# *Modelling the boundaries of project fast-tracking*

Article

Accepted Version

It is advisable to refer to the publisher's version if you intend to cite from the work.  See Guidance on citing.

## www.reading.ac.uk/centaur

**CentAUR**

Central Archive at the University of Reading

Reading's research outputs online

# Modelling the boundaries of project fast-tracking

**Abstract**

Fast-tracking a project involves carrying out sequential activities in parallel, partially overriding their original order of precedence, to reduce the overall project duration. The current predominant mathematical models of fast-tracking are based on the concepts of activity sensitivity, evolution, dependency and, sometimes, information exchange uncertainty, and aim to determine optimum activity overlaps. However, these models require some subjective inputs from the scheduler and most of them neglect the merge event bias.

In this paper, a stochastic model for schedule fast-tracking is proposed. Relevant findings highlight the existence of a pseudo-physical barrier that suggests that the possibility of shortening a schedule by more than a quarter of its original duration is highly unlikely. The explicit non-linear relationship between cost and overlap has also been quantified for the first time. Finally, manual calculations using the new model are compared with results from a Genetic Algorithm through a case study.

## 1. Introduction

Fast-tracking involves performing activities, initially viewed as sequential, in parallel by overlapping their execution. It is considered to be one of the three most common schedule compression or acceleration techniques, along with activity "crashing" and activity "substitution" [1]. However, unlike activity crashing and substitution, which generally increase project costs directly, activity overlapping is thought to increase project risk due to an increase in the potential for change and/or rework [2,3] which can lead to increased costs.

The first papers describing the implementation of fast-tracking practices in construction projects were written by Ruby (1978) and Baker and Boyd (1983). Ruby described how "phased construction", as it was previously called, could significantly shorten plant construction projects by allowing an early start for certain long lead time project phases. Baker and Boyd noted all the challenges for successful fast-tracking of a Nuclear Power Plant construction project in the Gulf States. In this project, construction costs amounted to one million US dollars per extra day of execution.

Numerous publications analysing the practical considerations for implementation of fast-tracking practices in a number of settings have been published since then. Some examples include: construction of oil pools [6], automobile instrument panel development [7], subsea tieback pipeline projects [8], surface water conversion systems to reduce groundwater usage [9], fluid catalytic cracking plant revamps [10], installation of spectrographs in astronomy observatories [11], etc. These papers discuss the varied challenges and repercussions of a shortened schedule in a real context.

Hence, it is clear from all of these studies that fast-tracking is not risk-free, and overlapping dependent activities can negatively impact project performance and has the potential to raise project costs [12]. Indeed, recent reviews of large scale pipeline projects have shown that fast-tracking during the construction stage doubles the probability of project failure,

and concurrent design (fast tracking during the engineering phase) multiplies risk by a factor of four [13].

This is also probably why fast-tracking has attracted interest from an organisational point of view too. In this regard, several studies have examined how team coordination, the flow of information (including feedback loops) and some organisational structures can hamper or facilitate the implementation of fast-tracking practices [7,14–17].

From the mathematical point of view, a number of models have been developed to analyse activity overlapping and concurrent engineering. As these models are highly relevant to this study, they will be discussed separately in the literature review.

With regards scheduling algorithms and computational methods, Genetic Algorithms (GA) have been used most frequently in fast-tracking computer applications by researchers as they provide quicker and more accurate solutions in comparison with other non-linear optimisation methods [18–21]. Mixed Integer Linear and Non-Linear programming models [1], mixed algorithmic approaches (e.g. [22,23]), and other heuristic methods [2] have also been used. However, these latter methods have more commonly been applied to the simultaneous implementation of two or three time–cost trade-off techniques (crashing, overlapping and substitution), and have normally required simplifying assumptions such as a linear relationship between cost and overlapping, the independence of overlapping and crashing, single path scenarios, etc. Finally, branch and bound algorithms have also been developed and have proven to be particularly effective when fast-tracking resource-constrained schedules [24,25].

The current research attempts to make a contribution to the area of fast-track modelling by proposing a novel stochastic activity overlapping model. The model captures the type of information used by most previous models and algorithms, but employs an alternative parametrisation that allows for the use of simpler explicit expressions without any loss of

generality. New project-level insights will be provided concerning the relationship between predecessor(s)–successor overlap times, costs and the probability (risk) of an unsuccessful overlap. A case study will be used to compare manually calculated results from this model with those from a Genetic Algorithm. The difficulties of fast-tracking a project from a mathematical point of view will be considered in the *Discussion*, and a summary of the findings and model limitations will be presented in the *Conclusions*.

## 2. Materials and Methods

### 2.1. Literature review

The literature discussing the implications of overlapping and rework in product development and construction projects is plentiful [26]. Recently, Dehghan and Ruwnapura [27] and Dehghan et al. [20] presented a thorough review of the current and previous fast-tracking models for both product development and project design and execution. This review makes no attempt to be as comprehensive, but instead focuses on identifying the most well-known models and highlighting their major contributions and limitations. The models will be reviewed in chronological order; their most significant contributions are highlighted in Figure 1.
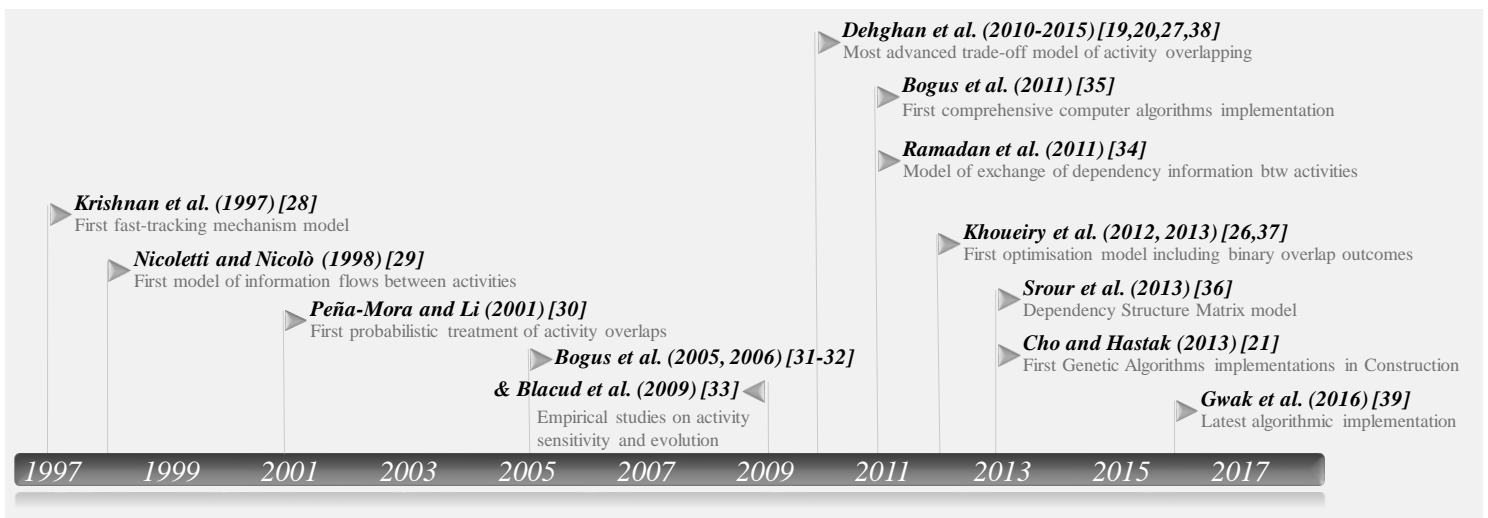


**Figure 1:** Timeline of major contributions to project fast-tracking research since 1997.

One of the first mathematical models of fast-tracking was published by Krishnan et al. in 1997 [28]. Their model focused on the activity-to-activity attributes of sensitivity (how quickly a predecessor activity releases information) and evolution (how quickly a successor activity progresses). Based on these concepts, they developed a model that, despite its simplicity, constituted the first good representation of the fast-tracking mechanism. The model proposed in this paper also includes the potential for incorporating different activity sensitivities and evolutions, but uses an alternative parametrisation.

In 1998, Nicoletti and Nicolò [29] developed a first decision support tool that modelled the information flows between activities in order to identify which activities should be overlapped and to what extent. The biggest limitation (simplification) of their model was that it assumed project completion time was not critical, which is not generally true. They did, however, include similar concepts to activity sensitivity and evolution, and managed to incorporate them into a complete project schedule for the first time.

Three years later, in 2001, Peña-Mora and Li [30] developed a dynamic planning and control methodology by integrating axiomatic design, concurrent engineering, the graphical evaluation and review technique (GERT) and system dynamics modelling. The major contribution of this work was the inclusion of a probabilistic view of activity overlaps. The main limitation of their model is that it can only be applied to pairs of activities. The model proposed here makes use of a similar probabilistic approach, but can be applied to complete schedules.

Between 2005 and 2009, Bogus et al. and Blacud et al. [31–33], contributed to the study of activity evolution and sensitivity through a series of expert interviews. They identified the aspects which make an activity more or less sensitive or make it evolve to a greater or lesser extent. Their studies took the first steps to translating the information gathered from real project contexts into inputs for mathematical models. However, their eminently qualitative approach

still require further research efforts before those contributions can be translated into fully quantitative models like the one proposed here.

Ramadan et al. in 2011 [34] developed a methodology for capturing and quantifying the exchange of dependency information between pairs of activities once it is known how sensitive they are and how they evolve. These attributes are indirectly accommodated in the model here as part of *Risk* which is related to the probability of achieving a successful overlap.

Following on from their previous studies on activity-to-activity sensitivity and evolution, Bogus et al. [35] implemented, still in 2011, one of the first comprehensive computer algorithms for optimising overlaps in complete schedules. This algorithm used Monte Carlo simulations to predict different discrete outcomes for each activity to obtain a more accurate understanding of the probability of rework. Monte Carlo simulations are also used in the model proposed here to estimate project duration.

In 2013, Cho and Hastak [21] developed one of the first Genetic Algorithms for fast-tracking dependent activities in construction projects. The main limitation of their model was that it considered projects with a single chain of critical activities. However, they were among the first researchers to consider compressing a schedule with multi-predecessor activities.

In the same year, Srour et al. [36] used the Dependency Structure Matrix (DSM) to improve the way dependent and interdependent relationships between activities are represented. Their approach had several advantages; for example, the DSM was able to represent (backward) feedback loops between activities. They also proposed a generic compression algorithm. However, in their work, dependencies between activities were eventually modelled based on subjective input from expert interviews. It is thus unclear from their work how risks from activity overlapping could be quantified for other project contexts.

In 2012 and 2013, Khoueiry et al. [26,37] developed another optimisation model for fast-tracking construction activities by calculating the probability of rework and its impact.

This model was possibly the first to include successful/unsuccessful overlap probabilities. The downside of this non-linear model was that it relied on the evaluation of multiple integral expressions for each overlap, making it impractical for implementation with a real project.

From 2010 to 2015, other group of researchers represented by Dehghan et al. [19,20,38] and Dehghan and Ruwanpura [27] have been among the most active researchers in the field of fast-track modelling. Over time, they have developed the most advanced trade-off model of overlapping and rework and have proposed Genetic Algorithms capable of handling multi-path networks. Their model also considers the potential for a cascade of overlaps between activities, multi-predecessor overlaps and several kinds of activity dependencies (i.e. finish to start (FS), start to start (SS), finish to finish (FF), and start to finish (SF)). The major limitations of their model as originally presented were: it handles the rework time extensions as deterministic duration values; rework durations are assumed to be linearly dependent on overlap durations; and both rework durations and cost functions require subjective input from the scheduler. The model presented later will propose alternative methods to address the first two limitations and reduce the need for the third.

In 2016, Gwak et al. [39] recently published an "exact" time–cost trade-off computational method to identify the optimum global overlaps between critical activities. It constitutes the latest and most computationally efficient algorithmic implementation of schedule fast-tracking to date. The proposed algorithm is capable of expeditiously finding the exact overlaps between critical activities, while approximately computing the probabilities of completing the fast-tracked project on time and/or on budget. However, the limitations of this model are: it offers two pre-set activity sensitivity and evolution levels which the scheduler subjectively chooses; the maximum overlap limits are also subjective; the cumulative probabilities of activity overlaps do not converge to 1; for the multi-predecessor case, probabilities are simply added instead of the predecessors being considered as simultaneous

independent events; rework durations are not treated stochastically; and it is unclear how the proposed algorithm identifies the cheapest combination of overlapping activities from different paths when there are several critical paths.

Finally, there has been some recent research on identifying and prioritising best practice [40] and developing first readiness assessment techniques [41] for faster fast-tracking or flash-tracking. Despite the fact that this area lacks mathematical models similar to those which have been developed for fast-tracking, it shows promise as an area for future research.

### 2.2. Model hypotheses and assumptions

It is commonly accepted that fast-tracking (or concurrent engineering) means carrying out in parallel, or partially overlapping, two or more activities from a project's critical path (or critical paths, if there are more than one). This entails to some extent overriding their original precedence relationships.

However, if we assume that a project schedule has been carefully considered and that any precedence relationships have been established for a reason, the maximum overlap of a particular activity with its immediate predecessor(s) should be equal to the duration the shortest predecessor. In other words, unless additional measures (normally involving additional costs) are implemented, theoretically, activities cannot be overlapped to the extent that a successor would start earlier than its predecessor(s), as could potentially occur when a successor's duration exceeds its predecessor's duration. This assumption is in line with previous fast-tracking models [20,27,39].

Similarly, no successor activity should be allowed to end earlier than its predecessor. This could occur when a predecessor's duration is longer than its successor's and the latter could in effect end up nested within the former. In this case, there would be a clear violation (as opposed to a partial override) of the associated precedence relationship. This assumption is

also included in most of the recent fast-track models in order to keep the network logic intact [20,26,27,37]. A tighter limit (less overlap) can be introduced, however, if there are other constraints, for example shared resources, space or material handling, between predecessors and their successors [39].

It is clear then, that there are limits to the overlap achievable, either precedence-related or imposed by the scheduler for technical reasons, and that unless special ad-hoc arrangements are made, the maximum overlap achievable will correspond to either the duration of the successor or the predecessor, whichever is the least.

Concerning the monetary aspects, though, there is still uncertainty about how best to model the costs associated with fast-tracking activities [37]. So far, it has only been agreed that the cost of fast-tracking is proportional to the cost of rework, and that the latter is proportional to the overlap. However, the proportions attributed to overlap and rework, or the probability of rework, are not clear. Some researchers have assumed a linear relationship between cost and overlap [20,27,37], and some exponential [26] or polynomial [39], but these choices have normally been based on subjective assumptions.

The model presented here takes a different approach. It is undeniable that there is an increase in the risk of needing to redo work corresponding to successor activities being brought forward and being overlapped with their predecessors. Rework implies there is an extra cost from repeating the fraction of the successor activity that has been overlapped and the possibility of having to re-start the successor activity at its original start date when the overlap causes unforeseen problems.

Fast-tracking is thus a double-edged sword: if all goes well, the Project Manager can shorten the project duration and there is no additional cost of rework; if not, the Project Manager will incur extra costs for rework and the successor activities will end up starting when originally planned (i.e. the project will cost more and there will be no schedule compression).

In the extreme, either a successor activity will be successfully overlapped, i.e. start earlier and incur no extra cost, or the successor will end as originally planned and an additional cost will be incurred corresponding to the fraction of the activity that was overlapped and needed to be redone. What it is very unlikely is that there is only one possible outcome and the duration of the "rework" is deterministic, i.e. the successor has a "new duration" which is the sum of its previous duration plus the duration of the estimated rework, but as it started earlier it will normally finish earlier as well (except when the rework duration is too long and/or too expensive, causing the overlap not to be time and/or cost-effective). This deterministic scenario, however, is what all previous research, with the exception of Peña-Mora and Li (2001), have assumed so far; and, maybe because of this oversimplification, it has been argued that fast-tracked projects are less predictable with regards their cost, time and quality [7,42,43].

Hence almost all previous models have calculated rework as the average duration extension after a negative overlap outcome (for example, a miscommunication of information between a predecessor and its successor). However, as stated previously, those average duration extensions (or rework) will not generally happen. Even if the schedule was assumed to be deterministic initially, there are now two stochastic outcomes for the overlapping activities: successful overlap and unsuccessful overlap. The schedule compression will therefore needed to be calculated using Stochastic Network Analysis (SNA) techniques. If not, due to the merge event bias [44,45], the actual project duration could be underestimated due to the influence of non-critical paths. Fortunately, merge event bias does not affect activity costs that, unlike project duration and compression, can be accurately represented by their average values due to their summative nature.

Therefore, concerning cost, a deterministic approach is reasonable. For instance, if the risk associated with the fast-track of an activity with a normal cost of 1,000$ is 50%, the cost of fast-tracking that activity would be on average 1,000×0.5=500$. Concerning time, however,

the model proposed here diverges from previous conceptions. For the sake of simplicity, let us first reflect on the consequences of fast-tracking (overlapping) an activity from a deterministic point of view.

If we examine the case when a successor activity would start (without fast-tracking) on day 10 and which is, when overlapped with its predecessor, brought forward by 4 days, the successor would start now on day 6. Let us assume that the risk of overlapping that successor with its predecessor is known and is 25%. Then, the average expected start date would be $10 \times 0.25 + 6 \times (1-0.25) = 7$ because the successor has a 25% chance that the activity will start on the original date of day 10 and a 75% chance the work will progress smoothly from its fast-tracked date of day 6. In this particular case, as assumed in previous models, the average amount of rework would be 1 day, which is the difference between starting on day 6 (a totally successful overlap) and on day 7 (an average overlap outcome).

However, further analysis reveals that the activity will never start on day 7. It will start on day 6 75% of the time, and on day 10 25% of the time. If we assume in successive (optimisation) compression calculations that the activity will start on day 7, we are imposing a misleading oversimplification, as any non-critical paths with floats below one day will remain as they are and the potential schedule compression which could be achieved by the activity starting on day 6 would be prevented.

Hence, contrary to what is widely believed, non-critical activities also need to be compressed below the target overlap for the critical path (i.e. the overlap achieved if it is successful, and not just below the "average" rework duration) or they will become compression bottlenecks. However, even if non-critical activities are overlapped such that they do not inhibit the successful overlap of critical activities, there is always the chance that they will themselves be unsuccessful. This means that the higher the number of subcritical paths, the lower the

potential average compression, as there will be more probability that at least one path will not be successfully compressed during the fast-tracking implementation.

Currently, there are no analytical solutions for the multi-path case with stochastic durations and exact compression can only be calculated by computer simulation [46,47]. Therefore, as a partial conclusion, fast-tracking a schedule necessarily involves at least a minimum consideration of the probabilistic dimension. The proposed model will address this specific issue.

### 2.3.Theoretical model

Let's consider two activities: a predecessor $p$ and a successor $s$, whose durations are $d_p$ and $d_s$ and whose costs are $c_p$ and $c_s$ respectively. The successor activity's cost $c_s$ can be divided into two components: the upfront cost $c_{sa}$ (which is incurred by simply initiating the successor) and the time dependent cost $c_{sb}$ (which is incurred as long as the activity is ongoing).

The "*overlap*", or $o$, is the number of days (or units of time) during which both activities are being carried out simultaneously.

In addition, there may be a time lag $l$ between the two activities (an amount of time after the predecessor finishes and before the successor should start), as well as a slack $k$ (or float) if the successor is not on the critical path. Both variables $l$ and $k$ are initially located in between the end of the predecessor and the start of the successor. These have been included merely for representation purposes, as the model focuses on the overlap $o$ between the activities, which is irrespective of their precedence relationship (FS, SS, SF, FF).

All these elements are shown in the schematic at the centre of Figure 2. Variable $f_t$ represents the amount of time a successor activity has been brought forward and is the absolute amount of time the successor starts sooner than planned. Parameters $\alpha$ and $\beta$ represent the sensitivity and evolution of the activities, respectively, and will be detailed later.
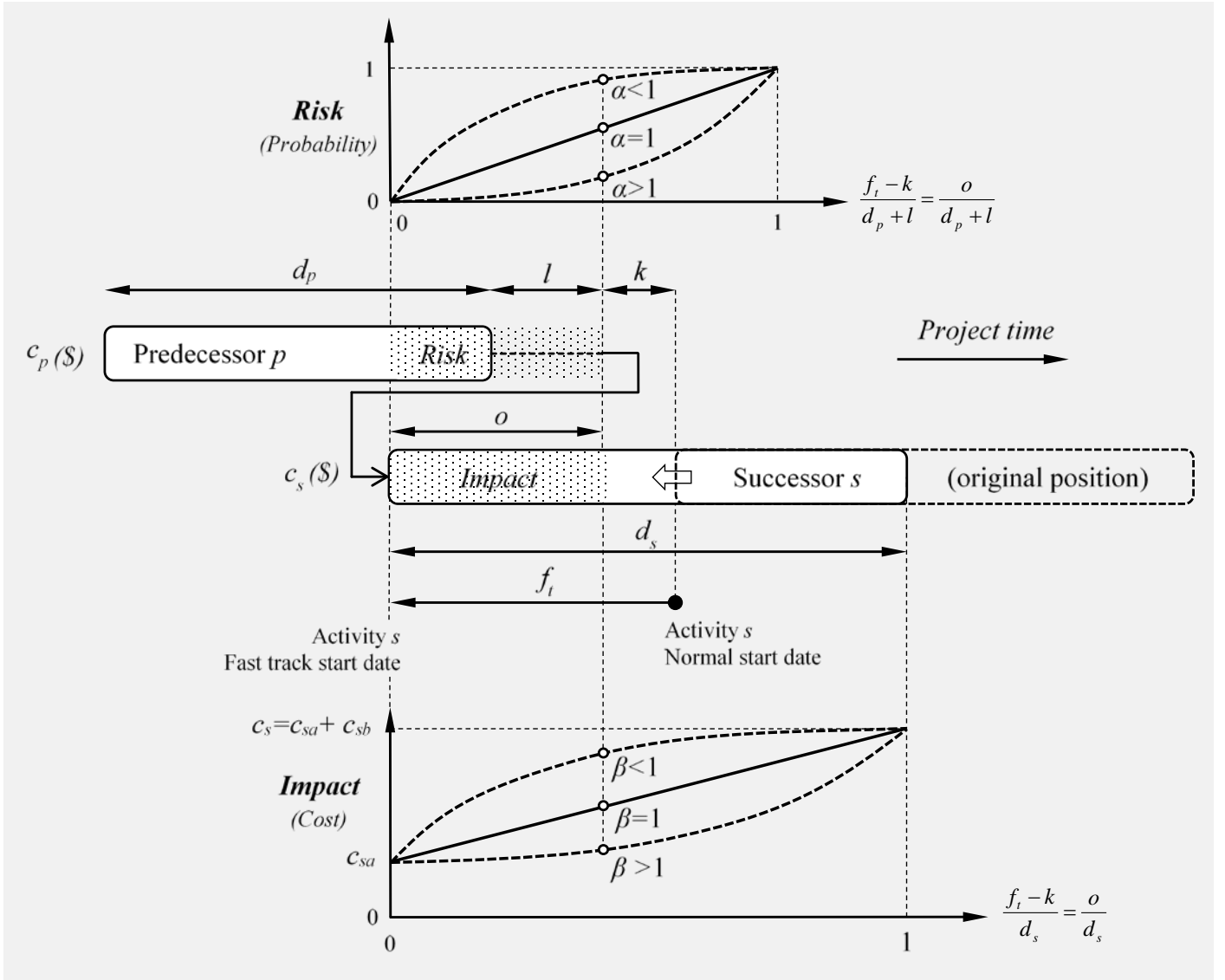
**Figure 2:** Fast-tracking risk and impact calculations

With all these elements known, as the successor is brought forward and the activities start to overlap (i.e. once $f_t > k$), the (average) cost increment ($\Delta c_s$) related to fast-tracking can be calculated as follows:

$$\Delta c_s = Risk \times Impact = \left(\frac{f_t - k}{d_p + l}\right)^{\alpha}\left(c_{sa} + \left(\frac{f_t - k}{d_s}\right)^{\beta} c_{sb}\right) = \left(\frac{o}{d_p + l}\right)^{\alpha}\left(c_{sa} + \left(\frac{o}{d_s}\right)^{\beta} c_{sb}\right)$$

with $\Delta c_s \leq c_s$, $0 \leq \alpha, \beta \leq +\infty$ and $0 \leq o \leq MIN(d_s, d_p + l)$

(1)

This way of modelling *Risk* obviously assumes that it increases proportionally to the overlap (for example, linearly when $\alpha=1$). When there is no overlap at all, *Risk*=0%, and it increases to 100% at the point where the overlap equals the predecessor's duration $d_p$ plus the lag $l$. Therefore, parameter $\alpha$ here represents the predecessor's sensitivity.

The *Impact* is clearly the proportion of the cost of the successor activity that has been overlapped (executed before it was originally intended) since that would be the part that would be repeated (and paid for again) if the overlap was unsuccessful. This component is similar to the "progress of the downstream activity" or "evolution" incorporated in previous models with a different approach. Here, the successor "evolution" is represented by a combination of the parameter $\beta$ (which models how quickly the time-dependent successor activity cost $c_{sb}$ is spent during execution; linearly for $\beta=1$) and $c_{sa}$ (the upfront successor activity cost).

Finally, the concepts of "dependency" and "changes uncertainty" are inherently included in the relationship between *Probability* and *Impact*, which in this case is their product. However, by choosing specific combinations of $\alpha$ and $\beta$, homogeneous (for $\alpha=1$) or heterogeneous (for $\alpha\neq1$) previous models' Poisson information exchange processes can also be incorporated as special cases within the proposed model [26,37,48].

Of particular interest, however, is the case where $c_{sa}\approx0$ (i.e. $c_s\approx c_{sb}$) and $\alpha=\beta=1$, as it could be representative of an average situation for most construction schedules. In construction projects, most activities consist of the execution of multiple homogeneous units (e.g. $m^3$ of concrete or excavation, $m^2$ of asphalt pavements, kg of reinforced steel) where the risk increase by overlapping two activities (modelled by $\alpha$), at least on average, is expected to be linearly proportional ($\alpha=1$) to the number of units that have already been executed in the successor by the time they start overlapping. Analogously, the successor's cost evolution (modelled by $\beta$) is also expected to grow linearly ($\beta=1$) with the successor's overlap, as the amount of executed

units over the activity total should also correspond proportionally to the total cost of the successor. Therefore, in this situation, expression (1) becomes:

$$\Delta c_s = Risk \times Impact = \left(\frac{f_t - k}{d_p + l}\right) \cdot \left(\frac{f_t - k}{d_s} \cdot c_s\right) = \frac{c_s \cdot (f_t - k)^2}{d_s(d_p + l)} = \frac{c_s \cdot o^2}{d_s(d_p + l)}$$

(2)

with $\Delta c_s \leq c_s$ and $0 \leq o \leq MIN(d_s, d_p + l)$

According to this expression, when all other variables are constant, the cost increment increases proportionally (on average) with the square of the overlap $o$. This would imply that a small amount of fast-tracking might have little effect in terms of cost. However, if a significant project compression is required, it would be better to spread the total overlap and introduce it in small amounts between many of the activities of the critical path, instead of having a significant overlap for only a few. Also, irrespective of the $\alpha$ and $\beta$ values used in expression (1), activity-to-activity overlap additional costs $\Delta c_s$ always increase nonlinearly, even though most previous models have resorted to this simplification in the past.

### 2.4. The multi-predecessor case and cascade of overlaps

If there is a successor activity $s$ with more than one predecessor $p_i$ *(i=1,2,...,n* predecessors), even when the end dates of the predecessors differ (due to different slack $k_i$ values), it is well understood that the fast-tracking cost increment ($\Delta c_s$) for activity $s$ will increase more rapidly than it would in the single-predecessor case.

In the multi-predecessor case, the *Risk* term of expression (1) will depend on the proportional overlap with each predecessor duration (plus the time lag $l_i$, if applicable). It is thus computed based on the probability of occurrence of independent statistical events, i.e. *overall Risk=1-Π(1-Risk$_i$)*, which is in line with some previous advanced models [27,36,38,42]. Other models have, for simplicity, assumed the probabilities can be added [39,49], or have used the longest average rework duration instead [50], but those simplifications are not correct.

Concerning the *Impact* term, it is obvious that the successor activity would only be repeated once, so the term would be similar to that in expression (1) with the exception that the proportion of successor activity that might be repeated would correspond to the maximum overlap ($o_{max}$) among all the individual overlaps ($o_i$) with each predecessor.

Expression (1) can thus be generalised for the case of $n$ multiple predecessors as:

$$\Delta c_s = overall\ Risk \times Impact = \left[1 - \prod_{i=1}^{n}\left(1 - \left(\frac{f_t - k_i}{d_{p_i} + l_i}\right)^{\alpha_i}\right)\right] \cdot \left[c_{sa} + \left(\frac{f_t - k_{min}}{d_s}\right)^{\beta} c_{sb}\right] =$$

$$= \left[1 - \prod_{i=1}^{n}\left(1 - \left(\frac{o_i}{d_{p_i} + l_i}\right)^{\alpha_i}\right)\right] \cdot \left[c_{sa} + \left(\frac{o_{max}}{d_s}\right)^{\beta} c_{sb}\right] \qquad (3)$$

with $\Delta c_s \leq c_s$ , $0 \leq \alpha, \beta \leq +\infty$ , $k_{min} = \min k_i$ and $0 \leq o_{max} = \max o_i \leq MIN(d_s, d_{p_i} + l_i)$

This expression is significant because many fast-tracking optimisation algorithms (mainly using Genetic Algorithms (e.g. [20,21,51]) have needed to resort to other iterative, or just mathematically simplified, expressions when considering multiple predecessors. Expression (3), however, provides the exact explicit expression for dealing with these cases.

Furthermore, expression (3) can be used to model the *cascade effect* among activities which are dependent through an intermediate activity (when a successor's successor overlaps with the first successor's predecessor). In these cases, it is only necessary to state a precedence relationship between the successor's predecessor and the successor's successor (with an FS arrow, for instance), irrespective of the fact that there is another activity in between. This way, the cascade effect is simply treated as a multi-predecessor case and no other precautions are needed.

## 3. Analysis and results

### 3.1. The boundaries of fast-tracking

So far we have modelled the cost increment related to fast-tracking an activity, but the time dimension has been neglected. However, the time dimension is paramount when understanding the effects and, particularly, the boundaries of fast-tracking. These boundaries will be explained with reference to the variables depicted in Figure 2.

As noted earlier, the *Risk* represents the probability of having to repeat an activity after it has been brought forward an amount of time $f_t$ (where $f_t=o+k$ when there is some slack $k$ left). As discussed above, overlapping involves risk, in that the activity brought forward might go back to its original position should the overlap be unsuccessful.

Now, despite that this approach can be inaccurate when estimating the whole project duration (see above), let us focus on the "average" expected time that a single activity can be "effectively" brought forward (hereafter referred to as *f*). This "effectively achieved overlap *f*" can be calculated easily by using a weighted average as follows as in a previous example:

$$f = k \cdot Risk + f_t(1 - Risk) = k \cdot Risk + (k + o)(1 - Risk) =$$
$$= k + o(1 - Risk) = k + o\left(1 - \left(\frac{o}{d_p + l}\right)^\alpha\right) \quad \text{with } f_t \geq k \tag{4}$$

Where $k$ is the maximum amount the successor activity will be able to be brought forward without increasing the risk of rework (i.e. when the time shift is simply taking up the slack $k$), and $f_t$ the "total" amount of time that the successor activity is being brought forward. It is interesting to observe that when *Risk=100%=1,* the successor activity will always have to be repeated and it will be re-started at $f_t=k$. Therefore, no matter what effort is made by the Project Manager to bring a successor activity forward, from a probabilistic point of view, the activity will remain where it was originally planned.

What is important for a Project Manager is to find where expression (4) reaches its maximum, i.e. to find out the maximum possible value of *f*. This approach was outlined by Dehghan et al. [20], but not fully developed from the analytical point of view. The way this

model has been generated greatly simplifies this task. After taking the derivative of the expression and observing at which value it equals zero, we have:

$$f \text{ is maximised when } \quad \frac{o}{d_p + l} = \left( \frac{1}{1+\alpha} \right)^{\frac{1}{\alpha}} \tag{5}$$

For the linear case ($\alpha$=1), which it has been suggested to capture the average case for most construction project schedules, $f$ is maximised when: $o=(d_p+l)/2$ or $f=k+(d_p+l)/4$ which corresponds to a *Risk*=50%.

This result is interesting since it means that "on average" fast-tracking cannot allow any successor to overlap its predecessor by more than 25% of the predecessor's duration. Therefore, for a given schedule, some activities will be successfully overlapped by 50% whereas others will have been unsuccessfully overlapped and will remain in their original position. In practical terms, this implies that it is impossible to shorten any schedule beyond 25% of its total duration, at least as long as activity "crashing" is not an option.

There have been a few empirical studies reported in the literature that also support this claim. Eldin [52], for example, examined four case studies and concluded that applying fast-tracking could reduce project schedules by up to 25% at best; and Attar et al. [53] found that fast-tracking practices were able to shorten underground construction projects by an average of 18%. Therefore, our model results seem to be in line with these two field studies measuring fast-tracking performance once these construction projects were finished.

### 3.2. The cost of fast-tracking

Using the effective time $f$ that an activity can be brought forward "on average" (from expression (4)), we can determine how much money the fast-tracking will cost in terms of schedule compression "effectively achieved". This requires two steps: first, to work out the value of $o$ in expression (4); and second, to use that value of $o$ in expression (1).

The main problem for the first step is that variable $o$ cannot be expressed explicitly in expression (4) unless $\alpha=1$ (the case where there is a linear increase in risk with activity overlap). For illustration purposes, it will be assumed in this paper that $\alpha = 1$, however, it would be a relatively easy process to numerically consider the cases where $\alpha \neq 1$, allowing a straightforward implementation of this model in future optimisation algorithms.

When $\alpha=1$, the value of variable $o$ can be calculated from expression (4) by solving this $2^{\text{nd}}$-order polynomial:

$$\frac{o^2}{d_p + l} - o + f - k = 0 \tag{6}$$

whose roots are:

$$o = \frac{d_p + l}{2}\left(1 \pm \sqrt{1 - \frac{4(f-k)}{d_p + l}}\right) \tag{7}$$

Of particular interest is the negative root, since the positive root leads to a larger value of overlap $o$ which would be more expensive as it would have a higher $\Delta c_s$ even though it would eventually achieve the same value of $f$. With this approach, for any amount we want to bring forward an activity $f$ on average, we can now calculate the corresponding overlap $o$ which needs to be reached. As explained above, the second step involves incorporating the negative root of $o$ from expression (7) in expression (1):

$$\Delta c_s = Risk \times Impact = \left(\frac{o}{d_p + l}\right) \cdot \left(c_{sa} + \left(\frac{o}{d_s}\right)^{\beta} c_{sb}\right) =$$

$$= \left(\frac{1}{2} - \sqrt{\frac{1}{4} - \frac{f-k}{d_p + l}}\right) \cdot \left(c_{sa} + \left(\frac{d_p + l}{2d_s}\left(1 - \sqrt{1 - \frac{4(f-k)}{d_p + l}}\right)\right)^{\beta} c_{sb}\right) \tag{8}$$

with $\Delta c_s \leq c_s$, $0 \leq \beta \leq +\infty$, $0 \leq o \leq MIN(d_s, d_p + l)$

and $k \leq f \leq k + MIN\left(\frac{d_p + l}{4}, d_s\right)$

Expression (8) is also of great interest, as it can be used to obtain a lower bound estimate of the average cost of fast-tracking a complete schedule ($\Delta c$) from the point of no effective compression ($f=0\%$) to the maximum possible overall average compression ($f=25\%$). In order to obtain that estimate, we simply need to assume that $k=0$; $\beta=1$; $c_{sa}=0$; $c_{sb}=c_c=$the sum of the critical activities' costs; and $d_s=d_p+l=d_m=$the average duration of the critical activities. If $f$ is now expressed as a per-unit fraction of overall compression ($f=0$ to $0.25$), with these substitutions, expression (8) is reduced to:

$$\Delta c = \left( \frac{1}{2} - \sqrt{\frac{1}{4} - f} \right)^2 c_c \quad \text{with } 0 \le f \le \frac{1}{4} \tag{9}$$

By rearranging expression (9), it can also be used to obtain an upper bound for the schedule compression that can be achieved per unit cost increment:

$$f = \sqrt{\frac{\Delta c}{c_c}} - \frac{\Delta c}{c_c} \quad \text{with } 0 \le \Delta c \le \frac{1}{4} c_c \tag{10}$$

Both expression (9) and expression (10) provide simple solutions for obtaining initial rough estimates. Indeed, for construction projects, the assumptions which underlie these expressions are quite reasonable. Most construction activities linearly incur cost as they progress, i.e. $\beta=1$, as they are normally executed in units (for example m$^3$ of concrete or excavation). The risk of overlap is also linearly proportional to the degree of progress of the predecessor at any given point, i.e. $\alpha=1$. For example, the risk of starting to assemble an HDPE pipe is proportional to the amount of sand bed that has already been prepared within the trench. Finally, many upfront costs (i.e. when $c_{sa}\neq0$) are recoverable. An exception to this might be when expensive equipment is bought and because of an incorrect specification due to evolution of the predecessor it needs to be repurchased.

A graph of expression (9) can be found in Figure 3. Also plotted is the equivalent average overlap *o* of critical activities required in order to achieve an average effective schedule compression *f*.
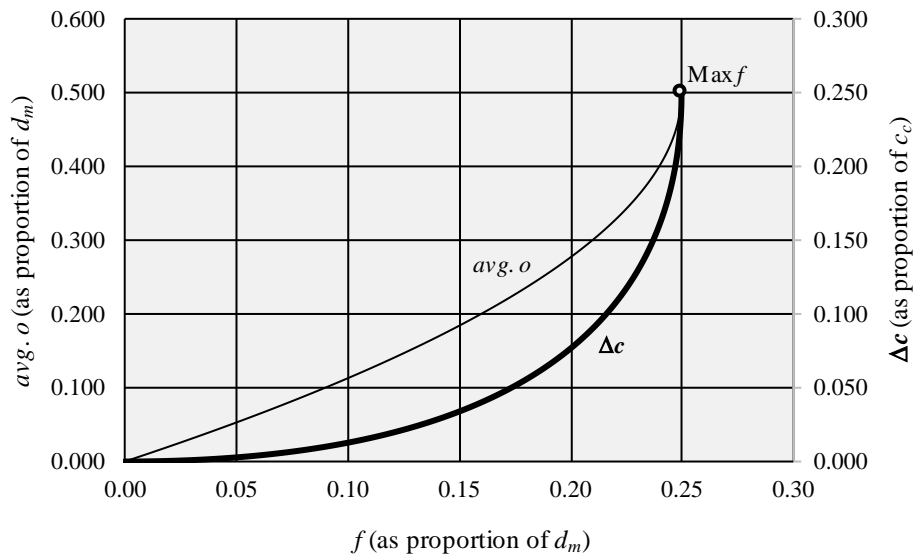


**Figure 3:** Average cost ($\Delta c$) of compressing a schedule (right axis) and average overlap (*o*) of critical activities (left axis) as a function of the effective compression (*f*)

It is important to remember that expressions (8) and (9) provide lower and upper bounds. As stated earlier, the schedule compression eventually achieved will be influenced by non-critical paths where two possible stochastic outcomes are possible for each activity (successful overlap and failed overlap). There are also other factors which will be explained later in the *Discussion*.

### 3.3. Case study

For illustration purposes, a case study is considered here despite the aim of this paper is not to illustrate a refined optimisation algorithm, rather a first algorithmic implementation. Simple parameter values used ($\alpha=\beta=1$, no time lags and $c_{sa}=0$) were chosen to examine activity duration variability versus previous model approaches resorting to deterministic average "rework" duration extensions. Concerning activity relationships, for the sake of simplicity only

finish to start (FS) relationships have been adopted. However, other non-FS logical relationships (e.g. SS, SF and FF) could also very easily have been implemented, as Lu and Lam proved in 2009 [54] that all non-FS precedence types can be reformulated as a FS relationship. Additionally and quite recently, Ballesteros-Pérez [55, figure 2] provided further graphical evidence about the feasibility of these transformations with and without activity lags.

The case study consists of a fictitious project with 12 activities whose precedence network is shown at the top of Figure 4. The duration, cost and slack for each activity are shown directly below each node (activity).

Critical activities have been highlighted in a dark grey colour in the schematic and their data are highlighted in the corresponding rows in the table below. The complete project duration initially equals 100 (days or other time units) and the cost is 1000 (money units). Before any overlap, there are two critical paths: 1-2-5-8-11-12 and 1-3-6-8-11-12.

When fast-tracking this case study with a manual approach, the objective will be to shorten the project schedule as much as possible by adjusting the overlaps between activities.

For the ease of calculation, the first three blocks of the table in Figure 4 summarise the information from the network schematic. The fourth block represents the minimum and maximum amount that each activity can be overlapped with its predecessor(s). The last block summarises the optimum solution (the one which maximises the overall schedule compression).

| Activity ID (sucessor, $s$) | Initial values | | | Predecessors ($p_i$) (FS relationips) | Predecessor durations ($d_{pi}$) | | | Overlap limits | | Activ. overlaps ($o$) (*Solution*) | Activ. overlaps cost ($\Delta c_s$) expression (3) | Overlap Risk expression (3) | Effective overlap ($f$) expression (4) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Dur. ($d_s$) | Cost ($c$) | Slack ($k$) | | | | | $o_{min}$ | $o_{max}$ | | | | |
| 1 | 10 | 50 | 0 | - | - | | | - | - | - | - | - | - |
| 2 | 25 | 50 | 0 | 1 | 10 | | | 0 | 5.00 | *5.00* | 5.00 | 0.50 | 2.50 |
| 3 | 25 | 100 | 0 | 1 | 10 | | | 0 | 5.00 | *5.00* | 10.00 | 0.50 | 2.50 |
| 4 | 20 | 50 | 5 | 1 | 10 | | | 0 | 5.00 | *0.00* | 0.00 | 0.00 | 0.00 |
| 5 | 25 | 100 | 0 | 2 | 25 | | | 0 | 12.50 | *12.50* | 25.00 | 0.50 | 6.25 |
| 6 | 25 | 100 | 0 | 3 | 25 | | | 0 | 12.50 | *12.50* | 25.00 | 0.50 | 6.25 |
| 7 | 20 | 50 | 5 | 3,4 | 25 | 20 | | 0 | 7.36 | *6.20* | 7.46 | 0.48 | 3.22 |
| 8 | 25 | 100 | 0 | 5,6 | 25 | 25 | | 0 | 8.33 | *8.33* | 18.52 | 0.56 | 3.70 |
| 9 | 20 | 50 | 5 | 6 | 25 | | | 0 | 12.50 | *3.33* | 1.11 | 0.13 | 2.89 |
| 10 | 25 | 100 | 5 | 7 | 20 | | | 0 | 10.00 | *9.63* | 18.53 | 0.48 | 4.99 |
| 11 | 10 | 100 | 0 | 8,9,10 | 25 | 20 | 25 | 0 | 5.75 | *5.75* | 33.17 | 0.58 | 2.43 |
| 12 | 5 | 150 | 0 | 11 | 25 | | | 0 | 5.00 | *5.00* | 30.00 | 0.20 | 4.00 |

Initial Project Cost = 1000

Fast tracking cost = 173.80

0.00

| Average compression actually achieved (average from project duration simulations) | **13.71** | (Deterministic) schedule compression | **18.88** |
|---|---|---|---|
| | 13.71% | $\min(f_2+f_5, f_3+f_6) + f_8 + f_{11} + f_{12}$ | 18.88% |

**Figure 4:** Case study and solution for maximum schedule compression.

Strictly speaking, it should not be necessary to set a maximum overlap ($o_{max}$) for each activity, as in theory, any optimisation algorithm should be able to identify the optimum overlap without this information. However, as a general rule, optimisation algorithms work better with narrower variable boundaries. Therefore, instead of simply limiting $o_{max}$ to the minimum of $d_p$ and $d_s$, $o_{max}$ was taken to be:

- For activities with a single predecessor: the minimum of $d_p/2$ and $d_s$ (see expression (5)).

- For activities with multiple predecessors (i.e. 7, 8 and 11): the value of $o$ that maximised $f$ (last column in the table) for values of $o$ between 0 and the minimum $d_p$, where $o \leq d_s$.

Using previous methods which model the amount of rework for each activity overlapped as an additional duration extension to that activity, this problem would have been very easy to solve. The procedure would have been as follows: overlap all the critical activities by their respective $o_{max}$ values and then overlap the activities which have become critical along the process so that the overlaps of those now critical paths are equal to the overlap of those initially critical activities minus the total rework. However, this is not actually the optimum solution when each activity has two possible outcomes, namely: successful overlap (where the probability of remaining overlapped is 1 minus the probability stated in the column "Overlap *Risk*"), or unsuccessful overlap where the activity returns to its original position (where the probability equals the "Overlap *Risk*").

For the stochastic case (the solution of which is in the column "Activity overlaps ($o$) (Solution)"), the maximum compression is achieved by compressing the critical activities to the maximum (i.e. $o=o_{max}$), whereas the initially non-critical activities that have later become critical are compressed by a minimum to allow the originally critical activities to become critical again. This means overlapping the emerging critical paths to a point at which their overall duration equals the current duration of the original critical path after having been compressed to its maximum extent. Let us illustrate this with our case study example. For

maximising schedule compression, all the critical activity overlaps are initially set to $o_{max}$. Then:

- Activity 4 has a slack of 5 days. This activity should not block the successful overlap of activity 3 which has also been overlapped by exactly 5 days. Therefore no compression is necessary in this case, and the cost due to overlap remains at zero.

- Activity 9 has a slack of 5 days, but in this case, activity 8 has been overlapped by 8.33 days. According to previous models, activity 8 has "effectively" been overlapped by 3.70 days (see the final column of the table). Therefore, in a deterministic case, activity 9 would not need to be overlapped as it would still have a remaining slack of 1.30 days. However, this is incorrect and activity 9 in fact needs to be overlapped by 3.33 days with its predecessor activity 6 as that is the only way it will not block activity 8's successful overlap. But it should not be overlapped any more, or the probability of having an unsuccessful overlap (bounce back) with activity 6 will increase and the average project duration will again be higher.

- Activities 7 and 10 also have a slack of 5 days, which is shared between the two. Their overlaps need to equal the overlaps of activities 6 and 8 minus the slack they had initially, i.e. they need a total overlap of 12.50+8.33-5=15.83. But how do we split this between the two activities? The aim is to find a pair of overlaps for activities 7 and 10 with the same risk value. This can be done either mathematically or numerically. The results shown in the table correspond to a *Risk* of 0.48 as can be seen in the last but one column. This optimum pair of overlaps ($o_7$=6.20 and $o_{10}$=9.63) minimises the overall risk of overlap failure, which ensures the critical path is compressed to its maximum possible value.

In order to obtain an average schedule compression, the set of overlap values shown in the column "Solution" were simulated 10,000 times. In these simulations, the activities were

sometimes successfully overlapped and sometimes not. The average compression for the total schedule from the project simulations was 13.71% as shown at the bottom centre in Figure 4.

However, if we had computed the effective overlaps for the critical path activities using a deterministic approach (shown at the bottom right corner), the results would have indicated that the actual compression achievable would have been 18.88%. The difference is quite significant (around 30%), even for such a simple project as this.

This problem was also solved by means of a Genetic Algorithm similar to the most recent GAs implemented in previous research (e.g. [20,21,51]). GAs are usually a good option for non-smooth optimisation problems, where they usually reach good (though frequently not optimal) solutions. Several mutation and crossover rates were tested and each time the objective function (the average project compression) was evaluated, 10,000 schedule simulations were carried out and the results averaged. The best solution identified allowed for a schedule compression of up to 14.67% (less than a 1% difference from the manually-obtained solution of 13.71%), but still underperforming the manual solution.

It is clear from all the models that the maximum schedule compression of 25% is not possible, which is mostly due to the stochastic variability of the activity overlaps. Additional reasons why the 25% was not achieved will be presented in the *Discussion*.

Finally, it is worth noting that the model could easily have been adapted to find the optimum set of overlaps to minimise the overall project cost for the case where, for example, there is a pay reward for each day the schedule is shortened. However, in this case, a manual approach would no longer have been possible, and that was not the aim of this paper.


## 4. Discussion

Project fast-tracking is increasingly being applied in the construction industry, as there is an increased demand for the expedient delivery of construction projects [53]. Common

examples of fast-tracking involve overlapping the traditionally sequential design and construction phases. More acute implementations involve overlapping construction activities and other work packages too.

Most construction works (e.g. buildings, bridges, highways, power plants, dams) include many activities. These can be considered unique endeavours and, most of the time, require an intensive coordination between large numbers of activities and resources. Unsuccessful coordination will eventually lead to project cost and/or time overruns. This is another reason why construction projects with short lead times have resorted to fast-tracking and concurrent engineering practices [56].

However, it has been established above that it is unlikely that fast-tracking will achieve a project compression beyond 25% of the original project duration. This conclusion is based on the assumption that *Risk* increases linearly ($\alpha$=1 in expressions (2), and (6) to (10)) with the overlap of a successor and its predecessor. This seems a reasonable argument for most construction projects as was explained before introducing expression (2) and seems to be aligned with empirical results (e.g. [52,53]). However, if the risk increased by a power other than unity, the maximum compression achievable would have been different to 25%. However, it is clear from expression (5), where *f* is mostly governed by $\alpha$, that *f* is not particularly sensitive to small variations in $\alpha$ (25% above or below).

It is also worth highlighting that an approximate maximum compression of 25% might be achievable in a situation where a project has activities with similar durations (less than a 50% difference in duration between successors and their respective predecessors). When this is not the case, for example when there are interspersed activities which are significantly shorter than others, each activity can only be overlapped with its predecessor(s) by a maximum of its own duration and the successor can only overlap with a shorter activity by half of the duration of the shorter activity. In other words, when there are some critical activities which are

significantly shorter than other critical activities, schedule compression is drastically reduced. There are, however, other reasons why a compression of 25% is difficult to achieve:

- The first activity in a schedule cannot be fast-tracked since it does not have a predecessor, hence its duration cannot be compressed (unless it is crashed; despite this is a complementary but totally different compression technique). However, this restriction is only relevant to projects with a small number of activities and/or where the first activity is significantly longer than the rest of activities.

- When activity durations are uncertain and/or variable, it is difficult to anticipate in advance to what degree two activities can be overlapped. As a consequence, if they are overlapped too much there is a greater risk of rework (and thus schedule delay), and if they are insufficiently overlapped, an optimum compression of 25% will not be achieved.

- The nodes at which several activities converge generally limit the amount of overlap, as seen in expression (3); again this reduces overall compression to below 25%.

- Finally, as it has been shown in the case study, the higher the number of critical and non-critical paths, the lower the maximum average overall compression which can be achieved.

With all of these limitations, it is to be expected that in real project settings, maximum schedule compressions will generally remain between 10% and 20% as some field studies [52,53] seemed to confirm. However, the upside is that with these lower rates of compression, the additional costs due to fast-tracking ($\Delta c$) will also be limited (in the case study, they remained at 17%).

## 5. Conclusions

This paper presents a new model of project fast-tracking. The way this model is stated has several advantages, namely it can be used for explicitly estimating the costs due to activity overlaps and it reduces the subjectivity of the input information compared to previous models.

Also, the proposed model includes previous models as special cases, as employs similar constructs (but with different parametrisation) for variable attributes such as activity sensitivity, evolution, dependency, etc. An additional advantage is that the multi-predecessor case can be easily accommodated by the model for any number of predecessor activities, and the cascade effect can also be modelled as an instance of the multi-predecessor case.

The findings of the analysis indicate that on average, it is very difficult for any schedule to be fast-tracked with a shortening in excess of 25% of the original project duration. A number of factors that might prevent this maximum compression threshold from being achieved have also been discussed.

The activity-to-activity overlap cost has been analysed for the case where the risk grows linearly with the overlap. This case has been extrapolated to an average schedule and it was shown that the maximum average costs arising from fast-tracking will not generally exceed 25% of the overall cost of the critical activities.

A case study of fast-tracking a project schedule was examined through manual calculations using the mathematical expressions developed and Genetic Algorithms. This case study has highlighted that unless Monte Carlo simulations are employed, the actual average schedule compression due to fast-tracking will be significantly overestimated. This assumption has been a common simplification in almost all previous models, as without exception, they have treated the effects of activity overlapping as a duration increase due to rework.

The model presented still has some limitations. For example, the model does not simultaneously handle activity crashing and fast-tracking, which could be advantageous for practical purposes. Also, real-life construction projects include far more activities and logical precedence relationship types (other than finish to start, FS) than the presented simplified project network. Other logical (non-FS) relationships can be easily converted to FS, as justified earlier, but a higher number of activities will remain a challenge for future algorithms, as the

complexity of the optimisation problem increases, not only with the number of activities, but also with the number of precedence relationships among them.

Hence, for future real project implementations, researchers will still need to develop refined optimisation algorithms that outperform the Genetic Algorithm proposed here. However, the mathematical expressions behind the evaluation of the objective function – the real contribution of this paper – will remain approximately the same. Most of these expressions have been made explicit and this will allow future algorithms to evaluate more efficiently what many previous models have had to evaluate iteratively. Finally, future implementations will need to be generalised for accommodating resource-constrained schedules, which, also due to their higher complexity, have not been considered here.

## Acknowledgements

## References

[1]     J.E.V. Gerk, 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.

[2]     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.

[3]     A. Van der Werf, Risk Mitigation in Fast Tracking Projects, in: AACE International. Transactions of the Annual Meeting, 2003. http://www.costandvalue.org/download/?id=1908 (accessed January 16, 2017).

[4] D.I. Ruby, Fast-tracking plant projects, Plant Engineering (Barrington, Illinois). 32 (1978) 121–123. (Magazine) ISSN 0032-082X.

[5] A. Baker, K. Boyd, Fast-tracking for nuclear power plant construction, International Journal of Project Management. 1 (1983). doi:10.1016/0263-7863(83)90019-4.

[6] M. Zaver, An integrated approach to project management, Wainwright West project, JPT, Journal of Petroleum Technology. 50 (1998) 1–2. doi:10.2118/49064-MS.

[7] V. Krishnan, Managing the simultaneous execution of coupled phases in concurrent product development, IEEE Transactions on Engineering Management. 43 (1996) 210–217. doi:10.1109/17.509986.

[8] M.V. Murray, Fast tracking for the future, Pipes and Pipelines International. 46 (2001). (Magazine) ISSN 0032-020X.

[9] S.K. Miller, R. Ortega, Managing fast track design and construction projects - How to stay on schedule and within budget, in: Proceedings of the 2006 Pipeline Division Specialty Conference - Pipelines 2006: Service to the Owner, 2006. doi:10.1061/40854(211)34.

[10] M. Whatley, S. Golden, T. Barletta, Fast-tracking an FCC revamp, Petroleum Technology Quarterly. 11 (2006). http://www.revamps.com/media/files/Article/189050ea/222.pdf (access. Jan.16, 2017).

[11] M.L. García-Vargas, J. Caballero, A. Pérez-Calpena, P. Amado, W. Seifert, M. Azzaro, H. Mandel, A. Quirrenbach, I. Ribas, A. Reiners, E. Guenther, L. Gesa, D. Galadí, J. Aceituno, CARMENES: Management of a schedule-driven project, in: Proceedings of SPIE - The International Society for Optical Engineering, 2016. doi:10.1117/12.2232433.

[12] F. Khoramshahi, J.Y. Ruwanpura, Speed up of project delivery through application of effective fast tracking strategies in the engineering phase, in: ISEC 2011 - 6th International Structural Engineering and Construction Conference: Modern Methods and Advances in Structural Engineering and Construction, 2011. doi:10.3850/978-981-08-7920-4-S1-C10-cd.

[13] M.B. Schoenhardt, V.C. Pardais, M.R. Marino, Why projects fail (and what we can do about it), in: Proceedings of the Biennial International Pipeline Conference, IPC, 2014. doi:10.1115/IPC2014-33515.

[14] Z. Tang, B. McCabe, Project coordination for fast-tracking construction projects, in: Proceedings, Annual Conference - Canadian Society for Civil Engineering, 2003. ISBN: 9781894662048.

[15] C. Terwiesch, C.H. Loch, A. De Meyer, Exchanging Preliminary Information in Concurrent Engineering, Organization Science. 13 (2002) 402–419. doi:10.1287/orsc.13.4.402.2948.

[16] B. Prasad, Concurrent Engineering Fundamentals: Integrated product and process organization, Prentice Hall, Upper saddle river, N.J., 1996. doi:10.13140/RG.2.1.2613.0005.

[17] F. Peña-Mora, M. Park, Dynamic Planning for Fast-Tracking Building Construction Projects, Journal of Construction Engineering and Management. 127 (2001) 445–456. doi:10.1061/(ASCE)0733-9364(2001)127:6(445).

[18]    A. Delgoshaei, M.K.M. Ariffin, B.T.H. Tuah Baharudin, Pre-emptive resource-constrained multimode project scheduling using genetic algorithm: A dynamic forward approach, Journal of Industrial Engineering and Management. 9 (2016) 732. doi:10.3926/jiem.1522.

[19]    R. Dehghan, K. Hazini, J.Y. Ruwanpura, Optimum activity overlapping using genetic algorithms, in: ISEC 2011 - 6th International Structural Engineering and Construction Conference: Modern Methods and Advances in Structural Engineering and Construction, 2011. doi:10.3850/978-981-08-7920-4-S1-CP19-cd.

[20]    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.

[21]    A.K. Cho, M. Hastak, Time and cost-optimized decision support model for fast-track projects, Journal of Construction Engineering and Management. 139 (2013) 90–101. doi:10.1061/(ASCE)CO.1943-7862.0000570.

[22]    T.A. Roemer, R. Ahmadi, Concurrent Crashing and Overlapping in Product Development, Operations Research. 52 (2004) 606–622. http://pubsonline.informs.org/doi/abs/10.1287/opre.1040.0125 (accessed January 16, 2017).

[23]    T.A. Roemer, R. Ahmadi, R.H. Wang, Time-Cost Trade-Offs in Overlapped Product Development, Operations Research. 48 (2000) 858–865. http://pubsonline.informs.org/doi/abs/10.1287/opre.48.6.858.12396?journalCode=opre (accessed January 16, 2017).

[24]    M. Vanhoucke, D. Debels, The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects, Computers and Industrial Engineering. 54 (2008) 140–154. doi:10.1016/j.cie.2007.07.001.

[25]    M. Vanhoucke, Setup times and fast tracking in resource-constrained project scheduling, Computers and Industrial Engineering. 54 (2008) 1062–1070. doi:10.1016/j.cie.2007.11.008.

[26]    Y. Khoueiry, I. Srour,  a Yassine, An optimization-based model for maximizing the benefits of fast-track construction activities, Journal of the Operational Research Society. 64 (2013) 1137–1146. doi:10.1057/jors.2013.30.

[27]    R. Dehghan, J.Y. Ruwnapura, Model of trade-off between overlapping and rework of design activities, Journal of Construction Engineering and Management. 140 (2014) 1–13. doi:10.1061/(ASCE)CO.1943-7862.0000786.

[28]    V. Krishnan, S. Eppinger, D. Whitney, A Model Based Framework to Overlap Product Development Activities, Management Science. 43 (1997) 437–451. http://pubsonline.informs.org/doi/abs/10.1287/mnsc.43.4.437 (accessed Jan. 16, 2017).

[29]    S. Nicoletti, F. Nicolò, A concurrent engineering decision model: Management of the project activities information flows, International Journal of Production Economics. 54 (1998) 115–127. doi:10.1016/S0925-5273(97)00108-4.

[30]    F. Peña-Mora, M. Li, Dynamic Planning and Control Methodology for Design/Build Fast-Track Construction Projects, Journal of Construction Engineering and Management. 127 (2001) 1–17. doi:10.1061/(ASCE)0733-9364(2001)127:1(1).

[31]    S. Bogus, K. Molenaar, J. Diekmann, Concurrent engineering approach to reducing design delivery time, Journal of Construction Engineering and Management. 131 (2005) 1179–1185. doi:10.1061/ASCE0733-93642005131:111179.

[32]    S.M. Bogus, K.R. Molenaar, J.E. Diekmann, Strategies for overlapping dependent design activities, Construction Management and Economics. 24 (2006) 829–837. doi:10.1080/01446190600658529.

[33]    N. a. Blacud, S.M. Bogus, J.E. Diekmann, K.R. Molenaar, Sensitivity of Construction Activities under Design Uncertainty, Journal of Construction Engineering and Management. 135 (2009) 199–206. doi:10.1061/(ASCE)0733-9364(2009)135:3(199).

[34]    M.M. Ramadan, I.M. Srour, M.A. Abdul-Malak, A.A. Yassine, Qualifying and quantifying the exchange of information between pairs of design activities, in: ISEC 2011 - 6th International Structural Engineering and Construction Conference: Modern Methods and Advances in Structural Engineering and Construction, 2011. doi:10.3850/978-981-08-7920-4-S1-C02-cd.

[35]    S. Bogus, J. Diekmann, K. Molenaar, C. Harper, S. Patil, J. Lee, Simulation of overlapping design activities in concurrent engineering, ASCE Journal of Construction Engineering & Management. 137 (2011) 950–957. doi:10.1061/(ASCE)CO .1943-7862.0000363.

[36]    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.

[37]    Y. Khoueiry, I. Srour, A. Yassine, An Optimization Model for Maximizing the Benefits of Fast-Tracking Construction Projects, in: Construction Research Congress ASCE, 2012: pp. 247–257. http://ascelibrary.org/doi/abs/10.1061/9780784412329.026 (accessed January 17, 2017).

[38]    R. Dehghan, J.Y. Ruwanpura, F. Khoramshahi, Activity overlapping assessment in construction, oil and gas projects, in: Construction Research Congress 2010: Innovation for Reshaping Construction Practice - Proceedings of the 2010 Construction Research Congress, 2010. doi:10.1061/41109(373)118.

[39]    H.-S. Gwak, S.-H. Son, Y.-J. Park, D.-E. Lee, Exact Time–Cost Tradeoff Analysis in Concurrency-Based Scheduling, Journal of Construction Engineering and Management. 142 (2016) 4016054. doi:10.1061/(ASCE)CO.1943-7862.0001164.

[40]    R.B. Austin, P. Pishdad-Bozorgi, J.M. de la Garza, Identifying and Prioritizing Best Practices to Achieve Flash Track Projects, ASCE Journal of Construction Engineering & Management. 142 (2016) 1–12. doi:10.1061/(ASCE)CO.1943-7862.0001061, 04015077.

[41]    P. Pishdad-Bozorgi, J.M. de la Garza, R.B. Austin, Readiness Assessment for Flash Tracking, ASCE, Journal of Construction Engineering & Management. 142 (2016) 1–5. doi:10.1061/(ASCE)CO.1943-7862.0001190, 06016005.

[42]    A.A. Alhomadi, R. Dehghan, J.Y. Ruwanpura, The predictability of fast-track projects, Procedia Engineering. 14 (2011) 1966–1972. doi:10.1016/j.proeng.2011.07.247.

[43]  A.A. Alhomadi, R. Dehghan, J.Y. Ruwanpura, Time, cost and quality predictability in fast-track projects, in: 4th Construction Specialty Conference, Montréal, Québec, 2013: p. CON-203-1 to CON-203-9. doi:10.13140/RG.2.1.1162.9288.

[44]  M. Vanhoucke, Project Management with Dynamic Scheduling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-25175-7.

[45]  K.R. MacCrimmon, C.A. Ryavec, An Analytical Study of the PERT Assumptions, Operations Research. 12 (1964) 16–37. doi:10.1287/opre.12.1.16.

[46]  D. Gong, R. Hugsted, Time-uncertainty analysis in project networks with a new merge-event time-estimation technique, International Journal of Project Management. 11 (1993) 165–173. doi:10.1016/0263-7863(93)90049-S.

[47]  P. Pontrandolfo, Project duration in stochastic networks by the PERT-path technique, International Journal of Project Management. 18 (2000) 215–222. doi:10.1016/S0263-7863(99)00015-0.

[48]  C.H. Loch, C. Terwiesch, Communication and Uncertainty in Concurrent Engineering, Management Science. 44 (1998) 1032–1048. doi:10.1287/mnsc.44.8.1032.

[49]  L. Grèze, R. Pellerin, P. Leclaire, N. Perrier, CIGI2011: A heuristic method for resource-constrained project scheduling with activity overlapping, Journal of Intelligent Manufacturing. 25 (2014) 797–811. doi:10.1007/s10845-012-0719-5.

[50]  T.K. Lim, C.Y. Yi, D.E. Lee, D. Arditi, Concurrent construction scheduling simulation algorithm, Computer-Aided Civil and Infrastructure Engineering. 29 (2014) 449–463. doi:10.1111/mice.12073.

[51]  M.A. Hossain, D.K.H. Chua, Overlapping design and construction activities and an optimization approach to minimize rework, International Journal of Project Management. 32 (2014) 983–994. doi:10.1016/j.ijproman.2013.10.019.

[52]  N.N. Eldin, Concurrent engineering: a schedule reduction tool, Journal of Construction Engineering and Management. 123 (1997) 354–362. doi:10.1061/(ASCE)0733-9364(1997)123:3(354).

[53]  A. Attar, M.A. Boudjakdji, N. Bhuiyan, K. Grine, S. Kenai, A. Aoubed, Integrating numerical tools in underground construction process, Engineering, Construction and Architectural Management. 16 (2009) 376–391. doi:10.1108/09699980910970851.

[54]  M. Lu, H.-C. Lam, Transform Schemes Applied on Non-Finish-to-Start Logical Relationships in Project Network Diagrams, Journal of Construction Engineering and Management. 135 (2009) 863–873. doi:10.1061/(ASCE)CO.1943-7862.0000062.

[55]  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.

[56]  C.J. Anumba, A.N. Baldwin, D. Bouchlaghem, B. Prasad, A.F. Cutting-Decelle, J. Dufau, M. Mommessin, Integrating Concurrent Engineering Concepts in a Steelwork Construction Project, Concurrent Engineering. 8 (2000) 199–212. doi:10.1177/1063293X0000800304.