AIChE

The Multimode Resource Constrained Multiproject Scheduling Problem: Alternative Formulations

Juan Camilo Zapata, Bri Mathias Hodge, and Gintaras V. Reklaitis
School of Chemical Engineering, Purdue University, West Lafayette, IN 47907

DOI 10.1002/aic.11522
Published online July 7, 2008 in Wiley InterScience (www.interscience.wiley.com).

Tactical management of development pipelines is concerned with the allocation of resources and scheduling of tasks. Though these decisions have to be made in the presence of uncertainty, to make the problem solvable it is customary to use deterministic MILP formulations of the multi-mode RCMPSP that are reevaluated after important uncertainties are realized. In spite of the major simplifications attained by down- playing the stochastic nature of the problem, the curse of dimensionality limits the exact solution of the formulations to very small systems. The curse is mainly caused by 3 factors: the indexing of the task execution modes the indexing of time periods, and the discrete character of the resources. Three models that attempt to overcome these limitations are proposed and compared. Results show that despite the theoretical advantages of the strategies used, the alternative formulations are limited to problems in the same range of applicability of the conventional multi-mode formulation. © 2008 American Institute of Chemical Engineers *AIChE J,* 54: 2101–2119, 2008

Keywords: organization, mathematical modeling, portfolio of new products, scheduling, continuous time formulations

Introduction

The multimode resource-constrained multiproject scheduling problem (multimode RCMPSP) is an extension of the conventional resource constrained scheduling problem in which the duration of each task is a function of the level and type of resources committed to it, and the project interactions resulting from the utilization of shared resources are taken into consideration.[1] The relevance of capturing the flexibility in the allocation of resources and the corresponding changes in the completion times can be illustrated with a highly idealized example. Consider a pharmaceutical company that received FDA approval for three different products. All the products have to go through three sequential pre-launch tasks (A1, A2, and A3), whose durations and resource requirements are presented in Table 1. It is assumed that there are three groups of renewable resources (R1, R2, and R3) that are completely specialized on each of the tasks. Specifically, RX can only be allocated to AX. The types of resources within each group are aggregated and its total availability is represented in terms of dollars (Table 2). For simplicity the objective function maximizes the total nondiscounted profit obtained from the three projects, assuming that the negative cash flow for each activity is equal to the resource requirements, and that the positive cash flows correspond to the net profit (Table 3) for the remaining patent life, 50 months for all three projects. Figure 1 shows the resulting schedules for the cases in which each task can only be assigned a single combination of resources (mode 1), and the more flexible one in which either mode 1 or mode 2 can be allocated. The consideration of multiple modes made the more resource intensive (and more profitable) project 1 more attractive by

Correspondence concerning this article should be addressed to G. V. Reklaitis at reklaiti@ecn.purdue.edu.

Table 1. Task Durations and Resource Requirements for the Motivating Example

Activities	Projects	Duration (months)		Resource requirements (M$)	
		Mode 1	Mode 2	Mode 1	Mode 2
A1	1	4	6	24	16
	2	4	6	18	12
	3	4	6	18	12
A2	1	8	12	30	20
	2	8	12	21	14
	3	8	12	15	10
A3	1	10	15	39	26
	2	10	15	36	24
	3	10	15	36	24

Table 2. Renewable Resources Availability

Resources	Availability (M$)
1	40
2	40
3	75

Table 3. Project Monthly Expected Net Profits After Market Introduction

Project	Returns (M$)
1	10
2	7
3	5

allowing the minimization of the reduction in the profits from the other two projects caused by the delay in their market introduction. The change in the schedule led to an increase in the portfolio profit from $279 million to $296 million.

This example shows that even without efficiency gains the ability to allocate resources in a flexible manner has a potential to enhance the value of the portfolio of projects, and therefore has to be part of the set of decision variables used to optimize the performance of the portfolio.

In the development of new products/services the decisions made at the tactical/operational level are mainly concerned with determining the optimal allocation of resources and scheduling the tasks required to complete each project in the pipeline. This optimization problem can be viewed basically as a multimode RCMPSP. The main difference is that technological and market uncertainties may cause the failure of some of the projects at different stages in the development process. These events force the remaining work scheduled on the terminated projects to be canceled, and the resources to be reallocated, resulting in schedules for the surviving projects that may be completely different from the ones planned before the failure.[2] The presence of success/failure, task duration, and resource requirement uncertainties make the underlying problem a stochastic optimization problem. However, the complexity of a multistage stochastic program that accounts for all the uncertainties and project interactions is outside the set of effectively solvable problems.[3,4] For this reason it is customary to generate a schedule by using deterministic MILP formulations that are reevaluated every time an important uncertainty is realized. In spite of the major simplifications attained in the formulations by ignoring or downplaying the stochastic nature of the problem, the curse of dimensionality still restricts their exact solution to very small systems. This limitation is mainly caused by the indexing of three factors: the indexing of the task execution modes, time periods, and the discrete character of the resources. The bulk of large-scale problems consider a single task execution mode or a very limited number of modes, and are solved by heuristics or tailored branch and bound algorithms.[5] In this work, we propose and evaluate three different formulations that attempt to provide exact solutions to the deterministic version of the tactical/operational decision making problem found in the development of new products when preemption is not allowed and the task sequence is predetermined. To achieve this goal three concepts are exploited, namely, continuously divisible resources, continuous time representations, and short time horizons.

The rest of the article is organized as follows. The next section provides a literature review. The section Conventional Multimode RCMPSP describes the conventional multimode RCMPSP, its limitations, and the reasoning behind the strategies used to overcome them. The problem is formalized in the section Problem Definition, and the corresponding formulations are presented in the section Mathematical Formulations. The section Example provides results and discussion for a specific case study. Concluding remarks and perspectives are presented in the last section.

Literature Review

The project scheduling literature treats the project scheduling problem with multiple task execution modes from three different perspectives, namely, time/cost trade-off, time/resource trade-off, and multimode resource allocation.[1,6] The time/cost trade-off strategy does not explicitly consider resource decisions; instead it uses a cost vs. duration function to describe the interaction between tasks duration and resource requirements. In the time/resource trade-off case it is assumed that a single resource is committed to each task, and that the duration of each task is a decreasing function of the allocated amount of the corresponding resource. The multimode problem is an extension of the other two strategies. It indexes different blends (modes) of resources to capture not just the time/cost and time/resource trade-offs, but also the resource/resource trade-offs. In practice, decision makers can tailor the group of resources assigned to undertake each task, increasing or decreasing the duration of the task accordingly. The resources can also be sourced internally or externally. In the first case the different options are made available by the multiple levels and fields of expertise present in the organization, while in the second case the options are determined by the outsourcing opportunities offered by the market. Clearly the multimode approach provides the most comprehensive framework for realistically representing the flexibilities that decision makers have. However, exact solution of this kind of problem requires customized branch and bound algorithms, even for relatively simple systems.[1] Only heuristics and metaheuristics can provide solution, though suboptimal, to mid and large sized problems.[1,5,7,8] Most of the efforts in the project scheduling literature have been on the algorithmic side. The process systems literature, on the other hand, has tried to leverage its experience with short-term batch scheduling to reformulate the problem. Initially, the reformulations were aimed

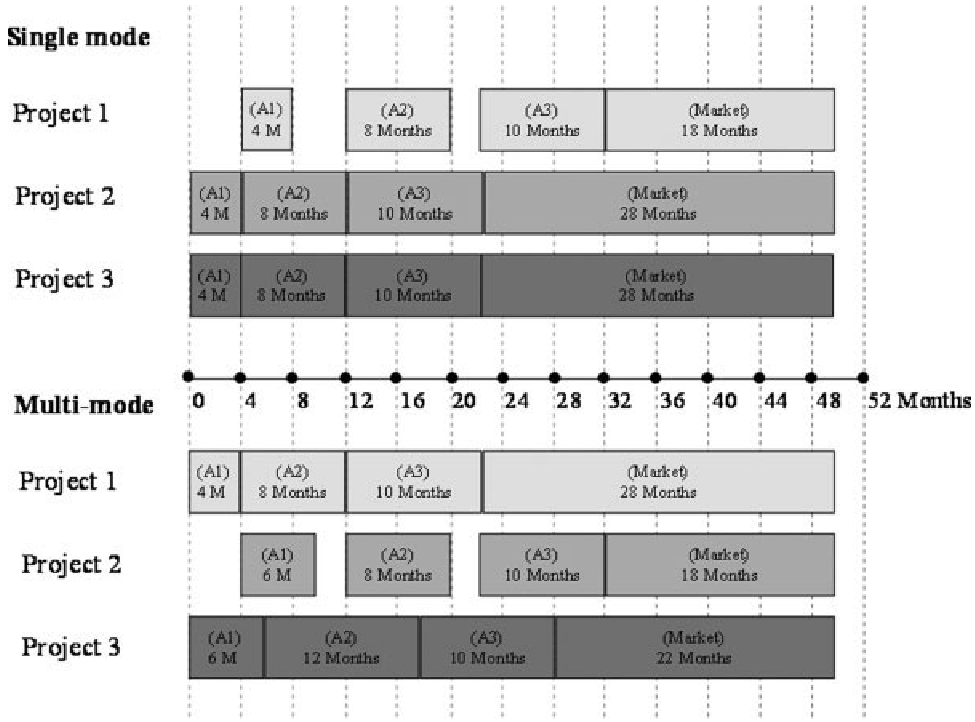

Figure 1. Schedules for the motivating example.

toward capturing the effects of the success/failure uncertainties by using the expectation operator in the objective function and some of the constraints. In this line of thought, Schmidt and Grossmann[9] proposed a single mode, slot-based continuous time formulation with unlimited resources, and Homkomp[10] proposed a resource constrained single mode discrete time formulation. In subsequent work, the focus changed to the development of realistic models that could be solved for large-scale problems. Jain and Grossmann[11] extended Schmidt's work by including resource constraints, and Maravelias and Grossmann[4] further developed that model by allowing the allocation of different levels of resources and capacity expansion. Subramanian et al.[3,12] proposed a "rolling horizon"-type strategy that combines short-term reactive scheduling with simulation. Under this strategy, Homkomp's deterministic RCMPSP for a short-term horizon is solved and its outcome is used to control a discrete event simulation of the system until the optimization has to be implemented again.

It is important to highlight that the similarities between project scheduling and short-term batch scheduling, first recognized by Schmidt and Grossman,[9] are exploited in this work. Specifically, sequence dependence[13] and time events[14] for continuous time representation are used in the proposed formulations.

Conventional Multimode RCMPSP

Formulation

In this version of the conventional formulation the modes typically represented by one index are split into two indexes, combinations, and multiples. Though this change does not have any impact on the number of variables or constraints, it helps to draw the parallel between the conventional model and the proposed models. In this formulation, every mode requires a predefined number of resources and results in a known duration for each task. Therefore, it is not necessary to add distinctive variables for the start and finish times. A single set of binary variables is enough to develop the model:

x_{pijmt} = 1 if task i of project p is completed at the beginning of time t with multiple m of combination j, 0 otherwise.

The complete formulation is given by Eqs. 1–5, where R_{pimjk} is the number of resources k allocated to task i of project p if multiple m of combination j is used, and P_{pimj} represents the duration of task i of project p if multiple m of combination j is used. Constraint (2) ensures that only one multiple of one combination is allocated to each task. Constraint (3) enforces the resource capacity limits, and constraints (4) and (5) guarantee that the precedence relations are satisfied.

$$\max \sum_{p} \sum_{i \in I_p} \sum_{m} \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} W_{pij}\, e^{-rt}\, x_{pimjt} \tag{1}$$

$$\sum_{m} \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} x_{pimjt} \leq 1 \quad \forall\, p, i \in I_p \cap I^{T}_{\lceil \mathrm{EFT} \rceil} \tag{2}$$

$$\sum_{p} \sum_{i \in I_p} \sum_{m} \sum_{j \in J_{pi}} R_{pimjk} \sum_{q=\max\{t+1, \lceil \mathrm{EFT}_{pij} \rceil\}}^{t+p_{pimj}} x_{pimjq} \leq R_k^{\max} \quad \forall k, t = 1, \ldots, T \tag{3}$$

$$\sum_{m} \sum_{w \in J_{ph}} \sum_{t \geq \lceil \mathrm{EFT}_{phw} \rceil} t x_{phmwt} - \sum_{m} \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} (t - p_{pimj}) x_{pimjt} \leq T \left(1 - \sum_{m} \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} x_{pimjt} \right) \quad \forall p, i \in I_p \cap I^{T}_{\lceil \mathrm{EFT} \rceil}, h \in H_{pi} \tag{4}$$

$$\sum_{m}\sum_{w\in J_{pi}}\sum_{t\geq\lceil \mathrm{EFT}_{pij}\rceil} x_{pimjt} - \sum_{m}\sum_{w\in J_{ph}}\sum_{t\geq\lceil \mathrm{EFT}_{phw}\rceil} x_{phmwt} \leq \left(1-\sum_{m}\sum_{j\in J_{pi}}\sum_{t\geq\lceil \mathrm{EFT}_{pij}\rceil} x_{pimjt}\right) \quad \forall p, i\in I_p \cap I^T_{\lceil \mathrm{EFT}\rceil},\ \ h\in H^*_{pi} \tag{5}$$

Weaknesses and strategies to address them

In the conventional formulation of the multimode RCMPSP, the indexing of the task execution modes results in a rapid increase in problem size with the number of modes. This scaling issue can be moderated by recognizing that a mode is an assembly of different types of resources (e.g. four of resource x, eight of resource y, and four of resource w) that can be characterized in terms of two factors. The first factor is the combination, represented by the resource makeup ratios (e.g. 25% resource x, 50% resource y, and 25% resource w) or any other set of values that represent the same relative amounts of resources (e.g. two of resource x, four of resource y, and two of resource w) and the second factor is the multiple of the combination (e.g. two times). The use of continuously divisible resources allows the representation of the different multiples of each combination by a single continuous variable, instead of a set of indexed binary variables as is done in the conventional formulation. This use of continuously divisible resources is a nonstandardized version of the intensity concept used in project scheduling problems with variable-intensity activities.[15,16] The index reduction is especially efficient when the number of combinations is small. For instance, if outsourcing and in-house development are the only possible combinations, the use of continuously divisible resources allows the development of a formulation in which the sizes of the variable sets are increased by a factor of 2. This gives a very modest increase in the problem size compared with the $2 \times M$ factor observed in a conventional multimode strategy, where M represents the number of multiples taken into consideration for each of the two possible combinations. The price of implementing this problem size reduction strategy is the need to use a linear approximation for the relationship between task durations and the multiples of each combination of resources. In addition, if the resources are not allowed to be used in multiple projects at the same time, the fractional outcome of the solution needs to be rounded. Though it is well known that such a policy may lead to suboptimal solutions, the significant number of resources in a large-scale problem decreases the policy's negative impact.

The second aspect that hinders the scaling of the conventional multimode RCMPSP formulation is the use of a discrete representation of the time domain. It is well known that such a strategy significantly limits the set of solvable problems, especially when the durations of tasks differ widely and/or time horizons are long.[17] This limitation is caused by the need to add a new indexed binary variable for each additional time period considered, resulting in exponential growth of the problem size from a solution perspective. The use of continuous time representations enables the development of formulations whose size is independent of the length of the time periods and the time horizon. However, as with the mode-related problem size reduction, a price has to be paid for removing the time index. In this case the formulations obtained tend to relax poorly due to the required use of big-M type constraints.

The standard approach when formulating a RCMPSP is to assume that the selected time horizon is long enough to allow the scheduling of all the tasks of every project in the pipeline. However, in large-scale applications long time horizons usually result in problems that are not solvable due to the substantial number of variables that need to be considered. A simple way around this difficulty is to develop a short-term schedule and implement it until a significant deviation in the real system is encountered, due to the realization of uncertainties, and a new schedule needs to be computed. From a formulation perspective this means that the model has to be able to handle the case in which not all of the tasks are completed in the given time horizon, and to allow tasks to start that will not finish within such a time period. It should be noted that the assumption behind this strategy is that the extra complexities introduced into the model to allow these additional degrees of freedom are overridden by the gains of solving a smaller problem. Also notice that this kind of rolling horizon strategy may tend to be myopic, and therefore it requires that the decisions weigh in some way the long-term effects.

Before moving to the formulations proposed, it is important to highlight that the work of Maravelias and Grossmann[4] is a crucial step forward when compared with the conventional formulation. It uses a continuous representation of the time domain and allows the duration of each task to vary according to the number of resource allocated to it. However, the different types of resources that can be allocated to a task have the same impact on the duration, and when multiple resources are allocate the combined effect is additive. In addition, every additional unit of each resource type requires an extra binary indexed variable. The first limitation does not make it possible to represent the allocation of resources in a realistic manner, and the second one restricts the model from being used for systems with a significant number of resources.

Problem Definition

The goal of the problem is to determine which tasks have to be scheduled in the given time horizon, their sequence, their timing, and the types and levels of resources that need to be allocated to them. To be able to solve this problem it is assumed that the following information is given: (a) a collection of projects, $p \in P$, with known expected returns; (b) a multistage nonpreemptive flowshop development plan for each project, I_p, in which the probability of success at the end of each task is specified; (c) a set of renewable resources, $k \in K$, in limited supply; (d) sets of combinations of resources, $j \in J_{pi}$, suitable to complete the tasks in each project; (e) a set of factors that relates the duration of each task and the level of the combination allocated, β; (f) the minimum, $L^{\min}$, and maximum, $L^{\max}$, combination multiples that can be allocated to each task based on the values used to represent the resource makeup ratios; (g) the maximum or minimum task durations when completed by a specific combination; and (h) a scheduling horizon, T.

*As proposed by one of the referees an alternative to this constraint is $\sum_m \sum_{j\in J_{pi}} \sum_{t\geq\lceil \mathrm{EFT}_{pi}\rceil} x_{pimjt} \leq \sum_m \sum_{w\in J_{ph}} \sum_{t\geq\lceil \mathrm{EFT}_{phw}\rceil} x_{phmwt}$.

Mathematical Formulations

To begin, two formulations that attempt to address the problem of obtaining exact solutions for the multimode RCMPSP by using continuous time representations, continuously divisible resources and short-term horizons were developed. Anticipating the characteristic poor relaxations of models based on continuous time representations, a third formulation based on a discrete time representation was also conceived.

Discrete time formulation

To guarantee that at every point in time resource and precedence constraints are not violated, it is necessary to keep track of when tasks are started and completed, the multiples of the combinations used to undertake them, and the cumulative resource usages. For this purpose, two sets of binary variables and three sets of continuous variables need to be defined:

$x^{\mathrm{s}}_{pijt} = 1$ if task i of project p is started at the beginning of time t with combination *j, 0* otherwise.

$x^{\mathrm{f}}_{pijt} = 1$ if task i of project p is finished at the beginning of time t with combination *j, 0* otherwise.

L^{s}_{pijt}: Multiple of combination j engaged at time t to undertake task i of project p.

L^{f}_{pijt}: Multiple of combination j disengaged at time t from task i of project p.

R_{kt}: Total amount of resources k that are being used at time t.

Notice that in spite of the continuous character of L^{s}_{pijt} and L^{f}_{pijt}, they are implicitly discretized by the discrete time representation. Therefore, if the resource constraints are binding, the combination multiples, L^{s}_{pijt} and L^{f}_{pijt}, only can take those values that lead to integer completion times. On the other hand, if the resource constraints are not binding the combination multiples will take any value that allows the completion of the task before the next scheduled task starts at an integer time.

Assignment Constraints. Only one combination can be assigned to a task if it is started (finished) during the given time horizon:

$$\sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} x^{\mathrm{s}}_{pijt} \leq 1 \quad \forall\, p, i \in I_p \cap I^T_{\lceil \mathrm{EST} \rceil} \tag{6}$$

$$\sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} x^{\mathrm{f}}_{pijt} \leq 1 \quad \forall\, p, i \in I_p \cap I^T_{\lceil \mathrm{EFT} \rceil} \tag{7}$$

To reduce the size of the problem, only those tasks whose earliest start time (EST) and earliest finish time (EFT) are within the time horizon are included. Notice that the use of a discrete time representation requires the EST and EFT values to be rounded up to the closest integer of the time unit chosen.

Time and Combination Multiple Constraints. Constraint (8) ensures the appropriate duration of each task given the multiple of the combination assigned to it. The third term on the left-hand side represents the duration of a given task when the minimum level of resources is allocated plus $\beta_{pij} L^{\min}_{pij}$, while the fourth term corresponds to the reduction in the task duration caused by the allocation of resources above the minimum level. Constraints (9) and (10) guarantee that the combination multiple allocated to a task when started is deallocated when completed. Note that constraints (8) and (9) are big-M constraints with the time horizon, T, and the maximum multiple of a particular combination, $L^{\max}_{pij}$, as the M factors. Constraints (11) and (12) impose minimum and maximum bounds on the multiples of each combination.

$$-\sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} t x^{\mathrm{f}}_{pijt} + \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} t x^{\mathrm{s}}_{pijt} + \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} \alpha_{pij} x^{\mathrm{s}}_{pijt} - \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} \beta_{pij} L^{\mathrm{s}}_{pijt} \leq T \left(1 - \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} x^{\mathrm{f}}_{pijt} \right) \quad \forall\, p, i \in I_p, j \in J^T_{\lceil \mathrm{EFT}_{pi} \rceil} \tag{8}$$

$$\sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} L^{\mathrm{s}}_{pijt} - \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} L^{\mathrm{f}}_{pijt} \leq L^{\max}_{pij} \left(1 - \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} x^{\mathrm{f}}_{pijt} \right) \quad \forall\, p, i \in I_p, j \in J^T_{\lceil \mathrm{EFT}_{pi} \rceil} \tag{9}$$

$$\sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} L^{\mathrm{s}}_{pijt} - \sum_{t \geq \lceil \mathrm{EFT}_{pij} \rceil} L^{\mathrm{f}}_{pijt} \geq 0 \quad \forall\, p, i \in I_p, j \in J^T_{\lceil \mathrm{EFT}_{pi} \rceil} \tag{10}$$

$$L^{\min}_{pijt} x^{\mathrm{s}}_{pijt} \leq L^{\mathrm{s}}_{pijt} \leq L^{\max}_{pijt} x^{\mathrm{s}}_{pijt} \quad \forall\, p, i \in I_p, j \in J_{pi}, t \in [\lceil \mathrm{EST}_{pi} \rceil, T] \tag{11}$$

$$L^{\min}_{pijt} x^{\mathrm{f}}_{pijt} \leq L^{\mathrm{f}}_{pijt} \leq L^{\max}_{pijt} x^{\mathrm{f}}_{pijt} \quad \forall\, p, i \in I_p, j \in J_{pi}, t \in [\lceil \mathrm{EFT}_{pij} \rceil, T] \tag{12}$$

Resource Constraints. Constraint (13) corresponds to the "mass balance" of each resource. The number of resources k allocated or deallocated is given by the product of the combination multiple (L^{s}_{pijt} or L^{f}_{pijt}), and the combination ratios or factors (ρ_{pijk}). Constraint (14) models the resource capacity limits, which, owing to the discrete nature of the time representation, can be easily varied in time to capture any future expansions or contractions.

$$R_{kt} = R_{k(t-1)} + \sum_p \sum_{i \in I_p \cap I^t_{\mathrm{EST}}} \sum_{j \in J_{pi}} \rho_{pijk} L^{\mathrm{s}}_{pijt} - \sum_p \sum_{i \in I_p \cap I^t_{\mathrm{EFT}}} \sum_{j \in J_{pi}} \rho_{pijk} L^{\mathrm{f}}_{pijt} \quad \forall\, t, k \tag{13}$$

$$R^{\max}_{kt} \geq R_{kt} \geq 0 \quad \forall\, k, t \tag{14}$$

Precedence Constraints. Constraint (15) enforces the completion of preceding tasks before a given task can be started. Not all the tasks can be started and finished in a short-term horizon, and therefore constraint (16) has to be added to prevent constraint (15) from being satisfied with $x^{\mathrm{f}}_{phwt} = 0$

$$\sum_{w \in J_{ph}} \sum_{t \geq \lceil \mathrm{EFT}_{phw} \rceil} t x^{\mathrm{f}}_{phwt} - \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} t x^{\mathrm{s}}_{pijt} \leq T \left(1 - \sum_{j \in J_{pi}} \sum_{t \geq \lceil \mathrm{EST}_{pi} \rceil} x^{\mathrm{s}}_{pijt} \right) \quad \forall\, p, i \in I_p \cap I^T_{\lceil \mathrm{EST} \rceil}, h \in H_{pi} \tag{15}$$

$$\sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EST}_{pi} \rceil} x^{s}_{pijt} - \sum_{w \in J_{ph}} \sum_{t \geq \lceil \text{EFT}_{phw} \rceil} x^{f}_{phwt} \leq 1 - \sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EST}_{pi} \rceil} x^{s}_{pijt} \quad \forall\, p, i \in I_p \cap I^{T}_{\lceil \text{EST} \rceil}, h \in H^{*}_{pi} \tag{16}$$

Tightening Constraints. It was found that in the relaxed problem tasks could be finished at time 0 without violating the task duration constraint (8). To avoid this situation, (17) and (18) were added to the formulation. These constraints enforce a lower bound in the finish time of the tasks, which corresponds to the case when the resources that can complete them in the minimum possible time are allocated.

$$\sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} t x^{f}_{pijt} \geq \sum_{w \in J_{ph}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} t x^{f}_{phwt} + \sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} \alpha_{pij} x^{f}_{pijt} - \max[\beta_{pij} L^{\max}_{pij}] - T \left(1 - \sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} x^{f}_{pijt} \right) \quad \forall p, i \in I_p \cap I^{T}_{\lceil \text{EFT} \rceil}, \quad h \in H_{pi} \tag{17}$$

$$\sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} t x^{f}_{pijt} \geq \sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} \alpha_{pij} x^{f}_{pijt} - \max[\beta_{pij} L^{\max}_{pij}] \quad \forall\, p, i \in I_p \cap I^{T}_{\lceil \text{EFT} \rceil} \tag{18}$$

Objective Function. From a financial perspective the decisions on how to manage a portfolio of projects need to be geared toward the maximization of the expected net present value (ENPV). However, the implementation of a short-term rolling horizon strategy does not allow the use of an objective function that computes such a quantity. Therefore, taking into account that the internal resources in an R&D organization are committed regardless of the project they are assigned to, and that external resources are limited by a budget, expected annual returns are used as weighting factors, W_{pij}, in the objective function. The expected values are calculated, assuming statistical independence between the tasks, by multiplying the forecasted annual returns for each project and the probabilities of success of the given task, the tasks down stream of it and the tasks that can take place in parallel with it. In addition, exponential discounting, with rate r, is used to capture the time value of money according to the market (e.g. importance of first mover advantage, patent protection, competitor's position) and technological risks perceived for each project.

Though this objective function is only based on variables that mark the completion of the different tasks, it can be extended to include terms characterized by variables that mark their starting points. This "simplified" objective was selected to allow for a more significant numerical comparison between the different formulations as the precedence based continuous time formulation is limited to the use of completion variables in the objective function, and the conventional formulation is limited to the use of a single type of variable (i.e. start or completion).

$$\max \sum_{p} \sum_{i \in I_p} \sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EFT}_{pij} \rceil} W_{pij}\ e^{-rt}\ x^{f}_{pijt} \tag{19}$$

*As proposed by one of the referees an alternative to this constraint is $\sum_{j \in J_{pi}} \sum_{t \geq \lceil \text{EST}_{pi} \rceil} x^{s}_{pijt} \leq \sum_{w \in J_{ph}} \sum_{t \geq \lceil \text{EFT}_{phw} \rceil} x^{f}_{phwt}$.

Summarizing, the discrete time formulation of the problem, M1, consists of Eq. 19 as the objective function and Eqs. 6–18 as the set of constraints.

Precedence-based continuous time formulation

The first continuous time formulation is based on the ideas on precedence and concurrence developed by Mendez et al.[13] This strategy, instead of keeping track of the resources being used at each point in time, determines which activities take place in parallel, calculates the total resource requirements of such activities and forces these requirements to be less than the capacities available. For this purpose two sets of binary variables and two sets of continuous variables need to be defined:

$x_{pij}=1$ if task i of project p is completed with combination j, 0 otherwise.

$Y_{pip'i'}=1$ if task i' of project p' has been completed after starting task i of project p, 0 otherwise.

C_{pi}: Completion time of task i of project p.

L_{pij}: Multiple of combination j used to complete task i of project p.

The most attractive aspect of this formulation is that it is completely time independent. However, the problem size reduction gains obtained in the time dimension are offset to some degree by the additional variables required to linearize the resource availability constraints and the objective function.

Assignment Constraints. Constraint (20) guarantees that only one combination is assigned to each task.

$$\sum_{j \in J_{pi}} x_{pij} \leq 1 \quad \forall\, p, i \in I_p \cap I^{T}_{\text{EFT}} \tag{20}$$

Combination Multiple Constraints. In this formulation, the variables are completely time independent, which means that no synchronization between start and finish variables has to be implemented. Therefore, only bounds have to be imposed on the multiples of each combination:

$$L^{\min}_{pij} x_{pij} \leq L_{pij} \leq L^{\max}_{pij} x_{pij} \quad \forall\, p, i \in I_p \cap I^{T}_{\text{EFT}}, j \in J_{pi} \tag{21}$$

Resource Constraints. The resource constraints require the determination of whether tasks that are allocated combinations that include a given resource take place in parallel. This determination is performed via pair wise comparisons that relate the starting and finishing points of two tasks. Based on the binary variables $Y_{pip'i'}$, that take a value equal to 1 if task i' of project p' is completed after starting task i of project p, it is possible to determine if tasks i and i' are simultaneously executed, at least to some extent. Specifically, if $Y_{pip'i'} + Y_{p'i'pi}$ is equal to 2 there is some overlap; if the sum is equal to 1 this indicates that the tasks are in some kind of sequence; and a result equal to zero implies that one or both of the tasks were not scheduled. Constraint (22) considers every task [in the same project (I_{pip}) or in the other projects in the pipeline ($I_{p'}$)] that according to each project plan has the potential of taking place in parallel with task i and assigns the corresponding values to $Y_{pip'i'}$. Constraint (23) uses such values, the combination multiples, L_{pij}, and ρ_{pijk} (combination ratios or factors) to enforce the resource constraints. This constraint exhibits bilinear terms ($L_{p'i'j'} Y_{pip'i'}$ and $L_{p'i'j'} Y_{p'i'\text{pi}}$) that following Glover[18] can be eliminated by replacing them with the linearization variables $Z^{d}_{pip'i'j'}$ and

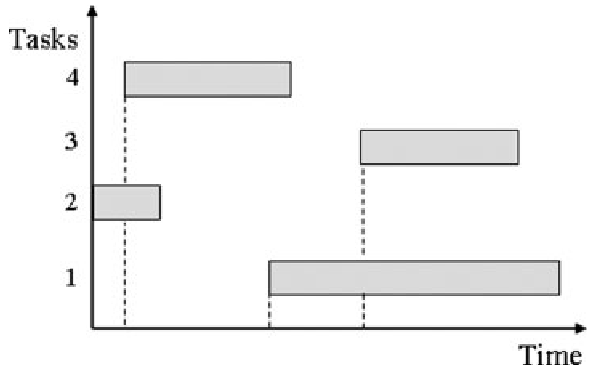

Figure 2. Schedule of tasks for project.

$Z^r_{pip'i'j'}$, resulting in constraint (24), and by appending constraints (25)–(28) to the formulation.

$$C_{p'i'} - C_{pi} + \sum_{j \in J_{pi}} [\alpha_{pij} x_{pij} - \beta_{pij} L_{pij}] + \leq T y_{pip'i'} + T\left(2 - \sum_{j \in J_{pi}} x_{pij} - \sum_{j \in J_{p'i'}} x_{p'i'j}\right)$$
$$\forall\, p, i \in I_p \cap I^T_{\text{EFT}}, p', i' \in (I_{pip} \vee I_{p'}) \cap I^T_{\text{EFT}} \quad (22)$$

$$\sum_{j \in J_{pi}} \rho_{pijk} L_{pij} + \sum_{p'} \sum_{i' \in (I_{p'} \cap I^T_{\text{EFT}}) \vee (I_{pip} \cap I^T_{\text{EFT}})} \sum_{j' \in J_{p'i'}} \rho_{p'i'j'k} L_{p'i'j'} \times (y_{pip'i'} + y_{p'i'pi} - 1) \leq R^{\max}_k \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, k \quad (23)$$

$$\sum_{j \in J_{pi}} \rho_{pijk} L_{pij} + \sum_{p'} \sum_{i' \in (I_{p'} \cap I^T_{\text{EFT}}) \vee (I_{pip} \cap I^T_{\text{EFT}})} \sum_{j' \in J_{p'i'}} \rho_{p'i'j'k} \times (z^d_{pip'i'j'} + z^r_{pip'i'j'} - L_{p'i'j'}) \leq R^{\max}_k \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, k \quad (24)$$

$$L^{\max}_{p'i'j'} y_{pip'i'} \geq z^d_{pip'i'j'} \geq 0 \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, p', i' \in (I_p, \vee I_{pip}) \cap I^T_{\text{EFT}}, j' \in J_{p'i'} \quad (25)$$

$$L_{p'i'j'} \geq z^d_{pip'i'j'} \geq L_{p'i'j'} - L^{\max}_{p'i'j'}(1 - y_{pip'i'}) \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, p', i' \in (I_p, \vee I_{pip}) \cap I^T_{\text{EFT}}, j' \in J_{p'i'} \quad (26)$$

$$L^{\max}_{p'i'j'} y_{p'i'pi} \geq z^r_{pip'i'j'} \geq 0 \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, p', i' \in (I_p, \vee I_{pip}) \cap I^T_{\text{EFT}}, j' \in J_{p'i'} \quad (27)$$

$$L_{p'i'j'} \geq z^r_{pip'i'j'} \geq L_{p'i'j'} - L^{\max}_{p'i'j'}(1 - y_{p'i'pi}) \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, p', i' \in (I_p, \vee I_{pip}) \cap I^T_{\text{EFT}}, j' \in J_{p'i'} \quad (28)$$

It is relevant to mention that the pairwise nature of the comparisons used in this formulation to determine which tasks take place in parallel leads to conservative resource availability constraints. This arises from the fact that when a task overlaps with two or more tasks that do not overlap between them constraint (24) interprets such a condition as if all the tasks were overlapping. This situation can be better explained with a simple example. Assume that there is a project p in the pipeline that has to go through four tasks. The four tasks can take place in parallel, and are the only ones that require resource k in order to be completed. Also, assume that there is only one combination suitable for each, with $\rho_{pi1k}=1$, and that the schedule generated by the optimization is the one presented in Figure 2. In that case, the optimizer arrived to the solution by enforcing the following constraints for resource k [constraint (23)]:

$$L_{p1k} + L_{p3k} + L_{p4k} \leq R^{\max}_k$$
$$L_{p2k} + L_{p4k} \leq R^{\max}_k$$
$$L_{p3k} + L_{p1k} \leq R^{\max}_k$$
$$L_{p4k} + L_{p2k} \leq R^{\max}_k$$

where the first constraint is conservative because tasks 4 and 3 are not taking place in parallel.

Precedence and Completion Time Constraints. Constraint (29) enforces the completion of preceding tasks before a given task can be started. Notice that if a task does not have any precedence, (29) still holds with $C_{ph}=0$. Constraint (30) has to be added to prevent (29) from being satisfied in that case. To enforce the time horizon constraint (31) was initially included, but it was dropped in favor of (32), which provides a tighter relaxation.

$$C_{ph} - C_{pi} + \sum_{j \in J_{pi}} [\alpha_{pij} x_{pij} - \beta_{pij} L_{pij}] \leq T\left(1 - \sum_{j \in J_{pi}} x_{pij}\right) \forall p, i \in I_p \cap I^T_{\text{EFT}}, h \in H_{pi} \quad (29)$$

$$\sum_{j \in J_{pi}} x_{pij} - \sum_{w \in J_{ph}} x_{phw} \leq 1 - \sum_{j \in J_{pi}} x_{pij} \quad \forall p, i \in I_p \cap I^T_{\text{EFT}}, \ h \in H_{pi} \quad (30)$$

$$C_{pi} \leq T \quad \forall\, p, i \in I_p \cap I^T_{\text{EFT}} \quad (31)$$

$$C_{pi} \leq T \sum_{j \in J_{pi}} x_{pij} \quad \forall\, p, i \in I_p \cap I^T_{\text{EFT}} \quad (32)$$

Tightening Constraints. A similar behavior as the one described in the discrete time formulation was observed in this model. In the relaxed problem tasks were completed at time 0 without violating constraint (29). Initially, constraints (33) and (34) were included to enforce the completion of each task at least in the minimum possible time. However, it was observed that the net effect of these tightening constraints was negative. Specifically, the increase in the solution time of the relaxed problem was more significant than the tightening of the relaxed feasible region.

$$C_{pi} \geq C_{ph} + \sum_{j \in J_{pi}} \alpha_{pij} x_{pij} - \max[\beta_{pij} L^{\max}_{pij}] - M_3\left(1 - \sum_{j \in J_{pi}} x_{pij}\right) \quad \forall\, p, i \in I_p \cap I^T_{\text{EFT}}, h \in H_{pi} \quad (33)$$

$$C_{pi} \geq \sum_{j \in J_{pi}} \alpha_{pij} x_{pij} - \max[\beta_{pij} L^{\max}_{pij}] \quad \forall\, p, i \in I_p \cap I^T_{\text{EFT}} \quad (34)$$

Objective Function. The objective function has the same structure as the one used in the discrete time formulation. However, the combination of the exponential function used for discounting and the continuous time representation makes it nonlinear. Schmidt and Grossman[9] showed that to avoid solving a MINLP this kind of function can be linearized without introducing significant error. The approximation uses a piecewise linear function with G segments between the grid points a_{pig}, transforming (35) into (36) and forcing the definition of two additional sets of continuous variables and one set of binary variables:

λ_{pijg}: Weight between 0 and 1 of the grid point g in the convex combination used to approximate the term $e^{-rC_{pi}}\ x_{pij}$.

S_{pij}: Slack variable used in the piecewise linearization of the objective function for task i of project p when completed with combination j.

$v_{pijg}=1$ if the grid point g is part of the convex combination used to approximate the term $e^{-rC_{pi}}\ x_{pij}$.

The linearization also requires the addition of a new set of constraints to the formulation. Constraint (37) calculates the linearization weights, while constraint (38) guarantees that the condition $\sum_g \lambda_{pijg} = 1$ for a convex combination is only enforced when a task is completed with a given combination j. Constraint (38) also guarantees that the weights are 0 if the task is not completed or a different combination is used to complete it. Constraint (39) forces the slack variable to be 0 when the combination j is assigned to task i of project. Constraint (40) requires the weights to be 0 if the corresponding grid point is not part of the convex combination, and constraints (41) and (42) ensure that only two consecutive points in the grid are part of the convex combination.

$$\max \sum_p \sum_{i\in I_p\cap I^T_{\mathrm{EFT}}} \sum_{j\in J_{pi}} W_{pij}\ e^{-rC_{pi}}\ x_{pij} \tag{35}$$

$$\max \sum_p \sum_{i\in I_p\cap I^T_{\mathrm{EFT}}} \sum_{j\in J_{pi}} W_{pij} \sum_g \lambda_{pijg}\ e^{a_{pig}} \tag{36}$$

$$\sum_g \lambda_{pijg} a_{pig} - s_{pij} = -rC_{pi} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi} \tag{37}$$

$$\sum_g \lambda_{pijg} = x_{pij} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi} \tag{38}$$

$$s_{pij} \le a_{piG}(1 - x_{pij}) \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi} \tag{39}$$

$$\lambda_{pijg} \le v_{pijg} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi}, g \tag{40}$$

$$\sum_g v_{pijg} = 2x_{pij} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi} \tag{41}$$

$$v_{pijg-1} + v_{pijg+1} \ge v_{pijg} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi}, g \tag{42}$$

Summarizing, the precedence based continuous time formulation of the problem, M2, consists of Eq. 36 as the objective function and Eqs. 20–22, 24–30, 32, and 37–42 as the set of constraints.

Continuous time formulation based on events

In this formulation, the time horizon is divided into intervals whose start and end points are dictated by the realization of events, specifically the start of each task (event points).[19] It follows the same idea used in the discrete time formulation, namely, it keeps track of (a) when tasks are started and completed, (b) the multiples of the combinations used to complete each task, and (c) the cumulative resource usage. However, the fact that the duration of time periods are unknown makes it more difficult to signal the end of task and the consequent release of the resources. Therefore, in addition to the start and finish variables, it is necessary to include "in process" variables. This additional complexity is reflected in the number of sets of variables required by the formulation, three sets of binary variables and six sets of continuous variables:

$x^{\mathrm{s}}_{pijn}=1$ if task i of project p is started at event point n with combination *j, 0* otherwise.

$x^{\mathrm{f}}_{pijn}=1$ if task i of project p is finished at or before event point n with combination *j, 0* otherwise.

x^{p}_{pijn} if task i of project p is being processed at event point n with combination j, 0 otherwise.

L^{s}_{pijn}: Multiple of combination j engaged at event point n to undertake task i of project p.

L^{f}_{pijn}: Multiple of combination j disengaged at or before event point n from task i of project.

L^{p}_{pijn}: Multiple of combination j being used at event point n to undertake task i of project.

R_{kn}: Total amount of resource k that is being used at event point n.

T^{f}_{pin}: Finish time of task i of project p (it is 0 for all the event points before the one in which it is started and takes the value of the finish time of the given task for all the remaining event points).

T_n: Time that corresponds to event point n.

Notice that the consideration of task starts as the only event determining event points requires the definition of T^{f}_{pin} to keep track of when tasks are finished, and allows the derivation of the starting time from T_n and x^{s}_{pijn}, making the definition of T^{s}_{pin} unnecessary.

In the following development, this version of the model will be initially formulated as a hybrid (generalized disjunctive program)/(mixed integer linear program) (GDP/MILP)[14,20] and later transformed into an MILP.

Assignment Constraints. Only one combination can be assigned to a task if it is started (finished) during the given time horizon.

$$\sum_{j\in J_{pi}} \sum_n x^{\mathrm{s}}_{pijn} \le 1 \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EST}} \tag{43}$$

$$\sum_{j\in J_{pi}} \sum_n x^{\mathrm{f}}_{pijn} \le 1 \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EFT}} \tag{44}$$

Time and Combination Multiple Constraints. To better understand the origin of the constraints included in this section the disjunctions from which they were derived are first presented. There are three different disjunctive cases for each suitable combination and each task of each project, at each point in time: (i) a task starts at event point n with combination j (x^{s}_{pijn}), (ii) a task is in process at event point n with combination j (x^{p}_{pijn}), and (iii) a task finishes or is already finished at event point n with combination j (x^{f}_{pijn}). Notice that in order to generate the set of disjunctions we need to define three new sets of variables:

T^{s}_{pijn}: Start time of task i of project p when combination j is used to undertake it (it is 0 for all the event points but the one when the task is started).

T^{f}_{pijn}: Finish time of task i of project p when combination j is used to undertake it (it is 0 for all the event points before the one in which it is started and takes the value of the finish time of the given task for all the remaining event points).

D_{pijn}: Duration of task i of project p that starts at event point n using combination j (it is 0 for all the event points but the one when the task is started).

Disjunction (45) captures the logic related to the start of a task. It states that the finish time for a task i at event point n processed with combination j is dictated by the multiple of the combination, L^{s}_{pijn}. This multiple has to lie within the given bounds, and be equal to L^{p}_{pijn} if the task is in process in the next event point or to L^{f}_{pijn} if the task is finished at or before the next event point. On the other hand, if the task is not started the finish time, T^{f}_{pijn} is kept unchanged.

$$\begin{pmatrix} x^{s}_{pijn} \\ D_{pijn} = \alpha_{pij} - \beta_{pij} L^{s}_{pijn} \\ T^{f}_{pijn} = T^{s}_{pijn} + D_{pijn} \\ L^{s}_{pijn} = L^{p}_{pijn+1} + L^{f}_{pijn+1} \\ L^{\min}_{pij} \le L^{s}_{pijn} \le L^{\max}_{pij} \\ L^{p}_{pijn} = L^{f}_{pijn} = 0 \end{pmatrix} \vee \begin{pmatrix} \neg x^{s}_{pijn} \\ T^{s}_{pijn} = 0 \\ D_{pijn} = 0 \\ T^{f}_{pijn} = T^{f}_{pijn-1} \\ L^{s}_{pijn} = 0 \end{pmatrix} \tag{45}$$

Disjunction (46) captures the logic related to the processing of a task. If task i is in process with combination j, the finish time, T^{f}_{pijn}, is kept unchanged, and the multiple L^{p}_{pijn} has to lie within the given bounds, and be equal to the one in the previous period, regardless of whether the period was a start period (L^{s}_{pijn-1}) or a processing period (L^{p}_{pijn-1}). If the task is not in process, the finish time, T^{f}_{pijn}, increases if the task starts at event point n with combination j, and stays the same if the task does not start.

$$\begin{pmatrix} x^{p}_{pijn} \\ T^{f}_{pijn} = T^{f}_{pijn-1} \\ D_{pijn} = 0 \\ L^{p}_{pijn} = L^{s}_{pijn-1} + L^{p}_{pijn-1} \\ L^{\min}_{pij} \le L^{p}_{pijn} \le L^{\max}_{pij} \\ L^{f}_{pijn} = L^{s}_{pijn} = 0 \end{pmatrix} \vee \begin{pmatrix} \neg x^{p}_{pijn} \\ T^{f}_{pijn} \ge T^{f}_{pijn-1} \\ L^{p}_{pijn} = 0 \end{pmatrix} \tag{46}$$

Disjunction (47) captures the logic related to the completion of a task. If task i is finished at event point n or before with combination j, the finish time, T^{f}_{pijn}, has to be less than or equal to the start time of the interval, T_n. Also the multiple L^{f}_{pijn} has to lie within the given bounds and be equal to the one in the previous period, whether the period was a start period (L^{s}_{pijn-1}) or a processing period (L^{p}_{pijn-1}). If the task is not finished, the finish time, T^{f}_{pijn}, increases if the task starts at event point n with combination j, and stays the same if the task does not start or is in process.

$$\begin{pmatrix} x^{f}_{pijn} \\ T^{f}_{pijn} \le T_n \\ D_{pijn} = 0 \\ L^{f}_{pijn} = L^{s}_{pijn-1} + L^{p}_{pijn-1} \\ L^{\min}_{pij} \le L^{f}_{pijn} \le L^{\max}_{pij} \\ L^{p}_{pijn} = L^{s}_{pijn} = 0 \end{pmatrix} \vee \begin{pmatrix} \neg x^{f}_{pijn} \\ T^{f}_{pijn} \ge T^{f}_{pijn-1} \\ L^{f}_{pijn} = 0 \end{pmatrix} \tag{47}$$

To complete the description of the disjunctions it should be acknowledged that the use of binary variables to characterize them is technically an abuse of notation. A rigorous presentation would instead require the use of Boolean variables. The transformation of the disjunctions into MILP constraints was implemented via convex hull (Raman and Grossmann[20] and references hereafter) and big-M constraints, the first option being the primary strategy when both were suitable. As for the duration, start and finish times, constraints (48)–(52) are derived from disjunction (45), whereas constraint (53) derives from disjunction (47). The duration of each task is given by constraint (48). Constraints (49) and (50) ensure the equality for T^{f}_{pijn} when the task is started, while constraints (51) and (52) activate the corresponding equality when the task is not started. Notice that the condition for the finish time given by disjunction (46) is captured in constraints (51) and (52).

$$D_{pijn} = x^{s}_{pijn}\alpha_{pij} - \beta_{pij} L^{s}_{pijn} \tag{48}$$

$$T^{f}_{pijn} \le T^{s}_{pijn} + D_{pijn} + T(1 - x^{s}_{pijn}) \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}},\ j \in J_{pi}, n \tag{49}$$

$$T^{f}_{pijn} \ge T^{s}_{pijn} + D_{pijn} - T(1 - x^{s}_{pijn}) \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}},\ j \in J_{pi}, n \tag{50}$$

$$T^{f}_{pijn} \le T^{f}_{pijn-1} + T x^{s}_{pijn} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \tag{51}$$

$$T^{f}_{pijn} \ge T^{f}_{pijn-1} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \tag{52}$$

$$T^{f}_{pijn} \le T_n + T(1 - x^{f}_{pijn}) \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \tag{53}$$

In terms of the combination multiples, the use of the convex hull reformulation allows the determination of bounds for each case [constraints (54), (55), and (56)] and the constraints that propagate the value of the multiple allocated at the start of a task, L^{s}_{pijn}, to the event point at or before which the task is completed [constraints (57), (58), (59)].

$$L^{\min}_{pij} x^{s}_{pijn} \le L^{s}_{pijn} \le L^{\max}_{pij} x^{s}_{pijn} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \tag{54}$$

$$L^{\min}_{pij} x^{f}_{pijn} \le L^{f}_{pijn} \le L^{\max}_{pij} x^{f}_{pijn} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EFT}}, j \in J_{pi}, n \tag{55}$$

$$L^{\min}_{pij} x^{p}_{pijn} \le L^{p}_{pijn} \le L^{\max}_{pij} x^{p}_{pijn} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \tag{56}$$

$$L^{s}_{pijn} = L^{p}_{pijn-1} + L^{f}_{pijn+1} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \tag{57}$$

$$L^{p}_{pijn} = L^{s}_{pijn-1} + L^{p}_{pijn-1} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \tag{58}$$

$$L^{f}_{pijn} = L^{s}_{pijn-1} + L^{p}_{pijn-1} \quad \forall\, p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \tag{59}$$

The number of variables and constraints in the formulation can be significantly reduced by aggregating constraints. Notice that because of constraint (43) each of the tasks can be assigned to at most one of the combinations. This ensures that when a task is started, T^{f}_{pijn} will be different from 0 only for one combination, allowing the substitution of $\sum_{j \in J_{pi}} T^{f}_{pijn}$ by T^{f}_{pin}. In the same way, constraint (43) supports the substitution of $\sum_{j \in J_{pi}} T^{s}_{pijn}$ by T^{s}_{pin}. Summing over j and substituting (48) and $T^{s}_{pin} = T_n$, constraints (49) and (50) are aggregated into (60) and (61). By a similar process constraints (51), (52) and (53) are aggregated into (62), (63), and (64). In addition, according to the definition of T^{f}_{pin} if the task is started, the difference in its value with respect to the one in the previous event point is given by the duration of the task. Based on this observation the formulation can be tightened by replacing constraint (63) by (65).

$$T^{\mathrm{f}}_{pin} \leq T_n + \sum_{j \in J_{pi}} \alpha_{pij} x^{\mathrm{s}}_{pijn} - \sum_{j \in J_{pi}} \beta_{pij} L^{\mathrm{s}}_{pijn} + T\left(1 - \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn}\right) \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, n \quad (60)$$

$$T^{\mathrm{f}}_{pin} \geq T_n + \sum_{j \in J_{pi}} \alpha_{pij} x^{\mathrm{s}}_{pijn} - \sum_{j \in J_{pi}} \beta_{pij} L^{\mathrm{s}}_{pijn} + T\left(1 - \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn}\right) \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EST}}, n \quad (61)$$

$$T^{\mathrm{f}}_{pin} \leq T^{\mathrm{f}}_{pin-1} + T \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EST}}, n \backslash \{0\} \quad (62)$$

$$T^{\mathrm{f}}_{pin} \geq T^{\mathrm{f}}_{pin-1} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EST}}, n \backslash \{0\} \quad (63)$$

$$T^{\mathrm{f}}_{pin} \leq T_n + T\left(1 - \sum_{j \in J_{pi}} x^{\mathrm{f}}_{pijn}\right) \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, n \backslash \{0\} \quad (64)$$

$$T^{\mathrm{f}}_{pin} \geq T^{\mathrm{f}}_{pin-1} + \sum_{j \in J_{pi}} \alpha_{pij} x^{\mathrm{s}}_{pijn} - \sum_{j \in J_{pi}} \beta_{pij} L^{\mathrm{s}}_{pijn} \quad \forall\, p, i \in I_p \cap I^T_{\mathrm{EST}}, n \backslash \{0\} \quad (65)$$

In the case of the combination multiple constraints, simplifications can be attained by realizing that at every event point n, at most one of L^{p}_{pijn}, L^{f}_{pijn}, L^{s}_{pijn} can be nonzero. This allows the grouping of constraints (57), (58), and (59) into constraint (66). Finally, following Maravelias and Grossmann,[14] it is possible to eliminate x^{p}_{pijn} from (56) and consequently from the problem, resulting in constraint (67).

$$L^{\mathrm{p}}_{pijn} + L^{\mathrm{f}}_{pijn} = L^{\mathrm{s}}_{pijn-1} + L^{\mathrm{p}}_{pijn-1} \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, j \in J_{pi}, n \backslash \{0\} \quad (66)$$

$$L^{\min}_{pij}\left(\sum_{n'<n} x^{\mathrm{s}}_{pijn'} - \sum_{n' \leq n} x^{\mathrm{f}}_{pijn'}\right) \leq L^{\mathrm{p}}_{pijn} \leq L^{\max}_{pij}\left(\sum_{n'<n} x^{\mathrm{s}}_{pijn} - \sum_{n' \leq n} x^{\mathrm{f}}_{pijn'}\right) \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, j \in J_{pi}, n \quad (67)$$

Additional constraints are required to enforce the time horizon and the ordering of the event points. Constraint (68) guarantees that the last event point, N falls within the time horizon, while constraint (70) forces the event points to increase in a monotonic manner.

$$T_N \leq T \quad (68)$$

$$T_1 = 0 \quad (69)$$

$$T_{n+1} \geq T_n \quad \forall n \backslash \{N\} \quad (70)$$

Resource Constraints. The concept of "mass balance" for resources described in the discrete time formulation is also used in this formulation. However, notice that the unknown length of the time intervals only allows one to use a time invariant capacity, $R^{\max}_k$ and to disregard those tasks whose EST and EFT are greater than the time horizon, T (i.e. it is not possible to disregard specific intervals for variables that can be started or completed within the given time horizon).

$$R_{kn} = R_{kn-1} + \sum_p \sum_{i \in I_p \cap I^T_{\mathrm{EST}}} \sum_{j \in J_{pi}} \rho_{pijk} L^{\mathrm{s}}_{pijn} - \sum_p \sum_{i \in I_p \cap I^T_{\mathrm{EFT}}} \sum_{j \in J_{pi}} \rho_{pijk} L^{\mathrm{f}}_{pijn} \quad \forall\, k, n \quad (71)$$

$$R_{kn} \leq R^{\max}_k \quad \forall k, n \quad (72)$$

Precedence Constraints. Constraint (73) enforces the completion of preceding tasks before a given task can be started. Not all the tasks can be started and finished in a short-term horizon; therefore, constraint (74) has to be added to prevent (73) from being satisfied with $T^{\mathrm{f}}_{phn} = 0$.

$$T^{\mathrm{f}}_{phn} - T^{\mathrm{s}}_{pin} \leq T\left(1 - \sum_j x^{\mathrm{s}}_{pijn}\right) \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, h \in H_{pi}, n \quad (73)$$

$$\sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn} - \sum_{w \in J_{ph}} \sum_{n' \leq n} x^{\mathrm{f}}_{phwn'} \leq 1 - \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn} \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, h \in H_{pi}, n \quad (74)$$

Tightening Constraints. No task can be in process or be finished at time 0, which allows the inclusion of (75) and (76) to fix the corresponding multiples of all the combinations. Initially, only constraints (77)–(79) were considered as a way to prevent the completion of tasks at time 0 in the relaxed problem without the need to use big M-constraints. However, it was found that the formulation tends to exhibit better performance when (80) and (81) are also included. Finally, (82) guarantees that in the relaxed problem x^{p}_{pijn} is within the [0, 1] range.

$$L^{\mathrm{p}}_{pij0} = 0 \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, j \in J_{pi} \quad (75)$$

$$L^{\mathrm{f}}_{pij0} = 0 \quad \forall p, i \in I_p \cap I^T_{\mathrm{EFT}}, j \in J_{pi} \quad (76)$$

$$T^{\mathrm{s}}_{pin} \geq EST_{pi} \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn} \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, n \quad (77)$$

$$T^{\mathrm{f}}_{pin} \geq EFT_{pi} \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn} \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}}, n \quad (78)$$

$$T^{\mathrm{f}}_{pin} \geq EFT_{pi} \sum_{j \in J_{pi}} x^{\mathrm{f}}_{pijn} \quad \forall p, i \in I_p \cap I^T_{\mathrm{EFT}}, n \quad (79)$$

$$T^{\mathrm{f}}_{pin} \geq T^{\mathrm{f}}_{phn} + \sum_{j \in J_{pi}} \alpha_{pij} x^{\mathrm{s}}_{pijn} - \max[\beta_{pij} L^{\max}_{pij}] - T\left(1 - \sum_{j \in J_{pi}} x^{\mathrm{s}}_{pijn}\right) \quad \forall p, i \in I_p \cap I^T_{\mathrm{EST}},\ h \in H_{pi}, n \quad (80)$$

$$T^{f}_{pin} \geq \sum_{j \in J_{pi}} \alpha_{pij} x^{s}_{pijn} - \max[\beta_{pij} L^{\max}_{pij}] \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, h \in H_{pi}, n \quad (81)$$

$$1 \geq \sum_{n' < n} x^{s}_{pijn'} - \sum_{n' \leq n} x^{f}_{pijn'} \geq 0 \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}},\ j \in J_{pi}, n \quad (82)$$

Objective Function. The objective function (83) has the same form as the one used in the two previous formulations, and, as in the precedence based model, needs to be linearized. Notice that the only differences compared with (35) lies is the use of T^{f}_{pin} and x^{s}_{pijn} in the places corresponding to C_{pi} and x_{pij} and the additional summation over n. From that perspective the derivation of the linearized objective function and required constraints (84)–(90) follows the same logic described for the precedence-based continuous time formulation.

$$\max \sum_{p} \sum_{i \in I_p \cap I^{T}_{\mathrm{EST}}} \sum_{j \in J_{pi}} \sum_{n} W_{pij}\ e^{-rT^{f}_{pin}} x^{s}_{pijn} \quad (83)$$

$$\max \sum_{p} \sum_{i \in I_p \cap I^{T}_{\mathrm{EST}}} \sum_{j \in J_{pi}} W_{pij} \sum_{g} \lambda_{pijg}\ e^{a_{pig}} \quad (84)$$

$$\sum_{g} \lambda_{pijg} a_{pig} - s_{pij} = -rT^{f}_{piN} \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi} \quad (85)$$

$$\sum_{g} \lambda_{pijg} = \sum_{n} x^{s}_{pijn} \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi} \quad (86)$$

$$s_{pij} \leq a_{piG} \left(1 - \sum_{n} x^{s}_{pijn} \right) \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi} \quad (87)$$

$$\lambda_{pijg} \leq v_{pijg} \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, g \quad (88)$$

$$\sum_{g} v_{pijg} = 2 \sum_{n} x^{s}_{pijn} \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi} \quad (89)$$

$$v_{pijg-1} + v_{pijg+1} \geq v_{pijg} \quad \forall p, i \in I_p \cap I^{T}_{\mathrm{EST}}, j \in J_{pi}, g \quad (90)$$

Summarizing, the event-based continuous time formulation of the problem, M3, consists of Eq. 84 as the objective function and Eqs. 43, 44, 54, 55, 60–62, 64–82, and 85–90 as the set of constraints. To conclude, it is important to highlight that the number of event points is a parameter of the formulation. Therefore, an iterative procedure, such as the one described by Maravelias and Grossmann,[14] is required. Specifically, they increase the number of event points by 1 until there is no improvement. Notice, however, that although this strategy is aimed at reducing the computational burden of solving the problem by implementing a structured search, it does not guarantee optimality.

Example

The formulations developed are evaluated with the four cases resulting from the two project portfolios and three project portfolio that can be constructed from a set of three projects (P1, P2, and P3). In each case two time horizons (two and three quarters) and two resource availability levels (100% and 40%) are considered. The 100% resource availability setting represents an environment in which resources are plentiful and the optimization is mainly driven by the way how combinations are allocated. On the other hand, the 40% availability setting creates a very tight environment in which both the combination amounts and the selection of the tasks to be scheduled are relevant. In each project all the tasks have the option of being completed in-house or outsourced. Each of these two combinations is a mix of the three types of resources (e.g. scientists, engineers, technicians) available in the corresponding setting (internal or external). Notice that for simplicity only the resources required for the actual implementation of the tasks were considered, which implicitly disregards the additional overhead involved in outsourcing. All the relevant information about projects and resources is provided in the Appendix.

The baseline for the evaluation of the formulations proposed is provided by an instance of the conventional formulation with five multiples for each of the two possible combinations (in-house and outsourced), for a total of 10 modes. This baseline formulation, M4, is solved for three different time buckets: 10, 5, and 1 days. All the models are implemented using ILOG's concert technology and solved with CPLEX 10.1 under the Unix environment in a Sun Enterprise 450 server with 300 MHz Ultra Sparc II processors and 4 GB in RAM.

To provide better insights into the behavior of each of the formulations a set of relative comparisons was implemented based on the results from the different portfolios and scenarios considered. The comparisons were divided in four groups: value of the objective function within each scenario, solution time within each scenario, effects of changes in the number of projects and timeline, and effects of changes in the availability of resources. Notice that for the sake of conciseness only pronounced and/or unexpected behaviors are presented and analyzed.

Objective function value

The most surprising outcome in this comparison was the lower values obtained from the discrete time formulations, M1, compared with the corresponding ones from M4 in Table 5. In theory, M1, disregarding small rounding effects should provide solutions at least as good as the ones from M4. However, the rounding effects turned out to be more significant than expected. In very pathological cases, as the ones observed in Table 5, the strategy used for rounding the mode durations in M4 may remove enough processing time from borderline activities to allow them to be completed within the given time horizon. It is important to highlight that this finding was completely accidental, as all the parameters for the projects were randomly generated.

The second relevant aspect in this category is the quality of the solutions exhibited by the 5 days M4 formulation [M4(5)] in many of the cases. They were basically indistinguishable from the ones obtained from M3 and the finer more computationally expensive 1 day M4 formulation [M4(1)]. However, as is clear from Table 6, significant differences, as expected, are possible and arise in unpredictable fashion. This implies that though it is tempting to use the conventional formulation with coarse time buckets, the inability to predict the impact in the quality of the solution limits its use.

Solution time

The difference in orders of magnitude in the solution time between M4 and the other three formulations is perhaps the most noticeable result obtained in this study. However, a closer look at the tables and the solutions paths shows that in several cases the advantage of M4 is much less significant. As was described in the previous comparison, in spite of the good solutions obtained in many cases by using a "coarse" time bucket (i.e. 5 days), 1 day is the ideal choice for the problem at hand. The problem is that the use of the 1 day time bucket considerably increases the solution time. This is especially true for very tight problems, as it can be seen in Table 7. In addition, some of the results in Tables 5 and 6 show that in order to obtain the real optimal solution the size of the problems resulting from the M4(1) formulations has to be further increased by making the discretization of the multiples finer. In the case of M3 the solution times also do not provide the complete picture. In spite of the difficulty of providing optimality the M3 formulations are capable of finding good solutions (within 95% or more of the optimal solution or the best solution after 8 h) in 1000 s or less. However, this performance is entirely dependent on the adequate selection of the number of time events, which is a recursive very expensive computational process. In addition, the ability to quickly find good solutions is lost very quickly as the relaxed problem becomes a lot harder with the increase in the number of constraints caused by additional time events and/or tasks. An example of this situation is portfolio (P1, P2, P3) in Table 7. According to number of time events used for the two project portfolios and the solution of the classical formulation for the three project portfolio it is possible to estimate with considerable confidence that the number of events required for this case is between 8 and 10. However, when the formulation is solved for such numbers of time events, after 8 h the best solution is ~25% below the one reported in Table 7, which itself is considerably suboptimal.

Effects of changes in the number of projects and timeline

The precedence based continuous formulation, M2, is the most sensitive to the increase in the number of tasks (projects). Though the number of variables growths significantly with the number of tasks, the real negative effect on the performance comes from the sharp rise in the number of constraints. The additional constraints make the relaxed problems in the branch and bound tree much more difficult. This can be clearly seen in Table 7 by comparing the 4300 nodes explored in 8 h for portfolio (P1, P2, P3), and the 62,700, 101,000, and 116,100 nodes explored for portfolios (P1, P2), (P1, P3), and (P2, P3), respectively. In addition, from the node log (not presented) it is evident that the increase in the number of tasks leads to relaxed problems that provide considerably weaker bounds. Notice that though the same type of behavior is observed with the increase in the time horizon, the cause of the problem does not have anything to do with time. M2 is completely time independent, so the reduction in

Table 4. Computational Results 100% Resource Availability and 180 Days Time Horizon

Projects	Form.	Variables		Const.	Best Sln. (M$)	Nodes	Integrality Gap (%)	CPU Time (s)
		Binary	Continuous					
P1, P2	M1 (10)	479	593	1251	77.0720	910	0	42.66
	M1 (5)	921	1143	2243	77.3168	4937	0	339.37
	M2	206	149	1789	77.3489	3200	0	235.30
	M3 (6)	428	750	3370	77.3978	316,800	0	12,662.60
	M4 (10)	632	0	153	77.1650	0	0	1.71
	M4 (5)	1292	0	261	77.3635	0	0	7.51
	M4 (1)	6242	0	1125	77.3930	0	0	353.81
P1, P3	M1 (10)	409	523	1064	51.7325	56	0	4.95
	M1 (5)	789	1011	1932	56.0110	2149	0	101.83
	M2	138	118	1110	56.1580	179	0	15.27
	M3 (6)	344	606	2719	56.1770	40,200	0	951.14
	M4 (10)	519	0	142	51.8254	0	0	1.38
	M4 (5)	1022	0	250	56.0110	0	0	5.00
	M4 (1)	5071	0	1114	56.1756	0	0	159.39
P2, P3	M1 (10)	508	622	1301	61.0400	154	0	12.94
	M1 (5)	984	1206	2361	61.1888	3000	0	206.41
	M2	206	147	1789	61.2560	6200	0	333.40
	M3 (6)	396	690	3042	61.3437	282,900	0	10,032.20
	M4 (10)	727	0	153	56.8866	0	0	2.36
	M4 (5)	1450	0	261	61.1472	0	0	8.40
	M4 (1)	7209	0	1125	61.2546	0	0	322.44
P1, P2, P3	M1 (10)	698	812	1751	92.7798	2540	0	224.45
	M1 (5)	1347	1569	3157	97.0806	56,700	0	14,782.30
	M2	380	207	3709	87.4368†	118,500	11.13	28,800
	M3 (6)	584	1020	4544	97.4458†	256,000	1.11	28,800
	M4 (10)	939	0	170	92.6043	0	0	3.76
	M4 (5)	1882	0	278	92.9435	0	0	35.21
	M4 (1)	9261	0	1142	97.2115	100	0	1398.80

Table 5. Computational Results 100% Resource Availability and 270 Days Time Horizon

Projects	Form.	Variables: Binary	Variables: Continuous	Const.	Best Sln. (M$)	Nodes	Integrality Gap (%)	CPU Time (s)
P1, P2	M1 (10)	1074	1242	2541	99.6571	1900	0	428.32
	M1 (5)	2104	2434	4760	107.3850	26,000	0	7179.62
	M2	384	235	3368	115.0508†	370,300	0.15	28,800
	M3 (8)	668	1162	5128	115.3690†	336,000	2.12	28,800
	M4 (10)	1935	0	225	107.2570	0	0	7.33
	M4 (5)	3939	0	390	115.1990	0	0	22.98
	M4 (1)	19,449	0	1686	115.3020	0	0	958.86
P1, P3	M1 (10)	935	1103	2246	84.7360	2840	0	181.53
	M1 (5)	1833	2163	4201	92.5447	15,000	0	3948.39
	M2	346	222	2961	100.3720	142,294	0	10,240.40
	M3 (9)	702	1224	5381	110.5746†	423,900	0.62	28,800
	M4 (10)	1566	0	218	87.9904	0	0	5.69
	M4 (5)	3158	0	383	100.2810	0	0	20.02
	M4 (1)	15,968	0	1679	100.4740	0	0	714.62
P2, P3	M1 (10)	1085	1253	2545	80.2273	5500	0	438.36
	M1 (5)	2135	2465	4807	80.5412	6400	0	4624.51
	M2	340	221	2887	80.5366†	525,100	0.17	28,800
	M3 (10)	748	1302	5694	80.7313†	78,200	4.21	28,800
	M4 (10)	2001	0	225	80.0298	0	0	7.99
	M4 (5)	4049	0	387	80.4731	0	0	18.58
	M4 (1)	20,435	0	1683	80.6100	0	0	938.43
P1, P2, P3	M1 (10)	1547	1715	3582	132.0740	38,900	0	16,619.00
	M1 (5)	3036	3366	6719	105.8055	36,400	32.65	28,800
	M2	760	339	7533	99.7757†	17,000	48.80	28,800
	M3 (9)	1062	1845	8099	143.1041†	25,300	8.66	28,800
	M4 (10)	2751	0	253	135.1250	30	0	26.27
	M4 (5)	5573	0	418	147.2400	100	0	130.31
	M4 (1)	27,926	0	1714	147.8560	960	0	4344.1

the solution performance is actually caused by the consideration of additional tasks.

The second continuous time formulation (M3) also exhibits a steep reduction in the solution performance caused by the increases in the number of constraints resulted from the growth in the number of tasks. However, in this case the effects are not only related to the number of tasks, but to the combined effect of additional tasks and additional time

Table 6. Computational Results 40% Resource Availability and 180 Days Time Horizon

Projects	Form.	Variables: Binary	Variables: Continuous	Const.	Best Sln. (M$)	Nodes	Integrality Gap (%)	CPU Time (s)
P1, P2	M1 (10)	479	593	1251	44.0462	3348	0	383.67
	M1 (5)	921	1143	2243	44.0931	194,000	55.65	28,800
	M2	206	149	1789	44.1520	40,700	0	2223.40
	M3 (5)	368	642	2861	44.1746	18,300	0	2666.30
	M4 (10)	632	0	153	33.2757	0	0	1.30
	M4 (5)	1292	0	261	38.5291	0	0	7.61
	M4 (1)	6242	0	1125	43.8158	200	0	1753.80
P1, P3	M1 (10)	409	523	1064	31.5980	67,300	0	888.35
	M1 (5)	789	1011	1932	31.6091	852,000	13.51	28,800
	M2	138	118	1110	35.8700	8300	0	344.82
	M3 (5)	296	519	2309	39.5150	19,100	0	838.48
	M4 (10)	519	0	142	29.2414	0	0	1.14
	M4 (5)	1022	0	250	29.2787	0	0	5.58
	M4 (1)	5071	0	1114	29.3314	0	0	621.28
P2, P3	M1 (10)	508	622	1301	38.0166	117,200	0	6851.28
	M1 (5)	984	1206	2361	29.9711	220,000	83.97	28,800
	M2	206	147	1789	38.0813	74,800	0	5462
	M3 (5)	340	590	2582	41.6652	16,200	0	1916.56
	M4 (10)	727	0	153	27.4119	0	0	1.53
	M4 (5)	1450	0	261	32.6513	0	0	6.03
	M4 (1)	7209	0	1125	37.9512	20	0	1053.70
P1, P2, P3	M1 (10)	698	812	1751	47.7286	198,000	24.88	28,800
	M1 (5)	1347	1569	3157	30.2372	80,200	183.65	28,800
	M2	380	207	3709	38.9497†	35,800	142.85	28,800
	M3 (7)	666	1167	5230	47.9122†	33,200	109.47	28,800
	M4 (10)	939	0	170	40.5722	0	0	3.20
	M4 (5)	1882	0	278	45.8254	10	0	20.85
	M4 (1)	9261	0	1142	51.1151	116	0	2090.70

Table 7. Computational Results 40% Resource Availability and 270 Days Time Horizon

Projects	Form.	Variables: Binary	Variables: Continuous	Const.	Best Sln. (M$)	Nodes	Integrality Gap (%)	CPU Time (s)
P1, P2	M1 (10)	1074	1242	2541	60.6808	98,100	63.19	28,800
	M1 (5)	2104	2434	4760	49.1499	24,500	102.84	28,800
	M2	384	235	3368	61.7028	62,700	86.75	28,800
	M3 (7)	594	1031	4532	90.8589*	86,000	20.09	28,800
	M4 (10)	1935	0	225	86.1412	75	0	42.160
	M4 (5)	3939	0	390	86.4292	1200	0	815.90
	M4 (1)	19,449	0	1686	86.5391	1900	12.36	28,800
P1, P3	M1 (10)	935	1103	2246	59.3630	130,800	39.31	28,800
	M1 (5)	1833	2163	4201	51.0486	36,300	55.95	28,800
	M2	346	222	2961	62.4711*	101,000	45.68	28,800
	M3 (6)	492	852	3701	71.4838*	192,000	12.48	28,800
	M4 (10)	1566	0	218	70.8651	20	0	19.92
	M4 (5)	3158	0	383	71.0021	189	0	171.76
	M4 (1)	15,968	0	1679	71.1412	5500	11.62	28,800
P2, P3	M1 (10)	1085	1253	2545	33.6586	46,300	136.85	28,800
	M1 (5)	2135	2465	4807	25.3090	26,400	217.03	28,800
	M2	350	221	2887	55.5545*	116,100	45.09	28,800
	M3 (7)	544	942	4083	66.6396*	75,800	21.97	28,800
	M4 (10)	2001	0	225	65.9674	193	0	52.09
	M4 (5)	4049	0	387	66.0247	22,200	0	2427.80
	M4 (1)	20,435	0	1683	66.1557	4900	12.97	28,800
P1, P2, P3	M1 (10)	1547	1715	3582	53.8610	34,100	145.01	28,800
	M1 (5)	3036	3366	6719	45.2034	7500	193.50	28,800
	M2	760	339	7533	19.6652*	4300	654.28	28,800
	M3 (7)	850	1471	6413	78.0900*	24,300	93.48	28,800
	M4 (10)	2751	0	253	88.0503	9534	0	1758.80
	M4 (5)	5573	0	418	93.3723	34,300	0	24,930
	M4 (1)	27,926	0	1714	93.6741	200	20.69	28,800

Values within parentheses are number of days per time bucket for the discrete time formulations, and number of time events for the continuous time formulation M3.
*Value from the linearized objective function.

events. This is evident by comparing portfolios (P1, P2, P3) and (P1, P2) in Tables 4 and 5.

The dependence of the continuous time formulations on time is an interesting result that deserves to be mentioned. In theory, these formulations make the model independent of time in terms of the length of the time horizon and the time buckets used to discretize it. However, during the evaluation process it was found that only the first condition is satisfied. Table 8 presents the results for a given case in which 1 and 20 days were used as time "buckets". Though additional tests were implemented, no clear relation was found between the solution time and the time "buckets." It is hypothesized that the variation in the solution times is caused by the changes in the behavior of the big-*M* constraints.

Effects of changes in the availability of resources

The reduction of the resource availability had a significant impact in the performance of formulation M1. By comparing the results in Tables 4 and 6 as well as Tables 5 and 7 it is clear that not only is the solution time significantly increased

Table 8. Effects of Time Units of Continuous Time Formulations (100% Resource Availability and 180 Days Time Horizon)

Projects	Form.	Time Unit (days)	Best Sln. (M$)	Nodes	CPU Time (s)
P1, P2	M2	1	77.3489	3200	235.30
		20	77.3489	5330	337.18
	M3(6)	1	77.3978	316,800	12,662.60
		20	77.3978	215,300	5327.51

Table 9. Comparative Summary of Formulations

Property	Formulation: M1	M2	M3	M4
Resource representation	Cont.	Cont	Cont	Disc.
Time representation	Disc.	Cont	Cont	Disc.
Objective function has to be linearized	No	Yes	Yes	No
Number of binary variables	<	<<<	<<	Baseline
Number of continuous variables	>>>	>	>>	Baseline
Number of constraints	>>>	>>>	>>	Baseline
Growth rate in binary variables with time	<	<<	<<<	Baseline
Growth rate in binary variables with projects	=	>>	=	Baseline
Growth rate in continuous variables with time	>>>	>>	>>>	Baseline
Growth rate in cont. variables with projects	>>>	>>>	>>>	Baseline
Growth rate in constraints with time	>>>	>>	>	Baseline
Growth rate in constraints with projects	>>	>>>	>>	Baseline
Favored formulation unlimited memory				■
Favored formulation unlimited time			■	

< and > indicates less and more than the baseline, respectively, and the number highlights the magnitude of the difference in a qualitative manner (i.e, the more the larger the difference).

but also the quality of the best solution deteriorates. It was found that the main drivers of this behavior were the loss of effectiveness of the cuts used by CPLEX and the increased difficulty in finding good solutions at the top of the branch and bound tree. M2 also showed in some cases significant reductions in the quality of the solutions (when compared with the ones from M3), something that is expected due to the use of conservative resource allocation constraints. However, the causes of the problem turned out to be the same as the ones found for M1. This result, as most of the ones in the comparisons above instead of providing clarity about the superiority of any of the formulations, just highlights the fact that the proper selection of a model is problem dependent

Conclusions

The use of continuous divisible resources and continuous time to address large-scale multimode RCMPSP has been explored in this article. Three different MILP models were developed. The proposed models differ in the strategy used to handle the time domain. The first formulation uses the standard discrete time representation, the second one relies on the idea of task precedence to eliminate time from the model, and the third is based on time events. A relative comparison, including the conventional discrete time multimode approach, shows that the reduction in the number of variables obtained by the use of continuous variables is offset by the increase in the number constraints and the corresponding negative effects in the relaxed model. This situation limits the applicability of the proposed models to the same scale of problems solvable by the conventional approach. However, in general terms, the selection of the most suitable formulation will depend on the computational resource available to the user. If memory and solution time are limited the best formulation will be dictated by the specific characteristics of the portfolio. On the other hand, if one of these computational resource constraints is relaxed, the conventional formulation, M4, is the top option for "unlimited" memory, whereas M3 is the best option for "unlimited" time. These results and the main characteristics of each formulation are summarized on a comparative basis in Table 9.

To conclude, it is important to recognize that in practice the number of tasks considered in a portfolio and the time horizons are much larger than the ones considered in the case study. Therefore, in spite of the modeling efforts to develop more efficient formulations, the use of heuristics, tailored branch and cut algorithms, and simulation-based strategies at present remains the only practical way to tackle such problems.

Notation

Indices

g = grid points (linearization objective function)
h = preceding tasks
i = task
j = combinations
k = resources
n = event points
p = projects
t = time periods
w = preceding tasks combinations

Sets

H_{pi} = set of tasks preceding task i of project p
I_p = set of tasks that are part of project p
I^t_{EST} = set of tasks that have an EST shorter than t
I^t_{EFT} = set of tasks that have an EFT shorter than t
I_{pip} = set of tasks within project p that can be processed in parallel with task i
J_{pi} = set of combinations that can be allocated to tasks i of project p
$J^t_{\lceil \mathrm{EFT}_{pi} \rceil}$ = set of combinations that can be allocated to tasks i of project p that guarantees and EFT shorter than t

Parameters

a_{pig} = location of grid point g for task i of project p
EST_{pi} = earliest start time of task i of project p
EFT_{pi} = earliest start time of task i of project p
EFT_{pij} = earliest start time of task i of project p if combination j is allocated ($\mathrm{EST}_{pi} + \alpha_{pij} - \beta_{pij} L^{\max}_{pij}$)
$L^{\max}_{pij}$ = upper bound for the multiple of combination j that can be assigned to task i of project p
$L^{\max}_{pij}$ = lower bound for the multiple of combination j that can be assigned to task i of project p
r = discounting rate
$R^{\max}_k$ = upper bound for renewable resource k
T = length of the time horizon
W_{pij} = objective function weighting factor for task i of project p when combination j is allocated; it is based on the expected annual returns of the project and the probabilities of success of different tasks
α_{pij} = duration of task i of project p when the minimum multiple of combination j is allocated plus $\beta_{pij} L^{\min}_{pij}$
β_{pij} = factor that transforms combination multiples into task durations
ρ_{pijk} = ratios (factors) of resources k that are part of combination j when it is allocated to task i of project p

Binary variables

v_{pijg} = 1 if the grid point g is part of the convex combination used to approximate the term $e^{-rC_{pi}}\, x_{pij}$ or $e^{-rT^{\mathrm{f}}_{pin}} \sum_n x^{\mathrm{s}}_{pijn}$
x_{pij} = 1 if task i of project p is completed with combination j, 0 otherwise
$x^{\mathrm{s}}_{pijt(n)}$ = 1 if task i of project p is started at the beginning of time t (time event n) with combination j, 0 otherwise
$x^{\mathrm{f}}_{pijt(n)}$ = 1 if task i of project p is finished at the beginning of time t (time event n) with combination j, 0 otherwise
x^{p}_{pijn} = 1 if task i of project p is being processed at event point n with combination j, 0
$y_{pip'i'}$ = 1 if task i' of project p' has been completed after starting task i of project p, 0 otherwise

Continuous variables

C_{pi} = completion time of task i of project p
D_{pijn} = duration of task i of project p that starts at event point n using combination j (it is 0 for all the event points but the one when the task is started)
L_{pij} = multiple of combination j used to complete task i of project
$L^{\mathrm{s}}_{pijt(n)}$ = multiple of combination j engaged at time t (time event n) to undertake task i of project p
$L^{\mathrm{f}}_{pijt(n)}$ = multiple of combination j disengaged at time t (time event n) from task i of project p
L^{p}_{pijn} = multiple of combination j being used at event point n to undertake task i of project
$R_{kt(n)}$ = total amount of resource k that is being used at time t (time event n)

S_{pij} = slack variable used in the piecewise linearization of the objective function for task i of project p when completed with combination j

T_n = time that corresponds to event point n

T^f_{pin} = finish time of task i of project p (it is 0 for all the event points before the one in which it is started and takes the value of the finish time of the given task for all the remaining event points)

T^s_{pijn} = start time of task i of project p when combination j is used to undertake it (it is 0 for all the event points but the one when the task is started)

T^f_{pijn} = finish time of task i of project p when combination j is used to undertake it (it is 0 for all the event points before the one in which it is started and takes the value of the finish time of the given task for all the remaining event points)

$z^d_{pip'i'j}$ = linearization variable for $L_{p'i'j}y_{pip'i'}$

$z^r_{pip'i'j}$ = linearization variable for $L_{p'i'j}y_{p'i'pi}$

λ_{pijg} = weight between 0 and 1 of the grid point g in the convex combination used to approximate the term $e^{-rC_{pi}}\,x_{pij}$ or $e^{-rT^f_{pin}}\sum_n x^s_{pijn}$

Literature Cited

1. Demeulemeester EL, Herroelen W. *Project Scheduling: A Research Handbook*. Boston: Kluwer Academic Publishers, 2002.
2. Blau GE, Pekny JF, Varma VA, Bunch PR. Managing a portfolio of interdependent new product candidates in the pharmaceutical industry. *J Prod Innovat Manage*. 2004;21:227–245.
3. Subramanian D, Pekny JF, Reklaitis GV. A simulation-optimization framework for research and development pipeline management. *AIChE J*. 2001;47:2226–2242.
4. Maravelias CT, Grossmann IE. Optimal resource investment and scheduling of tests for new product development. *Comput Chem Eng*. 2004;28:1021–1038.
5. Herroelen W. Project scheduling - Theory and practice. *Prod Oper Manage*. 2005;14:413–432.
6. Brucker P, Drexl A, Mohring R, Neumann K, Pesch E. Resource-constrained project scheduling: notation, classification, models, and methods. *Eur J Oper Resh*. 1999;112:3–41.
7. Salewski F, Schirmer A, Drexl A. Project scheduling under resource and mode identity constraints: model, complexity, methods, and application. *Eur J Oper Res*. 1997;102:88–110.
8. Weglarz J. *Project Scheduling: Recent Models, Algorithms, and Applications*. Boston: Kluwer, 1999.
9. Schmidt CW, Grossmann IE. Optimization models for the scheduling of testing tasks in new product development. *Ind Eng Chem Res*. 1996;35:3498–3510.
10. Honkomp SJ. Solving mathematical programming planning models subject to stochastic task success, PhD Thesis, School of Chemical Engineering, Purdue University, West Lafayette, IN, 1998.
11. Jain V, Grossmann IE. Resource-constrained scheduling of tests in new product development. *Ind Eng Chem Res*. 1999;38:3013–3026.
12. Subramanian D, Pekny JF, Reklaitis GV, Blau GE. Simulation-optimization framework for stochastic optimization of R and D pipeline management. *AIChE J*. 2003;49:96–112.
13. Mendez CA, Henning GP, Cerda J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput Chem Eng*. 2001;25:701–711.
14. Maravelias CT, Grossmann IE. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2003;42:4422–4422.
15. Tavares LV. A review of the contribution of operational research to project management. *Eur J Oper Res*. 2002;136:1–18.
16. Kis T. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Math Program*. 2005;103:515–539.
17. Floudas CA, Lin XX. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng*. 2004;28:2109–2129.
18. Glover F. Improved linear integer programming formulations of nonlinear integer problems. *Manage Sci*. 1975;22:455–460.
19. Castro P, Barbosa-Povoa A, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2001;40:2059–2068.
20. Raman R, Grossmann IE. Modeling and computational techniques for logic-based integer programming. *Comput Chem Eng*. 1994;18:563–578.

Appendix: Example Data

The precedence relations for each of the projects are presented in Figure A1 and the corresponding data in Tables A1–A10............

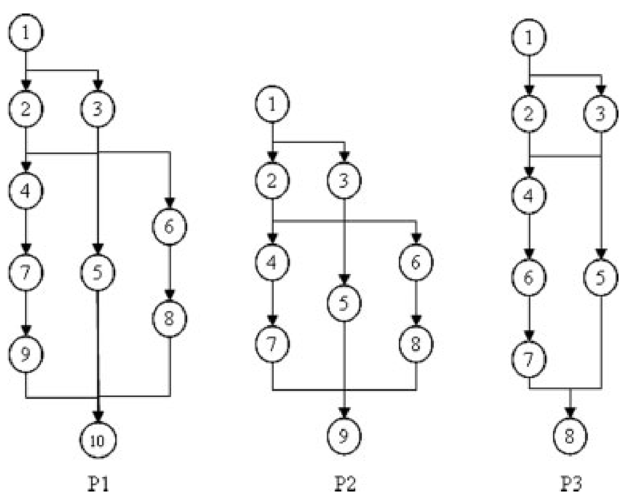

Figure A1. Project execution plans.

Table A1. Forecasted Annual Returns

Project	Returns (M$)
1	10
2	7
3	5

Table A2. Resource Capacities

Resource	Total Capacity
1	350
2	320
3	400
4	290
5	310
6	300

Table A3. Factors to Transform Combination Multiples Into Task Durations

Project	Combination	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	7.5	9.4	32	1.8	13.0	13.7	37.5	25.0	1.0	15.5
	Outsourced	5.0	6.3	22	1.2	9.0	9.0	25.0	16.5	0.8	10.5
2	In-house	8.1	10.8	24	2.0	8.0	13.5	1.0	25.5	35.0	
	Outsourced	5.4	7.2	16	1.3	5.5	9.0	0.5	17.0	23.0	
3	In-house	8.3	9.7	22	2.8	36.0	43.0	1.0	11.0		
	Out-sourced	5.4	6.6	15	2.0	24.0	28.5	0.8	7.5		

Table A4. Upper Bounds for Combination Multiples

Project	Combination	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	13	13	3	8	7	5	6	6	12	6
	Outsourced	10	10	2	6	6	4	5	5	10	5
2	In-house	12	12	3	7	6	4	11	6	5	
	Outsourced	10	10	2	6	5	3	9	5	4	
3	In-house	11	12	3	7	5	5	11	5		
	Outsourced	9	10	2	6	4	4	9	4		

Table A5. Lower Bounds for Combination Multiples

Project	Combination	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	5	6	2	3	5	2	4	4	7	4
	Outsourced	4	4	1	2	4	1	3	3	5	3
2	In-house	5	6	2	3	4	2	7	4	4	
	Outsourced	4	4	1	2	3	1	5	3	3	
3	In-house	4	5	2	3	4	3	6	3		
	Outsourced	3	4	1	2	3	2	4	2		

Table A6. Maximum Task Durations in Days

Project	Combination	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	120	158	85	30	103	72	200	90	10	60
	Outsourced	85	118	79	21	78	57	158	77	9	51
2	In-house	101	140	62	29	90	51	8	81	57	
	Outsourced	72	107	57	20	66	43	6	71	61	
3	In-house	110	136	60	28	97	182	10	49		
	Outsourced	78	98	55	21	89	150	9	41		

Table A7. Success Probability of Tasks that can Fail

Project	Task	Success Prob.
1	3	85
	5	95
	6	90
	7	98
2	3	90
	5	97
	8	83
3	3	88
	6	97
	7	93

Table A8. Resource Factors for the Composition Ratios (ρ_{pijk})

Project	Combination	Resource	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	1	14	8	9	10	15	15	11	14	9	10
		2	12	15	15	17	12	8	15	8	15	10
		3	15	15	15	16	15	15	9	8	14	10
	Outsourced	4	12	17	12	18	9	22	16	14	11	11
		5	9	18	24	20	14	24	16	11	16	9
		6	8	18	24	20	10	24	11	10	9	11
2	In-house	1	9	15	14	14	15	13	8	14	4	
		2	10	15	13	8	13	11	11	14	8	
		3	14	10	14	16	15	9	14	10	6	
	Outsourced	4	11	18	22	14	8	18	13	5	10	
		5	15	15	14	18	16	21	6	9	4	
		6	9	18	17	19	16	21	15	13	10	
3	In-house	1	11	17	13	15	12	9	15	5		
		2	6	10	13	13	5	12	15	8		
		3	16	10	9	8	13	12	12	10		
	Outsourced	4	12	16	13	18	14	21	9	6		
		5	18	15	13	10	14	21	14	9		
		6	7	17	13	9	5	14	18	12		

Table A9. Multiples for Each Combination (Conventional Formulation)

Project	Combination	Multiple	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	1	10	13	3	8	7	5	6	6	12	6
		2	11	11.25	2.75	6.75	6.5	4.25	5.5	5.5	10.75	5.5
		3	9	9.5	2.5	5.5	6	3.5	5	5	9.5	5
		4	7	7.75	2.25	4.25	5.5	2.75	4.5	4.5	8.25	4.5
		5	5	6	2	3	5	2	4	4	7	4
	Outsourced	1	10	10	2	6	6	4	5	5	10	5
		2	8.5	8.5	1.75	5	5.5	3.25	4.5	4.5	8.75	4.5
		3	7	7	1.5	4	5	2.5	4	4	7.5	4
		4	5.5	5.5	1.25	3	4.5	1.75	3.5	3.5	6.25	3.5
		5	4	4	1	2	4	1	3	3	5	3
2	In-house	1	12	12	3	7	6	4	11	6	5	0
		2	10.25	10.5	2.75	6	5.5	3.5	10	5.5	4.75	0
		3	8.5	9	2.5	5	5	3	9	5	4.5	0
		4	6.75	7.5	2.25	4	4.5	2.5	8	4.5	4.25	0
		5	5	6	2	3	4	2	7	4	4	0
	Outsourced	1	10	10	2	6	5	3	9	5	4	0
		2	8.5	8.5	1.75	5	4.5	2.5	8	4.5	3.75	0
		3	7	7	1.5	4	4	2	7	4	3.5	0
		4	5.5	5.5	1.25	3	3.5	1.5	6	3.5	3.25	0
		5	4	4	1	2	3	1	5	3	3	0
3	In-house	1	11	12	3	7	5	5	11	5	5	0
		2	9.25	10.25	2.75	6	4.75	4.5	9.75	4.5	4.75	0
		3	7.5	8.5	2.5	5	4.5	4	8.5	4	4.5	0
		4	5.75	6.75	2.25	4	4.25	3.5	7.25	3.5	4.25	0
		5	4	5	2	3	4	3	6	3	4	0
	Outsourced	1	9	10	2	6	4	4	9	4	0	0
		2	7.5	8.5	1.75	5	3.75	3.5	7.75	4	0	0
		3	6	7	1.5	4	3.5	3	6.5	3	0	0
		4	4.5	5.5	1.25	3	3.25	2.5	5.25	2.5	0	0
		5	3	4	1	2	3	2	4	2	0	0

Table A10. Task Duration for Each Multiple of Each Combination in Days (Conventional Formulation)

Project	Combination	Multiple	Task 1	2	3	4	5	6	7	8	9	10
1	In-house	1	60	92	53	21	77	31	125	40	5	29
		2	75	108.5	61	23.3	83.5	41.3	143.8	52.5	6.3	36.8
		3	90	125	69	25.5	90	51.5	162.5	65	7.5	44.5
		4	105	141.5	77	27.8	96.5	61.8	181.3	77.5	8.8	52.3
		5	120	158	85	30	103	72	200	90	10	60
	Outsourced	1	55	80	57	16	60	30	108	44	5	30
		2	62.5	89.4	62.5	17.2	64.5	36.8	120.5	52.3	6	35.3
		3	70	98.9	68	18.4	69	43.5	133	60.5	7	40.5
		4	77.5	108.3	73.5	19.6	73.5	50.3	145.5	68.8	8	45.8
		5	85	118	79	21	78	57	158	77	9	51
2	In-house	1	44	75	38	21	74	24	4	30	22	0
		2	58.3	91.3	44	23	78	30.8	5	42.8	30.8	0
		3	72.5	107.5	50	25	82	37.5	6	55.5	39.5	0
		4	86.8	123.8	56	27	86	44.3	7	68.3	48.3	0
		5	101	140	62	29	90	51	8	81	57	0
	Outsourced	1	40	64	41	15	55	25	4	37	38	0
		2	48.1	74.8	45	16.3	57.8	29.5	4.5	45.5	43.8	0
		3	56.3	85.5	49	17.5	60.5	34	5	54	49.5	0
		4	64.4	96.3	53	18.8	63.3	38.5	5.5	62.5	55.3	0
		5	73	107	57	20	66	43	6	71	61	0
3	In-house	1	52	68	38	17	61	96	5	27	5	0
		2	66.5	85	43.5	19.8	70	117.5	6.3	32.5	4.75	0
		3	81	102	49	22.5	79	139	7.5	38	4.5	0
		4	95.5	119	54.5	25.3	88	160.5	8.8	43.5	4.25	0
		5	110	136	60	28	97	182	10	49	4	0
	Outsourced	1	46	58	40	13	65	93	5	26	0	0
		2	54.1	67.9	43.8	15	71	107.3	6	26	0	0
		3	62.3	77.7	47.5	17	77	121.5	7	33.5	0	0
		4	70.4	87.6	51.3	19	83	135.8	8	37.3	0	0
		5	79	97	55	21	89	150	9	41	0	0

Manuscript received Jan. 15, 2008, and revision received Apr. 3, 2008.