A Stochastic Model and Algorithms for Determining Efficient Time-Cost Tradeoffs for a Project Activity

Pedro Godinho[1]

CeBER and Faculty of Economics, University of Coimbra

Avenida Dias da Silva, 165, 3004-512 Coimbra, Portugal

Phone: +351 239790571

Fax: + 351 239790514

E-mail: pgodinho@fe.uc.pt

João Paulo Costa

INESC-Coimbra, CeBER and Faculty of Economics, University of Coimbra

Avenida Dias da Silva, 165, 3004-512 Coimbra, Portugal

E-mail: jpaulo@fe.uc.pt

Abstract: We consider a stochastic model for the time-cost tradeoffs of an activity. In this model the activity can be undertaken by using several different resources, and the resource in use may change according to the way the activity is evolving. We present two algorithms for identifying strategies that are in a predefined neighborhood of the efficient set: one of them is based on a tree structure and the other is based on dynamic programming. Both algorithms take advantage of some mathematical properties of the model in order to reduce their running time and memory requirements. We present the results of some computational tests, as well as an application example. We conclude that the dynamic programming algorithm performs quite well, although it is sometimes necessary to adjust the parameters related to the neighborhood of the efficient set to be able to have reasonable running times.

Keywords: *Stochastic modeling, Activity management, Multicriteria analysis, Time-cost tradeoff*

MSC Codes: 90-08 Computational methods; 90B36 Scheduling theory, stochastic; 90B50 Management decision making, including multiple objectives

[1] Corresponding author

1. Introduction

This paper considers the stochastic project management problem of finding efficient time-cost trade-offs for project activities that can be undertaken by using several different resource combinations. Finding the efficient time-cost tradeoffs is a fundamental problem in the management of real-life activities. Project management models have usually dealt with this problem at the aggregate project level by reducing it to a single criterion: either a makespan related criterion (minimizing the makespan or maximizing the probability of concluding the project in a given time interval), subject to cost and/or resource related constraints (e.g., Golenko-Ginzburg and Gonik 1998; Heilmann 2001; Mokhtari *et al.* 2011), or minimizing cost subject to deadline related constraints (e.g., Guthjar *et al.* 2000; Tereso *et al.* 2004; Klerides and Hadjiconstantinou 2010; Mokhtari *et al.* 2010; Godinho and Branco 2012; Guthjar 2015; Kang and Choi 2015). In some cases it is important to explicitly generate the efficient time-cost combinations, instead of reducing the problem to a single criterion. Such an approach is followed by Godinho and Costa (2007) and Kiliç *et al.* (2008), among others.

Although several authors have pointed out the importance of properly taking the managerial flexibility into account (e.g., Jørgensen and Wallace 2000), most project management models ignore the possibility of adapting the management strategy (e.g., by changing the resource being used) to the way the project is evolving. The importance of properly taking risk and managerial flexibility into account has also been stressed in the capital budgeting literature, particularly in real option models. Such models usually focus on the exploration phase of a project, although the implementation or building phase of an activity can also have an important impact on the financial success of the project. Some authors have explicitly modeled the implementation phase, usually considering the project to be a single activity with a stochastic duration (e.g., Majd and Pindyck 1987; Pindyck 1993; Friedl 2002).

The efficient management of an activity may thus be relevant, both in the context of analyzing the implementation phase of a project, or as a step in the definition of an optimal strategy for managing a project represented by a network.

This leads us to the questions:

- How do we model the management of the implementation phase of a stochastic project activity, so that project managers can choose a strategy for the application of available resources in order to achieve a given time-cost combination?
- How do we determine efficient strategies for such a model?

In this paper we consider activities that can be performed by using different resources, and in which the resource being used can be changed throughout the duration of the activity. In this regard, a strategy is a plan defining which resources will be used in each situation to undertake an activity. The use of a resource entails a given cost per unit of time. We consider that the performance speed of the resources is stochastic, so, if there are problems and the activity is performed more slowly, we will have a higher cost. This means that instead of defining a static plan for the use of resources, it is important to change the resources in a reaction to how the activity is evolving. For a model with some detail, obtaining the exact efficient time-cost frontier may be impracticable due to the scale of the problem. So we aim to achieve a manageable approximation of the efficient frontier, with approximation errors that are defined as an input to the solution algorithms.

In this paper, therefore, our main objective is to define an effective algorithm that approximates the efficient time-cost frontier for an activity, considering that different resources can be used to undertake it and that the resource being used at any given time can be changed frequently over the duration of the activity. We intend to obtain solutions that present small and controllable approximation errors.

In most previous works, the stochastic time-cost tradeoff problem has been considered by using static, or non-adaptive, policies, which assume that crashing decisions or decisions concerning activity modes are made at the beginning of the project, without the possibility of postponing some decisions until an activity starts. There are, however, some exceptions, with some authors considering adaptive policies. Tereso *et al.* (2004) assume that the resources allocated to some activities are fixed, but for others activities they can be chosen according to the way the project is developing. Godinho and Branco (2012) and Kang and Choi (2015) present adaptive models in which the decision concerning the mode for performing an activity is made at the beginning of the activity, according to a policy that takes into account the time at which the activity starts.

In the model proposed in this paper, we consider an analysis of adaptive strategies at the level of the activity, assuming that managers can make decisions concerning the activity as a reaction to the way it is developing. In the previously mentioned adaptive approaches, the authors assume that, after the mode for performing an activity is chosen, no further changes will be made. By considering decisions made while the activity is developing, we are using a more detailed level of analysis. As far as we know, such an analysis has been very rarely considered in the literature, since most authors consider the probability distributions of the time and cost of each activity to be predetermined.

We must also mention that different approaches have been followed by other authors, concerning the way that uncertainty is modeled or handled. Ke *et al.* (2010) propose three models, each one using a fuzzy approach in which the duration and the cost of the activities are fuzzy and the manager can change the duration of the activity by undertaking actions that also affect its cost. Mahdiraji *et al.* (2011) use a grey mathematical programming model in which intervals are used instead of crisp values to represent the activity duration and cost. Although such approaches may be appealing for some decision makers, they may not be suitable for handling a large number of short uncertain periods, as we will have when we model an activity in detail. In fact, the works we could find using such approaches consider the level of the project, not the level of the activity which we want to take into account in this work.

Cohen *et al.* (2007) propose a robust approach for finding the minimum cost for the project, ensuring that a predetermined due date is met. The approach is adaptive, resorting to linear decision rules, and robust, using interval-type and ellipsoidal-type uncertainty sets for the activity durations. The robustness of the model ensures that the model constraints are satisfied by all possible realizations of the data in the uncertainty sets, and it guarantees that a worst-case cost is minimized in the uncertainty sets. As in the approaches based on the use of fuzzy sets and intervals, in this approach it is difficult to define reasonable uncertainty sets for short time periods, which must be done when we want to define strategies to be used within an activity.

Other authors use multi-objective approaches that consider other objectives besides time and cost. Diao *et al.* (2011) present an approach for analyzing the time-cost-quality tradeoff in a deterministic setting. Salmasnia *et al.* (2012) consider the time-cost-quality tradeoff in a stochastic setting. Azaron *et al.* (2007) consider the total direct costs of the activities, the mean project completion time and the variance of the completion time. Azaron and Tavakkoli-Moghaddam (2007) add a fourth objective - the probability of the project completion time not exceeding a given threshold - and consider a context in which new projects are generated and activities associated with successive projects contend for resources. Such approaches usually require some assumptions, for example a deterministic analysis (in the case of Diao *et al.* 2011), the independence of activity durations, or the existence of a predefined multi-objective method to be used in the analysis (in the cases of Azaron *et al.* 2007 and Azaron and Tavakkoli-Moghaddam 2007). In the case of Salmasnia *et al.* (2012), the authors simplify the problem by focusing on finding the resource allocation that maximizes the probability of the objectives lying in predefined intervals, based on a linear approximation of the relation between this probability and the levels of resources. Additionally, these approaches are not adaptive, since they do not allow decisions to be based on the way a project is developing. So, they cannot be easily applied to the definition of an adaptive strategy for undertaking an activity, which we want to develop.

We concluded that alternative approaches found in the literature for handling projects were not developed for managing a single activity. Among the works found in the literature, Laslo (2003) and, particularly, Godinho and Costa (2007) are the works that are closest to ours. Laslo (2003) analyzes the distributions of the time and cost of an activity when the amount of work and the idle times are stochastic and the performance speed is defined before the beginning of the activity. Like us, the author is concerned with modeling the time-cost tradeoff of an activity but, differently from us, the author does not consider changing the resource being used according to the way the activity is evolving.

Godinho and Costa (2007) propose a model for identifying strategies that lead to efficient time-cost tradeoffs for undertaking a single activity. The authors model activity evolution as stochastic and assume that different resources can be used to undertake it. They also assume that the activity cost and time distributions can be summarized in aggregate values for these criteria. They present some mathematical results with a view to identifying efficient strategies, and they also address the problem of finding a small set of strategies whose times and costs are close to those of efficient strategies. However, Godinho and Costa (2007) work with approximations of the cost and time of a strategy that may induce important approximation errors, in fact making the model unsuitable for real-size problems. Godinho and Costa (2007) illustrate the model and the algorithms with an example of a very limited size.

More effective algorithms are thus needed to be able to handle real problems. This paper follows the model described by Godinho and Costa (2007), but it presents new mathematical results that enable a more effective identification of strategies that are close to the efficient ones. We are able to rigorously define a neighborhood of the set of efficient strategies and ensure that the strategies we obtain are in that neighborhood. We use these results to develop two algorithms: a procedure based on tree structures and a dynamic programming algorithm. The computational tests show that the dynamic programming algorithm performs much better, in terms of both memory requirements and running time.

The main contribution of this paper is thus its proposal of a new, effective algorithm that allows for a good approximation of the efficient time-cost frontier that can be attained by managing the implementation phase of a stochastic activity. It offers the basis for developing useful tools for managers dealing with activities under uncertain time-cost settings. Although we use aggregate values of time and cost, we notice that the use of such aggregate criteria values may be an intermediate step towards identifying a small set of representative strategies whose complete time and cost distributions may then be used in an integrated model.

The paper is structured as follows. The next section provides a description of the model. The notation is slightly different from that in Godinho and Costa (2007), for a clearer presentation of the new results and algorithms. Section 3 presents new mathematical results. Section 4 introduces the algorithms for identifying the sets of strategies. Section 5 presents the results of the computational tests performed with the algorithms. Section 6 shows an application example, for illustration purposes. Some final remarks are presented in the last section. An appendix contains the proof of the mathematical results.

2. The model

In this paper we follow the model proposed by Godinho and Costa (2007). We have made some adjustments to the notation to better present the mathematical results and the new algorithms that we describe.

There are two main components in the model: *resources*, which are used to perform the activity (they correspond to the *processes* used by Godinho and Costa, 2007), and *strategies*, which are plans that define which resource will be used in each situation to undertake the activity.

This section presents the most important features of the model. In subsection 2.1, we set out the main ideas and assumptions underlying the model. Subsection 2.2 describes the most important notation for the resources. Subsection 2.3 presents the notation for the strategies, and some basic definitions concerning them.

2.1. Main ideas and assumptions

We consider an activity that can be undertaken by using the same resources, or combinations of resources, throughout its duration. Each resource, or combination of resources, corresponds to what is sometimes termed a *mode* of undertaking the activity (see, e.g., Tereso *et al.* 2004; Godinho and Branco 2012; Guthjar 2015). To simplify the explanation, we will hereafter use the term *resource* both for the case in which there is only one relevant resource and for the case where a combination of different resources must be taken into account.

The goal of the model is to determine the efficient time-cost combinations that can be achieved for the activity. We assume that the resource being used to undertake the activity can be changed along the execution of the activity, and we define strategies that determine which resource should be chosen, according to the way the activity is evolving. A strategy will define which resource should be used in each possible situation.

Risk is introduced through the speed with which the activity is performed. The same resource could perform the activity faster if things go well, or slower if there are problems. Differences in the speed are also considered by other authors, either directly (as in Golenko-Ginzburg and Gonik 1998, p. 151) or indirectly through uncertainty in the duration or the "work content" (the term used by Tereso *et al.* 2004, p. 1) of an activity. We consider that the use of a resource entails a given cost per unit of time so, if there are problems and the activity is performed more slowly the cost will be higher.

The use of resources is discretized by considering a minimum *utilization time* for each resource, and it is assumed that the resource is used in multiples of this utilization time. A *resource utilization unit* corresponds to using the resource for this minimum utilization time. We also assume that the resource being used to undertake the activity may be changed at the end of each utilization unit, that is, at the end of the utilization time. This utilization time depends on the resource: different resources can require different minimum utilization times (e.g., internal resources can probably be changed more frequently than outsourced resources). When there are no constraints preventing resources from being changed in continuous time, the definition of resource-dependent utilization times can be used to introduce some flexibility in the model, since the use of different resources may be modelled with different detail. In spite of this flexibility, we believe that in most cases it makes sense to define a utilization time for each resource that is the same as the calendar unit of the project.

We assume that it is possible to quantify the fraction of the activity that has been completed, at each time. This fraction of the activity can be seen as the quotient between the expected effort that would be needed to reach the current state of the activity and the total expected effort needed to complete the activity. For example, if an activity consists of testing a given number of identical items, the completed fraction of the activity at a given time will be the percentage of items that have been tested.

We will now summarize and discuss the main assumptions of the model.

Assumption 1: An activity can be undertaken using the same resources or combinations of resources throughout its duration, and the cost of using a resource for a unit of time is constant.

In project management, the work breakdown structure consists of dividing deliverables and project work into smaller, more manageable components, leading to specific actions that must be performed, which are the activities. So, the activities tend to be naturally homogeneous and the same resources are generally used throughout all the activity. Also, the cost of each resource for a unit of time is usually constant in projects (at least when the project time frame is not very long). We thus believe that this assumption will be met for almost all project activities.

Assumption 2: At each time it is possible to quantify the fraction of the activity that has been completed.

Earned value techniques can be applied to quantify the fraction of an activity that has been completed. The concept of earned value measures the amount of work that has been accomplished up to a given date, or in a given time period, regarding a project or an activity (see, e.g., Project Management Institute 2005). Dividing the earned value by the total amount of work involved in an activity allows us to calculate the fraction of the activity that has been completed.

Among the techniques that can be used to measure earned value, we can point out the fixed formula, the weighted milestone, the percent complete and the apportioned effort. The fixed formula is used for efforts that span one or two periods, and it credits a fixed percentage of work at the start of the work and the remaining at the completion. Efforts that are related to the completion of tangible end products or services, with a duration longer than two periods, can be measured with the weighted milestone or the percent complete techniques. The former technique divides the work to be completed into segments, each ending with an observable milestone, and assigns a value to each milestone. The percent complete technique uses indicators of the cumulative progress made against the plan for each task. If there are no objective indicators, this technique must rely on estimates made by workers or by project managers, and it can thus become very subjective. However, if there are objective indicators (e.g., number of units produced, which can be compared with the number of units to be completed, for a given product), this technique can be both objective and very useful. The apportioned effort technique is used for support activities that have a direct relationship with other activities (possible examples may be quality assurance and inspection activities). In this technique, the earned value of the support activity is determined based on the earned value of the base activity, usually assuming that the support activity corresponds to a predetermined percentage of the base activity. For more information on techniques for measuring earned value see, for example, Project Management Institute (2005).

The model presented in this paper was developed for activities whose completion can be quantified in some detail, preferably in an objective way. So, it is particularly suitable for activities whose earned value can be determined by using objective indicators and the percent complete technique. Producing or testing a given number of items, installing a given number of components, and many construction-related tasks are examples of activities whose earned value can easily be quantified in this way. Activities whose earned value can be measured using the weighted milestone technique can also be used with this model, provided that the number of segments is large enough to allow the activity advance to be quantified in some detail.

For other activities, it may be harder to quantify the percentage of completion. For example, for activities whose earned value can only be based on subjective assessments (e.g., several research and development activities where there may be some uncertainty concerning how close we are to complete the activity), the results of the model may become less reliable. Also, for activities for which the earned value is measured considering only two periods (using the fixed formula), the application of the model may become impossible, due to insufficient detail.

Assumption 3: All the resources that can be used to undertake the activity are available throughout its duration, and the resource being used to perform the activity can be changed at the end of each utilization unit.

The model assumes that resources can be reallocated from other uses (e.g., operational work) and used in the activity. The cost of using a resource in the project must reflect the opportunity cost of diverting it from its alternative uses. We also assume that whenever the resource being used to perform the activity is changed, setup times and costs are incurred.

Assumption 4: The choice of a strategy for undertaking the activity is based on an aggregated cost and an aggregated time; the aggregated cost is calculated as the expected value of the probability distribution of the cost, and the aggregated time is calculated as the expected value of the time, based on adjusted probabilities.

Each strategy will lead to a probability distribution for time and for cost. The model uses aggregated values for these criteria to identify efficient strategies. We use the aggregation procedures proposed by Godinho and Costa (2007). These authors argue that cost risk will usually be diversifiable; therefore, according to financial theory, no risk premium should be required to bear that risk, and the expected value of the cost will be suitable. For the time criterion, it is assumed that the time risk cannot usually be hedged, and time-adjusted probabilities allow the model to take this risk into account. By defining larger time-adjusted probabilities for longer times it is also possible to penalize the strategies that may lead to a longer activity duration.

Assumption 5: The fraction of the activity completed in each resource utilization unit is defined by a discrete probability distribution with two possible, strictly positive, outcomes (or, equivalently, a resource utilization unit may lead to two different advances of the activity).

The use of a discrete distribution with two possible outcomes per unit of time is common in real options models (such as the binomial model). Still, this may be considered quite a strong assumption for project management. Our model, however, intends to be only an approximate representation of the reality, and a representation based on two outcomes can capture the common situation in which either "everything is going as planned" or "a problem occurred". We note that by making the resource utilization time flexible in this model smaller time units can be used, which, considered over a longer period, could lead to a more realistic representation of the reality (for example, by considering time units of one hour for the utilization of a resource instead of one-day time units we are, in fact, allowing a wider range of outcomes per day). We also note that extending the model and the algorithms presented in this paper to more than two outcomes per resource utilization unit is almost immediate. However, we must point out that both considering smaller time units and considering more than two outcomes will add complexity and computational effort.

The fact that we assume that each fraction of the activity undertaken at a given time unit is strictly positive prevents us from considering cases where the activity could be pushed back by unexpected events. In most activities there is no possibility of being pushed back, but we must nonetheless point out that this is a limitation of the model.

We would also like to mention that considering a predefined minimum time for the utilization of each resource could lead to situations where we are very close to the end of the activity, and in fact we would need to use the resource for less than that minimum time in order to complete it. In this case, since the model always works with multiples of the resource utilization time there will in fact be a loss of effort – in spite of requiring less than a resource utilization unit, the model requires a complete utilization unit to be expended. Whenever there are real life constraints for using the resource for less than the minimum utilization time, this will not be a limitation; however, if it is possible to reallocate a resource in continuous time, there will be an approximation error. If short utilization times are used the approximation errors will be small.

Assumption 6: In the discrete probability distribution considered in assumption 5, the probabilities of the outcomes only depend on whether the largest or the smallest advance of the activity occurred in the last utilization of a resource.

This assumption specifies that the probability of “things going well” when a resource is used depends on whether or not “things went well” the last time a resource was used. If there were problems last time (i.e., if the completed fraction of the activity was the smallest one), this will usually mean that there is a larger probability of having problems again (i.e., the completed fraction of the activity once again being the smallest one). On the other hand, if everything went well the last time a resource was used the probability of having problems will usually be smaller. Although it can be argued that a longer history could be used to define the probabilities (for example, we could look at what happened in the last two or three units of utilization of a resource), we believe that in most cases this assumption will provide enough flexibility for realistically modeling the way an activity is developing.

2.2. Notation related to the resources

Most parameters needed to apply the model concern the resources. There are *n* resources that can be used to undertake the activity. A resource is usually identified by symbol *i* or *j* , and the set of resources is identified as $N = \{1, ..., n\}$.

As explained above, we allow different resources to have different utilization times (although it will usually be reasonable to consider a utilization time that is the same as the calendar unit of the project, for all resources). For resource *i*, the utilization time will be denoted by $t_i (t_i > 0)$. The constant cost associated with a unit utilization of resource *i* (that is, the cost of using resource *i* for a time t_i) is denoted by $c_i (c_i \geq 0)$.

For the resources, we must also consider the setup times and cost both for changing the resource being used and for using a resource at the start of the activity. $t_{i,j} \geq 0$ and $c_{i,j} \geq 0$ are, respectively, the setup time for and cost of switching from resource i to resource j. $t_{0,i} \geq 0$ and $c_{0,i} \geq 0$ are, respectively, the setup time for and the setup cost of using resource i at the beginning of the activity.

We assume that the performance speed of the different resources is stochastic. We assume that when there is a utilization unit of resource *i* the fraction of the activity that is completed follows a discrete distribution with two possible values: a smaller advance a_i and a larger advance $a_i + b_i$ (with $a_i > 0$ and $b_i \geq 0$). The probability of each of these fractions being completed will depend on state of the activity. To formalize this, we denote by Δx_i the fraction of the activity that is completed by a utilization unit of resource *i*. Then

$$\Delta x_i = a_i + b_i \cdot B\left(p_i\left(\theta\right)\right), \qquad (1)$$

with θ being the state of the activity, $p_i\left(\theta\right)$ being a probability that depends of θ and $B\left(p\right)$ being a random variable that follows a Bernoulli distribution with probability p of taking value 1. As explained before, the state of the activity, θ, could be defined as a function of the history of the activity, but we believe that it is enough to take into account whether "everything went as planned" or "there was a problem" the last time a resource was used.

We define two states: $\theta \in \Theta = \left\{\langle + \rangle, \langle - \rangle\right\}$, with $\theta = \langle + \rangle$ meaning that the activity had a large advance the last time a resource was used, and $\theta = \langle - \rangle$ meaning that the activity had a small advance (we assume that when the activity is at the beginning, the advance probability in (1) will be $p_i\left(\theta_0\right)$, with a predetermined $\theta_0 \in \Theta$).

For each resource, two probabilities must be defined for the fraction of the activity that is performed by a resource utilization unit: $p_i\left(\langle + \rangle\right)$, which is the probability of performing a larger fraction of the activity after a larger fraction was also performed in the previous resource utilization unit; and $p_i\left(\langle - \rangle\right)$, which is the probability of performing a larger fraction of the activity when a smaller fraction was performed in the previous resource utilization unit.

The aggregation of time is based on the use of time-adjusted probabilities, which can elicited taking into account the initial probabilities $p_i\left(\langle + \rangle\right)$ and $p_i\left(\langle - \rangle\right)$ (for details on the procedure that can be used to obtain time-adjusted probabilities, please see Godinho and Costa 2007, p. 321). The time-adjusted probabilities can be defined as:

- $p_i^{T,+}\left(\langle + \rangle\right)$: time-adjusted probability corresponding to initial probability $p_i\left(\langle + \rangle\right)$, to be used when a bigger advance in the initial utilization unit of resource *i* leads to a longer aggregated time;
- $p_i^{T,-}\left(\langle + \rangle\right)$: time-adjusted probability corresponding to initial probability $p_i\left(\langle + \rangle\right)$, to be used when a smaller advance in the initial utilization unit of resource *i* leads to a longer aggregated time;

- $p_i^{T,+}\left(\langle - \rangle\right)$: time-adjusted probability corresponding to initial probability $p_i\left(\langle - \rangle\right)$, to be used when a bigger advance in the initial utilization unit of resource *i* leads to a longer aggregated time;
- $p_i^{T,-}\left(\langle - \rangle\right)$: time-adjusted probability corresponding to initial probability $p_i\left(\langle - \rangle\right)$, to be used when a smaller advance in the initial utilization unit of resource *i* leads to a longer aggregated time.

The application of these probabilities is explained in the end of the next sub-section.

2.3. Notation and concepts related to the strategies

A strategy is a plan that defines which resources will be used to complete the activity or a fraction of the activity. A strategy starts with a utilization unit of a resource. As defined in (1), using the resource enables a smaller or a larger fraction of the activity to be performed. The resource choices made after that initial utilization of a resource may depend on the outcome of that utilization – that is, different “sub-strategies” could be followed depending on whether a larger or a smaller advance occurred with this first resource utilization. To take this into account we represent a strategy *s* as a triple:

$$s = \left(i, s^-, s^+\right) \quad (2)$$

In this notation, i is the first resource to be used, and s^- and s^+ are the strategies to be followed when the smallest advance (a_i) and the largest advance ($a_i + b_i$) occur, respectively. This notation is recursive in that the definition of a strategy uses the definition of the strategies that will be used for each possible outcome of the first resource utilization unit. In this context, strategies s^- and s^+ will sometimes be referred to as *sub-strategies* or *branches* of strategy *s*. The empty strategy, denoted by $\varnothing$, will be used to signal the end of the strategy: for example, if the activity is completed after an advance $a_i + b_i$, then $s^+ = \varnothing$.

A strategy *s* may not guarantee the completion of the activity. The maximum fraction of the activity that is guaranteed to be completed by using *s* is denoted by $x(s)$, and can be defined as

$$\begin{cases} x(\varnothing)=0 \\ x\left(\left(i,s^-,s^+\right)\right)=\min\left\{a_i+x\left(s^-\right);a_i+b_i+x\left(s^+\right)\right\} \end{cases} \qquad (3)$$

A strategy *s* with $x(s)\geq 1$ allows completion of the whole activity, and it is called a *complete strategy*. The set of complete strategies is denoted by $S(1)$.

A strategy with $x(s)<1$ is called a *partial strategy*, since it does not allow the completion of the whole activity (a partial strategy *s* can be used to complete the activity only if a fraction of at least $1-x(s)$ of the activity has already been performed). The set of partial strategies with $x(s)\geq x$ is denoted by $S(x)$.

Each strategy has a probability distribution for time and for cost. To make decisions it is useful to work with aggregated values instead of their distributions. In this paper we follow the aggregation procedure proposed by Godinho and Costa (2007). The authors argue that cost risk will usually be diversifiable; therefore, according to financial theory, no risk premium should be required to bear that risk, and the expected cost can be used.

The expression for calculating the cost of a partial strategy must consider four different cases: an empty strategy; a strategy that requires a resource utilization unit and ends immediately afterwards; a strategy that starts with a required resource utilization unit and ends if the largest activity advance occurs (i.e., the activity is completed only if "everything goes as planned"); and a strategy that starts with a resource utilization unit but, whatever the outcome of this resource utilization, it does not guarantee the completion of the activity. These four cases are considered in the following expression for the cost of a partial strategy:

$$C(s,\theta)=\begin{cases} 0,\text{ if } s=\varnothing \\ c_i,\text{ if } s=(i,\varnothing,\varnothing) \\ c_i+\left(1-p_i(\theta)\right)\cdot\left[C\left(s^-,\langle-\rangle\right)+c_{i,i^-}\right],\text{ if } s=\left(i,\left(i^-,s^{--},s^{-+}\right),\varnothing\right) \\ c_i+\left(1-p_i(\theta)\right)\cdot\left[C\left(s^-,\langle-\rangle\right)+c_{i,i^-}\right]+p_i(\theta)\cdot\left[C\left(s^+,\langle+\rangle\right)+c_{i,i^+}\right], \\ \qquad\qquad\text{if } s=\left(i,\left(i^-,s^{--},s^{-+}\right),\left(i^+,s^{+-},s^{++}\right)\right) \end{cases} \qquad (4)$$

where $s^{--},s^{-+},s^{+-},s^{++}$ are strategies and i^-,i^+ are resources. Note that since the strategy is an argument for the cost and the first resource to be used is the first element in the triple that defines the strategy, the cost function does indeed account for the first resource used in the strategy (the same thing happens with the calculation of time, in (7)).

A complete strategy allows the completion of the whole activity, so there is no need to use resources before starting a complete strategy. If θ_0 is the state of the activity at its beginning and *i* is the first resource to be used, then the cost of a complete strategy $s=\left(i,s^-,s^+\right)$ can be defined as:

$$C_C\left(s\right)=c_{0,i}+C\left(s,\theta_0\right) \qquad (5)$$

The aggregation of time is based on the use of adjusted probabilities. This is because time risk cannot usually be hedged, and such time-adjusted probabilities allow the model to take this risk into account. The proposed approach adjusts differently the same advance probability $p_i\left(\theta\right)$, according to whether the corresponding branch leads to a longer or a shorter aggregated time. Probabilities corresponding to branches that lead to longer times are usually adjusted upwards to penalize longer times.

The time-adjusted probability applied when a larger fraction of the activity is undertaken by a utilization unit of resource *i* is denoted by $p_i^T\left(\theta\right)$, with θ being the state of the activity at the beginning of the strategy (of course, when a smaller fraction of the activity is undertaken, the probability is $1-p_i^T\left(\theta\right)$). $p_i^T\left(\theta\right)$ can take one of two different values, according to the branch of the strategy that leads to a shorter aggregated time. If the first branch leads to a shorter aggregated time, it takes the value $p_i^{T,+}\left(\theta\right)$, otherwise it takes the value $p_i^{T,-}\left(\theta\right)$. Formally:

$$p_i^T\left(\theta\right)=\begin{cases} p_i^{T,+}\left(\theta\right), \text{ if } T\left(s^-,\langle-\rangle\right)+t_{i,i^-}<T\left(s^+,\langle+\rangle\right)+t_{i,i^+} \\ p_i^{T,-}\left(\theta\right), \text{ if } T\left(s^-,\langle-\rangle\right)+t_{i,i^-}\geq T\left(s^+,\langle+\rangle\right)+t_{i,i^+} \end{cases} \qquad (6)$$

Using (6), and following the same logic as in (4), we define the time of a partial strategy as:

$$T\left(s,\theta\right)=\begin{cases} 0, \text{ if } s=\varnothing \\ t_i, \text{ if } s=\left(i,\varnothing,\varnothing\right) \\ t_i+\left(1-p_i^T\left(\theta\right)\right)\cdot\left[T\left(s^-,\langle-\rangle\right)+t_{i,i^-}\right], \text{ if } s=\left(i,\left(i^-,s^{--},s^{-+}\right),\varnothing\right) \\ t_i+\left(1-p_i^T\left(\theta\right)\right)\cdot\left[T\left(s^-,\langle-\rangle\right)+t_{i,i^-}\right]+p_i^T\left(\theta\right)\cdot\left[T\left(s^+,\langle+\rangle\right)+t_{i,i^+}\right], \\ \qquad\qquad \text{if } s=\left(i,\left(i^-,s^{--},s^{-+}\right),\left(i^+,s^{+-},s^{++}\right)\right) \end{cases} \qquad (7)$$

where $s^{--}, s^{-+}, s^{+-}, s^{++}$ are strategies and i^{-}, i^{+} are resources.

If θ_0 is the state of the activity at the beginning and *i* is the first resource to be used, then the aggregated time of a complete strategy $s = \left(i, s^{-}, s^{+}\right)$ can be defined as:

$$T_C\left(s\right) = t_{0,i} + T\left(s, \theta_0\right) \qquad (8)$$

The strategies and the corresponding times and costs are the outputs of the algorithms. There are, however, two parameters concerning the algorithms that should be defined by the user: the maximum acceptable approximation errors for time and cost, denoted by ε_T and ε_C, respectively. The algorithms that we propose look for strategies whose time and cost are at least close to those of an efficient strategy, in the sense that no other strategy is able to simultaneously improve the time by more than ε_T and improve the cost by more than ε_C. There strategies are termed $\left(\varepsilon_T, \varepsilon_C\right) - efficient$ strategies.

3. Mathematical results

Our goal is to identify the efficient strategies that enable the completion of the activity. The complexity of the problem led us to exclude mathematical programming methods, which would lead to very large, probably impossible to solve, problems. A possible alternative approach might be using meta-heuristics. In fact, several different meta-heuristics have been used in the stochastic time-cost tradeoff problem, like the ant system approach used by Mokhtari et al (2011) or the electromagnetism-like algorithm used by Godinho and Branco (2012). However, the issue in this model, is the representation of strategies. Strategies may have very large dimensions, and the size of different strategies may vary for the same problem. This would make it very difficult to apply usual meta-heuristics to the presented model.

Instead, we chose to rely on the use of some mathematical properties of the model, and derive new ones, in order to develop algorithms for obtaining efficient strategies. Godinho and Costa (2007) provide a theorem stating that a complete strategy that contains a dominated partial strategy is also dominated. This theorem is very useful since it allows us to work only with efficient partial strategies when we are looking for the set of efficient complete strategies. The authors also present a result for calculating a “reduced set of strategies” that is guaranteed to contain a strategy in a given neighborhood of each element of the original set of efficient strategies. In this paper we use some different concepts that allow us to more effectively find an approximation to the set of efficient strategies.

Definition 1: A complete strategy $s \in S(1)$ is $(\varepsilon_T, \varepsilon_C)-\mathit{efficient}$, for $\varepsilon_T \geq 0, \varepsilon_C \geq 0$, if there is no other complete strategy $s' \in S(1)$ for which, simultaneously, $C_C(s') < C_C(s) - \varepsilon_C$ and $T_C(s') < T_C(s) - \varepsilon_T$.

Definition 2: A set of complete strategies $\bar{S} \subset S(1)$ is a *full set of* $(\varepsilon_T, \varepsilon_C)-\mathit{efficient\ strategies}$ if:

a. $\forall s, s_1 \in \bar{S}, \left[T_C(s) - T_C(s_1)\right] \cdot \left[C_C(s) - C_C(s_1)\right] < 0$ (no strategy in $\bar{S}$ is dominated by another strategy belonging to $\bar{S}$)

b. $\forall s' \in S(1) \exists s \in \bar{S} : C_C(s) \leq C_C(s') + \varepsilon_C \wedge T_C(s) \leq T_C(s') + \varepsilon_T$

Part b of Definition 2 states that a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies is guaranteed to contain a strategy that is in a neighborhood of each efficient strategy. So, such a set is an approximation to the set of efficient strategies. The next property states that the strategies of a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies are indeed $(\varepsilon_T, \varepsilon_C)-\text{efficient}$, despite the fact that this condition is not explicitly imposed.

Property 1: All strategies in a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies are $(\varepsilon_T, \varepsilon_C)-\text{efficient}$.[2]

We now extend the previous definitions to the partial strategies. For this, we define $S_i(x)$ as the set of strategies that allow the execution of a fraction *x* of the activity and start by using resource i, that is, the set strategies of the form $s = (i, s^-, s^+)$ with $x(s) \geq x$.

Definition 3: A partial strategy $s \in S(x)$ is $(\varepsilon_T, \varepsilon_C, x, \theta, i)-\mathit{efficient}$, for $\varepsilon_T \geq 0, \varepsilon_C \geq 0$, $x \in]0,1[$, $\theta \in \Theta$, $i \in N$, if $s \in S_i(x)$ and there is no other partial strategy $s' \in S_i(x)$ such that, simultaneously, $C(s', \theta) < C(s, \theta) - \varepsilon_C \cdot x$ and $T(s', \theta) < T(s, \theta) - \varepsilon_T \cdot x$.

Definition 4: A set of partial strategies $\bar{S} \subset S_i(x)$ is a *full set of* $(\varepsilon_T, \varepsilon_C, x, \theta, i)-\mathit{efficient}$ *strategies* if:

[2]The proofs of the mathematical properties are given in the Appendix.

a. $\forall s, s_1 \in \bar{S}, \left[T(s,\theta) - T(s_1,\theta)\right] \cdot \left[C(s,\theta) - C(s_1,\theta)\right] < 0$ (no strategy in $\bar{S}$ is dominated by another strategy belonging to $\bar{S}$)

b. $\forall s' \in S_i(x) \exists s \in \bar{S} : C(s,\theta) \leq C(s',\theta) + \varepsilon_C \cdot x \wedge T(s,\theta) \leq T(s',\theta) + \varepsilon_T \cdot x$

Property 2: All the strategies belonging to a full set of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies are $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$.

We now present a procedure that is guaranteed to provide full sets of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies.

Let $\bar{S}_i(\theta, x)$ be obtained using the following recursive procedure:

1. For $0 < x \leq a_i$, let $\bar{S}_i(\theta, x) = \{(i, \varnothing, \varnothing)\}$. That is, $\bar{S}_i(\theta, x)$ is just composed of the strategy that consists of one utilization unit of resource *i*.
2. For $a_i + b_i \geq x > a_i$, define:

$$U_i(\theta, x) = \left\{ s = (i, s^-, \varnothing) : s^- \in \bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i) \right\} \qquad (9)$$

   This means that $U_i(\theta, x)$ is the set composed of all partial strategies obtained by starting to use resource *i*, and then using a strategy from $\bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i)$.

3. For $1 \geq x > a_i + b_i$, define:

$$U_i(\theta, x) = \left\{ s = (i, s^-, s^+) : s^- \in \bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i), s^+ \in \bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i - b_i) \right\} \quad (10)$$

   This means that $U_i(\theta, x)$ is the set composed of all strategies obtained by starting to use resource *i*, and then using a strategy from $\bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i)$ if the advance a_i occurs, and a strategy from $\bigcup_{j=1}^{n} \bar{S}_j(\theta, x - a_i - b_i)$ if the activity advances by $a_i + b_i$.

4. For $1 \geq x > a_i$, $\bar{S}_i(\theta, x)$ is obtained from $U_i(\theta, x)$ by discarding all strategies that are dominated by other strategies in $U_i(\theta, x)$ and some strategies s' for which there exists $s \in \bar{S}_i(\theta, x)$ with:

$$T(s,\theta) - T(s',\theta) \leq \varepsilon_T \cdot a_i \qquad (11)$$

$$C(s,\theta) - C(s',\theta) \leq \varepsilon_C \cdot a_i \qquad (12)$$

Property 3: Consider the sets of strategies obtained through steps 1-4 above. The sets $\bar{S}_i(\theta, x)$ are full sets of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies.

This result provides a way to find full sets of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies, which are partial strategies. In fact, our ultimate goal is to determine sets of complete strategies. For given values of $\varepsilon_T \geq 0, \varepsilon_C \geq 0$, if we calculate the full sets of $(\varepsilon_T, \varepsilon_C, x = 1, \theta, i) - \text{efficient}$ strategies for all $\theta \in \Theta, i \in N$, and afterwards if we combine all these sets and remove the dominated strategies from the set thus obtained, we will get a full set of $(\varepsilon_T, \varepsilon_C) - \text{efficient}$ strategies: a set of complete strategies that is an approximation to the set of efficient strategies.

We now assume that we want to find the full sets of $(\varepsilon_T, \varepsilon_C) - \text{efficient}$ strategies for some predefined values of $\varepsilon_T \geq 0, \varepsilon_C \geq 0$. Note that if we want the set of efficient strategies (instead of an approximation to this set), it suffices to make $\varepsilon_T = \varepsilon_C = 0$.

In the next section we present two algorithms for calculating these sets. Both these algorithms are in fact ways of implementing the procedure presented above to obtain full sets of $(\varepsilon_T, \varepsilon_C, x = 1, \theta, i) - \text{efficient}$ partial strategies, for all $i \in N, \theta \in \Theta$, and afterwards using these sets to obtain a full set of $(\varepsilon_T, \varepsilon_C) - \text{efficient}$ strategies.

4. Algorithms for identifying efficient strategies

In this section we present two possible algorithms for identifying efficient strategies. The first is based on the use of tree structures that are close to the representation of strategies defined by (2), and the second is a dynamic programming algorithm.

4.1. A tree-based algorithm

Expression (2) defines the concept of strategy in a way that is closely related to a tree structure. In fact, it defines a root and two branches, each of which may either be empty or have a similar structure. Fig. 1 depicts a tree representing strategy $\left(1,\left(1,\left(2,\varnothing,\varnothing\right),\varnothing\right),\left(2,\left(1,\varnothing,\varnothing\right),\varnothing\right)\right)$.

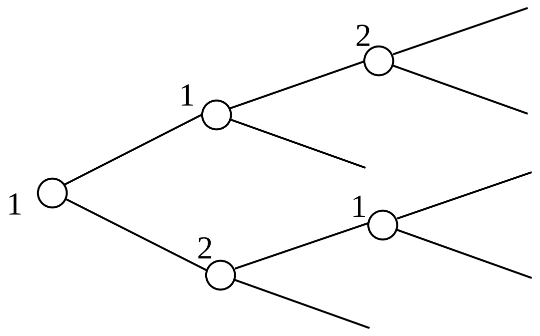

Fig. 1 Tree representing the strategy $\left(1,\left(1,\left(2,\varnothing,\varnothing\right),\varnothing\right),\left(2,\left(1,\varnothing,\varnothing\right),\varnothing\right)\right)$ (it is assumed that the upper branch always corresponds to a smaller activity advance)

A tree with only event nodes may be used for representing a strategy, such as the one shown in Fig. 1. An event node in Fig. 1 represents the use of a resource. The problem of identifying $\left(\varepsilon_T,\varepsilon_C\right)$ – efficient strategies can be defined as a search in decision trees, by adding decision nodes that represent the resource choices. Before each utilization of a resource there is a choice, corresponding to a decision node in the tree. So, before each resource utilization unit we must consider the choice of resource by adding decision nodes. The beginning of a tree explicitly representing such choices is depicted in Fig. 2.

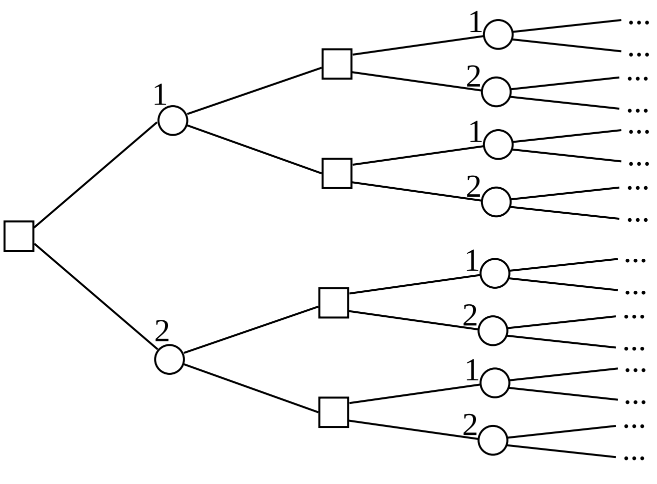

Fig. 2 Decision tree with nodes representing the choices of resources. It is assumed that *N*= {1;2}

A possible way of finding a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies consists of building the complete decision tree (of the type shown in Fig. 2), making all possible combinations of decisions and only keeping enough strategies to ensure that we have a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies. However, this way of finding $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies is impractical since the decision trees will usually be huge, and the possible combinations of decisions will be extremely large. However, the mathematical results presented earlier allow us to define a more effective algorithm based on such decision tree structures.

The idea underlying this algorithm is to identify the strategies without explicitly building the tree. We start at the tree root and recursively find the partial strategies that are candidates to be a part of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies. At each decision point (each decision node of the decision tree) we consider every possible resource and recursively find all the partial strategies that start by using that resource and are candidates to be a part of the set of complete strategies we are looking for. After finding all such strategies at a decision node, we discard those that are not required for a full set of $(\varepsilon_T, \varepsilon_C, x, \theta, i)-\text{efficient}$ strategies and return to the previous level. At the root, we get a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$ strategies. Algorithm 1 shows a simplified description of this procedure.

Algorithm 1 – Tree-based Algorithm

```
Function $\text{Main}(\varepsilon_T, \varepsilon_C, \theta_0, n, \text{resources } 1,...,n)$
    $\bar{S} \leftarrow \varnothing$
    For all resources $i = 1,...,n$ do
        $\bar{S} \leftarrow \bar{S} \cup \text{Find\_Efficient\_Strategies}(\varepsilon_T, \varepsilon_C, 1, \theta_0, i)$
    End for
    Remove from $\bar{S}$ all dominated strategies, as well as those not required for a full set of $(\varepsilon_T, \varepsilon_C)-\text{efficient}$
    strategies
    Return $\bar{S}$
End Main

Function $\text{Find\_Efficient\_Strategies}(\varepsilon_T, \varepsilon_C, x, \theta, i)$
    If $(x \le a_i)$ then
        Return $\{(i, \varnothing, \varnothing)\}$
    End if
    $\bar{S}_i(\theta, x) \leftarrow \varnothing, B^- \leftarrow \varnothing, B^+ \leftarrow \varnothing,$
    For all resources $j = 1,...,n$ do
        $B^- \leftarrow B^- \cup \text{Find\_Efficient\_Strategies}(\varepsilon_T, \varepsilon_C, x - a_i, \langle - \rangle, j)$
    End for
```

```
        If $\left(x > a_i + b_i\right)$ then
            For all resources $j = 1,...,n$ do
                $B^+ \leftarrow B^+ \cup \text{Find\_Efficient\_Strategies}\left(\varepsilon_{\mathrm{T}}, \varepsilon_{\mathrm{C}}, x - a_i - b_i, \langle + \rangle, j\right)$
            End for
        Else
            $B^+ \leftarrow \{\varnothing\}$
        End if
        For all $s_1 \in B^-, s_2 \in B^+$ do
            $\bar{S}_i(\theta, x) \leftarrow \bar{S}_i(\theta, x) \cup \{(i, s_1, s_2)\}$
        End for
        Remove from $\bar{S}_i(\theta, x)$ all dominated strategies, as well as those that are not required for a full set of
        $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies, according to (11)-(12)
        Return $\bar{S}_i(\theta, x)$
End Find_Efficient_Strategies
```

This approach allows an intuitive representation of the problem. However, its dimension is exponential in the average number of tree levels, making it impractical in many cases, even with computational support. In the next subsection we propose a different algorithm that is expected to have a better performance.

4.2. A dynamic programming algorithm

A full set of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies will be valid for all values of *x* belonging to an interval. The main idea underlying the algorithm we now present is to find the intervals of values of *x* for which the full sets of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies are valid, and calculate these sets only once. We start with small values of *x*, and proceed to successively larger values of *x*, until the full sets of $(\varepsilon_T, \varepsilon_C, x, \theta, i) - \text{efficient}$ strategies are calculated for all resources and for $x = 1$. Note that calculating full sets of strategies for intervals of values of *x* enables us to achieve significant savings in the running time of the algorithm (in relation to the tree-based algorithm), and we also achieve a more compact representation and storage of strategies. In fact, a partial strategy may be generated and represented only once, and pointers can be used to reference it whenever it is necessary. Since a partial strategy may occur within multiple complete strategies and multiple times in the same complete strategy, this approach could result in very significant savings of memory space.

We start by presenting a useful mathematical result stating that a full set of $(\varepsilon_T,\varepsilon_C,x=x',\theta,i)-$efficient strategies remains so for $x''>x'$ as long as all strategies in the set are valid to undertake a fraction x'' of the activity.

Property 4: Assume that $\bar{S}_i(\theta,x')$ is a full set of $(\varepsilon_T,\varepsilon_C,x=x',\theta,i)-$efficient strategies. If $x''>x'$ and $\forall s\in\bar{S}_i(\theta,x'),x(s)\geq x''$, then $\bar{S}_i(\theta,x')$ is also a full set of $(\varepsilon_T,\varepsilon_C,x=x'',\theta,i)-$efficient strategies.

In order to define the algorithm, we provide some additional notation. When $\bar{S}_i(\theta,x)$ is a full set of $(\varepsilon_T,\varepsilon_C,x,\theta,i)-$efficient strategies for all $x\in X$, we will denote it as $\bar{S}_i(\theta,X)$. Note that from step 1 of the procedure presented in Section 3, it is clear that $\bar{S}_i(\theta,]0,a_i])=\{(i,\varnothing,\varnothing)\}$, $\forall i\in N,\theta\in\Theta$. Additionally, Property 4 can be restated as: if $x''>x'$ and $\forall s\in\bar{S}_i(\theta,x'),x(s)\geq x''$, then $\bar{S}_i(\theta,[x',x''])=\bar{S}_i(\theta,x')$.

The algorithm works as follows. Consider a state θ and a resource i and define $l(\theta,i)$ as the largest value of x for which $\bar{S}_i(\theta,x)$ has already been determined. We choose the pair (θ',j) for which $l(\theta',j)$ is smaller. This way we are sure that we know $\bar{S}_i(\theta,l(\theta',j))$, for all resources $i\in N$ and all states $\theta\in\Theta$.

We now determine $\bar{S}_j(\theta',l(\theta',j)+\delta)$, for $\delta>0$ arbitrarily small – that is, we calculate the full set of $(\varepsilon_T,\varepsilon_C,x,\theta=\theta',i=j)-$efficient strategies for values of x immediately after $l(\theta',j)$. To define such a set, we consider all sets of strategies (j,s_1,s_2) with $s_1\in\bar{S}_{k^-}(\langle-\rangle,l(\theta',j)-a_j+\delta)$, $s_2\in\bar{S}_{k^+}(\langle+\rangle,l(\theta',j)-a_j-b_j+\delta)$, $k^-,k^+\in N$. From the set of strategies thus obtained we remove all dominated strategies, as well as those that are not required for a full set of $(\varepsilon_T,\varepsilon_C,x,\theta,i)-$efficient strategies, according to (11)-(12). We thus get $\bar{S}_j(\theta',l(\theta',j)+\delta)$. We find the largest value of x for which all strategies of this set are valid, that is, the value of $r=\min_{s\in S}\{x(s)\}$ for $S=\bar{S}_j(\theta',l(\theta',j)+\delta)$. According to Property 4, r is the right endpoint of an interval I for which $\bar{S}_j(\theta',l(\theta',j)+\delta)$ is a full set of

$(\varepsilon_T, \varepsilon_C, x, \theta, i)-$efficient strategies. Therefore we can say $\bar{S}_j\left(\theta', \left]l(\theta', j), r\right]\right) = \bar{S}_j\left(\theta', l(\theta', j) + \delta\right)$.

We now know $\bar{S}_j(\theta', x)$ for all $x \le r$, so $l(\theta', j) = r$. We find the new pair (θ', j) that minimizes $l(\theta', j)$, and proceed in the same way. The algorithm stops when we are able to calculate $\bar{S}_i(\theta_0, 1)$, for all $i \in N$. We now present the algorithm.

Algorithm 2 – Dynamic programming algorithm

```
Function Main$(\varepsilon_C, \varepsilon_T, \theta_0, n, \text{resources } 1,\dots,n)$
    For all resources $i = 1,\dots,n$, states $\theta = \langle - \rangle, \langle + \rangle$ do
        $\bar{S}_i(\theta, ]0, a_i]) \leftarrow \{(i, \varnothing, \varnothing)\}$, $l(\theta, i) \leftarrow a_i$
    While $l(\theta_0, i) < 1$, for any resource $i \in N$, do
        $(\theta', j) \leftarrow (\text{state, resource})$ with smaller $l(\theta', j)$
        Update_Set_of_Strategies$(\theta', j)$
    End do
    $S \leftarrow \varnothing$
    For all resources $i = 1,\dots,n$ do
        $S \leftarrow S \cup \bar{S}_i(\theta, 1)$
    End for
    Remove from $S$ all dominated strategies, as well as those not required for a full set of $(\varepsilon_T, \varepsilon_C)-$efficient strategies
    Return $S$
End Main

Procedure Update_Set_of_Strategies$(\theta', j)$
    $B^- \leftarrow \varnothing, B^+ \leftarrow \varnothing$
    For all resources $i = 1,\dots,n$ do
        $B^- \leftarrow B^- \cup \bar{S}_i(\langle - \rangle, l(\theta', j) - a_j + \delta)$, for $\delta > 0$ arbitrarily small
    End for
    If $(l(\theta', j) \ge a_j + b_j)$ then
        For all resources $i = 1,\dots,n$ do
            $B^+ \leftarrow B^+ \cup \bar{S}_i(\langle + \rangle, l(\theta', j) - a_j - b_j + \delta)$, for $\delta > 0$ arbitrarily small
        End for
    Else
        $B^+ \leftarrow \{\varnothing\}$
    End if
    $S \leftarrow \varnothing$
    For all $s_1 \in B^-, s_2 \in B^+$ do
        $S \leftarrow S \cup \{(j, s_1, s_2)\}$
    End for
    Remove from $S$ all dominated strategies, as well as those that are not required for a full set of $(\varepsilon_T, \varepsilon_C, x, \theta, i)-$efficient strategies, according to (11)-(12)
```

```
        $r \leftarrow \min_{s \in S} \{x(s)\}$
        $\bar{S}_i(\theta', ]l(\theta', j), r]) \leftarrow S$
        $l(\theta', j) \leftarrow r$
End  Update_Set_of_Strategies
```

5. Computational tests

Several computational tests were performed to assess the performance of the algorithms for identifying sets of strategies. Both algorithms were implemented in Borland Delphi. This section presents the results of a few of these tests.

The tests whose results we present aimed to determine how the computational times of both algorithms behave when we change two characteristics of the problem: the number of strategies and the size of the underlying tree, as considered by the tree-based algorithm.

A number of file sequences were generated for these tests. The first element of the sequence is generated by defining some parameters as constants and the rest as samples of given uniform distributions. The other elements of each sequence are defined through sequential changes in given parameters. Twenty-five sequences were generated for each set of parameter distributions, and such a set of sequences will be referred to as a "series".

In the tests presented here, the initial costs and setup times were set to 0, for all resources. Probabilities were set to $p_i(\langle - \rangle), p_i(\langle + \rangle) \in [49\%, 51\%]$, $p_i^{T,-}(\langle - \rangle), p_i^{T,-}(\langle + \rangle) \in [40\%, 45\%]$ and $p_i^{T,+}(\langle - \rangle), p_i^{T,+}(\langle + \rangle) \in [55\%, 60\%]$, for all resources $i \in N$. For the approximation of the set of efficient strategies, parameter values $\varepsilon_T = \varepsilon_C = 0.3$ were used.

Test series 1 and 2 analyze the impact of an increasing number of strategies in the execution time of the algorithms. The increase in the number of strategies is achieved by reducing the setup times and costs: lower setup times and costs will usually make it more attractive to change the resource in use, increasing the number of efficient strategies.

Series 1 considers two resources. Resource 1 allows the activity to proceed at a faster pace, but entails a higher cost. Resource parameters were set to $a_1 \in [0.075, 0.080]$, $a_1 + b_1 \in [0.095, 0.100]$, $a_2 \in [0.085, 0.090]$, $a_2 + b_2 \in [0.100, 0.105]$, $t_1 = 1.0$, $c_1 = 2.0$, $t_2 = 2.2$ and $c_2 = 1.1$. Setup times and costs are reduced from $t_{12} = c_{12} = t_{21} = c_{21} = 8$ to $t_{12} = c_{12} = t_{21} = c_{21} = 0$. The results are summarized in Fig. 3.

Series 2 considers three resources, providing different time/cost combinations. Resource parameters were set to $a_i, a_i + b_i \in [0.11, 0.12], i \in \{1;2;3\}$, $t_1 = 1$, $c_1 = 3$, $t_2 = 2$, $c_2 = 2$, $t_3 = 3$ and $c_3 = 1$. Set-up times and costs are reduced from $t_{i,j} = c_{i,j} = 8, i, j \in \{1;2;3\}, i \neq j$ to $t_{i,j} = c_{i,j} = 0, i, j \in \{1;2;3\}, i \neq j$. The results are summarized in Fig. 4.

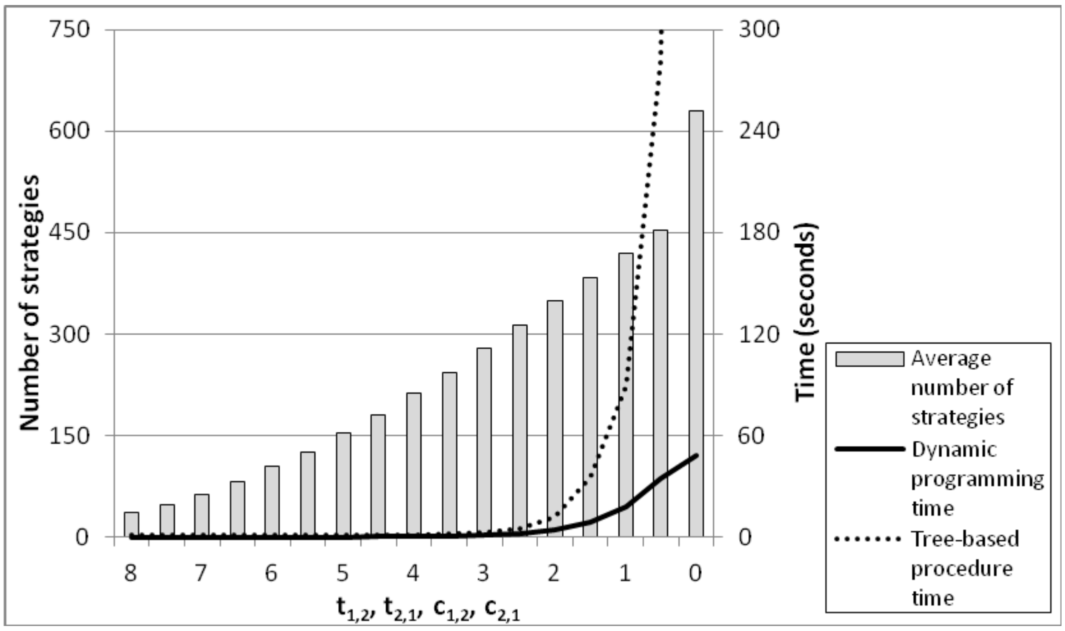

Fig. 3 Results of series 1 of computational tests. The series considers two resources, and varies the setup time and cost (depicted in the horizontal axis). The bars show the average number of strategies generated (values in the left vertical axis) and the lines depict the running times of both algorithms (values in the right vertical axis).

The results of series 1 and 2 show that as the setup times and costs decrease the number of generated strategies increases, and the run time of both algorithms also increases. The dynamic programming algorithm always performs much better than the tree-based algorithm. The performance differences are particularly noteworthy when the number of strategies is larger. In fact, for the case in which all the setup costs and times are null, the average running time of the tree-based algorithm was left out of the charts, since it was much bigger that the remaining times (1155 seconds for series 1 and 969 seconds for series 2).

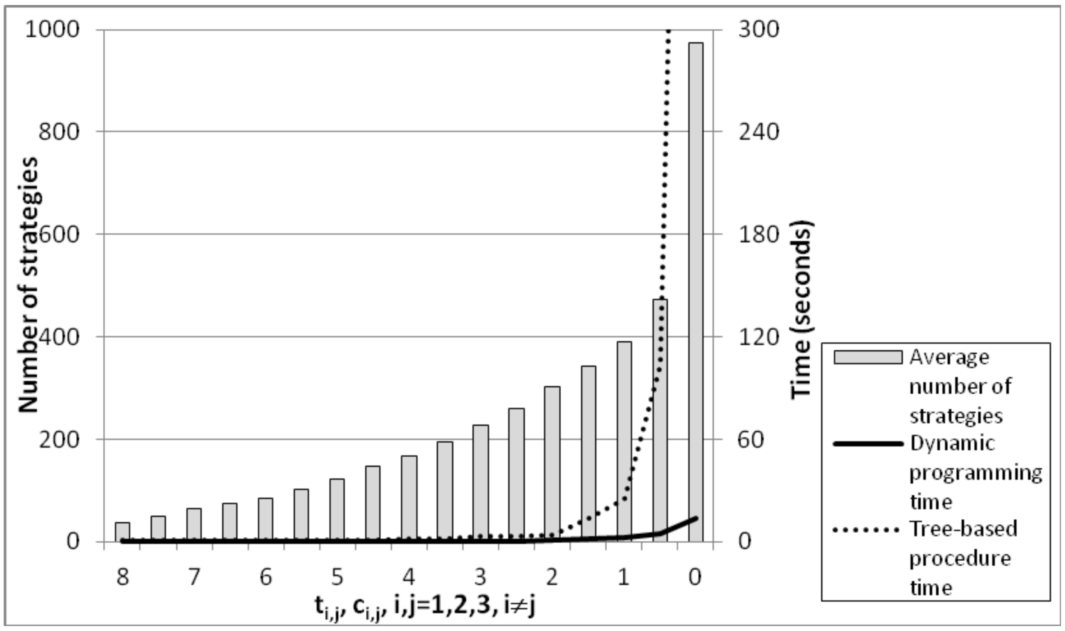

Fig. 4 Results obtained in series 2 of computational tests. The series considers three resources, and varies the setup times and costs (depicted in the horizontal axis). The bars show the average number of strategies generated (values in the left vertical axis) and the lines depict the running times of both algorithms (values in the right vertical axis)

Series 3 considers two resources. In this series, we reduced the values $a_i, a_i + b_i, i = 1,2$, to increase the number of units of resource utilization (and the size of the tree). At the same time, we defined very large setup times and costs to keep the number of strategies small; defining high setup times and costs means the resource being used never changes, leading to two efficient strategies, each corresponding to using just one of the resources to complete the activity. Resource 1 allows the activity to proceed at a faster pace, but entails a higher cost. For both resources we initially defined $a_i, a_i + b_i \in [0.2000, 0.2100]$, and reduced the values according to a geometric progression of ratio 0.9 until we reached $a_i, a_i + b_i \in [0.0027, 0.0028]$. Setup times and costs were fixed at $t_{1,2} = c_{1,2} = t_{2,1} = c_{2,1} = 10^6$. The results are summarized in Fig. 5.

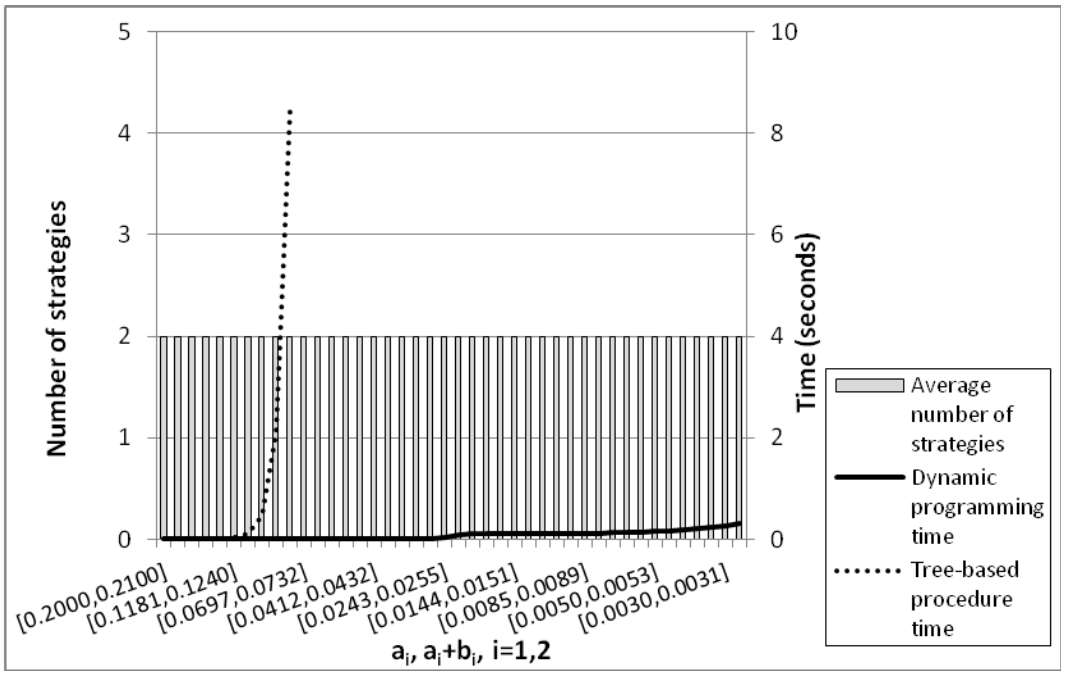

Fig. 5 Results obtained in series 3 of computational tests. The series considers two resources, and varies a_i and $a_i + b_i$ (depicted in the horizontal axis). The bars show the average number of strategies generated (values in the left vertical axis) and the lines depict the running times of both algorithms (values in the right vertical axis)

In series 3, the tree-based algorithm could no longer be run when the values of $a_i, a_i + b_i$ reached about 0.07, due to insufficient memory. However, the dynamic programming algorithm could be applied to all the instances of this series, and the execution times were always shorter than 1 second. Note that although the number of efficient strategies is small, the tree-based algorithm will require very large amounts of memory space to store each strategy when the number of tree levels is large. The dynamic programming algorithm uses a more efficient approach to strategy storage and avoids multiple copies of identical partial strategies. Therefore, it is not surprising that this algorithm could be executed when the activity advances are small, whereas the first one could not.

In the other tests that we performed (not reported here for the sake of economy of space), the dynamic programming algorithm was always faster than the tree-based algorithm. In several cases it was impossible to run the tree-based algorithm due to insufficient memory, and it was still possible to run the dynamic programming algorithm (the opposite never arose). However, the dynamic programming algorithm was unable to execute some instances of several series, usually when the values $a_i, a_i + b_i$ were small and the number of strategies was large.

We were able to conclude that the dynamic programming algorithm is better than the tree-based algorithm in terms of both memory requirements and running time. We also concluded that reducing the number of strategies to be generated could make the calibration of parameters $\varepsilon_T, \varepsilon_C$ very important, allowing the dynamic programming algorithm to be applied to some instances. This means that when it is impossible to run the algorithm for given values of $\varepsilon_T, \varepsilon_C$, it might be possible to run it if we increase these values. The application presented in the next section provides one such example.

6. An application example

We will now illustrate the application of the model with a small example. We consider a mechanical design activity (Zhongtu *et al.* 2006). We assume that it is possible to use a team of medium-skilled designers at low cost, or include some high-skilled designers, thereby increasing the cost. Using just medium-skilled designers allows the development to proceed at a good pace as long as no difficulties are met. However, when difficulties arise the activity advance slows down quite significantly and the probability of overcoming those difficulties (thus speeding up the activity) is small. Adding high-skilled designers increases the cost, but it simultaneously increases the pace of activity advance and considerably increases the probability of overcoming the difficulties when they arise.

It is possible to add or remove the high-skilled designers during the course of the activity, but this incurs additional costs and delays. When designers are added to the activity some time must be spent briefing them. When they are withdrawn there is the cost of familiarizing them with other tasks.

To model this example, we will consider two resources. Resource 1 consists of using just medium-skilled designers, and resource 2 consists of adding some highly skilled designers. We consider a utilization unit of 1 day for each resource, and assume that the cost of using resource 1 is 1 unit/day and the cost of using resource 2 is 2.5 units/day.

Resource 1 has $a_1 = 0.030$ and $b_1 = 0.015$, meaning that when the activity is proceeding normally it advances by 4.5%/day ($a_1 + b_1$), and when difficulties arise it advances by 3.0%/day (a_1). Resource 2 has $a_2 = 0.045$ and $b_2 = 0.015$, meaning that when the activity is proceeding normally it advances by 6.0%/day, and when difficulties arise it advances by 4.5%/day.

When the activity proceeds normally on a given day, the probability of finding problems the next day is 25% if resource 1 is being used, and 20% if resource 2 is being used. When difficulties have arisen, the probability of keeping proceeding at a slower pace is 75% if resource 1 is used and 40% if resource 2 is used.

When resource 1 is being used, the addition of high-skilled designers (that is, the transition to resource 2) implies a delay of 2 days and a cost of 5 units. When resource 2 is being used, the removal of high-skilled designers implies a cost of 1.5 units, but no additional delay. Using resource 1 at the beginning of the activity requires a starting time of 2 days, and a corresponding cost of 2 units. Using resource 2 at the beginning of the activity requires a starting time of 1 day, and a cost of 2.5 units.

All these parameters are summarized in Table 1. We assume that certainty equivalents of uncertain times were elicited, leading to the time-adjusted probabilities shown in the same table.

Table 1 – Parameters used in the example

$a_1 = 0.030$	$t_{12} = 2.0$	$p_1(\langle - \rangle) = 25\%$	$p_1^{T,-}(\langle - \rangle) = 20.0\%$
$b_1 = 0.015$	$c_{12} = 5.0$	$p_1(\langle + \rangle) = 75\%$	$p_1^{T,+}(\langle - \rangle) = 27.5\%$
$a_2 = 0.045$	$t_{21} = 0.0$	$p_2(\langle - \rangle) = 60\%$	$p_1^{T,-}(\langle + \rangle) = 72.5\%$
$b_2 = 0.015$	$c_{21} = 1.5$	$p_2(\langle + \rangle) = 80\%$	$p_1^{T,+}(\langle + \rangle) = 80.0\%$
$t_1 = 1.0$	$t_{01} = 2.0$		$p_2^{T,-}(\langle - \rangle) = 57.0\%$
$c_1 = 1.0$	$c_{01} = 2.0$		$p_2^{T,+}(\langle - \rangle) = 64.0\%$
$t_2 = 1.0$	$t_{02} = 1.0$		$p_2^{T,-}(\langle + \rangle) = 78.0\%$
$c_2 = 2.5$	$c_{02} = 2.5$		$p_2^{T,+}(\langle + \rangle) = 85.0\%$

We started by trying to use this example with $\varepsilon_T = 0$ and $\varepsilon_C = 0$ (that is, we tried to find the efficient strategies). In this case, we were unable to run the algorithms, due to insufficient memory space. The same happened when we used $\varepsilon_T = 0.1$ and $\varepsilon_C = 0.03$. We tried with $\varepsilon_T = 0.3$ and $\varepsilon_C = 0.05$, but we only used the dynamic programming algorithm since we were still unable to run the example with the tree-based algorithm. We obtained 759 strategies in the full set of $(\varepsilon_T = 0.3, \varepsilon_C = 0.05) -$efficient strategies, and the execution time was about 2 minutes (113 seconds). The corresponding time-cost combinations are shown in Fig. 6.

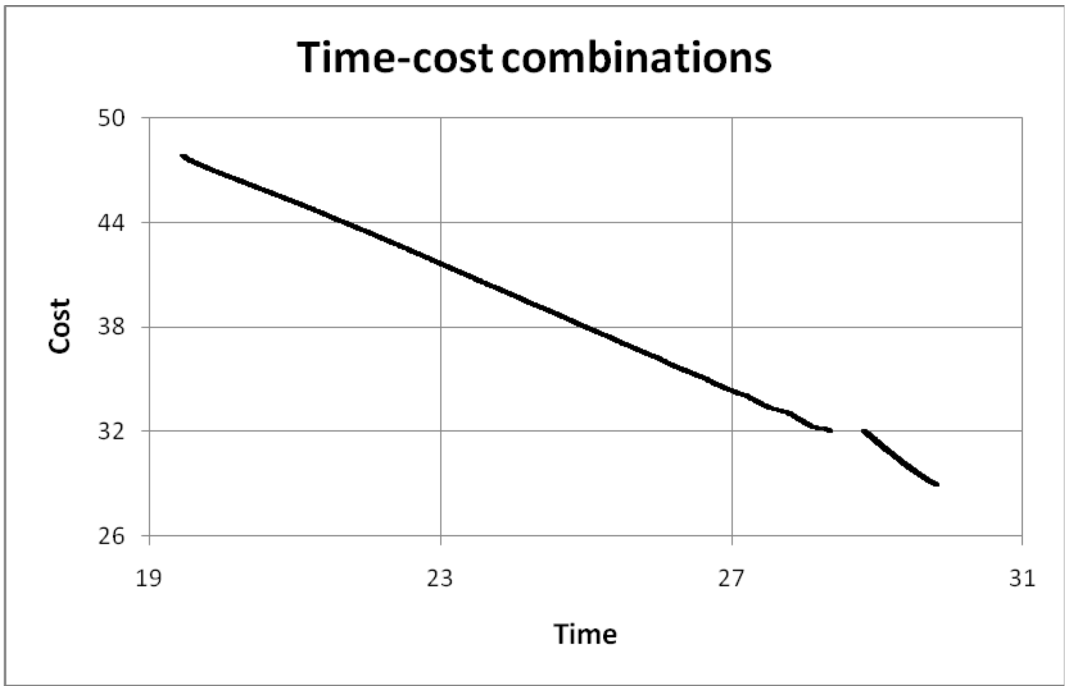

Fig. 6 Time/cost combinations for a full set of $(\varepsilon_T = 0.3, \varepsilon_C = 0.05)-$efficient strategies

We then increased the values of ε_T and ε_C once again to get an idea of the impact of these parameters. As these values increase, the number of strategies decreases and their quality tends to worsen, but the time required to generate them also decreases. For example, for $\varepsilon_T = 1.20$ and $\varepsilon_C = 0.20$, we got 189 strategies, and the execution of the algorithm only took 2 seconds. Fig. 7 shows two sections of the chart with the time-cost combinations of the strategies generated with $\varepsilon_T = 0.3$ and $\varepsilon_C = 0.05$ (base set), and $\varepsilon_T = 1.20$ and $\varepsilon_C = 0.20$ (second set). It can be seen that the second set has significantly fewer strategies, and that the strategies of the second set tend to be slightly worse (in one section of the chart, all of them are dominated by strategies from the base set). Once again, this provides an idea of the trade-off between a better approximation to the set of efficient strategies and the running time of the algorithm.

Finally, looking at Fig. 6 and the left side of Fig. 7, it is possible to see that the time-cost combinations are almost linear in some parts, and in general show an almost piecewise linear shape (although the right side of Fig. 7 shows some departures from linearity). This could be caused by several factors specific to the example, like the existence of only two resources and the closeness between original probabilities and time-adjusted probabilities. Still, even in this simple case, a discontinuity can be found in Fig. 6, which may be of great relevance to project managers: in the area close to the discontinuity it is possible to gain a significant saving in aggregate time with a small increase in aggregate cost. These discontinuities might make it quite difficult to estimate, at the outset, the shape of the time-cost combinations. Even if it were possible to know the shape of the time-cost combinations at the outset, the proposed algorithms are important because they also provide very important additional information: the strategies that allow managers to obtain given time-cost combinations.

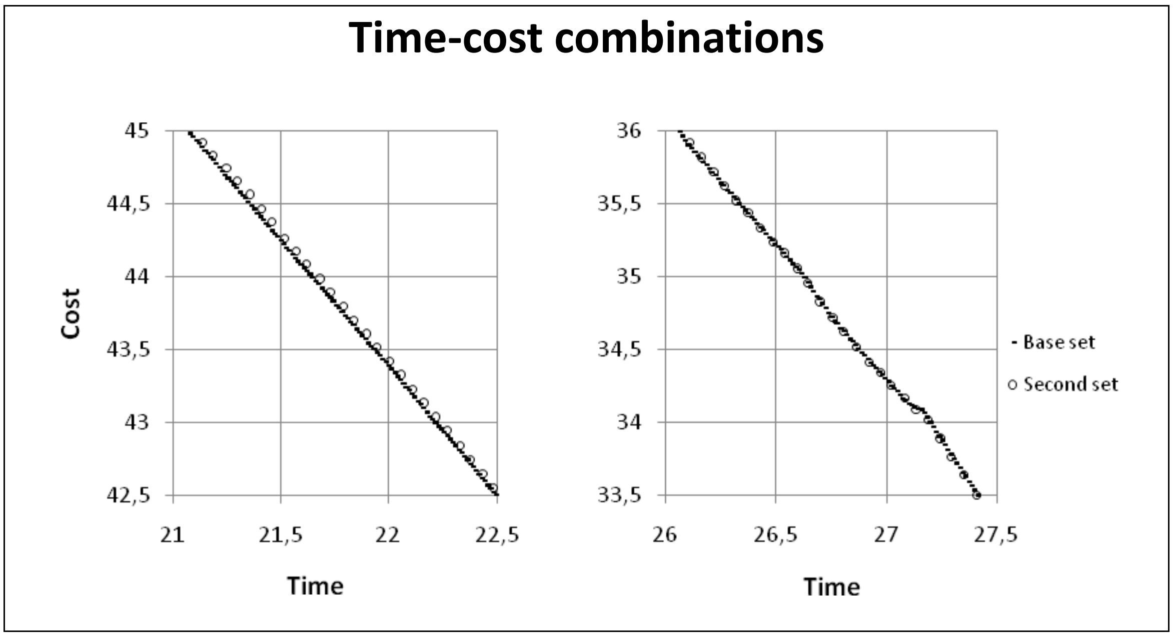

Fig. 7 Two sections of the plot showing time/cost combinations for two full sets of $(\varepsilon_T, \varepsilon_C)-$efficient strategies. Base set with $\varepsilon_T = 0.30, \varepsilon_C = 0.05$, second set with $\varepsilon_T = 1.20, \varepsilon_C = 0.20$.

7. Final remarks

This paper has considered a stochastic model for the time-cost tradeoffs of project activities. We aimed at identifying strategies that are “near” the efficient set. We have rigorously defined what we mean by “near” by introducing the concept of $(\varepsilon_T, \varepsilon_C)-$efficient strategies, where the parameters $\varepsilon_T, \varepsilon_C$ can be seen as defining a neighborhood of the set of efficient strategies. We have presented some mathematical properties for an effective identification of these strategies, and proposed two algorithms for generating them: one based on a tree structure and the other based on dynamic programming.

The computational tests showed that the dynamic programming algorithm performs much better, both in terms of memory requirements and in terms of running time. In some instances, even the dynamic programming algorithm can be very demanding in memory space and/or running time, particularly if the number of $(\varepsilon_T, \varepsilon_C)-$efficient strategies is large. However, the number of strategies can be reduced by increasing the values of $\varepsilon_T, \varepsilon_C,$ allowing us to obtain an approximation to the efficient frontier. We have presented an application example in which we had to increase the values of $\varepsilon_T, \varepsilon_C$ in order to obtain a set of strategies.

In the future, we intend to integrate this model into complete project networks or real options models. One way to do so could be to use the model to identify a limited set of relevant strategies and then use the complete probability distributions of time and cost for the identified strategies in the project network or real options model. This way we might be able to integrate the detailed management of an activity into a more complex project model.

References

Azaron A, Katagiri H, Sakawa M (2007) Time-cost trade-off via optimal control theory in Markov PERT networks. Ann Oper Res 150(1):47-64.

Azaron A, Tavakkoli-Moghaddam R (2007) Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach. Eur J Oper Res 180(3):1186-1200.

Cohen I, Golany B, Shtub A (2007) The stochastic time-cost tradeoff problem: a robust optimization approach. Networks 49(2):175-188.

Diao X, Li H, Zeng S, Tam V, Guo H (2011). A pareto multi-objective optimization approach for solving time-cost-quality tradeoff problems. Technol Econ Dev Econ 17(1):22-41.

Friedl G (2002) Sequential Investment and Time to Build. Schmalenbach Bus Rev 54:56-79.

Godinho P, Branco FG (2012) Adaptive policies for multi-mode project scheduling under uncertainty. Eur J Oper Res 216(3):553-562.

Godinho P, Costa JP (2007) A Stochastic Multimode Model for Time-Cost Tradeoffs under Management Flexibility. OR Spectr 29(2):311-334.

Golenko-Ginzburg D, Gonik A (1998) A Heuristic for Network Project Scheduling with Random Activity Durations Depending on the Resource Allocation. Int J Prod Econ 55(2):149-162.

Gutjahr WJ, Strauss C, Wagner E (2000) A Stochastic Branch-And-Bound Approach to Activity Crashing in Project Management. INFORMS J Comput 12(2):125-135.

Gutjahr WJ (2015) Bi-Objective Multi-Mode Project Scheduling Under Risk Aversion. Eur J Oper Res 246(2):421-434.

Hannan AM, Hajiagha SHR, Pourjam R (2011) A grey mathematical programming model to time-cost trade-offs in project management under uncertainty. In: Proceedings of 2011 IEEE International Conference on Grey Systems and Intelligent Services, pp 698-704.

Heilmann R (2001) Resource-Constrained Project Scheduling: A Heuristic for the Multi–mode Case. OR Spectr 23(3):335-357.

Jørgensen T, Wallace SW (2000) Improving Project Cost Estimation by Taking into Account Managerial Flexibility. Eur J Oper Res 127(2):239-251.

Kang C, Choi BC (2015) An adaptive crashing policy for stochastic time-cost tradeoff problems. Comput Oper Res 63:1-6.

Ke H, Ma W, Gao X, Xu W (2010) New fuzzy models for time-cost trade-off problem. Fuzzy Optim Decis Mak, 9(2):219-231.

Kiliç M, Ulusoy G, Şerifoğlu FS (2008) A Bi-objective Genetic Algorithm Approach to Risk Mitigation in Project Scheduling. Int J Prod Econ 112(1):202-216.

Klerides E, Hadjiconstantinou E (2010) A Decomposition-based Stochastic Programming Approach for the Project Scheduling Problem under Time/Cost Trade-off Settings and Uncertain Durations. Comput Oper Res 37(12):2131-2140.

Laslo Z (2003) Activity Time-Cost Tradeoffs under Time and Cost Chance Constraints. Comput Ind Eng 44:365-384.

Majd S, Pindyck R (1987) Time to Build, Option Value and Investment Decisions. J Financ Econ 18:7-27.

Mokhtari H, Aghaie A, Rahimi J, Mozdgir A (2010) Project Time–Cost Trade-off Scheduling: A Hybrid Optimization Approach. Int J Adv Manuf Technol 50(5-8):811-822.

Mokhtari H, Kazemzadeh RB, Salmasnia A (2011) Time-cost tradeoff analysis in project management: An ant system approach. IEEE Trans Eng Manag 58(1):36-43.

Pindyck R (1993) Investments of Uncertain Cost. J Financ Econ 3:53-76.

Project Management Institute (2005) Practice Standard for Earned Value Management. Newtown Square, PA:PMI.

Salmasnia A, Mokhtari H, Abadi INK (2012) A robust scheduling of projects with time, cost, and quality considerations. Int J Adv Manuf Technol 60(5-8):631-642.

Tereso A, Araújo M, Elmaghraby S (2004) Adaptive Resource Allocation in Multimodal Activity Networks. Int J Prod Econ 92(1):1-10.

Zhongtu L, Qifu W, Liping C (2006) A Knowledge-Based Approach for the Task Implementation in Mechanical Product Design. Int J Adv Manuf Technol 29:837-845.

Appendix: Mathematical proofs

Proof of Property 1:

Denote by $\bar{S}$ a full set of $(\varepsilon_T, \varepsilon_C)-$efficient strategies and consider a strategy $s \in \bar{S}$. Assume there is a strategy $s' \in S(1)$ such that $C_C(s') < C_C(s) - \varepsilon_C$. From part b of Definition 2, we know that there is a strategy $s_1 \in \bar{S}$ such that $C_C(s_1) \le C_C(s') + \varepsilon_C$ and $T_C(s_1) \le T_C(s') + \varepsilon_T$. This means that $C_C(s) > C_C(s_1)$ and since, from part a of Definition 2, s_1 does not dominate

s, it must be that $T_C(s) < T_C(s_1)$, so $T_C(s') > T_C(s) - \varepsilon_T$. Similarly, if $T_C(s') < T_C(s) - \varepsilon_T$, it is possible to show that $C_C(s) < C_C(s') + \varepsilon_C$. Therefore s is $(\varepsilon_T, \varepsilon_C)$ – efficient .■

Proof of Property 2:

This proof follows the same lines of the proof of Property 1.■

Proof of Property3:

For the sake of simplicity, we prove this property assuming that $p_i^{T,-}(\theta) = p_i^{T,+}(\theta) = p_i^T(\theta)$ (time-adjusted probabilities are independent of whether a large or small activity advance occurred before). The complete proof can be obtained by considering separately the different combinations of the signs that $\left[T\left(s'^{-}, \langle - \rangle\right) + t_{i,j1}\right] - \left[T\left(s'^{+}, \langle + \rangle\right) + t_{i,j2}\right]$ and $\left[T\left(s''^{-}, \langle - \rangle\right) + t_{i,j1}\right] - \left[T\left(s''^{+}, \langle + \rangle\right) + t_{i,j2}\right]$ may have in case c (defined below). The complete proof is available as Supplementary Material.

Step 4 of the procedure guarantees that no dominated strategies from $U_i(\theta, x)$ are included in $\bar{S}_i(\theta, x)$, therefore part a of Definition 4 holds. Let us now prove that part b also holds. To prove that part b of Definition 4 holds we should prove that for each $s' \in S_i(x)$ there exists $s \in \bar{S}_i(\theta, x)$ such that:

$$T(s,\theta) - T(s',\theta) \le \varepsilon_T \cdot x \qquad \text{(A1)}$$

$$C(s,\theta) - C(s',\theta) \le \varepsilon_C \cdot x \qquad \text{(A2)}$$

Let us consider 3 cases:

a) $0 < x \le a_i$;

b) $a_i + b_i \ge x > a_i$;

c) $1 \ge x > a_i + b_i$.

In order to prove that (A1) and (A2) hold, we will show that they hold in case a), and that in cases b) and c) they will hold for x if they also hold for x' such that $x' \le x - a_i$. This way we prove by induction that they hold in all cases.

Let us consider case a). In this case $s = (i, \varnothing, \varnothing)$ is the only strategy belonging to $\bar{S}_i(\theta, x)$. s also belongs to $S_i(x)$, and other strategies $s' \in S_i(x)$ must have the same or high times and costs. So, for all $s' \in S_i(x)$, $T(s,\theta) \le T(s',\theta)$ and $C(s,\theta) \le C(s',\theta)$, thus proving (A1) and (A2) for this case.

Let us now consider case b). Assume $s' = (i, s'^{-}, \varnothing)$ and that (A1) and (A2) hold for $x' \le x - a_i$, for all resources $i \in N$. Let us show that they hold for a partial strategy $s \in \bar{S}_i(\theta, x)$. Let j be the first resource to be used in strategy s'^{-}. By the induction hypothesis, there is $s'' = (i, s''^{-}, \varnothing) \in U_i(\theta, x)$ such that $s''^{-} \in \bar{S}_j(\langle - \rangle, x - a_i)$, and (A1) and (A2) hold for s'^{-} and s''^{-}.

$$\begin{aligned}
T(s'',\theta) - T(s',\theta) &= \left[t_i + \left(1 - p_i^T(\theta)\right)\cdot\left[T\left(s''^{-},\langle - \rangle\right) + t_{i,j}\right]\right] - \\
&\quad - \left[t_i + \left(1 - p_i^T(\theta)\right)\cdot\left[T\left(s'^{-},\langle - \rangle\right) + t_{i,j}\right]\right] \\
&= \left(1 - p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle - \rangle\right) - T\left(s'^{-},\langle - \rangle\right)\right] \\
&\le \left(1 - p_i^{T,-}(\theta)\right)\cdot\varepsilon_T\cdot(x - a_i) \\
&\le \varepsilon_T\cdot(x - a_i)
\end{aligned}$$

We would similarly show that $C(s'',\theta) - C(s',\theta) \le \varepsilon_C\cdot(x - a_i)$. A strategy $s \in U_i(\theta, x)$ will be included in $\bar{S}_i(\theta, x)$ such that $T(s,\theta) - T(s'',\theta) \le \varepsilon_T\cdot a_i$ and $C(s,\theta) - C(s'',\theta) \le \varepsilon_C\cdot a_i$. So:

$$\begin{aligned}
T(s,\theta) - T(s',\theta) &\le \varepsilon_T\cdot(x - a_i) + \varepsilon_T\cdot a_i \le \varepsilon_T\cdot x \\
C(s,\theta) - C(s',\theta) &\le \varepsilon_C\cdot(x - a_i) + \varepsilon_C\cdot a_i \le \varepsilon_T\cdot x
\end{aligned}$$

Let us now consider case c). Assume $s' = (i, s'^{-}, s'^{+})$ and that (A1) and (A2) hold for $x' \le x - a_i$, for all resources $i \in N$. Let us show that they hold for $s \in \bar{S}_i(\theta, x)$. Let j^{-} and j^{+} be the first resources to be used in strategies s'^{-} and s'^{+}, respectively. By the induction hypothesis, there is $s'' = (i, s''^{-}, s''^{+}) \in U_i(\theta, x)$ such that $s''^{-} \in \bar{S}_{j1}(\langle - \rangle, x - a_i)$, $s''^{+} \in \bar{S}_{j2}(\langle + \rangle, x - a_i - b_i)$, and (A1) and (A2) hold for s'^{-} and s''^{-}, and for s'^{+} and s''^{+}. Let us now consider (A1). We have:

$$
\begin{aligned}
&T(s'',\theta)-T(s',\theta)= \\
&\quad =\Big[t_i+\big(1-p_i^T(\theta)\big)\cdot\Big[T\big(s''^-,\langle-\rangle\big)+t_{i,j^-}\Big]+p_i^T(\theta)\cdot\Big[T\big(s''^+,\langle+\rangle\big)+t_{i,j^+}\Big]\Big]- \\
&\qquad -\Big[t_i+\big(1-p_i^T(\theta)\big)\cdot\Big[T\big(s'^-,\langle-\rangle\big)+t_{i,j^-}\Big]+p_i^T(\theta)\cdot\Big[T\big(s'^+,\langle+\rangle\big)+t_{i,j^+}\Big]\Big] \\
&\quad =\big(1-p_i^T(\theta)\big)\cdot\Big[T\big(s''^-,\langle-\rangle\big)-T\big(s'^-,\langle-\rangle\big)\Big]+p_i^T(\theta)\cdot\Big[T\big(s''^+,\langle+\rangle\big)-T\big(s'^+,\langle+\rangle\big)\Big] \\
&\quad \le\big(1-p_i^T(\theta)\big)\cdot\varepsilon_T\cdot(x-a_i)+p_i^T(\theta)\cdot\varepsilon_T\cdot(x-a_i-b_i) \\
&\quad \le\varepsilon_T\cdot(x-a_i)
\end{aligned}
$$

We would similarly show that $C(s'',\theta)-C(s',\theta)\le\varepsilon_C\cdot(x-a_i)$. According to step 4 of the procedure defining $\bar{S}_i(\theta,x)$, a strategy $s\in U_i(\theta,x)$ will be included in $\bar{S}_i(\theta,x)$ such that $T_P(s,\theta)-T_P(s'',\theta)\le\varepsilon_T\cdot a_i$ and $C_P(s,\theta)-C_P(s'',\theta)\le\varepsilon_C\cdot a_i$. So:

$$
\begin{aligned}
T_P(s,\theta)-T_P(s',\theta)&\le\varepsilon_T\cdot(x-a_i)+\varepsilon_T\cdot a_i \\
&\le\varepsilon_T\cdot x
\end{aligned}
$$

$$
\begin{aligned}
C_P(s,\theta)-C_P(s',\theta)&\le\varepsilon_C\cdot(x-a_i)+\varepsilon_C\cdot a_i \\
&\le\varepsilon_C\cdot x
\end{aligned}
$$

So, for each $s'\in S_i(x)$ there exists a $s\in\bar{S}_i(\theta,x)$ such that (A1) and (A2) hold, completing the proof.■

Proof of Property 4:

Since $\forall s\in\bar{S}_i(\theta,x'),x(s)\ge x''$, we know that $\bar{S}_i(\theta,x')\subset S(x'')$. Part a of Definition 4 is independent of x, so if it holds for x' it will also hold for x''. As for part b, note that $x''>x'\Rightarrow S(x'')\subset S(x')$, since a strategy that completes a fraction x'' of the activity will also allow completion of a fraction $x'<x''$. So, if part b holds for all $s'\in S(x')$, it will also hold for all strategies $s''\in S(x'')\subset S(x')$, thus completing the proof.■

Determining Efficient Time Cost Tradeoffs for an Activity – Supplementary Material

Proof of Property 3:

Step 4 of the procedure guarantees that no dominated strategies from $U_i(\theta, x)$ are included in $\bar{S}_i(\theta, x)$, therefore part a of Definition 4 holds. Let us now prove that part b also holds. To prove that part b of Definition 4 holds we will prove that for each $s' \in S_i(x)$ there exists $s \in \bar{S}_i(\theta, x)$ such that:

$$T(s,\theta) - T(s',\theta) \le \varepsilon_T \cdot x \qquad (13)$$

$$C(s,\theta) - C(s',\theta) \le \varepsilon_C \cdot x \qquad (14)$$

Let us consider 3 cases:

a) $0 < x \le a_i$;

b) $a_i + b_i \ge x > a_i$;

c) $1 \ge x > a_i + b_i$.

In order to prove that (13) and (14) hold, we will prove that they hold in case a), and that in cases b) and c) they will hold for x if they also hold for x' such that $x' \le x - a_i$. This way we prove by induction that they hold in all cases.

Let us consider case a). In this case $s = (i, \varnothing, \varnothing)$ is the only strategy belonging to $\bar{S}_i(\theta, x)$. s also belongs to $S_i(x)$, and other strategies $s' \in S_i(x)$ must have larger or equal times and costs (since a different strategy belonging to $S_i(x)$ must start by using resource *i* and then use a different resource in at least one branch, it cannot have a smaller time nor a smaller cost than s). So, for all $s' \in S_i(x)$, $T(s,\theta) \le T(s',\theta)$ and $C(s,\theta) \le C(s',\theta)$, thus proving (13) and (14) for this case.

Let us now consider case b). Assume $s' = (i, s'^{-}, \varnothing)$ and that (13) and (14) hold for $x' \le x - a_i$, for all resources $i \in N$. Let us prove that they will hold for a partial strategy $s \in \bar{S}_i(\theta, x)$. Let $j = f(s'^{-})$. There is $s'' = (i, s''^{-}, \varnothing) \in U_i(\theta, x)$ such that $s''^{-} \in \bar{S}_j(\langle - \rangle, x - a_i)$, and (13) and (14) hold for s'^{-} and s''^{-}.

$$
\begin{aligned}
T(s'',\theta)-T(s',\theta) &= \left[t_i+\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle-\rangle\right)+t_{i,j}\right]\right]- \\
&\quad -\left[t_i+\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s'^{-},\langle-\rangle\right)+t_{i,j}\right]\right] \\
&= \left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle-\rangle\right)-T\left(s'^{-},\langle-\rangle\right)\right] \\
&\le \left(1-p_i^{T,-}(\theta)\right)\cdot\varepsilon_T\cdot\left(x-a_i\right) \\
&\le \varepsilon_T\cdot\left(x-a_i\right)
\end{aligned}
$$

We would similarly show that $C(s'',\theta)-C(s',\theta)\le\varepsilon_C\cdot(x-a_i)$. A strategy $s\in U_i(\theta,x)$ will be included in $\bar{S}_i(\theta,x)$ such that $T(s,\theta)-T(s'',\theta)\le\varepsilon_T\cdot a_i$ and $C(s,\theta)-C(s'',\theta)\le\varepsilon_C\cdot a_i$. So:

$$
\begin{aligned}
T(s,\theta)-T(s',\theta) &\le \varepsilon_T\cdot(x-a_i)+\varepsilon_T\cdot a_i\le\varepsilon_T\cdot x \\
C(s,\theta)-C(s',\theta) &\le \varepsilon_C\cdot(x-a_i)+\varepsilon_C\cdot a_i\le\varepsilon_T\cdot x
\end{aligned}
$$

Let us now consider case c). Assume $s'=\left(i,s'^{-},s'^{+}\right)$ and that (13) and (14) hold for $x'\le x-a_i$, for all resources $i\in N$. Let us prove that they will hold for $s\in\bar{S}_i(\theta,x)$. Let $j1=f\left(s'^{-}\right)$ and $j2=f\left(s'^{+}\right)$. There is $s''=\left(i,s''^{-},s''^{+}\right)\in U_i(\theta,x)$ such that $s''^{-}\in\bar{S}_{j1}\left(\langle-\rangle,x-a_i\right)$, $s''^{+}\in\bar{S}_{j2}\left(\langle+\rangle,x-a_i-b_i\right)$, and (13) and (14) hold for s'^{-} and s''^{-}, and for s'^{+} and s''^{+}. Let us now consider (13). In order to properly handle the time-adjusted probabilities, we will divide case c) into four different sub-cases:

c.i) $\begin{cases} T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}\ge T\left(s'^{+},\langle+\rangle\right)+t_{i,j2} \\ T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}\ge T\left(s''^{+},\langle+\rangle\right)+t_{i,j2} \end{cases}$

c.ii) $\begin{cases} T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}< T\left(s'^{+},\langle+\rangle\right)+t_{i,j2} \\ T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}< T\left(s''^{+},\langle+\rangle\right)+t_{i,j2} \end{cases}$

c.iii) $\begin{cases} T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}< T\left(s'^{+},\langle+\rangle\right)+t_{i,j2} \\ T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}\ge T\left(s''^{+},\langle+\rangle\right)+t_{i,j2} \end{cases}$

c.iv) $\begin{cases} T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}\ge T\left(s'^{+},\langle+\rangle\right)+t_{i,j2} \\ T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}< T\left(s''^{+},\langle+\rangle\right)+t_{i,j2} \end{cases}$

Cases c.i) and c.ii) are handled in a similar way, and the same thing happens for c.iii) and c.iv). So, we will analyse $T\left(s'',\theta\right)-T\left(s,\theta\right)$ for c.i) and c.iii).

In sub-case c.i):

$$\begin{aligned}
&T\left(s'',\theta\right)-T\left(s',\theta\right)= \\
&\quad=\left[t_i+\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}\right]+p_i^{T,-}(\theta)\cdot\left[T\left(s''^{+},\langle+\rangle\right)+t_{i,j2}\right]\right]- \\
&\qquad-\left[t_i+\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}\right]+p_i^{T,-}(\theta)\cdot\left[T\left(s'^{+},\langle+\rangle\right)+t_{i,j2}\right]\right] \\
&\quad=\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle-\rangle\right)-T\left(s'^{-},\langle-\rangle\right)\right]+ \\
&\qquad+p_i^{T,-}(\theta)\cdot\left[T\left(s''^{+},\langle+\rangle\right)-T\left(s'^{+},\langle+\rangle\right)\right] \\
&\quad\le\left(1-p_i^{T,-}(\theta)\right)\cdot\varepsilon_T\cdot\left(x-a_i\right)+p_i^{T,-}(\theta)\cdot\varepsilon_T\cdot\left(x-a_i-b_i\right) \\
&\quad\le\varepsilon_T\cdot\left(x-a_i\right)
\end{aligned}$$

In sub-case c.iii):

$$\begin{aligned}
&T\left(s'',\theta\right)-T\left(s',\theta\right)= \\
&\quad=\left[t_i+\left(1-p_i^{T,-}(\theta)\right)\cdot\left[T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}\right]+p_i^{T,-}(\theta)\cdot\left[T\left(s''^{+},\langle+\rangle\right)+t_{i,j2}\right]\right]- \\
&\qquad-\left[t_i+\left(1-p_i^{T,+}(\theta)\right)\cdot\left[T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}\right]+p_i^{T,+}(\theta)\cdot\left[T\left(s'^{+},\langle+\rangle\right)+t_{i,j2}\right]\right] \\
&\quad\le\left[t_i+T\left(s''^{-},\langle-\rangle\right)+t_{i,j1}\right]-\left[t_i+T\left(s'^{-},\langle-\rangle\right)+t_{i,j1}\right] \\
&\quad\le\varepsilon_T\cdot\left(x-a_i\right)
\end{aligned}$$

In this sub-case, notice that the first inequality comes from expressions that define the sub-case.

In all sub-cases we get $T\left(s'',\theta\right)-T\left(s',\theta\right)\le\varepsilon_T\cdot\left(x-a_i\right)$. We would similarly show that $C\left(s'',\theta\right)-C\left(s',\theta\right)\le\varepsilon_C\cdot\left(x-a_i\right)$ (there would be no need to consider the sub-cases in order to show it). According to step 4 of the procedure defining $\overline{S}_i\left(\theta,x\right)$, a strategy $s\in U_i\left(\theta,x\right)$ will be included in $\overline{S}_i\left(\theta,x\right)$ such that $T\left(s,\theta\right)-T\left(s'',\theta\right)\le\varepsilon_T\cdot a_i$ and $C\left(s,\theta\right)-C\left(s'',\theta\right)\le\varepsilon_C\cdot a_i$. So:

$$\begin{aligned}
T\left(s,\theta\right)-T\left(s',\theta\right)&\le\varepsilon_T\cdot\left(x-a_i\right)+\varepsilon_T\cdot a_i \\
&\le\varepsilon_T\cdot x
\end{aligned}$$

$$\begin{aligned}
C\left(s,\theta\right)-C\left(s',\theta\right)&\le\varepsilon_C\cdot\left(x-a_i\right)+\varepsilon_C\cdot a_i \\
&\le\varepsilon_C\cdot x
\end{aligned}$$

So, for each $s' \in S_i(x)$ there exists a $s \in \bar{S}_i(\theta, x)$ such that (13) and (14) hold, completing the proof.■