

BUFFER SIZING FOR THE CRITICAL CHAIN PROJECT MANAGEMENT METHOD

A. Geekie¹ and H. Steyn²

¹Sasol Technology Project Management
Sasol Technology, South Africa
adrian@geekiegroun.com

²Department of Engineering and Technology Management
University of Pretoria, South Africa
herman.steyn@up.ac.za

ABSTRACT

Methods for sizing project and feeding buffers for critical chain project management are investigated. Experiments indicate that - in the absence of bias, and for certain classes of bias - buffer consumption is independent of the mean duration of a chain. Generally the popular method - a buffer size equal to 50% of the longest path leading to it - gives rise to excessively large buffers. Buffers sized according to the square root of the sum of squares perform well in the absence of bias, but with bias present the performance is unacceptably poor. A new approach to buffer sizing is proposed.

OPSOMMING

Metodes vir groottebepaling van projek- en saamvloeibuffers vir kritieke-ketting projekbestuur word ondersoek. Eksperimente dui daarop dat - in die afwesigheid van onewewigtigheid, en vir sekere tipes onewewigtigheid - bufferverbruik onafhanklik is van die gemiddelde lengte van 'n ketting. Oor die algemeen veroorsaak die metode van buffergrootte - gelyk aan 50% van die langste pad wat tot die buffer lei - onnodige groot buffers. Buffers bepaal met die metode van die vierkantwortel van die som van kwadrate, vaar goed in die afwesigheid van onewewigtigheid, maar vaar onaantoonbaar swak wanneer onewewigtigheid teenwoordig is. 'n Nuwe metode vir die bepaling van buffergrootte word voorgestel.

¹ The author was enrolled for the MPM programme at the Department of Engineering and Technology Management, University of Pretoria

1. INTRODUCTION

In 1997 Goldratt proposed a new approach to project management in his novel *Critical Chain* [1], which outlines the application of the Theory of Constraints to project management. Goldratt's method is known as *Critical Chain Project Management* (CCPM).

CCPM is intuitively appealing and appears to offer a number of advantages over the traditional scheduling and control methods. These include the elimination of special cause variation as well as the aggregation of project risk in a project buffer; the protection of the critical chain by way of feeding buffers; the late starting of non-critical chains; the use of resource buffers to warn of impending work; and buffer monitoring for managing project schedules. Figure 1 illustrates the key features of a critical chain project schedule for a single project. The critical chain is identified as the longest path through the network, taking into account both resource and logic dependencies. The estimated durations are shorter than those for a critical path schedule because risk allowances have been removed at task level and inserted at project level in the project buffer. The project buffer protects the promised project delivery date from delays in the estimated project schedule. Feeding buffers are inserted wherever non-critical chains join the critical chain. These protect the critical chain from delays in non-critical chains. Resource buffers are inserted at every point where work passes from one resource to another on the critical chain. Unlike the feeding and project buffers, these do not contribute to the overall schedule duration: they are simply warning windows. Non-critical paths are pushed to their late start, less the feeding buffer duration. Finally, buffer monitoring and control is instituted.

Appropriate sizing and management of buffers are crucial to the successful implementation of CCPM. This paper contributes to CCPM by providing a review of existing buffer sizing and management methodologies; by investigating the impact of various properties of chains of tasks on the extent of buffer consumption; and also by comparing the performance of two well-known buffer sizing methods through Monte Carlo simulation. The research is restricted to single project execution and limited to feeding and project buffers. Multiple projects, capacity, drum, and resource buffers are not investigated.

The objectives of this paper are, first, to review buffer sizing and management techniques available to project managers for use in CCPM, to determine the properties of path completion time distributions that impact on the extent of buffer consumption for cases with and without bias, and second, to compare the performance of two buffer sizing techniques through modelling for cases with and without bias.

2. EXISTING BUFFER SIZING AND BUFFER MONITORING METHODS

A review of CCPM literature revealed seven buffer sizing methodologies and three approaches to buffer management. Each of these is summarized below.

2.1 The 'cut and paste' method (C&PM)

The first buffer sizing method is the one proposed by Goldratt in *Critical Chain* [1]. Reduce the duration of each activity by 50% and let the buffer size equal one half the duration of the longest path feeding into the buffer. This implies a 25% reduction in project duration. Tukel, Rom and Duni Eksioglu [2] refer to this as the cut and paste method (C&PM). Leach [3] refers to this as the '50% of the Chain' method, and clarifies by stating that one should not count gaps in the chain or path when applying this method. Leach cites the advantages and disadvantages of this method. The two major advantages are that it is simple to apply and that it usually provides a large enough buffer. The disadvantage is that it does not allow one to account for known variation in the feeding path.

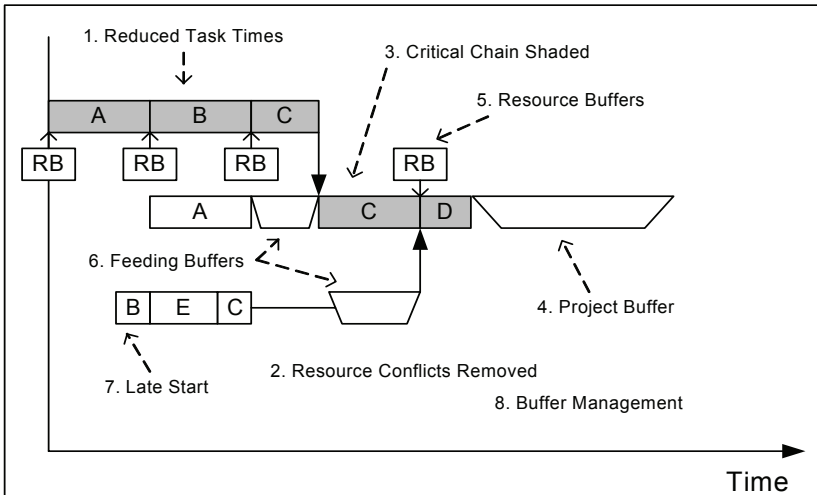


Figure 1: Key features of CCPM schedules [3]

2.2 The square root of the sum of squares method (SSQ)

In this method the size of the buffer is set to the square root of the sum of the squares of the difference between the low risk duration and the mean duration for each task along the chain leading to the buffer. Leach [3] states that, if the path branches upstream, one should use only the longest chain or the largest result considering each chain. Since one is specifically trying to account for the variation in task durations, adopting the largest result, considering each chain is probably the preferred approach, even though it will necessarily involve more computation. Leach does not specify what the low risk duration is for a task. Presumably it would be a duration with a probability greater than 90% of being achieved. Tukel et al [2] refer to this method as the root square error method (RSEM). The advantage of the SSQ method is that it allows one to account for known variation in task duration; the disadvantage is that it may lead to undersized buffers for long chains.

2.3 Bias plus SSQ method

This method is a combination of the *50% of the Chain* method and the SSQ method. The buffer is made up of a fixed portion added to the calculated SSQ buffer. The fixed portion accounts for variation that includes bias but is much smaller than 50% of the chain. Bias is any factor that can make projects take longer than planned but not shorter than planned [3]. Leach [3] provides some guidelines for the sizing of the fixed or bias portion of the buffer, but adds that experience should be used in determining this part of buffer size. Leach’s guidelines are summarized in Table 1. It should be noted that Leach states that these allowances are not necessarily additive.

Cause of bias	Range of buffer size
Omissions	Not specified
Path merging (more than five parallel paths)	Up to 20%
Errors	5%-20%
Special cause variation	0%-30%
Failure to report necessary rework	0%-20%

Table 1: ‘Bias’ buffer sizing [3]

Leach [3] further recommends that the project buffer should not be less than 25% of the critical chain. He urges caution if a path has fewer than ten activities, or if one activity constitutes more than 20% of the duration of a path.

2.4 Adaptive procedure with resource tightness (APRT)

Tukel et al [2] propose two buffer sizing methods. The first method attempts to account for what the authors define as ‘resource tightness’. Essentially they make their buffer equal to the standard deviation of the path leading to the buffer, scaled by a factor that is calculated by taking resource tightness into account. The buffer size is calculated as:

$$Buffer_Size = K \times \sigma_{feeding_path} \quad (1)$$

Where

$$K \propto \frac{r}{Rav} \quad (2)$$

where r is resource usage and Rav is resource availability. The feeding path standard deviation is calculated assuming the applicability of the central limit theorem, which states that the mean duration of the path or chain is equal to the sum of the mean durations of the tasks making up the chain, and the variation of the chain is equal to the sum of the variations of the tasks making up the chain [4]. The standard deviation is simply the square root of the variation.

2.5 Adaptive procedure with density (APD)

The second additional method that Tukul et al [2] propose attempts to account for the extent of precedence relationships in a network. The authors argue that for a given number of tasks the likelihood of a delay increases as the number of precedence relationships increases. Stated differently, there is a greater level of interdependence between the tasks and, if one task is delayed, all of its successors will be delayed. The authors refer to the number of precedence relationships as the density of the network. They attempt to account for density effects through the application of this buffer sizing method. Again the buffer is set to the standard deviation of the path scaled by a factor. This time the factor is based on the network density, and is defined as:

$$K = 1 + \frac{TOTPRE}{NUMTASK} \quad (3)$$

where *TOTPRE* is the total number of precedence relationships on the sub-network under consideration, and *NUMTASK* is the number of tasks on that sub-network. The buffer size is then given by:

$$Buffer_Size = K \times \sigma_{feeding_path} \quad (4)$$

where σ is the standard deviation of the longest path on the sub-network under consideration.

2.6 Buffer proportional to relative dispersion

Shou and Yeo [5] suggest that all activities should be placed into one of four classifications, which they arbitrarily designate A, B, C, and D. A is said to have a *very low* level of uncertainty, B is said to have a *low* level of uncertainty, C is said to have a *high* level of uncertainty, and D a *very high* level of uncertainty. The authors suggest that the activities be classified based on their '*relative dispersion (RD)*', which is defined as:

$$RD = \frac{\sigma}{t_e} \quad (5)$$

where σ is the activity's standard deviation and t_e is the activity's mean duration. Unfortunately the authors do not state what range of RD values falls into each of the categories A, B, C, and D. Nonetheless, once activities have been classified they can be assigned a buffer in accordance with recommendations made by Shou et al (and one's appetite for risk). Shou et al's [5] recommendations of buffer size for three different levels of risk are summarized in Table 2. In all cases the percentages are mean task duration.

Classification	Low safety	Median safety	High safety
A	4%	8%	12%
B	12%	24%	36%
C	20%	40%	60%
D	28%	57%	85%

Table 2: Buffer sizes for different classes of activity [5]

2.7 Method to account for systemic errors

Trietsch [6] argues that the assumption of independence of individual task estimates, on which the *RSEM* or *SSQ* method presented above is based, is not valid. The independence assumption, he says, leads to “a highly counterintuitive and damaging conclusion that project buffers should become relatively negligible for projects with long chains of activities”. He also argues that, even when activity durations are independent by nature, they can still be subject to the same estimation error or bias. The following examples are cited. If the same optimist or pessimist estimates the activity durations, then all durations will be underestimated by the optimist and overestimated by the pessimist. Alternatively, pressure from management may result in all estimators giving estimates that are too short. Bad weather or the loss of a key employee could affect several activities on a project; and, finally, a booming economy could increase queuing time for several activities on the project.

Trietsch [6] presents a model to account for the *systemic errors* or bias as discussed above for a single chain of activities with no branching. His method provides a lower limit on the buffer size, in contrast to the *RSEM* or *SSQ* method which will give a relatively negligible buffer size as the number of tasks approaches infinity (buffer size relative to path mean duration). The mathematics of Trietsch’s model is beyond the scope of this research.

2.8 Buffer monitoring and schedule control

Herroelen and Leus [7] provide the following concise summary of CCPM buffer monitoring and schedule control: “As activities are completed, managers keep track of how much of the buffers are consumed. As long as there is some predetermined proportion of the buffer remaining, everything is assumed to go well. If activity variation consumes a buffer by a certain amount, a warning is raised. If it deteriorates past a critical point, corrective action should be taken.”

Essentially, therefore, buffer monitoring is based on two trigger points. If buffer consumption is below the first trigger point, do nothing, but continue to monitor the buffer. When the first trigger point is reached, take note, and plan recovery actions. If buffer consumption continues to increase and passes a second trigger point, implement the recovery plans. Trigger points can be absolute or relative [3]. This is best illustrated graphically (refer to Figure 2). The relative buffer management approach can be further refined such that trigger points are closer together when a low proportion of the chain has been completed, and further apart when a high proportion of the chain has been completed [3]. In other words, the warning zone is

narrow early in the project (when there is still scope for many things to go wrong) but wide as the project nears completion (when few items that can go wrong remain). This promotes early action to conserve the buffer early in the project, and less immediate intervention late in the project when the need to conserve the buffer has diminished.

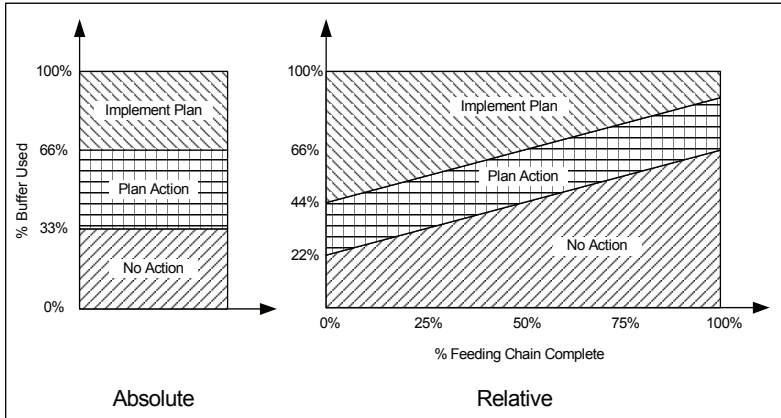


Figure 2: Buffer monitoring and schedule control

In addition to relative trigger points, a buffer consumption trend line can be plotted and action can be triggered on the basis of the trend. If, for example, the rate of buffer consumption increases, or if three consecutive measurements show an increase in buffer consumption, action can be taken [3].

The discussions to this point have summarized buffer sizing and buffer management methods available to project managers. In the sections that follow, properties of chains of tasks and their impact on buffer consumption are investigated, and a comparison is presented between the performance of the C&PM and the SSQ buffer sizing methods.

3. BUFFER CONSUMPTION AS A FUNCTION OF PATH MEAN DURATION AND STANDARD DEVIATION

3.1 The effect of mean duration and standard deviation of duration

The first experiment sets out to investigate the extent to which the mean duration of a path (or chain) and the standard deviation of that mean duration impact on buffer consumption. Modelling is first carried out for a case without bias, and is then extended for specific classes of bias. This experiment (along with other experiments in this research) is implemented in Microsoft Excel using Monte Carlo simulation software [8]. Path duration is modelled as a random variable with a normal distribution. In other words, the central limit theorem is assumed to be applicable. The path duration is subjected to Monte Carlo simulation over three thousand

cycles. The buffer is defined to commence at the path mean duration so that any modelled duration in excess of the mean duration is defined as consuming part of the buffer. The extent of buffer consumption, when it occurs, is calculated by subtracting the path mean duration from the modelled duration. Note that this calculation is only performed when buffer consumption occurs; one is not interested in cases where the buffer is not consumed. The model measures the extent of buffer consumption when buffer consumption occurs. The following tests were conducted with this model:

1. Mean duration of a path was held constant at forty time units, and path standard deviation was varied between one and ten time units. This test was designed to demonstrate the relationship between mean buffer consumption and standard deviation of path duration.
2. Standard deviation was held constant while mean duration was varied. Mean durations of twenty, forty, sixty, and one hundred time units were tested. The tests were repeated for different standard deviations. Standard deviations of one, five, and ten time units were used. This test was designed to demonstrate the relationship between path mean duration and buffer consumption.
3. The validity of the relationships identified in the first two sets of tests was tested by conducting tests on fifty 'random' paths or chains. The paths were given random mean durations of between zero and seventy time units, and standard deviations of between zero and thirty five time units.

The following results were generated with the model and tests described above:

1. As expected, mean and maximum buffer penetration increase with increasing standard deviation of the feeding path.
2. The extent of buffer penetration is completely independent of the path mean duration and depends only on the standard deviation. Buffer penetration is linearly related to standard deviation of the feeding path. This implies that a path with a mean duration of 5 and a standard deviation of 2 will have exactly the same mean buffer penetration as a path with a mean duration of 70 and a standard deviation of 2. The mean buffer penetration will obviously be a much larger percentage of 5 than it is of 70, but the absolute extent of mean buffer penetration will be identical in both cases.
3. In the absence of bias (i.e. variation is completely stochastic - there is no bias favoring or pushing towards longer durations) mean buffer penetration is independent of mean path duration. Mean duration has no impact on the extent of buffer penetration - it is the standard deviation of the feeding path alone that determines the extent of buffer penetration. Buffers should therefore *not* be sized on the basis of path mean duration, and (in the absence of bias) the practice of *basing* buffer sizes on the mean path duration makes no sense whatsoever.
4. Because mean buffer penetration is linearly related to standard deviation of the feeding path (point 2 above), buffer penetration percentage is linearly

related to *relative dispersion* of the feeding chain, where mean buffer penetration percentage is defined as mean buffer penetration divided by path mean duration and multiplied by one hundred, and relative dispersion as defined in equation 5.

5. The last result was found by serendipity, and there might be a mathematical explanation for it. It was hypothesised that the mean of the right hand side of the distribution might be equal to the mean buffer penetration (or perhaps mean of the right-hand side minus overall mean). However, numerical integration proves that mean buffer penetration percentage is equal to the difference between the mean of the right hand side and the mean of the left hand side of the normal distribution feeding the buffer, divided by the overall mean of that distribution, and expressed as a percentage. The next paragraph elaborates on this finding.

The last result presented above was discovered after observing the linear relationship between mean buffer penetration percentage and relative dispersion. An attempt was made to calculate an exact relationship. Initially it was thought the mean of the right hand side of the normal distribution would equate to the mean buffer penetration. This proved not to be the case. However, trial and error tests with the numerical integration set up to test the initial thinking revealed the relationship described in point 5 above. The means of the left and right hand sides of the normal distribution were calculated using the following relationship [9]:

$$\mu = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b x.e^{-\frac{(x-\mu)^2}{2\sigma^2}} .dx \quad (6)$$

The mean of the left hand side was calculated by setting the lower limit of the integral to six standard deviations less than the mean, and the upper limit of the integral equal to the mean. The mean of the right hand side was calculated by setting the lower limit equal to the mean, and the upper limit equal to six standard deviations greater than the mean. This is an approximation in both cases since positive and negative infinity should be used instead of the six standard deviations as applied here. A closed form solution to the integral is not possible, so numerical integration was performed using the trapezoidal rule from Swokowski [10]:

$$\int_a^b f(x)dx \approx \frac{b-a}{2n} [f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)] \quad (7)$$

In Figure 3 two sets of data are plotted. The first is a straight line obtained by numerical integration (by subtracting the mean of the left hand side of the distribution from the mean of the right hand side of the distribution) and therefore represents the theoretical exact solution. In order to generate the ‘exact’ line, the numerical integration had to be repeated for a range of feeding chain relative dispersions (from 0 to 0.5) and a linear regression had to be performed on the

output data. The second data set is the result of simulation 50 random paths, each with a random mean of between 0 and 70 time units and a random standard deviation of between 0 and 35 time units. The random paths were each subjected to 3,000 simulation cycles to test the proposition that the percentage of mean buffer penetration equals the difference between the mean of the right-hand side and the mean of the left-hand side of the feeding distribution. As the simulated data points closely match the theoretical proposition line, it is concluded that the proposition is true. The figure confirms the result described in point 5 above.

3.2 The impact of bias

The question arises as to whether or not the conclusions drawn in the previous section are valid when bias is present. Leach [3] describes bias very simply as “things that can make projects take longer but not shorter”. Essentially the methods presented and conclusions drawn above are based on the assumption that all task variation is stochastic; but in reality this assumption does not hold.

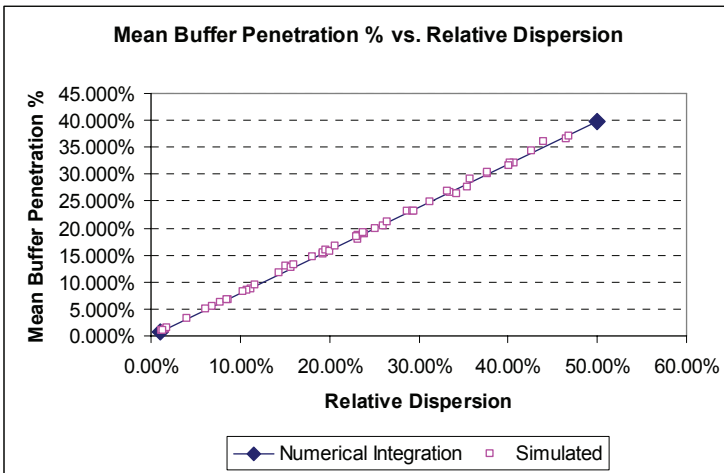


Figure 3: Mean buffer penetration % vs relative dispersion

The difficulty with testing bias is that bias can take on an infinite number of forms, and in virtually all cases the bias or amount of bias present is subjective. The results of the testing are, therefore, pre-determined by the modeller’s subjective decision on the extent of bias present. However, there is one form of bias which is not subjective and for which repeatable objective tests can be constructed. This form of bias is the merge point bias. Steyn [11] explains the merge point bias as follows. When two paths each have a probability of 50% of being completed on time, and a third activity or path can only commence when the first two are complete, the probability that the third activity will start on time is only 25%. (0.5x0.5). He further points out that, conventionally, many paths merge at the closure of a project, and

cites an example of only five paths merging for the completion of a project. In this case, if each path had a 50% probability of completion on time, the probability that the project would be completed on time is 0.55 - which is approximately 3%. In the case of this research, merge point bias was modelled by considering one critical chain having one main subordinate chain, and the main subordinate chain having three merging parallel chains. As per CCPM convention, only the main subordinate chain is buffered. The test configuration is illustrated in figure 4.

The tests presented above were repeated for merge point bias and two other subjective forms of bias. The first additional form of bias is termed 'gold plating bias'. In this form of bias, when a randomly simulated task duration is less than the mean duration, a random multiple of between zero and 1.5 times the difference between the mean duration and the simulated duration is added to the simulated duration. A realistic situation where this type of bias might occur is as follows. A software programmer completes his section of code ahead of schedule. Instead of passing it on to the next worker he decides to add an additional unnecessary visualization tool. He reasons that he will still finish ahead of the scheduled duration. Something then goes wrong, and in order to fix the problem he takes up to 1.5 times the time he would have saved had he just passed on the task when he completed it. Two sets of testing were conducted; one where this form of bias was applied at the task level, and one where this form of bias was applied at the chain level. The second additional form of bias is referred to as 'optimistic estimate bias'. In this form of bias it is assumed that the project scheduler has under-estimated the necessary task durations either as a result of optimism or as a result of pressure from management to cut project duration estimates. In other words, durations allowed for tasks are, for example, 90% of what they should really be.

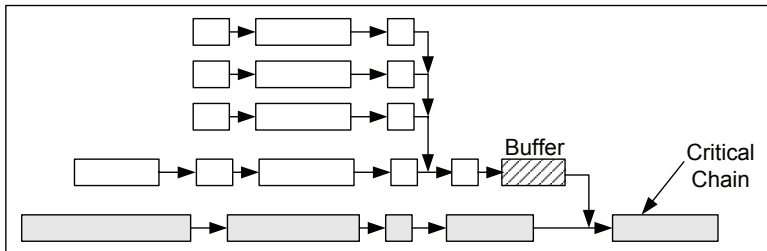


Figure 4: Illustration of merge bias

Here is a summary of the results obtained for the three forms of bias described above:

1. Mean path duration has no impact on the extent of buffer consumption for *merge point bias* and *gold plating bias* as defined here. As before, it is only standard deviation of the path that impacts on the extent of buffer consumption.
2. Buffer consumption increases with increasing feeding path standard deviations with bias present. This result is in line with results for no bias, which also

showed increasing buffer consumption with increasing standard deviation of path duration. The main difference with the bias cases is that the linear relationship between relative dispersion and buffer penetration percentage no longer holds.

3. Previously we showed that buffer consumption was completely independent of path mean duration in the absence of bias. With optimistic estimate bias present, mean buffer penetration is no longer independent of path mean duration. With optimistic estimate bias, buffer consumption increases with increasing path mean duration.

4. RELATIVE PERFORMANCE OF THE CUT AND PASTE METHOD (C&PM) AND THE SQUARE ROOT OF THE SUM OF SQUARES METHOD (SSQ)

The model and results described in sections 3.1 and 3.2 show that in the absence of bias - as well as for certain classes of bias - the extent to which chains consume their buffers is determined only by the path standard deviation and is completely independent of the path mean duration. This would indicate that sizing a buffer based on chain mean duration makes no sense at all. In other words, the C&PM method would appear to be nonsensical. However, there is a possibility that if certain forms of bias are introduced, the C&PM may perform better than the SSQ method and other more complex and difficult-to-apply buffer sizing methods.

The first model of this section sets out to test the C&PM against the SSQ method for the case where no bias is present. For this initial test a single chain or path is used. The number of tasks in the chain varies from five to ten, twenty, thirty, and finally forty tasks. Task durations are modelled using triangular distributions. For any given test all the tasks in a chain are identical, having the same minimum, most likely, and maximum durations. The variance in task duration, and hence the variance in chain duration, is altered by altering task minimum and task maximum duration while holding most likely and mean duration constant. Individual tasks are allowed to take on seven different variances. The tasks used in simulations are summarized in Table 3.

Task type	Min (a)	Most likely (c)	Max (b)	Task mean	Task variance
A	7	8	11	8.6667	0.722
B	6	8	12	8.6667	1.556
C	5	8	13	8.6667	2.722
D	4	8	14	8.6667	4.222
E	3	8	15	8.6667	6.056
F	2	8	16	8.6667	8.222
G	1	8	17	8.6667	10.722

Table 3: Task variance

For each chain or path length of five, ten, twenty, thirty and forty tasks, a set of simulations is conducted with each task type. In total, therefore, thirty-five sets of simulations are conducted. In each set of simulations the path mean duration is modelled, the percentage of times that the path buffer is exceeded is modelled, and the extent to which the buffer is exceeded (when it *is* exceeded) is modelled. Modelling is performed for both C&PM buffers and SSQ buffers.

The PERT method [12] was employed to compare the size of the SSQ and C&PM buffers to buffers with a particular level of certainty of not being exceeded. Essentially this was a 100% conventional application of the PERT method, with the only exception that task durations were modelled as triangular distributions rather than the special PERT-beta distribution. Table 4 summarizes the relative size of SSQ and C&PM buffers compared to particular certainty level buffers. Buffer size is expressed as a percentage of path mean duration.

	5 Task chains		40 Task chains	
Buffer Type	Type A Tasks	Type G Tasks	Type A Tasks	Type G Tasks
C&PM	50%	50%	50%	50%
SSQ	6.49%	23.4%	2.26%	8.28%
90% probable	5.62%	21.7%	1.99%	7.66%
99% probable	10.2%	39.3%	3.61%	13.90%

Table 4: Buffer size as a percentage of mean path duration

The table clearly shows that C&PM buffers are much larger than SSQ buffers, and that the difference in buffer size increases with increasing numbers of tasks in the chain. The table also indicates that C&PM buffers are excessive, given that they are much larger than buffers with a 99% probability of not being exceeded. The PERT calculation used to determine the 90% and 99% certain buffers can be reversed to calculate the level of certainty associated with a particular SSQ buffer size. This was done, and in all cases the SSQ buffer percentage certainty of not being exceeded was slightly in excess of 90%. This is as a result of the fact that the low risk task duration used in the calculation of the SSQ buffers was set to the task 90% probable duration. The simulated performance of the SSQ buffers matched the calculated PERT performance almost identically across all simulations for all task types and all numbers of tasks in the chains. The maximum discrepancy between simulated performance and calculated PERT performance was less than 0.85%. In respect of C&PM simulated buffer performance across all simulations for all task types and the full range of numbers of tasks in the chain, the C&PM buffer was exceeded in only one case. In the simulation for a five task chain with the highest variance task type, the C&PM buffer was exceeded in four out of 3,000 cycles, or 0.13% of the time. The conclusion one can draw from this is that, in the absence of bias, one would deliver 100% of projects on time if the C&PM buffer sizing approach was adopted. Unfortunately, one would never get to do any projects because of extremely uncompetitive promised deliveries. In the absence of bias, C&PM buffers are excessively and unnecessarily large and uncompetitive.

The above tests were extended to test the relative performance of the two buffer sizing methods in the presence of two forms of bias. The bias forms selected were optimistic estimate bias and gold plating bias. The two forms of bias were implemented exactly as described in section 3.2, except that in the case of gold plating bias the proportion of time added to the simulated duration was a factor of between zero and one times the difference between the mean and the simulated duration, rather than a factor of between zero and 1.5 times. The gold plating bias was applied at task level.

While the SSQ buffers performed well in the no bias case with only 7-8% of promised deliveries being missed, their performance was unacceptable for the optimistic estimate bias case. The best performance of the SSQ buffer was for the five task chain made up of the highest variance tasks. In this case the promised delivery was missed in 26% of cases. The worst performance occurred for the two lowest variance task types in the forty task chain and the lowest variance task type in the twenty task chain. In these three cases the promised delivery was missed in 100% of simulations. For optimistic estimate bias the SSQ buffer performance deteriorates for increasing number of tasks in the chain and reducing variance of the tasks. Even the best performance of 26% would be unacceptable to virtually all project managers. It is concerning to note that a comparatively small optimism factor of 10% leads to such a drastic collapse in performance of the SSQ buffers.

For the case of optimistic estimate bias, C&PM buffers were again almost never exceeded. Only in the case of the two highest variance task types for a five task chain was the C&PM promised delivery ever exceeded. The respective percentages of times that the C&PM promised delivery was not met were 0.2% and 0.6%. For all practical purposes the C&PM promised delivery was never exceeded in the case of the optimistic estimate bias.

The impact of gold plating is not as substantial as the optimistic estimate impact. The impact also shows a completely different trend. The best performance of the SSQ buffer occurred for the lowest variance task with five tasks in the chain. The promised delivery was exceeded in 11% of cases. The worst performance occurred for the highest variance task for forty tasks in the chain. The promised delivery was exceeded in 45% of cases. Generally speaking, in the case of gold plating bias, SSQ buffer performance deteriorates with increasing number of tasks in the chain and increasing task variance. The task variance effect is much less pronounced than is the case for optimistic estimate bias, and one could say that buffer performance remains relatively constant across the different task types.

The C&PM buffer performance in the case of gold plating bias was virtually perfect across all cases. Only in the case of the two highest variance task types in the five tasks in the chain was the C&PM promised delivery exceeded. The respective percentages of times that the promised delivery was exceeded were 0.03% and 0.16%. For all practical purposes the C&PM buffer was never exceeded.

In the presence of gold plating and optimistic estimate bias as defined here, one can conclude that SSQ buffers perform unacceptably, and that C&PM buffers remain excessive and uncompetitive. C&PM buffers may be appropriate in the cases of more extreme forms of bias.

5. CONCLUSION AND PROPOSED METHOD

The models presented in this research have demonstrated that in the absence of bias, and for certain forms of bias, buffer consumption is completely independent of feeding chain mean duration and is dependent solely on feeding chain standard deviation. Sizing buffers on the basis of the mean chain duration would, therefore, appear to be inadvisable. However, when certain forms of bias are introduced, a relationship between chain mean duration and the extent of buffer consumption does manifest itself. In these cases a buffer sized on the basis of chain mean duration would make sense. The best known buffer sizing method based on chain mean duration is the C&PM. Modeling presented here illustrates that C&PM buffers are unnecessarily large and uncompetitive. The performance of SSQ buffers in the absence of bias illustrates that they handle stochastic variation very successfully. However, the performance of SSQ buffers is unacceptably poor in the presence of bias. Given these observations, project schedulers should size buffers as a combination of a fixed portion proportional to the mean duration, but substantially smaller than the C&PM buffer and a variable portion calculated using the standard SSQ approach. This is essentially Leach's '*Bias plus SSQ*' approach, but - unlike Leach - this author would recommend sizing of the fixed portion on the basis of a database of previous project schedule performance for the organization concerned.

6. FURTHER RESEARCH

This research has made the specific findings summarized above, but has also demonstrated the usefulness of Monte Carlo simulations for modelling project schedule and buffer performance. As a general extension to this research, it is recommended that further simulations are undertaken to test the relative performance of some of the other buffer sizing techniques cited in this study. Additionally one could test the relative performance of different approaches to sizing the fixed portion of the combined SSQ / fixed portion buffer proposed above. This could perhaps lead to the development of a generic approach to sizing the fixed portion of the buffer. An investigation into the correlation between simulated performance and actual performance of project schedules would also be very useful. Finally, Monte Carlo simulation would also be a useful tool for investigating buffer *monitoring and control* approaches with a specific view to optimising intervention strategies.

7. REFERENCES

- [1] Goldratt, E.M. 1997. *Critical chain*. The North River Press, Great Barrington MA.
- [2] Tukel, O.I., Rom, W. R. & Duni Eksioglu, S. 2006. An investigation of buffer sizing techniques in critical chain scheduling, *European Journal of Operational Research*, 172, 401-416.
- [3] Leach, L.P., 2005. *Critical chain project management*. 2nd edition, Artech House, Inc., Norwood MA.

- [4] **Moder, J.J. & Philips, C.R.** 1985. *Project management with CPM and PERT*. Van Nostrand Reinhold Co., London.
- [5] **Shou, Y. & Yeo, K.T.** 2000. Estimation of project buffers in critical chain project management, *Proceedings of the IEEE international conference on management of innovation and technology*. ICMIT, 162-167.
- [6] **Trietsch, D.** 2005. The effect of systemic errors on optimal project buffers, *International Journal of Project Management*, 23, 267-274.
- [7] **Herroelen, W. & Leus, R.** 2001. On the merits and pitfalls of critical chain scheduling, *Journal of Operations Management*, 19, 559-577.
- [8] Software developed by Decision Support Services and available from <http://www.decisiontoolpak.com>
- [9] **Miller, I., Freund, J.E. & Johnson, R.A.** 1990. *Probability and statistics for engineers*. Prentice-Hall Inc., Englewood Cliffs, NJ.
- [10] **Swokowski, E.W.** 1998. *Calculus with analytic geometry*. Second alternate edition. PWS Kent Publishing, Boston MA.
- [11] **Steyn, H.** 2003. Comparisons between and combinations of different approaches to accelerate engineering projects, *South African Journal of Industrial Engineering*, 14 (2), 63-74.
- [12] **Malcolm, D.G., Roseboom, J.H., Clark, C.E. & Fazar, W.** 1959. Application of a technique for research and development program evaluation, *Operations Research*, 7 (5), 646-669.