

THANKS FOR DOWNLOADING THIS PAPER.

This is a post-refereeing version of a manuscript published by Elsevier.

Please, in order to cite this paper properly:

Ballesteros-Pérez, P., Elamrousy, K.M., González-Cruz, M^a.C. (2019) “Non-linear time-cost trade-off models of activity crashing: Application to construction scheduling and project compression with fast-tracking.” *Automation in Construction*, 97,229-240.

<https://doi.org/10.1016/j.autcon.2018.11.001>

The authors recommend going to the publisher’s website in order to access the full paper.

If this paper helped you somehow in your research, feel free to cite it.

Non-linear time-cost trade-off models of activity crashing:

Application to construction scheduling and project compression with fast-tracking

Authors:

Ballesteros-Pérez, Pablo ^a ; Elamrousy, Kamel Mohamed ^b ; González-Cruz, M^a Carmen ^{c*}

^a Senior Lecturer

School of Architecture, Building and Civil Engineering

Loughborough University

Loughborough, Leicestershire LE11 3TU

United Kingdom

Email: p.ballesteros-perez@lboro.ac.uk

Phone: +44 (0) 150 922 3771

^b BSc, MSc

School of the Built Environment

University of Reading

Whiteknights, Reading RG6 6DF

United Kingdom

Email: elamrousy@aucegypt.edu

^{c*} **Corresponding author**

Associate Professor.

Dpto. de Proyectos de Ingeniería. E.T.S.I. Industriales.

Universitat Politècnica de València.

Camino de Vera s/n, 46022 Valencia, Spain.

Phone: +34 963 879 863

Email: mcgonzal@dpi.upv.es

Non-linear time-cost trade-off models of activity crashing:

Application to construction scheduling and project compression with fast-tracking

Highlights

- Almost all time-cost trade-off models are wrongly assumed to be linear or discrete.
- Two new non-linear theoretical models are proposed for construction projects.
- They allow discrete/continuous, and deterministic/stochastic configurations.
- Comparison of crashing and fast-tracking is mathematically analysed.

Non-linear time-cost trade-off models of activity crashing: Application to construction scheduling and project compression with fast-tracking

Abstract

When shortening a project's duration, activity crashing, fast-tracking and substitution are the three most commonly employed compression techniques. Crashing generally involves allocating extra resources to an activity with the intention of reducing its duration. To date, the activity time-cost relationship has for the most part been assumed to be linear, however, a few studies have suggested that this is not necessarily the case in practice.

This paper proposes two non-linear theoretical models which assume either collaborative or non-collaborative resources. These models closely depict the two most common situations occurring during construction projects. The advantages of these models are that they allow for both discrete and continuous, as well as deterministic and stochastic configurations.

Additionally, the quantity of resources required for crashing the activity can be quantified.

Comparisons between the models and another recent fast-tracking model from the literature are discussed, and a Genetic Algorithm is implemented for a fictitious application example involving both compression techniques.

Keywords

Time-cost trade-off; crashing; scheduling; compression; fast-tracking; construction.

1. Introduction

Compressing a schedule with the intention of shortening a construction project's duration is sometimes necessary for meeting deadlines or mitigating delays [1]. Research in this area has remained very productive over the last 50 years within the disciplines of scheduling, project management, operational research and computer science. However, as will be seen later, most studies have incorporated schedule compression algorithms that take for granted certain pre-established relationships between activity durations and their costs [2].

The most common of these assumptions has been that the time-cost trade-off relationship is linear. This means that when an activity is compressed (shortened by adding resources) its cost increases linearly per unit of time compressed.

Other assumptions incorporated in compression algorithms modelling crashing have included: assuming (often arbitrary) limits for by how much the activities can be crashed [3]; discretising the trade-off relationship into a series of points (sometimes referred to as 'crashing modes') [4,5]; and neglecting the efficiency loss incurred as more resources work simultaneously on the same activity [2]. In particular, the latter simplification has been causing significant deviations between planned and actual durations in real construction projects, as recently analysed by Arashpour et al. [6–9] in off-site construction.

Many of these simplifications were originally made in order to reduce the computational effort required to obtain optimal (or near optimal) compressed schedules. Even though computers are nowadays far more powerful than they were fifty years ago, many of these assumptions are still being used. To date, the original time-cost trade-off models, and in particular their underlying assumption of linearity, have remained unchallenged [10] even in light of a number of studies which have proven that many of these assumptions do not hold true in real contexts (e.g. [2,11,12]).

Actually, the very earliest studies on crashing published in the early 1960s already recognised the non-linearity between time and cost at both project- and activity-level [13,14]. However, presenting hard empirical evidence was done sparsely and much later. Two of the most commendable efforts on this regard were made by Reda and Carr [2] in 1989 and Khosrowshahi [15] in 1997. Particularly, Reda and Carr [2] proved to what extent the variable number of resources allocated when crashing an activity was not linearly related to its time compression. They also illustrated how this made finding the project's shortest duration and cheapest cost a quite challenging problem in the construction of two concrete piers for a bridge. Khosrowshahi [15] illustrated again, but in much more detail, the same phenomenon when a series of 12 similar housing buildings were built in Hong Kong. Other minor works providing additional empirical evidence have also been published since then. Some will be recounted later.

This paper offers a new alternative to the well-established linear and discrete trade-off models developed in the past. The alternative consists of a model that represents much more realistically the activity duration and cost correlation. The correlation model proposed can adopt two configurations depending on whether or not resources work collaboratively. It also involves realistic variables and parameters that are usually known, or can be reasonably approximated, when executing construction projects. Finally, it also allows for calculation of the number of resources required for achieving different levels of compression.

Hence, the proposed model offers a new, more accurate alternative for evaluating the subcomponents (activity times and costs) of the objective function (generally the project compression and/or total cost) in scheduling compression problems. Future optimisation algorithms that implement the models proposed here will thus be more representative.

The paper will be structured as follows: the *Literature review* will provide an overview of the most relevant areas that previous research on schedule compression has both

focused on and neglected. The *Materials and methods* section will describe the model proposed with its two variants. It will be used to calculate the costs associated with each of the cases and the results will be compared with a recently developed fast-tracking model from the literature. In the *Application example* section, the model will then be applied to a fictitious project in which the activity crashing and fast-tracking models will be employed both simultaneously and separately. Finally, a joint *Discussion and Conclusions* section will summarise the contributions of this research to the body of knowledge and their implications for the scientific and professional community. In addition, some areas for future research and development will be suggested.

2. Literature review

This paper is concerned with improving the representativeness of time-cost trade-off mathematical relationships so that construction schedules can be crashed in a more informed manner. The first reference to activity crashing and the network compression problem dates back to 1961, when Fulkerson proposed a linear programming approach for finding the least cost curve for a project composed of many individual ‘jobs’ [13]. In Fulkerson’s model, each job had associated crashed and normal completion times, and the cost of each job varied linearly between these limits of time.

At a similar time, Kelley [14] applied the Critical Path Method to the compression problem proposing the mathematical basis for many future algorithmic implementations. He also employed linear programming within his research.

Four years later, Meyer and Shaffer [16] applied integer linear programming to the crashing problem, and proposed three non-linear activity time-cost trade-off models. In particular, they proposed: (1) Non-increasing, bounded, piece-wise linear, continuous, and

non-convex curves; (2) curves bounded and defined only at discrete points; and (3) bounded discontinuous curves. However, they did not provide any information about where these curves came from, the applicability of each type of curve, or how to estimate the curves from real project data.

Since the publication of Meyer and Shaffer's paper, the number of algorithmic implementations of the crashing problem has been steadily increasing. Surprisingly, hardly any of these later algorithms have considered crashing models which are not either linear or discrete. A few exceptions where non-linear models have been implemented include: Vrat and Kriengkrairut (1986) [11], Nonobe and Ibaraki (2006) [17], and Goh and Hall (2013) [18] who used convex piece-wise linear functions; Deckro et al. (1995) [10] and Liberatore and Pollack-Johnson (2006) [19] who employed quadratic models; Khosrowshahi (1997) [15] who used cubic polynomial curves; and Diaby et al. (2011) [12] who incorporated negative-exponential curves. Unfortunately, little if any evidence has been provided as to why these alternatives are superior and/or more realistic than their linear or discrete counterparts.

Although the underlying activity crashing model has remained unchallenged for many years, the number of optimisation algorithms published has continued to increase. Optimisation algorithms are designed to help find feasible combinations of activity durations and costs, so that (generally) a project's total cost is minimised. An optimisation algorithm essentially tries to navigate efficiently through the space of feasible (as well as sometimes unfeasible) solutions. All optimisation algorithms thus need a model to evaluate the merit of each solution, which is the focus of this study.

Examples of optimisation techniques being applied to the schedule crashing problem, implementing either linear or discrete models, are countless. The following is a representative list whose items have been approximately ordered in decreasing frequency order from the literature:

- Heuristics
- Linear programming
- Genetic algorithms
- Fuzzy programming
- Mixed Integer programming
- Non-linear (normally mixed integer) programming
- Simulated annealing
- Branch and bound algorithms
- Dynamic programming
- Ant colony optimisation
- Particle swarm optimisation
- Tabu search
- Integer programming
- Constraint programming
- Harmony search
- Imperialist competitive algorithms
- Combinations of the above techniques.

There have been many differences among these implementations. Some have considered limited resources (e.g. [17,20–23]) and some have allowed activities to split [24]. Most have incorporated continuous trade-off curves, although some have included discrete curves (e.g. [5,22,23,25]) or, less frequently, piecewise linear curves (e.g. [11,18,26,27]). In most cases, both upper and lower crashing boundaries have been included, however in some cases lower bounds have been excluded to allow non critical activity stretching (opposite of compressing) [3,28,29]. Moreover, a few have considered flexible activity predecessor relationships (e.g. [19,30]). Most algorithms have required input of perfect information, but

some work with either fuzzy [31–33] or stochastic information [34,35]. Furthermore, some have included a correlation between activity durations [36] while most have not. Finally, although there have been some commercial computer applications developed from the algorithms proposed (e.g. [37,38]), in most cases, research has remained more theoretical.

Many of these algorithmic implementations have also allowed for other primary and/or secondary objectives in addition to total project cost minimisation. Among the many alternative, sometimes complementary, objectives, are restraining the maximum total project duration [39], achieving a minimum project quality threshold [40,41], maximising the project net cash-flow [42,43], maximising the client's satisfaction [44], and avoiding resource over-allocations [45] and/or idle times [22].

However, to the best of our knowledge, a truly theoretically-based model that attempts to represent actual time-cost trade-off relationships has never been published in the scientific literature. This is particularly striking as many studies have discussed that the time-cost relationship is certainly not linear (e.g. [2,4,10]).

Conversely, application of crashing in real life construction projects has remained limited. Some examples where crashing has been used include references [2] and [15] provided in the *Introduction*. Other lesser examples are other works published on the construction of a highway in India [46], a high-rise building in Korea [47], as well as the production and distribution of batteries within a supply chain [48]. Furthermore, Ng et al. [49] conducted a survey in 2004 to find out how both Hong Kong clients and contractors chose which activities to crash when expediting projects. Surprisingly, cost was regarded as one of the least important factors.

Furthermore, there have been limited attempts to combine activity crashing algorithms with other compression techniques (i.e. activity overlapping or fast-tracking, and

activity substitution) [50–52]. Examples of algorithms combining two or three of these techniques include: Liberatore and Pollack-Johnson's [19] quadratic mixed integer programming algorithm; Dodin and Eliman's [53] mixed integer linear program; Gerik and Qassim's [54] mixed-integer nonlinear programming algorithm; Hazini et al.'s [50,55,56] genetic algorithms; Meier et al.'s [57,58] heuristic and evolutionary algorithms; and Abuwarda and Hegazy's [59] flexible activity precedence relations for crashing and overlapping activities.

In summary, most algorithmic implementations of crashing, whether combined or not with activity overlapping and/or substitution, have been incorporating ill-based models for evaluating sets of activity compressions and costs. This has not prevented the schedule compression problem from becoming a mathematical playground for many mathematicians, operational researchers, computer scientists and project managers. However, there is clearly an opportunity to improve the representativeness of past and future algorithms to improve applicability to practical situations.

3. Materials and methods

In this section, we will formulate two non-linear crashing models for the case of collaborative and non-collaborative resources. For that purpose, the next subsections will develop sequentially the mathematical expressions for calculating the activity crashed durations (in subsection 3.1), the efficiency loss as multiple resources are working together (in subsection 3.2), and the activity crashed costs (in subsection 3.3). Finally, all variables (activity durations, number of resources and costs) will be connected with each other in subsection 3.4.

Most papers on shortening project schedules have been written for the construction sector, although a number of their findings and recommendations can easily be extrapolated to other industries [60]. That being said, this paper will also focus on construction projects.

3.1. Activity crashing models

To begin, we define activity i as a task or a set of relatively homogeneous tasks that can be performed by a single type or well-balanced team of renewable resources. The term ‘renewable’ implies the resources are available on a period-by-period basis and their total use at any given time point may be constrained. Examples of renewable resources are manpower, machines, tools, equipment, and space [61]. By well-balanced team, we mean that a number of different resources contribute together towards the completion of the activity and that if the activity needs to be expedited, a multiple number of these resources will be necessary.

We can thus define:

d_i Initial duration of activity i when only one resource (team) is allocated to the activity. d_i can be stochastic and defined by μ_i and σ_i , or deterministic.

d_{s_i} Crashed duration of activity i when the number of resources (teams) is ≥ 1 (when the number of resources equals 1, then $d_{s_i} = d_i$)

n_i Number of resources (or resource teams) allocated to the execution of activity i

μ_i Average duration of activity i when $n_i=1$

σ_i Duration standard deviation of activity i when $n_i=1$

η_i Performance coefficient of activity i as more resources are allocated to it. This coefficient varies between $[0,1]$ and is explained later in more detail.

As shown at the top of Figure 1, and for the sake of simplicity, the duration d_i of an activity i will be modelled here by a Normal distribution. This assumption has been employed

in many previous works and provides good approximations in most instances [62]. With regards the crashing of activity i , there are two possible options depending on whether or not the required resources collaborate.

<Insert Figure 1 here>

In the case of collaborative resources, there is shared and non-pre-allocated work. This means that all resources or teams work until all work is complete. Resource teams who have finished their share of work will help other teams who have not, until all work is done. As a result, there will be no idle resources at any point during the activity execution. In construction settings, this generally depicts a crashed activity executed by (resource) teams from the ‘same subcontractor’. It is depicted at the bottom left of Figure 1, where d_{i_x} corresponds to the duration of each sub-activity on which a single resource (team) is working.

In contrast, with non-collaborative resources, there is pre-allocated and non-shared work. This means that each resource team works on its own pre-allocated share of the work, and work is not shared with other teams. As a result, top-performing (quickest) resources will remain idle or move to another activity instead of helping another team complete their work for the original activity. This scenario corresponds to a crashed activity being carried out by resources from different subcontractors. It is obviously less efficient than the previous crashing option, where all resources remain collaborative. This case is shown at the bottom right in Figure 1.

From the mathematical point of view, the two cases exhibit significant differences. In the case of collaborative resources, the total activity duration once crashed (d_{s_i}) will correspond to the average of its sub-activities’ durations. However, as resources are collaborative, they all complete their work at the same time. This is defined mathematically as:

$$d_{s_i} \sim \text{avg} \{ d_{i_x} \} \quad \text{with } x=1, 2, \dots, n_i \text{ resources} \quad (1)$$

Introducing the performance coefficient η_i (which reflects the resource efficiency loss as more resources work simultaneously) and considering that the sum of work (sub-activities d_{i_x}) undertaken by each resource is equal to the total original amount of work, gives:

$$d_{i_x} \sim N \left(\frac{\mu_i}{\eta_i n_i}, \frac{\sigma_i}{\eta_i \sqrt{n_i}} \right) = \frac{1}{\eta_i} N \left(\frac{\mu_i}{n_i}, \frac{\sigma_i}{\sqrt{n_i}} \right) \quad (2)$$

This is because if $\eta_i=1$ is assumed (no efficiency loss irrespective of the number of resources), the sum of all d_{i_x} would equal the original Normal distribution representing d_i .

We can thus conclude that:

$$d_{s_i} \sim \text{avg}_{x=1,2,\dots,n_i} \{ d_{i_x} \} \approx \frac{1}{\eta_i} N \left(\frac{\mu_i}{n_i}, \frac{\sigma_i}{\sqrt{n_i}} \right) = \frac{d_i}{\eta_i n_i} \quad (3)$$

On the other hand, in the case with non-collaborative resources, the total crashed activity duration will correspond to the time spent by the slowest resource (or team). It is possible to model this situation by:

$$d_{s_i} \sim \max \{ d_{i_x} \} \quad \text{with } x=1, 2, \dots, n_i \text{ resources} \quad (4)$$

The duration of each sub-activity d_{i_x} is also represented by equation (2) and, as d_{s_i} cannot be simplified, it becomes:

$$d_{s_i} \sim \max_{x=1,2,\dots,n_i} \left\{ \frac{1}{\eta_i} N \left(\frac{\mu_i}{n_i}, \frac{\sigma_i}{\sqrt{n_i}} \right) \right\} \quad (5)$$

3.2. Resource performance coefficient

Both crashing options described above include a performance coefficient η_i that reflects the reduction in efficiency which occurs as additional resources are allocated to an activity. This coefficient will always depend on the specific nature of the construction

activity, but it is generically modelled here via the following expression as a function of n_i (the number of resources) and a per-unit dimensionless coefficient α_i :

$$\eta_i = \frac{1}{n_i^{\alpha_i}} = n_i^{-\alpha_i} \quad \text{with } \alpha_i \in [0, 1] \quad (6)$$

This relationship is shown graphically in Figure 2.

<Insert Figure 2 here>

In the limit of $\alpha_i=1$, as more resources are added, the productivity decrease is such that it negates the contributions of the added teams. In contrast, when $\alpha_i=0$, all resources remain 100% efficient irrespective of the number of resources (resources are perfectly coordinated).

3.3. Cost of crashing

Having modelled the possible duration of activity i once crashed (d_{s_i}), it is necessary to quantify the cost of the crashed activity, taking into consideration η_i . This requires the following variables:

c_i Total cost of activity i

f_i Fixed cost of activity i

v_i Variable cost of activity i

r_i Cost of non-renewable resources (e.g. materials, energy)

m_{i_x} Resource (team) mobilisation unit cost. By default it will be assumed that

$m_{i_1} = m_{i_2} = \dots = m_{i_n} = m_i$ unless explicitly stated otherwise.

Hence:

$$c_i = f_i + d_i v_i \quad (7)$$

Where:

$$f_i = r_i + n_i m_{i_x} = r_i + n_i m_i \quad (8)$$

$$d_i v_i \approx \sum_{x=1}^{n_i} d_{i_x} \cdot v_i \approx n_i d_{s_i} v_i = \frac{1}{\eta_i} d_i v_i = n_i^{\alpha_i} d_i v_i \quad (9)$$

It is worth noting that both crashing alternatives (i.e. with collaborative and non-collaborative resources) have exactly the same cost. This is because each resource contributes to the cost as long as it is working, and on average, the total work carried out by the resources is the same in both crashing alternatives. Instead, it is the final crashed duration which differs (non-collaborative resources are less efficient than collaborative ones, hence the duration of an activity carried out by collaborative resources will always be equal to or shorter than the duration of an activity carried out by non-collaborative resources).

Hence, the cost of a crashed activity can be written using any of these three equivalent expressions that will be used indistinctively from now on:

$$c_i = r_i + n_i m_i + \frac{1}{\eta_i} d_i v_i = r_i + n_i m_i + n_i^{\alpha_i} d_i v_i = r_i + n_i (m_i + d_{s_i} v_i) \quad (10)$$

It may be useful on some occasions to understand the additional cost associated with crashing an activity. This cost increment can be easily calculated as detailed below for both crashing alternatives. More specifically, if c_{i_f} and c_{i_o} represent the final activity cost (after crashing with $n_i > 1$) and the initial activity cost (without crashing when $n_i = 1$), the difference Δc_i becomes:

$$\begin{aligned} \Delta c_i = c_{i_f} - c_{i_o} &= \left(r_i + n_i m_i + \frac{1}{\eta_i} d_i v_i \right) - (r_i + m_i + d_i v_i) = (n_i - 1) m_i + \left(\frac{1 - \eta_i}{\eta_i} \right) d_i v_i = \\ &= (n_i - 1) m_i + \left(\frac{1 - n_i^{-\alpha_i}}{n_i^{-\alpha_i}} \right) d_i v_i = (n_i - 1) m_i + (n_i^{\alpha_i} - 1) d_i v_i \end{aligned} \quad (11)$$

In expression (11), the value of n_i will be different depending on whether or not the resources are collaborative.

3.4 Time-Cost trade-off curves

Now that both the duration and costs of crashing an activity are known, the relationship between these two, i.e. the time-cost trade-off curve, can be calculated. Additionally, the number of resources n_i will be related to both these variables for the two crashing alternatives. The relationship between these elements is depicted in Figure 3.

<Insert Figure 3 here>

To calculate c_i as a function of d_{s_i} in the case of collaborative resources, we first need an explicit value for n_i , which can be worked out from expression (3) as follows:

$$d_{s_i} \approx \frac{d_i}{\eta_i n_i} = \frac{d_i}{n_i^{-\alpha_i} n_i} = \frac{d_i}{n_i^{(1-\alpha_i)}} \rightarrow n_i \approx \left(\frac{d_i}{d_{s_i}} \right)^{\frac{1}{1-\alpha_i}} \quad (12)$$

By combining (12) and (10), the total cost of a crashed activity with collaborative resources becomes:

$$c_i = r_i + n_i (m_i + d_{s_i} v_i) \approx r_i + (m_i + d_{s_i} v_i) \left(\frac{d_i}{d_{s_i}} \right)^{\frac{1}{1-\alpha_i}} \quad (13)$$

Possible simplifications for this expression include:

- $d_i = \mu_i$ → Deterministic approach (instead of the more general stochastic $d_i = N(\mu_i, \sigma_i)$)
- $m_i = 0$ → No resource mobilisation cost
- $\alpha_i = 0$ → No resource performance decrease with $n_i > 1$ (i.e. η_i is always equal to 1)

Under these conditions, the cost of crashing would be constant:

$$c_i = r_i + \mu_i v_i \quad (14)$$

However, assuming all these conditions are true simultaneously may be unrealistic. Hence, the probability of the cost remaining constant, or even linear as d_{s_i} varies, are very slim. This proves mathematically that the current models in the literature deviate significantly from reality.

Similarly, the cost of crashing in the case of non-collaborative resources also requires determining a value for n_i . However, this is not directly possible from expression (5). Indeed, a value for the variable n_i cannot be extracted from the Normal distribution in (5), and a simplification is required for the computation of the maximum of the d_{i_x} durations which correspond to independent and identically-distributed (*iid*) Normal distributions.

In this case, a problem arises as the Normal distribution is not a max-stable distribution; it is only sum-stable (the sum of normally-distributed variables is still normally-distributed). This is in fact the reason why in the crashing alternative with collaborative resources, it was possible to work out the value of n_i in (3).

Therefore, the only suitable method for computing the maximum of expression (5) is to replace the Normal distribution representing the original activity duration d_i by a max-stable distribution. There are only two max-stable distributions available for this purpose: the Gumbel and the Fréchet. Both distributions are used predominantly in natural sciences (most noticeably hydrology) and economics to model the statistical distribution of maxima of series of values. Besides, as highlighted above, both are max-stable which means that the maxima of Gumbel- or Fréchet-distributed values are still Gumbel- or Fréchet-distributed. As the Gumbel distribution is slightly simpler and its parameters have physical meaning, it is the preferred candidate in this case.

The Gumbel distribution is defined as a function of two parameters: θ (location) and φ (scale), and will henceforth be referred to as $G(\theta, \varphi)$. Using the method of moments, the Gumbel distribution that represents the distribution of the duration of each sub-activity i (d_{i_x}), is:

$$d_{i_x} \sim \frac{1}{\eta_i} N\left(\frac{\mu_i}{n_i}, \frac{\sigma_i}{\sqrt{n_i}}\right) \approx \frac{1}{\eta_i} G(\theta_i, \varphi_i) = \frac{1}{\eta_i} G\left(\frac{\mu_i}{n_i} - \frac{0.45\sigma_i}{\sqrt{n_i}}, \frac{0.78\sigma_i}{\sqrt{n_i}}\right) \quad (15)$$

Knowing that the distribution of the maximum of a Gumbel distribution corresponds to

$$\max_{x=1,2,\dots,n_i} \{G_x(\theta_i, \varphi_i)\} = G(\theta_i + \varphi_i \text{LN } n_i, \varphi_i) \quad (16)$$

where $\text{LN } n_i$ represents the natural logarithm of n_i , the Gumbel distribution representing the duration of the crashed activity d_{s_i} in the case of non-collaborative resources is:

$$\begin{aligned} d_{s_i} &\sim \max_{x=1,2,\dots,n_i} \left\{ \frac{1}{\eta_i} N_x\left(\frac{\mu_i}{n_i}, \frac{\sigma_i}{\sqrt{n_i}}\right) \right\} \approx \max_{x=1,2,\dots,n_i} \left\{ \frac{1}{\eta_i} G_x\left(\frac{\mu_i}{n_i} - \frac{0.45\sigma_i}{\sqrt{n_i}}, \frac{0.78\sigma_i}{\sqrt{n_i}}\right) \right\} = \\ &= \frac{1}{\eta_i} G\left(\frac{\mu_i}{n_i} + \frac{\sigma_i}{\sqrt{n_i}}(0.78\text{LN } n_i - 0.45), \frac{0.78\sigma_i}{\sqrt{n_i}}\right) \end{aligned} \quad (17)$$

As the intention here is to work out the value of n_i , a reasonable estimate can be found using the expected value of the distribution above, i.e. $(\theta + 0.57721\varphi)$, and thus:

$$d_{s_i} \approx E(d_{s_i}) = \frac{\mu_i}{\eta_i n_i} + \frac{0.78\sigma_i \text{LN } n_i}{\eta_i \sqrt{n_i}} = \frac{\mu_i}{n_i^{1-\alpha_i}} + \frac{0.78\sigma_i \text{LN } n_i}{n_i^{1/2-\alpha_i}} \quad (18)$$

Unfortunately, it is still not possible to work out the value of n_i unless additional simplifications (e.g. $\sigma_i=0$) are made. Therefore, there will be no better option than working with the implicit equation stated in (18).

For the sake of simplicity, the remaining expressions relating d_{s_i} (the activity's crashed duration), n_i (the number of resources allocated to the activity) and c_i (the cost of the crashed activity) will not be justified step by step for both alternatives here. Instead, they are summarised in Table 1.

Table 1 shows how either d_{s_i} , n_i or c_i can be obtained once the value of one of the other two is known for both collaborative resources (top) and non-collaborative resources (bottom). Namely, in each row one of these three variables is assumed to be the independent variable and one or several approximated explicit or implicit expressions are provided to calculate the value of the other two. Indication of whether these variables can be continuous or discrete, as well as deterministic, stochastic or both, have also been provided. Table 1 contains the core expressions that future optimisation algorithms will require when playing with the values of activities crashed durations, crashed costs or number of resources when trying to find the optimum compressed schedule.

<Insert Table 1 here>

From Table 1 it is evident that for crashing problems with collaborative resources, if d_{s_i} or n_i are independent variables, the other two (dependent) variables can be obtained via explicit expressions. In contrast, when the cost is the independent variable, at least one of the other two dependent variables will have to be obtained via an implicit expression.

Analogously, for the case of non-collaborative resources, the only independent decision variable that allows generating both stochastic and deterministic values of the other two is n_i . When the decision variable is either d_{s_i} or c_i , the associated stochastic expressions are not straightforward and would necessarily be mere approximations. However, these are not paradoxically great limitations, as when resources work non-collaboratively, the number of resources n_i should be considered as a discrete rather than continuous variable in most

cases. Fortunately, it is easy to infer from the value of n_i both the deterministic and stochastic values of the other two variables d_{s_i} and c_i . It can thus be said that with the above analysis, the most common situation has been adequately represented, and the remaining scenarios will be left for future and more advanced mathematical studies.

4. Crashing vs. Fast-tracking

Fast-tracking a schedule involves partially overlapping critical activities and overriding their original precedence relationships to some extent, with the intention of shortening the overall project duration [52]. Both activity crashing and fast-tracking are schedule compression techniques, but, as seen earlier, few algorithms have considered the use of both in the same project [51].

One of the more advanced models on project fast-tracking was proposed by Ballesteros-Pérez in 2017 [63]. Ballesteros-Pérez's model allows fully stochastic analysis of activity overlaps which is an important feature allowing for a fair comparison to be made with the model(s) proposed here. Ballesteros-Pérez's model and the one developed in this paper will also later be combined when proposing an algorithmic implementation for a fictitious project.

In order to compare and combine these models, it is first necessary to homogenise the notation. The adapted variables from Ballesteros-Pérez's fast-tracking model are as follows:

- o_i Duration overlap between the successor activity i and its predecessor
- d_{p_i} (Crashed or not) duration of the predecessor(s) of activity i
- d_{s_i} (Crashed or not) duration of successor activity i (the one that will start earlier after an overlap with its predecessor(s)).
- l_i Time lag between a successor i and its predecessor

- c_{sa_i} Up-front cost of successor activity i (which is incurred immediately by simply starting the execution of activity i)
- c_{sb_i} Time dependent cost of successor activity i (which is incurred as long as the activity i is ongoing)
- β_i Parameter representing predecessor activity sensitivity (how quickly the risk of having to repeat the predecessor grows as the overlap increases) ($0 \leq \beta_i \leq +\infty$)
- γ_i Parameter representing the successor activity's evolution of cost expenditure (how quickly money is spent during the execution of the activity) ($0 \leq \gamma_i \leq +\infty$)

It must be noted that, each activity i can be now crashed and/or overlapped with its predecessor(s). From the very beginning we referred to the crashed duration of an activity as d_{s_i} , precisely with the intention of matching the notation with Ballesteros-Pérez's fast-tracking model later. This way, an activity i is noted, at the same time, as the successor i . Later compression algorithms will be able to cover the whole range of n activities ($i=1,2,\dots,n$) and check whether an activity i each time will just be crashed, brought forward (overlapped with its predecessor(s)) or both.

Therefore, using these variables, the (extra) cost (Δc_i) due to fast-tracking (overlapping) an activity i with its predecessor can be obtained from the following expression:

$$\Delta c_i = \text{Risk} \times \text{Impact} = \left(\frac{o_i}{d_{p_i} + l_i} \right)^{\beta_i} \left(c_{sa_i} + \left(\frac{o_i}{d_{s_i}} \right)^{\gamma_i} c_{sb_i} \right) \quad (19)$$

where $\Delta c_i \leq c_i$ and $0 \leq o_i \leq \text{MIN}(d_{s_i}, d_{p_i} + l_i)$. For more details on the justification and development of this expression, including the multi-predecessor case, please refer to [63].

However, according to Ballesteros-Pérez (2017), some of these variables can be simplified based on some minor conditions that hold for most construction projects:

$l_i = 0$ Time lags tend to be zero or very close to zero, especially for critical activities

$c_{sa_i} = 0$ No up-front cost as, in most occasions, even if materials have been purchased for the execution of activity i , their use will be proportional to the degree of progress of the activity (i.e., they will not be used up just by starting activity i)

$\beta_i = \gamma_i = 1$ On average, all Project activities increase their risks and cost, respectively, linearly as they increase the overlap with their predecessor(s)

$c_i = c_{sb_i}$ As $c_{sa_i} = 0$, c_i (the cost of crashing activity i) will correspond to c_{sb_i} now.

Therefore, in this case, the cost increment due to fast-tracking can be reduced to:

$$\Delta c_i = f(o_i) = \frac{o_i^2 c_i}{d_{p_i} d_{s_i}} \quad (20)$$

where $c_i = r_i + n_i(m_i + d_{s_i} \cdot v_i)$ from expression (10).

Concerning the type of compression that is normally cheaper (crashing vs fast-tracking), it can easily be seen that the cost of fast tracking is $o_i^2 / (d_{p_i} d_{s_i})$ times the crashed activity's total cost c_i . However, as Ballesteros-Pérez showed, an overlap o_i bigger than the minimum of d_{p_i} and d_{s_i} is illogical. Hence the maximum possible value of $o_i^2 / (d_{p_i} d_{s_i})$ is 1 (when the length of the predecessor and successors coincide, that is, $d_{p_i} = d_{s_i}$), and this will grow slowly for low values of o_i . In short, the maximum cost due to fast-tracking (overlapping) will be equal to the activity cost c_i , but it will remain much lower most of the time.

The cost of crashing, on the other hand, does not have a ceiling and, as expression (10) shows, its growth is a function of: m_i (resource unit mobilisation cost), v_i (activity variable

duration cost) and α_i , (parameter modelling the efficiency decrease as more resources are added). These, as r_i (the cost of non-renewable resources), remain constant when an activity is crashed. Hence, the only way an activity cost c_i will not grow significantly as more teams are allocated to it will be when both m_i and v_i are very small compared to r_i , and α_i is close to zero. At this point, no clear boundaries can be identified, as there are many variables involved which can produce many different combinations. However, it seems unlikely that in most cases, m_i , v_i and α_i will all remain sufficiently low that the cost increase is less than that caused by fast-tracking.

Therefore, a general conclusion is that, for lower levels of compression, fast-tracking will be a less expensive and thus a better option for most projects. However, with fast-tracking, compression is very limited (indeed it is mathematically impossible for a project to be fast-tracked beyond 25%, and generally not beyond 10 to 20%, of its original duration [63]). This means that, for significant levels of compression, activity crashing alone or in combination with fast-tracking will provide better (cheaper) configurations. This provides further justification for combining the two models.

5. Application example

The aim of this research was to build an improved model to more accurately evaluate the objective function for time-cost trade-off crashing optimisation algorithms. However, in order to show a first algorithmic application, a Genetic Algorithm (GA) implementing crashing and fast-tracking separately and jointly for a fictitious 10-activity project is discussed. Among common metaheuristics, GAs offer some of the quicker and simpler, but also effective, options. They are also usually available within commercial solvers included by default in spreadsheet software like Microsoft Excel[®]. Moreover, GAs have successfully

been used in crashing problems in the past (e.g. [47,55,56,64]). For all these reasons, we will use a GA for solving multiple instances of our fictitious project while finding quick and good (although maybe suboptimal) solutions.

In this section the contents will be structured as follows. First, the fictitious project and its characteristics will be described in subsection 5.1. Next, compression results and its interpretation will be addressed in subsection 5.2. All information from this application example has been appended as *Supplemental online material*.

5.1 The fictitious project

The Activity-on-Arrow network of the 10-activity fictitious project is represented in Figure 4. This figure represents a set of generic tasks (activities) connected with some precedence (finish-start) relationships, along with all the information required to crash and overlap these activities. For this example, the (average) total project duration is assumed to be 200 days and the direct cost is 5,000 monetary units (m.u.). In addition, there is an indirect cost of 25 m.u. per day. Hence, the default (initial) total cost of the project is 10,000 m.u.

<Insert Figure 4 here>

To discuss how the input information required for these models could be gathered from real contexts, we can differentiate between duration- and cost-related information. Regarding duration-related information (variables μ_i , σ_i , α_i): μ_i corresponds to the deterministic (average) duration of each activity which is presented in most construction project Gantt charts, and σ_i could, for example, be approximated from PERT three-point estimates. However, α_i has been proposed in this research for the first time, hence it has not as yet been formally analysed. As explained previously, this parameter can only vary between $\alpha_i=0$ (when multiple resources working simultaneously do not affect each other's

performance) and $\alpha_i=1$ (when the efficiency loss is so high that with the addition of a new team, the final duration of the activity will be unaffected). A project manager, therefore, should be able to approximate the value of α_i between 0 and 1 in practice. Values of: $\alpha_i=0.333$, 0.500 and 0.666 have been proposed here to reflect activities where extra resource allocation would cause a low, medium or high impact (respectively) on resource performance.

Regarding the cost-related parameters, most construction managers should arguably be aware of the values of r_i , m_i and v_i for their projects, or at least be able to make reasonable estimates (as a fraction of total cost c_i for example). Here, an arbitrary set of values has been proposed.

With all these data, we use our GA to evaluate the objective function (total project cost) combining both crashing and fast-tracking for each activity. More specifically, the algorithm will be used to minimise the average total project cost while determining the schedule compression achieved

5.2 Crashing and fast-tracking results

The optimal compression results after applying the two crashing models and Ballesteros-Pérez's fast-tracking model (jointly and separately) are shown in Figures 5 and 6. Both implementations can be found as supplemental online material under tabs *Case study a)* (corresponding to Figure 5) and *Case study b)* (Figure 6).

<Insert Figure 5 here>

The first set of solutions corresponds to crashing scenarios in which all activities are completed by collaborative resources; the duration d_{s_i} and/or the overlap o_i of each activity are the decision variables. The number of resources n_i and the activity costs c_i are both

stochastic, and average values are displayed in the figures. Both d_{s_i} and o_i are taken as continuous variables. Furthermore, it is convenient to point out that variable n_i admits fractional values, as not all resource teams have to be working full-time on the relevant task.

The second set of solutions (Figure 6) assumes non-collaborative resources. In this occasion, the number of resources n_i and/or the overlap o_i of each activity are the decision variables. Activity durations d_{s_i} and costs c_i are considered as deterministic, n_i as a discrete variable, and o_i as a continuous variable. Variable n_i in this case is assumed to be an integer, as work has been distributed among multiple teams and fractions of a team are not physically possible.

<Insert Figure 6 here>

Many interpretations can be extracted from these results; only the most relevant will be included here and they focus on the lower tables of Figures 5 and 6. Firstly, with the crashing-only scenarios, the levels of compression achieved are significantly lower when resources are non-collaborative (14.8%) versus collaborative (26.9%). Similarly, the minimised project total cost achieved is significantly lower with collaborative resources (9,365 vs 9,722 m.u.).

Secondly, the fast-tracking-only solution is the same for both sets of analyses as only a single fast-tracking model has been applied. Nevertheless, it is interesting to see how the overall level of compression achieved is lower with fast-tracking than with crashing (13.8% vs 26.9% and 14.8%), although the compression in which is achieved is at a substantially lower cost (215 m.u. vs 709 and 462 m.u.). These results are in line with the theoretical discussion in the previous section.

Finally, combining both techniques (crashing and fast-tracking) gives optimal results. For instance, in Figure 5, although the schedule compression is lower with both techniques

than with crashing only (23.2% vs 26.9%), the total cost is reduced from 9,365 to 9,271 m.u. Of potentially more interest is the case in Figure 6 where the crashing-only option has been partially limited due to the discrete nature of n_i . In this case, with a marginal cost contribution from fast-tracking of only 25 m.u., the schedule compression increases from 14.8% (crashing-only) to 21.2% (crashing and fast-tracking). In addition, the project total cost decreases significantly from 9,722 to 9,427 m.u..

This is aligned with what had been stated mathematically: fast-tracking is only capable of short compressions but it is highly cost-effective, and by combining both compression techniques, higher schedule compressions and/or cost improvements can be achieved.

Finally, for the interested reader a spreadsheet has been provided as supplemental online material. This spreadsheet compares the performance (in terms of compression achieved when minimising the total project cost) between the ubiquitous linear crashing models and the non-linear models proposed here. No fast-tracking is included in this spreadsheet, as linear models, due to their deterministic nature, are not compatible with stochastic fast-tracking models like the one presented earlier.

No detailed interpretations of the comparison presented in the spreadsheet will be provided here, as a thorough comparison between linear and non-linear models exceeds the aim of this paper. However, direct comparisons evidence that, both in the example of Figure 5 (with collaborative resources) and Figure 6 (with non-collaborative resources), the results obtained with linear approximations do not resemble the costs or compressions obtained by non-linear models. This is to be expected, as previous models were not formulated using the same variables. More experiments, however, will be required to generalise these preliminary conclusions.

6. Discussion and conclusions

In 1961 the scheduling crashing problem was mathematically formulated for the first time. Since then, a predominant part of the scientific literature has proposed a wide range of optimisation techniques. However, most research has assumed that the activity time-cost trade-off relationship, upon which overall project compression and cost depend, is linear and/or discrete even though a number of authors have provided strong empirical evidence indicating that many of the original model assumptions do not hold in real life construction projects.

In this paper, a new non-linear model for time-cost trade-off activity crashing has therefore been proposed. The primary variables of this model are the activity crashed durations, crashed costs and number of resources involved in the activity compression. Mathematical formulae expressing two of these variables as a function of a third have been proposed. Straightforward expressions have also been provided for situations covering resources working collaboratively and non-collaboratively. The former represents the situation where resources are from the same subcontractor, whereas the latter represents the situation where resources are from different subcontractors. Both variants have been almost fully developed mathematically by means of explicit or implicit expressions, and brief guidance about their implementation in real contexts has been provided through a worked example. Although the mathematical formulation of these models is relatively simple, it allows for discrete and/or continuous decision variables, as well as deterministic and stochastic analyses. These features, along with a set of parameters with a physical meaning in most cases, confers the model high levels of flexibility and applicability.

Comparisons highlighting the advantages and disadvantages of activity crashing and fast-tracking have also been mathematically analysed and discussed, and a fictitious

application example involving a small project employing the two compression techniques jointly and separately has been developed. All scenarios of the example have been solved with Genetic Algorithms, a popular optimisation technique which has been employed by many researchers for addressing scheduling problems in the past. Despite further and more extensive validation is required, results from both the theoretical and example analyses seem to suggest that activity crashing is preferable, although more expensive, when higher schedule compressions are required. In contrast, fast-tracking is limited to lower compressions but is more cost-efficient. A combination of both techniques can thus result in the best overall outcome.

Finally, concerning future research, it is obvious that although the proposed model could still be refined, the most relevant gains would be expected from the model being integrated into optimisation algorithms. Since the combination of crashing and fast-tracking constitutes an NP-hard problem in the strong sense, future algorithms aimed at obtaining faster and more efficient solutions would be of great value. As an example, the polynomial time repeated cuts algorithm recently proposed by Hochbaum [65] for problems with convex time-cost trade-offs may provide a potential application of the work developed here.

The proposed models are still resorting, in some cases, to implicit expressions. It is of great interest to keep working on their mathematical formulation so as to develop expressions that, despite being slightly less accurate, are explicit and can be more efficient from the computational point of view.

Finally, a next step to be taken is to combine the non-linear and fast-tracking models described with the third scheduling compression technique not mentioned in this paper: activity substitution. Despite its being more qualitative in nature, by combining the three schedule compression techniques construction project managers will have a fully-featured framework for effective schedule compression.

Acknowledgements

This research was supported by the CIOB Bowen Jenkins Legacy Research Fund (reference BLJ2016/BJL.01) and by NERC under the Environmental Risks to Infrastructure Innovation Programme (reference NE/R008876/1) at the University of Reading.

References

- [1] W.R. Querns, Project expediting, in: 33rd Annual Meeting of the American Association of Cost Engineers (AACE International) ISSN: 00657158, Publ. by AACE, 1989: p. D.3.1-D.3.4.
- [2] R. Reda, R.I. Carr, Time-Cost Trade-Off Among Related Activities, *Journal of Construction Engineering and Management*. 115 (1989) 475–486. doi:10.1061/(ASCE)0733-9364(1989)115:3(475).
- [3] S.M. El-Sayegh, R. El Haj, Time-cost-float optimization in construction projects, in: *Civil-Comp Proceedings*, Civil-Comp Press, 2015: p. Volume 108. ISSN: 17593433.
- [4] M.M. Cusack, A simplified approach to the planning and control of cost and project duration, *Construction Management and Economics*. 3 (1985) 183–198. doi:10.1080/01446198500000014.
- [5] Ö.H. Bettemir, M. Talat Birgönül, Network analysis algorithm for the solution of discrete time-cost trade-off problem, *KSCE Journal of Civil Engineering*. 21 (2017) 1047–1058. doi:10.1007/s12205-016-1615-x.
- [6] M. Arashpour, R. Wakefield, N. Blismas, J. Minas, Optimization of process integration and multi-skilled resource utilization in off-site construction, *Automation in Construction*. 50 (2015) 72–80. doi:10.1016/j.autcon.2014.12.002.
- [7] M. Arashpour, R. Wakefield, N. Blismas, T. Maqsood, Autonomous production tracking for augmenting output in off-site construction, *Automation in Construction*. 53 (2015) 13–21. doi:10.1016/j.autcon.2015.03.013.
- [8] M. Arashpour, R. Wakefield, B. Abbasi, E.W.M. Lee, J. Minas, Off-site construction optimization: Sequencing multiple job classes with time constraints, *Automation in Construction*. 71 (2016) 262–270. doi:10.1016/j.autcon.2016.08.001.
- [9] M. Arashpour, Y. Bai, G. Aranda-mena, A. Bab-Hadiashar, R. Hosseini, P. Kalutara, Optimizing decisions in advanced manufacturing of prefabricated products: Theorizing supply chain configurations in off-site construction, *Automation in Construction*. 84 (2017) 146–153. doi:10.1016/j.autcon.2017.08.032.
- [10] R.F. Deckro, J.E. Hebert, W.A. Verdini, P.H. Grimsrud, S. Venkateshwar, Nonlinear time/cost tradeoff models in project management, *Computers and Industrial Engineering*. 28 (1995) 219–229. doi:10.1016/0360-8352(94)00199-W.

- [11] P. Vrat, C. Kriengkrairut, A goal programming model for project crashing with piecewise linear time-cost trade-off, *Engineering Costs and Production Economics*. 10 (1986) 161–172. doi:10.1016/0167-188X(86)90010-8.
- [12] M. Diaby, J.M. Cruz, A.L. Nsakanda, Project crashing in the presence of general non-linear activity time reduction costs, *International Journal of Operational Research*. 12 (2011) 318. doi:10.1504/IJOR.2011.042919.
- [13] D.R. Fulkerson, A Network Flow Computation for Project Cost Curves, *Management Science*. 7 (1961) 167–178. doi:10.1287/mnsc.7.2.167.
- [14] J.E. Kelley, Critical-Path Planning and Scheduling: Mathematical Basis, *Operations Research*. 9 (1961) 296–320. doi:10.1287/opre.9.3.296.
- [15] F. Khosrowshahi, The optimum project duration and cost curve for Hong Kong public housing projects, *Engineering, Construction and Architectural Management*. 4 (1997) 249–269. doi:10.1108/eb021052.
- [16] W.L. Meyer, L.R. Shaffer, Extending CPM for Multiform Project Time - Cost Curves, *Journal of the Construction Division*. 91 (1965) 45–68.
<http://cedb.asce.org/CEDBsearch/record.jsp?dockey=0013714> (last access 3rd May 2018)
- [17] K. Nonobe, T. Ibaraki, A metaheuristic approach to the resource constrained project scheduling with variable activity durations and convex cost functions, in: *International Series in Operations Research and Management Science*, Springer New York LLC, 2006: pp. 225–248. ISSN: 08848289.
- [18] J. Goh, N.G. Hall, Total Cost Control in Project Management via Satisficing, *Management Science*. 59 (2013) 1354–1372. doi:10.1287/mnsc.1120.1653.
- [19] M.J. Liberatore, B. Pollack-Johnson, Extending project time–cost analysis by removing precedence relationships and task streaming, *International Journal of Project Management*. 24 (2006) 529–535. doi:10.1016/j.ijproman.2006.04.004.
- [20] R. Deckro, J. Hebert, Resource constrained project crashing, *Omega*. 17 (1989) 69–79. doi:10.1016/0305-0483(89)90022-4.
- [21] L. Sunde, S. Lichtenberg, Net-present-value cost/time tradeoff, *International Journal of Project Management*. 13 (1995) 45–49. doi:10.1016/0263-7863(95)95703-G.
- [22] T. Ahn, S.S. Erenguc, The resource constrained project scheduling problem with multiple crashable modes: A heuristic procedure, *European Journal of Operational Research*. 107 (1998) 250–259. doi:10.1016/S0377-2217(97)00331-7.
- [23] S.S. Erenguc, T. Ahn, D.G. Conway, The resource constrained project scheduling problem with multiple crashable modes: An exact solution method, *Naval Research Logistics*. 48 (2001) 107–127. doi:10.1002/1520-6750(200103)48:2<107::AID-NAV1>3.0.CO;2-9.
- [24] M. Hariga, A. Shamayleh, F. El-Wehedi, Integrated time-cost tradeoff and resources leveling problems with allowed activity splitting, *International Transactions in Operational Research*. (2016). doi:10.1111/itor.12329.
- [25] W. Peng, H. Yu, DTCTP with renewable and nonrenewable resources, in: *2008 Chinese Control and Decision Conference*, IEEE, 2008: pp. 170–175. doi:10.1109/CCDC.2008.4597292.

- [26] E.Y.H. Lin, An alternate solution approach towards the time-cost tradeoff problem on activity-on-arc project network, *International Journal of Operations and Quantitative Management*. 17 (2011) 49–57. ISSN: 10821910.
- [27] I.-T. Yang, Performing complex project crashing analysis with aid of particle swarm optimization algorithm, *International Journal of Project Management*. 25 (2007) 637–646. doi:10.1016/j.ijproman.2006.11.001.
- [28] W.S. Khalaf, L.W. June, Cost reduction for the project completion in shortest possible duration by stretching noncritical activities, *Australian Journal of Basic and Applied Sciences*. 3 (2009) 4526–4533. <http://www.ajbasweb.com/old/ajbas/2009/4526-4533.pdf> (last access 3rd May 2018)
- [29] W.S. Khalaf, L.W. June, M.R.B. Abu Bakar, L.L. Soon, A linear programming approach to maximize savings by stretching noncritical activities, *Australian Journal of Basic and Applied Sciences*. 4 (2010) 5649–5657. <http://psasir.upm.edu.my/24822/> (last access 3rd May 2018)
- [30] S. Perera, Linear programming solution to Network compression, *Journal of the Construction Division*. 106 (1980) 315–326. <http://cedb.asce.org/CEDBsearch/record.jsp?dockey=0009721> (last access 3rd May 2018)
- [31] S.-T. Liu, Fuzzy activity times in critical path and project crashing problems, *Cybernetics and Systems*. 34 (2003) 161–172. doi:10.1080/01969720302865.
- [32] F.-T. Lin, Time-cost tradeoff problem based on confidence-interval estimates and level $(1 - \alpha)$ fuzzy numbers, *International Journal of Innovative Computing, Information and Control*. 4 (2008) 2551–2564. <http://www.ijcic.org/07-is01-11-1.pdf> (last access 3rd May, 2018)
- [33] E. Ehsani, N. Kazemi, E.U. Olugu, E.H. Grosse, K. Schwindl, Applying fuzzy multi-objective linear programming to a project management decision with nonlinear fuzzy membership functions, *Neural Computing and Applications*. 28 (2017) 2193–2206. doi:10.1007/s00521-015-2160-0.
- [34] R.A. Bowman, Stochastic gradient-based time-cost tradeoffs in PERT networks using simulation, *Annals of Operations Research*. 53 (1994) 533–551. doi:10.1007/BF02136842.
- [35] G. Zhu, J.F. Bard, G. Yu, A two-stage stochastic programming approach for project planning with uncertain activity durations, *Journal of Scheduling*. 10 (2007) 167–180. doi:10.1007/s10951-007-0008-x.
- [36] S. Cho, Z. Covaliu, Sequential estimation and crashing in PERT networks with statistical dependence, *International Journal of Industrial Engineering : Theory Applications and Practice*. 10 (2003) 391–399. ISSN: 10724761.
- [37] A.F. Chalabi, T.W. Staehr, R. Lai, Interactive construction control system for project planning and estimating, in: *Proceedings of the International Conference on Fire Safety*, ASCE, 1980: pp. 140–150. <https://bit.ly/2rg19Ie> (last access 3rd May 2018)
- [38] W.S. Khalaf, L.W. June, A linear programming approach for the project controlling, *Research Journal of Applied Sciences*. 4 (2009) 202–212. ISSN: 1815932X. Stable URL: <http://medwelljournals.com/abstract/?doi=rjasci.2009.202.212>

- [39] G.Y. Abbasi, A.M. Mukattash, Crashing PERT networks using mathematical programming, *International Journal of Project Management*. 19 (2001) 181–188. doi:10.1016/S0263-7863(99)00061-7.
- [40] A.J.G. Babu, N. Suresh, Project management with time, cost, and quality considerations, *European Journal of Operational Research*. 88 (1996) 320–327. doi:10.1016/0377-2217(94)00202-9.
- [41] J.Y. Kim, C.W. Kang, I.K. Hwang, A practical approach to project scheduling: considering the potential quality loss cost in the time–cost tradeoff problem, *International Journal of Project Management*. 30 (2012) 264–272. doi:10.1016/j.ijproman.2011.05.004.
- [42] O. Icmeli, S.S. Erenguc, The resource constrained time/cost tradeoff project scheduling problem with discounted cash flows, *Journal of Operations Management*. 14 (1996) 255–275. doi:10.1016/0272-6963(95)00025-9.
- [43] H. Nikoomaram, F.H. Lotfi, J. Jassbi, M.R. Shahriari, A new mathematical model for time cost trade-off problem with budget limitation based on time value of money, *Applied Mathematical Sciences*. 4 (2010) 3107–3119. <https://pdfs.semanticscholar.org/1c93/905720a390685304c07862d9162a7a6c9f06.pdf> (last access 3rd May 2018)
- [44] D. Kuchta, P. L'Ebraly, E. Ptasińska, Agile-similar approach in traditional project management a generalisation of the crashing model, in: *Lecture Notes in Business Information Processing*, SciTePress, 2016: pp. 318–326. ISBN: 9789897581878.
- [45] T. Hegazy, W. Menesi, Heuristic Method for Satisfying Both Deadlines and Resource Constraints, *Journal of Construction Engineering and Management*. 138 (2012) 688–696. doi:10.1061/(ASCE)CO.1943-7862.0000483.
- [46] C. Arun, B.N. Rao, Knowledge based decision support tool for duration and cost overrun analysis of highway construction projects, *Journal of the Institution of Engineers (India): Civil Engineering Division*. 88 (2007) 27–33. ISSN: 03731995.
- [47] H.-S. Lee, S. Roh, M.-S. Park, H.-G. Ryu, Optimal option selection for finishing works of high-rise building, *KSCE Journal of Civil Engineering*. 14 (2010) 639–651. doi:10.1007/s12205-010-0904-z.
- [48] A.A. Elimam, B. Dodin, Project scheduling in optimizing integrated supply chain operations, *European Journal of Operational Research*. 224 (2013) 530–541. doi:10.1016/j.ejor.2012.09.007.
- [49] T. Ng, C. Sai On, M.M. Kumaraswamy, K.K.M. Choy, Selection of activities to be crashed for mitigating construction delays, *HKIE Transactions Hong Kong Institution of Engineers*. 11 (2004) 42–47. doi:10.1080/1023697X.2004.10667942.
- [50] K. Hazini, R. Dehghan, J.Y. Ruwanpura, Enhancing schedule compression process using evolutionary optimization techniques, in: *ISEC 2013 - 7th International Structural Engineering and Construction Conference: New Developments in Structural Engineering and Construction*, Research Publishing Services, 2013: pp. 1267–1272. doi:10.3850/978-981-07-5354-2-CPM-6-60.
- [51] R. Dehghan, K. Hazini, J. Ruwanpura, Optimization of overlapping activities in the design phase of construction projects, *Automation in Construction*. 59 (2015) 81–95. doi:10.1016/j.autcon.2015.08.004.

- [52] I.M. Srour, M.A.U. Abdul-Malak, A.A. Yassine, M. Ramadan, A methodology for scheduling overlapped design activities based on dependency information, *Automation in Construction*. 29 (2013) 1–11. doi:10.1016/j.autcon.2012.08.001.
- [53] B. Dodin, A.A. Elimam, Integration of equipment planning and project scheduling, *European Journal of Operational Research*. 184 (2008) 962–980. doi:10.1016/j.ejor.2006.09.104.
- [54] J.E.V. Gerik, R.Y. Qassim, Project Acceleration via Activity Crashing, Overlapping, and Substitution, *IEEE Transactions on Engineering Management*. 55 (2008) 590–601. doi:10.1109/TEM.2008.927786.
- [55] K. Hazini, R. Dehghan, J. Ruwanpura, A heuristic method to determine optimum degree of activity accelerating and overlapping in schedule compression, *Canadian Journal of Civil Engineering*. 40 (2013) 382–391. doi:10.1139/cjce-2012-0380.
- [56] K. Hazini, R. Dehghan, J. Ruwanpura, An evolutionary optimization method to determine optimum degree of activity accelerating and overlapping in schedule compression, *Canadian Journal of Civil Engineering*. 41 (2014) 333–342. doi:10.1139/cjce-2013-0194.
- [57] C. Meier, T.R. Browning, A.A. Yassine, U. Walter, The Cost of Speed: Work Policies for Crashing and Overlapping in Product Development Projects, *IEEE Transactions on Engineering Management*. 62 (2015) 237–255. doi:10.1109/TEM.2015.2411514.
- [58] C. Meier, A.A. Yassine, T.R. Browning, U. Walter, Optimizing time–cost trade-offs in product development projects with a multi-objective evolutionary algorithm, *Research in Engineering Design*. 27 (2016) 347–366. doi:10.1007/s00163-016-0222-7.
- [59] Z. Abuwarda, T. Hegazy, Flexible Activity Relations to Support Optimum Schedule Acceleration, *Journal of Construction Engineering and Management*. 142 (2016) 6016004. doi:10.1061/(ASCE)CO.1943-7862.0001193.
- [60] A. Sols, Fast-tracking and crashing projects: Comprehensive analysis of reasons and of implementation strategies, in: 2016 International Annual Conference of the American Society for Engineering Management, ASEM 2016. <http://www.proceedings.com/33163.html> (last access 3rd May 2018)
- [61] M. Vanhoucke, *Project Management with Dynamic Scheduling*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-25175-7.
- [62] P. Ballesteros-Pérez, M-PERT. A manual project duration estimation technique for teaching scheduling basics, *Journal of Construction Engineering and Management*. 143 (2017) 4017063. doi:10.1061/(ASCE)CO.1943-7862.0001358.
- [63] P. Ballesteros-Pérez, Modelling the boundaries of project fast-tracking, *Automation in Construction*. 84 (2017) 231–241. doi:10.1016/j.autcon.2017.09.006.
- [64] K. Hazini, R. Dehghan, J. Ruwanpura, A heuristic method to determine optimum degree of activity accelerating and overlapping in schedule compression, *Canadian Journal of Civil Engineering*. 40 (2013) 382–391. doi:10.1139/cjce-2012-0380.
- [65] D.S. Hochbaum, A polynomial time repeated cuts algorithm for the time cost tradeoff problem: The linear and convex crashing cost deadline problem, *Computers & Industrial Engineering*. 95 (2016) 64–71. doi:10.1016/J.CIE.2016.02.018.

<i>Crashing option a): Collaborative resources (shared non pre-allocated work)</i>		
d_{s_i}	n_i	c_i
Independent (decision) variable <i>Continuous</i>	$\left(\frac{d_i}{d_{s_i}}\right)^{\frac{1}{1-\alpha_i}}$	$r_i + (m_i + d_{s_i} v_i) \left(\frac{d_i}{d_{s_i}}\right)^{\frac{1}{1-\alpha_i}}$
$\frac{d_i}{n_i^{(1-\alpha_i)}}$	Independent (decision) variable <i>Discrete or Continuous</i>	$r_i + n_i m_i + n_i^{\alpha_i} d_i v_i$
Implicit expression $c_i = r_i + (m_i + d_{s_i} v_i) \left(\frac{d_i}{d_{s_i}}\right)^{\frac{1}{1-\alpha_i}}$ Or once n_i is known by: $\frac{d_i}{n_i^{(1-\alpha_i)}}$	Implicit expression $c_i = r_i + n_i m_i + n_i^{\alpha_i} d_i v_i$ Or once d_{s_i} is known by: $\frac{c_i - r_i}{m_i + d_{s_i} v_i}$	Independent (decision) variable <i>Continuous</i>

<i>Crashing option b): Non collaborative resources (Pre-allocated non-shared work)</i>		
d_{s_i}	n_i	c_i
Independent (decision) variable <i>Continuous</i>	(Approximated & Deterministic) implicit expression $d_{s_i} = \frac{\mu_i}{n_i^{1-\alpha_i}} + \frac{0.78 \sigma_i LN n_i}{n_i^{\frac{1}{2}-\alpha_i}}$	Once n_i is known, then: $r_i + n_i (m_i + d_{s_i} v_i)$
Deterministic $\frac{\mu_i}{n_i^{1-\alpha_i}} + \frac{0.78 \sigma_i LN n_i}{n_i^{\frac{1}{2}-\alpha_i}}$ Stochastic $G(\theta_i, \varphi_i)$ with $\theta_i = \frac{\mu_i}{n_i^{1-\alpha_i}} - \frac{\sigma_i}{n_i^{\frac{1}{2}-\alpha_i}} (0.78 LN n_i - 0.45)$ $\varphi_i = \frac{0.78 \sigma_i}{n_i^{\frac{1}{2}-\alpha_i}}$	Independent (decision) variable <i>Discrete</i>	Once the (deterministic or stochastic) value of d_{s_i} is known, replace it in this expression: $r_i + n_i (m_i + d_{s_i} v_i)$
(Approximated & Deterministic) expression once n_i is known: $\frac{\mu_i}{n_i^{1-\alpha_i}} + \frac{0.78 \sigma_i LN n_i}{n_i^{\frac{1}{2}-\alpha_i}}$	(Approximated & Deterministic) implicit expression $c_i = r_i + n_i m_i + n_i^{\alpha_i} \mu_i v_i + 0.78 n_i^{\frac{1}{2}+\alpha_i} \sigma_i v_i LN n_i$	Independent (decision) variable <i>Continuous</i>

Table 1: Summary of the mathematical relationships between the crashed activity duration (d_{s_i}), the number of resources required (n_i) and the crashed activity cost (c_i) in the case of both collaborative (top) and non collaborative (bottom) resources activity crashing

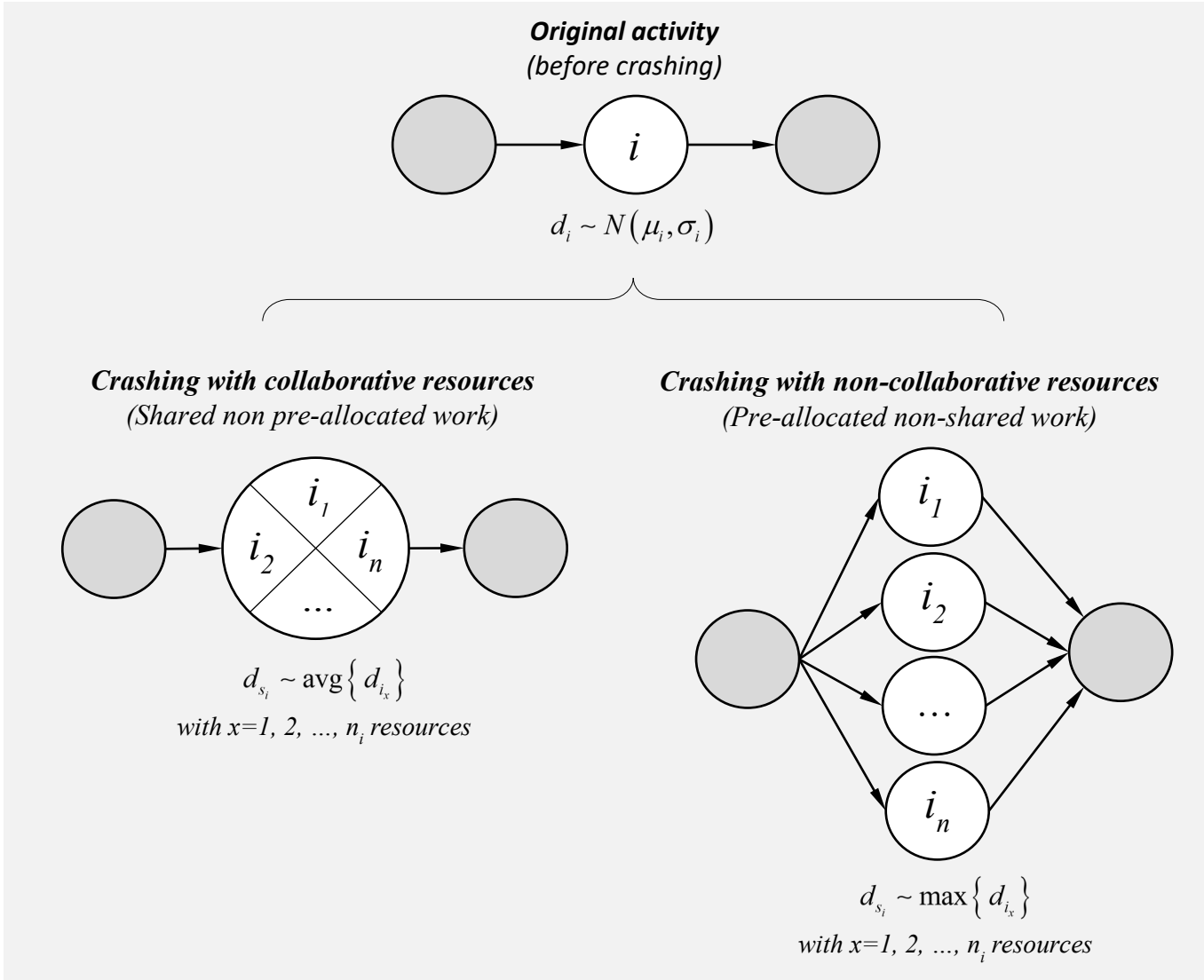


Figure 1: Two alternative crashing scenarios

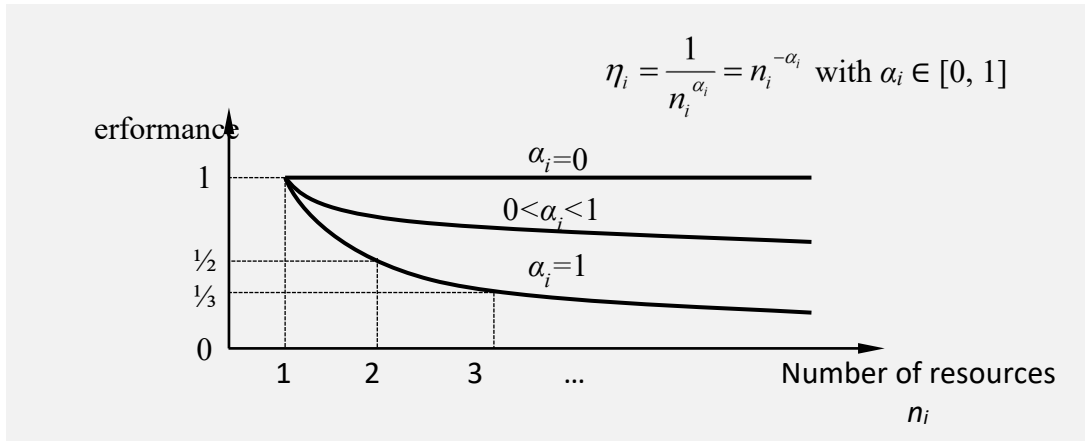


Figure 2: Relationship between resource performance coefficient (η_i) and number of resources (n_i)

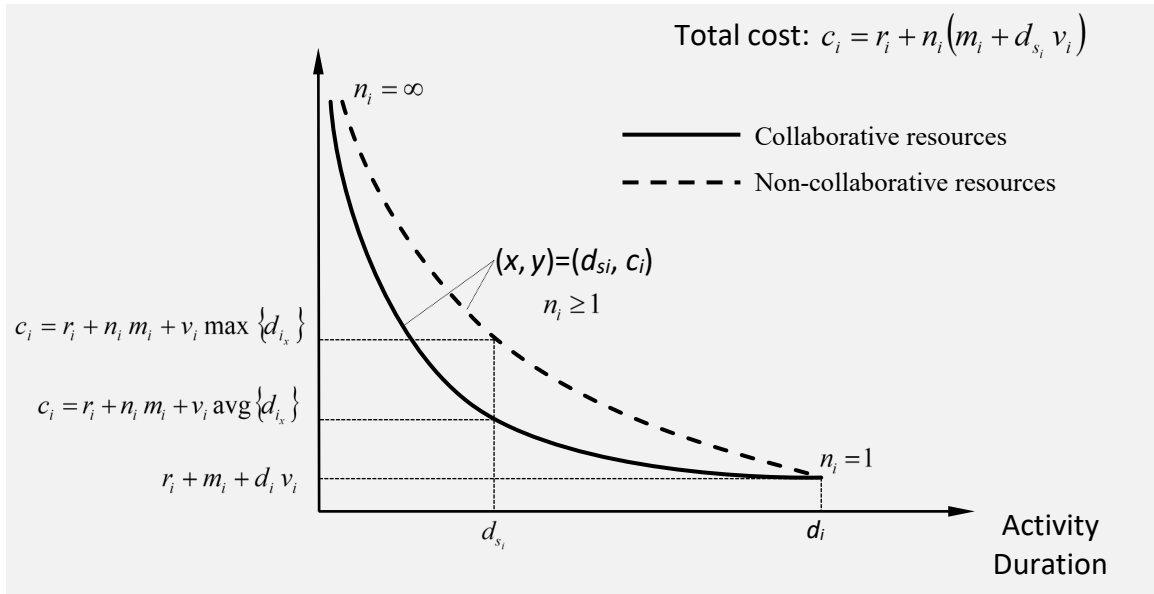


Figure 3: Activity time-cost trade-off curves for both crashing alternatives

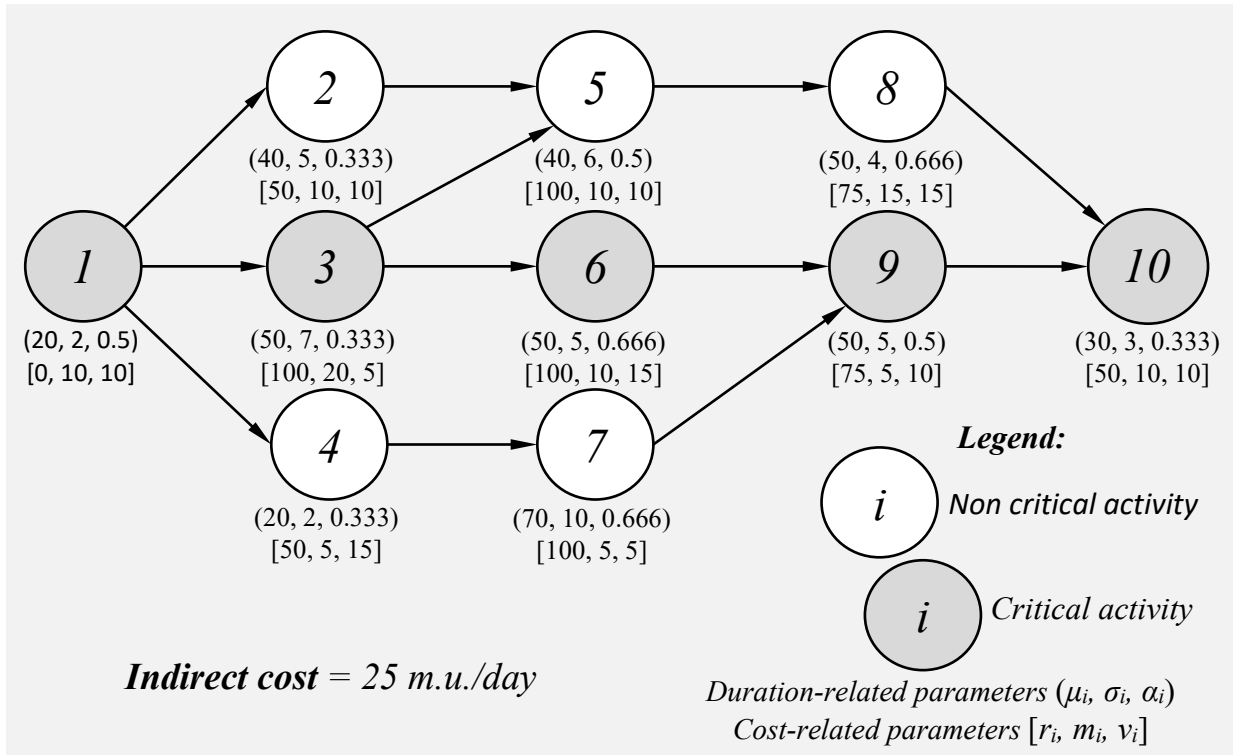


Figure 4: Activity on Node (AoN) network for the fictitious case study with all required activity information

<i>Activity (ID)</i>		1	2	3	4	5	6	7	8	9	10
<i>Initial values</i>	μ_i	20	40	50	20	40	50	70	50	50	30
	σ_i	2	5	7	2	6	5	10	4	5	3
	α_i	0.5	0.333	0.333	0.333	0.5	0.666	0.666	0.666	0.5	0.333
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	r_i	0	50	100	50	100	100	100	75	75	50
	m_i	10	10	20	5	10	10	5	15	5	10
	v_i	10	10	5	15	10	15	5	15	10	10
<i>Best solutions found</i>	<i>Act. (ID)</i>	1	2	3	4	5	6	7	8	9	10
<i>Initial (uncompressed) schedule</i>	d_{s_i}	20.00	40.00	50.00	20.00	40.00	50.00	70.00	50.00	50.00	30.00
	o_i	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<i>Crashing only</i>	d_{s_i}	14.02	32.66	30.99	15.56	36.50	49.84	65.34	50.00	38.15	13.04
	o_i	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n_i	2.03	1.36	2.05	1.46	1.20	1.01	1.23	1.00	1.72	3.49
<i>Fast-tracking only</i>	d_{s_i}	20.00	40.00	50.00	20.00	40.00	50.00	70.00	50.00	50.00	30.00
	o_i	-	0.00	10.00	0.00	0.00	0.00	0.00	0.00	15.00	25.00
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<i>Crashing & Fast-tracking</i>	d_{s_i}	16.29	35.59	35.59	17.82	38.32	50.00	68.62	49.65	50.00	25.00
	o_i	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.00	25.00
	n_i	1.51	1.19	1.66	1.19	1.09	1.00	1.06	1.02	1.00	1.31

	Project Duration		Activity costs (by default)		Compression costs		(minimise)	Avg. compress.
	(days)	% compress.	Direct costs	Indirect costs	Crashing	Fast-tracking	Total cost	cost/day
<i>Initial (uncompressed) schedule</i>	200.00	0.0%	5,000	5,000	0	0	10,000	0
<i>Crashing only</i>	146.22	26.9%	5,000	3,656	709	0	9,365	13.18
<i>Fast-tracking only</i>	172.45	13.8%	5,000	4,311	0	215	9,526	7.81
<i>Crashing & Fast-tracking</i>	153.69	23.2%	5,000	3,842	231	198	9,271	9.26

Figure 5: First set of solutions for the case study: Collaborative resources.
Independent (decision) variables: d_{s_i} (continuous) and/or o_i (continuous).
Dependent variables: n_i and c_i (both continuous and stochastic)

<i>Activity (ID)</i>		1	2	3	4	5	6	7	8	9	10
<i>Initial values</i>	μ_i	20	40	50	20	40	50	70	50	50	30
	σ_i	2	5	7	2	6	5	10	4	5	3
	α_i	0.5	0.333	0.333	0.333	0.5	0.666	0.666	0.666	0.5	0.333
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	r_i	0	50	100	50	100	100	100	75	75	50
	m_i	10	10	20	5	10	10	5	15	5	10
	v_i	10	10	5	15	10	15	5	15	10	10
<i>Best solutions found</i>	<i>Act. (ID)</i>	1	2	3	4	5	6	7	8	9	10
<i>Initial (uncompressed) schedule</i>	d_{s_i}	20.00	40.00	50.00	20.00	40.00	50.00	70.00	50.00	50.00	30.00
	o_i	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<i>Crashing only</i>	d_{s_i}	15.22	40.00	34.86	13.56	40.00	50.00	70.00	50.00	50.00	20.34
	o_i	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	n_i	2.00	1.00	2.00	2.00	1.00	1.00	1.00	1.00	1.00	2.00
<i>Fast-tracking only</i>	d_{s_i}	20.00	40.00	50.00	20.00	40.00	50.00	70.00	50.00	50.00	30.00
	o_i	-	0.00	10.00	0.00	0.00	0.00	0.00	0.00	15.00	25.00
	n_i	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<i>Crashing & Fast-tracking</i>	d_{s_i}	15.22	40.00	34.86	13.56	40.00	50.00	70.00	50.00	50.00	20.34
	o_i	-	0.00	7.61	0.00	0.00	0.00	0.00	0.00	0.00	20.34
	n_i	2.00	1.00	2.00	2.00	1.00	1.00	1.00	1.00	1.00	2.00

	Project Duration		Activity costs (by default)		Compression costs		(minimise)	Avg. compress.
	(days)	% compress.	Direct costs	Indirect costs	Crashing	Fast-tracking	Total cost	cost/day
<i>Initial (uncompressed) schedule</i>	200.00	0.0%	5,000	5,000	0	0	10,000	0
<i>Crashing only</i>	170.42	14.8%	5,000	4,261	462	0	9,722	15.61
<i>Fast-tracking only</i>	172.45	13.8%	5,000	4,311	0	215	9,526	7.81
<i>Crashing & Fast-tracking</i>	157.63	21.2%	5,000	3,941	462	25	9,427	11.48

Figure 6: Second set of solutions for the case study: Non-collaborative resources.
Independent (decision) variables: n_i (discrete) and/or o_i (continuous).
Dependent variables: d_{s_i} and c_i (both discrete and deterministic)