Expert Systems with Applications 42 (2015) 3159–3175

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A comparison of the performance of various project control methods using earned value management systems

Jeroen Colin [a], Mario Vanhoucke [a,b,c,*]

[a] Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium
[b] Technology and Operations Management, Vlerick Business School, Reep 1, 9000 Gent, Belgium
[c] Department of Management Science and Innovation, University College London, Gower Street, London WC1E 6BT, United Kingdom

ARTICLE INFO

Article history:
Available online 11 December 2014

Keywords:
Project management
Schedule control
Earned value management (EVM)
Simulation

ABSTRACT

Recent literature on project management has emphasised the effort which is spent by the management team during the project control process. Based on this effort, a functional distinction can be made between a top down and a bottom up project control approach. A top down control approach refers to the use of a project control system that generates project based performance metrics to give a general overview of the project performance. Actions are triggered based on these general performance metrics, which need further investigation to detect problems at the activity level. A bottom up project control system refers to a system in which detailed activity information needs to be available constantly during the project control process, which requires more effort. In this research, we propose two new project control approaches, which combines elements of both top down and bottom up control. To this end, we integrate the earned value management/earned schedule (EVM/ES) method with multiple control points inspired by critical chain/buffer management (CC/BM). We show how the EVM/ES control approach is complementary with the concept of buffers and how they can improve the project control process when cleverly combined. These combined top down approaches overcome some of the drawbacks of traditional EVM/ES mentioned in the literature, while minimally increasing the effort spent by the project manager. A large computational experiment is set up to test the approach against other control procedures within a broad range of simulated dynamic project progress situations.

1. Introduction

In this paper we focus our research on the control process during the execution of a project. Among the different approaches published in the project management literature, substantial distinctions exist with respect to the work breakdown structure (WBS) level at which the control process is performed, and consequently the effort which is spent during the process and the accuracy of potential actions triggered by such a process. We will restrict our attention to schedule control in this research.

Ultimately, the objective of the control process is finishing the project within a given deadline. It is assumed that the level of detail that has to be available for the project manager during project control corresponds to the effort spent during the control process. This research will not focus on the possible actions to be taken to bring the project back on track, we therefore refer to Herroelen and Leus (2001), Bowman (2006) and Vanhoucke (2011) for an illustration of possible actions that can be incorporated into a dynamic project control experiment. Rather, we will discuss the project control process itself and analyse its performance based on whether or not it produces correct warning signals. We consider the project baseline schedule to be a given and will not discuss different objective functions that can be taken into account during project planning (Liang, 2010) under the availability of limited resources. The reader is referred to a recent survey written by Hartmann and Briskorn (2010) on that topic.

Fig. 1 shows a classification of project control procedures according to the effort invested by the project manager during the project control process. The purpose of this figure is not to give an exhaustive list of the control procedures published in literature or to provide a bullet-proof classification for these control methods. Rather, we wish to express the reduced effort spent by a project manager when only high WBS level information needs to be recorded and processed at each review period during the top down project control process. This *top down* and *bottom up* classification

* Corresponding author at: Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium.
E-mail addresses: jeroen.colin@ugent.be (J. Colin), mario.vanhoucke@ugent.be (M. Vanhoucke).

http://dx.doi.org/10.1016/j.eswa.2014.12.007
0957-4174/

Top down	
EVM	Jacob and Kane (2004), Fleming and Koppelman (2010)
⋮	
EVM-1PB	Lipke et al. (2009)
EVM-STL	Colin and Vanhoucke (2014b)
EVM-SNB	Section 2.2.3
CC/BM	Goldratt (1997), Herroelen and Leus (2001)
EVM-FPB	Section 2.2.2
EVM-LPB	Lipke (2012)
⋮	
CPM	Kelley and Walker (1959)
Bottom up	

Fig. 1. Control process effort classification.

was previously used in the research of Vanhoucke (2011). A *bottom up* procedure requires an intensive and detailed control on all the activities at the lowest WBS level. With this information, a reliable estimate for the final project duration can be calculated and actions can be taken accordingly to meet a given project deadline. Alternatively, a *top down* procedure will only consider a single aggregate performance metric calculated at the top level of the WBS during project control. Only if necessary, additional effort can be spent by drilling down the WBS, in search of those activities that need actions, to ensure a timely completion of the project.

At the bottom of Fig. 1, the critical path method (CPM) is classified as requiring much project control effort by the project manager. CPM is one of the earliest reported approaches for project planning which can also be used in practice to form a basis for prediction of the total project duration and to check progress against a produced baseline during project execution (Kelley & Walker, 1959). At each review period during project control, progress details on all the activities need to be reported. If the CPM algorithm is updated with actual durations for those activities that have been finished, expected finish times for the activities that are in progress and the baseline estimates for the other activities, it produces a reliable estimate for the final project duration. Updating the information for all individual activities can become a cumbersome and disruptive task for project teams in projects with a large number of activities (Lipke, Zwikael, Henderson, & Anbari, 2009).

At the top of Fig. 1, earned value management is classified as demanding less project control effort by the project manager. EVM was originally developed in the 60's by the U.S. Department of Defence as a project cost and schedule control procedure that evaluates and reports performance metrics calculated at high levels of the WBS. Fleming and Koppelman (2010) brought EVM under the attention of researchers, and recent publications have produced the earned schedule method (ES; Lipke et al. (2009)), new project cost and duration forecasting methods using artificial intelligence (Cheng & Wu, 2009; Cheng & Roy, 2010; Wauters & Vanhoucke, 2014) and fuzzy logic (Moslemi Naeni & Salehipour, 2011), and dynamic EVM systems for monitoring (Lee, Peña-Mora, & Park, 2006) and visualisation (Chou, Chen, Hou, & Lin, 2010) of the performance of a project. Jacob and Kane (2004) argue that an aggregate look of the project performance at the highest level of the WBS might lead to misinterpretations of the real project performance and errors in the reported warning signals, and state that the EVM performance measures should be used at lower WBS levels, obviously leading to an increased effort for the project manager. On the contrary, Lipke et al. (2009) argue that the use of EVM on lower levels of the WBS is a cumbersome and often disruptive tasks for the project manager and EVM/ES needs to be applied at high levels of the WBS. These conflicting views on the optimal level for project control using EVM/ES have inspired the work in this paper.

More precisely, we will propose control points for a project at a level of the WBS in-between the top level of traditional EVM and the bottom level of the CPM. These can be interpreted as intermediate levels of the WBS at which the EVM/ES performance measures are calculated. In doing so, they replace the use of control accounts (CA) in an EVM system. Control accounts are natural management points for planning and control, since they represent the work assigned to one responsible organisational element in the WBS. However, since the WBS ordering in control accounts is brought forth by organisational or practical considerations, these control accounts have little or no correspondence to the baseline schedule, i.e. the logic followed during the execution. The proposed control points in this research can also be interpreted as locations in the project activity network. We will therefore frequently refer to the placement of a control point in the network, as calculated from the baseline schedule. The reader should note, that the timing of control points is not discussed in this research (Partovi & Burton, 1993; Raz & Erel, 2000). Each discussed control approach will have an equal number of observations, distributed uniformly over the duration of the project. While the research on the timing of control points is concerned with minimising the number of observation points during the execution of the project (Golenko-Ginzburg & Laslo, 2001), our objective is to investigate the effect of grouping activities into subsets. These subsets are then controlled separately, which should minimise the probability that a deviation from the baseline schedule goes unnoticed and endangers the project deadline. We will propose control points for different control approaches in this research, while incorporating the concept of buffers from the critical chain/buffer management (CC/BM) methodology into the EVM/ES control process to include project baseline information in a structured manner:

- Two new EVM/ES approaches will be introduced in this paper. Similar to the CC/BM approach, buffers are added as control points to the project in each feeding path that enters the critical path (EVM-FPB). EVM/ES performance measures will then be used to monitor the progress of both the critical path and all the feeding paths. While this can lead to a high number of control points, a second approach will also be presented to reduce the number of control points by adding buffers on subnetworks instead of on all feeding paths (EVM-SNB).
- We will test these newly proposed control approach against three additional EVM/ES procedures found in the literature. The traditional EVM/ES control methodology makes use of a single control point at the top level of the WBS, and is therefore labelled as EVM-1PB (1 project buffer). This control methodology can be extended by statistical tolerance limits (EVM-STL)

as proposed by Colin and Vanhoucke (2014). Vanhoucke (2010b) has shown that the EVM methodology shows a reliable behaviour when project networks are close to a serial network, and therefore, Lipke (2012) has proposed to monitor the project performance on the dynamic longest path of the project (EVM-LPB) to improve its performance. This approach requires the calculation of the critical path at each review period and lies closer to the traditional CPM method in Fig. 1.

It should be noted that the control approaches discussed and tested in this paper are inspired by the CC/BM control mechanisms without having the intention to completely follow the CC/BM methodology initially proposed by Goldratt (1997). The critical chain/buffer management project planning and control approach can also be characterised as an in-between top down and bottom up control approach in terms of the control effort on Fig. 1. CC/BM is Goldratt's theory of constraints translated to the field of project management (Rand, 2000). Shortly after Goldratt's initial formulations, CC/BM was found to be popular in both project management practice and literature (Steyn, 2002; Van de Vonder, Demeulemeester, Herroelen, & Leus, 2005). However, more than a purely algorithmic matter, CC/BM is believed to be an important strategic decision and its underlying assumptions and principles have all been challenged in literature. Primarily, the focus of CC/BM is to protect the makespan of the project. From a project control stance, CC/BM accepts that the project in progress is subject to considerable uncertainty, but tries to mitigate risks through focusing on the critical chain and by building in buffers.

Buffers are added to the baseline schedule in three forms. A *project buffer* is added at the end of the project as a way to merge the build-in contingencies that are usually found in individual activities, ultimately reducing the total project duration. The main argument for creating this project buffer is not that there is in general too much contingency built into schedules but rather that it is built in at the wrong place (Steyn, 2000). In order to reduce the contingencies at the activity level, it is conjectured that the activity durations need to be cut aggressively, as much as halving (50% rule) the original estimates (Goldratt, 1997; Herroelen & Leus, 2001). The second type of buffer added in CC/BM, the *feeding buffer*, has also been the subject of debate. Van de Vonder et al. (2005), Trietsch (2005) and Tukel, Rom, and Eksioglu (2006) tested different methods to accurately estimate the optimal buffer size to be added to all non-critical paths that feed into the critical chain. The activities on these non-critical paths are scheduled as late as possible to minimise the work in progress (WIP, the amount of partially finished work in the project) and to decrease the chance of rework if design problems are discovered. Moreover, gating tasks (i.e. tasks that have only a dummy start node as a predecessor) should not start before the scheduled start time, while non-gating tasks should be started as soon as they can, when work becomes available. This last principle is denoted the *roadrunner mentality*. Herroelen and Leus (2001) question this practice as it is not sure whether this is always beneficial with respect to WIP reduction. Furthermore, the implementation of this roadrunner mentality forces the user to maintain two different schedules, i.e. a baseline schedule and a projected schedule. The reader is referred to Herroelen and Leus (2001) for a detailed discussion on the merits and pitfalls of the CC/BM methodology. A third type of buffer, the *resource buffer*, is placed whenever a resource is assigned to an activity on the critical chain and the previous task on the critical chain is performed by another resource. Resource buffers usually take the form of an advance warning signal or idle time that can be created around the resource to create some kind of protective capacity. Since resources are not taken into account in this paper, we will not discuss resource buffers any further.

In this research, we did not have the intention to completely follow the CC/BM methodology. Rather than building buffers into the baseline schedule to mitigate risks, we partly adopt the nomenclature and schedule characteristics of the buffers, from the CC/BM methodology, to define control points throughout the project. At these control points, the schedule performance of the project is checked dynamically during project progress to act as an early warning signal to trigger potential actions. The first control point will always be situated at the project buffer, and will monitor the schedule performance of different subsets of the project activities depending on which project control approach is being used (i.e. EVM-1PB, EVM-STL, EVM-FPB, EVM-SNB or EVM-LPB). Additional control points can be added to monitor the consumption of slack for feeding path buffers (for EVM-FPB) or for subnetwork buffers (EVM-SNB). The use of these buffers will be formalised in Section 2. The number of control points utilised in each project control approach can be interpreted as the effort that needs to be spend by the project manager during the project control process. The EVM-1PB and EVM-STL method employ only one control point, while supplementary control points can be formulated for the EVM-FPB and EVM-SNB control approaches. However, it will be shown further in this paper that the use of EVM/ES performance measures on feeding or subnetwork buffers can lead to an improved ability to protect the project deadline without increasing the number of control points too dramatically. Discussing the number of control points for the EVM-LPB control approach directly, might give the reader the wrong impression that the control effort spent in EVM-LPB is similar to EVM-1PB and EVM-STL. EVM-LPB however, requires calculations of the CPM at each review period and therefore assumes activity level progress data, even though the schedule performance is interpreted at the project buffer. In terms of control effort the EVM-LPB method exceeds the other discussed control approaches. We will test all five control approaches over a broad range of project progress simulations in the computational experiments outlined in Section 3. The results for these experiments will be discussed in Section 4 and general conclusions and a discussion will be given in Section 5.

2. Methodology

In this section, the EVM-FPB and EVM-SNB procedures will be introduced using a small fictitious project example. In Section 2.1, we introduce the notations used in the proposed methodology. We will then demonstrate our methodology in Section 2.2, by illustrating how EVM/ES is integrated within the concept of buffers that act as control points in the schedule. We will formalise the EVM-FPB approach in Section 2.3 and the EVM-SNB approach in Section 2.4.

2.1. Notation and project scheduling

Fig. 2 is used to present the EVM-FPB and EVM-SNB control procedures in this section. The activity network of a small project with

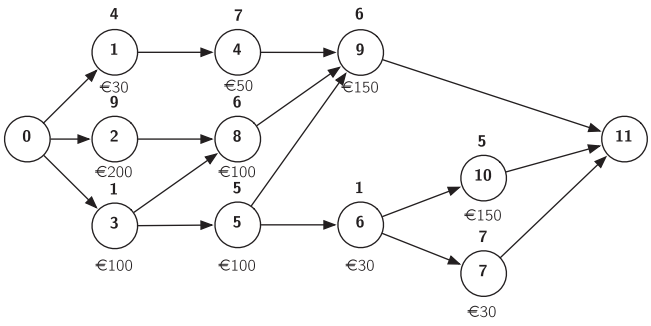

Fig. 2. An illustrative project network.

10 non-dummy activities is represented in an activity-on-the-node diagram. The baseline estimates for the activity durations are expressed in days as shown above the nodes, while the daily costs for the activities are denoted below the nodes in the network. Further notations for the network are formalised in Table 1.

The total planned cost for an activity can be found by multiplying the periodic cost below the node with its baseline estimate duration above the node. For activity i, the total planned cost is given by its budget at completion, $BAC^{\{i\}} = c_i * \widehat{d}_i = (€30) * 4 = €120$. Without loss of generality, the planned value for an activity i is assumed to accrue linearly from 0 at the starting time to $BAC^{\{i\}}$ at the planned finish time. The earned value for an activity i is conjectured the follow this same linear accrue between its real starting time and its real finish time. This distribution of value over the activity duration is believed to model most realistic situations where costs are expressed in man-hours and was used previously in studies by Vanhoucke (2010a) and Colin and Vanhoucke (2014, submitted for publication). We refer to the papers by Fleming and Koppelman (2010) and Vanhoucke (2011) for further details on the calculations of EVM, and to Lipke and Vaughn (2000) for the earned schedule (ES) calculations. A summary of the EVM/ES notations used in this paper is given in Table 1.

Table 1
Project network and EVM notations.

Project characteristics	
$G = (\mathcal{N}, \mathcal{A})$	Project activity network
$\mathcal{N}$	Set of activities (nodes) in the project
$\mathcal{A}$	Precedence relations (arcs) in the project
$\delta_{\mathcal{N}}$	Deadline for the project (in days)
$\mathcal{P}_i$	Set of predecessors of activity $i : \{j \in \mathcal{N} \mid (j,i) \in \mathcal{A}\}$
$\widehat{d}_i$	Baseline estimate for the duration of activity i (in days)
d_i	Real activity duration of activity i (in days)
c_i	Periodic cost for the activity i
Earned value management	
Project EVM/ES key metrics	
BAC^x	Budget at completion for the set of activities x
PV_t^x	Planned value for the set of activities x at period t
EV_t^x	Earned value for the set of activities x at period t
ES_t^x	Earned schedule for the set of activities x at period t
Project EVM/ES performance metrics	
$SV_t^x = EV_t^x - PV_t^x$	Schedule variance for the set of activities x at period t
$SPI_t^x = EV_t^x / PV_t^x$	Schedule performance index for the set of activities x at period t
$SV(t)_t^x = ES_t^x - t$	Schedule variance using earned schedule for the set of activities x at period t
$SPI(t)_t^x = ES_t^x / t$	Schedule performance index using earned schedule for the set of activities x at period t
with:	
$t = 1, \ldots, T$	Current time period
T	Total duration of the project

For the illustrative project network of Fig. 2, the critical path is found by a single forward/backward pass through the network. Fig. 3 shows the critical path of the example network and the resulting earliest start schedule in a Gantt chart. The critical path consists of activities 2-8-9, resulting in a planned project makespan of 21 days. During the execution of the project, the real duration d_i of an activity i can be different from its baseline estimate $\widehat{d}_i$. The slack present in the baseline schedule can then be consumed and non-critical paths might become critical, influencing the total project duration. In Section 2.2, we will illustrate and validate a project control model using EVM/ES that monitors both the performance of the activities on the critical path as well as the consumed slack by non-critical activities feeding into the critical path. This consumption of the slack can then act as a signal to trigger potential actions. To that end, we present 10 fictitious project executions of the example project in Table 2 in which each activity has a real duration d_i that might differ from its baseline estimate $\widehat{d}_i$.

2.2. Illustration of the project control model

In this section, we will demonstrate the new project control approach using EVM/ES. We will improve the schedule control performance of EVM/ES by calculating the performance metrics at multiple control points in the baseline schedule, instead of calculating them for all activities at the highest WBS level. These control points are not chosen arbitrarily, but are a direct consequence of the critical path baseline schedule of the project. The first control point will monitor the critical path, and will be situated at the project buffer. This will be discussed in Section 2.2.1. Non-critical activities on a path that feeds into the critical path can also become critical when their slack is consumed. In order to monitor the consumption of slack throughout the project, the EVM-FPB method considers every feeding path as a control point for which EVM/ES calculations are to be made. Section 2.2.2 discusses this approach. EVM-FPB is likely to lead to a large number of control points in the project since each feeding path is controlled individually, and correspondingly, the project control effort will be high. The EVM-SNB procedure tries to reduce the control effort through reduction of the number of control points in the project. To this end, feeding paths are combined to form a subnetwork. Section 2.2.3 shows how the slack of such a subnetwork can be controlled dynamically.

2.2.1. Project buffer

The dummy end node of Fig. 2 is transformed into a single project buffer in Fig. 3. This project buffer will act as a control point using EVM to predict whether the project deadline is likely to be met. The deadline is chosen in this example as a fixed percentage i.e. 30% of the project planned duration, as shown in the Gantt chart.

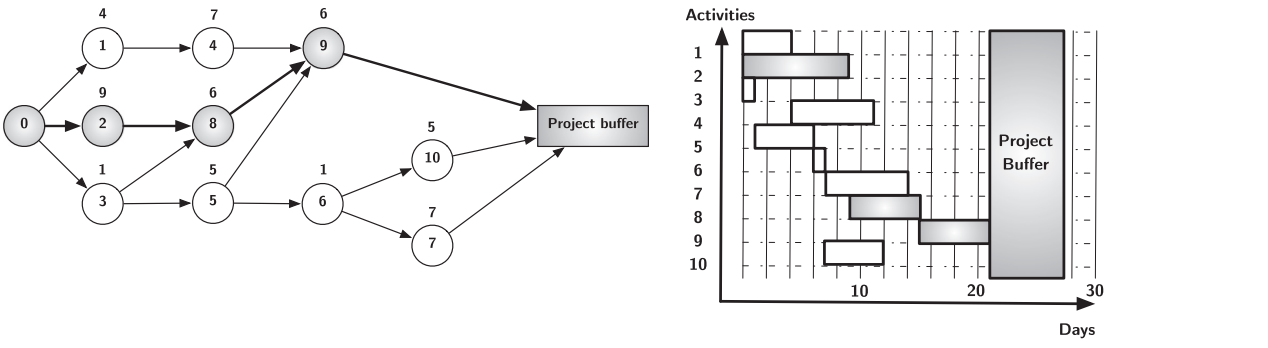

Fig. 3. Critical chain for the illustrative project network.

Table 2
Activity durations for fictitious project executions.

	Simulation run	Activities										Path 3-5-6-7
		1	2	3	4	5	6	7	8	9	10	14
e_i		4	9	1	7	5	1	7	6	6	5	
	1	6	17	3	10	6	2	8	11	10	10	19
	2	6	10	7	13	8	3	7	7	4	9	25
	3	3	9	1	13	9	3	10	9	9	6	23
	4	5	14	2	12	4	1	5	10	7	8	12
$\forall i \in \mathcal{N}$:	5	8	9	1	5	10	1	12	8	12	3	24
$d_i \in \mathbb{N}$	6	3	15	1	5	4	1	14	9	10	8	20
	7	6	9	1	10	4	1	8	9	10	9	14
	8	7	11	2	7	9	1	5	4	6	3	17
	9	6	15	1	7	5	1	8	11	6	8	15
	10	7	9	1	8	5	1	10	6	5	5	17

The project buffer can act as a control point for EVM/ES in different ways, dependent on which control approach is being used. Let Y_t^x denote the EVM/ES measure Y calculated at a time period t for a set of activities x. Y can then represent each of the project EVM/ES performance metrics shown in Table 1. At the project buffer, the EVM/ES performance metric Y_t^x will be controlled dynamically to evaluate whether the project is likely to meet the deadline. The specific EVM control approach which is used, determines the set of activities x for which Y is calculated at each time period t. The EVM control approaches introduced in the following sections (EVM-FPB and EVM-SNB) will only control the critical activities in the project buffer. The other EVM approaches (EVM-1PB, EVM-STL and EVM-LPB) discussed in this paper have a different set x but all of them are controlled at this project buffer control point. We will provide more details on the set of activities x for those control approaches later in this paper.

2.2.2. Feeding path buffer

In order to present the EVM-FPB control procedure, we will restrict our attention to the non-critical path 3-5-6-7 that feeds into the critical path. This path has an expected duration of 14 days and 7 days of slack. This slack is used as the feeding path buffer F_1 as shown in Fig. 4. Table 3 introduces the notations used to discuss feeding path buffers in the EVM-FPB control approach and subnetwork buffers in the EVM-SNB control approach.

Consider a dynamic project control approach that calculates the expected path length ($\widehat{\mathrm{RD}}_{F_j}$) as the sum of the real activity durations for the activities that are finished, the expected duration of the activities in progress and the baseline estimate duration for all activities in the future. The expected buffer penetration can then be calculated for each time period as

$$100\frac{\widehat{\mathrm{RD}}_{F_j} - \mathrm{PD}_{F_j}}{s_j^*} \quad (1)$$

Table 3
EVM-SPB and EVM-SNB buffer notations and project baseline schedule information.

Notations for a buffer F_j	
$\mathcal{F}_j$	Set of activities of the feeding path/subnetwork
$\mathcal{A}_{F_j}$	Set of precedence relations $(i,j) \in \mathcal{A} \mid i,j \in \mathcal{F}_j$
s_j^*	Slack of the last activity of the feeding path/subnetwork
PD_{F_j}	Planned duration of the feeding path/subnetwork
RD_{F_j}	Real duration of the feeding path/subnetwork
$\widehat{\mathrm{RD}}_{F_j}$	Expected real duration of the feeding path/subnetwork
Baseline schedule information	
s_i	Slack of activity i in the project baseline schedule
FT_i	Planned finish time of activity i in the project baseline schedule
ST_i	Planned start time of activity i in the project baseline schedule

This percentage can be less than 0% when the path is performed ahead of schedule, between 0% and 100% when the existing slack is expected to cope with the delays, or more than 100% when the slack will be consumed entirely and the critical path is potentially endangered.

Fig. 5 shows the buffer penetration of three example runs (1, 2 and 3) from Table 2 along the duration of the feeding path. More precisely,the x-axis denotes the current path duration and the y-axis denotes the expected buffer penetration. As an example, activity 3 is finished after day 3 in run 1, and given the expected remaining durations of the other activities, the expected path duration equals 16. Consequently, the expected buffer penetration after day 3 is $100 * (16 - 14)/7 = 28.57\%$. The final buffer penetration at day 19 for the project run 1 for F_1 is $100 * (19 - 14)/7 = 71.42\%$. As long as the final buffer penetration is less then 100%, the slack is sufficient to protect the critical path against delays in the feeding path 3-5-6-7. For the second and the third fictitious project executions, this is clearly not the case as they both end beyond 100% buffer penetration. We have added a suggestion for a straightforward

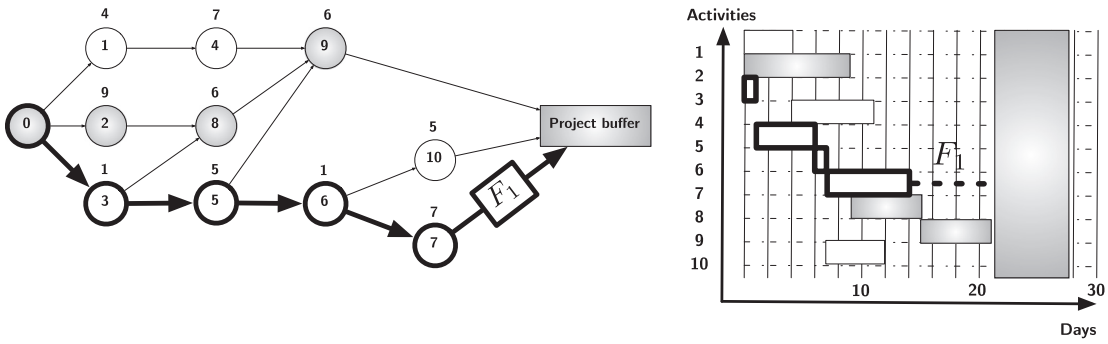

Fig. 4. Path 3-5-6-7 as the feeding path buffer F_1.

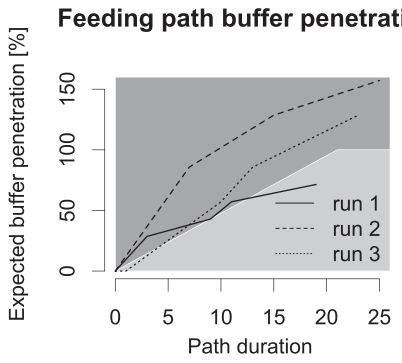

Fig. 5. Expected buffer penetration of F_1.

tolerance limit to trigger early warning signals that could indicate the need to take actions, during the execution of the feeding path. This is done by the lightgrey and darkgrey shaded areas on the graph in Fig. 5. The line which divides the two areas assumes that the 100% buffer is linearly spread over the path duration. From Fig. 5, we notice that this straightforward tolerance limit would trigger a signal for all of the three presented example runs. Since the final consumption of the slack for run 1 is lower than 100%, the warning signal could be interpreted as false. The signals for run 2 and 3 however, express a genuine need for corrective actions since they both end with more than 100% of the slack consumed.

This simple tolerance limit approach can be easily replicated using the standard EVM/ES performance metrics. EVM/ES provides a birds-eye view on the project performance by calculating schedule performance metrics at the top levels of the WBS. Moreover, it is known from literature (Vanhoucke, 2011) that these performance metrics perform best when measuring the schedule performance of a project with a network that lies close to a completely serial network. Consequently, it is an obvious step to consider our feeding path buffer, which acts as a control point for a 100% serial non-critical feeding path, as an EVM/ES performance control point, since it has the guarantee to provide reliable schedule estimates along the complete lifetime of the feeding path.

If Y_t^x denotes the EVM/ES measure Y calculated at a time period t for a set of activities x, then $ES_t^{\mathcal{F}_1}$ is the earned schedule calculated at a period t for the activities in the buffer F_j. If $ES_t^{\mathcal{F}_1}$ and $SPI(t)_t^{\mathcal{F}_1}$ are calculated for the feeding path, the first three runs of Table 2 result in the lines depicted in Fig. 6. The x-axis denotes the feeding path percentage complete, which always equals $ES_t^{\mathcal{F}_1}/PD_{F_1}$, and the y-axis denotes the calculated $SPI(t)_t^{\mathcal{F}_1}$. Let us now evaluate $SPI(t)_t^{\mathcal{F}_1}$ for run 1 at the same days as we did when we discussed the expected buffer penetration. After 3 days ($t = 3$), only the work that was scheduled to be performed in one day ($ES_3^{\mathcal{F}_1} = 1$) has been

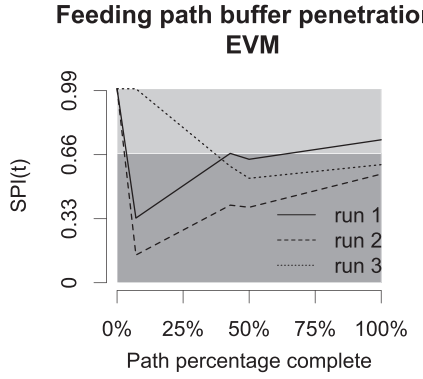

Fig. 6. Dynamic SPI(t) control of path F_1.

accomplished. Therefore, after 7.14% (1/14) of the feeding path planned duration, the $SPI(t)_t^{\mathcal{F}_1} = 1/3 = 0.33$. At the completion of the feeding path, it has taken ($t = 19$) days to complete the work that was scheduled to be done in 14 days, and hence the final $SPI(t)_1^{\mathcal{F}_1}$ equals 0.73. Instead of depicting the expected buffer penetration along the duration of the path and controlling it against a linear tolerance limit as was done in Fig. 5, the $SPI(t)_t^{\mathcal{F}_1}$ can be measured directly against a static threshold. In this approach, the expected path length ($\widehat{RD}_{F_1}$) is calculated using the formula of Lipke and Vaughn (2000) as:

$$\widehat{RD}_{F_1} = \frac{PD_{F_1}}{SPI(t)_t^{\mathcal{F}_1}} \tag{2}$$

This formula assumes that the current SPI(t) is representative for the schedule performance of the rest of the feeding path. Given a maximum buffer consumption of 7 days and a maximum feeding path duration of 21 days, the minimum allowable SPI(t) value is eaqual to 14/21 = 0.67. The SPI(t) is not allowed to drift below this value to avoid more than 100% of the slack being consumed. This minimal value is depicted by the border between the lightgrey and darkgrey area in Fig. 6. This is the same as distributing the buffer over the length of the path linearly, such as was done in the graph of Fig. 5 and consequently, the same conclusions can de drawn for the fictitious project executions depicted in Table 2.

2.2.3. Subnetwork feeding buffer

In the previous section, the slack of a single non-critical path feeding into the critical path was controlled using an EVM/ES measurement system. This provides the management with a reliable estimate whether or not this feeding path is likely to become critical at one point. If this approach is followed for all the non-critical paths that feed into the critical path, the total number of control points can become very large. In this section, we will present an alternative control approach (EVM-SNB) aiming to reduce this high number of control points. This approach combines all the non-critical paths feeding into the critical path in a subnetwork. EVM/ES performance metrics will be used to control all the activities of the subnetwork instead of the individual feeding paths. Calculating the slack for such a subnetwork is no longer trivial, since there might be more than one activity that feeds into the critical paths at the same place in the network. Consequently, tolerance limits are difficult or impossible to derive analytically for the slack present in the baseline schedule. When the subnetwork buffer includes more than one path, we will no longer use the analytical procedure to calculate tolerance limits followed in Fig. 6. However, we will show how an alternative approach using statistical tolerance limits will lead to a figure that is as easily interpretable.

Fig. 7 shows the subnetwork buffer F_1 that now consists of all of the activities of a subnetwork that feeds into the critical path at the project buffer, i.e. $\mathcal{F}_1 = \{3, 5, 6, 7, 10\}$. The subnetwork buffer F_1 is composed out of the two feeding paths 3-5-6-7 and 3-5-6-10 that both feed into the critical path at the same node, with each a different slack value. Path 3-5-6-7 is constrained by the project makespan and has a slack of 7 days, as depicted in Fig. 7. Path 3-5-6-10 is planned to finish two days earlier and its slack is 9 days. Both paths can become critical when they consume more than their slack during the execution of the project. Instead of controlling each path individually using the approaches outlined in Section 2.2.2, we wish to reduce the number of control points by combining them into a subnetwork buffer. To that end, statistical tolerance limits for a performance measure $Y_p^{\mathcal{F}_1}$, calculated for the subnetwork, should be derived. The tolerance limits for $Y^{\mathcal{F}_1}$ should now be calculated for each percentage project complete p, ranging between 10% and 90% instead of for each time instance t (Colin & Vanhoucke, 2014).

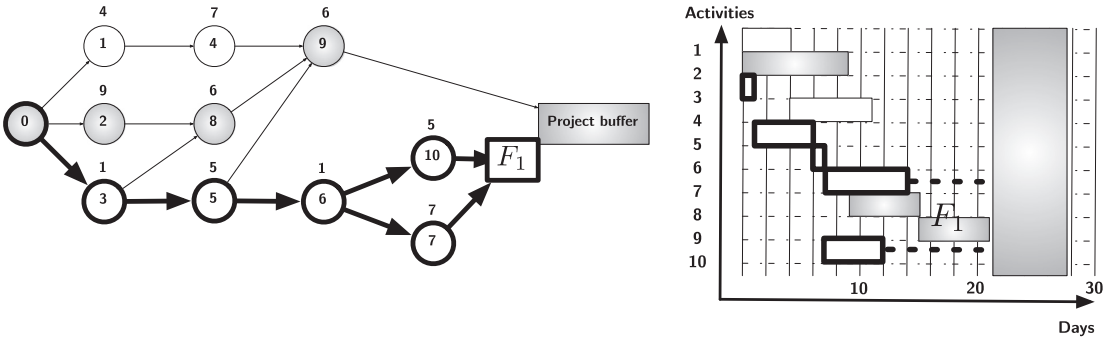

Fig. 7. F_1 consisting of the sub-network 3-5-6-7-10.

In order to calculate these statistical tolerance limits, we need a reference set of project executions. The right side of Table 4 shows the duration of the two paths in our subnetwork. In order for the subnetwork not to infer with the critical path, the two paths have to finish in less than 21 days after the start of the project. When we remove all project progress runs from Table 4 that consume more than their predefined slack for any of the two paths, the remaining project progress runs (with * in Table 4) can be used to calculate sample quantiles for $SPI(t)^{\mathcal{F}_1}$ that act as thresholds that should not be exceeded, just as the minimal SPI(t) tolerance limit depicted in Fig. 6. The $SPI(t)_p^{\mathcal{F}_1}$ values for all 10 fictitious project executions of Table 2 for p varying between 10% and 90%, with increments of 10%, are also presented in Table 4. The calculation of these $SPI(t)_p^{\mathcal{F}_1}$ values was not as straightforward as in the case of the single feeding path, since the subnetwork in F_1 is not serial. Instead, they required the $PV_t^{\mathcal{F}_1}$ and $EV_t^{\mathcal{F}_1}$ curves to be calculated first. The $PV_t^{\mathcal{F}_1}$ and $EV_t^{\mathcal{F}_1}$ curves for the first three fictitious project executions are shown in Fig. 8. From the $PV_t^{\mathcal{F}_1}$ and $EV_t^{\mathcal{F}_1}$ curves, $ES_t^{\mathcal{F}_1}$ can then be calculated and consequently, the $SPI(t)_p^{\mathcal{F}_1}$ values can be found.

As an example, Fig. 9 shows two grey shaded areas divided by the 0% quantile (minimum) of the empirical distribution of $SPI(t)^{\mathcal{F}_1}$. This sample quantile is also presented in Table 5. The lightgrey area represents the sample values for the runs indicated by * in Table 4. The darkgrey area represents all $SPI(t)^{\mathcal{F}_1}$ values which are lower than the 0% sample quantiles found from the set of runs indicated with *. The three lines in Fig. 9 again represent the first three fictitious project executions of Table 4. The first fictitious project execution uses 100% of its slack for the path 3-5-6-10, but does not consume all of the slack for the path 3-5-6-7. Since there is not more than 100% consumption of the subnetwork slack, the critical path is not endangered in this fictitious project execution, as shown by the full line ending in the lightgrey area in Fig. 9. The second and the third fictitious project execution consume more than 100% of the slack in either the path 3-5-6-7 or the path

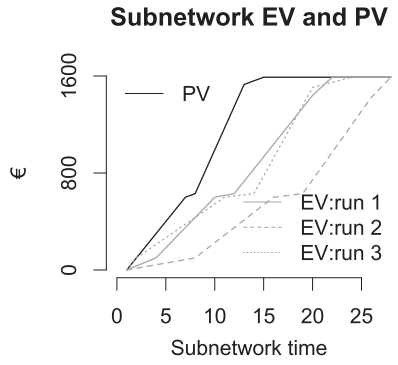

Fig. 8. PV and EV curves for the subnetwork FB_1.

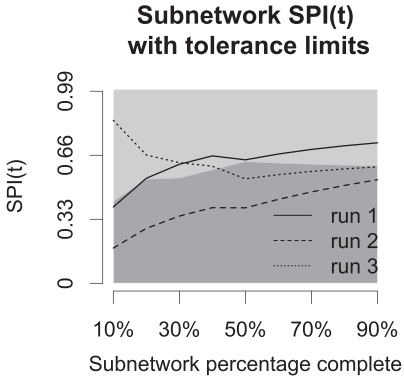

Fig. 9. Dynamic SPI(t) tracking of FB_1, with tolerance limits.

Table 4
SPI(t) calculated for the subnetwork in F_1.

Simulation run		Subnetwork percentage complete									Path duration	
		10%	20%	30%	40%	50%	60%	70%	80%	90%	3-5-6-7	3-5-6-10
*	1	0.36	0.45	0.55	0.64	0.64	0.54	0.51	0.56	0.61	19	21
	2	0.16	0.24	0.31	0.38	0.39	0.38	0.39	0.43	0.47	25	27
	3	0.97	0.86	0.74	0.63	0.54	0.49	0.49	0.53	0.57	23	19
*	4	0.54	0.68	0.82	0.96	1.00	0.90	0.81	0.85	0.89	12	15
	5	0.96	0.84	0.71	0.58	0.58	0.60	0.60	0.60	0.59	24	15
*	6	1.02	1.07	1.13	1.18	1.17	0.78	0.65	0.66	0.68	20	14
*	7	1.02	1.07	1.13	1.18	1.17	0.79	0.69	0.77	0.85	14	15
*	8	0.50	0.52	0.53	0.54	0.58	0.59	0.63	0.70	0.76	17	15
*	9	1.00	1.00	1.00	1.00	1.00	0.65	0.65	0.75	0.84	15	15
*	10	1.00	1.00	1.00	1.00	1.00	0.82	0.76	0.78	0.80	17	12

* These runs are used to produce the sample quantile of Table 5.

Table 5
SPI(t) sample quantiles for the subnetwork in F_1.

Sample quantile	Subnetwork percentage complete								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
0%	0.42	0.54	0.54	0.58	0.63	0.62	0.61	0.61	0.61

3-5-6-10. They are correctly classified by our tolerance limits, since they end up below the 0% sample quantile in the darkgrey area. The use of the 0% sample quantile would have correctly signalled that F_1 was likely to interrupt or endanger the critical path for runs 2 and 3. When Fig. 9 is compared to Figs. 5 and 6, we notice that the false warning signal for run 1 has been removed through the use of statistical tolerance limits on the subnetwork buffer. The use of the feeding path buffer could have possibly led to a overreaction, while the use of the subnetwork buffer would have only shown genuine warning signals, with respect to endangering the critical path of the project.

We have illustrated in this section that the statistical tolerance approach to setting limits on EVM/ES metrics allows the project manager to control the consumed slack within a subnetwork. In doing so, the number of control points throughout the project, and correspondingly the project control effort, can be reduced. Applying the bird's eye perspective of EVM/ES on these specific subnetworks in the project, alongside the critical path, will improve project schedule control. We will further formalise the EVM-FPB and EVM-SNB approaches in Sections 2.3 and 2.4 and test their performance in the remainder of this paper.

2.3. Combining EVM with feeding path buffers (EVM-FPB)

In this section, we formalise the EVM-FPB procedure to combine the use of EVM with feeding path buffers in two parts. In Section 2.3.1, a recursive method is introduced to determine the number of feeding path buffers and their place in the project schedule. Section 2.3.2 calculates the tolerance levels for the EVM/ES performance metrics to be used, while controlling the project buffer and the feeding path buffers.

2.3.1. Recursive addition of feeding path buffers

The algorithm starts with an earliest start schedule of the project network $G(\mathcal{N},\mathcal{A})$ in which 1 is used as the index for the dummy start node and N is the index for the dummy end node. For each activity i in the earliest start schedule, the slack s_i is calculated as the difference between its latest start time and its earliest start time. Consequently, a recursive search is invoked to determine the number and place of the feeding path buffers. $\mathcal{C}=\{i\in\mathcal{N}|s_i=0\}$ will be used to denote the set of activities that lie on the critical path and the set $\mathcal{E}$ is used to store the activities that are still eligible for being passed as argument to the recursive function presented in Algorithm 1.

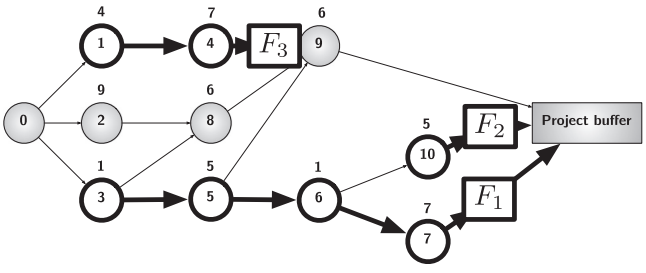

Fig. 10. Feeding path buffers for the example project network.

Algorithm 1: $AddFeedingPath(i)$

```
E = E \ {i}
// Non-critical predecessors
for k ∈ P_i ∩ (E \ C) do
    if (FT_k ≠ ST_i) ∧ (F_j ≠ {}) then
        j = j + 1
        F_j = {}
    end
    F_j = F_j ∪ {k}
    AddFeedingPath(k)
end
if (E \ C ≠ {}) ∧ (F_j ≠ {}) then
    j = j + 1
    F_j = {}
end
// Critical predecessors
for k ∈ P_i ∩ C do
    AddFeedingPath(k)
end
```

Algorithm 1 shows the recursive function to add a total of J feeding path buffers F_j, which should be controlled using EVM/ES performance metrics. After initialisation ($j=1$; $\mathcal{F}_j=\{\}$; $\mathcal{E}=\mathcal{N}\setminus\{1\}$) the dummy end node should be given as argument to the function *AddFeedingPath* (*AddFeedingPath*(N)). Recursively, each node i in the project network will then be visited and non-critical activities will be added to an appropriate feeding path buffer F_j. The algorithm first visits all non-critical predecessors of the node i. The predecessor k will be added to the current feeding path F_j if the start time of i is equal to the finish time of k. Otherwise, a new empty buffer F_{j+1} will be created and the predecessor activity k will be added to this new buffer. Secondly, all critical activities will also be visited in the current recursive level in order to run backwards along the critical path until the dummy start node is reached. For our experiments we will always choose the non-critical predecessor k that leads to the path with the longest sequence of activities first. More precisely, this is recursively done by selecting the predecessor $k\in\mathcal{P}_i\cap(\mathcal{E}\setminus\mathcal{C})$ that will lead to the longest path when continuing the recursive search. Searching for the longest possible paths for each feeding buffer leads to a reduction in the number of feeding path buffers in comparison with the random selection of a path for each feeding buffer. However, the recursive search can be applied differently without loss of generality, possibly leading to a higher number of buffers, and hence, a higher number of control points during project progress. Fig. 10 shows the addition of $J=3$ feeding path buffers to the project example network using our recursive search algorithm.

2.3.2. Setting tolerance limits for EVM-FPB

The EVM-FPB project control approach extends the use of the traditional EVM/ES project control approach to control points on each buffer, aiming at monitoring the schedule progress measured by the SV(t) and SPI(t) performance metrics. The first control point is always situated at the project buffer (or dummy end node) and is used to control the critical path. To that end, either $\text{SV(t)}_t^{\mathcal{C}}$ or $\text{SPI(t)}_t^{\mathcal{C}}$ should be monitored dynamically during the project execution against a tolerance level to act as a threshold for early warning signals. Suppose we have a project deadline that can be written as:

$$\frac{(100+\%pb)}{100}\text{PD}_C \tag{3}$$

with $\%pb$, the project buffer, i.e. the maximum allowable delay expressed as a percentage of the planned duration of the project.

Based on the $\%pb$ value, the tolerance limits for $\text{SV(t)}_t^{\mathcal{C}}$ can be expressed as a function of the time t, where t varies between the starting time of the first activity on the critical path until the finish of the last activity of the critical path, as

$$\frac{-\%pb}{100+\%pb}t \tag{4}$$

Likewise, the tolerance limit for $\text{SPI(t)}_t^{\mathcal{C}}$ can be set as

$$\frac{100}{100+\%pb} \tag{5}$$

which shows that its value is independent from the time t.

The feeding path buffers act as additional control points to be controlled alongside the critical path. Analogously to the project buffer, if the slack of a feeding path F_j is written as a percentage of the planned duration of the feeding path: $\%fpb = 100s_j^*/\text{PD}_{F_j}$, then the tolerance limits are $-\%fpb/(100+\%fpb)t$ and $100/(100+\%fpb)$ for $\text{SV(t)}_t^{\mathcal{F}_j}$ and $\text{SPI(t)}_t^{\mathcal{F}_j}$ respectively, where t is the time since the planned start of the first activity of F_j.

2.4. Combining EVM with subnetwork buffers (EVM-SNB)

In this section, we formalise the EVM-SNB procedure using control points on subnetworks. In Section 2.4.1, a recursive method is introduced to determine the subnetwork buffers and their place in the baseline schedule. Section 2.4.2 calculates the tolerance levels for the EVM/ES performance metrics to be used while controlling the critical path and the subnetwork buffers.

2.4.1. Recursive addition of subnetwork buffers

Similar to the EVM-FPB approach the procedure starts with an earliest start schedule, and relies on the recursive search of Algorithm 2 to determine and add the subnetwork buffers.

Algorithm 1: *AddFeedingPath*(i)

```
E = E \ {i}
// Non-critical predecessors
for k ∈ P_i ∩ (E \ C) do
    if (FT_k ≠ ST_i) ∧ (F_j ≠ {}) then
        j = j + 1
        F_j = {}
    end
    F_j = F_j ∪ {k}
    AddFeedingPath(k)
end
if (E \ C ≠ {}) ∧ (F_j ≠ {}) then
    j = j + 1
    F_j = {}
end
// Critical predecessors
for k ∈ P_i ∩ C do
    AddFeedingPath(k)
end
```

After initialisation ($\mathcal{F}_1 = \{\},\ \mathcal{E} = \mathcal{N} \setminus \{1\}$) the dummy end node and $j = 1$ should be given as arguments to the function *AddSubNetwork* (*AddSubNetwork*$(N, 1)$) to produce a total of J^* subnetwork buffers. For a node i in the activity network, the recursive search will first visit all non-critical predecessors k and will add them in a subnetwork buffer F_j. If all non-critical predecessors have been visited, the search is continued along the critical path and a new subnetwork buffer F_{j+1} is created as long as there are still non-critical activities eligible, given by $\mathcal{E} \setminus \mathcal{C} \neq \{\}$, and $\mathcal{F}_j$ is not empty. Fig. 11 shows how $J^* = 2$ subnetwork buffers are added to the example project network.

The number of subnetwork buffers J^* will represent a significant decrease in the number of control points, compared to the number of feeding path buffers J as calculated by Algorithm 1. This difference will be the largest for project network structures close to parallel. In the extreme, for a complete parallel project, the procedure outlined in Algorithm 2 will result in $J^* = 1$, while Algorithm 1 will produce $J = N - 2$ feeding path buffers.

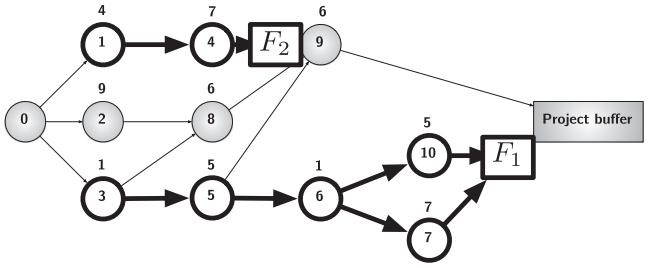

Fig. 11. Subnetwork buffer for the example project network.

2.4.2. Setting tolerance limits for EVM-SNB

The tolerance limits to control the subnetwork buffers in the EVM-SNB procedure are calculated from sample quantiles of a Monte Carlo simulation performed prior to the project progress as illustrated in Section 2.2.3. These sample quantiles can be calculated for any EVM/ES performance metric Y and for any αth quantile of the empirical distribution function approximated from the simulation. We refer to Hyndman and Fan (1996) and Colin and Vanhoucke (2014) for more detail on the calculations of the αth quantile from the simulated set of observations $\{Y_{i,p}^x | i \in \mathcal{M}\}$, where $Y_{i,p}^x$ is used to denote the observation of an EVM/ES performance metric Y, calculated for a set of activities x in a run i ($1 \leqslant i \leqslant n$) at p percentage of the project complete. The α^{th} quantile $\hat{Q}(\alpha)_p$ at a percentage complete p, is the number q for which the probability of drawing a value from Y_p less than q is at most α.

In order to find appropriate runs to form the reference set $\mathcal{M}$, a Monte Carlo simulation is performed. From the simulation output, all runs for which the delays in the subnetwork buffers exceed the slack of that subnetwork are removed, and the remaining simulation runs form the reference $\mathcal{M}$ from which statistical tolerance limits can be calculated. In the EVM-SNB approach, tolerance limits need to be calculated for the set of activities x, equal to $\mathcal{C}$ for the critical path and x equal to $\mathcal{F}_j$ for the subnetwork buffers ($j \in \{1, \ldots, J^*\}$).

The distributional input to the Monte Carlo simulation should be chosen such that it accurately reflects the real uncertainty that is encountered in practice in project activity durations. Probability density functions for all activities can be obtained from expert judgement or from calibration to historical data and the specific settings for the uncertainty modelling in our study will be introduced in Section 3.

3. Computational experiments

In this section, we introduce an extensive simulation experiment in which we compare the performance of the EVM-FPB and EVM-SNB control approaches against EVM-1PB, EVM-STL, EVM-LPB. Section 3.1 will formally present the EVM-1PB, EVM-STL, EVM-LPB control approaches. Monte Carlo simulation experiments are well established in the project management research literature (Van Slyke, 1963; Williams, 1995; Bowman, 2006). The project dataset and the dynamic project progress model for these experiments will be introduced in Section 3.2. Section 3.3 will discuss the input modelling for the activity durations used in the. Finally, we will elucidate the measures used to quantify and compare the performance of the project control approaches in Section 3.4.

3.1. EVM project control procedures

The EVM-FPB and EVM-SNB procedures combine the bird's eye perspective of EVM with cleverly assigned project control points to incorporate baseline schedule information. In our computational

experiments, we investigate whether this combination is likely to improve the performance of the project control process in protecting the project makespan, without excessively increasing the effort spent in the process. We will therefore compare the introduced project control procedures with three alternative EVM control methods encountered in literature. We will first give a short overview of the alternative approaches in Section 3.1.1 and second, we will present the main characteristics of the five project control procedures and the details on their implementation in this study in Section 3.1.2.

3.1.1. Introduction from the project control approaches from literature

The traditional **EVM-1PB** procedure relies on the earned schedule performance measures SV(t) and SPI(t) calculated for the complete project. It is labelled as EVM-1PB, since it uses only one control point that is situated at the project buffer. However, unlike the EVM-FPB and EVM-SNB procedures, this control point includes the performance of all the activities (i.e. the set $\mathcal{N}$) and not only those that lie on the critical path. If the project deadline is written as in Eq. (3), then tolerance limits can be calculated for $SV(t)_t^{\mathcal{N}}$ and $SPI(t)_t^{\mathcal{N}}$ using Eqs. (4) and (5). A warning signal for schedule control is said to be produced by the EVM-1PB procedure at a time t if $SV(t)_t^{\mathcal{N}}$ or $SPI(t)_t^{\mathcal{N}}$ are lower than their tolerance limits. This control approach for projects was previously discussed by Anbari (2003) as the use of "Target Performance Charts", and implemented by Vanhoucke (2012) in his study on top-down project control using EVM/ES.

The **EVM-STL**, previously described by Colin and Vanhoucke (2014), also uses only one control point at the project buffer that includes the performance of all activities (the set $\mathcal{N}$). However, the difference lies in the calculation of the tolerance limits on the EVM/ES performance metrics at each p percentage project completion ($SV_p^{\mathcal{N}}$, $SPI_p^{\mathcal{N}}$, $SV(t)_p^{\mathcal{N}}$ and $SPI(t)_p^{\mathcal{N}}$). Similar to the procedure outlined in Section 2.4.2, this is done by estimating sample quantiles ($\hat{Q}(\alpha)_p$) from the Monte Carlo simulation runs in a set $\mathcal{M}$, prior to the start of the execution phase of the project. A warning signal is said to be produced when at any percentage complete p during the execution of the project, the measured EVM/ES performance metrics are lower than their corresponding tolerance limits.

Inspired by the observation that the EVM/ES metrics are fully reliable for serial networks, the longest path EVM (**EVM-LPB**) control approach was proposed by Lipke (2012). It also employs only one control point at the project buffer, but the EVM/ES performance metrics are now only calculated for the activities lying on the dynamic longest path $\widehat{\mathcal{C}}_t$. This dynamic longest path can be perceived as a dynamic (i.e. dependent of the time t) estimate of the real critical path of the project, and therefore might be different for different values of t. It assumes that at each time instance the critical path method calculations are made to establish the longest path in the schedule, given the baseline estimates for the durations of the activities that have not been finished and the real durations for those that have been finished. This procedure require full knowledge at the activity level of the project and is therefore classified as closest to the CPM method in Fig. 1 with respect to the control effort. If the project deadline is written as in Eq. (3), then tolerance limits can be calculated for $SV(t)_t^{\mathcal{C}_t}$ and $SPI(t)_t^{\mathcal{C}_t}$ using Eqs. (4) and (5). A warning is said to be signalled by the EVM-LPB control procedure if the current schedule performances are lower than their tolerance limits.

3.1.2. Implementation and comparison

Table 6 summarises the main characteristics of the five project control procedures that are compared in this computational experiment. For EVM-1PB, EVM-STL and EVM-LPB the main references are provided in the second column and for EVM-FPB and EVM-SNB, we refer to the corresponding algorithms presented this paper. The third column of Table 6 presents the number of control points $J^{\#}$ for each control procedure. For the EVM-LPB approach, this amounts to N, which indicates that at each review period $k = 1 \ldots K$ all N activities of the project need to be consulted in order to produce a warning signal.

In order to present an objective comparison of the five project control procedures in this paper, we have restricted their implementation such that they use a single EVM/ES performance metric, so that for all of them Y^x equals $SPI(t)^x$. Moreover, a total number of control periods K over the lifetime of the project was kept fixed for each project control procedure. Therefore, the time t at which the $SPI(t)^x$ is calculated for a set of activities x is chosen such that it coincides with $K = 19$ distinct intervals along the project complete axis ($t = t(p),\ \forall p = 5\%, 10\%, \ldots 95\%$). The fourth column of Table 6 presents the sets of activities x for each control procedure and the corresponding tolerance limits are shown in the fifth column.

The reader should note that an additional index α has also been added to the tolerance limits of the EVM-1PB, EVM-FPB and EVM-LPB project control procedures. $\%pb_\alpha$ and $\%fpb_\alpha$ thereby respectively represents a "virtual" project and feeding path buffer that allows the tolerance limits for these control procedures to be calculated with more flexibility. This was done to ensure a maximal *effectiveness* for all control procedures in our computational experiment, in order to allow us to compare them based on their *efficiency* and *control effort*. We will go into more detail concerning these performance measures in Section 3.4.

3.2. Dynamic project progress model

The simulation experiment, deployed in this computational experiment, uses the project network dataset of Vanhoucke (2011) to test the performance of the control approaches. The 900 networks were generated using RanGen (Demeulemeester, Vanhoucke, & Herroelen, 2003; Vanhoucke, Coelho, Debels, Maenhout, & Tavares, 2008), and ensure a maximal diversity in terms of network structure within the dataset. The project progress for this research is performed using Monte Carlo simulations in P2 Engine (Vanhoucke, 2014), a scripting-enabled project scheduling and control tool, previously used by Vanhoucke (2011) and Colin and Vanhoucke (2014, submitted for publication). Project progress

Table 6
Overview of the control procedures discusses in the computational experiments.

Name	References/algorithm	$J^{\#}$	Sets of activities	Tolerance limits
EVM-1 PB	Vanhoucke (2012) and Anbari (2003)	1	$x = \mathcal{N}$	$\frac{100}{100+\%pb_\alpha}$
EVM-STL	Colin and Vanhoucke (2014)	1	$x = \mathcal{N}$	$\widehat{Q}(\alpha)_p$
EVM-FPB	*AddFeedingPath*	$J+1$	$x = \begin{cases} \mathcal{C} & j = 1 \\ \mathcal{F}_j & j = 2 \ldots J+1 \end{cases}$	$\begin{cases} \frac{100}{100+\%pb_\alpha} \\ \frac{100}{100+\%fpb_\alpha} \end{cases}$
EVM-SNB	*AddSubNetwork*	$J^* + 1$	$x = \begin{cases} \mathcal{C} & j = 1 \\ \mathcal{F}_j & j = 2 \ldots J^*+1 \end{cases}$	$\widehat{Q}(\alpha)_p$
EVM-LPB	Lipke (2012)	N	$x = \widehat{\mathcal{C}}_t$	$\frac{100}{100+\%pb_\alpha}$

is simulated from fictitious project executions, where actual durations for the activities deviate from the baseline estimate durations given the applied modelling of the input uncertainty. The EVM/ES control data calculated during the fictitious project executions are based on the assumption that both PV and EV accrue linearly for an activity during its execution.

3.3. Simulation input modelling

In order to test and validate the performance of the different control approaches, two Monte Carlo simulations are needed for each project. After establishing the subnetwork buffers through Algorithm 2, a first simulation will be used to calculate the tolerance limits for the EVM-SNB approach as described in Section 2.4.2. Subsequently, a second simulation allows us to calculate the performance of the project control approach, as will be discussed in Section 3.4. Both these simulations are run with appropriate settings for the distributions to model uncertainty on activity durations.

In this research, we have chosen to model uncertainty at the activity level by both variation and risk. Variation on the activity level of a project can be expressed by applying probability density functions on the real activity durations and will be discussed in Section 3.3.1. In addition to variation, project management literature often recounts other types of uncertainty occurring during execution (Loch, De Meyer, & Pich, 2006). Risk, as discussed in Section 3.3.2, is regularly used to denote (un)likely events that can have a potentially disastrous impact on multiple activities in the project and the overall project objectives. This type of uncertainty is often overlooked in project control simulation, since there is no generalised way to model correlation structures or the occurrence of events. A solution to overcome this shortcoming is provided by the linear association approach that looks at statistical dependencies between activities in the project (Trietsch, Mazmanyan, Govergyan, & Baker, 2012). In our experimental model, we will incorporate both variation in the form of activity-specific probability density functions for the duration and risks by using linear association, as will be discussed in Section 3.3.3.

3.3.1. Variation model

We will model variation on the activity durations through probability density functions from the generalised beta distribution family. Generalised beta functions have long been associated with PERT estimates in project management (McBride & McClelland, 1967) for their ability to accurately mimic the behaviour of the random input processes driving the system (Kuhl, Lada, Steiger, Wagner, & Wilson, 2007). The probability density function for a random variable X is

$$f_X(x|a,b,\theta_1,\theta_2) = \begin{cases} \frac{\Gamma(\theta_1+\theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)} \frac{(x-a)^{\theta_1-1}(b-x)^{\theta_2-1}}{(b-a)^{\theta_1+\theta_2-1}} & \text{if } a \leqslant x \leqslant b \\ 0 & \text{if } x < a \vee b < x \end{cases} \quad (6)$$

where $\Gamma()$ denotes the gamma function and θ_1 and θ_2 are the shape parameters. AbouRizk, Halpin, and Wilson (1994) suggest an approximation for fitting the beta distribution to historical data or subjective estimates, when uncertainty is characterised by a long tail, as often described in project management. This approximation produces estimates for the beta shape parameters through a single auxiliary parameter, calculated from PERT-style three point estimates. In Vanhoucke (2011), these approximations have been used to test the performance of EVM time-forecasting methods on a large fictitious dataset. However, since large variation instances for fixed values for the average of the activity duration can not be produced using the approximation of AbouRizk et al. (1994), we propose the use of a parameter vector $\boldsymbol{\omega} = (a, b, m, \mu)$, with estimates for the minimum, the maximum, the mode and the mean of the distribution expressed as a fraction of the baseline estimate duration $\widehat{d}_i$. In doing so, we can write the probability density function of the random duration D_i of activity i as:

$$f_{D_i}\left(d_i|a\widehat{d}_i, b\widehat{d}_i, \theta_1(\boldsymbol{\omega}), \theta_2(\boldsymbol{\omega})\right) = f_X(x|a,b,\theta_1(\boldsymbol{\omega}),\theta_2(\boldsymbol{\omega}))\widehat{d}_i \quad (7)$$

$$= \beta(\boldsymbol{\omega})\widehat{d}_i \quad (8)$$

with $x = d_i/\widehat{d}_i$ and

$$\begin{cases} \theta_1(\boldsymbol{\omega}) = -\frac{(b+a-2m)(a-\mu)}{(m-\mu)(a-b)} & \text{(a)} \\ \theta_2(\boldsymbol{\omega}) = \frac{(b+a-2m)(b-\mu)}{(m-\mu)(a-b)} & \text{(b)} \end{cases} \quad (9)$$

where the shape parameters (Eq. (9)) are independent from the baseline estimate $\widehat{d}_i$ and can be found by solving the set of Eq. (10) through substitution.

$$\begin{cases} \mu = \frac{\theta_1 b + \theta_2 a}{\theta_1+\theta_2} & \text{(a)} \\ m = \frac{(\theta_1-1)b+(\theta_2-1)a}{\theta_1+\theta_2-2} & \text{(b)} \end{cases} \quad (10)$$

3.3.2. Risk model

We have chosen to model risk using the linear association approach of Trietsch et al. (2012). In the absence of a generalised way to model the potential impact of events on multiple activities through a correlation matrix, Trietsch et al. (2012) and Trietsch and Baker (2012) suggested to use a positive random variable B to model dependencies between activities. The bias term B is easily perceived as a consistent over- or underestimation of activity duration, or a project-wide effect of uncertain events. In a critical chain context, Trietsch (2005) investigated the effect of a systemic error on setting an optimal project buffer. A set of positive random variables Z_i are said to be linearly associated if $Z_i = BY_i$, where $\{Y_i\}$ is a set of positive random variables with the same cardinality. Trietsch et al. (2012) show that the inclusion of a bias term B is indispensable when a project model is calibrated to historical data of nine projects. According to Trietsch et al. (2012), B is in practice influenced by both additive and multiplicative causes and is therefore assumed to follow a lognormal distribution. We reason that the generalised beta family of distributions has an advantage in simulation input modelling over the lognormal distribution. It is a more intuitive distribution, which originates from practical applications, and which allows $f(x) = 0$ for $x < \infty$. The lognormal distribution has $\lim_{x\to\infty} f(x) = 0$, a theoretical limit that is not really intuitive (every positive duration having non-zero probability of occurring), nor is it practically reproducible in a computerised simulation experiment.

3.3.3. Combined input model

For each run in the simulation experiment, numbers will be generated from the generalised beta distribution to model both activity variation and risk. To that purpose, a bias term B will first be sampled for each run in the simulation experiment from a generalised beta probability distribution $\beta(\boldsymbol{\omega}_R)$. Subsequently, for all activities $i = 1 \ldots N$ in this fictitious project execution, an activity duration d_i is drawn from the distribution $\beta(\boldsymbol{\omega}_V)B\widehat{d}_i$. This distribution can be seen as a generalised beta distribution $\beta(\boldsymbol{\omega}_V)$ with a independently chosen parameter vector that represents the variation $\boldsymbol{\omega}_V$, which is applied to a scaled baseline estimate $(B\widehat{d}_i)$ of the real activity duration.

These assumptions and characteristics allow the use of a limited number of parameter vectors $\boldsymbol{\omega}$ (either $\boldsymbol{\omega}_R$ or $\boldsymbol{\omega}_V$) in our computational experiment. Table 7 depicts the selection of parameter vectors from which $\boldsymbol{\omega}_R$ and $\boldsymbol{\omega}_V$ are chosen in the simulations. Preliminary calculations and visualisation in the statistical programming language R (R Core Team, 2013) provided validation for the

Table 7
Parameter vectors for the generalised beta distribution.

General performance experiment			Sensitivity experiment					
			Mean			*Standard deviation*		
$\boldsymbol{\omega}$	(a,b,m,μ)	σ	$\boldsymbol{\omega}$	(a,b,m,μ)	σ	$\boldsymbol{\omega}$	(a,b,m,μ)	σ
$\boldsymbol{\omega_1}$	$(0.1,1.2,0.7,0.6)$	0.38						
$\boldsymbol{\omega_2}$	$(0.2,4,0.9,1)$	0.30	$\boldsymbol{\omega}_{2\mu_1}$	$(0.2,4,0.51,0.7)$	0.3	$\boldsymbol{\omega}_{2s_1}$	$(0.2,4,0.40,1)$	0.50
			$\boldsymbol{\omega}_{2\mu_2}$	$(0.2,4,0.90,1.0)$		$\boldsymbol{\omega}_{2s_2}$	$(0.2,4,0.70,1)$	0.42
			$\boldsymbol{\omega}_{2\mu_3}$	$(0.2,4,1.22,1.3)$	0.3	$\boldsymbol{\omega}_{2s_3}$	$(0.2,4,0.85,1)$	0.35
			$\boldsymbol{\omega}_{2\mu_4}$	$(0.2,4,1.57,1.6)$		$\boldsymbol{\omega}_{2s_4}$	$(0.2,4,0.92,1)$	0.28
			$\boldsymbol{\omega}_{2\mu_5}$	$(0.2,4,1.89,1.9)$	0.3	$\boldsymbol{\omega}_{2s_5}$	$(0.2,4,0.96,1)$	0.20
						$\boldsymbol{\omega}_{2s_6}$	$(0.2,4,0.98,1)$	0.15
$\boldsymbol{\omega_3}$	$(0.9,4,1.3,1.4)$	0.38						

suggested parameter vectors of Table 7 during our research. The first set of parameter vectors ($\boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ and $\boldsymbol{\omega_3}$) are chosen to emulate practical distributions, with realistically shaped tails that represent scenarios were the average activity duration is lower than, equal to or higher than its baseline duration. The standard deviations of $\boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ and $\boldsymbol{\omega_3}$ equal 0.38, 0.30 and 0.38 respectively and hence, an appropriate choice for both $\boldsymbol{\omega_R}$ and $\boldsymbol{\omega_V}$ from this set of parameter vectors is sufficient to simulate large samples for $d_i/\widehat{d}_i$ with standard deviations up to 0.5. Samples of activity durations with standard deviation up to 0.5 were found to adequately represent the largest variation instances from the sizeable collection of empirical data provided by Batselier and Vanhoucke (2014). For more than 90% of the projects found in the database of Batselier and Vanhoucke (2014), the sample standard deviation was found to be less than 0.5. This database currently consists of 51 real-life projects with actual progress data and is very diverse with respect to sector, planned duration and budget at completion characteristics of the projects. The second set of parameter vectors ($\boldsymbol{\omega}_{2\mu_1}, \ldots \boldsymbol{\omega}_{2\mu_5}$) is derived from $\boldsymbol{\omega_2}$ to test the influence of a change in the mean of the distribution. If the mean is changed, the mode needs to change accordingly to keep a realistic shape for the distribution function, and to ensure that the standard deviation remains the same as for $\boldsymbol{\omega_2}$, $\sigma = 0.3$. The last set of parameter vectors ($\boldsymbol{\omega}_{2s_1}, \ldots \boldsymbol{\omega}_{2s_6}$) will be used to explore the influence of a change in the standard deviation. Is it only necessary to change the mode of the generalised beta distribution to effectively change the standard deviation, while keeping the mean $\mu = 1$ unchanged.

3.4. Performance measurement

In the computational results section, we will compare the *efficiency* of all project control methods under such conditions that the effectiveness is always equal to 100%. To that purpose, we also measure the *control effort* of each project control method as a proxy of the amount of time a project manager has to spend in monitoring project progress using each method. Both the efficiency and control effort are quantified in the following two subsections.

3.4.1. Efficiency

The efficiency of project control methods have been defined earlier by Vanhoucke (2012) using a corrective actions model that is used when projects are in danger. However, since in the current study no corrective actions are implemented, we have adopted the efficiency definition of project control. Instead, we rely on a model that assumes that each control method is always able to detect deviations from the plan. More precisely, we have set the α parameter, used for setting the tolerance limits of the EVM-1PB, EVM-FPB and EVM-LPB methods using virtual buffers or as sample quantiles for the EVM-STL and EVM-SNB methods, to such a value that the *detection performance* of each control method is equal to 100%. The detection performance has been used in the study of Colin and Vanhoucke (2014) and is equal to:

$$\mathrm{P}[S|RD > \delta_{\mathcal{N}}] = \frac{\sum_{i=1}^{n} \mathbb{1}_i(RD > \delta_{\mathcal{N}})\mathbb{1}_i(S)}{n}$$

Here, $\mathbb{1}_i(A)$ denotes the identity operator, which returns 1 if statement A is true and 0 if statement A is false for simulation run i. Moreover, a signal can be produced by any of the control points in the selected control procedure, so S is a logical disjunction over all control points:

$$S = \bigvee_{j=1}^{J^{\#}} S_j$$

where $J^{\#}$ represents the number of control points in the selected control procedure and S_j represents whether or not a signal is produced in control point j. Similarly, we can calculate the *probability of encountering false warning signals* (Colin & Vanhoucke, 2014) as:

$$\mathrm{P}[S|RD \leqslant \delta_{\mathcal{N}}] = \frac{\sum_{i=1}^{n} \mathbb{1}_i(RD \leqslant \delta_{\mathcal{N}})\mathbb{1}_i(S)}{n}$$

The efficiency of a control procedure expresses the probability that the project duration will exceed its final deadline, given the event that a signal S is reported. Therefore, Bayes' theorem allows us to calculate the efficiency of the project control procedure from the detection performance and the probability of encountering false warning signals as:

$$\begin{aligned}\mathrm{P}[RD > \delta_{\mathcal{N}}|S] &= \frac{\mathrm{P}[S|RD > \delta_{\mathcal{N}}]\ \mathrm{P}[RD > \delta_{\mathcal{N}}]}{\mathrm{P}[S]} \\ &= \frac{\mathrm{P}[S|RD > \delta_{\mathcal{N}}]\ \mathrm{P}[RD > \delta_{\mathcal{N}}]}{\mathrm{P}[S|RD > \delta_{\mathcal{N}}]\ \mathrm{P}[RD > \delta_{\mathcal{N}}] + \mathrm{P}[S|RD \leqslant \delta_{\mathcal{N}}]\ \mathrm{P}[RD \leqslant \delta_{\mathcal{N}}]}\end{aligned} \quad (11)$$

where $\mathrm{P}[RD > \delta_{\mathcal{N}}]$ and $\mathrm{P}[RD \leqslant \delta_{\mathcal{N}}]$ depend on the probabilistic outcome of the project simulations. This is affected by both the given the simulation input modelling (Section 3.3) and the underlying project network structure.

3.4.2. Control effort

For each of the control procedures discussed in this computational experiment, we will also directly measure the control effort. This measure counts the relative number of times a signal is reported by the project control approaches, and consequently the relative number of times that the activity level performance needs to be controlled during the execution of the project by drilling down the WBS (Vanhoucke, 2012). For each project control approach, we therefore calculate the control effort as:

$$\frac{\sum_{i=1}^{n}\sum_{k=1}^{K}\bigvee_{j=1}^{J^{\#}} S_{ijk}}{nK} \quad (12)$$

where $J^{\#}$ again represents the number of control points in the selected control procedure and S_{ijk} represents whether or not a signal is produced in simulation run i, for control point j at a time period k.

4. Results

Section 4.1 shows the general performance of the five project control approaches on the 900 project networks dataset. The network structure of a project can influence the performance of each of the project control approaches, as explored in Section 4.2. The robustness of the different approaches is discussed in Section 4.3.

4.1. General performance

The *general performance experiment* parameter vectors of Table 7 were applied to the 900 project networks dataset and the simulation results (with $n = 10,000$ Monte Carlo simulation runs on each project and for every scenario) were analysed. The parameter vectors $(\omega_1, \omega_2, \omega_3)$ were assigned to model both the dependencies between activities (ω_R) and the variation (ω_V) in the simulation experiment. We combined ω_2 with a parameter vector chosen from $\{\omega_1, \omega_2, \omega_3\}$ and assigned one of them to ω_R and the other to ω_V. In a second experiment, the role of the parameter vectors are switched and Table 8 shows the averaged results over these simulations.

The mean and standard deviation for the resulting distribution of $d_i/\widehat{d}_i$ are given in the upper part of Table 8. The lower part of Table 8 presents the average number of control points used in each project control procedure ($\overline{J^{\#}}$), and the efficiency (Eq. (11)) and control effort (Eq. (12)) that has been recorded for each control procedure. For all project networks and each general performance simulation scenario, we have chosen the deadline $\widehat{RD}$ as 1.3PD. The "virtual" buffers $\%pb_\alpha$ and $\%fpb_\alpha$ and the α^{th} sample quantiles were chosen such that the detection performance for all project control approaches could be kept fixed at 100%. The average number of control points used in a project control procedure $\overline{J^{\#}}$, is also calculated. For the EVM-FPB and EVM-SNB procedures, we have set the number of feeding path buffers and subnetwork buffers that are controlled simultaneously to those values such that the efficiency of the project control approach is maximised.

A general trend that can be seen from Table 8 is that the efficiency of project control procedures seems to increase drastically for an increasing mean of the activity duration distribution. This observed increase is the most profound for the control procedures that deploy project specific statistical tolerance limits, calculated from the additional simulation runs prior to the execution of the project (EVM-STL, EVM-SNB). At the same time, the control effort for these project control approaches decreases, while the effort increases for the other control approaches (EVM-1PB, EVM-FPB) or remains stable (EVM-LPB). We observed that given an increased mean in activity durations, the probability of meeting a given deadline will decrease, since the mean of the distribution of the real project duration can be expected to shift to larger values (Elmaghraby, 2000). In this scenario, it is then advantageous to use statistical tolerance limits, in order to produce warning signals at the right moments in the project, which ultimately reduces the project control effort.

Table 8 shows that the EVM-LPB method outperforms the other project control approaches in terms of the efficiency. This is due largely to a reduced probability of encountering false warning signals, which means that the EVM-LPB method is less likely to produce a signal when the project has a makespan that is smaller than the project deadline. This can be expected since the

Table 8
General performance experiment comparison of project control procedures.

	ω_R and ω_V chosen from:								
	(ω_1, ω_2)			ω_2			(ω_2, ω_3)		
$E[d_i/\widehat{d}_i]$	0.7			1.0			1.3		
$sd[d_i/\widehat{d}_i]$	0.38			0.3			0.38		
	$\overline{J^{\#}}$	Eq. (11) (%)	Eq. (12) (%)	$\overline{J^{\#}}$	Eq. (11) (%)	Eq. (12) (%)	$\overline{J^{\#}}$	Eq. (11) (%)	Eq. (12) (%)
EVM-1 PB	1	23	27	1	43	51	1	78	61
EVM-STL	1	12	31	1	43	27	1	78	23
EVM-FPB	3.1	50	26	2.9	54	37	1.9	83	43
EVM-SNB	1.1	12	33	1.2	46	27	1.4	80	25
EVM-LPB	1	58	100	1	55	100	1	82	100

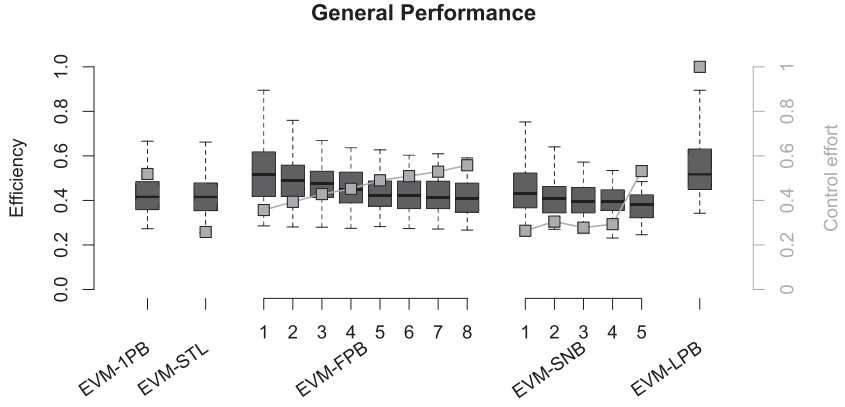

Fig. 12. General comparison of the project control procedures (for $\omega_R = \omega_V = \omega_2$).

EVM-LPB approach uses dynamic activity level data to calculate the expected project duration at each time instance during the project execution. However, the increased efficiency comes at a high cost of maximal control effort (100%) because the activity level performance needs to be updated at each control period during the execution of the project. Consequently, from Table 8, we conclude that the EVM-FPB approach is able to approximate the level of efficiency of the EVM-LPB approach, with much less the control effort. The maximal number of control points for the EVM-FPB approach can become very large, although Table 8 shows that on average only 2–3 control points need to be monitored simultaneously to produce these results.

Fig. 12 shows the observed efficiency and control effort for the second general performance scenario (where ω_R and ω_V are equal to ω_2). The number of control points is varied from low to high values for both the EVM-FPB and EVM-SNB methods. The boxplots show the distribution of the observed values of the efficiency, for which the axis is depicted to the left of the figure. The control effort is displayed using lightgrey boxes and is referenced against the axis on the right of Fig. 12. Overall, the same conclusions can be drawn from these as from Table 8. The EVM-FPB approach approximates the efficiency of the EVM-LPB approach with a smaller corresponding control effort. For the EVM-FPB and EVM-SNB approaches, the performance in shown in relation to the number of control points used, denoted along the x axis. Fig. 12 shows that the control effort increases for both the EVM-FPB and EVM-SNB approaches for increasing number of control points, while their efficiency decreases. This decrease in efficiency is due to the higher observed probability of encountering a false warning signal when multiple control points are used. For EVM-FPB, the increase in control effort is immediate, as soon as an additional control point is used alongside the control point placed at the project buffer to control the critical path. For the EVM-SNB approach, this effect only shows when more than 4 control points are used in combination with the control point at the project buffer.

In conclusion, Table 8 and Fig. 12 show that for the two control procedures, EVM-FPB and EVM-SNB, an optimal efficiency and control effort is obtained when only the deterministic critical path is monitored at the project buffer. For the EVM-FPB approach, the efficiency is comparable to the best efficiency observed in our study, which required dynamic calculations of the longest path in the project (EVM-LPB), for only a fraction of the control effort. However, the recorded efficiency and the control effort for all studied control approaches were produced using an index α. This α allowed us to ensure that the project control approaches all have a detection performance that equals 100%, while potentially reducing the probability of encountering false warning signals. Thereby, the efficiency of the control approaches could be maximised. The distributional choice made for ω_R and ω_V in the first stage simulation however, might influence the sensitivity of the control methods in the latter stage. Consider a real-life practical environment, where historical data is used to fuel a first simulation and consequently, through statistical analysis, tolerance limits (indexed by α) are calculated. If these tolerance limits are to be used in a practical dynamic control process, it is of utmost importance to know how sensitive these limits are to possible under- or overestimations in the distributional characteristics of the activity durations. In order to check this sensitivity, or rather the robustness against the distributional assumptions made, we conducted the robustness experiment in Section 4.3.

4.2. Impact of network structure

The impact of the network structure of a project on the project control performance is studied in this section. The data from Fig. 12 is restructured in Fig. 13 to show the relation of the efficiency and control effort to the underlying network structure of the project. Therefore, the serial/parallel (SP) indicator is denoted on the x axis. The efficiency for all control approaches is shown at the top of Fig. 13 and the control effort is show at the bottom. The closer a project network is to a full serial network, the closer the SP is to 100%. If all activities are allowed to be executed in parallel, the project network is said to resemble a full parallel network and the SP is equal to 0%. Fig. 13 confirms the earlier reported characteristic of

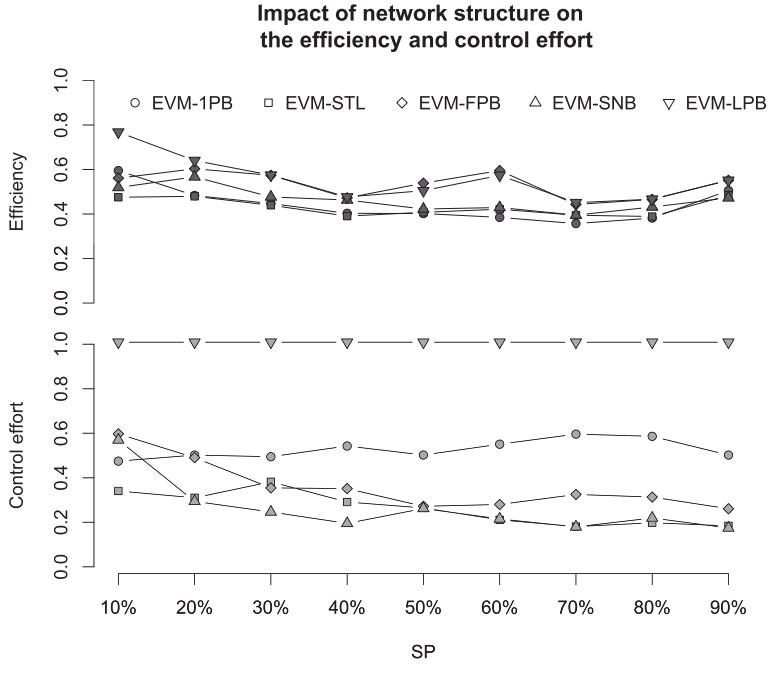

Fig. 13. Influence of the serial/parallel indicator on the project control procedures.

EVM that it becomes less efficient to obtain activity level performance data (EVM-LPB) for more serial network structures (Vanhoucke, 2010b). The efficiency of the other project control approaches remains relatively stable for all SP values. The control effort however, remains stable only for the EVM-1PB approach, and decreases for all other project control approaches (EVM-STL, EVM-FPB, EVM-SNB) for increasing SP values. This also confirms the findings of Vanhoucke (2010b, 2012) with respect to the improvement in the project control capabilities of the EVM/ES system for more serial network structures.

4.3. Robustness experiment

In this robustness experiment, the sensitivity of the project control approaches to possible under- or overestimations of the characteristics of the underlying distribution for the activity durations is tested. All the presented results for the efficiency and the control effort are calculated using tolerance limits which are indexed by α which optimises the efficiency, while keeping the detection performance fixed at 100%. Therefore, the tolerance limits are subject to the result of a simulation run, prior to the project execution. Consequently, errors in the estimates for the standard deviation and the mean of the activity durations might have an impact on the performance on the tolerance limits derived from these estimates. Fig. 14 shows the efficiency and project control effort for the EVM-1PB, EVM-STL, EVM-FPB, EVM-SNB and EVM-LPB control approaches. Two x-axes represent the changes in the mean and the standard deviations, which we will discuss separately in the remainder of this section. The two axis on the left of Fig. 14 are respectively used to reference the efficiency (at the top) and the control effort (at the bottom) against.

Mean. The left side of Fig. 14 shows the influence of a change in the mean on the performance of the project control procedures. The tolerance limits (indexed by α) are calculated from a statistical analysis of the simulation output where $\boldsymbol{\omega}_{2\mu_2}$ ($\mu = 1$) is assigned to both $\boldsymbol{\omega}_R$ and $\boldsymbol{\omega}_V$. The second stage simulation has another parameter vector ($\boldsymbol{\omega}_{2\mu_1}$, $\boldsymbol{\omega}_{2\mu_2}$, $\boldsymbol{\omega}_{2\mu_3}$, $\boldsymbol{\omega}_{2\mu_4}$ or $\boldsymbol{\omega}_{2\mu_5}$, with corresponding means $\mu = 0.7$, 1, 1.3, 1.6 or 1.9) assigned to model either the risk $\boldsymbol{\omega}_R$ or the variation $\boldsymbol{\omega}_V$. Consequently, Fig. 14 shows the averaged performance where either the risk or variation estimates are wrong. Fig. 14 confirms the earlier observation that the overall efficiency increase when the mean of the activity duration distribution increases. The project control procedures (EVM-STL and EVM-SNB) that require the calculation of sample quantiles prior to the execution of the project perform significantly worse than those that use the "virtual" buffers to derive tolerance limits (EVM-FPB and EVM-LPB). This effect shows most clearly when the mean of the activity distribution decreases. Moreover, we can again conclude that the EVM-FPB approach approximates best the observed efficiency of the EVM-LPB approach with significantly less control effort. This control effort for the EVM-FPB approach decreases with decreasing mean, and increases with increasing mean of the activity duration distribution.

Standard deviation. In order to test the potential effect of an under- or overestimation of the standard deviation of the distributions used to calculate the tolerance limits, the previous procedure was replicated with the parameter vectors $\boldsymbol{\omega}_{2s_1}$, $\boldsymbol{\omega}_{2s_2}$, $\boldsymbol{\omega}_{2s_4}$, $\boldsymbol{\omega}_{2s_5}$ and $\boldsymbol{\omega}_{2s_6}$, with a standard deviation of $\sigma = 0.15$, 0.2, 0.28, 0.35, 0.42 or 0.5. The tolerance limits where calculated using a distribution ($\boldsymbol{\omega}_2$) that has a standard deviation of 0.3, assigned to both the risk ($\boldsymbol{\omega}_R$) and the variation ($\boldsymbol{\omega}_V$) uncertainty modelling. Consequently, the simulations with $\boldsymbol{\omega}_{2s_1}$, $\boldsymbol{\omega}_{2s_2}$ or $\boldsymbol{\omega}_{2s_3}$ are used to model underestimates of the real standard deviation of the activity duration distribution, while the simulations with $\boldsymbol{\omega}_{2s_4}$, $\boldsymbol{\omega}_{2s_5}$ or $\boldsymbol{\omega}_{2s_6}$ are used to model overestimates. Fig. 14 again shows how that the EVM-FPB method approximates the efficiency of the EVM-LPB approach for varying values of the real standard deviation of the activity duration distribution. The observations for the other project control approaches are described along the following lines. The EVM-1PB project control approach seems to be less efficient for overestimates of the standard deviation and more efficient for underestimates. For the control procedures that require the calculation of sample quantiles prior to the execution of the project (EVM-STL and EVM-SNB), an opposite effect is observed. These

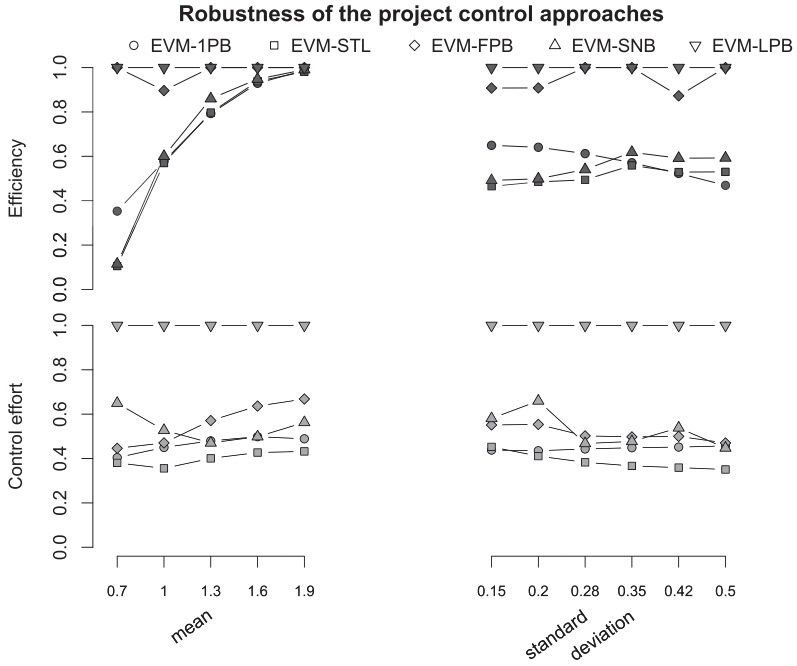

Fig. 14. Influence of a change in the mean and the standard deviation on the project control procedures.

control procedure are more efficient for overestimates, while their efficiency decreases for the underestimates of the real standard deviation of the activity duration distribution. Fig. 14 also shows that the control effort for the EVM-1PB, EVM-STL, EVM-FPB and EVM-SNB procedures decreases slightly when the real standard deviation of the activity duration distribution increases.

5. Discussion and conclusions

In order to minimise the effort spent by the project manager during the project control process, a top down earned value management/earned schedule (EVM/ES) method can be applied. However, the precise level of the WBS at which the EVM/ES control process should be conducted has been the subject of debate in literature. In this paper, we investigate five project control procedures using EVM/ES with different control points in the project. EVM-1PB calculates EVM/ES performance measures at the top WBS level and corresponds to the traditional use of EVM/ES. The EVM-STL control approach uses the same control point, but compares actual performance to statistical tolerance limits. The EVM-LPB control approach calculates the expected project duration from EVM/ES performance measures calculated for the dynamic longest path at each time instance during the execution of the project. In this research, we test two additional EVM/ES control approaches, inspired by the concept of buffers from the critical chain/buffer management (CC/BM) methodology. Buffers, which include baseline information in a structured manner will act as control points at which EVM/ES performance measures are calculated. Control points are added at each non-critical path feeding into the critical path in the EVM-FPB control approach. Since this can lead to a high number of control points, a second approach is presented to reduce the number of control points by adding control points on subnetworks buffers in the EVM-SNB approach. For both the EVM-FPB and EVM-SNB control approaches, we presented a recursive search algorithm to determine the number of control points and to find their place in the baseline schedule. Subsequently, a procedure was proposed to formulate tolerance limits for both control approaches. These tolerance limits were used in dynamic project progress simulations to produce warning signals, which indicated whether the project was likely to meet a given deadline or not. All the project control approaches were tested using a large computational experiment that includes a broad range of simulated dynamic project progress situations. They were then compared based on their efficiency and the required control effort.

The research methods used in this paper are either in line with recent literature or provide a valuable extension. Three main strengths of the proposed method are discussed along the following lines. First, the computational experiment was performed on a well known, large dataset of projects, which allows us to produce generalisable results. Second, the dynamic project progress simulations, performed for this research, deployed an uncertainty modelling for the activity durations in two stages. Linear association was chosen to represent the risk factor, in addition to variation modelling through probability density functions. Third, an extensive comparison of both newly developed project control approaches (EVM-FPB and EVM-SNB) and approaches from the literature (EVM-1PB, EVM-STL, EVM-LPB) was presented. The weaknesses of the research methods with which this study was conducted are summarised along the following lines. In order to allow a practical implementation of the proposed control approaches, historical data should be available within an organisation. It is also required that this data can be interpreted and calibrated to the uncertainty modelling that we propose in this paper. In addition, a minimum of theoretical knowledge on project scheduling theory should be present in the organisation, in order to calculate a baseline schedule for the project that needs to be performed. Finally, the results of in this paper are believed to be generalisable, but are subjected to some restrictions (i.e. results are only shown for the SPI(t)) and some adaptations that were made to the methods available in literature (i.e. the concept of "virtual" buffers is applied to ensure that all the control approaches have a detection performance that is equal to 100%).

The contribution to the theory on project management and control is threefold. First, we presented an introduction of the theoretical basis to the CC/BM methodology to the practice of project control using EVM/ES systems. Second, we have characterised two newly proposed control approaches (EVM-FPB and EVM-SNB) and compared them against alternatives. Moreover, this study also presents the first characterisation of the EVM-LPB method in a large computational experiment. Third, we have shown the mathematical basis on how to apply EVM/ES at different control points in a project baseline schedule and different ways in which tolerance limits can be calculated to produce warning signals for project control.

The practical implications of our study should also be given some thoughts. The results of our study shows that all EVM/ES project control procedures require some calibration (either by using the concept of "virtual" buffers or by the calculation of sample quantiles) to some prior data of project executions. This translates in this study to a detection performance that is equal to 100%. This means that not a single project execution, that does not meet the predefined deadline, goes unnoticed by the project control approaches. We have shown that the EVM-LPB control approach is the most efficient. However, this approach requires a dynamic recalculation of the longest path at each review period during the execution of the project. It therefore places a heavy burden on the project manager in terms of control effort. In order to reduce this control effort, our study shows that the EVM-FPB method should be preferred. Even when only the critical path is monitored during the execution of the project, the efficiency of the EVM-FPB method already approximates that of the EVM-LPB method. Especially when is dealt with projects with a more serial network structure, it becomes more beneficial to incorporate only the critical activities on a single EVM/ES control chart.

In order to draw further conclusions on the practical implications from this study, we advise that future research expands on the introductory experiment that is presented in this paper. A model that incorporates corrective actions could refine the measure for the efficiency that is proposed in this paper. This model could then investigate whether the warning signals that are produced by the control approaches lead to a better decision making in the project control practice. Moreover, a case study could be implemented to test whether, from an EVM/ES accounting perspective, it would be attainable to switch from the control account-oriented approach to a structure that is dictated by the baseline schedule of the project. In addition, the preliminary findings on the concept of "virtual" buffers for the critical path and the feeding paths could be expanded. It could then be tested whether these can be translated to physical buffers in order to build contingencies into the baseline schedule.

References

AbouRizk, S., Halpin, D., & Wilson, J. (1994). Fitting beta distributions based on sample data. *Journal of Construction Engineering and Management, 120*, 288–305.

Anbari, F. (2003). Earned value project management method and extensions. *Project Management Journal, 34*(4), 12–23.

Batselier, J., & Vanhoucke, M. (2014). Construction and evaluation framework for a real-life project database. *International Journal of Project Management*. http://dx.doi.org/10.1016/j.ijproman.2014.09.004.

Bowman, R. A. (2006). Developing activity duration specification limits for effective project control. *European Journal of Operational Research, 174*(2), 1191–1204.

Cheng, M.-Y., & Roy, A. F. (2010). Evolutionary fuzzy decision model for construction management using support vector machine. *Expert Systems with Applications, 37*(8), 6061–6069.
Cheng, M.-Y., & Wu, Y.-W. (2009). Evolutionary support vector machine inference system for construction management. *Automation in Construction, 18*(5), 597–604.
Chou, J.-S., Chen, H.-M., Hou, C.-C., & Lin, C.-W. (2010). Visualized EVM system for assessing project performance. *Automation in Construction, 19*(5), 596–607.
Colin, J. & Vanhoucke, M. (submitted for publication). *A multivariate approach to statistical project control using earned value management*. (For an update, see www.projectmanagement.ugent.be).
Colin, J., & Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega The International Journal of Management Science, 49*, 107–122.
Demeulemeester, E., Vanhoucke, M., & Herroelen, W. (2003). Rangen: A random network generator for activity-on-the-node networks. *Journal of Scheduling, 6*, 17–38.
Elmaghraby, S. (2000). On criticality and sensitivity in activity networks. *European Journal of Operational Research, 127*, 220–238.
Fleming, Q., & Koppelman, J. (2010). *Earned value project management* (3rd ed.). Newton Square, PA: Project Management Institute.
Goldratt, E. (1997). *Critical chain*. Great Barrington, MA: North River Press.
Golenko-Ginzburg, D., & Laslo, Z. (2001). Timing control points via simulation for production systems under random disturbances. *Mathematics and Computers in Simulation, 54*(6), 451–458.
Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207*, 1–15.
Herroelen, W., & Leus, R. (2001). On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management, 19*, 559–577.
Hyndman, R. J., & Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician, 50*(4), 361–365.
Jacob, D., & Kane, M. (2004). Forecasting schedule completion using earned value metrics? Revisited. *The Measurable News, Summer, 1*, 11–17.
Kelley, J., & Walker, M. (1959). *Critical path planning and scheduling: An introduction*. Ambler, PA: Mauchly Associates.
Kuhl, M. E., Lada, E. K., Steiger, N. M., Wagner, M. A., & Wilson, J. R. (2007). Introduction to modeling and generating probabilistic input processes for simulation. In S. Henderson, B. Biller, M. Hsieh, J. Shortle, J. Tew, & R. Barton (Eds.), *Proceedings of the 2007 winter simulation conference* (pp. 63–76). New Jersey: Institute of Electrical and Electronics Engineers.
Lee, S. H., Peña-Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction, 15*(1), 84–97.
Liang, T.-F. (2010). Applying fuzzy goal programming to project management decisions with multiple goals in uncertain environments. *Expert Systems with Applications, 37*(12), 8499–8507.
Lipke, W. (2012). Speculations on project duration forecasting. *The Measurable News, 3*, 3–7.
Lipke, W., & Vaughn, J. (2000). Statistical process control meets earned value. *CrossTalk: The Journal of Defense Software Engineering*, 28–29 (June:16–20).
Lipke, W., Zwikael, O., Henderson, K., & Anbari, F. (2009). Prediction of project outcome: The application of statistical methods to earned value management and earned schedule performance indexes. *International Journal of Project Management, 27*, 400–407.
Loch, C., De Meyer, A., & Pich, M. (2006). *Managing the unknown: A new approach to managing high uncertainty and risk in project*. New Jersey: John Wiley and Sons Inc.
McBride, W. J., & McClelland, C. W. (1967). Pert and the beta distribution. *IEEE Transactions on Engineering Management* (4), 166–169.
Moslemi Naeni, L., & Salehipour, A. (2011). Evaluating fuzzy earned value indices and estimates by applying alpha cuts. *Expert Systems with Applications, 38*(7), 8193–8198.
Partovi, F. Y., & Burton, J. (1993). Timing of monitoring and control of cpm projects. *IEEE Transactions on Engineering Management, 40*(1), 68–75.
R Core Team (2013). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.
Rand, G. K. (2000). Critical chain: The theory of constraints applied to project management. *International Journal of Project Management, 18*(3), 173–177.
Raz, T., & Erel, E. (2000). Optimal timing of project control points. *European Journal of Operational Research, 127*(2), 252–261.
Steyn, H. (2000). An investigation into the fundamentals of critical chain project scheduling. *International Journal of Project Management, 19*, 363–369.
Steyn, H. (2002). Project management applications of the theory of constraints beyond critical chain scheduling. *International Journal of Project Management, 20*, 75–80.
Trietsch, D. (2005). The effect of systemic errors on optimal project buffers. *International Journal of Project Management, 23*(4), 267–274.
Trietsch, D., & Baker, K. R. (2012). Pert 21: Fitting PERT/CPM for use in the 21st century. *International Journal of Project Management, 30*(4), 490–502.
Trietsch, D., Mazmanyan, L., Govergyan, L., & Baker, K. R. (2012). Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation. *European Journal of Operational Research, 216*, 386–396.
Tukel, O. I., Rom, W. O., & Eksioglu, S. D. (2006). An investigation of buffer sizing techniques in critical chain scheduling. *European Journal of Operational Research, 172*(2), 401–416.
Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2005). The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics, 97*, 227–240.
Van Slyke, R. (1963). Monte Carlo methods and the PERT problem. *Operations Research, 11*, 839–860.
Vanhoucke, M. (2010a). Measuring time – improving project performance using earned value management. *International series in operations research and management science* (Vol. 136). Springer.
Vanhoucke, M. (2010b). Using activity sensitivity and network topology information to monitor project time performance. *Omega The International Journal of Management Science, 38*, 359–370.
Vanhoucke, M. (2011). On the dynamic use of project performance and schedule risk information during project tracking. *Omega The International Journal of Management Science, 39*, 416–426.
Vanhoucke, M. (2012). Measuring the efficiency of project control using fictitious and empirical project data. *International Journal of Project Management, 30*, 252–263.
Vanhoucke, M. (2014). Integrated project management and control: First come the theory, then the practice. In *Management for professionals*. Springer.
Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research, 187*, 511–524.
Wauters, M., & Vanhoucke, M. (2014). Support vector machine regression for project control forecasting. *Automation in Construction, 47*, 92–106.
Williams, T. (1995). What are PERT estimates? *Journal of the Operational Research Society, 46*, 1498–1504.