The calculation of buffer size considering activity schedule risk

Nanfang Cui

School of Management, Huazhong University of Science & Technology
haojinying@126.com

Jinying Hao

School of Management, Huazhong University of Science & Technology

Abstract
A new method is proposed to calculate the buffer size based on the activity schedule risk. Compared with Root Square Error (RSE), the simulation experiments indicate that this method gets better robust performance with smaller buffer and avoids the critical chain rupture and non-critical chain overflow to a certain extent.

Keywords: Activity Schedule Risk, Critical Chain Project Management, Buffer Size

INTRODUCTION

Glodratt (1997) proposed the critical chain project management (CCPM) method for scheduling and controlling projects based on the theory of constraints (TOC), and it has proven to be a popular and effective approach regarding both project scheduling and project control under enormous complexity and uncertainty (Bevilacqua et al. 2009; Ma et al. 2014; Peng and Huang, 2014; Yang and Fu, 2014).

In CCPM, buffer management (BM) is a valuable control tool for coping with uncertainty and complexity. It can reduce project schedule risk due to the risk aggregation theory. Thus, buffers can deal with schedule variability during project execution and ensure the on-time completion of a project. In BM, feeding buffer (FB) is inserted where the non-critical chain and critical chain converge, hence, it can ensure the non-critical activities do not impact the start of critical chain activities. Furthermore, project buffer (PB) is placed at the end of the critical chain to ensure the project completion time. Therefore, the size of buffers means the manager's anticipation of uncertainty and it will directly determine the project completion time as well as the schedule risk. The vast majority of the research related to BM have concentrated on calculating buffer size (see Bie et al. 2012; Tukel et al. 2006) and improving the monitoring of the project schedule (Herroelen and Leus, 2005; Hu et al. 2015).

To calculate the buffer size, Monte Carlo simulation (Bevilacqua et al. 2009; Hoel and Taylor, 1999; Li, 1989; Schatteman et al. 2008; Zhang et al. 2015), fuzzy mathematics (Zuo 2010), queuing theory (Yang and Gao, 2012) and entropy (Bie et al. 2012; Zhang et al. 2014) were used in the aspect of activity duration. And more, Tukel et al. (2005) put forward two methods for determining feeding buffer sizes, one of which incorporated resource tightness while the other used network complexity.

About project monitoring, Herroelen and Leus (2001) included an activity crashing mechanism in their factorial experiment based on the three-stage buffer control system. Leach (2005) set buffer trigger points to track the actual execution of a project. Bie and Cui (2010) proposed a realistic buffer monitoring method by calculating the buffer size and time instant dynamically. In a more recent study, Zhang et al. (2015) used the grey prediction model to establish an effort buffer deviation monitoring and control model for software projects. Colin and Vanhoucke (2015) proposed two new project control approaches with multiple control points for the purpose of minimizing the effort spent, in which the earned value management/earned schedule method and buffer management are combined. And Hu et al. (2015) monitored the project by evaluating the probability of successful project completion relative to the crashing cost.

Though these buffer monitoring methods can well indicate the project schedule as a whole and decide whether to take control action or not, they failed to clarify which activities deserve more of management control. To deal with this defect, we introduce the activity schedule risk (ASK) method to calculate the buffer size. This method analyzes every activity and considers not only activity uncertainty but also the activity schedule risk in the whole process of project execution. Buffers are distributed to activities with high activity schedule risk which will have a high impact on project completion or contribute more to the delay of a project. Then the corresponding chain will get more FB and PB that will effectively improve the project schedule performance. The new ASR method is compared with RSE method (Newbold, 1998), whose advantages include that it can utilize known task variation and that it will not generate very large or very small buffer sizes based on the length of the chain according to Tukel et al. (2006) and Bie et al. (2012). Our computational studies show that, compared with the previous RSE method, our ASK method provides better performance with smaller feeding buffers.

The remainder of this paper is structured as follows. The second part explains the process of calculating activity schedule risk. The third part proposes an algorithm including using activity schedule risk to determine buffer size, and the integrated critical chain project management framework is propounded. The fourth section utilizes an example to illustrate the algorithm, including the detailed process of calculating buffers in ASK method and RSE method. In the fifth section, simulation experiments are executed to indicate the superiority in comparison. The final section includes the conclusions of this study.

ACTIVITY SCHEDULE RISK

The Definition of Activity Schedule Risk

Activity's risk consists of the risk occurring probability and its resulting loss cost. Managers must consider both probability and risk impact. To reduce the probability, most managers aim at improving schedules to get higher practicability and stronger robustness. Buffers can improve the stability of project schedule and reduce the risk of delay.

In this paper, activity schedule risk is defined to be the product of occurring probability and the cost of delay. Based on detailed analysis of every activity, buffers can be calculated reasonably, thus, mangers can take specific measures for each activity, and ensure the good project schedule performance.

The Calculation of Activity Schedule Risk

The following definitions will be used throughout this paper:

N	set of all activities
A	set of arrow line; it represents the precedence constraints
G	original project network (considering only precedence constraints and $G = (N, A)$)
R	arrow line set; it represents the resources flow precedence constraints
RG	resource flow network (considering available resources and $RG = (N, R)$)
$G^{\sim}$	the sum of G and RG ($G^{\sim}$=$G \cup RG$)
T	all precedence relationships in $G^{\sim}$(T= $A \cup R$)
i	activity i is activity j's precursor activity in T ($1 \leq i \leq n,\ 1 \leq j \leq n$)
D_i	the baseline duration of activity i
$F(j)$	the realized starting time of activity j
$S(j)$	the planned starting time of activity j
ω_j	unit cost of delay for activity j
$LPL(i,j)$	the longest path from activity i to activity j in $G^{\sim}$
RS_j	the schedule risk of activity j

The schedule risk of each activity reflects its hazard when the realized starting time is delayed. RS is defined in previous section and it can be calculated by multiplying the risk occurring probability by its resulting loss cost, as presented in the Eq. (1).

$$RS_j = rs_j \times \omega_j \tag{1}$$

rs_j reflects the delaying probability of starting time. It can be calculated by Eq. (2).

$$rs_j = P\big(F(j) > S(j)\big) = \sum_{(i,j)\in T(A\cup R)} P(D_i > S(j) - S(i) - LPL(i,j)) \tag{2}$$

RS can be calculated as follows:

Step 1: Use the branch-and-bound algorithm (Demeulemeester and Herroelen, 2014) to generate the initial baseline schedule S without considering resource flow network RG..

Step 2: RG is obtained based on the resource flow in S, then, $G^{\sim}$ is ascertained.

Step 3: Find out all direct and indirect precursor activity of activity j in the scope of $G^{\sim}$, then calculate $LPL(i,j)$.

Step 4: Calculate $P\big(F(j) > S(j)\big)$ based on $S(j)$, $S(i)$, $LPL(i,j)$ and D_i (subject to a lognormal distribution in this paper).

Step 5: Use Eq. (1) to calculate activity schedule risk RS_j, j runs from 1 to n.

THE CALCULATION OF BUFFER SIZE

In this paper, buffer is calculated based on activity schedule risk. This method firstly calculates the schedule risk of each activity. Then, limited amount of scattered buffers are added to activities with higher schedule risk. The high schedule risk activities may impact project completion more severely and contribute a great deal to the delay of project. Consequently, buffers to these special activities are more crucial and effective. The sub-problems of where to insert buffers and how many buffers should be added can be solved by one iterative algorithm (Cui et al. 2014). Then, according to CCPM, a critical chain and several non-critical chains are obtained. The buffers to protect individual activity are centralized to protect the chain to which the activity belongs. That is to say, scattered buffers belonging to the critical chain are centralized to calculate project buffer, and others pertaining to each non-critical chain are centralized respectively to calculate feeding buffers.

The algorithm to calculate PB and FB can now be stated as follows:

Step 1: Considering only sequence constraints, use the branch-and-bound method to generate a baseline schedule S which has the shortest completion time S_n^0. Set value of δ_n, the deadline of project, which should never be passed due to resulting unacceptable disaster.

Step 2: Considering both sequence constraints and resource constraints, calculate each activity schedule risk RS_j by the second part of section "ACTIVITY SCHEDULE RISK", and arrange them in decreasing order. Meanwhile, the sum of activity schedule risks $\sum_{j\in N} RS_j$ can be obtained. Add one unit of buffer to the activity with the highest RS, then both this activity and all its direct and indirect successor activities are delayed one unit of time. Update the schedule and planned starting time $S(j)$, then each schedule risk RS_j will be changed.

Step 3: Recalculate the scheduling completion time S_n, each activity schedule risk RS_j and the sum of activity schedule risk $\sum_{j\in N} RS_j$ in new schedule plan. If the new scheduling project completion time is not over the limit we set ($S_n \leq \delta_n$) and the new sum of all activity schedule risks does not increase, the new schedule plan proves to be available and it will be used as baseline schedule in next iteration. The progress will go to step 2, otherwise will go to step 4.

Step 4: Find out one activity j which not only has smaller RS than the all executed activities, but also has higher RS than activities which had not been executed in the previous process. If $RS_j = 0$, the algorithm will end, else add one unit of buffer to the activity j. Both activity j and its direct and indirect successor activities will be delayed one unit of time. Update the schedule and process goes to step 3.

Finishing the above four steps, the scattered buffers belong to each activity can be obtained. Then the critical chains and non-critical chains are obtained according to CCPM (Tian and Cui, 2009). When there is more than one critical chain, we can choose one critical chain optionally and others belonging to non-critical chains. Scattered buffers belonging to the critical chain are centralized to calculate project buffer, and others pertaining to each non-critical chain are centralized respectively to calculate feeding buffers. FB and PB can get though Eqs. (3) and (4).

$$FBI = \sum_{j\in NCCI} buffer(j) \tag{1}$$

$$PB = \sum_{j\in CC} buffer(j) \tag{1}$$

$NCCI$	the non-critical chain numbered I
buffer(j)	buffers belong to activity j
CC	the only critical chain
FBI	the feeding buffer of NCCI

In next section, an example is used to illustrate the whole process.

THE ILLUSTRATE OF ALGORITHM

The algorithm is illustrated by a project network pat 1 choosing from Patterson. Pat 1 consists of 12 non-dummy activities and consumes three kinds of resources with a constant availability of 2, 1 and 2 units. The project network is shown in Fig. 1.

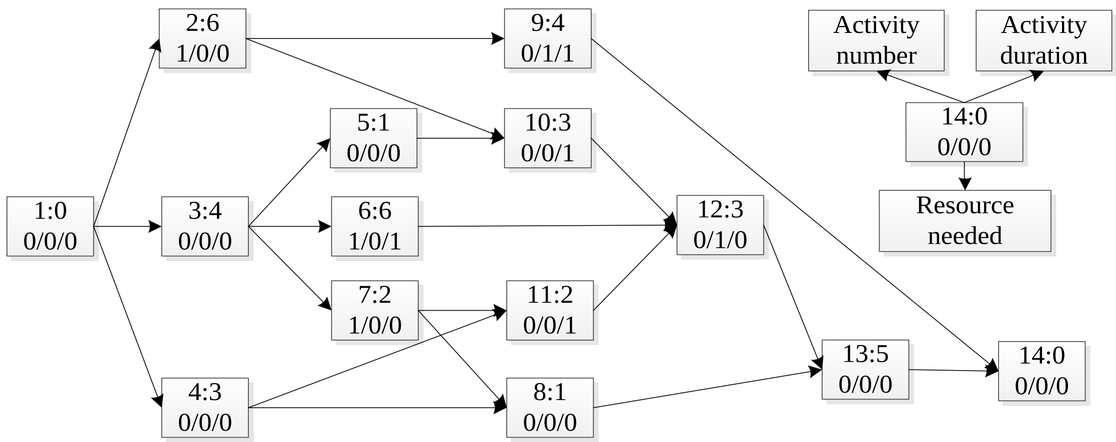

Figure 1 – Patterson pat1 project network

Branch-and-bound method is used to get baseline schedule with shortest completion time. The result of baseline schedule *S* equals to (0, 0, 0, 0, 4, 4, 6, 8, 14, 6, 9, 11, 14, 19), and the shortest completion time of pat 1 is 19. The Gantt chart of S is shown in Fig. 2.

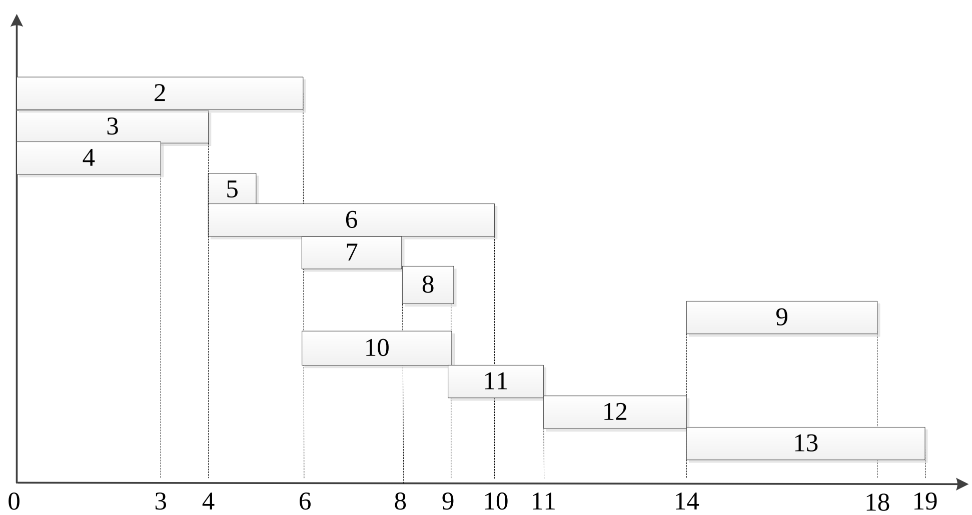

Figure 2 – Baseline schedule in branch-and-bound

Critical chain and non-critical chains can be obtained by the method proposed by Tian (2009). For pat 1, critical chain contains activities 1-2-10-11-12-13-14 and others activities belong to 6 non-critical chains. NCC1 contains activity 3 and activity 5, and it converges to activity 10. NCC2 contains activity 4, converges to activity 11. NCC3 contains activity 7, converges to activity 11. NCC4 contains activity 6, converges to activity 12. NCC5 contains activity 8, converges to activity 13. NCC6 contains activity 9, converges to activity 14.

Activity durations are supposed to follow a lognormal distribution. The expected values are durations in Fig. 1 and the standard deviation σ equals to 0.3 temporarily. The activity unit delay costs ω equal to (0, 3, 1, 4, 7, 9, 6, 6, 5, 2, 4, 3, 1, 10) and the deadline δ_n is supposed to be 25. According to the iterative algorithm in Section "THE CALCULATION OF BUFFER SIZE", buffers pertaining to every activity equal to (0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1). In details, activities 5, 6, 8, 9, 11, 12 and 14 gain one unit of buffer and others activities gain zero unit of buffer. Then FB and PB can be calculated like follows.

NCC1 {3-5}, *FB1=buffer (3) +buffer (5) =0+1=1*, and *FB1* is inserted to activity 10.

NCC2 {4}, *FB2.1=buffer (4) =0.* NCC3 {7}, *FB2.2=buffer (7) =0.* Both NCC2 and NCC3 converge to activity 11, so the buffer inserted to activity 11 equals to the max of *FB2.1* and *FB2.2.* In details, *FB2=max {FB2.1, FB2.2} =0.*

NCC4 {6}, *FB3=buffer (6) =1* and *FB3* is inserted to activity 12.

NCC5 {8}, *FB4=buffer (8) =1* and *FB4* is inserted to activity 13.

NCC6 {9}, *FB5=buffer (9) =1* and *FB5* is inserted to activity 14.

CC {1-2-10-11-12-13-14}, *PB=buffer (1) +buffer (2) +buffer (10) +buffer (11) +buffer (12) +buffer (13) +buffer (14) =0+0+0+1+1+0+1=3.*

The critical chain schedule is shown in Fig. 3.

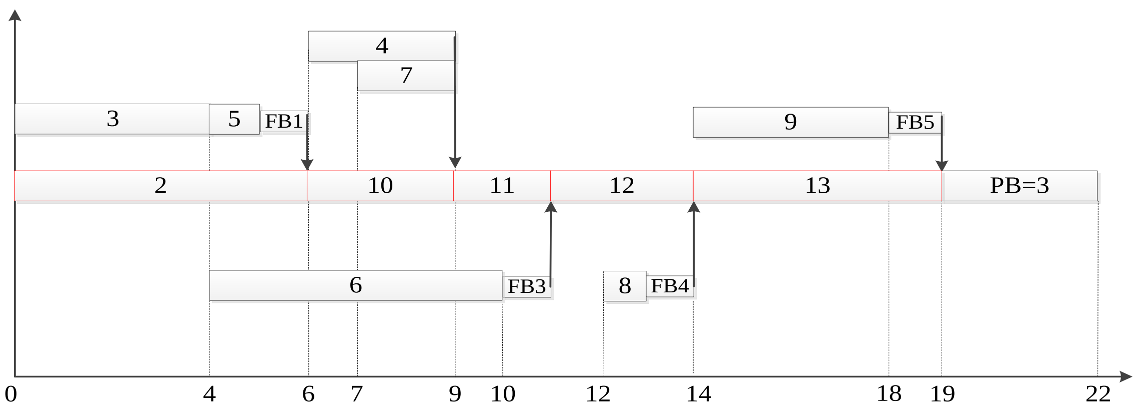

Figure 3 – The critical chain schedule of ASK method

To test the performance of new method, buffers are recalculated according to RSE method. Under same assumptions that activities durations follow a lognormal distribution, the expected values are durations in Fig. 1 and the standard deviation σ equals to 0.3, buffers can be obtained like follows.

$$FBI = \sqrt{\sum_{i\in I} VAR_i} \quad (5)$$

$$PB = \sqrt{\sum_{i\in CC} VAR_i} \quad (6)$$

VAR_i the variance of activity i's duration ($VAR_i = (D_i)^2 \times (\exp(\sigma^2) - 1)$)

σ_i the standard deviation of activity i.

According to Eqs. (5) and (6), buffers in RSE method are *FB1=2, FB2=1, FB3=2, FB4=1*, *FB5=2 PB=3.* After inserting buffers, new schedule can be obtained. New Gantt chart is shown in Fig. 4.

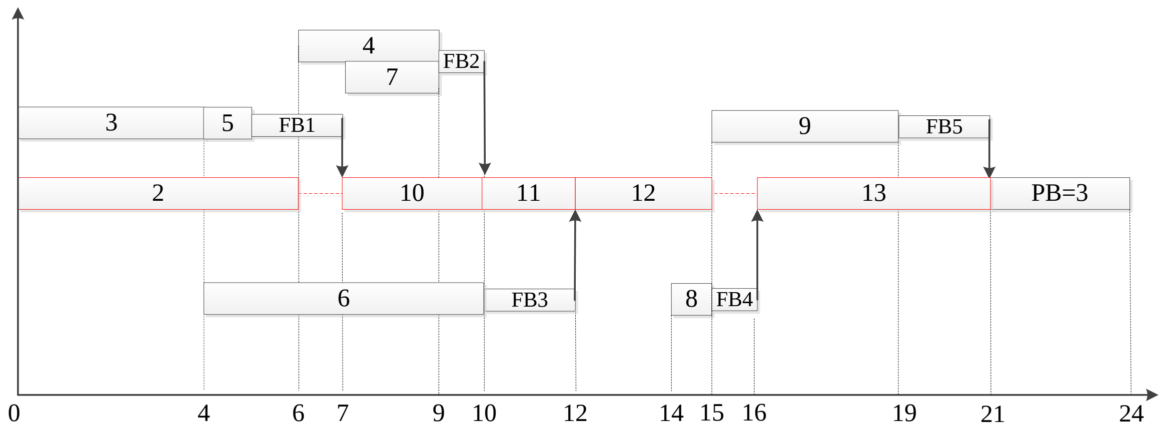

Figure 4 – The critical chain schedule of RSE method

As is shown in Fig. 4, critical chain rupture appears in the new schedule. It is the oversize of FB which causes resource conflict again.

SIMULATION EXPERIMENTS

Simulation experiments are executed to compare the new method which considers activity schedule risk with the traditional RSE method. The two different schedule plans are shown in Fig. 3 and Fig. 4. In practical situation, uncertainty will influence the stability of execution, thus, test of robustness performance is essential.

Experimental Environment

The three settings of the simulation are as follows:

(a) Activity durations probability distribution and simulation

To response the uncertainty, all activity durations are supposed to follow a logarithmic normal distribution, and the durations D_i in project network are expected value. In simulation, activity durations u_i are random that $u_i = ln(D_i) - \sigma^2/2$. It can be expressed by $lognrnd(u(i), \sigma)$ in MATLAB. Simulation experiments are executed which have three different degrees of uncertainty. In ascending order, the standard deviations are 0.3, 0.6 and 0.9.

(b) Project execution strategy

In CCPM, the execution follows roadrunner scheduling strategy in which activities will start as early as possible and activities belong to critical chain have the priority to start.

(c) Performance evaluation index.

Project completion time, *PCT*: This is a cost indicator referring to the simulated project completion time.

TPCPs: This indicator refers to the probability that a project is completed within the projected due date (deadline δ_n). The function is $TPCP = prob(S_N \leq \delta_n)$, which is efficiency indicators.

The Result of Simulation Experiments

To guarantee the compare of average completion rate is meaningful, the limit completion time δ_n must be the same in simulations. In this paper, $\delta_n = PB_{RSE\ \sigma=0.6} + S(n) = 6 + 19 =$

25. FB and PB are calculated in two methods, the results are shown in Table 1.

Table 1 – Buffer sizes under different σ

Standard Deviation	$\sigma = 0.3$ ($\delta_n = 25$)		$\sigma = 0.6$ ($\delta_n = 25$)		$\sigma = 0.9$ ($\delta_n = 25$)	
	RSE	ASK	RSE	ASK	RSE	ASK
FB1	2	1	3	0	5	0
FB2	1	0	2	1	4	0
FB3	2	1	4	1	7	0
FB4	1	1	1	2	2	3
FB5	2	1	3	2	5	3
PB	3	3	6	2	11	3
SUM of FB	8	4	13	6	23	6

The simulations are executed 1000 times, the average *PCT* and average *TPCPs* are shown in Table 2.

Table 2 – The result of average robustness

Standard Deviation	$\sigma = 0.3$ ($\delta_n = 25$)		$\sigma = 0.6$ ($\delta_n = 25$)		$\sigma = 0.9$ ($\delta_n = 25$)	
	RSE	ASK	RSE	ASK	RSE	ASK
PCT	23.4544	23.0636	25.1998	24.8379	28.3825	27.5616
TPCPs	0.8250	0.8930	0.6380	0.6530	0.5120	0.5390

Analyzing Table 1 and Table 2, two points are found:

(a) In situations with different uncertainties, ASK method can get smaller FB than RSE method.

(b) In situations with different uncertainties, ASK method can get better average completion time and average completion rate in pat 1. This conclusion is the same with Li Ming (2013) (Smaller FB will not cause prominent influence to robustness performance).

CONCLUSIONS

In this paper, ASK method calculates buffer sizes based on activity schedule risk. It considers activity uncertainty and evaluates schedule risk of every activity. To protect the project efficiently, limited amount of scattered buffers are added to high schedule risk activities. Using CCPM theory, a critical chain and several non-critical chains are obtained. Then, FB and PB can be calculated.

The experimental results indicate that the new method can get smaller average completion time and larger average rate of project completion with smaller buffers, while avoiding or reduce critical chain rupture and non-critical chain overflow. In conclusion, the new method utilizes smaller feeding buffers and project buffer to protect the project from delay and effectively reduce resource conflict when feeding buffers are inserted.

BIBLIOGRAPHY

Goldratt, E. M. 1997. Critical Chain. The North River Press, New York.

Bevilacqua, M., Ciarapica, F. E., Giacchetta, G. 2009. Critical chain and risk analysis applied to high-risk industry

maintenance: A case study. International Journal of Project Management 27(4): 419–432.
Ma, G., Wang, A., Li, N., Gu, L., and Ai, Q. 2014. Improved critical chain project management framework for scheduling construction projects. Journal of Construction Engineering and Management 140(12): 1–12.
Peng, W., Huang, M. 2014. A critical chain project scheduling method based on a differential evolution algorithm. International Journal of Production Research 52(13): 3940–3949.
Yang, S., and Fu, L. 2014. Critical chain and evidence reasoning applied to multi-project resource schedule in automobile RandD process. International Journal of Project Management 32(1): 166–177.
Yang, J. B. 2007. How the critical chain scheduling method is working for construction. Cost Engineering 49(4): 25–32.
Bie, L., Cui, N., and Zhang, X. 2012. Buffer sizing approach with dependence assumption between activities in critical chain scheduling. International Journal of Production Research 50(24): 7343–7356.
Tukel, O. I., Rom, W. O., and Eksioglu, S. D. 2006. An investigation of buffer sizing techniques in critical chain scheduling. European Journal of Operational Research 172(2): 401–416.
Herroelen, W., and Leus, R. 2005. Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research 165(2): 289–306.
Hu, X., Cui, N., and Demeulemeester, E. 2015. Effective expediting to improve project due date and cost performance through buffer management. International Journal of Production Research 53(5): 1460–1471.
Li, C. 1989. Improvements in Triangular Distribution in Monte Carlo Simulation Method. Chinese Journal of Marine Geology and Quaternary Geology 9(4): 97-105.
Hoel, K., Taylor, S. G. 1999. Quantifying Buffers for Project Schedules. Production and Inventory Management Journal 40(2): 43-47.
Damien, H., Herroelen, W., Stijn, V. V. 2008. Methodology for Integrated Risk Management and Proactive Scheduling of Construction Projects. Journal of Construction Engineering and Management 134: 885-893.
Bevilacqua, M., Ciarapica, F. E., Giacchetta, G. 2009. Critical Chain and Risk Analysis Applied to High-risk Industry Maintenance: A Case Study. International Journal of Project Management 27: 419-432.
Yang, X., Gao, P. 2012. Time Buffer Quantitative Study in Critical Chain Project Management. Chinese Journal of Shanghai Management Science 3: 62-66.
Zhang, J., Song, X., Diaz, E. 2014 Buffer Sizing of Critical Chain based on Attribute Optimization. Concurrent Engineering：Research and Applications 22(3): 253-264.
Herroelen, W., and Leus, R. 2001. On the merits and pitfalls of critical chain scheduling. Journal of Operations Management 19(5): 559–577.
Leach, L. P. 2005. Critical Chain Project Management (2nd ed.). Artech House.
Bie, L., and Cui, N. 2010. Research on dynamic buffer monitoring in critical chain project management. Chinese Journal of Management Science 18(6): 97–103.
Hu, X., Cui, N., and Demeulemeester, E. 2015. Effective expediting to improve project due date and cost performance through buffer management. International Journal of Production Research 53(5): 1460–1471.
Colin, J., and Vanhoucke, M. 2015. A comparison of the performance of various project control methods using earned value management systems. Expert Systems with Applications 42(6): 3159–3175.
Tian, W. Cui, N. 2009. The Identification of Critical Chain and Non-critical Chain in Critical Chain Project Management. Chinese Journal of Industrial Engineering and Management 14(2): 88-93.
Cui, N., Zhao, Y., Hu, X. 2014. The Buffer Setting Method of Robustness project Scheduling. Chinese Control and Decision 29(2): 368-372.
Demeulemeester, E., Herroelen, W. 2014. A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem. Chinese Management Science 38(12):1803-1818.
Tukel, O. I., Rom, W. O., Sandra, D. E. 2005. An investigation of buffer sizing techniques in critical chain scheduling. European Journal of Operational Research: 401-416.
Hu, H., Cui, N., Demeulemeester, E. 2015. Effective expediting to improve project due date and cost performance through buffer management. International Journal of Production Research 53(5): 1460-1471.
Li, M., Xu, Z., Yu, J. 2013. The Setting Method of Feeding Buffer in Critical Chain based on Parkinson's Law. Chinese Journal of Computer Integrated Manufacturing System 19(12): 3177-3183.