*Original article*

# CONSTRUCTION PROJECT CRASHING WITH UNCERTAIN CORRELATED NORMAL AND CRASH TASK DURATIONS AND COSTS: AN INTEGRATED STOCHASTIC PRACTICAL APPROACH

Dimitrios D. Kantianis

*Department of Economic and Regional Development, Panteion University of Social and Political Sciences, Athens, Greece*

*Received: 9 January 2023*
*Revised: 6 March 2023*
*Accepted: 27 March 2023*

**Abstract:** The research aims at developing an integrated mathematical spreadsheet modelling approach for the practical solution of the stochastic crashing problem in construction project planning. The proposed project crashing methodology is founded upon a synthesis of traditional PERT/CPM network scheduling, Monte Carlo simulation, and linear programming. The main contribution of the introduced model to the existing project management literature is that it produces frequency histograms and relevant statistics for optimum project crash make span and additional cost for project compression, by assuming uncertainty and correlation simultaneously for both normal and crash durations and direct expenses of project activities. The implementation of the model is automated in Microsoft Excel© through VBA coding. The research is anticipated to assist built environment academics and professionals to improve decision-making effectiveness in the planning of construction projects.

**Keywords:** Construction; Crashing; Optimisation; Planning; Uncertainty.

## 1. INTRODUCTION

The 'project crashing' problem in construction occurs when the project duration is not compliant with the scheduled baseline or when the owner wishes to accelerate the process for using the constructed facility earlier than the agreed contract deadline. In such situations, there is a need for compressing the project execution by narrowing the finish times of several critical activities to less than normal estimated times but increasing concurrently the direct costs due to the additional resources required (Figure 1). Such an acceleration may include the use of more efficient equipment, hiring more workers or subcontracting selected parts of the work. Therefore, it is important to model the trade-off relationship between time saved and additional cost for crashing to identify which tasks to crash and to assess the associated expenses required to compress these activities (Ahipasaoglu et al., 2019). Project compression in construction scheduling has been recognised since the 1950s, simultaneously with the development of the critical path method (CPM) by Kelley and Walker (1959) and the program evaluation and review technique (PERT) reported in Malcolm et al. (1959). Since then, a vast amount of literature has proposed various solutions to the basic deterministic project crashing problem. However, the literature addressing the *stochastic* version of the problem is very limited (Herroelen & Leus, 2005).



Corresponding author. Email: dkant@panteion.gr
ISSN 2560-4961 (online)

https://doi.org/10.56889/pdsd6032

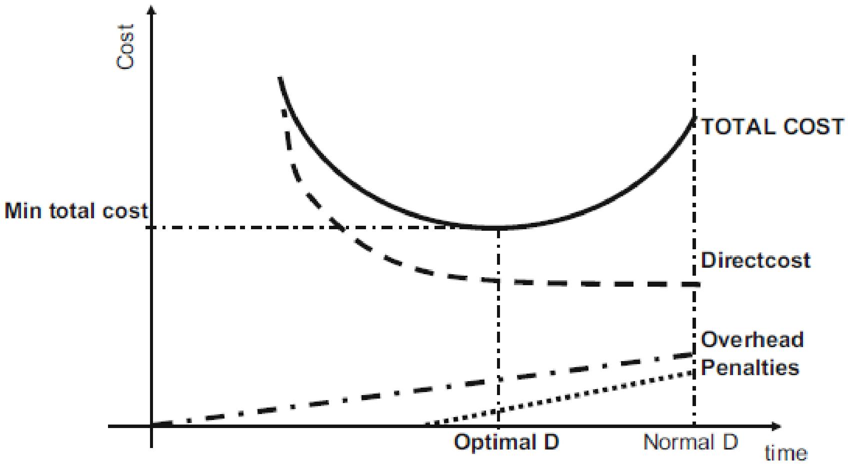



**Figure 1:** A typical project crashing graph (Source: De Marco, 2011)

Project managers widely acknowledge that the inherent uncertainty, arising from both internal and external diverse sources, including technical, managerial, or commercial issues, is critical for delivering successful construction projects (Hillson, 2002). Moreover, as new endeavours, projects require implementation of previously untried designs and physical production processes while accomplishing demanding restrictions within usually tight time and cost boundaries. According to Harrison and Lock (2017), construction is prone to damage through uncertainties and risks and, it is not surprising, that many projects fail by wide margins to meet their targets. Hulett (2016) pointed out that schedule activity durations are better understood as probabilistic estimates of possible durations rather than single-point figures about how long the activity will last. Thus, typical deterministic models for construction planning with fixed figures suffer from the assumption of absolute certainty and disregard random events that normally arise during project design and execution. Consequently, in practice, expected project duration and cost are frequently underestimated (Möhring, 2001). In conventional CPM, each activity in the project network is assigned a single duration value which represents the 'best guess' estimate of the expected time required to complete the activity (Willis, 1985). As a result, project duration is also calculated as a single figure by summing the deterministic time estimates of the critical activities. However, in construction, real-life durations are most often *not* known in advance with certainty. Thus, more than 60 years ago, the classic PERT technique introduced the estimation of potential variability in the expected project makespan (Malcolm et al., 1959). PERT can be used either for estimating the probability that a project will be completed by a specific deadline or to construct a selected confidence interval for upper and lower project makespan values. The methodology may also be used for cost estimation. The following formula is used to find the expected mean duration $\bar{x}$ of an activity *i*:

$$\bar{x}_i = (a_i + 4 \cdot m_i + b_i)/6 \qquad (1)$$

where: a-*optimistic*, m-*most likely*, and b-*pessimistic* are the three duration estimates to complete the activity under three distinct working conditions (Figure 2).

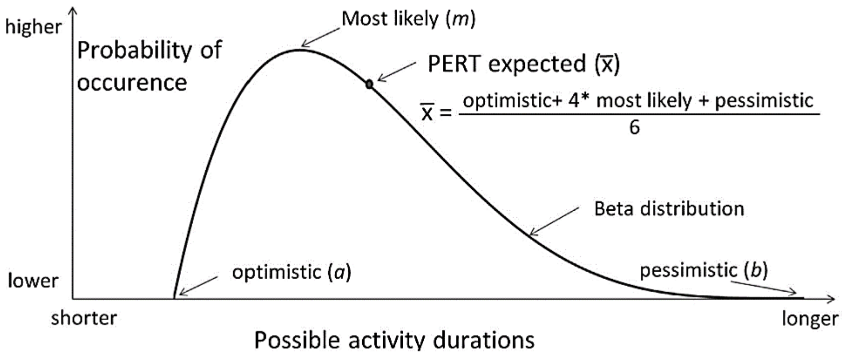



**Figure 2:** Classic PERT beta distribution for possible durations (Source: Hajdu & Isaac, 2016)

The estimation of the *variance* of activity *i* which describes the uncertainty associated with its duration, assumes six (6) *standard deviations* between optimistic and pessimistic times:

$$\sigma_i^2 = [(b_i - a_i)/6]^2 \qquad (2)$$

PERT analysis is critically explained in several project management textbooks (see Klastorin, 2004) and Tables for z-values for areas under the standard normal curve (probabilities) are included in most standard statistics textbooks (e.g., Montgomery & Runger, 2014). Roos and den Hertog (2020) stated that PERT is still widely used in practice (see for instance, Onifade et al., 2017) and the technique is included in most project management software packages because of its low computational effort; it only requires three-point estimation of the likely values mainly from historical data or/and subjective expert judgment by built environment professionals.

PERT/CPM planning approaches were subsequently extended to Monte Carlo simulation (MCS) method which permits the analysis of the distribution of the critical path without the restricted PERT assumptions (Van Slyke, 1963). MCS also facilitates a clearer understanding of effective construction cost planning. Therefore, assigning probabilistic estimates is not limited to project activity durations, but probability distribution functions can also be attributed to project direct expenses. Expert judgment is often producing these estimates to arrive at a frequency distribution for the final total project cost. This cost distribution is then used by management to put aside a budget reserve, to be used when contingency plans are necessary to respond to uncertain events (Kwak & Ingall, 2007). MCS is used in the research as the stochastic attribute to the developed project crashing model. However, a common source of error in MCS is the assumption of independency between random variables, so that changes in one variable does not affect other variables. According to Touran and Wiser (1992), this supposition may result in inaccurate estimates in actual construction projects. This research assumes the existence of a positive strong correlation not only for uncertain normal durations and direct costs of activities but also between uncertain crash times and additional costs for project crashing.

## 2. LITERATURE REVIEW ON THE STOHASTIC PROJECT CRASHING PROBLEM

The review of relevant literature focuses on the stochastic project crashing problem beyond traditional PERT/CPM techniques. Coskun (1984) developed an approach to the stochastic crashing problem based on chance constrained programming by converting the probabilistic problem to an equivalent deterministic formulation with normally distributed activity times having known mean and standard deviation. Wollmer (1985) introduced a stochastic version of the traditional deterministic linear compression definition, using a limited set of crashing options with task durations following discrete distributions. Dodin (1985) proposed a procedure to find an approximate cumulative density function (cdf) for project makespan with stochastic task durations. Johnson and Schou (1990) defined activity times by continuous random variables and suggested the use of three rules when selecting tasks to compress (the common CPM lowest cost slope; the highest criticality index;

and the least cost expected value). The authors, based on a simulation study of a single small sample project, concluded that as the size of problems increases, the likelihood of multiple critical paths would likely lead to larger differentials in the expected cost of different rules. Gutjahr et al. (2000) described a stochastic branch-and-bound algorithm for solving a discrete (binary) version of stochastic crashing whereas activity times must be either normal or crashed. Feng et al. (2000) presented a novel hybrid approach by combining stochastic simulation with genetic algorithms and also highlighted the need to develop more efficient algorithms.

Thus, selected subsequent attempts to analyse project crashing problems under uncertainty are summarised as follows. Mitchell and Klastorin (2007) formulated the objective function with direct costs, indirect costs, and penalty costs, presenting a stochastic compression project heuristic based on decomposition of PERT networks into serial and parallel subnets. Eshtehardian et al. (2008) established a multiobjective fuzzy time-cost model. Aghaie and Mokhtari (2009) considered a new hybrid approach for the stochastic project crashing problem based on ant colony optimization technique and MCS by using discrete and exponentially distributed functions for activity costs and durations, respectively. Ke et al. (2009) constructed two models for stochastic compression with both chance-constrained programming and dependent-chance programming. Mokhtari et al. (2010) developed a hybrid optimization approach based on cutting plane method and MCS for stochastic crashing of PERT networks. Ke (2014) presented an uncertain random time-cost trade-off model with dependent chance programming that was built with a crisp equivalent model for the case that uncertain random parameters in the problem are partly random and partly uncertain variables. The proposed model was solved through the integration of uncertain random simulation and genetic algorithms. Kang and Choi (2015) considered a stochastic time-cost trade-off problem to determine the required level of crashing activities so that the expected summation of crashing and tardiness costs is minimized; they proposed a threshold policy that makes crashing decisions contingent on projects' current status, i.e., crashing an activity to compensate delayed starting time from a predetermined threshold. Yang and Morton (2022) recently proposed a branch-and-cut decomposition algorithm in which spatial branching of the first stage continuous variables and linear programming approximations for the recourse problem are sequentially tightened. The algorithm was tested by the authors with multiple improvements and showed that the solution time can be significantly reduced over the direct solution of the problem.

It could be argued that the construction project crashing problem is rather limited in literature, but it still is an active field of research with practical implications to all stakeholders engaged in projects (clients, project managers, and contractors). Therefore, the purpose of the paper is to develop an integrated stochastic crashing method that can be practical and easy to implement for construction professionals. The suggested mathematical model which assumes that normal as well as crash task durations and costs are uncertain and strongly correlated, is founded upon classic PERT/CPM network techniques and MCS. Transparency in the introduced approach is guaranteed by the model being setup in an Excel spreadsheet. Finally, it is automated through (three discrete) VBA (Visual Basic for Applications) codes that can also be combined together in a single ('one-click') code.

## 3. STOHASTIC PROJECT CRASHING INTEGRATED MODEL DEVELOPMENT

The rationale behind the research methodology with its associated mathematical formulation which is used to develop the new integrated stochastic crashing model is explained as follows:

### 3.1 Stochastic project network definition

Any stochastic project analysis requires a realistic and consistent deterministic work schedule, created with the use of information from historical projects and best practices for the network logic and the appropriateness of the sequencing and phasing of activities (Mubarak, 2015). Nowadays, project scheduling practice is implemented almost entirely on AoN networks; the technique is more flexible because of its simplicity (Vanhoucke, 2013) and its enhanced modelling capabilities closer to reality (Hajdu, 2013), and it is the type of network analysis used in this research.

A construction project is defined as an acyclic and directed graph G = (N, P) consisting of a set of interacting activities (or tasks) to be executed with no interruptions with required ***uncertain*** durations and direct expenses for their completion. Each activity normally requires resources. Resources may be of different types, including financial resources, human resources, machinery, equipment, materials, energy, etc. The work breakdown structure of the project provides a decomposition of these activities into i). a set of nodes (vertices) N which consists of $n$ activities $i = \{1, \ldots, n\}$ to be scheduled *plus* two auxiliary (dummy, with zero duration and cost) activities 0 and $n$+1 representing project start and finish, respectively, and ii). a set of arcs (edges) P representing the technological precedence relationships (constraints) between activities $i$. A precedence relationship is defined as a pair of activities ($a$, $b$) where $a \neq b$, denoting that beginning time of activity $a$ affects earliest start time of activity $b$. A normal execution duration $\boldsymbol{x}_a$ is assigned as a ***random variable*** (***r.v.***) (with bold style henceforward) to each activity $a$ and a time lag $\delta_{ab}$ to each pair $(a, b)\in$P. The temporal constraint then is $\delta_{ab} \leq s_b - s_a$ with $s_a$ and $s_b$ being the start times of activities $a$ and $b$, respectively. Since $(a, b)\in$P, activity $b$ cannot start earlier than $\delta_{ab}$ time units (normally working weeks or days) after the start of activity $a$. If $\delta_{ab} = x_a$, the above inequality constraint is referred to as immediate precedence constraint between activities $a$ and $b$ (Schwindt, 2006). AoN network analysis then consists of (Oxley & Poskitt, 1996): (1) calculating the earliest finish ($EF_i$) times of activities $i$ by a forward pass through the network and selecting the longest path (i.e. the final earliest completion time gives project duration); (2) calculating the latest finish ($LF_i$) times of activities $i$ by a backward pass through the network and selecting the longest path (the final latest finish time is the same as its earliest completion time and gives the same project duration); (3) calculating the total float ($TF_i$) of activities $i$ which is either latest start times *minus* earliest start times ($LS_i - ES_i$) or latest finish times *minus* earliest finish times ($LF_i - EF_i$) (since both give the same results); and (4) identifying the critical activities, i.e. the ones with zero total float, to determine the critical path and the project duration under normal execution condition $T_n$. The complete project network definition is as follows:

G — an acyclic and directed graph, where G = (N, P)

N — set of nodes in project network, each node representing an activity $i$

P — set of arcs in project network, representing the technological precedence relationships between activities $i$, with each activity pair $(a, b)\in$P where $a \neq b$, denoting that starting time of activity $a$ affects earliest start time of activity $b$

$i$ — activity to be executed with no interruption, where $i = \{0, 1, \ldots, n, n+1\}\in$N, with 0 and $n$+1 being the two auxiliary (dummy) activities representing project start and finish, respectively

$\boldsymbol{x}_i$ — ***r.v.*** – normal execution duration assigned to each activity $i$ ($\boldsymbol{x}_i \geq 0$), with $x_0 = x_{n+1} = 0$

$ES_i$ — earliest start time of activity $i$

$EF_i$ — earliest finish time of activity $i$

$LS_i$ — latest start time of activity $i$

$LF_i$ — latest finish time of activity $i$

$TF_i$ — total float (or slack) of activity $i$ where:

$$TF_i = LF_i - EF_i = LS_i - ES_i \quad (3)$$

$\delta$ — time lag to each arc $(a, b)\in$P, where:

$$\delta_{ab} + s_a \leq s_b \quad (4)$$

being the temporal constraint with $s_a$ and $s_b$ the start times of activities $a$ and $b$; if $(a, b)\in$P, activity $a$ cannot start earlier than $\delta_{ab}$ time units after the start of activity $a$; if $\delta_{ab} = \boldsymbol{x}_a$, constraint (4) is referred to as the immediate precedence constraint between activities $a$ and $b$ assuming a finish-to-start (FS) relationship with zero leads or lags (FS=0). This is the best-known type of precedence relationship (Demeulemeester & Herroelen, 2006) and it is used in this study. Thus, an activity can only start as soon as all its predecessor activities have finished.

$T_n$ — normal execution project duration

The next step in constructing the model

considers the project crashing (or project compression) problem setting.

### 3.2 Project crashing problem mathematical formulation

At first, the ***random*** estimates $\boldsymbol{y}_i$ for duration and $\boldsymbol{Y}_i$ for direct cost are assigned to each activity $i$ for crash completion. Another random estimate $\boldsymbol{X}_i$ is assigned to each task $i$ representing the direct cost for execution under normal condition. Assuming that cost is a linear function of time, as in the original CPM (Kelley, 1961), the marginal crashing cost for each task $i$ is calculated, which is the additional cost for shortening each activity by one time unit. Hence, if one reduces by one time unit the normal duration $\boldsymbol{x}_i$ of the critical task with the lowest marginal cost of crashing (or else with the minimum crash cost slope), it is possible to shorten the project duration by one time unit, at the expense of the crash direct cost of that critical task. The process is repeated until another network path(s) becomes critical and to the point where all critical activities can be fully compressed. Then, a maximum crashing point is determined where project duration corresponds to the minimum possible (compressed) project completion and cannot be crashed any further. The mathematical formulation of the project crashing problem is described as follows:

$\boldsymbol{y}_i$ — ***r.v.*** – crash execution duration assigned to each activity $i$ ($0 \leq \boldsymbol{y}_i \leq \boldsymbol{x}_i$), with $y_0 = y_{n+1} = 0$

$r_i^{max}$ — maximum time reduction in duration of activity $i$, where:

$$r_i^{max} = \boldsymbol{x}_i - \boldsymbol{y}_i \qquad (5)$$

$r_i$ — time reduction in duration of activity $i$ for crashing the project ($0 \leq r_i \leq r_i^{max}$)

$s_i$ — start time of activity $i$ when crashing the project ($s_i \geq 0$)

$e_i$ — end time of activity $i$ when crashing the project, where:

$$e_i = s_i + \boldsymbol{x}_i - \boldsymbol{y}_i \qquad (6)$$

$T_c$ — crash execution project duration, i.e., earliest possible completion after project crashing

$t$ — time units for project duration, where: $t = \{0, 1, 2, \ldots, T_c, \ldots, T_n\}$, with $0 < T_c < T_n$

$\boldsymbol{X}_i$ — ***r.v.*** – direct cost for normal completion ($\boldsymbol{x}_i$) of activity $i$

$\boldsymbol{Y}_i$ — ***r.v.*** – direct cost for crash completion ($\boldsymbol{y}_i$) of activity $i$

$C_x$ — total project direct cost for normal completion ($T_n$), where:

$$C_x = \sum \boldsymbol{X}_i \qquad (7)$$

$C_y$ — total project direct cost for crash completion ($T_c$), where:

$$C_y = \sum \boldsymbol{Y}_i \qquad (8)$$

$A_i$ — additional direct cost for crash completion of activity $i$, where:

$$A_i = \boldsymbol{Y}_i - \boldsymbol{X}_i \qquad (9)$$

$b_i$ — additional direct cost per time unit saved from crashing activity $i$, or crash cost slope, where:

$$b_i = (\boldsymbol{Y}_i - \boldsymbol{X}_i) / (\boldsymbol{y}_i - \boldsymbol{x}_i) = -(A_i / r_i^{max}) \qquad (10)$$

$C_i$ — crash direct cost for activity $I$, where:

$$C_i = b_i \cdot r_i \qquad (11)$$

$C_c$ — total project direct cost for crash completion, where:

$$C_c = \sum C_i = \sum (b_i \cdot r_i) \qquad (12)$$

$C_t$ — total project cost for crash completion, where:

$$C_t = C_x + C_c = \sum \boldsymbol{X}_i + \sum (b_i \cdot r_i) \qquad (13)$$

The linear programming (LP) mathematical model for solving the project crashing problem is:

*Objective function*

to minimize crash execution project duration $T_c$, where:

$$T_c = s_{n+1} + x_{n+1} - {}_{n+1} \qquad (14)$$

*Subject to* (*Constraints*)

$$r_i \leq r_i^{max} \text{ (maximum reduction in activity duration)} \qquad (15)$$

$r_i \geq 0$ (non-negativity for reduction in activity duration) (16)

$s_i \geq 0$ (non-negativity for activity start times) (17)

$T_c \geq 0$ (non-negativity for project duration) (18)

$T_c \leq T_n$ (maximum project duration constraint) (19)

$s_{i+1} \geq s_i + \boldsymbol{x}_i - r_i$ (general start time precedence constraint) (20)

The formulated LP model, i.e., the objective function (14) and constraints (15)-(20), can be put into a spreadsheet with a built-in optimisation tool, e.g., Microsoft Excel© (Excel) with Solver add-in (Solver) developed by Frontline Systems© (www.solver.com), which is the software selected in this research. The use of spreadsheets has become a matter of routine for construction estimators and managers by providing ease of implementation with flexible presentation facilities, adaptability to new information, and capability to incorporate uncertainty by generating uniform random numbers and simulating iterations from specific probability distributions (Cooper et al., 2005). Solver can handle efficiently both linear (through the Simplex algorithm) and non-linear LP optimisation problems and is used to solve the project crashing linear optimisation problem. The key to calculate the optimal solution to a LP problem with Solver is to set up a spreadsheet model that tracks every decision variable of interest to management (e.g., time and cost), to identify the changing cells, i.e., the variable cells, to set the target cell, i.e., the cell that contains the objective function, and to specify all required constraints (Moore & Weatherford, 2001).

### 3.3 Modelling correlation between random time and cost variables

The uncertain nature of construction production environment is embedded into the project crashing modelling process by assigning a normal probability distribution function to the four random estimates: normal duration $\boldsymbol{x}_i \sim N(\mu_x, \sigma_x)$, crash duration $\boldsymbol{y}_i \sim N(\mu_y, \sigma_y)$, normal direct cost $\boldsymbol{X}_i \sim N(\mu_X, \sigma_X)$, and crash direct cost $\boldsymbol{Y}_i \sim N(\mu_Y, \sigma_Y)$ of work activities. The normal distribution is chosen because it is widely used and intuitively simple (Yang, 2005), and basically can be described by only two parameters familiar to most construction managers, the mean and the standard deviation:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (21)$$

Its main shortcoming is that it is symmetrical by definition and cannot be skewed like, for instance, the triangular distribution or the beta distribution (Vose, 2008). To overcome this inflexibility, in this research, the mean and variance of the assigned normal distribution functions are derived from the previously presented traditional PERT equations (1) and (2). As a result, the sole initial input required by the model from construction experts is simply the classic PERT three-point estimates a, m, and b. The research further assumes a strong positive correlation for each of both *normal* ($\boldsymbol{x}_i$, $\boldsymbol{X}_i$) and *crash* ($\boldsymbol{y}_i$, $\boldsymbol{Y}_i$) random pairs of task durations and direct costs, with a Pearson's positive correlation coefficient $\rho$ with: $+0{,}50 < \rho < +1{,}00$ (Lind et al., 2017). The correlation coefficient $\rho$ is a value ranging from –1 to +1 and represents the desired degree of correlation between two variables during sampling. A positive relationship between the variables, i.e., when the value sampled for the one variable is high, the value sampled for the second variable will also tend to be high, are indicated by positive values. On the contrary, an inverse relationship between two variables is indicated by a negative correlation coefficient value, i.e., when the value sampled for the one variable is high, the value sampled for the second variable will tend to be low. Thus, when a randomly selected normal or crash duration is closer to either the pessimistic or the optimistic value, then, the corresponding random normal or crash cost will in turn be selected closer to either the pessimistic or the optimistic cost estimate. To accurately model this assumed correlation, four standard normal independent random variables, namely $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{Z}_3$, and $\mathbf{Z}_4$ corresponding to random variables $\boldsymbol{x}_i$, $\boldsymbol{X}_i$, $\boldsymbol{y}_i$, and $\boldsymbol{Y}_i$, respectively, are randomly generated for each activity from the inverse cumulative standard normal distribution N(0, 1) using NORMSINV Excel function. The standard normal distribution is derived from the above equation (21) for $\mu = 0$ and $\sigma = 1$:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (22)$$

If each random pair ($\boldsymbol{x}_i$, $\boldsymbol{X}_i$) and ($\boldsymbol{y}_i$, $\boldsymbol{Y}_i$) is also assumed to follow a *bivariate* normal joint probability distribution (Feng et al., 2000; Garvey et al., 2016) with parameters $\mu_x$, $\sigma_x$, $\mu_X$, $\sigma_X$, $\rho_{xX}$ and $\mu_y$, $\sigma_y$, $\mu_Y$, $\sigma_Y$, $\rho_{yY}$, respectively, then the following equations can be used to produce the required correlated random values:

$$\boldsymbol{x}_i = \sigma_x \cdot \mathbf{Z}_1 + \mu_x \quad (23)$$

$$\boldsymbol{X}_i = \sigma_X \cdot [\rho_{xX} \cdot \mathbf{Z}_1 + (1 - \rho^2_{xX})^{1/2} \cdot \mathbf{Z}_2] + \mu_X \quad (24)$$

$$\boldsymbol{y}_i = \sigma_y \cdot \mathbf{Z}_3 + \mu_y \quad (25)$$

$$\boldsymbol{Y}_i = \sigma_Y \cdot [\rho_{yY} \cdot \mathbf{Z}_3 + (1 - \rho^2_{yY})^{1/2} \cdot \mathbf{Z}_4] + \mu_Y \quad (26)$$

$$+0{,}50 < \rho_{xX}, \rho_{yY} < +1{,}00 \quad (27)$$

with: $\{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \mathbf{Z}_4\} \sim N(0, 1)$.

Generally, in MCS the total variance is underestimated when disregarding correlation between variables (Touran & Wiser, 1992).

### 3.4 Embedding Monte Carlo simulation

Construction managers are highly interested in obtaining the probability density function (pdf) of several critical outputs in their projects, such as normal and accelerated completion times, normal and increased direct costs due to project crashing, or the additional direct costs for speeding up the production process. A pdf can provide stakeholders better insight into the randomness of project performance, assisting in more effective decision-making on bidding, budgeting, and scheduling (Yao & Chu, 2007). The stochastic method selected in the research is Monte Carlo simulation (MCS). Using MCS, it is possible to calculate different probabilistic sets of artificial but more realistic task durations and costs, and then to apply a deterministic scheduling procedure to each set of these durations and costs. The result is the estimation of frequency distributions of project completion time and cost, so that the probability of meeting particular project deadlines or budgets can be assessed. MCS generates estimates by randomly calculating a feasible value for each critical variable from a statistical probability distribution function which represents the range and pattern of possible outcomes. To ensure that the chosen values are representative of these uncertain outputs, thousands repetitive deterministic calculations, known as iterations, are made (Bennett & Ormerod, 1984). MCS relies both on the central limit theorem (CLT) and the law of large numbers. The CLT states that the average of a sample of observations drawn from a population with any distribution shape is approx. normally distributed. More precisely, given a distribution with mean μ and variance $\sigma^2$, the sampling distribution of the mean approaches a normal distribution with a mean μ and a variance $\sigma^2/n$ as the sample size $n$ increases. The 'law of large numbers' states that if a stochastic process is sampled repeatedly, the mean value converges to the *true* expected value. The CLT holds, i.e., the sampling distribution of the mean approaches a normal distribution, no matter what the shape of the original distribution is.

Each time the LP problem (14)-(20) is solved using Solver, a probabilistic (stochastic) minimum project duration corresponding to its stochastic maximum total project direct cost (being the sum of normal direct cost and the additional cost for crashing activities) is estimated and stored. Furthermore, the generation of random values for normal ($\boldsymbol{x}_i$, $\boldsymbol{X}_i$) and crash ($\boldsymbol{y}_i$, $\boldsymbol{Y}_i$) pairs of task durations and direct costs, to be inserted in the LP solving process, also results in the calculation of the probabilistic normal project duration $T_n$ and normal total direct cost $C_x$, for each repetition.

### 3.5 Outline of sequential steps in model implementation

The required ten sequential steps in implementing the new stochastic project crashing model are:

*Step* (1)
Consult construction expert judgment, or analyse data from similar past projects, on expected uncertainty levels in both normal and crash durations and associated normal and crash direct costs of each activity, in the form of three-point estimates, i.e., optimistic (a), most likely (m), and pessimistic (b).

*Step* (2)
Use the classic PERT formulae (1) and (2) with the three-point estimates established in *Step* (1) to calculate the expected values for mean μ and standard deviation σ of each activity's random normal and crash durations (***x***, ***y***) together with normal and crash direct costs (***X***, ***Y***).

*Step* (3)
Generate uniform random z-values for each activity from the inverse cumulative standard normal distribution N(0, 1), using Excel's NORMSINV built-in function, to model the assumed strong positive correlation with a correlation coefficient value *randomly* chosen from +0,50 to +1,00 between pairs of normal durations and direct costs (***x***, ***X***), and of crash durations and direct costs (***y***, ***Y***), respectively.

*Step* (4)
Assuming that each pair of estimates for normal durations and direct costs, and crash durations and direct costs of each task, are *correlated* random variables that jointly follow a bivariate normal distribution N(μ, σ, *ρ*) with mean μ and standard deviation σ as calculated in *Step* (2) for each task, generate random pairs of estimates (***x***, ***X***) and (***y***, ***Y***) using equations (23)-(27).

*Step* (5)
Develop an acyclic and directed activity-on-node (AoN) project network that depicts accurately and realistically the technological precedence relationships between project tasks that must be executed with no interruption, including two auxiliary (dummy) project start and finish nodes. Use the random durations and direct costs generated in *Step* (4) to perform traditional CPM calculations to find the critical activities and to identify the critical path, for both normal and crash execution conditions.

*Step* (6)
Assuming a linear duration-direct cost relationship for each activity, calculate random maximum crash times $r^{max}$ and crash costs per time unit saved (e.g., per week or per day) or else slope *b*.

*Step* (7)
Formulate a LP mathematical model with the minimisation of project duration as the objective function (equation 14) which is subject to constraints (15)-(20). Solve this LP problem by an efficient LP algorithm or software (e.g., Solver) using the results from *Steps* (5)-(6) and store the results for the following variables: i). minimum project duration after project crashing $T_c$; ii). maximum additional total direct cost for project crashing $C_c$; iii). normal execution project duration $T_n$; iv). normal execution total project direct cost $C_x$; and v). total project cost for crash completion $C_t$.

*Step* (8)
Repeat *Steps* (3)-(7) iteratively (e.g., 1000 repetitions or more) to conduct MCS for the randomly generated correlated normal and crash durations (***x***, ***y***), and normal and crash direct costs (***X***, ***Y***) of each activity. The outcomes from *Step* (7) for output variables i). to v). are again stored after each iteration.

*Step* (9)
Use the results from *Step* (8) to calculate values for mean, variance, standard deviation, kurtosis, and skewness and to construct relevant frequency histograms and cumulative distribution *S*-curves, for the output variables in *Step* (7).

*Step* (10)
Calculate probabilities and lower/upper limits for confidence intervals, associated with critical project crashing duration and cost information to support managerial decision-making. The flowchart that follows (see Figure 3) describes the above ten steps for model implementation. The model is further automated with three discrete VBA codes (i. for generating random values, ii. for crashing with Solver, and iii. for propagating crash duration and additional cost for crashing results) that can also be executed with a (one-click) single button (see Appendix A).

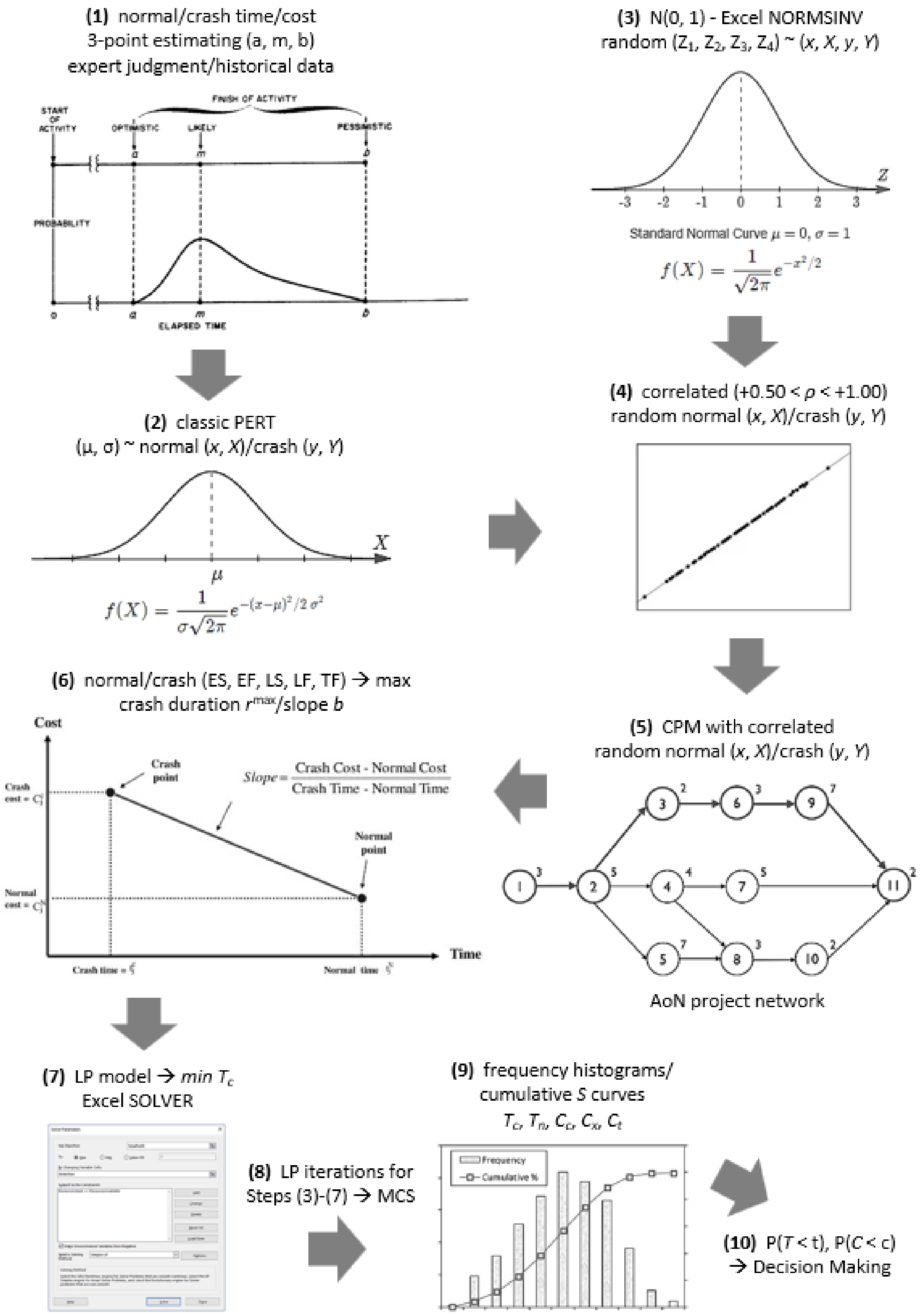



**Figure 3:** Flowchart with the sequential steps for model implementation

## 4. NUMERICAL EXAMPLE

The model is used for scheduling the construction of a residential building project taken from Ragsdale (2008: pp. 674). Table 1 summarizes all required CPM calculations for the example project, assuming a finish-to-start with no leads or lags precedence relationship between work activities. Total project duration under normal (most likely) working conditions is estimated to 46 weeks.

**Table 1:** CPM calculations for expected normal durations (in weeks)

| Activity | Description | Immediate Predecessors | Normal Duration | ES | EF | LS | LF | Total Float | Critical |
|---|---|---|---|---|---|---|---|---|---|
| A | Excavate | | 3 | 0 | 3 | 0 | 3 | 0 | Y |
| B | Lay foundations | A | 4 | 3 | 7 | 3 | 7 | 0 | Y |
| C | Rough plumbing | B | 3 | 7 | 10 | 22 | 25 | 15 | N |
| D | Frame | B | 10 | 7 | 17 | 7 | 17 | 0 | Y |
| E | Finish exterior | D | 8 | 17 | 25 | 17 | 25 | 0 | Y |
| F | Install HVAC | D | 4 | 17 | 21 | 21 | 25 | 4 | N |
| G | Rough electric | D | 6 | 17 | 23 | 19 | 25 | 2 | N |
| H | Sheetrock | C; E; F; G | 8 | 25 | 33 | 25 | 33 | 0 | Y |
| I | Install cabinets | H | 5 | 33 | 38 | 33 | 38 | 0 | Y |
| J | Paint | H | 5 | 33 | 38 | 35 | 40 | 2 | N |
| K | Final plumbing | I | 4 | 38 | 42 | 38 | 42 | 0 | Y |
| L | Final electric | J | 2 | 38 | 40 | 40 | 42 | 2 | N |
| M | Install flooring | K; L | 4 | 42 | 46 | 42 | 46 | 0 | Y |

Figure 4 below shows the constructed Gantt chart for the project with expected normal duration and total slack for each task (in weeks). Critical path is A-B-D-E-H-I-K-M.

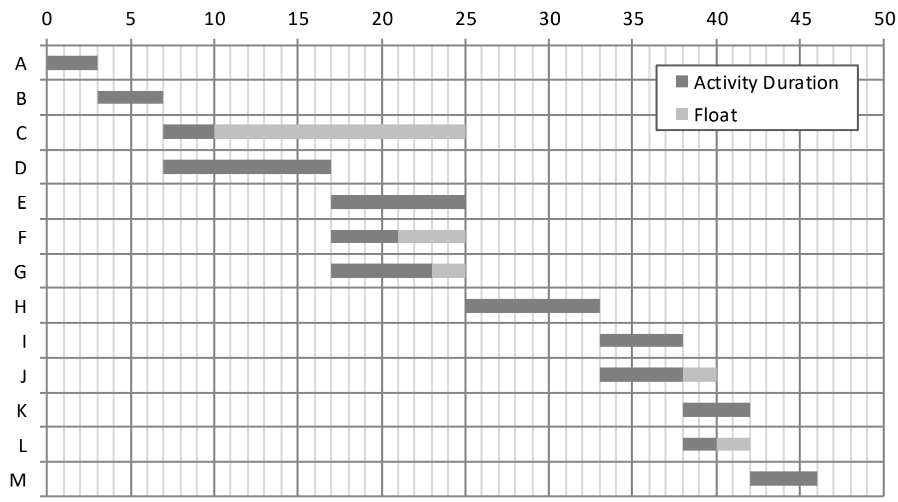



**Figure 4:** Gantt chart for the housing building project for expected normal durations (in weeks)

Figure 5 demonstrates that there is a weak linear relationship (no significant correlation) between project crash duration $T_c$ (in weeks) and additional cost for crashing the project $C_c$ (in €), after 1000 iterations.

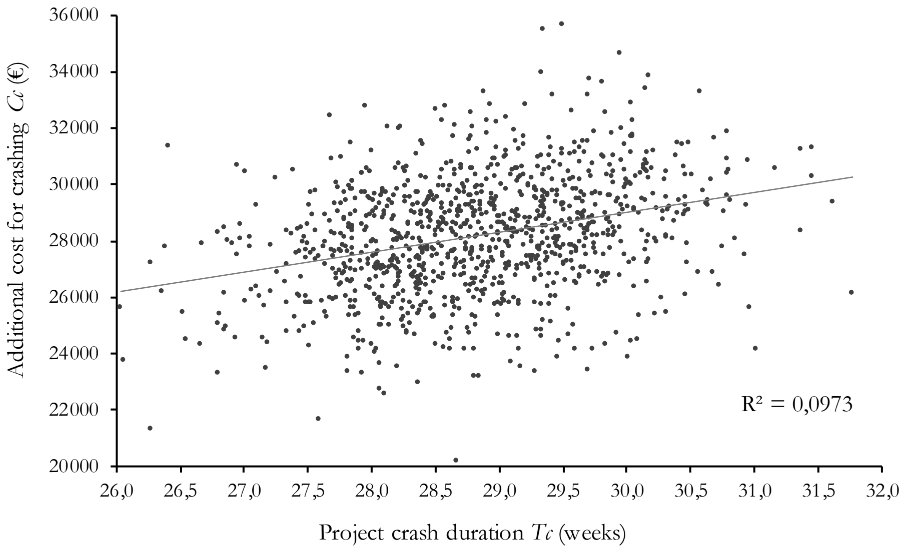



**Figure 5:** Weak correlation between crash duration ($T_c$) and additional cost for crashing ($C_c$)

Figure 6 and Table 2 present a frequency histogram and descriptive statistics respectively, for project crash duration $T_c$ after 1000 iterations. The mean crash makespan is 28,83 weeks with a standard deviation of 0,90 weeks and a coefficient of variation of 3,14%. Min value is 26,03 weeks and max value is 31,77 weeks.

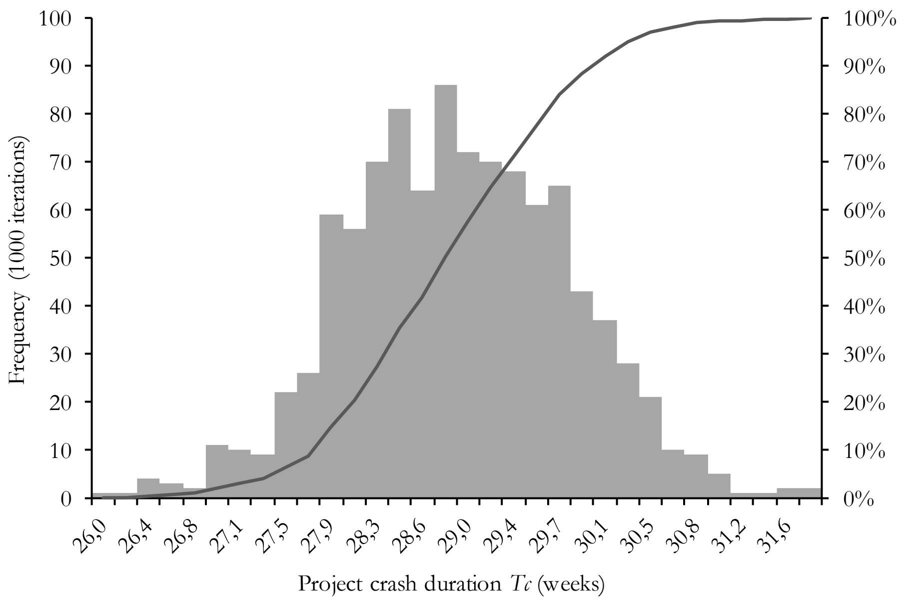



**Figure 6:** Frequency histogram (1000 iterations) for project crash duration $T_c$ (in weeks)

A 95% confidence interval can be constructed for crash project duration, with a lower limit of 28,77 weeks and an upper limit of 28,88 weeks.

**Table 2:** Descriptive statistics for project crash duration $T_c$ (in weeks)

| ***Tc*** | |
|---|---|
| Mean | **28,83** |
| Standard Error | 0,03 |
| Median | 28,80 |
| Coefficient of Variation | **3,14%** |
| Standard Deviation | **0,90** |
| Sample Variance | 0,82 |
| Kurtosis | -0,0021 |
| Skewness | 0,0117 |
| Range | 5,74 |
| Minimum | **26,03** |
| Maximum | **31,77** |
| Sum | 28826,54 |
| Count | 1000 |

Figure 7 and Table 3 present a frequency histogram and descriptive statistics respectively, for the additional cost for crashing the project $C_c$ after the same 1000 iterations. The mean crash cost is €28.181,08 with a standard deviation of €2.063,23 and a coefficient of variation of 7,32%. Min value is €20.231,63 and max value is €35.720,06.

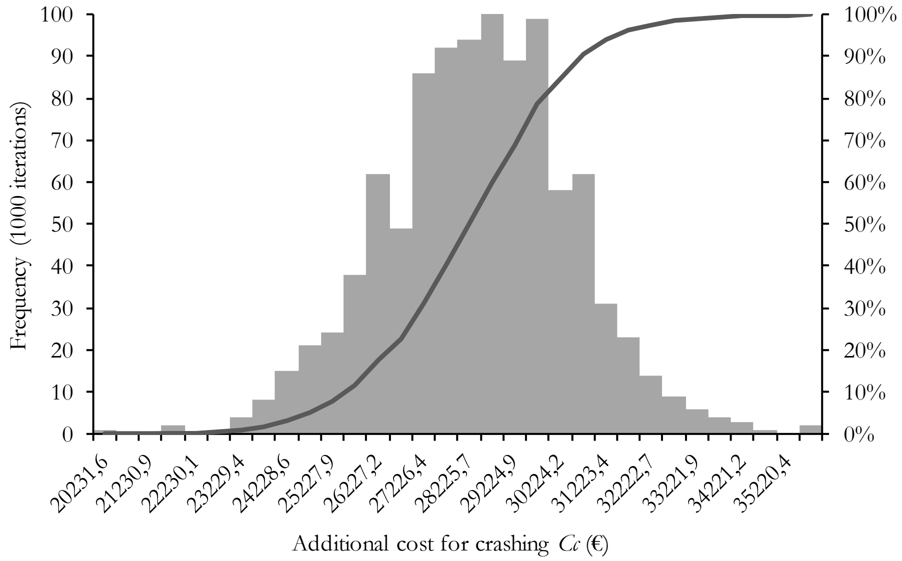



**Figure 7:** Frequency histogram (1000 iterations) for additional cost for project crashing $C_c$ (in €)

Again, a 95% confidence interval can be constructed for additional cost for crashing the project, with a lower limit of €28.053,20 and an upper limit of €28.308,96.

**Table 3:** Descriptive statistics for additional cost for project crashing $C_c$ (in €)

| | ***Cc*** |
|---|---|
| Mean | **28181,08** |
| Standard Error | 65,25 |
| Median | 28238,41 |
| Coefficient of Variation | **7,32%** |
| Standard Deviation | **2063,23** |
| Sample Variance | 4256919,55 |
| Kurtosis | 0,3856 |
| Skewness | -0,0343 |
| Range | 15488,43 |
| Minimum | **20231,63** |
| Maximum | **35720,06** |
| Sum | 28181081 |
| Count | 1000 |

## 5. CONCLUSION

In the field of construction project scheduling, finding the distribution shape of completion time and cost in a PERT/CPM network is still an active research area. Determining a project's duration/ direct cost relationship can be critical to all construction professionals and especially to clients by emphasizing the significance of 'crashing' the project for earliest completion on the maximisation of their capital outlay. However, deterministic time-cost optimisation models for project crashing suffer from the assumption of complete information and disregard the uncertainty which is endemic in the construction production process. As a result, the expected normal as well as crash project durations and expenses are often underestimated in practice.

The research presented herein aims at developing an integrated spreadsheet model for the practical solution of the stochastic project crashing problem in construction planning. The introduced mathematical model is based on the integration of classic PERT/CPM network techniques, probabilistic MCS, and LP optimisation. The main contribution of the model to the existing project planning literature is that it produces frequency histograms and relevant statistical information for optimum project crash makespan and additional cost for project compression, by assuming uncertainty and correlation simultaneously for both normal and crash durations and direct expenses of project activities. The implementation of the model is automated in Excel with VBA coding. The research is anticipated to assist both academics and professionals operating in the built environment towards more effective decision-making in planning construction projects.

## REFERENCES


Aghaie, A., & Mokhtari, H. (2009). Ant colony optimization algorithm for stochastic project crashing problem in PERT networks using MC simulation. *The International Journal of Advanced Manufacturing Technology*, 45, 1051-1067.

Ahipasaoglu, S. D., Natarajan, K., & Shi, D. (2019). Distributionally robust project crashing with partial or no correlation information. *Networks*, 74(1), 79-106.

Bennett, J., & Ormerod, R. N. (1984). Simulation applied to construction projects. *Construction Management and Economics*, 2(3), 225-263.

Cooper, D. F., Grey, S., Raymond, G., & Walker, P. (2005). *Project risk management guidelines*. Wiley.

Coskun, O. (1984). Optimal probabilistic compression of PERT networks. *Journal of Construction Engineering and Management*, 110(4), 437-446.

Demeulemeester, E. L., & Herroelen, W. S. (2006). *Project scheduling: a research handbook*, 49, Springer Science & Business Media.

De Marco, A. (2011). Project Management for Facility Constructions: A Guide for Engineers and Architects. Springer.

Dodin, B. (1985). Approximating the distribution functions in stochastic networks. *Computers & Operations Research*, 12(3), 251-264.

Eshtehardian, E., Afshar, A., & Abbasnia, R. (2008). Time–cost optimization: using GA and fuzzy sets theory for uncertainties in cost. *Construction Management and Economics*, 26, 679-691, DOI: 10.1080/01446190802036128.

Feng, C., Liu, L., & Burns, S. A. (2000). Stochastic Construction Time-Cost Trade-Off Analysis. *Journal of Computing in Civil Engineering*, 14(2), 117-126.

Garvey, P. R., Book, S. A., & Covert, R. P. (2016). *Probability methods for cost uncertainty analysis: A systems engineering perspective*. CRC Press.

Gutjahr, W. J., Strauss, C., & Wagner, E. (2000). A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing*, 12, 125-135.

Hajdu, M., & Isaac, S. (2016). Sixty years of project planning: history and future. *Organization, Technology and Management in Construction: an International Journal*, 8(1), 1499-1510.

Hajdu, M. (2013). Effects of the application of activity calendars on the distribution of project duration in PERT networks. *Automation in Construction*, 35, 397-404.

Harrison, F., & Lock, D. (2017). *Advanced project management: a structured approach*. Routledge.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2), 289-306.

Hillson, D. (2002). Extending the risk process to manage opportunities. *International Journal of Project Management*, 20(3), 235-240.

Hulett, D. (2016). *Practical schedule risk analysis*. Routledge.

Johnson, G. A., & Schou, C. D. (1990). Expediting projects in PERT with stochastic time estimates. *Project Management Journal*, 21(2), 29-34.

Kang, C., & Choi, B. C. (2015). An adaptive crashing policy for stochastic time-cost tradeoff problems. *Computers & Operations Research*, 63, 1-6.

Ke, H. (2014). Uncertain random time-cost trade-off problem. *Journal of Uncertainty Analysis and Applications*, 2(23).

Ke, H., Ma, W., & Ni, Y. (2009). Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem. *Applied Mathematics and Computation*, 215(1), 308-313.

Kelley, J. E. (1961). Critical Path Planning and Scheduling: Mathematical Basis. *Operational Research Journal*, 9(3), 167-169.

Kelley, J. E., & Walker, M. R. (1959). Critical-path planning and scheduling. In Papers presented at the December 1-3, 1959, *eastern joint IRE-AIEE-ACM computer conference*, 160-173.

Klastorin, T. (2004). *Project Management: Tools and Tradeoffs*. John Wiley and Sons Inc.

Kwak, Y. H., & Ingall, L. (2007). Exploring Monte Carlo simulation applications for project management. *Risk management*, 9(1), 44-57.

Lind, D. A., Marchal, W. G., & Wathen, S. A. (2017). *Statistical techniques in business & economics*. McGraw-Hill Education.

Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. (1959). Application of a technique for research and development program evaluation. *Operations Research*, 7(5), 646-669.

Mitchell, G., & Klastorin, T. (2007). An effective methodology for the stochastic project compression problem, *IIE Transactions*, 39(10), 957-969, DOI: 10.1080/07408170701315347.

Möhring, R. H. (2001). Scheduling under uncertainty: Bounding the makespan distribution. In *Computational Discrete Mathematics* (79-97). Springer, Berlin, Heidelberg.

Mokhtari, H., Aghaie, A., Rahimi, J., & Mozdgir, A. (2010). Project time–cost trade-off scheduling: a hybrid optimization approach. *The international journal of advanced manufacturing technology*, 50(5), 811-822.

Montgomery, D. C., & Runger, G. C. (2014). *Applied statistics and probability for engineers*. 6th ed., John Wiley and Sons.

Moore, J. H., & Weatherford, L. R. (2001). *Decision Modeling with Microsoft Excel*, 6th ed., Prentice Hall.

Mubarak, S. A. (2015). *Construction project scheduling and control*. John Wiley & Sons.

Onifade, M. K., Afolabi, J., & Babawale, A. (2017). Application of Project Evaluation and Review Technique (PERT) in Road Construction Projects in Nigeria. *European Project Management Journal*, **7**, 3-13.

Oxley, R., & Poskitt, J. (1996). *Management Techniques Applied to the Construction Industry*, 5th ed., Oxford: Blackwell Science.

Ragsdale, C. T. (2008). *Spreadsheet modeling and decision analysis: a practical introduction to business analytics*. 5th ed., Nelson Education.

Roos, E., & den Hertog, D. (2020). Reducing conservatism in robust optimization. *INFORMS Journal on Computing*, 32(4), 1109-1127, https://doi.org/10.1287/ijoc.2019.0913

Schwindt, C. (2006). *Resource allocation in project management*. Springer Science & Business Media.

Touran, A., & Wiser, E. P. (1992). Monte Carlo technique with correlated random variables. *Journal of Construction Engineering and Management*, 118(2), 258-272.

Van Slyke, R. M. (1963). Monte Carlo methods and the PERT problem. *Operations Research*, 13, 141-143.

Vanhoucke, M. (2013). *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*, 2nd ed., Springer.

Vose, D. (2008). *Risk analysis: a quantitative guide*. John Wiley & Sons.

Willis, R. J. (1985). Critical path analysis and resource constrained project scheduling—theory and practice. *European Journal of Operational Research*, 21(2), 149-155.

Wollmer, R. D. (1985). Critical path planning under uncertainty. *Mathematical Programming Study*, 25, 164-171.

Yang, H., & Morton, D. P. (2022). Optimal crashing of an activity network with disruptions. *Mathematical Programming*, 194, 1113-1162, https://doi.org/10.1007/s10107-021-01670-x.

Yang, I.-T. (2005). Impact of budget uncertainty on project time-cost tradeoff, *IEEE Transactions on Engineering Management*, 52(2), 167-174, doi: 10.1109/TEM.2005.844924

Yao, M. J., & Chu, W. M. (2007). A new approximation algorithm for obtaining the probability distribution function for project completion time. *Computers & mathematics with Applications*, 54(2), 282-295.

## Appendices

A. VBA codes for the automation of the stochastic project crashing model (see also Appendix B):

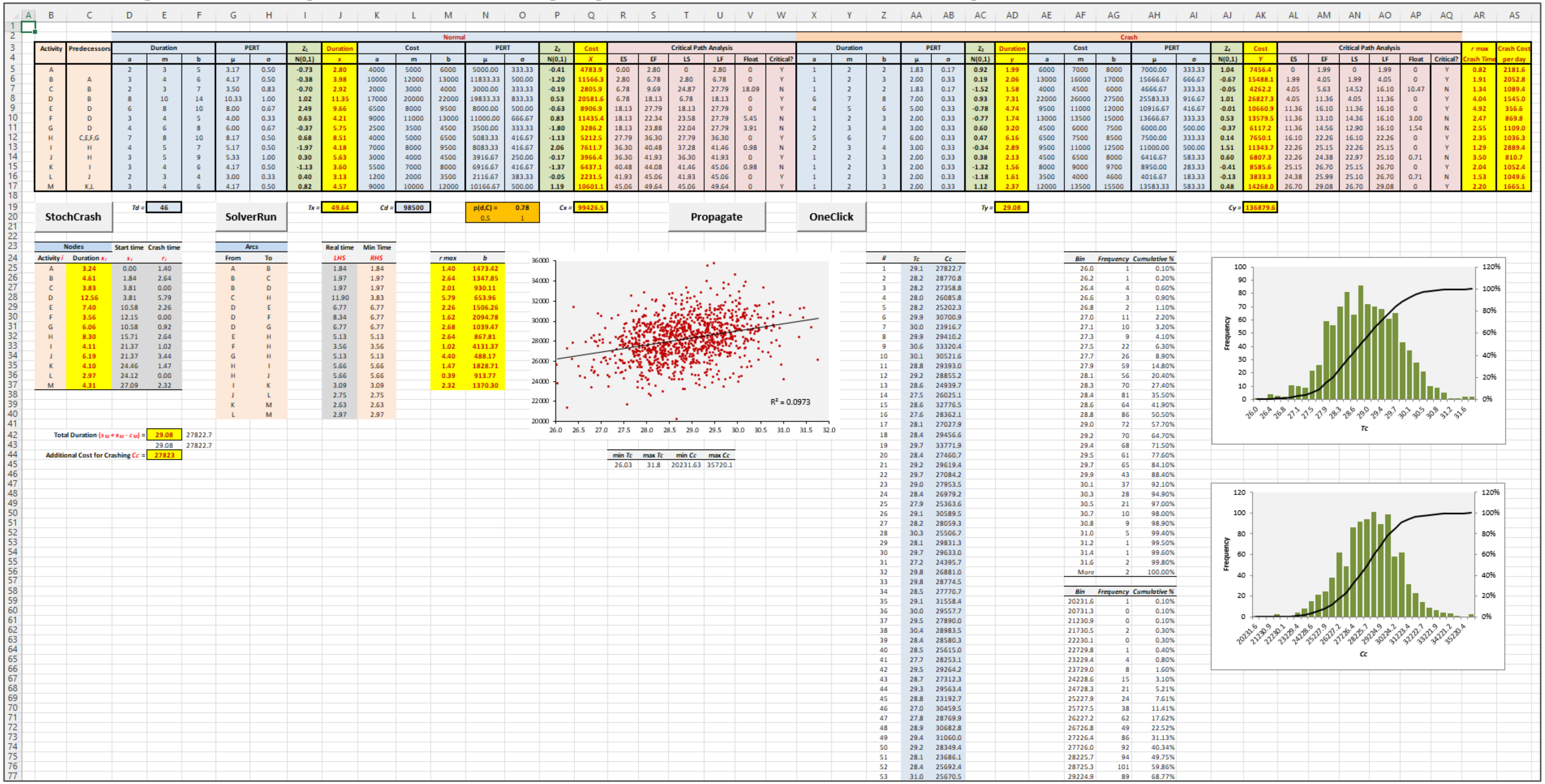
```
Sub StochCrash()
' StochCrash Macro
    Calculate
    Range("J5:J17").Select
    Selection.Copy
    Range("C25:C37").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Range("AR5:AS17").Select
    Application.CutCopyMode = False
    Selection.Copy
    Range("M25:N37").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
End Sub

Sub SolverRun()
' SolverRun Macro
    SolverReset
    SolverOk SetCell:="$E$42", MaxMinVal:=2, ValueOf:=0, By Change:="$D$25:$E$37", _
        Engine:=2, EngineDesc:="Simplex LP"
    SolverAdd CellRef:="$D$25:$E$36", Relation:=3, FormulaText:="0"
    SolverAdd CellRef:="$E$25:$E$37", Relation:=1, FormulaText:="$M$25:$M$37"
    SolverAdd CellRef:="$J$25:$J$40", Relation:=3, FormulaText:="$K$25:$K$40"
    SolverOk SetCell:="$E$42", MaxMinVal:=2, ValueOf:=0, ByChange:="$D$25:$E$37", _
        Engine:=2, EngineDesc:="Simplex LP"
    SolverSolve userFinish:=True
End Sub

Sub Propag()
' Propag Macro
    Range("E42:F42").Select
    Selection.Copy
    Range("E43:F43").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Selection.Copy
    Range("AA25:AB25").Select
    Selection.Insert Shift:=xlDown
End Sub

Sub OneClick()
    Call StochCrash
    Call SolverRun
    Call Propag
End Sub
```

B. Excel snapshot with the implementation of the developed spreadsheet model for the numerical example:



Normal

| Activity | Predecessors | Duration a | Duration m | Duration b | PERT μ | PERT σ | $Z_1$ N(0,1) | Duration x | Cost a | Cost m | Cost b | PERT μ | PERT σ | $Z_2$ N(0,1) | Cost X | ES | EF | LS | LF | Float | Critical? |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | 2 | 3 | 5 | 3.17 | 0.50 | -0.73 | 2.80 | 4000 | 5000 | 6000 | 5000.00 | 333.33 | -0.41 | 4783.9 | 0.00 | 2.80 | 0 | 2.80 | 0 | Y |
| B | A | 3 | 4 | 6 | 4.17 | 0.50 | -0.38 | 3.98 | 10000 | 12000 | 13000 | 11833.33 | 500.00 | -1.20 | 11566.3 | 2.80 | 6.78 | 2.80 | 6.78 | 0 | Y |
| C | B | 2 | 3 | 7 | 3.50 | 0.83 | -0.70 | 2.92 | 2000 | 3000 | 4000 | 3000.00 | 333.33 | -0.19 | 2805.9 | 6.78 | 9.69 | 24.87 | 27.79 | 18.09 | N |
| D | B | 8 | 10 | 14 | 10.33 | 1.00 | 1.02 | 11.35 | 17000 | 20000 | 22000 | 19833.33 | 833.33 | 0.53 | 20581.6 | 6.78 | 18.13 | 6.78 | 18.13 | 0 | Y |
| E | D | 6 | 8 | 10 | 8.00 | 0.67 | 2.49 | 9.66 | 6500 | 8000 | 9500 | 8000.00 | 500.00 | -0.63 | 8906.9 | 18.13 | 27.79 | 18.13 | 27.79 | 0 | Y |
| F | D | 3 | 4 | 5 | 4.00 | 0.33 | 0.63 | 4.21 | 9000 | 11000 | 13000 | 11000.00 | 666.67 | 0.83 | 11435.4 | 18.13 | 22.34 | 23.58 | 27.79 | 5.45 | N |
| G | D | 4 | 6 | 8 | 6.00 | 0.67 | -0.37 | 5.75 | 2500 | 3500 | 4500 | 3500.00 | 333.33 | -1.80 | 3286.2 | 18.13 | 23.88 | 22.04 | 27.79 | 3.91 | N |
| H | C,E,F,G | 7 | 8 | 10 | 8.17 | 0.50 | 0.68 | 8.51 | 4000 | 5000 | 6500 | 5083.33 | 416.67 | -1.13 | 5212.5 | 27.79 | 36.30 | 27.79 | 36.30 | 0 | Y |
| I | H | 4 | 5 | 7 | 5.17 | 0.50 | -1.97 | 4.18 | 7000 | 8000 | 9500 | 8083.33 | 416.67 | 2.06 | 7611.7 | 36.30 | 40.48 | 37.28 | 41.46 | 0.98 | N |
| J | H | 3 | 5 | 9 | 5.33 | 1.00 | 0.30 | 5.63 | 3000 | 4000 | 4500 | 3916.67 | 250.00 | -0.17 | 3966.4 | 36.30 | 41.93 | 36.30 | 41.93 | 0 | Y |
| K | I | 3 | 4 | 6 | 4.17 | 0.50 | -1.13 | 3.60 | 5500 | 7000 | 8000 | 6916.67 | 416.67 | -1.37 | 6437.1 | 40.48 | 44.08 | 41.46 | 45.06 | 0.98 | N |
| L | J | 2 | 3 | 4 | 3.00 | 0.33 | 0.40 | 3.13 | 1200 | 2000 | 3500 | 2116.67 | 383.33 | -0.05 | 2231.5 | 41.93 | 45.06 | 41.93 | 45.06 | 0 | Y |
| M | K,L | 3 | 4 | 6 | 4.17 | 0.50 | 0.82 | 4.57 | 9000 | 10000 | 12000 | 10166.67 | 500.00 | 1.19 | 10601.1 | 45.06 | 49.64 | 45.06 | 49.64 | 0 | Y |

Crash

| Activity | Duration a | Duration m | Duration b | PERT μ | PERT σ | $Z_3$ N(0,1) | Duration y | Cost a | Cost m | Cost b | PERT μ | PERT σ | $Z_4$ N(0,1) | Cost Y | ES | EF | LS | LF | Float | Critical? | r max Crash Time | Crash Cost per day |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 2 | 2 | 1.83 | 0.17 | 0.92 | 1.99 | 6000 | 7000 | 8000 | 7000.00 | 333.33 | 1.04 | 7456.4 | 0 | 1.99 | 0 | 1.99 | 0 | Y | 0.82 | 2181.6 |
| B | 1 | 2 | 3 | 2.00 | 0.33 | 0.19 | 2.06 | 13000 | 16000 | 17000 | 15666.67 | 666.67 | -0.67 | 15488.1 | 1.99 | 4.05 | 1.99 | 4.05 | 0 | Y | 1.91 | 2052.8 |
| C | 1 | 2 | 2 | 1.83 | 0.17 | -1.52 | 1.58 | 4000 | 4500 | 6000 | 4666.67 | 333.33 | -0.05 | 4262.2 | 4.05 | 5.63 | 14.52 | 16.10 | 10.47 | N | 1.34 | 1089.4 |
| D | 6 | 7 | 8 | 7.00 | 0.33 | 0.93 | 7.31 | 22000 | 26000 | 27500 | 25583.33 | 916.67 | 1.01 | 26827.3 | 4.05 | 11.36 | 4.05 | 11.36 | 0 | Y | 4.04 | 1545.0 |
| E | 4 | 5 | 6 | 5.00 | 0.33 | -0.78 | 4.74 | 9500 | 11000 | 12000 | 10916.67 | 416.67 | -0.01 | 10660.9 | 11.36 | 16.10 | 11.36 | 16.10 | 0 | Y | 4.92 | 356.6 |
| F | 1 | 2 | 3 | 2.00 | 0.33 | -0.77 | 1.74 | 13000 | 13500 | 15000 | 13666.67 | 333.33 | 0.53 | 13579.5 | 11.36 | 13.10 | 14.36 | 16.10 | 3.00 | N | 2.47 | 869.8 |
| G | 2 | 3 | 4 | 3.00 | 0.33 | 0.60 | 3.20 | 4500 | 6000 | 7500 | 6000.00 | 500.00 | -0.37 | 6117.2 | 11.36 | 14.56 | 12.90 | 16.10 | 1.54 | N | 2.55 | 1109.0 |
| H | 5 | 6 | 7 | 6.00 | 0.33 | 0.47 | 6.16 | 6500 | 7500 | 8500 | 7500.00 | 333.33 | 0.14 | 7650.1 | 16.10 | 22.26 | 16.10 | 22.26 | 0 | Y | 2.35 | 1036.3 |
| I | 2 | 3 | 4 | 3.00 | 0.33 | -0.34 | 2.89 | 9500 | 11000 | 12500 | 11000.00 | 500.00 | 1.51 | 11343.7 | 22.26 | 25.15 | 22.26 | 25.15 | 0 | Y | 1.29 | 2889.4 |
| J | 1 | 2 | 3 | 2.00 | 0.33 | 0.38 | 2.13 | 4500 | 6500 | 8000 | 6416.67 | 583.33 | 0.60 | 6807.3 | 22.26 | 24.38 | 22.97 | 25.10 | 0.71 | N | 3.50 | 810.7 |
| K | 1 | 2 | 3 | 2.00 | 0.33 | -1.32 | 1.56 | 8000 | 9000 | 9700 | 8950.00 | 283.33 | -0.41 | 8585.6 | 25.15 | 26.70 | 25.15 | 26.70 | 0 | Y | 2.04 | 1052.4 |
| L | 1 | 2 | 3 | 2.00 | 0.33 | -1.18 | 1.61 | 3500 | 4000 | 4600 | 4016.67 | 183.33 | -0.13 | 3833.3 | 24.38 | 25.99 | 25.10 | 26.70 | 0.71 | N | 1.53 | 1049.6 |
| M | 1 | 2 | 3 | 2.00 | 0.33 | 1.12 | 2.37 | 12000 | 13500 | 15500 | 13583.33 | 583.33 | 0.48 | 14268.0 | 26.70 | 29.08 | 26.70 | 29.08 | 0 | Y | 2.20 | 1665.1 |

StochCrash  $Td$ = 46  SolverRun  $Tx$ = 49.64  $Cd$ = 98500  p(d,C) = 0.78 (0.5, 1)  $Cx$ = 99426.5  Propagate  OneClick  $Ty$ = 29.08  $Cy$ = 136879.6

| Nodes: Activity $i$ | Duration $x_i$ | Start time $s_i$ | Crash time $r_i$ |
|---|---|---|---|
| A | 3.24 | 0.00 | 1.40 |
| B | 4.61 | 1.84 | 2.64 |
| C | 3.83 | 3.81 | 0.00 |
| D | 12.56 | 3.81 | 5.79 |
| E | 7.40 | 10.58 | 2.26 |
| F | 3.56 | 12.15 | 0.00 |
| G | 6.06 | 10.58 | 0.92 |
| H | 8.30 | 15.71 | 2.64 |
| I | 4.11 | 21.37 | 1.02 |
| J | 6.19 | 21.37 | 3.44 |
| K | 4.10 | 24.46 | 1.47 |
| L | 2.97 | 24.12 | 0.00 |
| M | 4.31 | 27.09 | 2.32 |

| Arcs: From | To | Real time LHS | Min Time RHS |
|---|---|---|---|
| A | B | 1.84 | 1.84 |
| B | C | 1.97 | 1.97 |
| B | D | 1.97 | 1.97 |
| C | H | 11.90 | 3.83 |
| D | E | 6.77 | 6.77 |
| D | F | 8.34 | 6.77 |
| D | G | 6.77 | 6.77 |
| E | H | 5.13 | 5.13 |
| F | H | 3.56 | 3.56 |
| G | H | 5.13 | 5.13 |
| H | I | 5.66 | 5.66 |
| H | J | 5.66 | 5.66 |
| I | K | 3.09 | 3.09 |
| J | L | 2.75 | 2.75 |
| K | M | 2.63 | 2.63 |
| L | M | 2.97 | 2.97 |

| r max | b |
|---|---|
| 1.40 | 1473.42 |
| 2.64 | 1347.85 |
| 2.01 | 930.11 |
| 5.79 | 653.96 |
| 2.26 | 1506.26 |
| 1.62 | 2094.78 |
| 2.68 | 1039.47 |
| 2.64 | 867.81 |
| 1.02 | 4131.37 |
| 4.40 | 488.17 |
| 1.47 | 1828.71 |
| 0.39 | 913.77 |
| 2.32 | 1370.30 |

Total Duration ($s_M + x_M - c_M$) = 29.08  27822.7
29.08  27822.7
Additional Cost for Crashing $Cc$ = 27823

| min Tc | max Tc | min Cc | max Cc |
|---|---|---|---|
| 26.03 | 31.8 | 20231.63 | 35720.1 |

| # | Tc | Cc |
|---|---|---|
| 1 | 29.1 | 27822.7 |
| 2 | 28.2 | 28770.8 |
| 3 | 28.2 | 27358.8 |
| 4 | 28.0 | 26085.8 |
| 5 | 28.2 | 25202.3 |
| 6 | 29.9 | 30700.9 |
| 7 | 30.0 | 23916.7 |
| 8 | 29.9 | 29410.2 |
| 9 | 30.6 | 33320.4 |
| 10 | 30.1 | 30521.6 |
| 11 | 28.8 | 29393.0 |
| 12 | 29.2 | 28855.2 |
| 13 | 28.6 | 24939.7 |
| 14 | 27.5 | 26025.1 |
| 15 | 28.6 | 32776.5 |
| 16 | 27.6 | 28362.1 |
| 17 | 28.1 | 27027.9 |
| 18 | 28.4 | 29456.6 |
| 19 | 29.7 | 33771.9 |
| 20 | 28.4 | 27460.7 |
| 21 | 29.2 | 29619.4 |
| 22 | 29.7 | 27084.2 |
| 23 | 29.0 | 27953.5 |
| 24 | 28.4 | 26979.2 |
| 25 | 27.9 | 25363.6 |
| 26 | 29.1 | 30589.5 |
| 27 | 28.2 | 28059.3 |
| 28 | 30.3 | 25506.7 |
| 29 | 28.1 | 29831.3 |
| 30 | 29.7 | 29633.0 |
| 31 | 27.2 | 24395.7 |
| 32 | 29.8 | 26881.0 |
| 33 | 29.8 | 28774.5 |
| 34 | 28.5 | 27770.7 |
| 35 | 29.1 | 31558.4 |
| 36 | 30.0 | 29557.7 |
| 37 | 29.5 | 27890.0 |
| 38 | 30.4 | 28983.5 |
| 39 | 28.4 | 28580.3 |
| 40 | 28.5 | 25615.0 |
| 41 | 27.7 | 28253.1 |
| 42 | 29.5 | 29264.2 |
| 43 | 28.7 | 27312.3 |
| 44 | 29.3 | 29563.4 |
| 45 | 28.8 | 23192.7 |
| 46 | 27.0 | 30459.5 |
| 47 | 27.8 | 28769.9 |
| 48 | 28.9 | 30682.8 |
| 49 | 29.4 | 31060.0 |
| 50 | 29.2 | 28349.4 |
| 51 | 28.1 | 23686.1 |
| 52 | 28.4 | 25692.4 |
| 53 | 31.0 | 25670.5 |

| Bin | Frequency | Cumulative % |
|---|---|---|
| 26.0 | 1 | 0.10% |
| 26.2 | 1 | 0.20% |
| 26.4 | 4 | 0.60% |
| 26.6 | 3 | 0.90% |
| 26.8 | 2 | 1.10% |
| 27.0 | 11 | 2.20% |
| 27.1 | 10 | 3.20% |
| 27.3 | 9 | 4.10% |
| 27.5 | 22 | 6.30% |
| 27.7 | 26 | 8.90% |
| 27.9 | 59 | 14.80% |
| 28.1 | 56 | 20.40% |
| 28.3 | 70 | 27.40% |
| 28.4 | 81 | 35.50% |
| 28.6 | 64 | 41.90% |
| 28.8 | 86 | 50.50% |
| 29.0 | 72 | 57.70% |
| 29.2 | 70 | 64.70% |
| 29.4 | 68 | 71.50% |
| 29.5 | 61 | 77.60% |
| 29.7 | 65 | 84.10% |
| 29.9 | 43 | 88.40% |
| 30.1 | 37 | 92.10% |
| 30.3 | 28 | 94.90% |
| 30.5 | 21 | 97.00% |
| 30.7 | 10 | 98.00% |
| 30.8 | 9 | 98.90% |
| 31.0 | 5 | 99.40% |
| 31.2 | 1 | 99.50% |
| 31.4 | 1 | 99.60% |
| 31.6 | 2 | 99.80% |
| More | 2 | 100.00% |

| Bin | Frequency | Cumulative % |
|---|---|---|
| 20231.6 | 1 | 0.10% |
| 20731.3 | 0 | 0.10% |
| 21230.9 | 0 | 0.10% |
| 21730.5 | 2 | 0.30% |
| 22230.1 | 0 | 0.30% |
| 22729.8 | 1 | 0.40% |
| 23229.4 | 4 | 0.80% |
| 23729.0 | 8 | 1.60% |
| 24228.6 | 15 | 3.10% |
| 24728.3 | 21 | 5.21% |
| 25227.9 | 24 | 7.61% |
| 25727.5 | 38 | 11.41% |
| 26227.2 | 62 | 17.62% |
| 26726.8 | 49 | 22.52% |
| 27226.4 | 86 | 31.13% |
| 27726.0 | 92 | 40.34% |
| 28225.7 | 94 | 49.75% |
| 28725.3 | 101 | 59.86% |
| 29224.9 | 89 | 68.77% |

C. Excel formulae used in the numerical example (see also Appendix B):

| | | |
|---|---|---|
| G5 | =(D5+4*E5+F5)/6 | copied to G6:G17 |
| H5 | =SQRT((F5-D5)^2/36) | copied to H6:H17 |
| I5 | =NORMINV(RAND();0;1) | copied to I6:I17 |
| J5 | =G5+H5*I5 | copied to J6:J17 |
| J19 | =SUMIF(W5:W17;"Y";J5:J17) | |
| L19 | =SUM(L5:L17) | |
| N5 | =(K5+4*L5+M5)/6 | copied to N6:N17 |
| O5 | =SQRT((M5-K5)^2/36) | copied to O6:O17 |
| P5 | =NORMINV(RAND();0;1) | copied to P6:P17 |
| O19 | =RAND()*(O20-N20)+N20 | |
| Q5 | =N5+O5*(I5*$O$19+P5*(1-$O$19^2)^1/2) | copied to Q6:Q17 |
| Q19 | =SUM(Q5:Q17) | |
| R5 | {=MAX(IF(ISERR(FIND($B$5:$B$17;C5));0;$S$5:$S$17))} | copied to R6:R17 |
| S5 | =R5+J5 | copied to S6:S17 |
| T5 | =IF(U5-J5<0,0001;"0";U5-J5) | copied to T6:T17 |
| U5 | {=MIN(IF(ISERR(FIND(B5;$C$5:$C$17));MAX($S$5:$S$17);$T$5:$T$17))} | copied to U6:U17 |
| V5 | =IF(U5-S5<0,0001;"0";U5-S5) | copied to V6:V17 |
| W5 | =IF(U5-S5<0,0001;"Y";"N") | copied to W6:W17 |
| AA5 | =(X5+4*Y5+Z5)/6 | copied to AA6:AA17 |
| AB5 | =SQRT((Z5-X5)^2/36) | copied to AB6:AB17 |
| AC5 | =NORMINV(RAND();0;1) | copied to AC6:AC17 |
| AD5 | =AA5+AB5*AC5 | copied to AD6:AD17 |
| AD19 | =SUMIF(AQ5:AQ17;"Y";AD5:AD17) | |
| AH5 | =(AE5+4*AF5+AG5)/6 | copied to AH6:AH17 |
| AI5 | =SQRT((AG5-AE5)^2/36) | copied to AI6:AI17 |
| AJ5 | =NORMINV(RAND();0;1) | copied to AJ6:AJ17 |
| AK5 | =AH5+AI5*(AC5*$O$19+AJ5*(1-$O$19^2)^(1/2)) | copied to AK6:AK17 |
| AK19 | =SUM(AK5:AK17) | |
| AL5 | {=MAX(IF(ISERR(FIND($B$5:$B$17;C5));0;$AM$5:$AM$17))} | copied to AL6:AL17 |
| AM5 | =AL5+AD5 | copied to AM6:AM17 |
| AN5 | =IF(AO5-AD5<0,0001;"0";AO5-AD5) | copied to AN6:AN17 |
| AO5 | {=MIN(IF(ISERR(FIND(B5;$C$5:$C$17));MAX($AM$5:$AM$17);$AN$5:$AN$17))} | copied to AO6:AO17 |
| AP5 | =IF(AO5-AM5<0,0001;"0";AO5-AM5) | copied to AP6:AP17 |
| AQ5 | =IF(AO5-AM5<0,0001;"Y";"N") | copied to AQ6:AQ17 |
| AR5 | =IF(J5-AD5<0;"-";J5-AD5) | copied to AR6:AR17 |
| AS5 | =IF(AR5<1;(AK5-Q5)*AR5;(AK5-Q5)/AR5) | copied to AS6:AS17 |
| J25 | =VLOOKUP(H25;$B$25:$D$37;3)-VLOOKUP(G25;$B$25:$D$37;3) | copied to J26:J40 |
| K25 | =VLOOKUP(G25;$B$25:$C$37;2)-VLOOKUP(G25;$B$25:$E$37;4) | copied to K26:K40 |
| E42 | =D37+C37-E37 | |
| F42 | =SUMPRODUCT(E25:E37;N25:N37) | |
| E44 | =SUMPRODUCT(E25:E37;N25:N37) | |
| R45 | =MIN(AA25:AA1024) | |
| S45 | =MAX(AA25:AA1024) | |
| T45 | =MIN(AB25:AB1024) | |
| U45 | =MAX(AB25:AB1024) | |

D. Solver parameters used in the numerical example (see also Appendix B):