

## Research Article

# A DSM-Based CCPM-MPL Representation Method for Project Scheduling under Rework Scenarios

Guofeng Ma <sup>1</sup>, Ming Wu <sup>1</sup>, Keke Hao <sup>1</sup> and Shanshan Shang <sup>2</sup>

<sup>1</sup>Department of Construction Management and Real Estate, Tongji University, Shanghai 200092, China

<sup>2</sup>School of Business and Management, Shanghai International Studies University, 550 Dalian Road, Shanghai 201620, China

Correspondence should be addressed to Ming Wu; [wumingkey@126.com](mailto:wumingkey@126.com)

Received 9 July 2020; Revised 10 December 2020; Accepted 4 January 2021; Published 22 January 2021

Academic Editor: Chiara Bedon

Copyright © 2021 Guofeng Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rework risks caused by information flow interactions have become a major challenge in project scheduling. To deal with this challenge, we propose a model integrating the critical chain project management method, design structure matrix method, and max-plus method. Our model uses a start-to-start relationship of activities instead of the traditional finish-to-start relationship, which also allows overlaps between activities. We improve the accuracy of the rework safety time in two ways: (1) the overall overlapping effect is taken into consideration when calculating the rework time of an activity arising from the information flow interaction of its multiple predecessors overlapped with it; (2) the rework time arising from activity overlaps, the first rework time, and the second rework time are calculated as components of the rework safety time in our model, while the last one is ignored in traditional methods. Furthermore, the accuracy of time buffers is improved based on the improved rework safety time. Finally, we design the max-plus method to generate project schedules and appropriately sized time buffers. The empirical results show that the project schedule generated by the proposed method has a higher on-time completion probability, as well as more appropriately sized project buffers.

## 1. Introduction

Rework is defined as “the unnecessary effort of redoing a process or an activity implemented incorrectly the first time.” [1] Regarded as inevitable and epidemic in projects [2], rework has become a major impediment that adversely affects project performance. Many studies have shown the impact of rework risks on cost overrun and project delay. Love investigated 161 construction projects in Australia and concluded that rework was the main cause of overrun cost, which can reach 52% increased cost on average [3]. Similarly, a study by Barber et al. reported that rework costs can be as high as 23% of contract values [4]. Hwang et al. surveyed 381 projects in Singapore and found that more than 80% of corporations and 59% of projects have experienced client-related rework, resulting in a cost increase of 7.1% and a schedule delay of 3.3 weeks on average [5].

Considerable research studies have explored the causes of rework as it significantly impacts project performance.

Love argued that design and construction errors, as well as omissions and changes, are the main reasons of rework [3]. Ye et al. proposed several other reasons for rework involving contract management and management scope, etc. [6]. However, the most widely held view is that information flow interactions between activities should be responsible for rework [7, 8]. To reveal and model rework relationships among activities and other factors, several approaches have been developed such as cognitive mapping (CM) [9, 10] and system dynamics (SD) [11, 12].

Despite the awareness of the causes of rework risks, tools to manage rework and mitigate its adverse impact are still insufficient. Traditional project scheduling tools, such as the critical path method (CPM), project evaluation and review technique (PERT), and the Gantt chart (GC), have been widely used in project management for decades but are not applicable under rework scenarios, as they cannot deal with rework appropriately [13, 14]. For example, in the CPM network, the forward pass assumes that the maximum value

is propagated to as its successor start, and the backward pass assumes that it finishes as late as possible [15]. In this way, the so-called time difference can be treated as float to deal with rework or delays. Yet, such approach can only deal with part of delays. Due to these limitations, rework has not been managed or controlled effectively in practice, which commonly results in low efficiency of traditional project schedule management tools. This brings challenges for both practice and theory, thus appealing for novel methods to deal with such issue and predict the project schedule with consideration of rework scenarios. Therefore, the need exists for a new project scheduling approach to model rework relationships and establish reliable schedules under rework scenarios for effective project schedule prediction and control.

The design structure matrix (DSM), first proposed by Steward, proved to be a powerful tool in modeling rework relationships [16]. A DSM-based discrete-event Monte Carlo simulation approach was then presented by Browning and Eppinger [17] and expanded in later studies [18–20] to predict project duration. However, this approach is highly computationally demanding and time-consuming, also inapplicable for large projects. It is also unreasonable that an activity could not begin until all its precedent activities have been completed in the simulation process, which greatly limits its application. The critical chain project management (CCPM) method, first proposed by Rand and Goldratt [21], which applies the theory of constraints to project management, is another emerging and potential tool to manage information uncertainty by setting resource and time buffers in project schedules. Based on prior studies, Zhang et al. first combined CCPM with DSM to model rework relationships between activities and to address rework risks by setting time buffers that consider rework safety times [8]. However, this method calculates the rework safety time between an activity and its each predecessor, respectively. The overall effect of overlaps between an activity and its multiple predecessors on rework safety times as well as a learning curve is not considered, which yields the total rework safety time of an activity longer than required in practice. In addition, the second rework times are overlooked when calculating rework safety times. The activities in the generated project schedule meet  $FTS = 0$  relationship, which cannot effectively reflect the activity overlaps. Ma et al. developed a critical chain design structure matrix method to calculate the rework time [22]. However, all the rework times between an activity and its predecessors are added, which created a big project buffer. Therefore, further meaningful research is needed to integrate and refine current CCPM methods to better address rework risks. To sum, this study aims to predict a more accurate project schedule with consideration of rework scenarios and provide informative insights for practice.

In this study, we develop a quantitative model integrating a DSM into the CCPM-max-plus linear (CCPM-MPL) framework by considering various rework factors. The model considers the second rework time and the overall overlapping effect between a critical activity and its multiple predecessors when calculating rework safety time. The

formulas in the max-plus method to calculate time buffers using the root square error method (RSEM) are built and adjusted to generate project schedules with a start-to-start (STS) relationship. Project schedules generated with the proposed method absorb rework risks and other uncertainties, thereby enabling project practitioners to predict project duration more accurately.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature. Section 3 explains the proposed method in detail, including the project scheduling process with the max-plus method, the calculation model of rework safety times, and a refined buffer determination method. Section 4 evaluates the proposed method using an empirical analysis. Section 5 concludes this paper.

## 2. Literature Review

*2.1. Buffer Determination in the CCPM.* Ever since the CCPM method was first proposed, it has not only been increasingly applied in project scheduling but also been continually extended and refined [23]. There are three main types of buffers—the resource buffer (RB), the feeding buffer (FB), and the project buffer (PB)—which are distinguished by their respective locations and functions. RBs, which function as warnings and consume no time, are placed before critical activities to protect the chain from the risk of critical resource tightness [24]. FBs are placed at the convergence of the critical chain and noncritical chains to protect the chain from activity variants on noncritical chains [25]. PBs are placed at the end of schedules to protect against exceeding project durations [26]. The determination of buffer sizes depends on various factors, such as risk preferences of the project team, project complexity, personnel, and equipment capacities [27]. The cut-and-paste (CAP) method and RSEM are widely used among a number of buffer-sizing methods. The CAP method retains half of activity duration as safety time, estimates half of the sum of safety times on the critical chain to be the PB, and that of the noncritical chain to be the FB in order to guarantee project duration. However, as this procedure is linear, the buffer size increases linearly, which may cause an unnecessarily large amount of protection [28]. In order to improve the effectiveness of the CAP method, Shi and Gong proposed the RSEM, which is based on the large number law and the central limit theorem [29]. The RSEM calculates the square root of the total squares of the difference between the safety estimate time and the average estimate time for each activity on the critical chain as the PB and that on the noncritical chain as the FB. Zhang et al. argued that the RSEM outperformed the CAP method for large projects because the former was less affected by critical chain length [8]. Furthermore, considering that various project characteristics and attributes could influence buffer sizes, several modified buffer-sizing methods have been proposed. Tukul et al. [30] proposed the adaptive procedure with resource tightness (APRT) and the adaptive procedure with density (APD) methods to determine time buffers—both of which take resource tightness and network complexity into consideration. Shi et al. further calculated resource tightness using a fuzzy method and determined time buffers taking into

account network complexity and risk project managers' preference [31]. In addition, other researchers have also developed different kinds of buffer-sizing methods that consider other project attributes, including degree of project uncertainty [32], project network characteristics [33], and project teams' risk preferences [25]. In recent years, there has been some research focused on objective and quantitative float/buffer allocation to critical activities. With inspiration from voting theory, Su et al. created and validated a functioning method for float allocation that fairly mitigates risk [15]. The same authors further employed voting methods to apportion contract float to correlated activities in network schedules to projects from delays [34]. Table 1 lists the aforementioned buffer calculation methods.

Although great advances have been made on buffer-sizing methods, they all assume that project activities are independent and neglect the codependency of activities, as well as the effect of rework risks arising from information flow interactions on project duration. These factors resulted in smaller estimated project durations than necessary [8]. As previously mentioned, Zhang et al. proposed a refined DSM-based CCPM method, which took the codependency of activities and rework risks into account when determining buffer sizes [8], but this method still has several limitations on modeling rework and determining rework safety times, which deserves further research.

*2.2. DSM as a Process Management Technique.* DSM as a modeling tool for process architecture has been widely accepted and adopted in different application areas, such as project scheduling and product design [22, 38]. Process DSM uses matrix formation to visualize and analyze activities and logic relationships or dependencies between activities that include sequential, parallel, and coupled activities. In process DSM, an off-diagonal element  $A_{ij}$  represents information flow from activity  $j$  to activity  $i$  or dependency between the two activities. For a binary DSM, if the dependency between activity  $i$  and activity  $j$  exists,  $A_{ij}$  is often filled with symbol "x" or "."; otherwise, it will be null [39]. However, binary DSMs cannot effectively express the strength of information flows or dependencies between activities and are therefore limited to a qualitative analysis of process architecture. In order to overcome the deficiency, a numerical DSM, in which the bigger value of elements means a higher degree of dependency between activities, was developed to express dependencies between activities more accurately in order to quantitatively analyze process architecture [40]. Diagonal elements  $A_{ii}$  are used to represent a certain activity characteristic, such as duration of activity  $i$ , or are simply empty in most cases.

The matrix-based expression of DSMs is applicable for rework analysis, as it can clearly display rework-related, information flow interactivity [38]. Regarding DSMs, much related research has been performed to address rework risks. Browning and Eppinger first proposed a DSM-based discrete-event Monte Carlo simulation model to estimate project duration and cost [17]. They argued that processes with appropriate overlapping and rework,

instead of processes with the fewest feedback marks in the DSM, may lead to less project durations and reduced costs [38]. This framework was adopted by a number of subsequent studies and extended by considering additional constraints in process scheduling. For instance, Cho and Eppinger proposed a heuristic to schedule stochastic and resource-constrained projects under rework scenarios [18]. Yassine et al. proposed two kinds of genetic algorithm-based approaches to find near-optimal project durations for resource-constrained, multiple-project scheduling problems [39]. Overall, prior studies have explored the possibility and efficiency of using a DSM in project scheduling to predict and control rework and have reported significant progress. However, the DSM-based expression is hard to transfer to a traditional network diagram, which is commonly used in practice by project managers [41]. To apply a DSM effectively in project planning and scheduling, this paper developed an accurate DSM-based calculation model of rework duration and a DSM-based CCPM-MPL representation method.

### *2.3. The Effect of Overlaps between Activities on Rework Times.*

Zhang et al. introduced the concept of rework safety time, which is defined as the safety time caused by rework, to deal with rework risk [8]. The value of rework safety time is the total expected rework time of the activity. The initial safety time determined without considering rework is called activity safety time. The sum of activity safety time and rework safety time is called the total safety time.

Project teams face complex situations when an activity may be overlapped with multiple precedent activities, and information transmission from any one of them may cause its rework and affect its rework time. In prior studies [8, 18], the rework times caused by information transmission from its multiple predecessors were simply summed up to be the default rework amount. The calculation method is highly questionable because the accumulation of rework times may exceed the longest individual overlap time. Such overlap time should be the upper limit of rework time and occurs in the worst scenario in which the completed work has to be reworked completely. In order to relieve multiple predecessors' effect on rework calculation times, a formula developed by Dehghan is introduced and explained in the following section [42].

Furthermore, reworking an activity requires less effort compared to the first time in most cases because of the learning curve effect and the activity participants' adaption [17]; learning curves are utilized to accurately measure effects of rework risks on activity duration. Further study by Osborne found that reworked activities exhibited little further learning-curve effect after an initial rework [43]. Therefore, the learning curve for each activity is considered as a fixed value when calculating the rework safety time; it means that a fixed proportion of original activity duration will be taken for rework execution whether it is the first or subsequent rework.

TABLE 1: Buffer calculation methods.

Method	Buffer calculation
C&PM [35]	Buffer = (1/4) $\sum T_{90\%}$
RSEM [36]	Buffer = $\sqrt{\sum_{i=1}^n (S_i - A_i)^2}$ $S_i$ represents the duration with 90% completion probability $A_i$ represents the duration with 50% completion probability
APRT [30]	$r(i, q)$ represents resource requirements of $i$ for $q$ $Rav(q)$ represents the level of resource acquisition of $q$ $\sqrt{\sum_{i=1}^n VAR_i}$ represents the sum of duration variance of activity in the critical chain
APD [30]	$k = 1 + (TOTPRE/NUMTASK)$ Buffer = $k \cdot \sqrt{\sum_{i=1}^n VAR_i}$ TOTPRE represents the number of activity relationships in the chain NUMTASK represents the number of activities in the network
Buffer calculation considering resource tightness [37]	Buffer = $\sqrt{\sum_{i=1}^n [(1 + \alpha_i) \times \beta_i \times \sigma_{iy}]^2}$ $\alpha_i$ represents the physical resource tightness $\beta_i$ represents the information resource tightness $\sigma_{iy}$ represents the standard deviation of activities
Buffer calculation considering resource tightness, network complexity, and risk coefficient [31]	Buffer = $\tau [\sum_{i=1}^n [(1 + A(a_i) \times \beta_i \times \sigma_{iy})]^2]^{1/2}$ $\tau$ represents the buffer adjustment coefficient in the 5% risk level $A(a_i)$ represents the membership function of resource tightness $\beta_i$ represents the coefficient of the network schedule
Buffer calculation considering project rework [8]	Buffer = $\sqrt{\sum_{i=1}^n (ST_{0i} + ST_{Ri})^2}$ $ST_{0i}$ represents the safety duration of the activity $ST_{Ri}$ represents the safety time of rework

### 3. Methodology

In this section, we present project scheduling using a start-to-start relationship between activities with the max-plus method first. This part is the integration of CCPM, DSM, and max-plus method. DSM is used to represent the logic of activities, and the max-plus method is employed to decouple relationships among these activities to further determine the critical chain of the project. Based on these efforts, we then build a model that considers the learning curve effects and overlaps with multiple predecessors to calculate rework safety times accurately. Finally, we propose a refined buffer determination method that is expressed in max-plus algebra to generate appropriately sized time buffers and protect project schedules. Specifically, notations used in this section are listed in Table 2.

**3.1. Project Scheduling with a Max-Plus Method.** Four major operators of max-plus algebra are defined as follows. For two

variables  $x, y \in D$ , where  $D = R \cup \{-\infty\}$  and  $R$  is the real field,  $x$  and  $y$  satisfy

$$\begin{aligned}
x \oplus y &= \max(x, y), \\
x \wedge y &= \min(x, y), \\
x \otimes y &= x + y, \\
x \odot y &= -x + y.
\end{aligned} \tag{1}$$

For matrices  $X, Y \in D^{m \times m}$ , where  $[X]_{ij}$  and  $[Y]_{ij}$ , respectively, represent the  $(i, j)$  th elements of  $X$  and  $Y$ ,  $X$  and  $Y$  satisfy

$$\begin{aligned}
[X \oplus Y]_{ij} &= \max([X]_{ij}, [Y]_{ij}), \\
[X \wedge Y]_{ij} &= \min([X]_{ij}, [Y]_{ij}), \\
[X \otimes Y]_{ij} &= \bigoplus_{k=1}^m ([X]_{ik} + [Y]_{kj}), \\
[X \odot Y]_{ij} &= \bigwedge_{k=1}^m (-[X]_{ik} + [Y]_{kj}).
\end{aligned} \tag{2}$$

In addition, two basic unit elements for these operators are given by  $\varepsilon (\equiv -\infty)$  and  $e (\equiv 0)$ , and  $\otimes$  and  $\odot$  are prior to  $\oplus$  and  $\wedge$ . Previous studies on the CCPM-MPL framework have assumed that an activity could not begin until all its precedent activities were completed. That means, only the FTS relationship is taken into consideration in project scheduling, and no overlapping exists between activities. However, this assumption is highly questionable, considering the fact that precedent constraints between activities include, but are not limited to, the FTS relationship, and overlapping between activities is prevalent in actual projects [44]. Therefore, multiple relationships need to be considered in project scheduling and integrated into the CCPM-MPL framework to develop project schedules closer to actual circumstances.

Our work takes the STS relationship between activities into account by transforming it into a finish-to-finish (FTF) relationship, which is more suitable for embedment in a max-plus method. As shown in Figure 1(a), a STS relationship exists between activities  $A$  and  $B$  with the value of  $a$ , which means that activity  $B$  cannot start until activity  $A$  has started for at least  $a$  days, at which time  $a$  can be transformed into a corresponding value of the FTF relationship which is  $b$ , as shown in Figure 1(b). Let matrix  $F_{ss}$  represent the STS relationship between activities (see Figure 2(a)) and matrix  $F$  represent the corresponding FTF relationship (see Figure 2(b)). If the FTF relationship exists, the element is then filled with the corresponding value; otherwise, it will be filled with  $\varepsilon$ . The transforming process can be expressed as follows:

$$[F]_{ij} = [F_{ss}]_{ij} + [D]_{ii} - [D]_{jj}, \quad (3)$$

where  $D$  is the matrix of activity durations and filled by the diagonal elements with each activity's duration and the nondiagonal elements with negative infinity.

The CCPM-MPL framework adopted in this paper has been modified to make it more applicable to projects. First, as we consider the characteristics of projects, including uniqueness and one-time process, the start time of a certain project activity is regarded to be unaffected by completion of the same kind of activities in other projects previously completed. Second, for simplicity, external inputs and external outputs are not included in this study. Project scheduling processes in the CCPM-MPL representation are given in the following.

The earliest finish time for each activity is calculated as

$$x_E^+ = (F)^* \otimes D \otimes x_0, \quad (4)$$

where  $(F)^* = e_n \oplus F \oplus \dots \oplus (F)^{l-1}$ . The  $(i, j)$ th element of matrix  $(F)^*$  means the largest deviation of the finish time of activities  $j$  and  $i$  if activities  $i$  and  $j$  are on one or more paths at the same time; otherwise, it is  $\varepsilon$ . For simplicity, we set  $x_0 = \underbrace{(e, e, \dots, e)}_n^T$ , which is an  $n \times 1$  matrix and means the

beginning of activities in the project will not be affected by other projects with  $e$  representing zero.

TABLE 2: Notations in Section 3.

Name	Description
$A_{ij}$	Element in row $i$ and column $j$ of the design structure matrix
$F_{ss}$	Matrix represents the STS relationship between activities
$F$	Matrix represents the FTF relationship between activities
$\varepsilon$	Equal to $-\infty$
$e$	Equal to zero
$D$	Matrix represents activity durations
$x_0$	An $n \times 1$ matrix with all elements equal to zero
$x_E^+$	Earliest finish time of an activity
$x_E^-$	Earliest start time of an activity
$C_0$	A $1 \times n$ matrix with all elements equal to zero
$y_E$	Maximum of the earliest finish time for all activities
$x_L^-$	Latest start time of an activity
$[O]_{ij}$	Overlap duration between activity $i$ and $j$
$LC_i$	Learning curve of activity $i$
$[RT_0]_{ij}$	Rework time of activity $i$ caused by activity $j$
$[RT_1]_{ij}$	Rework time of activity $i$ caused by activity 1 to $n$
$[RT_2]_{ij}$	First rework time of activity $i$ caused by activity $j$
$ST_i$	Total safety time of activity $i$
$ST_{ri}$	Rework safety time of activity $i$
$ST_{oi}$	Activity safety time of activity $i$
$v'$	Location of feeding buffers

The earliest start time for each activity is calculated as the difference between its earliest finish time and duration according to the following equation:

$$x_E^- = D \odot x_E^+. \quad (5)$$

The corresponding output time, denoted as the maximum of the earliest finish time for all activities, is given by

$$y_E = C_0 \otimes (F)^* \otimes x_E^+, \quad (6)$$

where  $C_0 = \underbrace{(e, e, \dots, e)}_n$ , which is a  $1 \times n$  matrix with  $e$  representing zero. The latest start time for each activity is calculated as the difference between the project output time and the sum duration of activity  $i$ , as well as the duration of subsequent critical activities, according to the following equation:

$$x_L^- = [(F)^* \otimes D]^T \odot [x_0 \odot y_E]. \quad (7)$$

The total floats in all activities are obtained as

$$[TF]_i = [x_E^-]_i \odot [x_L^-]_i. \quad (8)$$

The critical chain is then determined by the set of activities  $\alpha$  that satisfy  $\{\alpha [TF]_\alpha = 0\}$ .

**3.2. Calculation of Rework Safety Times.** We define the rework safety time as the rework time caused by information flow interactions, including rework time arising from overlaps with precedent activities, the first rework times, and the second rework times. Therefore, the rework safety time of an activity is the sum of the three kinds of rework times above. Ignoring rework times when estimating the safety times may lead to smaller time buffers than necessary.

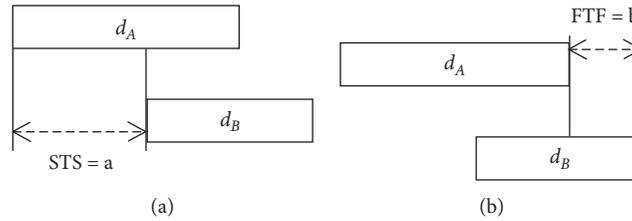


FIGURE 1: (a) Activities with the STS relationship; (b) the transformation of STS to the FTF relationship.

	A	B
A	$\epsilon$	$\epsilon$
B	a	$\epsilon$

(a)

	A	B
A	$\epsilon$	$\epsilon$
B	b	$\epsilon$

(b)

FIGURE 2: (a) Matrix  $F_{ss}$ . (b) Matrix  $F$ .

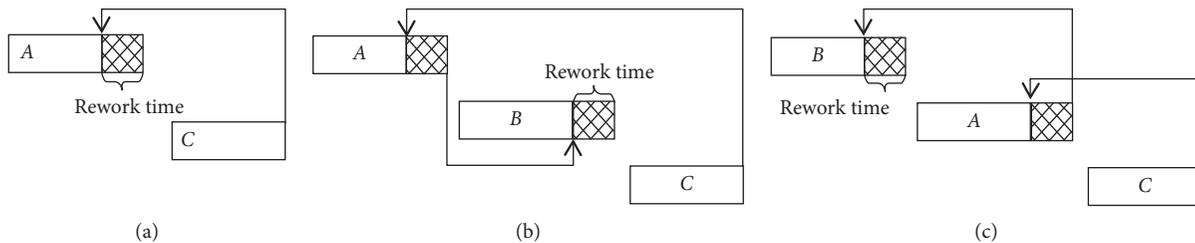


FIGURE 3: The first and second rework time. (a) The first rework, (b) the second rework caused by information transition from its precedent activity, and (c) the second rework caused by information feedback from its successive activity.

**3.2.1. Effects of Information Flow Interactions on Rework Risks.** As previously mentioned, information uncertainties are the main reasons for rework [7]. Such uncertain information may transmit or feedback from one activity to another, creating information flow. Information flow interactions may cause rework risks in projects and consequently cost more rework time. Although information transmission and feedback exist in both overlapping and nonoverlapping activities, information flow interactions in these two conditions are significantly different.

For nonoverlapping activities, rework risks mainly arise from information feedback from successive activities and information transmission from precedent activities after completion of their rework. As depicted in Figure 3, A is a precedent activity of B, and together, they perform sequentially according to relationships. After C has been completed, the performance information of C will be generated and transmitted from C to A, which may result in rework of A. Such a case is known as the first rework [45]. The first rework time is shown as the shadow area in Figure 3(a). After the completion of rework, A transmits or feeds back some revised information to B, which may cause rework of B. Such cases are called the second rework. The second rework times are shown as the shadow areas in Figures 3(b) and 3(c). Both

the first rework and the second rework should be taken into consideration in project scheduling to avoid that the planned project duration is much less than the actual duration.

In addition to the aforementioned two reasons, information flow interactions between overlapping activities are another major cause of rework. It is noteworthy that only overlapping-dependent activities can generate this type of rework risk for projects. Two activities overlapping without a dependent relationship will lead to no rework. As shown in Figure 4,  $t_A$  represents the completed time of activity A before A transmits preliminary information to its successive activity B.  $t_B$  represents the execution time of activity B when B receives information from A, while  $t_1$  represents the duration in which B continually receives information from A and transfers feedback information to A simultaneously until A is complete. Hence, the overlap between A and B is represented as the sum of  $t_B$  and  $t_1$ . The rework duration arising from information flow interactions between overlapping activities can be represented as  $RT_0$ , which is a function of overlapping duration—i.e.,  $t_A$  and  $t_1$ .

**3.2.2. Determining Rework Times Arising from Activity Overlaps.** In order to quantify the effects of rework risks on activity and project duration, four matrices are introduced to

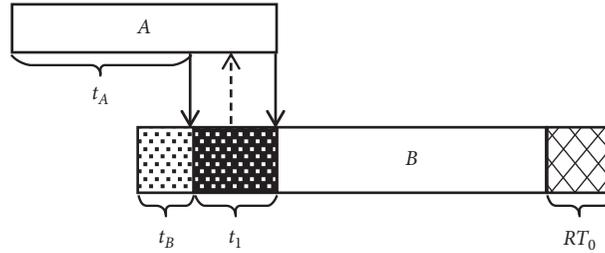


FIGURE 4: Information flow interaction between overlapping activities.

calculate rework times, including the rework probability matrix (RP), the rework intensity matrix (RI), the information output time-factor matrix ( $P$ ), and the information input time-factor matrix ( $S$ ). A detailed explanation is shown as follows: (1) matrix RP: the nondiagonal element  $[RP]_{ij}$  represents the rework probability of activity  $i$  caused by information from activity  $j$ . In particular, the elements in the upper triangle indicate the rework probabilities of upstream activities caused by information feedback from the downstream activities. The lower triangular elements indicate the rework probabilities of downstream activities caused by information transmission from the upstream activities. (2) Matrix RI: the nondiagonal element  $[RI]_{ij}$  represents the rework intensity of activity  $i$  caused by information from activity  $j$ , which is revealed as the percentage of duration of activity  $i$ . (3) Matrix  $P$ : the nondiagonal element  $[P]_{ij}$  means the ratio of the completed duration of activity  $j$  to its initial

activity duration when activity  $j$  begins to transfer information to activity  $i$ . (4) Matrix  $S$ : similar to matrix  $P$ , the nondiagonal element  $[S]_{ij}$  means the ratio of the completed duration of activity  $i$  to its initial activity duration when activity  $i$  begins to receive information from activity  $j$ .

To calculate rework times between overlapping activities, the overlap times ( $O_{ij}$ ) need to be determined first. The formula is shown as follows:

$$[F_{ss}]_{ij} = P_{ij} \cdot D_{jj} - S_{ij} \cdot D_{ii}. \quad (9)$$

Matrix  $F$  can then be achieved through equation (3), and the earliest start times of activities can be determined through equation (5). Based on the earliest start times of activities, the overlap duration ( $O_{ij}$ ) between activities  $i$  and  $j$  can be obtained by

$$O_{ij} = \begin{cases} \max(0, \min(D_{ii}, D_{jj} + [x_E^-]_j - [x_E^-]_i)): & \text{if } P_{ij} \text{ or } S_{ij} \neq 0, \\ 0: & \text{else.} \end{cases} \quad (10)$$

Therefore, the learning curve for each activity is modeled as a fixed value in this research, which means that a fixed proportion of original activity duration will be taken for rework execution whether it is the first or subsequent rework.

Based on the aforementioned definitions, the rework duration ( $RT_0$ ) in Figure 4 can be calculated by

$$[RT_0]_{ij} = [RP]_{ij} \cdot [RI]_{ij} \cdot LC_i \cdot [O]_{ij}, \quad (11)$$

where  $LC_i$  represents the learning curve of activity  $i$ .

Figure 5 shows a successor activity  $i$  with all of its entire predecessors ( $j = 1$  to  $n$ ), and each of the predecessors has an overlapping period with activity  $i$  ( $O_{ij}$ :  $j = 1$  to  $n$ ).  $[RT_0]_{ij}$  refers to the rework time of activity  $i$  caused by its predecessor  $j$ , and  $O_i$ , which satisfies  $O_i = \max(O_{i1}, O_{i2}, \dots, O_{in})$ , represents the maximum overlapping period of activity  $i$  with all its predecessors, while  $[RT_0]_i$  represents the rework time of activity  $i$  caused by its predecessors 1 to  $n$ .

According to probability theory, if two events ( $A$  and  $B$ ) are independent of each other and can occur simultaneously, the occurrence probability of the union of the two events, denoted as  $P(A \cup B)$ , can be calculated by

$$P(A \cup B) = P(A) + P(B) - P(A) \cdot P(B). \quad (12)$$

It should be noted that the probability principles are still valid when more than two events exist as well.

Since an activity can overlap with multiple predecessors simultaneously and the overlaps are not mutually exclusive, all of the reworks caused by individual overlaps between activity  $i$  and its different predecessors can be considered as nearly independent of each other. According to equation (11), the rework time resulting from individual overlaps can be calculated by  $[RT_0] = X_{ij} \cdot O_{ij}$  ( $j = 1, 2, \dots, n$ ), where  $[RP]_{ij} \cdot [RI]_{ij} \cdot LC_i$  is replaced with  $X_{ij}$  to represent the calculation process more concisely. For any overlapping time ( $O_{ij}$ ), at least  $X'_{ij}$  exists that satisfies  $[RT_0]_{ij} = X'_{ij} \cdot O_{ij} = X'_{ij} \cdot O_i$ . Therefore, the calculation process of  $X'_{ij}$  can be expressed with respect to individual rework duration ( $[RT_0]_{ij}$ ) and the maximum individual overlapping period ( $O_i$ ) as follows:

$$X'_{ij} = \frac{[RT_0]_{ij}}{O_i}. \quad (13)$$

The rework time can then be expressed as a function of the maximum individual overlapping period ( $O_i$ ) and the union of  $X'_{ij}$  ( $j = 1, 2, \dots, n$ ), that is,

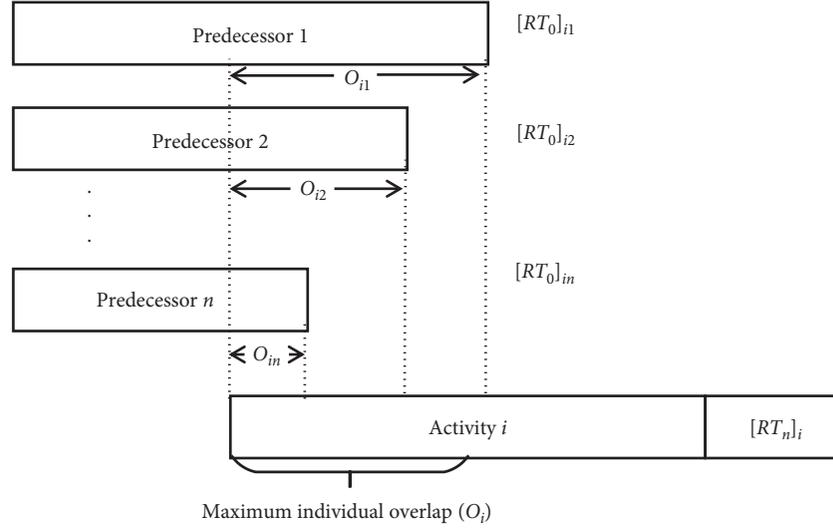


FIGURE 5: Effects of overlaps with multipredecessors on calculation of rework time.

$$[RT_0]_i = (X'_{i1} \cup X'_{i2} \cup \dots \cup X'_{in}) \cdot O_i \quad (14)$$

When  $j = 2$ , according to equations (12) and (13), the rework time is calculated by

$$[RT_0]_i = \frac{[RT_0]_{i1} + [RT_0]_{i2} + [RT_0]_{i3}}{O_i} - \frac{[RT_0]_{i1} \cdot [RT_0]_{i2} + [RT_0]_{i1} \cdot [RT_0]_{i3} + [RT_0]_{i2} \cdot [RT_0]_{i3}}{O_i^2} + \frac{[RT_0]_{i1} \cdot [RT_0]_{i2} \cdot [RT_0]_{i3}}{O_i^3} \quad (15)$$

Similar equations can be extended from equation (14) for the calculation of rework times in the case of  $j = 3, 4, \dots, n$ . When  $j = n$ , the calculation process is given as

$$[RT_0]_i = \left( \frac{\sum_{1 \leq j \leq n} [RT_0]_{ij}}{O_i} - \frac{\sum_{1 \leq j < k \leq n} [RT_0]_{ij} \cdot [RT_0]_{ik}}{O_i^2} + \dots + (-1)^{n-1} \cdot \frac{\prod_{1 \leq j \leq n} [RT_0]_{ij}}{O_i^n} \right) \cdot O_i \quad (16)$$

The rework time is more reasonable than that determined by traditional methods because of the following three reasons. First, the rework time does not exceed the longest individual overlapping period. Second, the result will never be less than the longest individual rework time, and third, the rework time will increase as the number of precedent activities increases.

**3.2.3. Determining the Times of the First and the Second Reworks.** Based on the previously defined parameters, including rework probability ( $[RP]_{ij}$ ), rework intensity ( $[RI]_{ij}$ ), learning curve ( $LC_i$ ), and activity duration ( $[D]_{ii}$ ), the first rework times of activity  $i$  ( $[RT_1]_{ij}$ ) caused by information feedback from activity  $j$ , shown as the shadow in Figure 3(a), can be obtained by [17]

$$[RT_1]_{ij} = [RP]_{ij} \cdot [RI]_{ij} \cdot LC_i \cdot [D]_{ii} \quad (17)$$

The total time of the first rework of activity  $i$  ( $[RT_1]_i$ ) caused by information feedback from all its successive activities can be obtained by

$$[RT_1]_i = \sum_{j=i+1}^n [RP]_{ij} \cdot [RI]_{ij} \cdot LC_i \cdot [D]_{ii} \quad (18)$$

Similarly, the total time of the second rework of activity  $i$  ( $[RT_2]_i$ ) caused by information feedback from all of its successive activities, shown as the shadow in Figures 3(b) and 3(c), can be obtained by

$$[RT_2]_i = \sum_{j=1}^n \sum_{k=\max(i,j)+1}^n [RP]_{jk} \cdot [RP]_{ij} \cdot [RI]_{ij} \cdot LC(i) \cdot [D]_{ii} \quad (19)$$

The total rework time of activity  $i$  ( $[RT]_i$ ), denoted as rework safety time ( $ST_{ri}$ ) for ease of understanding in

determining time buffers, can be obtained by summing up the rework time arising from activity overlaps ( $[RT_0]_i$ ), the time of the first rework ( $[RT_1]_i$ ), and that of the second rework ( $[RT_2]_i$ ), whose calculation process is shown as follows:

$$\begin{cases} [RT]_i = [RT_0]_i + [RT_1]_i + [RT_2]_i, \\ ST_{ri} = [RT]_i. \end{cases} \quad (20)$$

**3.2.4. Activity Sequence Optimization to Reduce Information Flow Interactions.** As previously mentioned, information flow interactions cause the codependency between activity durations and bring about rework risks. The rework times of activities are highly correlated to the intensity of information flow interactions. Noticeably, compared to information transmission, information feedback is more likely to lead to a higher intensity of information flow interactions. Therefore, we can reduce the intensity of information flow interactions and activity rework times by shortening the distance of the information feedback path through the optimization of the activity sequence [46]. It is introduced as the objective function to optimize the activity sequence:

$$f = \sum_{i=1}^n \sum_{j=i+1}^n [RP]_{i,j} \cdot (j - i), \quad (21)$$

where  $[RP]_{i,j}$  represents the  $(i, j)$ th value extracted from the rework probability matrix.

**3.3. Determining Buffer Sizes with the Refined RSEM in a Max-Plus Representation.** In this section, we improve the traditional RSEM by adopting more accurately calculated rework safety times and express them in max-plus algebra to make the calculation process more effective.

**3.3.1. Determining Project Buffers.** According to the central limit theorem, the activity safety times and rework safety times are first allocated at the end of the project, and the refined RSEM is then used to determine the project buffer. The principle of the refined RSEM is shown in Figure 6, in which  $d_i$  represents the estimated duration of activity  $i$  with a 50% completion rate, and  $ST_i$  denotes the total safety time of activity  $i$ .  $ST_i$  is composed of the rework safety time ( $ST_{ri}$ ) and the activity safety time ( $ST_{0i}$ ), which has a 90% completion rate. The project buffer can then be calculated with refined RSEM by

$$PB = \sqrt{\sum_{i \in CP} (ST_{0i} + ST_{ri})^2}. \quad (22)$$

To calculate PB effectively, a virtual activity, two vectors, and a matrix are defined, denoted as  $V_0$ ,  $ST^2$ ,  $ST_p^2$ , and  $T_p$ , respectively. Virtual activity  $V_0$  is added at the end of schedules and consumes no time. An assumption is made that a relationship exists that satisfies  $FTF = 0$  between  $V_0$

and each other activity. The  $i$ th element of  $ST^2$ , denoted as  $[ST^2]_i$ , means the square of total safety time of activity  $i$ , and the corresponding element of activity  $V_0$  is  $e$ . Vector  $ST_p^2$  is transformed from  $ST^2$  by

$$[ST_p^2]_i = \begin{cases} [ST^2]_i: & \text{if } [TF]_i = 0, \\ e: & \text{else.} \end{cases} \quad (23)$$

Matrix  $T_p$  is transformed from  $ST_p^2$  by

$$[T_p]_{ij} = \begin{cases} [ST_p^2]_i: & \text{if } j = i - 1, j = 1, 2, \dots, n - 1, \\ \varepsilon: & \text{else.} \end{cases} \quad (24)$$

To better interpret matrix  $T_p$ , a simple case example is shown in Figure 7. It is assumed that the square of total safety times of activities is  $[a, b, c]$ , and activity  $B$  is not on the critical chain. We then achieve the vector  $ST_p^2$  as  $[a, e, c, e]$  and matrix  $T_p$ .

The PB can then be represented in max-plus algebra as

$$PB = \sqrt{C_0^T \otimes T_p^* \otimes C_0}, \quad (25)$$

where  $C_0^T = (\underbrace{e, e, \dots, e}_{n+1})$ , which is a  $1 \times (n + 1)$  matrix with  $e$  representing zero.

**3.3.2. Determining Feeding Buffers.** In order to insert feeding buffers in place to protect the critical path from delays in noncritical paths, two operators, denoted as  $\text{diag}()$  and  $\#$ , are first introduced. For a vector  $x$ , a new matrix can be obtained by the operator  $\text{diag}()$  as  $[\text{diag}(x)]_{i,j} = \{x_i: \text{if } i = j, \varepsilon: \text{else}\}$ , and a new vector can be obtained by the operator  $\#$  as  $\{[x]_i\}^\# = \{e: \text{if } [x]_i \neq \varepsilon, \varepsilon: \text{if } [x]_i = \varepsilon\}$ .

In preparation of the location and size of the feeding buffer, another two vectors are introduced,  $[w]_i = \{e: i \in \alpha, \varepsilon: i \notin \alpha\}$  and  $[v]_i = \{e: i \in \beta, \varepsilon: i \notin \beta\}$ , where  $\alpha$  and  $\beta$  are the set of activities on the critical chain and those on the noncritical chain, respectively. Moreover, an adjacency matrix  $R$  is transformed from matrix  $F$  by  $[R]_{ij} = \{e: \text{if } [F]_{ij} \neq \varepsilon, \varepsilon: \text{else}\}$ . Vector  $v'$  is then introduced to locate the feeding buffers as follows:

$$[v']_y = [R_{\beta\alpha}^T \otimes w]_y, \quad (26)$$

where  $R_{\beta\alpha}$  is the adjacency matrix, which represents transitions from noncritical activities to critical ones, and can be obtained by  $R_{\beta\alpha} = \text{diag}(w) \otimes R \otimes \text{diag}(v)$ .  $\lambda$  represents the noncritical activity, one of whose successors is critical activity. Therefore, a feeding buffer should be inserted behind the activity  $\gamma$ .

The feeding buffer can then be determined by square rooting of the sum of  $[ST^2]_\eta$ , where  $\eta$  is the set of activities on a certain noncritical chain. The formula is expressed in max-plus algebra as

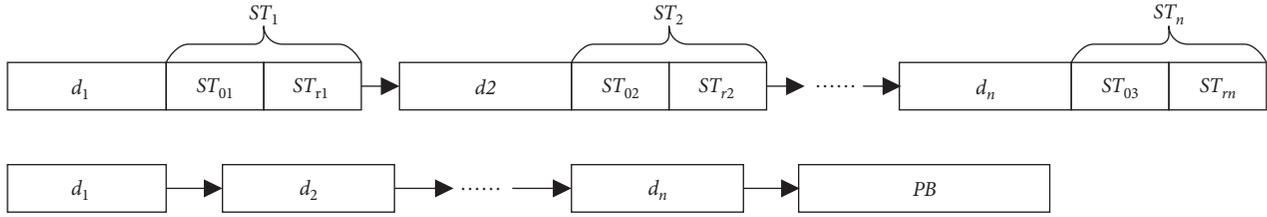


FIGURE 6: Buffer determination principle of the refined RSEM.

	A	B	C	$V_0$
A	$\epsilon$			
B	a	$\epsilon$		
C		e	$\epsilon$	
$V_0$			c	$\epsilon$

FIGURE 7: Matrix  $T_p$  (other elements  $\epsilon$  are omitted).

$$FB_0 = \sqrt{\text{diag}\left\{\left[R_{\beta\beta} \otimes \text{diag}(v'^{\#}) \otimes \text{diag}(ST^2)\right]^* \otimes C_1\right\} \otimes \text{diag}\left[\text{diag}(ST^2) \otimes \text{diag}(v') \otimes C_1\right] \otimes C_1}. \quad (27)$$

Furthermore, due to the constraint that the feeding buffer should not exceed its total float, the sizes of the feeding buffers should be adjusted by

$$[FB]_y = \min([FB]_y, [TF]_y). \quad (28)$$

#### 4. Case Study

This section presents an empirical analysis to validate the feasibility and effectiveness of the proposed method on addressing rework risks in project schedule management. Section 4.1 presents the project data used in this study. Section 4.2 describes the implementation process of the proposed method in detail, and Section 4.3 compares the schedule generated by the proposed method with that generated by traditional CCPM methods.

**4.1. Project Dataset.** The case used for the empirical analysis was derived from a feasibility study of a modular real estate construction first introduced by Sullivan [47] and further supplemented by Eppinger et al. [41]. It was a typical project with large-scale rework relationships and overlaps. The data were collected through interviews of individual team members and the group as a whole, and interviews to identify the information exchanges required to execute each task and to better understand how the tasks were completed.

The project data are shown in Figures 8(a)–8(d) and Table 3. The figure represents the initial predecessor time-factor matrix  $P_0$ , the initial successor time-factor matrix  $S_0$ , the initial rework probability matrix  $RP_0$ , and the initial rework impact matrix  $RI_0$ , respectively. Table 3 shows activities in the case and corresponding activity parameters including an optimistic duration completion rate of 90%, a pessimistic duration completion rate of 50%, and the most likely duration and learning curves for each activity.

**4.2. Implementation of the Proposed Method.** The case was implemented with the proposed method, and the steps are shown in Figure 9 and described in detail as follows:

**Step 1: optimize the activity sequence.** A genetic algorithm (GA), which has proven to be a useful tool in addressing complex project scheduling problems by a number of prior studies [48], was developed in MATLAB to address the activity-sequencing problem. Specially, according to previous research [41], the parameters of the GA including population size, number of generations, crossover probability, and the mutation probability were set as 50, 150, 0.95, and 0.08, respectively. Equation (21) was used as an objective function to generate a near-optimal activity sequence. Figure 10 shows the GA convergence process and

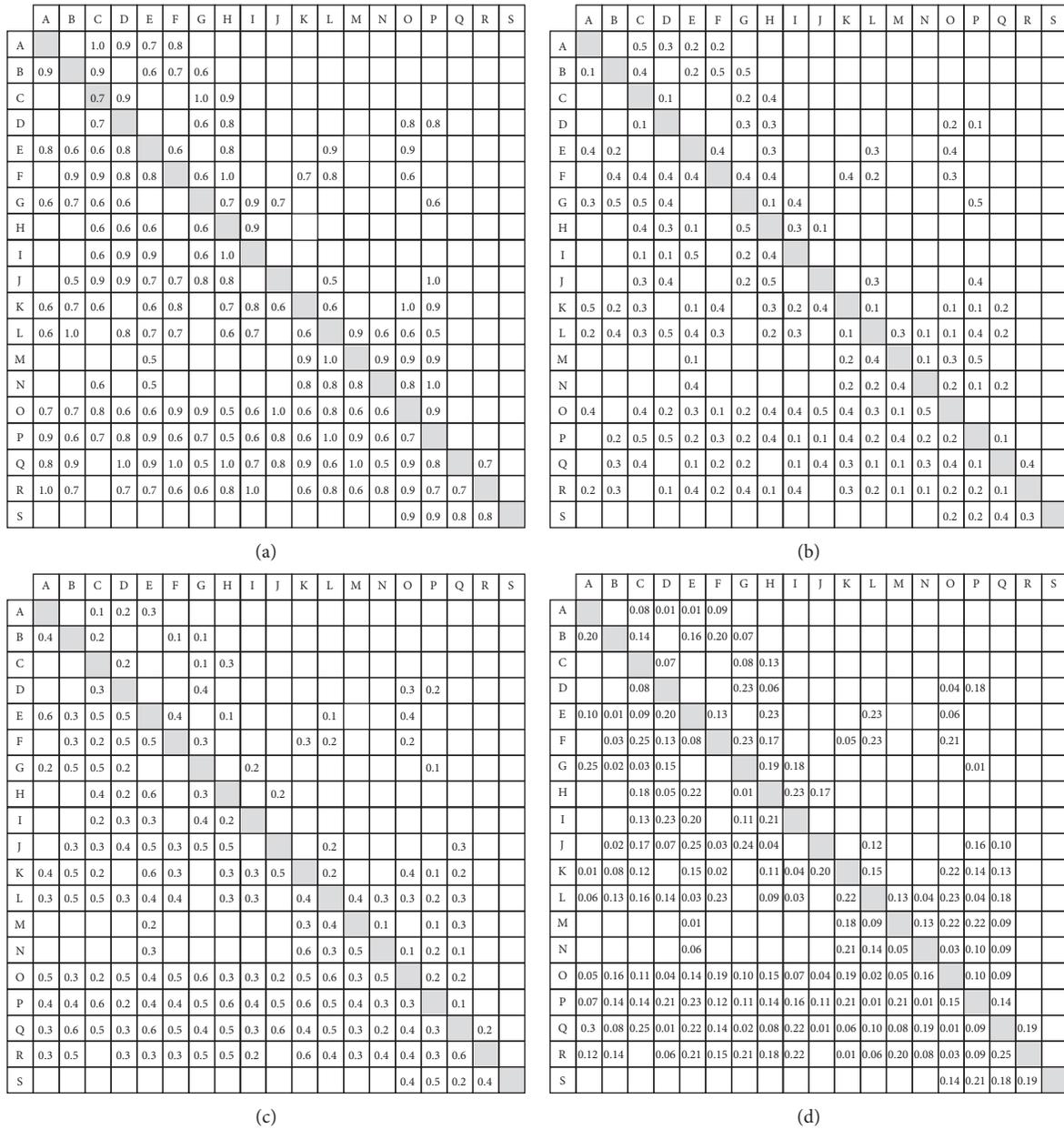


FIGURE 8: Project information initial matrices in the case project. (a) Matrix  $P_0$ . (b) Matrix  $S_0$ . (c) Matrix  $RP_0$ . (d) Matrix  $RI_0$ .

reported a near-optimal activity sequence as  $[F, C, A, J, R, K, P, D, M, O, B, H, E, N, L, G, Q, S]$ .

Step 2: update matrices  $P_0$ ,  $S_0$ ,  $RP_0$ , and  $RI_0$  into matrices  $P$ ,  $S$ ,  $RP$ , and  $RI$  according to the new activity sequence. The initial elements in the matrices were kept unchanged and transferred to the new places in the matrices in which the row and column represent corresponding original activity, respectively.

Step 3: generate matrices  $D$  and  $R$ . Matrix  $D$  was generated by filling the diagonal elements with a corresponding optimistic activity duration completion rate of 50%, as shown in Figure 11(a). Matrix  $R$  was derived from matrix  $P$  or  $S$  by replacing all of the nonnull elements with  $e$  and all of the null elements and

upper-triangle elements with  $\varepsilon$ , as shown in Figure 11(b).

Step 4: generate matrices  $F_{ss}$  and  $F$ . Matrix  $F_{ss}$  was derived from matrices  $P$ ,  $S$ , and  $D$  according to equation (9). Matrix  $F$  was derived from matrices  $F_{ss}$  and  $D$  according to equation (3).

Step 5: determine the time parameters of activities including  $x_E^+$ ,  $x_E^-$ , and  $x_L^-$ , as well as the critical chain and its length.  $x_E^+$ ,  $x_E^-$ , and  $x_L^-$  were calculated according to equations (4), (5), and (7) based on matrices  $F$  and  $D$ . The total float vector TF was generated by equation (8) as  $[0, 0, 4.90, 0, 2.52, 49.71, 0, 0, 0, 5.14, 1.30, 13.67, 1.94, 0, 2.71, 0, 0, 0, 0]$  and reported the critical chain (CC) as  $[F, C, I, K, P, D, E, L, G, Q, S]$ . The length of the

TABLE 3: Case activity parameters.

ID	Activity	Duration (day)			Learning curve
		Optimistic	Most likely	Pessimistic	
A	Perform preliminary market analysis	5	8	15	0.5
B	Evaluate marketability options	5	8	15	0.6
C	Engage feasibility consultants	3	4	7	0.3
D	Evaluate planning and zoning process	10	13	20	0.5
E	Perform massing study	20	24	35	0.6
F	Develop conceptual design	30	35	50	0.8
G	Identify external stakeholders	5	8	15	0.4
H	Identify permits and approvals	10	13	20	0.7
I	Complete phase 1 ESA	10	13	20	0.3
J	Evaluate consultants and contractors	10	13	20	0.6
K	Obtain rough construction costs	5	8	15	0.5
L	Determine highest and best use	3	5	9	0.5
M	Identify debt options	2	4	8	0.4
N	Identify equity options	3	4	5	0.5
O	Update financial underwriting	1	2	3	0.5
P	Reevaluate the organization strategy	2	3	5	0.6
Q	Estimate the schedule	3	4	7	0.6
R	Gain control of the site and/or client	3	4	7	0.6
S	Review and approve	3	4	7	0.7

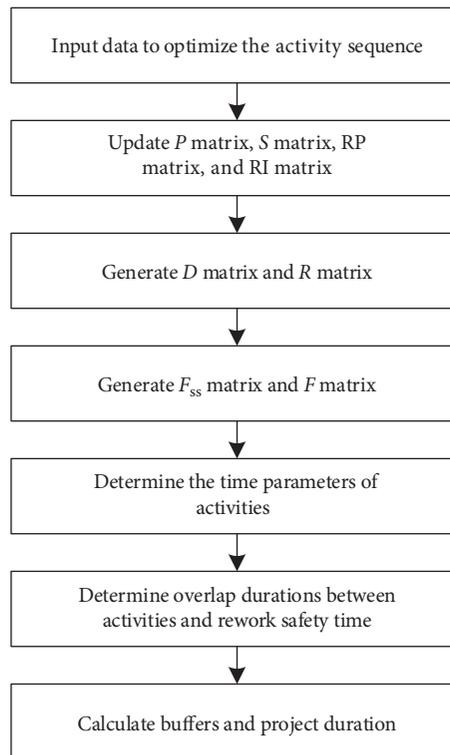


FIGURE 9: Flowchart of the simulation steps.

critical chain was calculated by equation (6), with a result of 68.67 days.

Step 6: determine overlap durations between activities and rework safety times. The overlap durations between activities were obtained by equation (10) based on the vector  $x_E^-$  and matrix  $D$ . Rework safety times including rework times arising from the activity overlap, the first rework times, and the second rework times were

determined by equations (11) through (20) with the result of [0.93, 0.13, 0.20, 1.36, 4.50, 3.19, 3.91, 2.90, 1.93, 0.35, 0.61, 0.26, 2.72, 3.81, 0.54, 0.80, 3.05, 0.24, 0.05].

Step 7: calculate buffers and project duration. Equations (23) through (28) calculated the buffers. The resulting project buffer was 40.03 days, and the feeding buffers were [0, 0, 4.90, 0, 2.52, 7.19, 0, 0, 0, 5.14, 1.3,



TABLE 4: Case project results.

	Project buffer (day)	Project duration (day)	Buffer consuming rate on average (%)	On-time completion probability (%)
The proposed method	40.03	108.70	89.39	85.40
Zhang's method	35.58	104.25	100.57	50.40
Original RSEM	33.43	102.11	107.04	34.60

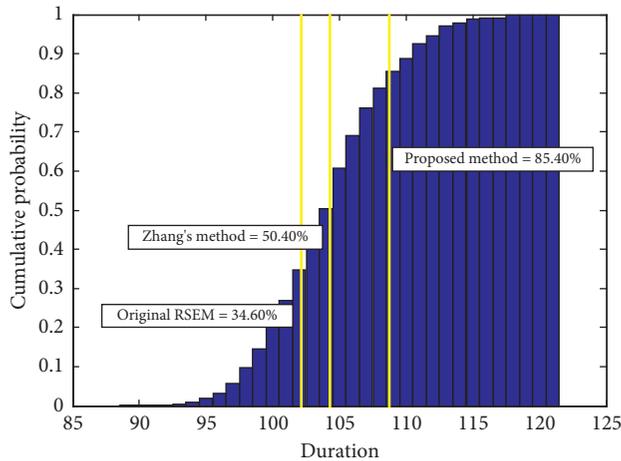


FIGURE 13: Comparison of on-time project completion probabilities between the proposed method, the original RSEM, and Zhang's method-based schedules.

10.26, 1.94, 0, 2.54, 0, 0, 0, 0]. The expected project duration was 108.70 days.

**4.3. The Proposed Method Compared with Traditional CCPM Methods.** To further validate our model, we compared the performance of the proposed method with traditional CCPM methods that used the original RSEM and the buffer determination method in Zhang et al.'s paper to determine the project buffer [8]. It is noted that the comparison is based on the optimized sequence of the activities. Previously, to access the effectiveness of the proposed methods in project scheduling under the rework scenarios, the Monte Carlo simulation approach was executed 500 times in the case project in which the duration of each project activity adhered to a beta distribution. Figure 12 shows the simulated actual project durations.

While all three methods employed the CCPM-MPL representation method proposed in the paper to generate project schedules, they differed in terms of determination of project buffer sizes. The original RSEM, as previously mentioned, disregarded rework safety times. Zhang's method overcame this problem, but overlooked the effects of the learning curve, overlapping with multiple predecessors, and the second rework on calculation results of rework safety times when determining the project buffer. Our model further factored in these considerations to generate a more accurate project buffer and reliable project schedule.

The project buffers and schedules obtained by these three methods and comparison results are presented in Table 4

and Figure 13. The results showed that the estimated project schedule durations generated with Zhang's method and the original RSEM were 104.25 days, including the project buffer of 35.58 days, and 102.11 days including the project buffer of 33.43 days, respectively. When the estimated project durations were superimposed in the simulated project durations using the Monte Carlo approach as depicted in Figure 13, it can be seen that the on-time completion probabilities were 50.40% and 34.60%, respectively. The results show that the schedule generated with the proposed method, which had an on-time completion probability of 85.40%, significantly outperformed schedules generated with the other two methods in ensuring on-time completion probability of the case project.

Furthermore, the proposed method outperformed Zhang's method and the original RSEM by providing an appropriately sized project buffer with a proper buffer consumption rate on average. As illustrated in Table 4, the buffer consumption rate of schedules generated with the proposed method was 89.39%, while that of the other two methods was 100.57% and 107.04%, respectively. The oversized buffer consumption rate means that, in many cases, the calculated project buffer is undersized and unable to protect the project from schedule delay.

## 5. Conclusion

In this paper, through an integration of the DSM and CCPM methods, as well as a max-plus method, a new project scheduling method was developed to generate schedules of projects under rework scenarios. Accurate rework safety times of activities were calculated by factoring in rework time arising from activity overlaps, the first rework time, and the second rework time. Time buffers were then determined with the RSEM considering the rework risk. The refined max-plus method was utilized to transform the logic ties, rework relationships, and other parameters of the activity into simple matrix operations, and such parameters are uniformly calculated by the max-plus method to realize the effective scheduling of the project. Formulas in the max-plus algebra were created to calculate buffers with the refined RSEM and generate project schedules with the STS relationship between activities. The empirical results showed that project schedules generated with the proposed method have appropriately sized project buffers and can ensure a higher on-time completion probability compared to those generated with traditional CCPM methods.

The paper contributes to the project schedule management research and practice in the following ways.

We have greatly extended the research framework of the CCPM-MPL representation method. The original CCPM-MPL framework was based on the logical tie in that FTS is equal to 0, and we developed the formula applicability to the situation that STS is not equal to 0. The original CCPM-MPL framework employed the C&PM method to calculate times buffers, and we designed the formula expressed in max-plus algebra for calculation of time buffers with the RSEM and adjusted the max-plus method to generate project schedules with the STS relationship existing between activities. Our model considers the learning curve effects and the overall overlapping effect to calculate rework safety times accurately. DSMs have been integrated into the framework, enabling the generated project schedules to deal with rework risks effectively. We also developed an accurate calculation model of rework safety times by factoring various rework factors and improved the calculation accuracy of rework time.

The proposed method is a promising and powerful tool for project practitioners. Project schedules generated with this proposed method can absorb rework risks and various other uncertainties, enabling practitioners to predict project duration accurately and have sufficient time to deal with rework risks during the project process.

There also exist several limitations of this study. Considering little attention to the STS relationship in the literature as well as the potential to incorporating overlaps of the STS relationship, this study makes an attempt to focus on the STS relationship, yet we also ignore other types of relationship, such as the STF relationship. There are many directions for future research efforts regarding the problem investigated in the current paper. Resource conflict continues to remain a critical challenge in project scheduling and deserves future research by extending the current CCPM-MPL framework. Additionally, the effect of activity overlap on the calculation of activity safety times was not considered when determining the project buffer and feeding buffers. This topic also merits future research. Besides, the current study validates the proposed method with a case study, and further application of this method should be investigated in different types of project and scenario.

## Data Availability

The data generated or analyzed during the study are available from the corresponding author upon request. Information about the journal's data-sharing policy can be found at [http://ascelibrary.org/doi/10.1061/\(ASCE\)CO.1943-7862.0001263](http://ascelibrary.org/doi/10.1061/(ASCE)CO.1943-7862.0001263).

## Disclosure

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSFC.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This material was based on the work supported by the National Natural Science Foundation of China (NSFC) under Grant no. 71671128. The authors thank the NSFC for its support.

## References

- [1] J. L. Ashford, *The Management of Quality in Construction*, Routledge, Abingdon, UK, 2002.
- [2] Sommerville and J. J. S. Survey, "Defects and rework in new build: an analysis of the phenomenon and drivers," vol. 25, no. 5, pp. 391–407.
- [3] P. E. D. Love, "Influence of project type and procurement method on rework costs in building construction projects," *Journal of Construction Engineering and Management*, vol. 128, no. 1, pp. 18–29, 2002.
- [4] P. Barber, A. Graves, M. Hall, D. Sheath, and C. Tomkins, "Quality failure costs in civil engineering projects," *International Journal of Quality & Reliability Management*, vol. 17, pp. 479–492, 2000.
- [5] B.-G. Hwang, X. Zhao, and K. J. Goh, "Investigating the client-related rework in building projects: the case of Singapore," *International Journal of Project Management*, vol. 32, no. 4, pp. 698–708, 2014.
- [6] G. Ye, Z. Jin, B. Xia, and M. Skitmore, "Analyzing causes for reworks in construction projects in China," *Journal of Management in Engineering*, vol. 31, no. 6, 2015.
- [7] N. Forcada, M. Gangoles, M. Casals, and M. Macarulla, "Factors affecting rework costs in construction," *Journal of Construction Engineering and Management*, vol. 143, 2017.
- [8] J. Zhang, X. Song, H. Chen, and R. Shi, "Determination of critical chain project buffer based on information flow interactions," *Journal of the Operational Research Society*, vol. 67, no. 9, pp. 1146–1157, 2016.
- [9] F. Ackermann, "Problem structuring methods "in the dock": arguing the case for soft OR," *European Journal of Operational Research*, vol. 219, no. 3, pp. 652–658, 2012.
- [10] M. Jingmond and R. Ågren, "Unravelling causes of defects in construction," *Construction Innovation*, vol. 15, no. 2, pp. 198–218, 2015.
- [11] S. Han and F. Peña-Mora, "A system dynamics model for assessing the impacts of design errors in construction projects," *Mathematical and Computer Modelling*, vol. 57, pp. 2044–2053, 2013.
- [12] Y. Li and T. Taylor, "Modeling the impact of design rework on transportation infrastructure construction project performance," *Journal of Construction Engineering and Management*, vol. 140, 2014.
- [13] M. Alam and U. Gühl, *Project-Management in Practice*, Wiley, Hoboken, NY, USA, 2016.
- [14] D. Dalcher, *The Oxford handbook of project management*. *Project Management Journal*, vol. 42, p. 93, 2011.
- [15] Y. Su, G. Lucko, and R. C. Thompson, "Application of voting theory to the float ownership problem," *Journal of Construction Engineering and Management*, vol. 144, no. 1, 2018.
- [16] D. Steward, "The design structure system: a method for managing the design of complex systems," *IEEE Transactions on Engineering Management*, 1981.
- [17] T. R. Browning and S. D. Eppinger, "Modeling impacts of process architecture on cost and schedule risk in product development," *IEEE Transactions on Engineering Management*, vol. 49, no. 4, pp. 428–442, 2002.

- [18] S.-H. Cho and S. D. Eppinger, "A simulation-based process model for managing complex design projects," *IEEE Transactions on Engineering Management*, vol. 52, no. 3, pp. 316–328, 2005.
- [19] H. M. E. Abdelsalam and H. P. Bao, "Re-sequencing of design processes with activity stochastic time and cost: an optimization-simulation approach," *Journal of Mechanical Design*, vol. 129, no. 2, pp. 150–157, 2007.
- [20] A. A. Yassine and L. A. Wissmann, "The implications of product architecture on the firm," *Systems Engineering*, vol. 10, no. 2, pp. 118–137, 2007.
- [21] G. K. Rand and E. M. Goldratt, "Critical chain," *The Journal of the Operational Research Society*, vol. 49, no. 2, p. 181, 1998.
- [22] G. Ma, K. Hao, Y. Xiao, and T. Zhu, "Critical chain design structure matrix method for construction project scheduling under rework scenarios," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1595628, 14 pages, 2019.
- [23] F. Ahlemann, F. El Arbi, M. G. Kaiser, and A. Heck, "A process framework for theoretically grounded prescriptive research in the project management field," *International Journal of Project Management*, vol. 31, no. 1, pp. 43–56, 2013.
- [24] G. K. Rand, "Critical chain: the theory of constraints applied to project management," *International Journal of Project Management*, vol. 18, no. 3, pp. 173–177, 2000.
- [25] M. Bevilacqua, F. E. Ciarapica, and G. Giacchetta, "Critical chain and risk analysis applied to high-risk industry maintenance: a case study," *International Journal of Project Management*, vol. 27, no. 4, pp. 419–432, 2009.
- [26] G. Ma, A. Wang, N. Li, L. Gu, and Q. Ai, "Improved critical chain project management framework for scheduling construction projects," *Journal of Construction Engineering and Management*, vol. 140, 2014.
- [27] E. Roghanian, M. Alipour, and M. Rezaei, "An improved fuzzy critical chain approach in order to face uncertainty in project scheduling," *International Journal of Construction Management*, vol. 18, no. 1, pp. 1–13, 2017.
- [28] B. Ashtiani, G.-R. Jalali, M.-B. Aryanezhad, and A. Makui, "New approach for buffer sizing in critical chain scheduling," in *Proceedings of the 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 1–4, pp. 1037–1041, Singapore, December 2007.
- [29] Q. Shi and T. Gong, "An improved project buffer sizing approach to critical chain management under resources constraints and fuzzy uncertainty," in *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1, pp. 486–490, Shanghai, China, November 2009.
- [30] O. I. Tukul, W. O. Rom, and S. D. Eksioğlu, "An investigation of buffer sizing techniques in critical chain scheduling," *European Journal of Operational Research*, vol. 172, no. 2, pp. 401–416, 2006.
- [31] Q. Shi, Y. T. Wang, and T. Gong, "An improved approach for project buffer sizing and evaluation," *System Engineering Theory and Practice*, vol. 32, pp. 1739–1746, 2012.
- [32] J. Zhang, R. Shi, and E. Diaz, "Dynamic monitoring and control of software project effort based on an effort buffer," *Journal of the Operational Research Society*, vol. 66, 2015.
- [33] L. Bi, B. Ren, D. Zhong, and L. Hu, "Real-time construction schedule analysis of long-distance diversion tunnels based on lithological predictions using a markov process," *Journal of Construction Engineering and Management*, vol. 141, 2015.
- [34] Y. Su, G. Lucko, and R. C. Thompson, "Apportioning contract float with voting methods to correlated activities in network schedules to protect construction projects from delays," *Automation in Construction*, vol. 118, 2020.
- [35] E. M. Goldratt, *Critical Chain: A Business Novel*, p. 1709, Routledge, Abingdon, UK, 2017.
- [36] R. C. Newbold, *Project Management in the Fast Lane: Applying the Theory of Constraints*, p. 55, CRC Press, Boca Raton, FL, USA, 1998.
- [37] J. Zhang, X. Song, and E. Diaz, "Project buffer sizing of a critical chain based on comprehensive resource tightness," *European Journal of Operational Research*, vol. 248, no. 1, pp. 174–182, 2016.
- [38] T. R. Browning, "Design structure matrix extensions and innovations: a survey and new opportunities," *IEEE Transactions on Engineering Management*, vol. 63, no. 1, pp. 27–52, 2016.
- [39] A. Yassine, O. Mostafa, and T. Browning, "Scheduling multiple, resource-constrained, iterative, product development projects with genetic algorithms," *Computers & Industrial Engineering*, vol. 107, 2017.
- [40] T.-L. Yu, A. A. Yassine, and D. E. Goldberg, "An information theoretic method for developing modular architectures using genetic algorithms," *Research in Engineering Design*, vol. 18, no. 2, pp. 91–109, 2007.
- [41] S. Eppinger, S. Bonelli, and A. Gonzalez, "Managing iterations in the modular real estate development process," in *Proceedings of the 15th International Dependency and Structure Modelling Conference, DSM*, pp. 37–44, Melbourne, Australia, August 2013.
- [42] R. Dehghan and J. Ruwanpura, "Model of trade-off between overlapping and rework of design activities," *Journal of Construction Engineering and Management*, vol. 140, 2014.
- [43] M. Y. Jaber and A. L. Guiffrida, "Learning curves for imperfect production processes with reworks and process restoration interruptions," *European Journal of Operational Research*, vol. 189, no. 1, pp. 93–104, 2008.
- [44] A. Chassiakos and S. Sakellariopoulos, "Time-cost optimization of construction projects with generalized activity constraints," *Journal of Construction Engineering and Management*, vol. 131, 2005.
- [45] Q. Yang, T. Yao, T. Lu, and B. Zhang, "An overlapping-based design structure matrix for measuring interaction strength and clustering analysis in product development project," *IEEE Transactions on Engineering Management*, vol. 61, 2013.
- [46] J. Zhang, S. Xiwei, H. Chen, and R. Shi, "Optimisation of critical chain sequencing based on activities' information flow interactions," *International Journal of Production Research*, vol. 53, pp. 1–11, 2015.
- [47] G. F. Ma and T. C. Zhu, "Application of DSM in process management of a PE commercial real estate development project," *Property Management*, vol. 36, no. 1, pp. 20–36, 2018.
- [48] J. Kim, "Genetic algorithm stopping criteria for optimization of construction resource scheduling problems," *Construction Management and Economics*, vol. 31, pp. 1–17, 2012.