Earned Schedule min-max: two new EVM metrics for monitoring and controlling projects

Ballesteros-Pérez, P. Ph.D.[1] ; Sanz-Ablanedo, E. Ph.D.[2];Mora-Melià, D. Ph.D.[3]

González-Cruz,Mª.C. Ph.D.[4]*;Fuentes-Bargues, J.L. Ph.D.[5]; Pellicer, E. Ph.D.[6]

[1*] Senior researcher. Dpto. de Ingeniería Mecánica y Diseño Industrial, Escuela Superior de Ingeniería.Universidad de Cádiz, Avda. Univ. de Cádiz 10, Puerto Real,11519Cádiz, Spain.

Phone: +34 956 483200, e-mail: pablo.ballesteros@uca.es

[2]AssociateProfessor. Dpto. Tecnología Minera, Topográfica y Estructuras.

Universidad de León. Avda. Astorga, s/n. 24400 Ponferrada, León. Spain.

Phone: +34 987 442 110, e-mail: esana@unileon.es

[3]AssociateProfessor. Dpto. de Ingeniería y Gestión de la Construcción.

Universidad de Talca. Camino Los Nicheskm.1, 3340000Curicó, Chile.

Phone: +56 951 265 044, e-mail: damora@utalca.cl

[4]AssociateProfessor. GIDDP, Dpto. de Proyectos de Ingeniería.

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

Corresponding author. Phone: +34 963 879 866, e-mail: mcgonzal@dpi.upv.es

[5]AssistantProfessor. GIDDP, Dpto. de Proyectos de Ingeniería.

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

Phone: +34 963 877000, e-mail: jofuebar@dpi.upv.es

[6]AssociateProfessor. Depto. de Ingeniería y Gestión de la Construcción.

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

Phone: +34 963 877 151, e-mail: pellicer@upv.es

Earned Schedule min-max:

Two new EVM metrics for monitoring and controlling projects

Abstract

Earned Value Management (EVM) is a well-known project management technique for monitoring project progress. Over the last 15 years, many promising EVM metrics have been proposed to get, among other improvements, betteractual project durationand cost estimates. Papers comparing the performance of all these metrics are, however,scarce and sometimes contradictory.

In this paper, asimulation and empirical comparison of 26 deterministic project duration forecasting techniques under the EVM frameworkis developed. Among them, two new metrics: Earned Schedule min (ES_{min}) and Earned Schedule max (ES_{max}) are proposed.$ES_{min}$and ES_{max}offer a new and simpler activity-level calculation approach of the traditional Earned Schedulemetric. Top performing (most accurate) metrics: Earned Schedule (*ES*), Earned Duration (*ED*) and Effective Earned Schedule (*ES(e)*)with Performance Factor 1 (*PF*=1), are slightly outperformed by the new metrics which also offer some interesting applications for enhanced project control.

Keywords: Earned Value; Earned Schedule; Earned Duration; project duration; deterministic techniques.

1. Introduction

Earned Value Management (EVM) was devised as a financial analysis tool within the United States Department of Defense in the 1960s. Since then it has become one of the most prominent techniques for monitoring project progress[1,2].

The biggest advantages of EVM are its (relative) simplicity and that it just needs the type of information (mostly activity percentages of completion and actual costs) that is gatheredfor many other purposes during the project execution stage.

Using the most recent terminology, EVM consists of three metrics named Planned Value (*PV*), Actual Cost (*AC*) and Earned Value (*EV*). *PV* represents the planned cumulative expenditure as the project progresses, that is, the planned cumulative cost. *AC* represents the actual cumulative expenditure as the project progresses (activity durations and costs will usually be different to the planned durations and costs from the *PV*).Finally, the *EV* is the cumulative expenditure of the project assuming that costs correspond to what was 'planned', but spent according to the 'actual' activity durations. *PV* and *EV* are very similar. They both represent the cumulative expenditure of the same budget (project planned cost), but the pace of that expenditure is 'as planned' for the *PV*, and 'as executed' for the *EV*[3].

Over the last 15 years, many other extensions and partial reformulations of EVM have been proposed (e.g. the Earned schedule (*ES*) [4], the p-factor [5], the Earned Duration Management (EDM) [6]) . Many of them address specific weaknesses or limitations of the EVM framework generally with the intention of better estimating the actual project duration and/or taking better proactive/corrective actions during project execution.

Also, despitemany pieces of research in the 2000s paid significant attention to the EVM cost forecasting accuracy (e.g.[7,8]), nowadays it is accepted that EVM is more accurate in the cost dimension than in the time dimension[9,10].This is probably to be expected as the project total cost mostly comes from the addition of its activity costs (whose sum converges to a Normal distribution). Whereas this is not the case of the project total duration, which is totally dependent, not just on the activity durations, but on theactivities order of execution. Thismay be a compelling reason why so many EVM extensions have been proposed over the last years trying to improve the actual project duration estimates. However, their advantages, even their actual forecasting accuracy remain uncertain, as many of these extensions have never been compared with each other and previous scientific studies have sometimes provided contradicting results.

Apart from deterministic project duration forecasting techniques, other more advanced techniques have also been proposed over the years (fuzzy logic, neural network analysis, Bayesian inference, Monte Carlo simulations, statistical learning and artificial intelligence methods, etc.) [11,12]. Deterministic techniques, despite generally less accurate, offer some advantages. They are easier to learn, and their results are generally easier to understand and communicate. The amount of information they require is also lower than non-deterministic techniques. Finally, calculations are generally quicker or at least much less computer-demanding than other alternative methods. For these reasons, deterministic project duration forecasting techniques still play a significant role in the project management practice nowadays. EVM has produced many of these techniques and this is the reason why this study focuses on those exclusively.

Hence, the main aim of this paper is toprovide numerical evidence on which deterministic EVMextensions and metrics are more accurate at predicting thereal duration of a project. For achieving this, we will resort toaset of simulated and real projects.The performance (accuracy) of the most relevant deterministic EVMextensions and metricspublishedtodate will be compared at different moments of the projects execution. Among the extensionscompared, another two new metrics based on the Earned Schedule (*ES*) metric will alsobe proposed. It will be shown how these new metrics, besides slightly outperforming the existing ones, are simpler to calculate and offer someinterestingapplications for enhanced project monitoring and control.

The paper will be structured as follows. In the literature review, the most recent and noticeable deterministic EVM-based project duration forecasting metrics will be reviewed and a representative summary of previous performance comparison studies will be provided. The materials and methods section will describe the simulated and real project datasets used to compare the forecasting methods, and the mathematical formulation of the new metrics $ES_{min}$and ES_{max}. The results section will detail the analysis and performance results of the 26 forecasting methods. The discussions will go over the weaknesses of most EVM-based methods, while alsoproposing some potential applications of $ES_{min}$and ES_{max}. Finally, the conclusions will highlight the major results and contributions, state the study limitations, and propose future research continuations.

2. Literature review

2.1.Project duration forecasting EVM extensions

For easier reference, allproject duration forecasting techniques[11]compared are presented upfront in Table 1. Project duration estimates in the EVM context have traditionally been noted as *EAC(t)* (project Estimate At Completion in time). The same terminology will be followed here. Each project duration forecasting technique includes two identifiers stated in the first two columnsof Table 1: a numerical ID, and a second referring to the metric they are based on. The mathematical expressions are displayed in the third column. Authors of every forecasting technique are stated in the fourthcolumn. A brief description of every group of methods is provided in the last column. The twonew proposed techniques are relayed to the last two rows.

[1]The terms *techniques* and *methods* are used indistinctly, and they refer to the mathematical expressions that produce a project duration estimate (in time units). The word *metric* refers to the auxiliary magnitudes that *techniques* or *methods* need to produce estimates. *Metrics* are generally the magnitudes that express the current project progress and can be expressed in either time or cost units. *Metrics* are just one of the variables in the *techniques/methods* mathematical expressions, but probably the most relevant, as they are generally the ones whose information is updated at each tracking period.

ID	Method	Expression	Author	Observations
1	*PV1*	$EAC(t)_{PV1} = PD - (EV - PV)(PD/BAC)$	(Anbari, 2004)	Methods 1 to 3 depend on a metric named Time Variance (*TV*). *TV* equals the ratio of Schedule Variance (*SV*) divided by the Planned Value rate (PV_{rate}). For simplification purposes, though, methods 1 to 3 have been directly expressed as a function of the most basic EVM metrics instead (avoiding the intermediate calculation of *TV*).
2	*PV2*	$EAC(t)_{PV2} = PD/SPI$		
3	*PV3*	$EAC(t)_{PV3} = PD/SCI$		
4	*ED1*	$EAC(t)_{ED1} = MAX(PD, AT) + AT(1 - SPI)$	(Jacob, 2003)	Methods 4 to 6 depend on an intermediate metric named Earned Duration (*ED'*). *ED'* is calculated as the multiplication of *AT* by *SPI*and has nothing to do with the *ED* metric used later in methods 14 and 15. *EAC(t)* expressions also skip the use of *ED'* and are expressed directly as a function of the most basic EVM metrics instead.
5	*ED2*	$EAC(t)_{ED2} = MAX(PD, AT)/SPI$		
6	*ED3*	$EAC(t)_{ED3} = MAX(PD, AT)/SCI + AT(1 - (1/CPI))$		
7	*ES1*	$EAC(t)_{ES1} = AT + PD - ES$	(Lipke, 2003)	Methods 7 to 9 follow the generic formula *AT*+(*PD-ES*)/*PF*and completely rely on the Earned Schedule (*ES*) metric. *ES*, unlike *TV* and *ED'*, cannot be expressed explicitly as a function of other EVM metrics, therefore this variable has been kept as is. Depending on the value of *PF* we find: method 7 (*PF*=1), 8 (*PF=SPI(t)*) or 9 (*PF=SCI(t)*).
8	*ES2*	$EAC(t)_{ES2} = AT + (PD - ES)/SPI(t)$		
9	*ES3*	$EAC(t)_{ES3} = AT + (PD - ES)/SCI(t)$		
10	*EDM1*	$EAC(t)_{EDM1} = PD - (TED - TPD)(PD/BAC(t))$	(Khamooshi & Golafshani, 2014)	Methods 10 to 15 were developed under the Earned Duration Management (EDM) framework. These methods are the counterpart of other EVM forecasting methods (namely methods 1,2, 4, 5, 7 and 8, respectively), where planned and actual activity costs are replaced by planned and actual activity durations. EDM by itself does not allow project cost forecasting unless we complement it with EVM. This is the reason why there is no EDM counterpart for EVM methods 3, 6 and 9.
11	*EDM2*	$EAC(t)_{EDM2} = PD/EDI$		
12	*EDM3*	$EAC(t)_{EDM3} = MAX(PD, AT) + AT(1 - EDI)$		
13	*EDM4*	$EAC(t)_{EDM4} = MAX(PD, AT)/EDI$		
14	*EDM5*	$EAC(t)_{EDM5} = AT + PD - ED$		
15	*EDM6*	$EAC(t)_{EDM6} = AT + (PD - ED)/DPI$		
16	*ESM1*	$EAC(t)_{ESM1} = AT + PD - ES(e)$	(Lipke, 2011)	Methods 16 to 18 are very similar to methods 7 to 9 (from the same author). These methods replace *ES* by the effective Earned Schedule (*ES(e)*). *ES(e)* is calculated the same way *ES is*, but from the fraction of *EV* that is adhered to the original schedule as measured by the p-factor (Lipke, 2004), that is, the effective Earned Value (*EV(e)*).
17	*ESM2*	$EAC(t)_{ESM2} = AT + (PD - ES(e))/SPI(t)(e)$		
18	*ESM3*	$EAC(t)_{ESM3} = AT + (PD - ES(e))/SCI(t)(e)$		
19	*ESM4*	$EAC(t)_{ESM4} = AT + (PD - ES)/SPI(t)_{\alpha=CI}$	(Elshaer, 2013)	Methods 19 to 21 are very similar to method 8. *SPI(t)* versions of these methods resort to *PV* and *EV* activity costs that come from a weighted sum. The weighting factors of each activity (planned and actual) cost are their respective Criticality Index (*CI*), Significance Index (*SI*) and Schedule Sensitivity Index (*SSI*).
20	*ESM5*	$EAC(t)_{ESM5} = AT + (PD - ES)/SPI(t)_{\alpha=SI}$		
21	*ESM6*	$EAC(t)_{ESM6} = AT + (PD - ES)/SPI(t)_{\alpha=SSI}$		
22	*XSM1*	$EAC(t)_{XSM1} = AT + (PD - ES)/T_{t,SPI(t)}$	(Khamooshi and Abdi, 2017)	Methods 22 to 24 are the counterpart of expressions 7 (methods 22 and 23) and 14 (method 24), but applying exponential smoothing techniques. These methods resort to different smoothing factors *T*.
23	*XSM2*	$EAC(t)_{XSM2} = AT + (PD - ES)(T_{t,AT}/T_{t,ES})$	(Batselier & Vanhoucke, 2017)	
24	*XSM3*	$EAC(t)_{XSM3} = AT + (PD - ED)/T_{t,EDI}$	(Khamooshi and Abdi, 2017)	
25	ES_{min}	$EAC(t)_{ES_{min}} = AT + PD - ES_{min}$	(This paper)	Methods 25 and 26 are the ones proposed in this paper and will be detailed in the *Materials and methods* section.
26	ES_{max}	$EAC(t)_{ES_{max}} = AT + PD - ES_{max}$		

Table 1. Project Duration forecasting methods (all variable names and mathematical details can be found in the *Supplemental online material*)

For the sake of clarity, every variable used in Table 1 is described along with its mathematical expression in Table S1 in the Supplemental online material. All readersare strongly encouraged to refer to that materialto look up all mathematical details and reproduce any calculation.

Quick inspection of Table 1 allows observing some evident patterns in the forecasting formulae. With some exceptions, most of them follow the generic expression *AT+(PD-ES)/PF*. *AT*confusingly stands for Actual Time and refers to the fraction of the project duration we have already consumed, that is, the current moment in time (normally associated with the last tracking period date). *PD* stands for Planned Duration and corresponds to the initial estimate of the project duration before the project started. *ES* stands for Earned Schedule, although this metric is replaced by Earned Duration metric in forecasting methods 10 to 15. *PF* is the Performance Factor that specifies at what pace the rest of the project will be executed. PF can equal 1 (which meansthe remaining duration of the project will be completed as initially planned). But it can also equal the *SPI* (Schedule Performance Index), *CPI* (Cost Performance Index), the *SCI* (Schedule Cost performance Index), and many other. All these possible *PF*values determine different speeds of execution depending on: the current schedule progress (*PF=SPI*), cost expenditure(*PF=CPI*) or schedule & cost combined (*PF=SCI*).

A last note is made about methods 19 to 21. These methods resort to the Criticality Index (*CI*), the Significance Index (*SI*) and the Schedule Sensitivity Index (*SSI*)which are not, strictly speaking, deterministic variables. They are actually obtained by (Monte Carlo) simulation in what is called Schedule Risk Analysis (SRA)[1]. Elshaer[14] also proposed using the Cruciality Factors based on Pearson's r, Spearman's $\rho$and Kendall's τcoefficients of correlation. However, Elshaer himself proved that the performance of those three forecasting methods was worse than the three presented here. Besides, Elshaer never detailed what to do when negative values of r, ρor τarise, a common situation when these variables are calculated by a limited number of simulations. For both reasons, only the three methods displayed as 19 to 21 have been compared here.

2.2.Previous studies comparing EVM metrics performance

Performance analyses comparing these metrics have been in short supply. The first was an exhaustive and extensive simulation study whose highlights can be found in Vanhoucke [17]. This study compared the project time performance of EVM in combination with SRA metrics, while also considering schedule networks topology. The main aim of this study was to validate the (by then) current methods to improve the corrective actions decision-making process during the project control stage considering project duration forecast accuracy. Methods 1 to 9 were compared being method 7 the top performer. The same set of 4100 network schedules generated for that study will be used here.

Elshaer[14] proposed merging EVM and SRA metrics resulting in the comparison of methods19 to 21. He used the same 4100 simulated projects dataset for comparison purposes. This study also included method 7 as benchmark, but apparently,method 19 proved to be more accurate.

Batselier and Vanhoucke[18] performedanother comparison involving methods 1 to 9, but this time with 23 real projects data instead of simulated projects. Method 7 was again the top performer. The same real project dataset will also be used later in this study.

Another study by Batselier and Vanhoucke [19] compared three new project duration forecasting techniques, separately and in combination with each other with the same simulated and real project datasets. Particularly, techniques 7, 15, 16 and 19 were compared and, method 7 was the most accurate (when it should have been method 19 according to Elshaer[14]).

Khamooshi and Abdi [15]compared methods 10 to 13, plus methods 22 and 24 on a different 19-project real dataset; results suggested that techniques 22 and 24, which included double linear exponential smoothing, were the best.

Batselier and Vanhoucke [16] suggested a new EVM metric with exponential smoothing (method 23) and compared it against method 7 obtaining a marginal accuracy improvement. However, the new proposed method came at the expense of adding a new (subjective) parameter whose calibration may not be possible in all project contexts.

De Andrade and Vanhoucke [20] compared methods 7 and 15, plus a combination of these, on a 14-project subset of the 23-project real dataset used by Batselier and Vanhoucke [18]. Results showed that method 7 was again the top performing and the combination of both methods did not seem to provide substantial advantages.

There have been many other studies suggesting new EVM metrics with interesting properties but whose benefits are difficult to generalize (or even implement) with the information that is gathered under the EVM framework. Representative examples of these may be Earned Incentive (EIM) [21] for projects that use time and/or cost incentives; or the mean lags metric [22] for a better measurement of the EVM metrics accuracy versus stability forenhanced project duration and cost forecasting. Also, Picornell et al. [23]proposed a new formulation focused on projects whose paymentsare based on unit-prices. However, these variants will not be considered in this study as their purpose substantially differ from the EVM metrics compared later.

As a conclusion, given the recent proliferation of EVM techniques and metrics it seems necessary to test all of them with the same benchmark (simulated and real)project datasets to identify which ones are better or, at least, under what conditions some of them perform better. This is the first major aim of the present paper. The second aim will be to propose two new metrics ($ES_{min}$and ES_{max}) and discuss their advantages.

3. Materials and methods

3.1.Simulated projects dataset

The simulated projects dataset consists of 4100 activity-on-node networks with 30 activities each plus two dummy activities (zero duration and cost) signaling the start and end of each project. This dataset is curated online by the University of Ghent's Operations Research &

Scheduling Research Group and is accessible here https://bit.ly/2OYl34Q along with other project datasets. Project information basically comprises the activity (deterministic) durations and the predecessors information. No resource information was used in this study.

This 4100-project dataset was generated using the RanGen2 algorithm. RanGen2 is a robust random network generator validated in recent studies [24,25] and capable of generating a wide range of different network topologies.The same set projects has also been used in many recent research studies on EVM (e.g. [26–28]).

The projects of this dataset were generated under pre-set values of four topological indicators:the serial-Parallel (*SP*), the Activity Distribution (*AD*), the Length of Arcs (*LA*), and the Topological Float (*TF*).The SP indicator describes how close a network is to a serial or parallel network. The *AD* describes the distribution of activities in its different network paths. The*LA* measures the distance between two activities in the project network. The *TF* measures the slack or float that each activity has at a topological level, that is, how dense the network is. All indicators range from 0% to 100%.These four topological indicators were initially proposed by Vanhoucke et al. [25] and slightly refined in Vanhoucke [17].They are considered representative and accurate descriptors of a network topology.

Another two network complexity indicator values have been provided for each project instance for comparison purposes: the Coefficient of Network Complexity (*CNC*) [29] and the Order Strength (*OS*) [30]. The values of all six indicators can be found for all network instances along with the performance results as *Supplemental online material*.More precisely, the 4100 project network topologies were generated by setting specific staggered values of the *SP*indicator from *SP*=0 (all project activities are in parallel) to SP=100% (all activities are in series). While the *SP* was set, the other indicators (*AD*, *LA* and *TF*) could vary freely when searching new random network configurations. Namely, the following series of *SP* values were used: 13%, 23%, 32%, 42%, 52%, 61%, 71%, 81%, and 90%. Extremes (0% and 100%) were not included in the analyses as they are not considered representative of real projects.

For the interested reader, this series of values was adopted (instead of a *SP* series with constant 10% intervals like 10%, 20%, 30% and so on) becauseof the total number of activities per project. It was mentioned that each project included 30 activities plus two dummy activities. In the analyses shown later, however, dummy activities were also given stochastic durations and costs different from zero. This allowed increasing the number of activities per project and marginally enhancing their representativity ('slightly bigger' projects). Hence, with 32 activities per project,rounded *SP* values of 10%, 20%, 30%, etc. were just not mathematically possible.

Another couple of substantial changes were implemented in the default artificial project dataset. These changes did no longer condition the network topology generation, but the generation of activity durations and costs.

Concerning the stochastic generation of activity durations, most studies referenced earlier resorted to triangular distributions. Triangular distributions are upper- and lower-bounded distributions whose limits are somehow subjectively set (normally by multipliers of the value used as the distribution mode). This approach may be too restraining at times. It is not uncommon that activities from real projects are significantly shortened, or much more frequently, lengthened, 2to 10 times their expected (planned) duration. It is also common that the differences among different (planned) activity durations alsoexceed those proportions. Capturing thosedifferent orders of magnitude cannot be effectively achieved with triangular distributions. This, as such spread triangles would end up resembling uniform distributions, rather than triangular distributions.

Instead, log-Normal distributions have been used here. Log-Normal distributions automatically exclude the possibility of negative durations, allow (if required) a higher concentration of values around the mean (or mode) and depend onjust two parameters, instead of three. Log-Normal distributions are quite simple and are effective at occasionally letting some activities take significantly higher or lower duration values. Finally, empirical studies have shown that this distribution models construction activity duration variability quite satisfactorily in the case of construction projects [31,32].

Therefore, athree-stage process was adopted for generating the activity durations. First, the ‘mean’ durationof each activity *i* (noted here as m_{di}) was stochastically generated by Monte Carlo simulation from a Log-Normal distribution with meanμ=0.5 and standard deviation σ=0.25, that is m_{di}~$10^{\text{Normal}(\mu=0.5,\ \sigma=0.25)}$. This, as we used logarithms with base 10 and the log-Normally distributed values are generated by calculating the antilogarithm. Second, the ‘coefficient of variation’(the ratio of the standard deviation and the mean)of each activity duration (CV_{di})was generated by simulation with a Uniform distribution ranging between 0.1 and 0.3, that is, CV_{di}~Uniform(a=0.1, b=0.3).In the latter expression, a and b are the Uniform distribution lower and upper bounds, respectively. At this stage, we have already created a series of mean and standard deviation duration values($m_{di}$and $s_{di}=CV_{di} \cdot m_{di}$, respectively)for each activity *i*. Particularly, the means represent the ‘planed’ durations (d_i)of each activityduring the (simulated) project execution, that is, $d_i=m_{di}$. The Third stage consists of generating a stochastic duration valuefor each activity(d'_i)around the previously log-Normally-generated m_{di} by using the expressiond'_i ~Normal(μ=m_{di},σ=s_{di}). Those areconsideredthe ‘actual’ activity durations during project execution.

The approach taken above allows generating a set of planned durations that sometimes have significant differences among activities. How often that happens can be easily controlled with the standard deviation values(s_{di})which are constantly changing for each project instance. The second advantage is that, once we have a set of activity durationswith sufficiently (but not excessively) spread durations, the actual (d'_i)versus the planned(d_i) durations can also differ substantially, that is, unlike triangular distributions, those values can be far from the mode at times. This is more challenging for the project duration forecasting methods compared.

A similar three-stepprocess was alsofollowed for the stochastic generation of activity costs.First, the Log-Normal distribution for generating the activity cost ‘means’(m_{ci})changed the prior activity durationmoments from μ=0.5 and σ=0.25to μ=4 and σ=0.5, that is

m_{ci}~$10^{\text{Normal}(\mu=4,\,\sigma=0.5)}$. The bigger μ and σ values herereflect that activity costs (in money units) are generally bigger magnitudes than activity durations (in time units). Similarly,this set of 'mean' costsequaled the 'planned' cost of each project activity, that is, c_i=m_{ci}. The second step (stochastic cost Coefficient of Variation generation) also followed a Uniform distribution from 0.1 to 0.3, that is, CV_{ci}~Uniform(a=0.1, b=0.3). Finally, the set of stochastic 'actual'costs (c'_i)for each activity*i*was generated from the two previous moments(m_{ci} and s_{ci} =$CV_{ci} \cdot m_{ci}$). However, in this generation step,a correlation coefficientρ_i between each activity duration and cost was introduced. Values of ρ_i were set to vary uniformly for each activity between 0 and 0.25, that is, ρ_i~Uniform(a=0, b=0.25).With this information, the specific expression for generating the c'_i values was $c'_i \sim \rho_i\, c_i \dfrac{d'_i}{d_i} + \sqrt{1-\rho_i^2} \cdot Normal\left(\mu = c_i, \sigma = s_{ci}\right)$.

The inclusion of ρ_i constitutes a significant addition versus previous comparison analyses too. It is obvious that most activities cost more when they last longer. Furthermore, among the project duration forecasting methods, there are some (like the Earned Duration Method (EDM)) which replace activity costs by activity durations. If both sets of values vary independently from each other, the effectiveness of all EDM-based methods could not be properly tested.The reader is referred to the*Supplemental online material* for further mathematical details.

3.2. Real projects dataset

The same research group at Ghent University (Belgium) curates a dataset comprising 125 real projects. The dataset is accessible here https://bit.ly/2Mi8mmE. The dataset was originally made public in two papers published by Batselier and Vanhoucke [33] and Vanhoucke et al. [34]. This dataset is also used regularly nowadays by other construction researchers in the area of scheduling (e.g.[2,32]). Projects encompass building, civil engineering, industrial and services, but most of them are construction-related. The country of origin is mostly Belgium (when the non-existence of aconfidentiality agreementallowed this piece of information to be disclosed). However, there are also projects from the Netherlands, Italy, USA and Azerbaijan.

Of those 125 projects, unfortunately, only 23 contain tracking informationto allow the application of EVM techniques. This tracking information includes the baseline schedule, activity percentages of completion and theactual duration and costs at different project stages. Those 23 projects are listed with their main characteristics in Table 2. For easier cross-reference, project ID codes on the first column have kept the original database codes. For further information, the reader is referred to Batselier and Vanhoucke [33].

ProjectID (m)	Project Name	Project Type	Planned Cost (€)	Actual Cost (€)	Planned Dur. (d)	Actual Dur. (d)	Nºactiv. (total)	Track. periods	SP (%)	AD (%)	LA (%)	TF (%)
C2011-05	Telecom System Agnes	Service	180,485.27	180,485.27	43	53	22	5	60	58	38	9
C2011-07	Patient Transport System	Service	180,759.44	191,065.06	389	444	69	23	70	70	7	8
C2011-12	Claeys-Verhelst Premises	Building	3,027,133.19	3,102,395.91	443	453	59	8	41	50	5	43
C2011-13	Wind Farm	Civil Eng.	21,369,835.51	26,077,764.74	525	600	167	120	27	36	0	48
C2012-13	Pumping Station Jabbeke	Industrial	336,410.15	350,511.31	125	140	75	28	64	59	3	27
C2013-01	Wiedauwkaai Fenders	Civil Eng.	1,069,532.42	1,314,584.58	152	152	49	6	48	45	0	68
C2013-02	Sewage Plant Hove	Civil Eng.	1,236,603.66	1,146,444.38	403	408	221	17	12	38	0	62
C2013-03	Brussels Finance Tower	Building	15,440,865.89	16,338,027.20	425	426	63	18	3	82	0	87
C2013-04	Kitchen Tower Anderlecht	Building	2,113,684.00	2,512,524.00	333	453	272	11	47	59	0	63
C2013-06	Govmnt. Office Building	Building	19,429,810.51	21,546,846.18	352	344	300	18	10	36	0	34
C2013-07	Family Residence	Building	180,476.47	175,030.65	170	174	63	11	40	44	3	25
C2013-08	Timber House	Building	501,029.51	576,624.05	216	235	53	13	29	42	0	47
C2013-09	Urban Develop.Project	Civil Eng.	1,537,398.51	1,696,971.79	291	360	72	10	34	51	6	16
C2013-10	Town Square	Civil Eng.	11,421,890.36	15,218,926.38	786	785	273	30	18	36	0	62
C2013-11	Recreation Complex	Building	5,480,518.91	5,451,028.00	359	277	209	20	27	44	0	32
C2013-12	Young Cattle Barn	Building	818,439.99	879,853.17	115	188	30	5	64	77	6	54
C2013-13	Office Finish. Works (1)	Building	1,118,496.59	955,929.22	236	217	12	9	20	49	33	6
C2013-15	Office Finish. Works (3)	Building	341,468.11	308,343.78	171	115	18	3	25	43	21	35
C2014-04	Compres. Station Zelzate	Industrial	62,385,597.58	65,526,930.04	522	844	25	36	95	100	0	100
C2014-05	Apartment Building (1)	Building	532,410.29	591,410.53	228	274	26	13	58	71	35	18
C2014-06	Apartment Building (2)	Building	3,486,375.47	3,599,114.11	547	611	30	19	57	75	46	15
C2014-07	Apartment Building (3)	Building	1,102,536.78	1,289,696.78	353	404	26	14	58	71	35	18
C2014-08	Apartment Building (4)	Building	1,992,222.09	2,380,299.86	233	275	43	13	44	29	11	14

Table 2. Real projects dataset

Some of these projects required substantial editing before they could be used in later analyses. Unlike previous project duration forecasting techniques, the two new metrics proposed in the next subsection, require some basic baseline activity information (mostly activity planned start dates and slacks). Some of these projects did not reflect the correct activity slacks as their activities had been partiallyshifted. That meant the plannedactivity start dates did not correspond to either the as soon as possible (ASAP) schedule, nor the as late as possible schedule (ALAP). Actual durations and costs, nor the actual start and finish dates were altered, though.

3.3. The ES_{min} and ES_{max} metrics

Besides comparing 24 deterministic EVM-based project duration forecasting methods, later analyses also include two new forecasting methods presented in this study for the first time. These two methods correspond to methods 25 and 26 in Table 1 and rely, respectively, on two new metrics named Earned Schedule min (ES_{min}) and Earned Schedule max (ES_{max}). Their mathematical expressions are:

$$EAC(t)_{ES_{min}} = AT + PD - ES_{min} \tag{1}$$

$$EAC(t)_{ES_{max}} = AT + PD - ES_{max} \tag{2}$$

From expressions (1) and (2), it is easy to appreciate that both methods share the same forecasting approach and the performance factor equals 1 (the value that would have divided the 'PD-ES_{min}' or 'PD-ES_{max}' terms). AT, as described earlier, correspond to the Actual Time (current duration elapsed since the project started, normally assumed as the date of the last tracking period).PD is the project planned duration (in time units).

Basically, what ES_{min} and ES_{max} do is measuring the project progress of its most advanced and delayed paths, respectively. For calculating ES_{min} and ES_{max}, it is necessary to calculate beforehand the Earned Schedule value of each activity i (noted as ES_i) at the current tracking period AT.ES_i differs from the classical Earned Schedule (ES) metric on the fact that they are calculated at the activity level, not atthe project level. However, this calculation is quite straightforward:

$$ES_i = SD_i + PC_i \cdot d_i \quad (3)$$

Where SD_i is activity *i*'s (earliest) planned start date;PC_i is activity *i*'s percentage of completion at (current) tracking period *AT*; and d_i corresponds to activity *i*'s planned duration. For expression (3) to work accurately, both SD_i and d_i magnitudes have to be expressed in working days, not in calendar days.

Known (3) for every activity at a particular*AT*, ES_{min} and ES_{max} can be calculated as follows from the basic baseline schedule information:

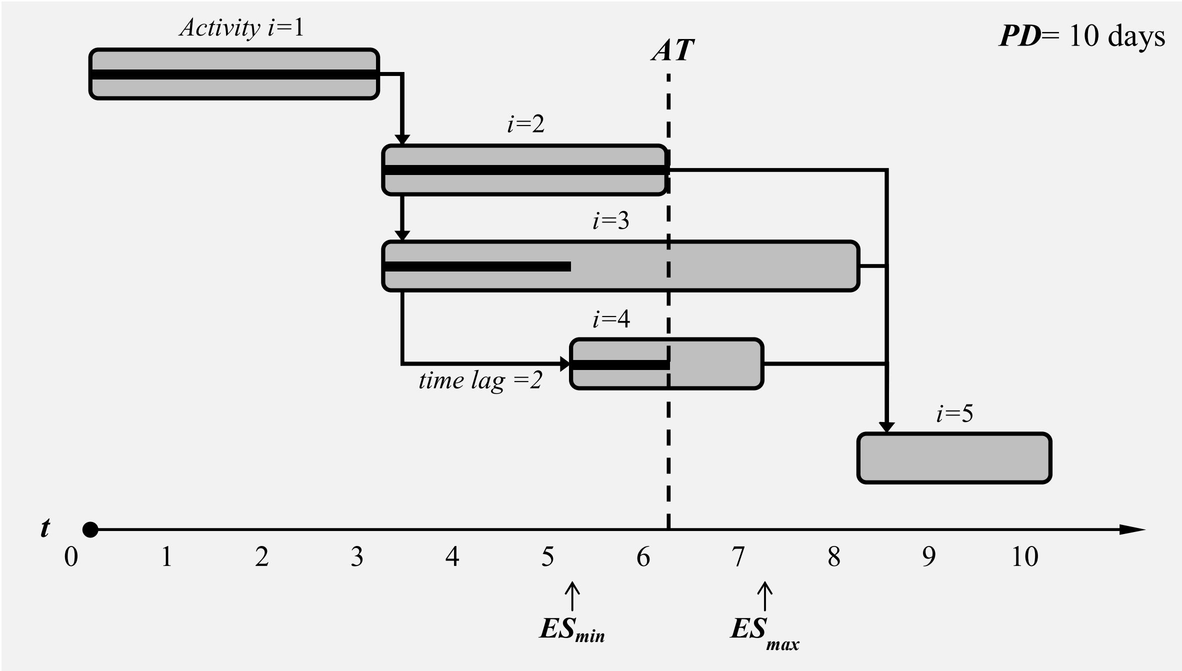
$$ES_{min} = MIN\left\{ES_i + s_i : PC_i \subset [0,1), i \in n\right\} \quad (4)$$

$$ES_{max} = MAX\left\{ES_i : PC_i \subset (0,1], i \in n\right\} \quad (5)$$

Where s_i is activity *i*'s (baseline) slack, while*n* denotes the set of all activities in the project.

Expression (4) denotes that ES_{min} is calculated as the minimum $ES_i$of all*unfinished* activities, that is, those activitieswhose percentage of completion range from 0% (included) to just below 100% (not included). The inclusion of s_i in expression (4) means that what expression (4) is actually doing, is using activities' latest start dates, that is, the ALAP schedule.

ES_{max} is even easier to calculate. Namely, ES_{max} is calculated as the maximum ES_i value of those activities which have already started. This time, activities that have a percentage of completion of 0% are not considered.To illustrate the simple calculations that these metrics require, a simple numerical example is provided in Figure 1. All data is reproduced graphically at the top and numerically at the bottom.



Activity (i)	***Predecessors***	SD_i(days)	d_i (days)	s_i (days)	PC_i(%)	ES_i(days)	ES_{min} (days)	ES_{max} (days)
1	-	0	3	0	1.00	3	-	3
2	FS1	3	3	2	1.00	6	-	6
3	FS1	3	5	0	0.40	5	5+0	5
4	FS1+2d	5	2	1	1.00	7	-	7
5	FS2,FS3,FS4	8	2	0	0.00	8	8+0	-
							$\boldsymbol{ES_{min}=5}$	$\boldsymbol{ES_{max}=7}$

Figure 1.ES_{min}&ES_{max} calculation schematic

Figure 1 depicts a project with 5 activities that was planned to last 10 days (or another time unit). The current date is AT= 6, that is, six days after the project started. Activity (planned) durations are represented by grey bars. The actual durations of all activities are represented by the thick black line inside each activity bar. Additionally, activity 4 had (from the baseline schedule) a compulsory time lag, which keeps her from starting earlier. As of AT’s date,we can see in the table below that Activities 1, 2 and 4 are completed (PC_1=PC_2= PC_4=100%). Activity 3 is completed at 40% (PC_3=40%), and activity 5 has not started yet(PC_5=0%).

With all this information, activity planned start dates (SD_i) and slacks (or floats) are very easy to calculate. The only non-critical activities are activities 2 and 3, whose slacks are 2 and 1 days, respectively. These variables could have also been calculated fromany scheduling software.

The next step consists of calculating all activities' ES_i values with expression (3). This is represented in the last but twocolumn. Finally, ES_{min} is calculated with expression (4) among those activities which are not complete andincluding those that have not even started ($0\% \leq PC_i < 100\%$). Analogously, ES_{max} is calculated with expression (5) among those activities which have started, no matter they are finished or not ($0\% < PC_i \leq 100\%$). Both calculations results in ES_{min}=5 days and ES_{max}=7 days.

The reader will have appreciated now that what expressions (4) and (5) areactually doing isjust calculating the equivalent (planned) date of progress (on the baseline schedule)of the most delayed and most advanced paths.This may seem not sophisticated at all. However, it will be seen later how these metrics outperform the rest in almost all project datasets.

4. Results

The error magnitude chosen to measure the deviations between each method's forecasted project duration at tracking period AT (generically referred to for every method as $EAC(t)_{AT}$), and the actual project$m$'s duration(generically referred as RD_m below, or as Real Duration RD, in the tables) is the Absolute Percentage Error (APE_{AT}). $APE_{AT}$is calculated for every project $m$and method at every tracking periodATas:

$$APE_{AT} = \frac{|RD_m - EAC(t)_{AT}|}{RD_m} \qquad (6)$$

The detailed4100-projectAPE_{AT}values can be found in the *Supplemental online material*.Subsequent tables will only present aggregatedresults, that is, average results either by tracking periodAT, project$m$ or both.

In the case of the simulated projects, when averaging the APE_{AT}results for all projects at the same tracking period, expression (6) will become the Mean Absolute Percentage Error (MAPE) at AT, that is:

$$MAPE_{AT} = \frac{1}{M}\sum_{m=1}^{M}\frac{|RD_m - EAC(t)_{AT}|}{RD_m} \quad (7)$$

Where m denotes each project (m=1,2,…, 4100 for the simulated dataset)and$M$ denotes the total number of projects, that is 4100.AT in expression (7) will mean specific homogeneous times of progress across all projects. In this study, we assume that AT= 0%, 10%, 20%, …, 90% and 100% of RD_m (the Real Duration of project m).

Averaging the results at the same AT with expression (7) is not possible in the real projects dataset due to their occasional extremely low number of tracking periods. For this dataset, the APEresults are presented averaged by project. When averaging the results for the same project m for all its$N$tracking periods, expression (6) becomes:

$$MAPE_m = \frac{1}{N}\sum_{AT=0}^{N} APE_{AT} \quad (8)$$

Particularly,AT=0 denotes the moment just before the project starts. Without exception, all forecasting methods at that moment assume that $EAC(t)_{AT=0=0\%RD} = PD$, that is, the Planned Duration. On the other hand, AT=$N$is assumed here to coincide with $\lfloor RD \rfloor$(the nearest rounded down integer of the Real duration of project m). That is the moment of time when there is exactly less than one day left to complete the project (the project will finish at some point the day after). This measurement is interesting as it allows detecting those forecasting methods that are near-sighted, that is, those incapable of providing with good project duration estimates,no matter the project is about to finish.

Finally, some Figures will also present in their last column the average results for all projects and tracking periods altogether. We will refer to these generically as the $MAPE$ values:

$$MAPE = \frac{1}{N}\sum_{AT=0}^{N} MAPE_{AT} = \frac{1}{M}\sum_{m=1}^{M} MAPE_m \quad (9)$$

4.1. Simulated projects performance results

Hence, for the 4100 different network instances (topologies), stochastic activity durations and (partially correlated) costs were generated. $MAPE_{AT}$ results are shown first in Figure 2 for the 26 forecasting methods (by rows) at different moments of project progress (by columns). Top performing (those with lower overall $MAPE_{AT}$ values) methods are highlighted in bold text. Among them, we can find method 7, one of the top performing methods in almost all previous comparison studies. But also method 14 –the equivalent of method 7 in the EDM framework–which had not been compared to date.

ID	Method	% *RD* 0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	Overall
1	*PV1*	0.053	0.055	0.054	0.053	0.051	0.049	0.046	0.044	0.043	0.040	0.043	0.048
2	*PV2*	0.053	0.353	0.226	0.164	0.150	0.084	0.076	0.052	0.047	0.042	0.043	0.117
3	*PV3*	0.053	0.403	0.257	0.189	0.201	0.109	0.100	0.073	0.069	0.064	0.063	0.144
4	*ED1*	0.053	0.085	0.073	0.059	0.053	0.049	0.048	0.046	0.044	0.040	0.022	0.052
5	*ED2*	0.053	0.353	0.226	0.164	0.150	0.084	0.076	0.052	0.047	0.040	0.021	0.115
6	*ED3*	0.053	0.397	0.248	0.177	0.186	0.093	0.082	0.054	0.048	0.040	0.021	0.127
7	***ES1***	**0.053**	**0.051**	**0.048**	**0.046**	**0.042**	**0.038**	**0.034**	**0.029**	**0.023**	**0.014**	**0.003**	**0.035**
8	*ES2*	0.053	0.118	0.089	0.074	0.062	0.053	0.043	0.035	0.026	0.015	0.003	0.052
9	*ES3*	0.053	0.158	0.111	0.089	0.073	0.060	0.049	0.038	0.028	0.015	0.003	0.062
10	*EDM1*	0.053	0.053	0.050	0.048	0.044	0.040	0.038	0.034	0.031	0.029	0.038	0.042
11	*EDM2*	0.053	0.182	0.122	0.087	0.065	0.054	0.046	0.038	0.032	0.029	0.038	0.068
12	*EDM3*	0.053	0.055	0.053	0.048	0.044	0.041	0.038	0.035	0.032	0.028	0.015	0.040
13	*EDM4*	0.053	0.182	0.122	0.087	0.065	0.054	0.046	0.038	0.032	0.026	0.014	0.065
14	***EDM5***	**0.053**	**0.051**	**0.048**	**0.045**	**0.042**	**0.037**	**0.033**	**0.028**	**0.022**	**0.013**	**0.003**	**0.034**
15	*EDM6*	0.053	0.115	0.087	0.071	0.060	0.050	0.042	0.034	0.024	0.014	0.003	0.050
16	***ESM1***	**0.053**	**0.051**	**0.048**	**0.046**	**0.042**	**0.038**	**0.034**	**0.029**	**0.023**	**0.014**	**0.003**	**0.035**
17	*ESM2*	0.053	0.119	0.090	0.074	0.063	0.053	0.044	0.035	0.026	0.015	0.003	0.052
18	*ESM3*	0.053	0.160	0.113	0.090	0.074	0.061	0.049	0.039	0.028	0.015	0.003	0.062
19	*ESM4*	0.053	0.102	0.088	0.074	0.062	0.052	0.044	0.035	0.026	0.015	0.003	0.050
20	*ESM5*	0.053	0.059	0.061	0.058	0.054	0.048	0.042	0.035	0.026	0.015	0.003	0.041
21	*ESM6*	0.053	0.130	0.111	0.086	0.066	0.053	0.042	0.033	0.024	0.013	0.003	0.056
22	*XSM1*	0.053	0.118	0.090	0.073	0.060	0.050	0.042	0.034	0.025	0.015	0.003	0.051
23	*XSM2*	0.053	0.117	0.089	0.072	0.061	0.052	0.043	0.035	0.025	0.015	0.003	0.051
24	*XSM3*	0.053	0.118	0.090	0.072	0.060	0.050	0.042	0.034	0.025	0.014	0.003	0.051
25	**ES_{min}**	**0.053**	**0.050**	**0.047**	**0.043**	**0.039**	**0.035**	**0.031**	**0.026**	**0.020**	**0.012**	**0.002**	**0.032**
26	**ES_{max}**	**0.053**	**0.052**	**0.051**	**0.049**	**0.046**	**0.041**	**0.036**	**0.030**	**0.023**	**0.013**	**0.002**	**0.036**

Figure 2. The 26 deterministic project duration forecasting methods' $MAPE_{AT}$ values on 4100 simulated projects by percentage of completion (top five performing methods in bold text)

Methods 22 to 24 in Figure 2resort to exponential smoothing. These three methods can be adjusted as a function of a single parameter named exponential smoothing constant. Particularly, method 22's and 24's smoothing constant is namedβ(see Table S1in the*Supplemental online material*). Its value, following the authors' recommendation, equaled 0.25. Method 23 resorted to a smoothing constant namedγ in Table S1. On the authors' recommendation, γ equaled 0.05 in all instances. Probably it goes without saying that, should these parameters had been allowed to vary dynamically during the project duration, their respective methods would have performed better. However, allowing this possibility does not seem fair to the other methods, mostly when the adjustment (calibration) of these smoothing constants is not easy (if possible) before the Real project Duration is known.With thesepremises, the red (denoting higher $MAPE_{AT}$ values) to green (denoting lower $MAPE_{AT}$ values) color gradient fromFigure 2easily allows identifying the top performing methods.

The difference between the first (method 25 or *EAC(t)* using ES_{min}) and second best (method 14 or *EAC(t)* using *ED*) methods seems very small. But is worth noticing how the former dominates all methods at all stages of project progress (lowest $MAPE_{AT}$ values from *RD*=0% up to 100%). This makes us believe that method 25 really performs better than any other, at least on average.

On the worst performing side of the spectrum, we can find methods 2, 3, 5 and 6, whose overall performance values (last column of Figure 2)greatly exceedthe default $MAPE_{AT=0\%}$= 0.053 (the same for all methods). This 0.053value is indicative of the average project duration variability imposed by the distributions and parameter values used in these simulations.Additionally, methods 1, 2, 3, 10 and 11 are clearly near-sighted, astheir $MAPE_{AT=100\%RD}$ values are comparatively very high.

Another interesting result from Figure 2 is that,with the exception of method 16 (which resorts to *ES(e)*), the top five performing methods are actually among the simplest. This raises the question about whether all the complexities added recently to the EVM frameworkwerereally necessary. Agoodexample can befound among the top performing methods 7 (depending on *ES*) and 16 (depending on *ES(e)*). Both methods were proposed by Lipke in 2003 and 2011, respectively, but despite the latter is mathematically much more complex than the former, they have performedalmost exactly the same in our dataset. Another example may be methods 7 and 14, which are the counterparts of the EVM and EDM frameworks.

Finally, it is worth highlighting that the best (method 25) and fifth best (method 26) correspond to the new ones proposed in this study. Both methods, as illustrated earlier, are extremely easy to calculate. Indeed, they may be the easiest of the 26. Also, despite method 26 (depending on ES_{max}) is the fifth now, it will outperform the rest when comparing real projects.

The same set of 4100 projects were also arranged by their Serial-Parallel (*SP*) values. This indicator describes how close a project network is to a project with all activities in parallel (*SP*=0%) or in series (*SP*=100%).Also, from this indicator one can indirectly infer the minimum number of paths that a project has.Acknowledging this, *MAPE*results for the 26 methods performance by *SP* value are shown in Figure 3.

		Parallel networks								Serial networks	
	SP=	13%	23%	32%	42%	52%	61%	71%	81%	90%	
ID	Method↓ M=	100	1300	100	100	1300	100	100	900	100	Overall
1	*PV1*	0.072	0.063	0.055	0.044	0.043	0.040	0.039	0.035	0.034	0.047
2	*PV2*	0.141	0.196	0.202	0.071	0.086	0.061	0.063	0.060	0.066	0.105
3	*PV3*	0.160	0.222	0.221	0.093	0.112	0.088	0.092	0.089	0.100	0.131
4	*ED1*	0.109	0.071	0.055	0.042	0.043	0.049	0.040	0.035	0.034	0.053
5	*ED2*	0.138	0.193	0.200	0.070	0.084	0.060	0.061	0.058	0.065	0.103
6	*ED3*	0.145	0.207	0.209	0.079	0.095	0.070	0.073	0.071	0.079	0.114
7	***ES1***	**0.053**	**0.045**	**0.042**	**0.033**	**0.031**	**0.029**	**0.028**	**0.025**	**0.024**	**0.034**
8	*ES2*	0.070	0.065	0.057	0.048	0.048	0.045	0.042	0.040	0.035	0.050
9	*ES3*	0.076	0.073	0.065	0.056	0.058	0.055	0.053	0.051	0.051	0.060
10	*EDM1*	0.069	0.057	0.049	0.039	0.036	0.034	0.031	0.027	0.026	0.041
11	*EDM2*	0.111	0.099	0.080	0.055	0.057	0.050	0.045	0.042	0.037	0.064
12	*EDM3*	0.071	0.055	0.048	0.038	0.035	0.032	0.030	0.026	0.024	0.040
13	*EDM4*	0.107	0.095	0.078	0.053	0.055	0.048	0.044	0.040	0.036	0.062
14	***EDM5***	**0.051**	**0.044**	**0.041**	**0.033**	**0.030**	**0.029**	**0.028**	**0.025**	**0.024**	**0.034**
15	*EDM6*	0.066	0.062	0.055	0.047	0.047	0.045	0.042	0.040	0.035	0.049
16	***ESM1***	**0.052**	**0.045**	**0.042**	**0.033**	**0.031**	**0.029**	**0.028**	**0.025**	**0.024**	**0.034**
17	*ESM2*	0.070	0.066	0.058	0.049	0.049	0.046	0.043	0.040	0.035	0.051
18	*ESM3*	0.076	0.074	0.066	0.057	0.058	0.055	0.054	0.052	0.052	0.060
19	*ESM4*	0.065	0.061	0.055	0.047	0.048	0.046	0.043	0.040	0.036	0.049
20	*ESM5*	0.056	0.050	0.046	0.039	0.038	0.036	0.035	0.034	0.031	0.041
21	*ESM6*	0.078	0.073	0.064	0.049	0.051	0.047	0.044	0.040	0.036	0.054
22	*XSM1*	0.069	0.063	0.056	0.048	0.048	0.045	0.042	0.040	0.035	0.050
23	*XSM2*	0.069	0.064	0.056	0.048	0.048	0.045	0.042	0.040	0.035	0.050
24	*XSM3*	0.069	0.063	0.056	0.048	0.048	0.045	0.042	0.040	0.035	0.050
25	$\boldsymbol{ES_{min}}$	**0.045**	**0.040**	**0.039**	**0.032**	**0.030**	**0.029**	**0.028**	**0.025**	**0.023**	**0.032**
26	$\boldsymbol{ES_{max}}$	**0.064**	**0.049**	**0.043**	**0.035**	**0.031**	**0.029**	**0.027**	**0.025**	**0.023**	**0.036**

Figure 3. The 26 deterministic project duration forecasting methods' *MAPE* values on 4100 simulated projects by Serial-Parallel (*SP*) values (top five performing methods in bold text)

Figure 3 shows near the topthe values of*SP* and*M.SP* values range, as described earlier approximately from 0% to 100%, without including the extremes for representativity purposes. *M*representsthe number of projects that were used to compute each column (out of the total 4100 project instances). Despite the obviously uneven distribution of projects in some *SP* values, all columns seem to have enough sample size to draw representative average results.

The major differences between Figures 2 and 3 are that the former second, third and fourth best methods are all even now (with an average *MAPE* of 0.034). The former best and fifth best methods (the two new ones proposed) keep their relative positions.Particularly, method 25 (the best) still dominates all methods in all columns.

Finally, all methods generally perform better (suffer from lower errors) towards serial networks, rather than parallel networks. This means that anticipating the duration of projects with more activities in parallel is more challenging than in serial projects. This is the result of the ‘merge event bias’phenomenon [35,36] which indirectly describes the increasing possibility of one path falling behind (underperforming respect to its planned work) as the number of paths increases.

4.2.Real projects performance results

All project duration forecasting methods were again compared in the real dataset consisting of 23 projects. Results are shown in Figure 4. However, as results are displayed in this occasion by project, for the sake of clarity, only the top performing methodsare displayed. Coincidentally, these top five performing methods are the same top five performing methods from the simulated projects (methods 7, 14, 16, 25 and 26). This provides reassurance on the robustness of these methods.

Project ID (m)	*ES1*	*EDM5*	*ESM1*	*XSM2*	γ	ES_{min}	ES_{max}	$ES_{min\text{-}max}$	δ
C2011-05	0.140	0.137	0.131	0.109	0.995	0.129	0.148	0.129	0.000
C2011-07	0.075	0.064	0.073	0.075	0.000	0.090	0.070	0.070	1.000
C2011-12	0.031	0.031	0.033	0.031	0.000	0.037	0.030	0.016	0.473
C2011-13	0.077	0.076	0.080	0.076	0.015	0.080	0.112	0.080	0.000
C2012-13	0.079	0.080	0.078	0.079	0.000	0.075	0.091	0.091	0.000
C2013-01	0.074	0.049	0.084	0.074	0.000	0.086	0.104	0.086	0.000
C2013-02	0.049	0.081	0.052	0.049	0.000	0.015	0.008	0.008	0.902
C2013-03	0.040	0.057	0.048	0.040	0.000	0.064	0.073	0.002	0.510
C2013-04	0.080	0.388	0.085	0.080	0.000	0.098	0.107	0.098	0.000
C2013-06	0.142	0.234	0.177	0.142	0.000	0.072	0.037	0.036	0.793
C2013-07	0.142	0.234	0.177	0.142	0.000	0.072	0.037	0.036	0.793
C2013-08	0.070	0.338	0.068	0.070	0.000	0.082	0.051	0.051	1.000
C2013-09	0.137	0.346	0.136	0.135	0.058	0.108	0.119	0.108	0.000
C2013-10	0.026	0.164	0.032	0.026	0.000	0.030	0.022	0.022	0.998
C2013-11	0.259	0.423	0.274	0.242	0.208	0.214	0.156	0.098	0.646
C2013-12	0.191	0.541	0.188	0.184	0.050	0.185	0.209	0.181	0.551
C2013-13	0.059	0.510	0.059	0.059	0.000	0.085	0.085	0.081	-
C2013-15	0.288	0.663	0.331	0.152	0.519	0.312	0.237	0.225	0.772
C2014-04	0.267	0.267	0.267	0.203	0.146	0.267	0.267	0.267	-
C2014-05	0.057	0.057	0.057	0.035	0.084	0.057	0.057	0.057	-
C2014-06	0.028	0.027	0.028	0.022	0.026	0.027	0.027	0.027	-
C2014-07	0.050	0.051	0.050	0.036	0.076	0.051	0.051	0.051	-
C2014-08	0.097	0.096	0.096	0.063	0.495	0.053	0.060	0.053	1.000
Avg.	0.107	0.214	0.113	0.092	0.116	0.100	0.094	0.081	0.469

Figure 4. $MAPE_m$ values over all tracking periods of the most accurate project duration forecasting techniques in the 23-project real dataset.

Additionally, method 23 (noted in Figure 4 as *XSM2*) and a new weighted forecasting method named $ES_{min\text{-}max}$ (not shown earlier) are included in this last comparison. Method 23 is one of the exponential smoothing forecasting techniques that depend on a (subjective) exponential smoothing constant named here as γ (with $\gamma \subset [0, 1]$). In the simulated experiments, this constant took the value of 0.05, not to give it an unfair advantage over the other methods. However, on comparing real projects, the value of this constant was optimized for each project to observe how much better othermethods may have got. The specific values of γ for every project are represented in the first grey shaded column and the $MAPE_m$ results of the *XSM2*method itself on the left of the latter shaded column.

$ES_{min\text{-}max}$ is just a weighted average from methods ES_{min} and ES_{max} at every tracking period AT, that is:

$$ES_{min\text{-}max} = \left(1-\delta\right) ES_{min} - \delta \cdot ES_{max} \qquad (10)$$

δ is the weighting factor (with $\delta \subset [0, 1]$) such as when δ=0 then $ES_{min\text{-}max}$=ES_{min}, and whenδ=1 then $ES_{min\text{-}max}$=ES_{max}. It can be seen as some sort of resource transferability factor. This, as δ can be understood as the proportion of resources that can be transferred from the most advanced path (identified with ES_{max}) to help the most delayed path (identified as ES_{min}) catch up. However, the real purpose of $ES_{min\text{-}max}$ method is to allow the comparison of the ES_{min} and ES_{max} metrics with method *XSM2* (as both methods have now one adjustable parameter). Numerical values of the δparameter are specified in the last column of Figure 4, and the $MAPE_m$ values of the $ES_{min\text{-}max}$ method in the penultimate column.

Results are, perhaps, unexpected. Leaving aside the *XSM2* and $ES_{min\text{-}max}$ methods for now, the top performing method is method 26 (depending on metric ES_{max}), despite not by a wide margin and not dominating the other methods in all projects either. The second best is method 25 (depending on ES_{min}). The third best (method 7 noted as *ES1*) and fourth best (method 16 noted as *ESM1*) remain close, whereas the fifth best (method 14 noted as *EDM5*) clearly falls behind (*MAPE*=0.214). On looking at these results,the additional mathematical complexity incorporated by methods 14 (*EDM5*) and 16 (*ESM1*) may be questioned again.

Also, it is striking that method 26 (relying on ES_{max}) provides the most accurate project duration estimates. It is necessary to remember that this metric calculates the project progress as a function of the most advanced path, whereas ES_{min} measures the progress as a function of the most delayed path. The only possible explanations for this result are that, either both metrics must have a higher stability compared to other metrics (as both depend on maxima and minima of many activities) and/or the actual project duration tend to remain in between these two boundaries most of the time. An average resource transferability factor δ of 0.469 may support the latter conclusion.

Finally, results from the one-variable *XSM2* and $ES_{min\text{-}max}$ methods are not surprising. Despite its extremely simple formulation, $ES_{min\text{-}max}$ performs better while resorting to a parameter that also has some physical meaning. The latter suggests that its adjustment may be possible, even subjectively, during project execution. Basically, the project manager may have to estimate what proportion of resources can be moved from the most advanced paths to the most delayed at every tracking period. With these estimates, the predictive power of ES_{min} and ES_{max} can be clearly enhanced. More research is necessary, however, to explore the proper calibration of δ, as well as a more refined reformulation of expression (10).

5. Discussion

Twenty-six deterministic EVM-based project duration methods that resort to different metrics have been compared. Results from the present study agree with someprevious studies on method 7 being the top performer (leaving aside the newones proposed here). However, method 7 either had not been compared with the latest methods (e.g. methods 10 to 15, 22 and 23) or had evenshown worse results (e.g. against methods 19 to 21 and 23)in recent studies. After comparing all methods under the same conditions in both simulated and real projects, method 7 stands out as the most powerful, yet simple, existingproject duration forecasting method. Only method 23, which resorts to an additional exponential smoothing constant, can very marginally outperform method 7. However, in the (more than likely) absence of a good constant calibration, method 23 is highly unlikely to beat method 7's performance.

Method 7, as can be seen in Table 1, resorts to one of the simplest mathematical expressions and is based on the Earned Schedule (*ES*). When it was published by Lipke in 2003 [4], this metric overcame two significant problems the EVM technique had had for a long time. First, it allowed to express in time, instead of money, the project duration-related performance. Second, despite probably unintentionally, it also avoided the bias that the two most relevant duration-related EVM metrics – the Schedule Variance (*SV*) and the Schedule Performance Index (SPI) – sufferwhen a late project is near the end[1]. This bias consists of *SV* and *SPI* converging to 0 and 1, respectively, indicating that the project is exactly on time, no matter the project may already be late (exceeded its Planned Duration *PD*).

However, two weaknesses remained whichwere inherited from the two metrics the *ES* is calculated from: the Planned Value (*PV*) and the Earned Value (*EV*). The *PV* constitutes the planned cost base line and is generally calculated from a deterministic schedule. However, this cost baseline constitutes a lower bound of a realistic *PV*. Nowadays, more realistic *PV* curves can obtainedfrom stochastic network analysis (SNA)[3]. SNA hasproven that project durationsare generally longer than what a deterministic analysis suggests. This means the actual *PV* curve should be partially stretched to the right, otherwise it will always produce an optimistic project completion date. A detailed discussion of this effect can be found in [3]. Unfortunately, this bias cannot be overcome unless we resort to stochastic techniques, which is not the case in the methods compared here.

The second weakness of *ES* comes from the *EV* metric itself. Broadly speaking, the *EV* grows as more activities are executed. This means that when a significant proportion of activities with high planned costs may be completed ahead of schedule the EV will increase rapidly. But a higher *EV* maynot indicatethe existence ofsome (maybe smaller) activities falling behind and causing eventually a project delay. Fortunately, this is the shortcoming that the two new proposed metrics have addressed. By being calculated at activity level, $ES_{min}$for example, can identify which paths are falling behind and provide a more accurate forecast of when the project will actually finish.

In the same vein, but now concerning the limitations of the project datasets used in this study, Figure 3 evidenced that the most challenging networks for forecasting methods are those with more activities in parallel. Actually, if it was not for the *merge event bias* phenomenon discussed earlier, the 26 deterministic methods might have been more accurate. As it happens with the *ES* metric, the challenge of all deterministic EVM-based metrics is to accurately measure current project progress. As discussed above, this is not easy as, somehow, the work performed in the most advanced paths is worth less than the work in paths that are causing a delay (the bottleneck). Dynamically updating this information is, however, not easy, as most EVM metrics do not discern where the work comes from. The inaccuracy of these metrics is then translated to the forecasting methods, which eventually produce worse project duration estimates.

Hence, the most challenging project networks for EVM metrics are predominantly parallel.The inclusion of more parallel networks in the datasets could have provided a higher discriminatory power. However, the vast majority of real projects do not resemble perfectly parallel networks, and if they did, it would be extremely unlikely that all their activity durations were exactly the same (the hardest scenario for all metrics). Therefore, the fact that real projects' Serial-Parallel (*SP*) values mostly fall between 0.3 and 0.7 (as in Table 2) partially disguises the limitations of the deterministic metricscompared here.On the other hand,these comparisonshaveindirectly allowed obtainingmore realistic estimates of the errors that these metrics and methods maysuffer in real project contexts.

Finally, two new metrics have been proposed in this study that, despite extremely simple to calculate, slightly outperform the rest. These metrics named ES_{min} and ES_{max}basically compute the project progress on the most delayed and most advanced paths at any tracking period, respectively. Both metrics perform similarly, but ES_{min} was better in the simulated projects, whereas ES_{max} outperformed the rest in the real projects. In the case of simulated projects, where corrective actions were not possible, the path that fell behind was the most likely to remain behind. This means that, despite not necessarily always, ES_{min} (once coupled in method 25) constitutes an average upper-bound of the project duration. Metric ES_{max} (once coupled in method 26) constitutes an average lower-bound of the project duration (the minimum the project will last). In real life projects, hence, the actual (final) project duration is likely to remain between these two boundaries most of the time.

Additionally, these two metrics have other practical applications. For example, the project manager can use them toidentify those bottleneck activities (the ones whose ES_i coincide with the ES_{min}). If the project needs to be brought back on track or if it just needs to be accelerated, resources need to be mobilized to these critical activities. Those resources should primarily come from those activities whose ES_i coincide with the ES_{max}. In other words, metrics

ES_{min} and ES_{max} allow identifying which activities are in need and those activities which can 'donate' resources. Of course, this assumes that resources are partially transferrable between delayed and advanced activities. If no transference is possible, the project may have to resort to other schedule compression techniques such as activity crashing [37]or fast-tracking [38].

6. Conclusions

Earned Value Management (EVM) is a prominent technique for monitoring project progress in both time and cost dimensions. One of the most common EVM applications involves forecasting the actual project duration. To this end, many EVM-based metrics and methods have been proposed over the last two decades. However, previous studies hadnot compared them alland/or had produced contradicting results on which perform better.

In this study, the performance of 26 deterministic EVM-based project duration forecasting methods has been compared in a set of 4100 simulated projects and 23 real projects.This set of 26 methods encompasses, to the best of the authors' knowledge, all deterministic methods published as of the submission of this paper.

Among the existing metrics, the top performing in both simulated and real project datasets have been the Earned Schedule (*ES*) [4] and Effective Earned Schedule (*ES(e)*) [13] in forecasting methods with Performance Factor 1 (*PF*=1). The Earned Duration (*ED*) metric [6] also performed very well with *PF*=1 in the simulated projects dataset, but fell slightly behind in the real dataset.

Additionally, two new metrics named ES_{min} and ES_{max} and their respective forecasting methods have been proposed. These metrics constitute a partial reformulation of the classical Earned Schedule (*ES*) metricproposed by Lipke in 2003 [4]. ES_{min} and ES_{max} are calculated at activity level, instead of project level, and have marginally outperformed all existingmetrics. Theirmajor advantage is that their calculation is extremely simple, requiring only somebasic schedule information (the activities planned start dates and slacks, andtheircurrent percentages of completion).

Finally, it has been discussed how the ES_{min} and ES_{max} metrics, apart from its higher performance results, alsohave the potential to be used aspowerfulproject control tools. This, asES_{min} and ES_{max}can be used to make decisions on what activities prioritize and how to distribute resources to achieve shorter project durations.$ES_{min}$and ES_{max} can alsobe completely decoupled from the EVM framework, as they do not rely on the Planned Value, Actual cost, nor Earned Value metrics. This significantly lowers the number of calculations to implement them, but also allows them to be used with any other project management framework (Earned Duration Management, for instance). Last of all, ES_{min} and ES_{max} can also be combined into a new metric named here$ES_{min\text{-}max}$, which has been shown to outperform all 26 methods in the real projects dataset. $ES_{min\text{-}max}$has beenprovisionallyexpressed as a weighted average of ES_{min} and ES_{max}via a single parameterδ. Thisparametercan be identified with the (average) potential transference of resources from those activities progressing faster to those activities currently delayed.

Study limitations have also been discussed and emphasize the substantial room for improvement regarding the discriminatory power of the simulated and real project networks used in this study. Despite both datasets are quite representative of real projects, project networks with a higher number of parallel activities could have posed more challenging scenarios for all EVM metrics, and allowed, perhaps, finding more significative differences among the compared methods.All the same, from a practical point of view, any EVM method (top performing methods included) must always be applied with a basic understanding of its underlying assumptions and limitations. Only this way, a project manager, on combining the metrics outputs with other schedule and contextual information, will be able to make better decisions and achieve various project objectives.

Future research will explore the capabilities of the two new metrics proposed (ES_{min} and ES_{max}) plus its combination into $ES_{min\text{-}max}$ for enhanced project monitoring and control.For example, we should be able to analyze how the (time) difference between ES_{min} and ES_{max} can be potentially used to assess how balanced the progress of a project is. This, as theES_{min} and

ES_{max} gap should ideally be always zero (meaning all paths are progressing at the same relative speed). Furthermore, the parameter δcould be used as an indication of how feasible is to bring back the balance between the different paths progress. This, becauseδrepresents the potential overall resource transference from the quickest to the slowest paths.All these continuations, though, are expected to be part of a separate paper.

Acknowledgements

The first author acknowledges the Spanish Ministry of Science, Innovation and Universities for his Ramon y Cajal contract (RYC-2017-22222). This work was also supported by the second author's "Estancias de movilidaden el extranjero José Castillejo para jóvenesdoctores, 2017 (grant ref. CAS17/00488)" and thefourthauthor's"Estancias de profesores e investigadores senior en centros extranjeros Salvador de Madariaga 2018 (grant ref. PRX18/00381)", the last twofrom the Spanish Ministryof Education, CultureandSports.

References

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

[2] A. Martens, M. Vanhoucke, An empirical validation of the performance of project control tolerance limits, Automation in Construction. 89 (2018) 71–85. doi:10.1016/J.AUTCON.2018.01.002.

[3] P. Ballesteros-Pérez, K.M. Elamrousy, On the limitations of the Earned Value Management technique to anticipate project delays, in: I. Press (Ed.), EURO MED SEC 2 - The Second European and Mediterranean Structural Engineering and Construction Conference: Responsible Design and Delivery of the Constructed Project Edited by Abdul-Malak, M., Khoury, H., Singh, A., and Yazdani, S., 2018: pp. 1-6. ISBN: 978-0-9960437-5–5. doi:10.14455/ISEC.res.2018.42.

[4] W. Lipke, Schedule is different, The Measurable News. Summer (2003) 31–34.Last accesses 5th November 2018: http://www.mycpm.org/news/measurable-news/

[5] W. Lipke, Connecting earned value to the schedule, The Measurable News. Winter (2004) 6–16. Last accessed 5th November 2018: http://www.mycpm.org/news/measurable-news/

[6] H. Khamooshi, H. Golafshani, EDM: Earned Duration Management, a new approach to schedule performance management and measurement, International Journal of Project Management. 32 (2014) 1019–1041. doi:10.1016/j.ijproman.2013.11.002.

[7] D.S. Jacob, Forecasting project schedule completion with earned value metrics, The Measurable News. 1 (2003) 7–9.Last accessed 5th November 2018: http://www.mycpm.org/news/measurable-news/

[8] F.T. Anbari, Earned value project management method and extensions, IEEE Engineering Management Review. 32 (2004) 97–97. doi:10.1109/EMR.2004.25113.

[9] O. Zwikael, S. Globerson, T. Raz, Evaluation of models for forecasting the final cost of a project, Project Management Journal. 31 (2000) 53.Last accessed 21st January 2019: https://www.pmi.org/learning/library/evaluation-models-forecasting-final-cost-1991

[10] M. Picornell, E. Pellicer, C. Torres-Machí, M. Sutrisna, Implementation of Earned Value Management in Unit-Price Payment Contracts, Journal of Management in Engineering. 33 (2017) 06016001. doi:10.1061/(ASCE)ME.1943-5479.0000500.

[11] P. Ballesteros-Pérez, M-PERT. A manual project duration estimation technique for teaching scheduling basics, Journal of Construction Engineering and Management. 143 (2017) 04017063. doi:10.1061/(ASCE)CO.1943-7862.0001358.

[12] E. Radziszewska-Zielina, G. Śladowski, M. Sibielak, Planning the reconstruction of a historical building by using a fuzzy stochastic network, Automation in Construction. 84 (2017) 242–257. doi:10.1016/J.AUTCON.2017.08.003.

[13] W. Lipke, Schedule Adherence and Rework, PM World Today. 13 (2011) 1–14. Last accessed 5th November 2018: http://www.earnedschedule.com/Docs/Schedule%20Adherence%20and%20Rework%20-%20PMWT%20(July%202011).pdf

[14] R. Elshaer, Impact of sensitivity information on the prediction of project's duration using earned schedule method, International Journal of Project Management. 31 (2013) 579–588. doi:10.1016/J.IJPROMAN.2012.10.006.

[15] H. Khamooshi, A. Abdi, Project Duration Forecasting Using Earned Duration Management with Exponential Smoothing Techniques, Journal of Management in Engineering. 33 (2017) 04016032. doi:10.1061/(ASCE)ME.1943-5479.0000475.

[16] J. Batselier, M. Vanhoucke, Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting, International Journal of Project Management. 35 (2017) 28–43. doi:10.1016/j.ijproman.2016.10.003.

[17] M. Vanhoucke, Measuring Time - Improving Project Performance Using Earned Value Management, 2010.Springer-Verlag US.doi: 10.1007/978-1-4419-1014-1.

[18] J. Batselier, M. Vanhoucke, Empirical Evaluation of Earned Value Management Forecasting Accuracy for Time and Cost, Journal of Construction Engineering and Management. 141 (2015) 05015010. doi:10.1061/(ASCE)CO.1943-7862.0001008.

[19] J. Batselier, M. Vanhoucke, Evaluation of deterministic state-of-the-art forecasting approaches for project duration based on earned value management, International Journal of Project Management. 33 (2015) 1588–1596. doi:10.1016/j.ijproman.2015.04.003.

[20] P.A. de Andrade, M. Vanhoucke, Combining EDM and EVM: ap proposed simplification for project time and cost management, Journal of Modern Project Management. (2017) 94–106. doi:10.19255/JMPM01410.

[21] L.P. Kerkhove, M. Vanhoucke, Extensions of earned value management: Using the earned incentive metric to improve signal quality, International Journal of Project Management. 35 (2017) 148–168. doi:10.1016/j.ijproman.2016.10.014.

[22] M. Wauters, M. Vanhoucke, A comparative study of Artificial Intelligence methods for project duration forecasting, Expert Systems with Applications. 46 (2016) 249–261. doi:10.1016/j.eswa.2015.10.008.

[23] M. Picornell, E. Pellicer, C. Torres-Machí, M. Sutrisna, Implementation of Earned Value Management in Unit-Price Payment Contracts, Journal of Management in Engineering. 33 (2017) 06016001. doi:10.1061/(ASCE)ME.1943-5479.0000500.

[24] E. Demeulemeester, M. Vanhoucke, W. Herroelen, RanGen: A random network generator for activity-on-the-node networks, Journal of Scheduling. 6 (2003) 17–38. doi:10.1023/A:1022283403119.

[25] M. Vanhoucke, J. Coelho, D. Debels, B. Maenhout, L. V. Tavares, An evaluation of the adequacy of project network generators with systematically sampled networks, European Journal of Operational Research. 187 (2008) 511–524. doi:10.1016/J.EJOR.2007.03.032.

[26] M. Vanhoucke, On the dynamic use of project performance and schedule risk information during project tracking, Omega. 39 (2011) 416–426. doi:10.1016/j.omega.2010.09.006.

[27] M. Wauters, M. Vanhoucke, Study of the stability of earned value management forecasting, Journal of Construction Engineering and Management. 141 (2014) 1–10. doi:10.1061/(ASCE)CO.1943-7862.0000947.

[28] J. Colin, M. Vanhoucke, Setting tolerance limits for statistical project control using earned value management, Omega. 49 (2014) 107–122. doi:10.1016/J.OMEGA.2014.06.001.

[29] E.M. Davies, An Experimental Investigation of Resource Allocation in Multiactivity Projects, Operational Research Quarterly (1970-1977). 24 (1973) 587. doi:10.2307/3008335.

[30] A.A. Mastor, An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques, Management Science. 16 (1970) 728–746. doi:10.1287/mnsc.16.11.728.

[31] D. Trietsch, L. Mazmanyan, L. Gevorgyan, K.R. Baker, Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation, European Journal of Operational Research. 216 (2012) 386–396. doi:10.1016/j.ejor.2011.07.054.

[32] J. Colin, M. Vanhoucke, Empirical Perspective on Activity Durations for Project-Management Simulation Studies, Journal of Construction Engineering and Management. 142 (2016) 04015047. doi:10.1061/(ASCE)CO.1943-7862.0001022.

[33] J. Batselier, M. Vanhoucke, Construction and evaluation framework for a real-life project database, International Journal of Project Management. 33 (2015) 697–710. doi:10.1016/J.IJPROMAN.2014.09.004.

[34] M. Vanhoucke, J. Coelho, J. Batselier, An Overview of Project Data for Integrated Project Management and Control, The Journal of Modern Project Management. 3(2) (2016) 6-21. doi:10.3963/JMPM.V3I3.158.

[35] P. Ballesteros-Pérez, S.T. Smith, J.G. Lloyd-Papworth, P. Cooke, Incorporating the effect of weather in construction scheduling and management with sine wave curves: Application in the United Kingdom, Construction Management and Economics. 36(12) (2018) 666-682. doi:10.1080/01446193.2018.1478109.

[36] H. Khamooshi, D.F. Cioffi, Uncertainty in Task Duration and Cost Estimates: Fusion of Probabilistic Forecasts and Deterministic Scheduling, Journal of Construction Engineering and Management. 139 (2013) 488–497. doi:10.1061/(ASCE)CO.1943-7862.0000616.

[37] P. Ballesteros-Pérez, K.M. Elamrousy, M.C. González-Cruz, Non-linear time-cost trade-off models of activity crashing: Application to construction scheduling and project compression with fast-tracking, Automation in Construction97 (2019) 229-240. doi: 10.1016/j.autcon.2018.11.001.

[38] P. Ballesteros-Pérez, Modelling the boundaries of project fast-tracking, Automation in Construction. 84 (2017) 231–241. doi:10.1016/j.autcon.2017.09.006.