Mathematical modelling of a hierarchical framework for controlling NPD projects under a hard time constraint

Mathematical modelling of a hierarchical framework for controlling NPD projects under a hard time constraint

A.B. Dragut[1], J.W.M. Bertrand[2]

Abstract

Based on the hierarchical framework introduced in Dragut, Bertrand (2002) for the control of New Product Development (NPD) projects under a hard time constraint, we formulate mathematically the project control. The relationships with the well-known mathematical project models are discussed. The framework splits the project horizon into a number of review periods. At the start of each review period the project state is reviewed in order incorporate the new information about the customer needs, and about the progress the engineers made in working on design tasks. This leads to the addition/deletion of design tasks, and to a stochastic solving time of the design tasks.

The paper contributes to the area of mathematical models for the organization of work in an NPD, and to the development of management-related control concepts in the NPD projects, both areas that present research opportunities according to Brown, Eisenhardt (1995).

Keywords: mathematical modelling, hierarchical control, NPD projects, dynamic product definition, planning, engineering.

1 Introduction

In a previous paper (Dragut, Bertrand, 2002) we introduced a hierarchical framework for the control of New Product Development (NPD) projects under a hard time constraint. The framework splits the project horizon into a number of review periods. At the start of each review period the state of the project is reviewed in order incorporate the new information about the customer needs (from the market), and about the progress the engineers made in working on design tasks. In response to the new information the framework considers the process of solving design tasks by the engineers, the allocation of design tasks to engineers, and the updating of the performance levels of design tasks. The framework furthermore assumes that decisions are taken in order to maximize at the deadline the expected market value of the new product. With a probability greater than a given safety margin, this new product will be delivered at the deadline, to the market.

The previous paper also gives a list of concepts, variables and relationships that enable the mathematical modelling of the process of performing the design tasks, of the dynamics in the design tasks that constitute the project, and of the relationship between expected marked performance of a product and the performance levels of the design tasks of the project. These concepts, variables

[1]Dept. of Operations Planning and Control, Faculty of Technological Management, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands (e-mail: {a.b.dragut@tm.tue.nl})

[2]Dept. of Operations Planning and Control, Faculty of Technological Management, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

and relationships take into account relevant knowledge in published literature or new product management, product development, system engineering, project planning and control, production scheduling and human performance management.

The current paper is a sequel to the previous paper and presents the mathematical modelling of the hierarchical control framework. Section 2 describes the aggregate decision process of periodically setting the design tasks performance levels given the update of the project status in view of the time and capacity remaining until the project deadline. Section 3 deals with the detailed planning process consisting of a re-scheduling decision function, its associated re-scheduling problem, and the allocation of design tasks to the engineers subject to workload constraints, and a common due date. In section 4 we present the engineering process that transforms the project state during each review period, by solving their allocated design tasks.

The aggregate decision process, as well as the detailed planning level allocation problem are specific for controlling NPD projects under a hard time constraint, and their formulation it is to the best of our knowledge new. The entire framework can be used for determining in probabilistic terms the expected NPD project outcome, as shown in section 5. It is not practical to solve separately for each review period the problems defined at the hierarchical structure levels, but the review periods can be easy linked using their approximate solutions. The paper is concluded in section 6.

The global variables of our NPD project model are:

T : the total number of review periods (review periods are numbered from 0 to $T-1$);

M : the total number of engineers;

$\overline{N}$: the initial number of design tasks;

N : an upper bound for the maximum number of design tasks during the whole project;

L : the number of performance levels per design task (levels are numbered from 0 to L, where 0 means that design task will not be performed any more);

Δ : the total number of customer needs considered;

h : the short time detailed planning horizon, which is a multiple number of review periods;

$c(n)$: the cost of performing one activity of the design task n; $n = 1, ..., N$

μ : the rate of the exponential distribution of an activity solving time.

2 The aggregate decision process

The aggregate decision problem formulation deals with the technological uncertainty, and market requirement variability (Huchzermeier, Loch, 2001; Bhattacharya et al., 1998).

At the beginning of each review period t, $(t, t+1]$, the aggregate controller, decides first whether to continue or not the NPD project. The abandonment is the result of either an expected exceeded NPD budget, or of a low product performance. In case of continuation the controller modifies the design tasks performance levels in an interactive process aiming at a maximal market payoff at the deadline. Thus, the targets on design tasks realization for the detailed planning level are provided, under the constraint of achieving the currently desired performance levels, at the deadline, with a certain probability. Decisions on allocating design tasks to resources (engineers) are not considered here.

A directed acyclic graph of design tasks reflects the precedence relations among design tasks at the beginning of each review period. We construct stages for our graph, by associating a representation into T independent sets to it: sets of unordered design tasks (no precedence relations between any two of them) and all having the same length of the longest path from the start dummy node to them (in the precedence graph), see Figure 1.

Figure 1.

Empty sets can be added in the T-partition if the controller wants to control more often than the number of independents sets. The concept of a T- stage network associates naturally the t-th decision moment with the allocation of the t-th set of the partition of design tasks.

Input parameters (at the beginning of review period t**):**

$\alpha(t)$: the required current safety margin for the probability of completing the project before the deadline; $\alpha(t) \in (0,1)$

$\beta(t)$: the required current safety margin for the probability of exceeding the maximal team solving capacity; $\beta(t) \in (0,1)$

$w_\delta(t)$: the customer need normalized weight δ**;** $\forall\ \delta = 1, ..., \Delta$;

$B(t)$: the current remaining NPD project budget;

$R_t := \left(\Lambda_t^0 \cup \Lambda_t^1 \cup ... \cup \Lambda_t^{T-1}, \mathcal{A}\left(\Lambda_t^0 \cup \Lambda_t^1 \cup ... \cup \Lambda_t^{T-1}\right)\right)$: the newly updated T-partite directed acyclic graph of unfinished design tasks precedence relations; $\Lambda_t^{t_0}$ is the current design tasks set to be allocated at t_0;

$X_i(t-1)$: the set of newly arrived design tasks (during review period $t-1$) concurrent with design tasks allocated/to be allocated during the i-th review period; In Dragut, Bertrand (2002) we assumed for each $i = 0, ..., t$ a general Markovian review period-dependent arrival process of new design tasks with a common contribution function and identical performance level structure.

$l_{\min}(\cdot, t) : \{1, ..N\} \to \{0, ..., L\}$: the minimal performance design task level function;

$l(\cdot, t) : \{1, ..N\} \to \{0, ..., L\}$: the achieved performance design task level function;

$A(n, t, l)$: the number of sequential activities planned for solving the design task n, at the performance level l, assuming the previous levels already solved. All activities are assumed to have the solving time exponentially distributed with the same mean time μ (see for empirical evidence Best, 1995; Reed, 1998), independent of the engineer which will perform them; $\forall\ n = 1, ..., N, \forall l = 1, ..., L$;

$\Theta(n, t, \delta)$: the normalized maximal contribution of the design task n in fulfilling the customer need δ ; $\Theta(n, t, \delta) \in (0, 1]$; $\forall\ n = 1, ..., N, \forall\ \delta = 1, ..., \Delta$;

Since N is an upper bound, we set to zero all the parameters depending on an artificial $n \leq N$.

Notation (at the beginning of review period t**):**

$N(t-1)$: the random variable giving the design tasks number arrived since the NPD project beginning until the end of review period $t-1$, $[t-1,t)$;

$S_n(t,l)$: the solving time of a design task n, if it is performed at the level l; $n = 1,...,N$. They are independent random variables Erlang-$\sum_{i=1}^{l} A(n,t,i)$, with mean $\frac{\sum_{i=1}^{l} A(n,t,i)}{\mu}$;

$C(t,l(\cdot,t),R)$: the completion time of the network of design tasks, R, if $l(\cdot,t)$ gives the design tasks performance levels;

$\overline{S}_\delta(t,l(\cdot,t))$: the normalized S-functions family giving the market payoff , function of the distance between the cumulated design tasks contribution per customer need δ, $\sum_{n=1,..,N} \Theta(n,t,\delta) \cdot \frac{l(n,t)}{L}$, and its ideal value; $\overline{S}_\delta(t,l(\cdot,t)) \in (0,1), \forall\delta = 1,...,\Delta$.

$M\left[\left(\Theta(t,\cdot,\delta)_{1\leq\delta\leq\Delta}, l(\cdot,t)\right)_{1\leq n\leq N}\right] := \prod_{\delta=1}^{\Delta}\left[\overline{S}_\delta(t,l(\cdot,t))\right]^{w_\delta(t)}$ (Yoshimura 1996), or $\sum_{\delta=1}^{\Delta} w_\delta(t)\left[\sum_{n=1,..,N}\Theta(n,t,\delta)\cdot\frac{l(n,t)}{L}\right]$ (Askin, Dawson, 2000) : the cumulated market payoff function, where $l(\cdot,t)$ gives the design tasks performance levels;

Decision function:

$\hat{l}(\cdot,t) : \{1,...,N\} \to \{0,...,L\}$: the desired design task performance level function, at the beginning of review period t

Then, the aggregate planing problem is:

$$\max_{\hat{l}(n,t),\ n\in\{1,...,N\}} M\left[\Theta\left(t,\hat{l}(\cdot,t),\delta\right)_{1\leq\delta\leq\Delta}\right] \tag{1}$$

subject to:

$$\Pr\left\{C\left(t,\hat{l}(\cdot,t),R_t\right) \leq (T-t)\right\} \geq \alpha(t) \tag{2}$$

$$\Pr\left\{\sum_{\substack{n\in\Lambda_t^0\cup\Lambda_t^1\cup...\cup\Lambda_t^{T-1} \\ or \\ \overline{N}+E[N(t-1)]+1\leq\ n\leq\overline{N}+E[N(T-1)],\ \text{for}\ t>0}} S_n\left(t,\hat{l}(n,t)\right) \leq M\cdot(T-t)\right\} \geq \beta(t) \tag{3}$$

$$\sum_{t_0=0}^{T-1}\left[\sum_{\substack{n\in\Lambda_t^{t_0} \\ or \\ \overline{N}+E[N(t_0-1)]+1\leq n\leq\overline{N}+E[N(t_0)],\text{ for }t_0>0}}\sum_{i=l(n,t)}^{\hat{l}(n,t)}A\left(n,t,i\right)\cdot c\left(n\right)\right]\leq B\left(t\right) \tag{4}$$

$$\hat{l}\left(n,t\right)\geq\min\left(l_{\min}\left(n,t\right),l\left(n,t\right)\right),\ n=1,...,N \tag{5}$$

¿From (2) the completion time of the NPD project defined by the design tasks desired performance must be smaller than the deadline with a probability at least the current safety margin $\alpha(t)$. The analytical evaluation of a general directed acyclic graph completion time distribution is a NP-complete problem, ($P\neq NP$), but Colajanni et al. (2000) gives a polynomial time algorithm for determining an upper bound.

¿From (3) the remaining total workload does not exceed the team remaining maximal solving capacity with a probability at least the required $\beta(t)$ margin. The workload is computed by adding the remaining solving times of the unfinished design tasks from R_t. This constraint takes into account the average random arrival of new design tasks. The computation of the time required to complete both the planned and additional expected design tasks can be done under the assumption of review period-dependent Poisson arrival processes of design tasks with a common contribution function and identical performance level structure.

The constraint (4) gives the condition of abandoning the project in the case of exceeding the remaining budget. The cost per activity may differ from one design task to another, refining the previous constraint.

¿From the last inequality the target performance level of a design task should at least the minimal target or what the engineers already achieved.

This aggregate project planning formulation studies the project risk in terms of the probability of obtaining a total duration, a total quality, and a total cost, achieving dynamically the product definition. During the aggregate decision making the controller has to assign numerical values to $\alpha(t)$, $\beta(t)$. Those choices depend on the risk that the controller is willing to take. The adoption of small values means that more time will be spent on the design tasks already allocated, since more uncertainty is allowed for the outcome of the NPD project. Adopting large values means precisely the opposite. A selection with a large $\alpha(t)$ and a small $\beta(t)$ means that the controller focuses more on the possibility of finishing the design tasks situated on most "stochastic critical paths ", than on the capacity issue.

3 The Detailed Planning Process

At the beginning of each review period t, $(t,t+1]$, a set of concurrent (planned or newly arrived to the team of engineers from the review period $t-1$)) design tasks ,$Y(t) \overset{notation}{:=} X_t\,(t-1) \cup \Lambda_t^t$, that can be solved in parallel is allocated to the team of engineers.

We also have three more other sets of design tasks subject to precedence constraints inherited from the project network:

- the set of updated design tasks already scheduled in previous review periods.

- the set of newly arrived design tasks to the team of engineers (from the review period $t-1$), $X\,(t-1) \overset{notation}{:=} \left(\bigcup_{i=0,\ldots,t-1} X_i\,(t-1) \right)$,which are concurrent with previously allocated design tasks.

- the set of design tasks left over after the allocation procedure has taken place at the detailed planning level for the previous review period $t-1$. If at the beginning of this review period the set $Y(t)$ is empty, they will be regarded as new design tasks to be allocated to the engineers, and $Y(t)$ will updated to contain them. The same happens with newly arrived design tasks that are concurrent with them. If the set $Y(t)$ is not empty, they will be included in the set $X\,(t-1)$, and will be subject first to scheduling, and then to workload and common due date constraints.

At the detailed planning level, first a re-scheduling decision is taken for some of the design tasks assigned to the engineers in previous review periods, in case of infeasibility problems. Second, the (re-)scheduling takes place for the team of engineers. The design sets considered for (re-)scheduling are not only the ones affected by the re-scheduling decision function, but also the set $X\,(t-1)$ of newly arrived design tasks to the team of engineers from the review period $t-1$, which are concurrent with previously allocated design tasks**.** The (re-)scheduling problem aims to minimize the expected maximum completion time of a set of design tasks subject to precedence constraints, to be solved by the M engineers working in parallel.

Afterwards, the concurrent design tasks, $Y(t)$, are allocated for each engineer on an individual basis. In order to investigate the problem more precisely, we need to model the dependency between engineers' productivity and their perceived work pressure. In Bowers et al., (1997) a first estimation of the work pressure is given by comparing, for a given engineer, the estimated duration for completing the allocated design tasks and the available time until the deadline. Design task solving times are stochastic, and there is a common due date $t+h$ for all those allocated to engineers (h is called short time planning horizon, and given as input). Thus, we use the probability $\overline{p}_h(A)$ of finishing all design tasks of the set A in time h. We also suppose we are given as input for each engineer m an optimal value $\alpha\,(m)$ for this probability, called optimal work pressure level. Therefore, we achieve efficiency for the engineers involved in the process by requiring for each engineer m that $|\overline{p}_h\,(A_m) - \alpha\,(m)| < \varepsilon$. In words, this requires that for a short time horizon h, the probability of finishing the design tasks allocated to m is close to an optimal subjective value (the closeness ε being also an input). This ensures both not overloading the engineers and not giving them too little work to do. Organizational psychology (see for a review

Wickers, 1996) shows that this dependency between the productivity and the work pressure is curvilinear, and this is the reason of the absolute value bound. The impact of work pressure on engineers productivity has been confirmed by experimental research in Oorschot (2001).

The allocation problem aims to maximize the value of the set of design tasks allocated in the short-time planning horizon, subject to workload constraints for each engineer, and with no precedence constraints between the tasks. The detailed planning level controller does not know the market payoff function, nor the customer needs. For optimization purposes we derive *design task value functions*, as an indication of design task realizations influence on the market payoff function. They are obtained by taking into account both the *design task contribution functions*, and the type of *cumulative market payoff function* (see Appendix).

The design tasks left over after the allocation procedure will be further maintained available at the detailed planning level. No due date will be set for them. This way of doing detailed planning reflects the particular purpose of due dates in NPD processes. Since the precedence relationships in between tasks are quite loose, the due dates are not mostly given to be met, but to ensure the efficiency of the involved engineers. So, while the deadline of the whole NPD project is a hard constraint, design tasks due dates are not like that.

Input parameters (at the beginning of review period t**):**

$\alpha(m)$: the optimal work pressure level for the engineer m, for all $m = 1, ..., M$.

ε : the closeness parameter, i.e. the allowed variation with respect to the optimal work pressure level of each engineer.

$\sigma_m^{\setminus}(t)$: the m-th engineer scheduled design tasks sequence from review periods prior to t, $m = 1, ..., M$;

$d(n,t)$: the due date of the design task n at the beginning of review period $t; n \in \sigma_m^{\setminus}(t)$;

$X_i(t-1)$: the set of newly arrived design tasks (during review period $t-1$) concurrent with design tasks allocated/to be allocated during the i-th review period, $i = 0, ..., t$;

$Z(t) := \bigcup_{m=1,...,M} \sigma_m^{\setminus}(t)$: the set of design tasks that were already scheduled in previous review periods;

$G_t = (J(G_t), \mathcal{A}(G_t))$: the directed, acyclic graph of precedence relations among design tasks at the detailed planning level, where $J(G_t) := Z(t) \cup \left(\bigcup_{i=0,...,t-1} X_i(t-1)\right)$**;**

$Y(t) := X_t(t-1) \cup \Lambda_t^t$:the set of concurrent (planned or newly arrived) design tasks to be allocated to the engineers at the beginning of review period t; $K := |Y(t)|$.

$\hat{l}(n,t)$: the n-th design task performance, as established at the aggregate decision level, $n \in J(G_t) \cup Y(t)$;

$A(n,t,l)$: the n-th design task number of planned activities, if its performance level is l, assuming the previous levels already solved; $l = 1, ..., \hat{l}(n,t)$, $n \in J(G_t) \cup Y(t)$; A design task that is in progress will start with the next activity not performed yet.

$V(n,t)$: the n-th design task value; $\forall\ n = 1, ..., N$;

Notation (at the beginning of review period t**):**

$X(t-1) := \bigcup_{i=0,\dots,t-1} X_i(t-1)$: the set of newly arrived design tasks (during review period $t-1$) concurrent with previously allocated ones;

$k|\sigma$: the real index (in the numbering of design task from 1 to N) of the k-th design task from the sequence σ, $\forall k = 1, ..., |\sigma|$

$C_n(t)$: the random variable denoting the n-th design task completion time, $n \in J(G_t)$

$E_n(t)$: the random variable denoting the n-th design task earliest start time, $n \in J(G_t)$

Pred (n) : the set of direct predecessors of the design task n in the graph G_t, $n \in J(G_t)$

$\widehat{S}_n(t)$: the random variable denoting the n-th design task solving time; $\forall n = 1, ..., N$

For an engineer, to solve one of the design tasks, n, scheduled to him means that he has to solve sequentially the list of activities planned for it at the beginning of the review period t, for each level up to $\hat{l}(n,t)$ (Dragut, Bertrand, 2002). Thus, we consider the solving time of any design task n as being a random variable distributed Erlang-$\sum_{i=1}^{\hat{l}(n,t)} A(n,t,i)$, with mean $\dfrac{\sum_{i=1}^{\hat{l}(n,t)} A(n,t,i)}{\mu}$.

3.1 Re-scheduling decision function

At the beginning of each review period t, $(t, t+1]$, the re-scheduling decision function decides whether for the updated design tasks in $Z(t) \cup X(t-1)$ we already have a partial schedule, satisfying the precedence relations among the design tasks according to a stochastic ordering. More explicitly, if a design task has direct predecessors in the set $J(G_t)$, then it will be assigned to an engineer such that its starting time will be greater than the completion time of all its predecessors according to a stochastic ordering.

Definition 1 *Given X_1, X_2 two random variables with distribution functions F_{X_1}, F_{X_2} we say that X_1 is stochastically smaller than X_2 (denoted by $X_1 \leq_{stoch} X_2$) if $F_{X_1}(z) \leq F_{X_2}(z), \forall z \geq 0$. The stochastic ordering is a partial order relationship among random variables and distribution functions which is closed under multiplication and convolution.*

We have $E_{k|\sigma_m^{\setminus}(t)}(t) := \sum_{i=1}^{k-1} \left[S_{i|\sigma_m^{\setminus}(t)}(t)\right]$ and $C_{k|\sigma_m^{\setminus}(t)}(t) := \sum_{i=1}^{k} \left[S_{i|\sigma_m^{\setminus}(t)}(t)\right]$, for any $k \in Z(t)$.

So if there exists m_0 and k_0 such that $E_{k|\sigma_m^{\backslash}(t)}(t) < \max_{stoch}\limits_{j\in\left\{Pred\left(k|\sigma_m^{\backslash}(t)\right)\right\}} C_j(t)$ then the design task k_0 and all its successors are added to $J(G_t)\backslash Z(t)$ and removed from the sequences assigned for the engineers, and implicitly from $Z(t)$.

3.2 (Re-)Scheduling of design tasks

The project controller of a design team will schedule to the engineers the design tasks from the set $J(G_t)\backslash Z(t)$ updated by the re-scheduling decision function. This is done mainly to minimize the expected makespan, because we want to avoid the delays that may occur due to the precedence constraints relating $J(G_t)\backslash Z(t)$ to the set of concurrent planned design tasks to be allocated next to the engineers.

As a result of solving the scheduling problem, we will obtain for each engineer m an optimal sequence $\sigma_m^{\backslash\backslash}(t)$ of design tasks. In front of this sequence we will add the sequence $\sigma_m^{\backslash}(t)$ of the design tasks on progress from previous control periods which at the beginning of review period t had the performance levels unchanged, and their previous scheduling satisfied the precedence relationships by means of stochastic ordering.

An instance of our type of scheduling problem is given by the directed, acyclic subgraph of G_t spanned by $J(G_t)\backslash Z(t)$. We assume that the engineer cannot solve more than one design task at the same time and no preemption is allowed. Then our (re-)scheduling problem is a variant of the known stochastic identical parallel machine scheduling with non-unit jobs and arbitrary precedence constraints aiming to minimize the expected maximum completion time. This problem was solved analytically for tree-like precedence constraints (see for a review Weiss, 1995), while the research of Foulds et al. (1991) and Neumann and Zimmermann (1998) gives polynomial heuristics even in more general types of precedence constraints.

Decision variables:

$\sigma_m^{\backslash\backslash}(t)$: the sequence of design tasks from $J(G_t)\backslash Z(t)$ scheduled for the engineer m in the beginning of review period t; $m = 1, ...M$.

The detailed planning level (re-)scheduling problem can be formulated as follows:

$$\min_{\left(\sigma_1^{\backslash\backslash}(t),...,\sigma_M^{\backslash\backslash}(t)\right)} \left\{ \max_{m=1,...,M} E\left[C_{\left|\sigma_m^{\backslash}(t)\right| |\sigma_m^{\backslash\backslash}(t)}(t) \right] \right\} \tag{6}$$

subject to

$$\forall k \in 1, ..., \left|\sigma_m^{\backslash\backslash}(t)\right|,\ \forall m = 1, ..., M$$
$$E_{k|\sigma_m^{\backslash\backslash}(t)}(t) \geq \max_{stoch} \left\{ C_j(t),\ j \in Pred\left(k|\sigma_m^{\backslash\backslash}(t)\right) \right\} \tag{7}$$

$$\forall n \in J(G_t) \setminus Z(t), \exists! m \in 1, ..., M \text{ s. t. } n \in \sigma_m^{\setminus\setminus}(t) \tag{8}$$

where

$$C_{k|\sigma_m^{\setminus\setminus}(t)}(t) := \sum_{i \in \sigma_m^{\setminus}(t)} \left[S_{i|\sigma_m^{\setminus}(t)}(t) \right] + \sum_{i=1}^{k} \left[S_{i|\sigma_m^{\setminus\setminus}(t)}(t) \right] \tag{9}$$

$$E_{k|\sigma_m^{\setminus\setminus}(t)}(t) := \sum_{i \in \sigma_m^{\setminus}(t)} \left[S_{i|\sigma_m^{\setminus}(t)}(t) \right] + \sum_{i=1}^{k-1} \left[S_{i|\sigma_m^{\setminus\setminus}(t)}(t) \right] \tag{10}$$

The condition 8 requires that each design task to be (re-)scheduled at the beginning of review period t has to be allocated to exactly one engineer.

3.3 Allocation of concurrent design tasks

After the (re-)scheduling, the design team project controller allocates the set $Y(t)$ of concurrent (planned or newly arrived) design tasks to engineers.

The allocation of design tasks to engineers maximizes the value of the allocated design tasks, subject to the following constraints : each design task is either allocated to exactly one engineer or is not allocated at all, and resources (engineers) have to work close to their corresponding optimal work pressure levels. This multiple choice knapsack problem is specific for NPD projects under hard time constraint and was solved in Dragut (2002). The stochasticity and the fact that to each engineer we can assign more than one design task, do not allow neither a mixed-integer formulation of the problem, nor a simpler formulation of the partial solutions to be eliminated, as required by more efficient algorithms (Ibaraki et al., 1978; Dyer et al.,1995).

Additional notation (at the beginning of review period t**):**

$v(A,t) := \sum_{n \in A} V(n,t)$: the cumulated value of the design task set $A \in P(Y(t))$; $u(\emptyset) = 0$;

$N_a(A,t) := \sum_{n \in A} \sum_{i=1}^{\hat{l}(n,t)} A(n,t,i)$: the total number of activities of the design tasks in the set $A \in P(Y(t))$;

$\overline{p}_h(A,t) := \Pr\left\{ \sum_{n \in A} \widehat{S}_n(t) < h \right\}$: the probability of solving the design tasks in time, where $A \in P(Y(t))$, and $\widehat{S}_n(t)$ are sums of $N_a(\{n\},t)$ i.i.d. exponential random variables of mean μ; each represents the duration of a design task k. Since the mean μ is the same for the activities of all tasks,

$$\overline{p}_h(A,t) = 1 - \sum_{i=0}^{N_a(A,t)-1} \frac{(\mu h)^i \exp(-\mu h)}{i!}$$

We also have to say that $\overline{p}_h\left(\{n_1,...,n_{|A|}\}\right)$ is easily computable from the number of activities of each task $n_i \in A$: we have assumed that the mean of the exponentials (for the activities) is the same for all tasks and thus we have again an Erlang distribution. This is also the reason we keep the design task's control at activity level.

Definition 2 *Given $Y(t) \neq \emptyset$ we say that $\pi_m = (A_1, A_2, ..., A_m)$ is an m-partition of the set $Y(t)$ if $\bigcup_{i=1}^{m} A_i = X$, $A_i \cap A_j = \emptyset$, and $A_i \neq \emptyset$ for all $i \neq j$, $i, j = 1, ..., m$.*

Under the assumption that we have enough design tasks to be allocated to the engineers, the allocation problem is:

$$\max_{\pi_{M+1}=(A_1,A_2,...,A_{M+1})\in\Pi_{M+1}} \left(\sum_{m=1,...,M} v\left(A_m,t\right)\right) \tag{A0}$$

where
$\Pi_{M+1} := \{\pi_{M+1}$ an $M+1$ partition of $Y(t)|. \left|\overline{p}_h\left(A_i,t\right) - \alpha\left(i\right)\right| < \varepsilon,\, i = 1,...,M\}$.
The set A_{M+1} will contain all the design tasks that we failed to allocate into the short time-planning horizon, to the M engineers of the team considered.

If after solving the aggregate level problem, there are not enough design tasks to be allocated to engineers, the engineers may receive design tasks from other projects to ensure their efficiency. From this project point of view they will have zero-value, and hence will be the last to be allocated.

4 Engineering process

During the review period t, $(t, t+1]$, the engineers work concurrently, each on its corresponding sequence of design tasks determined at the beginning of review period t in the detailed planning. Each design task has been allocated to only one engineer.

Input parameters (at the beginning of review period t):
$\sigma_m\left(t\right)$: the sequence of design tasks allocated to the engineer m during the detailed planning; $m = 1, ..., M$;

$\Upsilon(t) := \bigcup_{m=1}^{M} \{k|\, k \in \sigma_m\left(t\right)\}$: the set of all design tasks allocated to the team of engineers;

$A\left(n,t,l\right)$: the number of activities planned for solving the design task n, at the performance level l, assuming the previous levels already solved; $l = 1, ..., L$, $n \in \Upsilon(t)$;

$\hat{l}\left(n,t\right)$: the level at which the design task n must be performed, as established at the aggregate decision level, $n \in \Upsilon(t)$;

For solving a design task, n, we may assume that an engineer solves sequentially its planned activities, for each level up to $\hat{l}\left(n,t\right)$ (Dragut, Bertrand, 2002).

But, an engineer decision on choosing or not a specific design task to work on remains uncertain due to the large variety of variables implied and due to the lack of empirical data to support theories. Unlike machines, human beings perceive the concurrency and the relative urgency of design tasks. The sequence of design tasks allocated to an engineer, may contain more than one design task allocated at the beginning of the same review period. Those design tasks can be performed in parallel and their order in a sequence reflect only the optimality criteria of the scheduler from the detailed planning level.

During the solving process of the design tasks, other several disturbances concerning new activities arrival to the design tasks in progress, and addition/deletion of design tasks in the NPD project may occur (review: Dragut, Bertrand, 2002). The new activities are a result of the incapacity of solving the design task with its current description, so they have preemptive resume priority over the planned activities (Dragut, Bertrand, 2002).

All those uncertainties influence the execution of the schedule. Therefore, for the next period, at the aggregate decision level the controller will take into consideration the current shop status at the current review period end. This fact is consistent with the real life situations where, on a weekly basis, each engineer measures how much time was spent on solving each design task, what activities were solved, what activities were added. Thus, at the beginning of review period $t+1$, the output variables from the engineering process are:

$A(n,t+1,l)$: the number of activities of design task n , if design task n is performed at level l, assuming the previous levels already solved; $l=1,...,L$, $n \in \Upsilon(t)$;

5 Expected NPD project outcome

Before starting a new review period, the aggregate controller must update the previous network of design tasks according to the technological changes occurred (deletion/addition of new design tasks, arrival of new activities for the design tasks on which the engineers have worked). Based on previous levels mathematical description, one can link the review periods using approximate solutions for the detailed planning and engineering level problems, and assuming review period-dependent Poisson arrival processes of new design tasks with a common contribution function and identical performance level structure. Such a linkage mechanism can be further used to predict the expected NPD project outcome in terms of market payoff, and to derive optimal policies for achieving it.

Simple priority rules may schedule detailed planning level design task, giving an ordering among all the planned activities included in all the design tasks (Neumann, Zimmerman 1999; Dragut, Dragut, 2001). Thus, at the beginning of each stage t the linkage problem gives a queueing system with M parallel servers, and a common queue of $\widetilde{A}(t)=\sum\limits_{n\in Y(t)\cup Z(t)}\sum\limits_{l=l(n,t)+1}^{l(n,t+1)}A(n,t,l)$ planned activities with $Exp(\mu)$ distributed processing times. The solving process is disturbed by a $\lambda(t)$-Poisson arrival of unplanned activities ($\lambda(t)/\mu < M$) with preemptive resume priority. This simplification allowed the computation of transition probabilities of a Markovian decision process for the one customer need aggregate control of time constrained NPDs (Dragut, Dragut, 2001). Optimal policies can be derived for the case of more customer needs under similar conditions. Also, approximate policies can be derived using the neuro-dynamic techniques for finite-horizon nonstationary Markov decision processes (Garcia, Ndiaye, 1998).

6 Conclusions

Managing the new product definition is a complex managerial task. Based on recent research, this paper proposes a mathematical formulation of the NPD project management, which includes two new problems specific for the NPD projects under hard time constraint.

We model the real-life project, and formalize the quality-time-cost trade-offs underlying the NPD project mainly from the technological uncertainty point of view. This paper goal is to close the gap in between the mathematical models world and management practice providing a basis from which one can derive the constraints for computational models. Thus, it is facilitated the evaluation and acceptance of the computational results by the management practitioners.

References:

1. Askin, R.G., Dawson, D.W., 2000. Maximizing customer satisfaction by optimal specification of engineering characteristics. *IIE Transactions 32*, 9-20.

2. Best, J.B., 1995, Cognitive Psychology. West Publishing Company, St. Paul, Minneapolis.

3. Bhattacharya, S., Krishnan, V., Mahjan, V., 1998. Managing new product definition in highly dynamic environments. *Management Science 44,* 50-64.

4. Bowers, C.A., Braun, C.C., Morgan Jr., B.B., 1997. Team Workload: its meaning and measurement. In: M.T. Brannick, E. Salas, C. Prince, (Eds.) Team Performance Assessment and Measurement: theory, research, and applications. Lawrence Erlbaum Associates Inc. Publishers, New Jersey, pp. 85-108.

5. Brown, S.L., Eisenhardt, K.M., 1995. Product development: past research, present findings, and future directions. *Academy of Management Review 20,* 343-378.

6. Colajanni, M., Lo Presti, F., Tucci, S., 2000. A hierarchical approach for bounding the completion time distribution of stochastic task graphs. *Performance Evaluation 41,* 1-22.

7. Dragut, A.B. , Dragut, M., 2001. Upper Bound Decision Process for a Time-Constrained Radical New Product Development Process (TR-NPD)- A Finite Horizon Discrete-Time Markov Nonstationary Approach.In: Govaert, G., Janssen, J., Limnios, N. (Eds.) Proceedings of the X-th International Symposium on Applied Stochastic Models and Data Analysis, Imp. Finet-Compiègne, France, pp. 392-397.

8. Dragut, A.B., 2002. Heuristic allocation based on a dynamic programming state space representation. *Journal of Computational and Applied Mathematics*, 140 (1-2), 257-273.

9. Dragut, A.B., Bertrand, J.W.M., 2002. A General Framework for Controlling Time Constrained NPD Projects, submitted to *International Transactions in Operational Research.*

10. Dyer, M.E., Riha W.O., and Walker J., 1995, A hybrid dynamic programming/ branch-and-bound algorithm for the multiple-choice knapsack problem, *Journal of Computational and Applied Mathematics* 58, 43-54.

11. Foulds, L.R., Hoffman, D., Neumann, K., 1991. Stochastic identical parallel machine scheduling with OR precedence constraints. *Asia-Pacific Journal of Operational Research 8,* 1-25.

12. Garcia, F., Ndiaye S., 1998. A Learning Rate Analysis of Reinforcement Learning Algorithms in Finite-Horizon, , In J. Shavlik (Eds.) *Proceeedings of the 15-th International Conference on Machine Learning.* Morgan Kaufmann, San Francisco, CA, pp. 215-223

13. Ibaraki, T., Hasegawa, T., Teranaka, K., Iwase, J., 1978. The multiple-choice knapsack problem, *Journal of Operations Research Society Japan* 21, 59-95.

14. Huchzermeier, A., Loch, C.H., 2001. Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D. *Management Science 47,* 85-101.

15. Neumann, K., Zimmerman, J., 1999. Heuristic procedures for parallel-machine scheduling problems with stochastic precedence contraints. *Annals of Operations Research 83,* 115-136

16. Oorschot, van K.E., 2001. Analysing Radical NPD Projects from an Operational Control Perspective. PhD Thesis, Eindhoven University of Technology, The Netherlands.

17. Reed, S.K., 1988. Cognition, Theory and Applications. Brooks/Cole Publishing Company, Pacific Grove, California.

18. Weiss, G., 1995. A Tutorial in Stochastic Scheduling. In P.Chretienne, E.G. Coffman, Jr., J.K. Lenstra, Z. Liu (Eds.) Scheduling Theory and its Applications. John Wiley & Sons, New York, pp. 33-64.

19. Wichers, R.,1996. A theory of individual behavior. Academic Press, London.

20. Yoshimura, M., 1996. Design Optimisation for Product Life Cycle. In G.Q. Huang (Eds.) Design for X. Chapman & Hall, London, pp. 424-440.

Appendix:

If $M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},\hat{l}(\cdot,t)\right)_{1\le n\le N}\right]\overset{def}{:=}\sum_{\delta=1}^{\Delta}w_\delta(t)\left[\sum_{n=1,..,N}\Theta(n,t,\delta)\cdot\frac{\hat{l}(n,t)}{L}\right]$ (Askin, Dawson, 2000) then $\max_{\hat{l}(n,t),\ n\in\{1,\dots,N\}}M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},l(\cdot,t)\right)_{1\le n\le N}\right]=\max_{\hat{l}(n,t),\ n\in\{1,\dots,N\}}\sum_{n=1,..,N}\frac{\hat{l}(n,t)}{L}\cdot\left[\sum_{\delta=1}^{\Delta}w_\delta(t)\cdot\Theta(n,t,\delta)\right]$. An obvious choice for design task value functions is: $V(n,t):=\frac{\hat{l}(n,t)}{L}\cdot\left[\sum_{\delta=1}^{\Delta}w_\delta(t)\cdot\Theta(n,t,\delta)\right],\forall\ n=1,\dots,n\in J(G_t)\cup Y(t)$, where $\hat{l}(n,t)$ is the n-th design task desired performance level.

If $M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},\hat{l}(\cdot,t)\right)_{1\le n\le N}\right]\overset{def}{:=}\prod_{\delta=1}^{\Delta}\left[\overline{S}\left(t,\hat{l}(\cdot,t),\delta\right)\right]^{w_\delta(t)}$ (Yoshimura 1996) we have no direct decomposition indexed by the design tasks numbers.

Since

$$\begin{aligned}&\max_{\hat{l}(n,t),\ n\in\{1,\dots,N\}}M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},\hat{l}(\cdot,t)\right)_{1\le n\le N}\right]\\ \approx\ &\max_{\hat{l}(n,t),\ n\in\{1,\dots,N\}}\log\left\{M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},\hat{l}(\cdot,t)\right)_{1\le n\le N}\right]\right\}\end{aligned}$$

we define $V(n,t):=\log\left\{\prod_{\delta=1}^{\Delta}\left[\overline{S}_\delta\left(t,\hat{l}(n,t)\right)\right]^{w_\delta(t)}\right\}-V_{\min}+\varepsilon\ge\varepsilon>0$ to obtain positive design task value functions, where

$$\exists V_{\min}=\inf_{n\in J(G_t)\cup Y(t),\forall\delta=1,\dots,\Delta}\log\left\{\prod_{\delta=1}^{\Delta}\left[\overline{S}_\delta\left(t,\hat{l}(n,t)\right)\right]^{w_\delta(t)}\right\}\in(-\infty,0)$$

since $\overline{S}_\delta\left(t,\hat{l}(n,t)\right)\in(0,1),\forall n\in J(G_t)\cup Y(t),\forall\delta=1,\dots,\Delta$ represent a finite number of finite values.

If

$$\begin{aligned}LHS&\overset{notation}{:=}\sum_{n\in J(G_t)\cup Y(t)}V(n,t)\\ \le\ &\log\left\{M\left[\left(\Theta(t,\cdot,\delta)_{1\le\delta\le\Delta},\hat{l}(n,t)\right)_{n\in J(G_t)\cup Y(t)}\right]\right\}\overset{notation}{:=}RHS\end{aligned}$$

holds, then the detailed level allocation solution (which maximizes the value of the allocated design tasks) provides an heuristic lower bound for the aggregate decision targets.

$$\begin{aligned} LHS &= \sum_{n\in J(G_t)\cup Y(t)} \sum_{\delta=1}^{\Delta} w_\delta (t) \log \overline{S}_\delta \left(t, \hat{l}\,(n,t)\right) \\ &= \sum_{\delta=1}^{\Delta} w_\delta (t) \sum_{n\in J(G_t)\cup Y(t)} \log \overline{S}_\delta \left(t, \hat{l}\,(n,t)\right) \\ &= \sum_{\delta=1}^{\Delta} w_\delta (t) \log \prod_{n\in J(G_t)\cup Y(t)} \left[\overline{S}_\delta \left(t, \hat{l}\,(n,t)\right)\right] \end{aligned}$$

Since $\overline{S}_\delta (t, l\,(\cdot, t)) \in (0,1), \forall \delta = 1, ..., \Delta$ are normalized $S-$functions $LHS \leq \sum\limits_{\delta=1}^{\Delta} w_\delta (t) \log \left[\overline{S}_\delta \left(t, \hat{l}\,(n_0, t)\right)\right]$. Since $log\,(\cdot)$ and $\overline{S}_\delta (t, l\,(\cdot, t))$ (being $S-$ functions of the $\sum\limits_{n\ =1,..,N} \Theta\,(n,t,\delta) \cdot \frac{\hat{l}(n,t)}{L}$) are increasing functions of the levels:

$$LHS \leq \sum_{\delta=1}^{\Delta} w_\delta (t) \log \left\{ \sum_{n\in J(G_t)\cup Y(t)} \left[\overline{S}_\delta \left(t, \hat{l}\,(n,t)\right)\right] \right\} \leq RHS$$

Figure 1: Stages of the directed acyclic graph describing the state of the system at time instant t_0

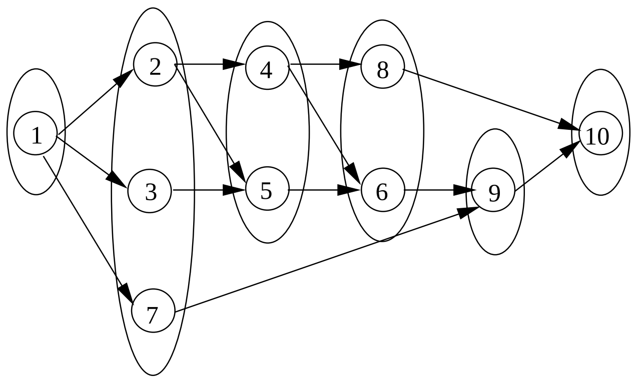

Figure 1: