CEJOR (2014) 22:37–52
DOI 10.1007/s10100-012-0264-5

ORIGINAL PAPER

Allocating slacks in stochastic PERT network

Javier Castro · Daniel Gómez · JuanTejada

Published online: 5 September 2012

Abstract The SPERT problem was defined, in a game theory framework, as the fair allocation of the slack or float among the activities in a PERT network previous to the execution of the project. Previous approaches tackle with this problem imposing that the durations of the activities are deterministic. In this paper, we extend the SPERT problem into a stochastic framework defining a new solution that tries also to maintain the good performance of some other approaches that have been defined for the deterministic case. Afterward, we present a polynomial algorithm for this new solution that also could be used for the calculation of other approaches founded in the deterministic SPERT literature.

Keywords Game theory · Project scheduling · PERT network · Slack allocation · Polynomial algorithms

1 Introduction

Given a PERT network, or a precedence relations among a set of activities, there exist too many real situations in which it is necessary to elaborate (previous to the execution

J. Castro (✉)· D. Gómez
Dpto. de Estadística e I.O. III, Escuela Universitaria de Estadística,
Universidad Complutense, Avda. Puerta de Hierro, s/n, 28040 Madrid, Spain
e-mail: jcastroc@estad.ucm.es

D. Gómez
e-mail: dagomez@estad.ucm.es

J. Tejada
Dpto. de Estadística e I.O. I, Facultad de Matemáticas,
Universidad Complutense, Plaza de Ciencias, s/n, 28040 Madrid, Spain
e-mail: jtejada@mat.ucm.es

of the project and with some time) a tight schedule or a timetable in order to determine when each activity must start. For example, if there exist difficulties to find, in a short period of time, a firm to execute an activity, is convenient to contract the execution of this activity with some time. This lack of firms availability is very common when there exist few firms that are capable of doing the activity satisfying the preferences of the Decision Maker relatives to quality and cost or when, even if there exist many, the demand is quite bigger than the offer in the market.

It is important to note that in these situations (in which there exist availability problems) don't allocate the slack of each activity (to determine when the activity should start) will produce a significant increase in the total cost and/or in the probability of delay the project. On the contrary, when there is not availability problems, a common approach is to use a JIT (Just in time) methodology (see Monden 1983). Let us observe that with this JIT politic, we only fix with the firms, previous of the execution of the project, the activities that can be done in the first place (i.e those activities that do not have predecessors). In the moment in which the decision maker needs someone who make a particular activity, he tries to find an enterprize or firm (see also Dodin 1985; Lida 2000 or Monhor 2010 for more details).

In the literature, the problems of elaborating a tight schedule or a calendar (previous to execution of the project) are often modeled as optimization problems with limited resources (see for example Rogalska et al. 2008; Talbot and Patterson 1978 or Azaron et al. 2005). However, this approach is not appropriate in some real situations for different reasons as: Decision maker has lack of information about costs and resources because the only available information that he has is the PERT network; There exit fixed costs and a fixed number of resources enough to the execution of the project; There exits a necessity to agree with all enterprizes involved in the project and thus a optimization model is not appropriate.

Taking into account that for the two first situations the optimization problem is not too relevant and for the third situation is necessary to agree with all enterprizes, some authors (see Bergantiños and Sánchez 2002a or Castro et al. 2008b) have been focused its attention to find compromise solutions (also referred as rules), based on game theory instead of optimization theory, that permits us to agree with all enterprizes involved in the project.

From mathematical point of view, and without any additional information that the PERT network, the previous problem is equivalent to the problem of assigning in an efficient way the total slack among activities without delay the project (see Castro et al. 2008b). In resource management, the most common approach to assign additional time to each non critical activity (without increasing the completion time of the whole project) is to calculate the possible slack for each activity at the beginning of the project, and then assign slack to the paths which have experienced delays during the execution of the project. This procedure presents some problems that were analyzed and exposed in Castro et al. (2008b), Bergantiños and Lorenzo (2008) or Bergantiños and Sánchez (2002b). Taking into account this problematic and with the aim of present rules that allow us to assign a priori extra time, Bergantiños and Sánchez (2002a) defined the SPERT problem from a game theory point of view.

The SPERT problem consists of the a priori assignment of additional time to each non critical activity without increasing the completion time of the whole project, which is represented in the corresponding PERT network. For the case in which this assignment is *efficient* and the slack of the SPERT problem coincides with the path slack of the PERT network (see for more details Castro et al. 2008b), the allocation of the slack among the activities (SPERT problem) coincides with the elaboration of a tight schedule. In order to allocate the slack, in Bergantiños and Sánchez (2002a), it is proposed two solutions for a Non-Transferable Utility game (or NTU game) based on the compromise value (one of these solutions, the Γ rule, is efficient and so produces a tight schedule). The second reference in the literature referred to rules for SPERT problems is also due to Bergantiños and Sánchez (2002b). In this work they extended the problem to a more general one, the problem with constraints and claims (PCC), and they defined another efficient rule, the Q^c rule. Shragowitz et al. (2003) considered the problem of slack allocation from a computational point of view and proposed a non-efficient algorithm. Recently, Castro et al. (2008b) defined an efficient rule (Q^d rule) to the SPERT problem. In that paper, the Q^d rule was compared with the previous rules showing some clear advantages. The Q^d rule was also characterized in terms of some desirable properties.

It is important to note that all previous rules have been defined only for deterministic PERT networks (i.e., networks in which the only information available is the expected duration of the activities). In this paper we define a rule (the Q^w rule) for stochastic SPERT problems that extends the rule defined in Castro et al. (2008b) maintaining its good performance. Finally, we present a polynomial algorithm that permits us to calculate the Q^w rule.

This paper has been organized as follows. Firstly, the SPERT model and the notation are introduced in Sect. 2. Secondly, a general proportional rule is defined in Sect. 3. Thirdly, a proportional rule in a stochastic context is introduced and discussed in Sect. 4. Fourthly, a polynomial algorithm for proportional rule are defined in Sect. 5. Finally, some conclusions are drawn in Sect. 6.

2 Preliminaries

2.1 PERT networks

A directed graph is a pair $G = (X, N)$ where $X = \{x_1, \dots, x_u\}$ is a finite set of nodes and $N = \{1, \dots, n\}$ is the collection of arcs. An arc $i \in N$ is given by $(x_{i,1}, x_{i,2})$, where $x_{i,1}, x_{i,2} \in X$.

A source is a node $x_b \in X$ such that there is no arc $(x_{i,1}, x_{i,2}) \in N$ with $x_{i,2} = x_b$. A sink is a node $x_e \in X$ such that there is no arc $(x_{i,1}, x_{i,2}) \in N$ with $x_{i,1} = x_e$.

Given a node $x \in X$, we define the sets of immediate predecessors activities of x as $PredIm(x) = \{i \in N / x_{i,2} = x\}$ and the immediate successors activities of x as $SucIm(x) = \{i \in N / x_{i,1} = x\}$. Let us observe that the activities in a PERT network are represented by arcs, and thus the sets of immediate successors and predecessors of a node are refereed to a set of arcs instead of nodes as is usually done in graph theory.

A stochastic PERT Network PE can be defined as a pair (G, D) where $G = (X, N)$ is a directed graph without cycles, a unique source and a unique sink, N is the set of arcs representing the set of activities in the project, and $D = (D_i)_{i \in N}$ is a random vector, being D_i the random variable that describes the duration of activity i. A path π between x and x^* is a collection of arcs $\{i_1, \ldots, i_p\}$ such that $x_{i_1,1} = x$, $x_{i_p,2} = x^*$ and $\forall k \in \{1, \ldots, p-1\}$ $x_{i_k,2} = x_{i_{k+1},1}$. A cycle is a path between x and itself. We denoted by $P(x, x^*)$ the set of all paths between x and x^*. A complete path, is a path between x_b and x_e. We denote by P the set of all complete paths.

An alternative definition of stochastic PERT network is given by a 3-tuple (N, P, D), where $N = \{1, \ldots, n\}$ is the set of activities or arcs of the graph, $P = \{\pi_1, \ldots, \pi_p\}$ is the set of complete paths in the graph and $D = (D_1, \ldots, D_n)$, is a random vector.

In the literature, it is usual to assume that the D_i is defined in terms of three parameters: the optimistic completion time (a_i), the pessimistic completion time (b_i), and the modal completion time (m_i). Obviously, we assume that $b_i > D_i > a_i \geq 0$.

The most common random distribution for these $\{D_i\}_{i=1,\ldots,n}$ variables are:

- Uniform distribution $U(a, b)$:

$$f(t) = \begin{cases} \frac{1}{b-a} & \text{if } a < t < b \\ 0 & \text{otherwise} \end{cases}$$

- Triangular distribution $T(a, m, b)$:

$$f(t) = \begin{cases} \frac{2}{(m-a)(b-a)}(t-a) & \text{if } a < t \leq m \\ \frac{2}{(m-b)(b-a)}(t-b) & \text{if } m < t < b \\ 0 & \text{otherwise} \end{cases}$$

- Beta distribution $\beta(a, \alpha, \varphi, b)$:

$$f(t) = \begin{cases} 0 & \text{if } t \leq a \\ \frac{(t-a)^\alpha (b-t)^\varphi}{\int_a^b (t-a)^\alpha (b-t)^\varphi dt} & \text{if } a < t < b \\ 0 & \text{if } t \geq b \end{cases}$$

- Normal distribution $N(\mu, \sigma)$:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \quad -\infty < t < \infty$$

We will denote by d_i the expected duration of the activity i and $d = (d_1, \ldots, d_n)$ the expected duration vector. If $d_i = 0$, then i is a fictitious activity which only indicates a precedence relation among activities. We denote the set of fictitious activities by $\widehat{N}$. The duration of a path π, denoted by d_π, is the sum of the durations of the activities in π, i.e. $\sum_{i\in\pi} d_i$.

A deterministic PERT Network is a stochastic PERT Network in which the only information available is the expected duration of the activities. We denote a deterministic PERT Network by (G, d) or (N, P, d).

The PERT time, T, is defined as the minimum time that the project needs to be finished, i.e. $T = \max_{\pi\in P} d_\pi$. Given a node $x \in X$, we define the earliest time of this node, T_x^E, as the minimum time required to finish all predecessor activities of the node x, i.e. $T_x^E = \max_{\pi \in P(x_b,x)} d_\pi = \max_{\{i \in PredIm(x)\}} (T_{x_{i,1}}^E + d_i)$. Given a node $x \in X$, we define the latest time of this node, T_x^L, as the maximum time required to finish all predecessor activities of the node x without delaying the project, i.e. $T_x^L = T - \max_{\pi \in P(x,x_e)} d_\pi = \min_{\{i \in SucIm(x)\}} (T_{x_{i,2}}^L - d_i)$. Obviously, $\forall\ x \in X$, $T_x^E \leq T_x^L$.

Given a complete path $\pi \in P$, we define the slack of π, ps_π, as the maximum time, in addition to d_π, that all the activities in π can use without delaying the project, i.e. $ps_\pi = T - d_\pi$. If $ps_\pi = 0$ then π is a critical path. We denote by $PS = (ps_1, \ldots, ps_p)$, the path slack vector. Given an activity $i \in N$, we define the activity slack as the maximum time, in addition to d_i that activity i can use without delaying the project, i.e. $as_i = \min_{\pi \in P/i \in \pi} ps_\pi = T_{x_{i,2}}^L - T_{x_{i,1}}^E - d_i$. If $as_i = 0$ then i is a critical activity. We denote by $AS = (as_1, \ldots, as_n)$, the maximum slack vector.

2.2 SPERT problem

A SPERT is given by a 2-tuple (PE, S), where:

- $PE = (N, P, D)$ is a stochastic PERT network.
- $S = (S_1, \ldots, S_p)$, is the path slack vector.

By default, $S = PS$. Given $M \subset N$, we will denote by $N|_M = M$, $P|_M = \{\pi \cap M/\pi \in P \text{ and } \pi \cap M \neq \emptyset\}$, $D|_M = \{D_i/i \in M\}$ and $S|_M = \{S_\pi/\pi \in P \text{ and } \pi \cap M \neq \emptyset\}$.

A feasible allocation for the SPERT problem is a vector $(x_i)_{i \in N} \in \Re^n$ such that $x_i \geq 0, \forall i \in N$ and $\sum_{i \in \pi} x_i \leq S_\pi, \forall \pi \in P$, and the set of feasible allocations is $F(PE, S)$. A rule is a function f that assigns to any problem (PE, S) a feasible allocation, i.e., $f(PE, S) \in F(PE, S)$.

Given a SPERT problem (PE, S), we define the maximum slack activity vector $AS(PE, S) = (as_1(PE, S), \ldots, as_n(PE, S))$ as follows: $as_i(PE, S) = \min_{\{\pi \in P/\ i \in \pi\}} S_\pi$. Let us observe that if $S = PS$ then $AS(PE, S) = AS$.

In order to show what is the relation between the SPERT problem and the tight schedule for PERT networks we introduce the following remark.

Remark 1 A tight schedule could be defined from a solution of the SPERT problem if its solution belongs to the Pareto boundary of $F(PE, PS)$ (the Pareto boundary of a set $Y \subset \Re^n$ is defined as $PB(Y) = \{y \in Y\ /\ \forall x \in \Re^n,\ x_i \geq y_i$ and $x \neq y$, then $x \notin Y\}$) and $S = PS$. In this schedule, a time window is assigned to each activity in such a way that if the activities durations are substituted by the length of the time window, all of them become critical. The tight schedule for the project could be defined according to the following steps:

1. Model the project as a SPERT problem with $S = PS$.
2. Find a solution x of the SPERT problem using a rule that guarantees that x belongs to the Pareto boundary.
3. Calculate the new duration for each activity i, $d_i^* = d_i + x_i$.
4. With these new durations, calculate the earliest start and earliest finish times for each activity.

The schedule can be formulated at the beginning of the project or at any moment in time during its execution. This flexibility is possible because the activities yet to be performed can always be represented in a PERT network. Consequently, a schedule based on this reduced PERT network can be prepared. Thus, a tool which is capable of producing project schedules is created for any point in time during its execution.

3 The proportional to the weights rule

In Castro et al. (2008b) it was defined and analyzed the $Q^d(PE, S)$ rule for $SPERT$ problems in deterministic $PERT$ networks. In that paper, the deterministic durations of the activities, $(d_i)_{i=1,\ldots,n}$, was used to measure the *a priori* importance of each of the activities. In a stochastic framework, it is necessary to consider more information to measure the importance of each activity. To this reason, in this section, we will extend the $Q^d(PE, S)$ rule considering a positive weight for non-fictitious activities, $W(PE, S) = (w_1(PE, S), \ldots, w_n(PE, S))$, that depends on the stochastic $SPERT$ problem, (PE, S).

From now on, and where there is non ambiguity with respect to $W(PE, S) = (w_1(PE, S), \ldots, w_n(PE, S))$ we will denote this parameter as $W = (w_1, \ldots, w_n)$.

Following the ideas of proportionality defined in Castro et al. (2008b), the $Q^w(PE, S)$ rule for slack allocation that we introduce in this section are proportional to a weight W. That is, if only one path π with slack exists, this slack must be distributed according to the following equation:

$$R_i^w = \frac{w_i}{\sum_{j \in \pi \cap (N \backslash \widehat{N})} w_j} S_\pi, \forall i \in \pi \cap (N \backslash \widehat{N}). \quad where \quad w_i > 0 \quad \forall i \in N \backslash \widehat{N}.$$

In a more general case, given a SPERT problem we have to consider the reduced SPERT problem (PE^1, S^1) where: $N^1 = \{i \in N \backslash \widehat{N} \;/\; as_i(PE, S) > 0\}$, $P^1 = P|_{N^1}$, $D^1 = D|_{N^1}$ and $S^1 = S|_{N^1}$.

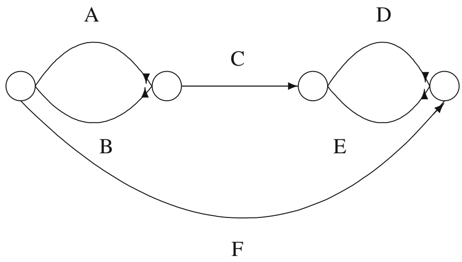
This reduced problem only contains activities that are not critical and not fictitious, that is, those activities that can receive extra time for their execution. For this reduced problem, a feasible solution that allocates the slack proportional to the weight is given by: $R^w(PE^1, S^1) = \left(R_i^w(PE^1, S^1)\right)_{i \in N^1}$ where $R_i^w(PE^1, S^1) = \lambda w_i, \ \forall i \in N^1$ and $\lambda = \max\{r \in \Re / rW^1 \in F(PE^1, S^1)\}$. The value of λ is $\lambda = \min_{\pi \in P^1} \frac{S_\pi}{\sum_{i \in \pi} w_i}$ as was proved in Castro et al. (2008b).

As happened with the $R^d(PE, S)$ rule for deterministic PERT networks, is necessary to provide an iterative algorithm to define the $Q^w(PE, S)$ rule in order to have a tight schedule.

The $Q^w(PE, S)$ rule

Step 1: Given a SPERT problem (PE, S), we define (PE^1, S^1) as $N^1 = \{i \in N \backslash \widehat{N} / as_i(PE, S) > 0\}$, $P^1 = P|_{N^1}$, $D^1 = D|_{N^1}$ and $S^1 = S|_{N^1}$ and $W = (w_1, \ldots, w_n)$. For each $i \in N^1$, we calculate: $R_i^w(PE^1, S^1) = \lambda^1 w_i$, where

Fig. 1 Network of Example 1



$$\lambda^1 = \min_{\pi \in P^1} \frac{S^1_\pi}{\sum_{i\in\pi} w_i}$$

Assuming that (PE^t, S^t) are known $\forall t \le k$: We define (PE^{k+1}, S^{k+1}) as:

Step k+1:

$N^{k+1} = \{i \in N^k \;/\; as_i(PE^k, S^k) > 0\}$
$P^{k+1} = P^k|_{N^{k+1}}$
$D^{k+1} = D^k|_{N^{k+1}}$
$S^{k+1}_{\pi^{k+1}} = S^k_{\pi^k} - \sum_{i\in\pi\cap N^k} R^w_i(PE^k, S^k),\ \forall\, \pi^{k+1} \in P^{k+1}$

For each $i \in N^{k+1}$, we calculate $R^w_i(PE^{k+1}, S^{k+1}) = \lambda^{k+1} w_i$

Final step: Computing the solution.
Stop in step $T \ge 1$ in which $N^T \ne \emptyset$ and $N^{T+1} = \emptyset$.
Compute the solution $Q^w(PE, S)$ as:

$$Q^w_i(PE, S) = \begin{cases} 0 & \text{if } i \in (N\backslash N^1) \\ \sum_{k=1}^{T_i} R^w_i(PE^k, S^k) & \text{if } i \in N^1 \end{cases}$$

where, for each $i \in N^1$, T_i $(1 \le T_i \le T)$ verifies $i \in N^{T_i}$ and $i \notin N^{T_i+1}$.

Example 1 Let us consider the PERT network in Fig. 1, where $N = \{A, B, C, D, E, F\}$, $P = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5\}$ where $\pi_1 = \{A, C, D\}, \pi_2 = \{A, C, E\}, \pi_3 = \{B, C, D\}, \pi_4 = \{B, C, E\}$ and $\pi_5 = \{F\}$, $D = (D_A, D_B, D_C, D_D, D_E, D_F) = (U(1,3), \beta(0,1,1,2), U(2,4), \beta(0,2,2,6), U(3,5), U(9,11))$ and $S = (S_{\pi_1}, S_{\pi_2}, S_{\pi_3}, S_{\pi_4}, S_{\pi_5}) = (2,1,3,2,0)$ and $W = (w_A, w_B, w_C, w_D, w_E, w_F) = (2,2,2,6,2,2)$.

In the first step, (N^1, P^1, D^1, S^1) is given by: $N^1 = \{A, B, C, D, E\}$, $P^1 = \{\{A, C, D\}, \{A, C, E\}, \{B, C, D\}, \{B, C, E\}\}$, $D^1 = (D_A, D_B, D_C, D_D, D_E)$, $S^1 = (2, 1, 3, 2)$ and $\lambda^1 = \min\{\frac{2}{10}, \frac{1}{6}, \frac{3}{10}, \frac{2}{6}\} = \frac{1}{6}$, therefore $R_i(N^1, P^1, D^1, S^1) = \frac{1}{6} w_i, \forall i \in N^1$.

In the second step, (N^2, P^2, D^2, S^2) is given by: $N^2 = \{B, D\}$, $P^2 = \{\{D\}, \{B, D\}, \{B\}\}$, $D^2 = (D_B, D_D)$, $S^2 = (2 - \frac{5}{3}, 3 - \frac{5}{3}, 2 - 1)$ and $\lambda^2 = \min\{\frac{1/3}{6}, \frac{4/3}{8}, \frac{1}{2}\} = \frac{1}{18}$ therefore $R_i(N^2, P^2, D^2, S^2) = \frac{1}{18} w_i, \forall i \in N^2$.

In the third and final step, (N^3, P^3, D^3, S^3) is given by: $N^3 = \{B\}$, $P^3 = \{\{B\}, \{B\}\}$, $D^3 = (D_B)$, $S^3 = (3 - \frac{5}{3} - \frac{8}{18}, 2 - 1 - \frac{2}{18})$ and $\lambda^3 = \frac{4}{9}$ therefore $R_B(N^3, P^3, D^3, S^3) = \frac{8}{9}$. The solution is $Q^w = (Q^w_A, \ldots, Q^w_F) = (\frac{1}{3}, \frac{4}{3}, \frac{1}{3}, \frac{4}{3}, \frac{1}{3}, 0)$.

4 On the election of the weight

In this section, we will find possible weights for $Q^w(PE, S)$ that permit us to guarantee some desirable properties of the proposed rule.

First at all, it is important to note that a necessary condition for the weight is the additivity (see below) in order to guarantee that the $Q^w(PE, S)$ rule satisfies the properties for slack allocation rules defined in Castro et al. (2008b). So, we will only consider additive weights. In order to define the additivity property for weights we introduce the following notation:

Let us observe that an activity can be split into several subactivities (or the opposite, several subactivities can be merged in one) in such a way the original random variable coincides with the sum of the random variables associated with its subactivities. Formally, we consider the SPERT problems (PE, S) and (PE^*, S^*) where: 1) $PE^* = (M, P|_M, D^*)$, $S^* = S|_M$, $M \subset N$, 2) there exists only one activity $k \in M$ such that $D^*_k = D_k + \sum_{j \in (N \setminus M)} D_j$, 3) $D^*_i = D_i \; \forall i \in M \setminus \{k\}$ and 4) $\forall \pi \in P$, $(\{k\} \cup (N \setminus M)) \subset \pi$ or $(\{k\} \cup (N \setminus M)) \cap \pi = \emptyset$.

Additivity (AD): Given the SPERT problems (PE, S) and (PE^*, S^*) as have been defined previously, we will say that the $W(PE, S)$ weight satisfies the additivity property if:

$$w_k(PE^*, S^*) = w_k(PE, S) + \sum_{j \in N \setminus M} w_j(PE, S) \quad and$$
$$w_i(PE^*, S^*) = w_i(PE, S) \quad \forall i \in M \setminus \{k\}.$$

It is easy to prove that the following weights verify the additivity property: the minimum and maximum values of D_i (a_i and b_i), the expected mean of D_i (d_i), the range of D_i ($b_i - a_i$) and, assuming independence among random variables, the variance of $D_i (Var(D_i))$.

In order to discriminate between the possible weights that are additive we introduce the *Symmetric Risk* property for the SPERT allocation rules. This property requires that when two activities are symmetric in the PERT network (two activities $i, j \in N$ are symmetric in the PERT network if $\forall \pi \in P \; i \in \pi$ if and only if $j \in \pi$), the risk of delay is the same.

Symmetric Risk (SR): We will say that the $f(PE, S)$ rule satisfies the Symmetric Risk property if for all (PE, S) and $\forall i, j \in N$, symmetric activities in the PERT network, the following holds:

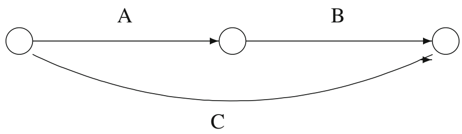
$$P(D_i \geq d_i + f_i(PE, S)) = P(D_j \geq d_j + f_j(PE, S)).$$

It is important to emphasize that when a rule does not satisfy this SR property it is possible that some activities receive more slack that they need and there are

Table 1 Solution of Example 2

	$w_i = a_i$	$w_i = b_i$	$w_i = d_i$	$w_i = Var(D_i)$	$w_i = b_i - a_i$
f_A	0	85.714	75	112.5	100
f_B	150	64.286	75	37.5	50
f_C	0	0	0	0	0

Fig. 2 Network of Example 2



other activities that receive less. We can see this pathology in the following example:

Example 2 Let (PE, S) be the $SPERT$ problem define as the PERT network in Fig. 2, $N = \{A, B, C\}$; $P = \{\{A, B\}, \{C\}\}$; $D = (\beta(0, 100, 200), \beta(50, 100, 150), \beta(300, 350, 400))$ and $S = (150, 0)$.

In Table 1, we show the final allocation given by the $Q^w(PE, S)$ rule using the additive weights minimum, maximum, expected mean, variance and range.

Let us observe that all weights except the range give more slack that $b_i - d_i$ (maximum slack that the activity i could need) for one of the two activities. In this example the range is the only weight that satisfies the *SR* property.

It is possible to extend this good behavior of the rule when we chose the range to a more general class of SPERT problems as we show in the following proposition.

Proposition 1 *Let (PE, S) be a SPERT problem. If, all random variables $\{D_i\}_{i=1,\ldots,n}$ have uniform distribution or all have triangular distribution with $m_i = a_i + k(b_i - a_i)$ and $k \in [0, 1]$, then, the Q^w rule satisfies the Symmetric Risk property if and only if the weight vector is proportional to the range, i.e. $w_i = \alpha(b_i - a_i)$ with $\alpha > 0$, $\forall i = 1, \ldots, n$.*

Proof It is important to note, that if two activities $i, j \in N$ are symmetric in the PERT network, then $Q_j^w w_i = Q_i^w w_j$ and thus $Q_i^w = \lambda w_i$ and $Q_j^w = \lambda w_j$. Following previous equality and SR definition we have:

$$\begin{aligned} P(D_i \geq d_i + Q_i^w(PE, S)) \quad &= \quad P(D_j \geq d_j + Q_j^w(PE, S)) \\ &\Longleftrightarrow P(D_i \geq d_i + \lambda w_i) = P(D_j \geq d_j + \lambda w_j). \end{aligned}$$

Now, this last equality are equivalent to

$$F_{D_i}(d_i + \lambda w_i) = F_{D_j}(d_j + \lambda w_j) \tag{1}$$

$F_{D_i}(t)$ being the distribution function of the random variable D_i. The rest of the proof are divided in two cases (uniform and triangular distribution):

1. If all random variables $\{D_i\}_{i=1,\ldots,n}$ have uniform distribution and taking into account that the uniform distribution function is

$$F_{U(a,b)}(t) = \begin{cases} 0 & t \leq a \\ \frac{t-a}{b-a} & a < t < b \\ 1 & t \geq b \end{cases},$$

   and $F_{U(a,b)}(t) = F_{U(0,1)}(\frac{t-a}{b-a})$, then (1) holds if and only if:

$$F_{U(0,1)}\left(\frac{d_i + \lambda w_i - a_i}{b_i - a_i}\right) = F_{U(0,1)}\left(\frac{d_j + \lambda w_j - a_j}{b_j - a_j}\right) \Longleftrightarrow \frac{d_i + \lambda w_i - a_i}{b_i - a_i} = \frac{d_j + \lambda w_j - a_j}{b_j - a_j} \Longleftrightarrow \frac{a_i + \frac{b_i - a_i}{2} + \lambda w_i - a_i}{b_i - a_i} = \frac{a_j + \frac{b_j - a_j}{2} + \lambda w_j - a_j}{b_j - a_j} \Longleftrightarrow \frac{w_i}{b_i - a_i} = \frac{w_j}{b_j - a_j}$$

   Let us observe that this last equality only holds if the weight $w_i = \alpha(b_i - a_i) \quad with \quad \alpha > 0 \, \forall i = 1, \ldots, n.$
2. If all random variables $\{D_i\}_{i=1,\ldots,n}$ have triangular distribution with $m_i = a_i + k(b_i - a_i)$ and taking into account that this triangular distribution function is

$$F_{T(a,a+k(b-a),b)}(t) = \begin{cases} 0 & t \leq a \\ \frac{(t-a)^2}{k(b-a)^2} & a < t \leq a + k(b-a) \\ 1 - \frac{(b-t)^2}{(1-k)(b-a)^2} & a + k(b-a) < t < b \\ 1 & t \geq b \end{cases}$$

   and $F_{T(a,a+k(b-a),b)}(t) = F_{T(0,k,1)}(\frac{t-a}{b-a})$, then (1) holds if and only if:

$$F_{T(0,k,1)}\left(\frac{d_i + \lambda w_i - a_i}{b_i - a_i}\right) = F_{T(0,k,1)}\left(\frac{d_j + \lambda w_j - a_j}{b_j - a_j}\right) \Longleftrightarrow \frac{d_i + \lambda w_i - a_i}{b_i - a_i} = \frac{d_j + \lambda w_j - a_j}{b_j - a_j} \Longleftrightarrow \frac{a_i + \frac{(1+k)(b_i - a_i)}{3} + \lambda w_i - a_i}{b_i - a_i} = \frac{a_j + \frac{(1+k)(b_j - a_j)}{3} + \lambda w_j - a_j}{b_j - a_j} \Longleftrightarrow \frac{w_i}{b_i - a_i} = \frac{w_j}{b_j - a_j}.$$

   Let us observe that this last equality only holds if the weight $w_i = \alpha(b_i - a_i) \quad with \quad \alpha > 0 \, \forall i = 1, \ldots, n.$

From now on we have taken the range for the Q^w rule taking into account its good behavior.

Remark 2 In Castro et al. (2008b), it is defined some desirable properties and two characterizations for the Q^d rule in deterministic SPERT problems. We can extend in a trivial way these properties and characterizations to the case in which the SPERT problem is stochastic and it easy to see that the Q^w rule with $w_i = b_i - a_i$ verifies these properties.

Remark 3 Although it is unreal to suppose that $b_i = a_i$ (taking into account that random variables model the time duration of an activity and they are continuous), from mathematical point of view we could solve this problematic with desirable properties redefining the weight w_i as $w_i^\epsilon = (b_i - a_i) + \epsilon d_i$ with $\epsilon > 0$. Obviously, $w_i^\epsilon > 0$ for all activity non-fictitious. Finally, we can calculate Q^w as $\lim_{\epsilon \longrightarrow 0} Q^{w^\epsilon}$. Let us observe that in deterministic PERT networks (in which we don't know the $b_i - a_i$ value) this calculation coincides with Q^d if we eliminate the non informative $b_i - a_i$ from the weight (i.e. $w_i^\epsilon = \epsilon d_i$).

5 A polynomial algorithm

In this section, we define and analyze an algorithm for the Q^w rule (*the Q^w rule algorithm*) when the SPERT problem represents the tight schedule problematic (i.e. $S = PS$).

Q^w rule algorithm; input: $PE = (G, D)$ **and** $W(PE, PS)$**.**

Begin
  k:=0; EndA:=0; $N^0 = \{i \in N \backslash \widehat{N}\}$; $d_i^0 = d_i \quad \forall i \in N$
  Calculate the PERT time (T) of the $PERT$ (G,d^0)
  While EndA=0
  Begin
    k:=k+1
    Calculate the activity slack (as_i^{k-1}) of the $PERT$ (G,d^{k-1})
    $N^k = \{i \in N^{k-1} \,/\, as_i^{k-1} > 0\}$
    Calculate $daux_i^k := \begin{cases} \frac{w_i}{as_i^{k-1}} & \text{if } i \in N^k \\ 0 & \text{otherwise} \end{cases}$
    If $daux_i^k = 0, \ \forall i \in N$ then
      EndA:=1
    Else
    Begin
      t:=1; EndStep:=0
      Find a critical path ($\pi^{k,t}$) in the $PERT$ (G,$daux^k$)
      While EndStep=0
      Begin
        Calculate $\lambda^{k,t} := \dfrac{T - \sum_{i \in \pi^{k,t}} d_i^{k-1}}{\sum_{\{i \in \pi^{k,t} \cap N^k\}} w_i}$
        Calculate $d_i^{k,t} := \begin{cases} d_i^{k-1} + \lambda^{k,t} w_i & \text{if } i \in N^k \\ d_i^{k-1} & \text{otherwise} \end{cases}$
        Calculate the PERT time ($T^{k,t}$) of the $PERT$ (G,$d^{k,t}$)
        If $T^{k,t} = T$ then
        Begin

Calculate $d_i^k := d_i^{k,t},\ \forall i \in N$
EndStep:=1
End
Else
Begin
t:=t+1
Find a critical path ($\pi^{k,t}$) in the $PERT$ (G,$d^{k,t-1}$)
End
End
End
End
$Q_i^w := d_i^k - d_i,\ \forall i \in N$
End

The following example shows how to apply *the Q^w rule algorithm* for the Example 1.

Example 3 First at all, let us observe that we have two loops in the previous algorithm. For the first one we have used the k letter and for the second one the t letter.

With $k = 0$, $N^0 = \{A, \ldots, F\}$, $d^0 = (d_A^0, \ldots, d_F^0) = (2, 1, 3, 3, 4, 10)$; $W = (w_A, \ldots, w_F) = (2, 2, 2, 6, 2, 2)$ and the PERT time is $T = 10$.

With $k = 1$, $AS^0 = (as_A^0, \ldots, as_F^0) = (1, 2, 1, 2, 1, 0)$, $N^1 = \{A, \ldots, E\}$ and $daux^1 = (daux_A^1, \ldots, daux_F^1) = (2, 1, 2, 3, 2, 0)$. In this iteration, for $t = 1$, $\pi^{1,1} = \{A, C, D\}$, $\lambda^{1,1} = \frac{10-(2+3+3)}{2+2+6} = \frac{1}{5}$, $d^{1,1} = (d_A^{1,1}, \ldots, d_F^{1,1}) = (2+\frac{2}{5}, 1+\frac{2}{5}, 3+\frac{2}{5}, 3+\frac{6}{5}, 4+\frac{2}{5}, 10) = (\frac{12}{5}, \frac{7}{5}, \frac{17}{5}, \frac{21}{5}, \frac{22}{5}, 10)$. As the PERT time in the network $(G, d^{1,1})$ is $T^{1,1} = \frac{51}{5} > 10$ we have other iteration, $t = 2$, in which $\pi^{1,2} = \{A, C, E\}$, $\lambda^{1,2} = \frac{10-(2+3+4)}{2+2+2} = \frac{1}{6}$, $d^{1,2} = (d_A^{1,2}, \ldots, d_F^{1,2}) = (2 + \frac{2}{6}, 1 + \frac{2}{6}, 3 + \frac{2}{6}, 3 + \frac{6}{6}, 4 + \frac{2}{6}, 10) = (\frac{7}{3}, \frac{4}{3}, \frac{10}{3}, 4, \frac{13}{3}, 10)$. Now, as the PERT time of the network $(G, d^{1,2})$ is $T^{1,2} = 10$, we finish the iteration $k = 1$ with $d^1 = (\frac{7}{3}, \frac{4}{3}, \frac{10}{3}, 4, \frac{13}{3}, 10)$.

With $k = 2$, $AS^1 = (as_A^1, \ldots, as_F^1) = (0, 1, 0, \frac{1}{3}, 0, 0)$, $N^2 = \{B, D\}$ and $daux^2 = (daux_A^2, \ldots, daux_F^2) = (0, 2, 0, 18, 0, 0)$. In this iteration, with $t = 1$, $\pi^{2,1} = \{B, C, D\}$, $\lambda^{1,1} = \frac{10-(\frac{4}{3}+\frac{10}{3}+4)}{8} = \frac{1}{4}$, $d^{2,1} = (d_A^{2,1}, \ldots, d_F^{2,1}) = (\frac{7}{3}, \frac{4}{3} + \frac{2}{4}, \frac{10}{3}, 4 + \frac{6}{4}, \frac{13}{3}, 10) = (\frac{7}{3}, \frac{11}{6}, \frac{10}{3}, \frac{11}{2}, \frac{13}{3}, 10)$. As the PERT time of the network $(G, d^{2,1})$ is $T^{2,1} = \frac{67}{6} > 10$, we have other iteration, $t = 2$, in which $\pi^{2,2} = \{A, C, D\}$, $\lambda^{2,2} = \frac{10-(\frac{7}{3}+\frac{10}{3}+4)}{6} = \frac{1}{18}$, $d^{2,2} = (d_A^{2,2}, \ldots, d_F^{2,2}) = (\frac{7}{3}, \frac{4}{3} + \frac{2}{18}, \frac{10}{3}, 4 + \frac{6}{18}, \frac{13}{3}, 10) = (\frac{7}{3}, \frac{13}{9}, \frac{10}{3}, \frac{13}{3}, \frac{13}{3}, 10)$. Now, as the PERT time of the network $(G, d^{2,2})$ is $T^{2,2} = 10$, we finish the iteration $k = 2$ with $d^2 = (\frac{7}{3}, \frac{13}{9}, \frac{10}{3}, \frac{13}{3}, \frac{13}{3}, 10)$.

With $k = 3$, $AS^2 = (as_A^2, \ldots, as_F^2) = (0, \frac{8}{9}, 0, 0, 0, 0)$, $N^3 = \{B\}$ and $daux^3 = (daux_A^3, \ldots, daux_F^3) = (0, \frac{4}{9}, 0, 0, 0, 0)$. In this iteration, for $t = 1$, $\pi^{3,1} = \{B, C, D\}$ or $\pi^{3,1} = \{B, C, E\}$, anyway it holds that $\lambda^{3,1} = \frac{10-(\frac{13}{9}+\frac{10}{3}+\frac{13}{3})}{2} = \frac{4}{9}$, $d^{3,1} = (d_A^{3,1}, \ldots, d_F^{3,1}) = (\frac{7}{3}, \frac{13}{9} + \frac{8}{9}, \frac{10}{3}, \frac{13}{3}, \frac{13}{3}, 10) = (\frac{7}{3}, \frac{13}{9}, \frac{10}{3}, \frac{13}{3}, \frac{13}{3}, 10)$. As the PERT time in the network $(G, d^{3,1})$ is $T^{3,1} = 10$ the $k = 3$ iteration finish with $d^3 = (\frac{7}{3}, \frac{13}{9}, \frac{10}{3}, \frac{13}{3}, \frac{13}{3}, 10)$.

With $k = 4$, $AS^3 = (as^3_A, \ldots, as^3_F) = (0, 0, 0, 0, 0, 0)$, $N^4 = \{\emptyset\}$, $daux^4 = (daux^4_A, \ldots, daux^4_F) = (0, 0, 0, 0, 0, 0)$ and thus the algorithm finishes. The final solution is $Q^w = (Q^w_A, \ldots, Q^w_F) = (\frac{1}{3}, \frac{4}{3}, \frac{1}{3}, \frac{4}{3}, \frac{1}{3}, 0)$.

Lemma 1 *For each k iteration, the computational complexity of the Q^w rule algorithm is bounded by mn^2 where m is the number of paths in the $PERT$ network.*

Proof Taking into account that the elemental operations of the algorithm (without including the number of steps in the second loop) is bounded by n^2, we only have to prove that the number of iterations (for each k) in the second loop (denoted by t) is bounded by m. Effectively, this is true because when the algorithm pass from t to $t+1$ at least one path $\pi \in P$ is excluded (in the rest of iterations) for the calculation of the $\lambda^{k,r}$ with $r \geq t + 1$.

Let us observe that the algorithm pass from t to $t+1$ when $T^{k,t} \neq T$. If this happen, then $T^{k,t} > T$ because $T^{k,t}$ is the PERT time in $(G, d^{k,t})$ and at least $\pi^{k,t}$ has duration T. And so, it can be seen from the calculation of λ that $\lambda^{k,t+1} < \lambda^{k,t}$ holds, which implies that at least $\pi^{k,t}$ is excluded for the calculation of the $\lambda^{k,r}$ with $r \geq t + 1$.

Proposition 2 *The solutions obtained by the Q^w rule algorithm and the Q^w rule coincide for SPERT problems with $S = PS$.*

Proof Taking into account that *the Q^w rule algorithm* and the Q^w rule are iterative process with the same stop-criteria (both processes finished when there is not slack to allocate in non-fictitious activities), is enough to prove that in each iteration (k) and for each activity (i), the slack allocation of the Q^w rule, denote by $R^w_i(PE^k, S^k)$ and the slack allocation of the Q^w rule algorithm ($d^k_i - d^{k-1}_i$) coincides. To prove that $R^w_i(PE^k, S^k) = d^k_i - d^{k-1}_i$ for all k, we will use completed induction.

For $k = 1$, $R^w_i(PE^1, S^1) = \lambda^1 w_i$ if $i \in \{j \in N \backslash \widehat{N} / as_j(PE^1, S^1) > 0\}$, 0 otherwise, and $d^1_i - d^0_i = \lambda^{1,t} w_i$ if $i \in \{j \in N \backslash \widehat{N} / as^0_j > 0\}$, 0 otherwise, t being the iteration in which $T^{1,t} = T$. As $as_j(PE^1, S^1) = as^0_j$ for all $j \in N$ when $S = PS$, is enough to prove that $\lambda^1 = \lambda^{1,t}$, t being the iteration of the Q^w rule algorithm in which $T^{1,t} = T$. From the definition $\lambda^{1,t}$ and the fact that $T^{1,t} = T$ it can be concluded that $\lambda^{1,t} W \in F(PE^1, S^1)$ and for all $\epsilon > 0$, $(\lambda^{1,t} + \epsilon) W \notin F(PE^1, S^1)$. As $\lambda^1 = \max\{r \in \Re / r W^1 \in F(PE^1, S^1)\}$, both values coincide.

Now, let us suppose that for $r \leq k$, $R^w_i(PE^r, S^r) = d^r_i - d^{r-1}_i$, we have to prove that $R^w_i(PE^{k+1}, S^{k+1}) = d^{k+1}_i - d^k_i$. This part of the proof mimics the case $k = 1$, taking into account that $as_j(PE^{k+1}, S^{k+1}) = as^k_j$ for all $j \in N$ when $S = PS$.

From the Lemma 1, it can be followed that the computational complexity of the Q^w rule algorithm is bounded by $n^3 m$, m being the number of paths. The following proposition reduces this complexity under the soft assumption that the weights belong to the natural numbers set.[1]

Proposition 3 *Let (PE, PS) be a SPERT problem, if $w_i \in N^+$, then the computational complexity of the Q^w rule algorithm is bounded by $n^3 n(w)$, where $n(w) = Min \left\{Max_{\pi \in P^1} \left\{\sum_{i \in \pi} w_i\right\}, m\right\}$*

[1] Let us observe that for the case in which the weight is the range, this assumption is not too restrictive since the maximum and the minimum expected duration are usually calculated in terms of workdays.

Proof Taking into account the Lemma 1, we only have to prove that the number of iterations (for each k) in the second loop (denoted by t) is bounded by $Max_{\pi\in P^1}\left\{\sum_{i\in\pi} w_i\right\}$. As $Max_{\pi\in P^1}\left\{\sum_{i\in\pi} w_i\right\} \geq \sum_{i\in\pi^{k,1}\cap N^k} w_i$ and $w_i \in N^+$, it is enough to prove that $\sum_{i\in\pi^{k,1}\cap N^k} w_i > \sum_{i\in\pi^{k,2}\cap N^k} w_i > \sum_{i\in\pi^{k,3}\cap N^k} w_i > \ldots$, since in each iteration at lest we reduce in one unit $\sum_{i\in\pi^{k,1}\cap N^k} w_i$ and this sum obviously is a positive integer.

Let t and $t+1$ be two consecutive sub-iterations. On one hand, $\pi^{k,t}$ is the critical path of the network $(G, d^{k,t-1})$ and thus

$$T^{k,t-1} = \sum_{j\in\pi^{k,t}} d_j^{k-1} + \lambda^{k,t-1} \sum_{j\in\pi^{k,t}\cap N^k} w_j \geq \sum_{j\in\pi^{k,t+1}} d_j^{k-1} + \lambda^{k,t-1} \sum_{j\in\pi^{k,t+1}\cap N^k} w_j. \tag{2}$$

On the other hand, as $\pi^{k,t+1}$ is the critical path of the network $(G, d^{k,t})$, $T^{k,t} > T$ holds and thus

$$T^{k,t} = \sum_{j\in\pi^{k,t+1}} d_j^{k-1} + \lambda^{k,t} \sum_{j\in\pi^{k,t+1}\cap N^k} w_j > \sum_{j\in\pi^{k,t}} d_j^{k-1} + \lambda^{k,t} \sum_{j\in\pi^{k,t}\cap N^k} w_j = T. \tag{3}$$

In Lemma 1 we proved that $\lambda^{k,t-1} > \lambda^{k,t}$. So is possible to rewrite $\lambda^{k,t-1} = \lambda^{k,t} + \epsilon$, with $\epsilon > 0$. Now, replacing this expression in the inequality (2) the following holds:

$$\sum_{j\in\pi^{k,t}} d_j^{k-1} + (\lambda^{k,t} + \epsilon) \sum_{j\in\pi^{k,t}\cap N^k} w_j \geq \sum_{j\in\pi^{k,t+1}} d_j^{k-1} + (\lambda^{k,t} + \epsilon) \sum_{j\in\pi^{k,t+1}\cap N^k} w_j \tag{4}$$

Now, let us suppose that $\sum_{j\in\pi^{k,t+1}\cap N^k} w_j \geq \sum_{j\in\pi^{k,t}\cap N^k} w_j$. If we used this inequality to the previous one, we have that:

$$\Leftrightarrow \sum_{j\in\pi^{k,t}} d_j^{k-1} + \lambda^{k,t} \sum_{j\in\pi^{k,t}\cap N^k} w_j \geq \sum_{j\in\pi^{k,t+1}} d_j^{k-1} + \lambda^{k,t} \sum_{j\in\pi^{k,t+1}\cap N^k} w_j.$$

Obviously, this last inequality is a contradiction with the inequality (3), so we can conclude that $\sum_{j\in\pi^{k,t+1}\cap N^k} w_j < \sum_{j\in\pi^{k,t}\cap N^k} w_j$ for all $t \geq 1$. $\square$

Remark 4 Although one can think that the previous proposition does not reduce considerably the bound of the computational complexity, it is important to emphasize that in real projects, even assuming that the maximum expected duration is 10 times the expected duration (i.e. $b_i \leq 10d_i$) and the minimum expected duration is 0, the $n(w)$ value with $w_i = b_i - a_i \quad \forall\, i = 1, \ldots, n$, is lower than $10T$. Let us observe that this amount is usually considerably lower than the number of paths.

6 Conclusions and final remarks

Taking into account that there exist too many real situations in which it is necessary to elaborate a tight schedule or a calendar from a PERT network previous to the execution of the project, the SPERT problem was defined from a game theory point of view. Nevertheless, few efforts have been dedicated to find efficient algorithms from computational point of view that permits us to tackle with real size SPERT problems. In this paper we have defined an efficient algorithm that permit us to solve real size stochastic SPERT problems. Let us observe that following the Remark 4 and the Footnote 1 we can guarantee that the Q^w rule algorithm is polynomial for real stochastic SPERT problems.

In addition with the definition of this new algorithm, we have extended the Q^d rule to the case in which the PERT network is stochastic. This extension (the Q^w rule) with $w_i = b_i - a_i$, presents the same desirable properties that presented the Q^d rule defined in Castro et al. (2008b).

Let us also to remember that the algorithm presented in this paper is only available for SPERT problems that represent the tight schedule problematic (i.e. $S = PS$). For SPERT problems this situation is the more interesting since when $S \neq PS$ the slack allocation problems can produce a non real schedule.

Obviously, this work leaves also some open questions. For example, in this paper we have been focused on uniform, triangular or beta random distributions. In Proposition 1 we present a desirable result for uniform and triangular random distributions. Taking into account that triangular random distributions can be viewed as convolutions of uniforms and the fact that, recently, in Monhor (2005), it is given a general formula that permits to explain more complex random distributions based on convolutions of uniform distributions, it could extend the Proposition 1 to a more general framework. How to do that is a question that merits to be studied in a future.

To conclude this paper, let us give the following three remarks.

Firstly, let us observe that another important situation in which is necessary the use of tight schedules is in the problem of allocating the cost that is produced for a delay in a project. In the problem of how to allocate the cost when a delay is produced we should determine the slot that each activity had, so is necessary to have an schedule as the starting point to decide for each activity its guilty or its associated payment of the total cost produced by the delay (see Brânzei et al. 2002, 2010; Castro et al. 2008a for example).

Secondly, we will like to emphasize that the Q^w rule proposed in this work can be understood also as a generalization of the Γ and Q^c rules, since these three rules are based on an iterative process in which the slack is allocated in a proportional way.

Finally, let us observe that given a specific calendar C, it is possible to find a weight w_c such that the Q^{w_c} rule proposed in this paper provide this calendar. Taking into account this important property of the rule, the algorithm of this paper could be used (changing adequately the weights) to tackle with the problem of finding calendars from an optimization point of view.

References

Azaron A, Perkgoz C, Sakawa M (2005) A genetic algorithm approach for the time-cost trade-off in PERT networks. Appl Math Comput 168:1317–1339

Bergantiños G, Sánchez E (2002a) NTU PERT games. Oper Res Lett 30:130–140

Bergantiños G, Sánchez E (2002b) The proportional rule for problems with constraints and claims. Math Soc Sci 43:225–249

Bergantiños G, Lorenzo L (2008) The equal award principle in problems with constraints and claims. Eur J Oper Res 188:224–239

Brânzei R, Ferrari G, Fragnelli V, Tijs S (2002) Two approaches to the Problem of Sharing Delay Costs in Joint Projects. Ann Oper Res 109:359–374

Brânzei R, Ferrari G, Fragnelli V, Tijs S (2010) A bonus-malus approach to project management. Cent Eur J Oper Res. doi:10.1007/s10100-010-0139-6

Castro J, Gómez D, Tejada J (2008a) A polynomial rule for the problem of sharing delay costs in PERT networks. Comput Oper Res 35:2376–2387

Castro J, Gómez D, Tejada J (2008b) A rule for slack allocation proportional to the duration in a PERT network. Eur J Oper Res 187:556–570

Dodin B (1985) Bounding the project completion time distribution in PERT networks. Oper Res 33:862–880

Lida T (2000) Computing bounds on project duration distribution for stochastic PERT. Nav Res Log 47(7):559–580

Monden Y (1983) Toyota production systems. Industrial Engineering and Management Press, Norcross

Monhor D (2005) On the relevance of the convolution of uniform distributions to the theory of errors. Acta Geod Geophys Hung 40:53–68

Monhor D (2010) A new probabilistic approach to the path criticality in stochastic PERT. Cent Eur J Oper Res. doi:10.1007/s10100-010-0151-x

Rogalska M, Bozejko W, Hejducki Z (2008) Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling. Autom Constr 18:24–31

Shragowitz E, Youseff H, Lu B (2003) Iterative converging algorithms for computing bounds on durations of activities in pert and pert-like models. J Comb Optim 7:5–22

Talbot F, Patterson JH (1978) Efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems. Manag Sci 24:1163–1174