

Research Article

Critical Chain Design Structure Matrix Method for Construction Project Scheduling under Rework Scenarios

Guofeng Ma, Keke Hao , Yu Xiao, and Tiancheng Zhu

Department of Construction Management and Real Estate, Tongji University, Shanghai 200092, China

Correspondence should be addressed to Keke Hao; 1631029@tongji.edu.cn

Received 5 October 2018; Accepted 1 April 2019; Published 21 April 2019

Academic Editor: Emilio Jiménez Macías

Copyright © 2019 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 have been a major challenge in the construction industry that constantly affects project schedules and threatens on-time project completion. Traditional project scheduling methods are not capable of modeling rework relationships between activities and mitigating the impact of resulting uncertainties during the development of project schedules. To address this challenge, a critical chain design structure matrix (CCDSM) method is proposed in this paper. The CCDSM method aims to develop construction project schedules that are adaptive to rework scenarios and robust against rework risks. The CCDSM method models and displays large-scale rework relationships among activities and introduces a new rework buffer to quantitatively represent the impact of rework instances in project schedules. A max-plus algorithm is adopted in CCDSM to transform complex logic relationships into simple matrix operations, reducing computational load of schedule generation. A case study was conducted to demonstrate the implementation of the CCDSM method and assess its effectiveness in addressing rework risks. The results showed that the CCDSM is a promising tool to generate schedules, which could improve on-time project completion rate and reduce impacts of varying rework scenarios on project execution.

1. Introduction

Rework has been regarded as one of the major challenges that can adversely affect project performance in the construction industry [1]. Hwang et al. [2] surveyed about Client-Related Rework (CRR) in 381 projects conducted by 51 construction companies and concluded that more than 80% companies and 59% projects experienced CRR, which increased project costs by 7.1% and caused 3.3 weeks' delay on average. Simpeh et al. [3] investigated 78 construction professionals and found that if the mean rework cost of a project reached 5.12%, the likelihood of the project exceeding its budget would be as high as 76%.

Extensive efforts have been made to manage rework-caused project delay and minimize the rework impact in construction projects. When used to predict the project duration as well as its on-time completion probability, traditional scheduling techniques, such as CPM, PERT, and Gantt Charts, cannot quantitatively measure activity rework time and its impact on project completion time [4] and hence are not able to directly take into account the rework risks

during project planning. Although buffers in the CCPM can aggregate uncertainties in project execution, which are reflected as blocks of resource or time redundancy in project schedules, it is still difficult to model large-scale rework relationships between activities using the CCPM [5]. To further address this challenge, a design structure matrix (DSM) method has been introduced, which is designed to represent the dependency or information flow between activities, providing an effective representation of rework instances [5]. However, one limitation of the DSM method exists that the matrix format is too concise to fully display certain attributes of activities, which leads the DSM method usually to be adapted as a project duration calculator rather than a process control tool [6]. Overall, most of the current tools for managing rework are reactive, can barely compute and diagram rework instances with satisfactory accuracy, and focus on measuring the rework risk of activity duration rather than that of overall project completion time. Due to these limitations, although in many cases project managers expect rework to occur, they lack reliable tools to fully assess and address these risks at the planning stage of the projects

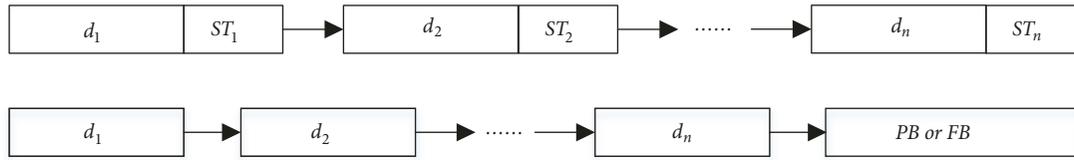


FIGURE 1: Concentration principle of buffer determination.

and to mitigate potential impacts of rework with proactive measures.

Accordingly, a new method, termed critical chain design structure matrix (CCDSM), is proposed in this paper. Built on the strength of the CCPM and DSM in project scheduling, the CCDSM aims to integrate these two methods to overcome their respective limitations in developing construction project schedules that are adaptive to rework instances. The CCDSM is designed in such a way that it not only models large-scale rework relationships among activities, improving the accuracy and effectiveness of project management, but also displays various attributes of activities, enabling analysis and visualization of the project schedule under rework scenarios. Moreover, the CCDSM adopts max-plus algorithm, a straightforward linear algebra algorithm, to transform complex logic relationships into simple matrix operations, which effectively reduces the computational load of generating schedules. The detailed design of the CCDSM and a case study that demonstrates the effectiveness of the CCDSM in addressing rework risks in construction projects are presented in the remainder of this paper.

2. Review of Relevant Works

2.1. Buffer Management and Critical Chain Scheduling in the CCPM. The CCPM, which has proved to be a powerful tool for project scheduling in many fields [7, 8], has been increasingly used in project scheduling and subject to a number of refinements and extensions [9]. There are two main components in the implementation of the CCPM, including buffer management and critical chain scheduling.

There are three major types of buffers, including resource buffer (RB), feeding buffer (FB), and project buffer (PB) in CCPM, which are distinguished by their positions and functions in the schedules [10]. RBs are set to protect the critical chain from the tightness of critical resources. They are used as warnings, consuming no time. FBs are set to protect the critical chain from the variation of tasks not on the critical chain and are placed where noncritical and critical activities converge. PBs are placed at the end of the critical chain to protect against exceeding project delivery dates. The concentration principle of buffer determination in CCPM is shown as Figure 1, where d_i represents the estimated duration of activity i and ST_i denotes the safety time of i . In fact, another two major time buffering methods existed for developing reliable schedules and protect them from effects of uncertainties. One is optimization-based time buffering method such as starting time criticality and Tabu search, which insert buffers in front of activities dispersedly

to ensure that each activity can be conducted on time in accordance with the project schedules. However, it is unknown whether such scattered buffers can ensure on-time completion of the entire project [11]. The other is simulation-based time buffering method. However, a major drawback of simulation-based methods exists that simulation is highly computationally demanding, which could become a significant problem in practice when large projects are involved or when project managers want to perform what-if analysis for a wide variety of scenarios [12]. The CCPM-based time buffering used in this study has several advantages. It calculates time buffers efficiently, provides explicit protection against stochastic variation and presents a set of reasonable control guidelines to ensure on-time completion of the entire project [13].

Determining the size of buffers depends on various factors, such as managerial experience and preferences of the project team, project circumstances, personnel and equipment capabilities, and so on [14]. A number of buffer sizing approaches have been developed in CCPM, among which the cut and paste method (C&PM) and the root square error method (RSEM) are widely used. The C&PM reduces the safety time of each activity by 50% and estimates half of the sum of safety time to be a buffer. However, the C&PM adopts a linear procedure, and the size of the calculated buffer increases linearly with the length of the critical chain. The RSEM calculates the square root of the total squares of the difference between two estimates, a safe estimate, and an average estimate, for each activity in the critical chain as the buffer size [15]. Tukul et al. [16] argued that the RSEM could be less affected by the length of the critical chain than the C&PM. In addition, Icmeli and Erenguc [17] proposed a resource utilization factor (RF) to calculate buffers. Roghanian et al. [18] took into account variation of fuzzy numbers for required resources when buffer sizing. Ghaari and Emsley [9] employed Monte Carlo simulation method to validate the efficiency of C&PM in buffer sizing and put forward buffer sizing schemes in multiproject environments considering the level of resource capacity. A major limitation of these existing buffer sizing approaches, however, is that they operate based on an assumption that project activity durations are independent of each other (Li et al., 2012). These approaches, therefore, are not appropriate to cope with rework, which involves extensively the mutual relationships between activities. In a recent study, Zhang et al. [19] proposed a CCPM-based scheduling method that, by integrating the DSM, took into account rework risks in the calculation of resource tightness and information constraints. However, this method failed to recognize that the rework

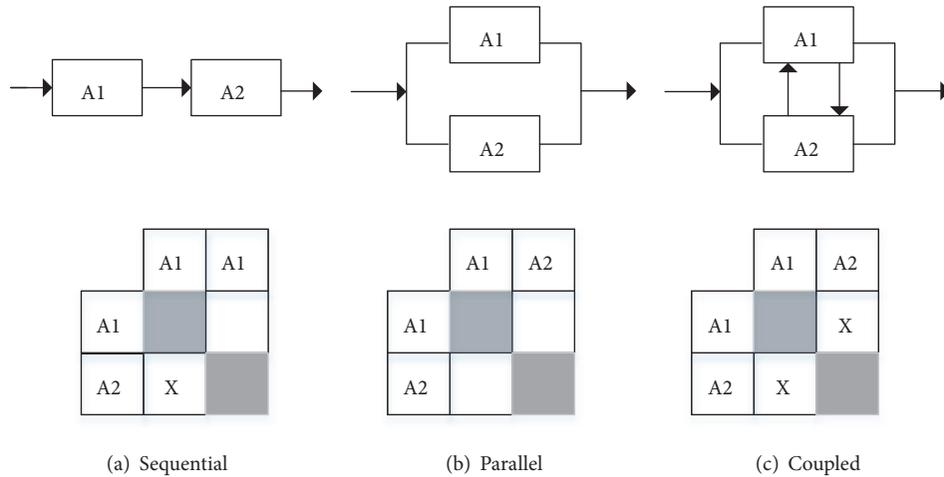


FIGURE 2: Different types of relationship between activities represented in the DSM.

risks could also impact the critical chain, and it identified the critical chain in a traditional manner.

With respect to critical chain scheduling, Francisco and Alfonso [20] extended traditional critical chain scheduling methods by using critical set and critical cloud to avoid ambiguity in the identification of critical tasks. Ma et al. [21] proposed a critical chain scheduling method with a scenario-based proactive robustness optimization. The method was able to yield higher probabilities of on-time project completion than traditional CCPM-based scheduling methods. Goto [22] developed a scheduling framework for projects under limited resources by representing the CCPM method in max-plus algebra to achieve a short lead time. Salama et al. [23] took into account the impacts of resource continuity and variability on activity durations and integrated linear scheduling and CCPM methods to identify multiple critical sequences. Despite active research in CCPM-based linear process scheduling in recent years, the challenge of rework in CCPM-based schedules has barely been addressed [24], where further research is needed.

2.2. DSM as a Process Management Technique. Since Steward [25] first introduced the DSM for the square-matrix-based models of processes, process architecture DSM models (process DSMs) have received the most attention among different application areas of DSM [6]. The DSM uses matrix format to uniformly represent three different logical relationships and dependencies, including sequential, parallel, and coupled, between activities, as shown in Figure 2. An off-diagonal element A_{ij} of the DSM matrix represents with a mark or a number the information flow from activity j to activity i or the dependency relationship between these two activities. If no dependency exists between activities j and i , it will be null or zero. Diagonal element A_{ii} is occasionally used to represent certain characteristics, such as duration, of the activity i , or simply nothing in most cases.

In this matrix-based display, elements above the diagonal are usually regarded as information feedback or iterative activities relationship [5]. Such display provides a useful tool

for analyzing rework and has given rise to a number of relevant studies. The first DSM-based, discrete event Monte Carlo simulation model was proposed by Browning [26], and later extended by Browning and Eppinger [27], to estimate project duration and cost and their variances. They argued that processes with the fewest feedback marks in the DSM may not be necessarily optimal and could be sped up with appropriate increase in overlapping and iterations [6]. They introduced rework probability (RP) matrix, rework impact (RI) matrix, and learning curve (LC) to represent and calculate rework duration [27]. For the element in row i and column j in $[RP]_{ij}$, if $j > i$, which means the element belongs to upper triangular, it describes the probability that the completion of activity i causing rework of activity j ; if $j < i$, which means the element belongs to lower triangular, it describes the probability that after activity j completes rework, the activity i will be influenced. $[RI]_{ij}$ represents the possibility of activity i to be reworked when rework is caused by activity j for $i, j = 1, \dots, n$. LC_i represents the ratio of activity i to be reworked because the participants may benefit from learning and adaption when rerunning the activity. Based on their definition, the values of the elements in these two matrices are between 0 and 1. The values of elements in the two matrices can be collected from historical data or estimated based on risk preferences of the project team. Then the expected rework time of activity i caused by activity j can be calculated by $RP_{ij} \times RI_{ij} \times d_i \times LC_i$, where d_i represents the duration of activity i .

A number of other DSM simulations have adopted the above framework, with certain extensions to account for additional constraints in process scheduling. For instance, Cho and Eppinger [28] proposed a heuristic for solving stochastic, resource-constrained project scheduling problems in an iterative project network. Lévardy and Browning [29] accounted for technical performance characteristics besides duration and cost, by setting up a superset of general classes of activities, each with modes that vary in terms of inputs, duration, cost, and expected benefits. Meanwhile, a number of studies have looked into transferring process DSM to a

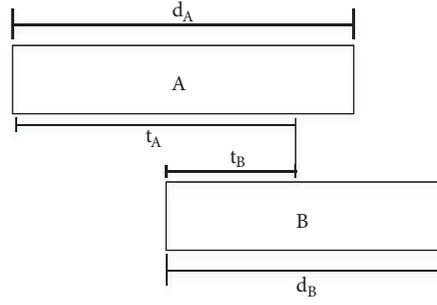


FIGURE 3: Information transfer process with overlapping.

TABLE 1: Nomenclature for all parameters in the paper.

Parameter	Notation	Parameter	Notation
A	Initial state matrix: $[A]_{ij}$ shows duration of activity i ; for the non-diagonal elements, if there is information delivered from j to i , then the value of $[A]_{ij}$ is non empty; else it is empty.	SD	The activity duration variance matrix
P_{ij}	The predecessor time factor from j to i	ES_i	Earliest start time of activity i
S_{ij}	The successor time factor from j to i	EF_i	Earliest finish time of activity i
D	Activity time matrix: the diagonal elements are activity duration and the non-diagonal elements are minus infinity	TT	The length of critical path
R	Activity relationship matrix: if there is information delivered from j to i , then the value of $[R]_{ij}$ is 0; else it is ε .	F	Activity deviation matrix: $[F]_{ij}$ shows the deviation between activity j and activity i
x_E^+	The early finish time for activity	x_0	0 vector: the number of dimension equals the number of activities
$[\]_{n \times 1}$	N-dimension vector of certain value	x_E^-	The early start time for activity
e	The activity element in matrix R, but the value is 0 in the algorithm	RT_i	The first and second rework time as well as rework time the last critical activity cause other activities to generate, respectively, for $i=1,2,3$
w	For the diagonal elements, if the corresponding activity is on the critical path, then the value is 0; or the value is ε .	TF	The total float of each activity
ε	Negative infinity in max-plus	v	For the diagonal elements, if the corresponding activity is on the non-critical path, then the value is 0; or the value is ε .
$[RI]_{ij}$	Rework impact matrix	$[RP]_{ij}$	Rework probability matrix
PT	Estimated project completion time	RwB_i	Rework buffer of activity i

more effective process management tool that is closer to traditional process management tools and expressions. A time factor was therefore introduced [30]. The time factor is defined based on the overlapping relationship between the activities. As shown in Figure 3, d_i and d_j are defined as the duration of activities i and j , respectively, t_i as the duration between the start of activity i and the time when information transition from activity i to activity j is completed, and t_j as the duration between the start of activity j and the time the information transition is completed. A predecessor time factor is defined as $P_{ji} = t_i/d_i$ and a successor time factor as $S_{ji} = t_j/d_j$, where $0 < P \leq 1, 0 \leq S < 1$. Two matrices, P and S , are defined by these two factors. The project duration

is then calculated according to (1), and a nomenclature is provided in Table 1.

$$\begin{aligned}
 ES_i^* &= \max \{ ES_j + P_{ij} \times P_{jj} - S_{ij} \times S_{ii} \} \\
 ES_i &= \max (ES_i^*, 0) \\
 EF_i &= ES_i + A_{ii} \\
 TT &= \max [EF_i]
 \end{aligned} \tag{1}$$

In short, prior research has explored the possibility of using the DSM to predict and control rework in project scheduling and planning and has achieved noticeable progress. However, DSM lacks the capability of functioning as

a stand-alone project management technique, for that DSM's matrix-based expression is difficult to be effectively converted to network diagram-based expression, sometimes leading to ambiguity and confusion [5]. Therefore, it is challenging to apply DSM to the process management of construction projects, and DSM needs to be extended and improved with further research efforts.

3. The Critical Chain Design Structure Matrix Method

In this section, we are going to demonstrate how to use the CCDSM method to generate the project schedule. We improve the max-plus algorithm to reduce computational load in generating project schedules and introduce rework buffers to evaluate and address rework risks.

3.1. Construction Project Scheduling with the Max-Plus Algorithm. A discrete event system is state-discrete and event-driven. Its state evolution depends on asynchronous discrete events occurring at discrete points over time [31]. Simple discrete event systems are usually linear systems in which state and output variables for all possible input variables and the initial state satisfy superposition principle of the system. A construction project can generally be seen as a linear system that consists of a number of discrete events.

The max-plus algorithm provides an alternative way to describe the discrete event system. Based on the max-plus algorithm, the status of all discrete events, denoted as vector $x(k)$, in a system can be used to describe the status of the system, and adjacent statuses are interchangeable with a limited number of linear changes, as follows:

$$\begin{aligned} x(1) &= A_1 x(0) \\ x(2) &= A_2 x(1) \\ &\vdots \\ x(k) &= A_k x(k-1) \end{aligned} \quad (2)$$

Based on the above equation, if the linear changes at every stage, $A_1, A_2 \dots A_k$, are known, then the final status of the system, $x(k)$, can be derived from the initial status of the system $x(0)$. Using this as a starting point, the max-plus algorithm introduces four types of operations to describe the discrete event systems. For two $m \times m$ matrices X and Y denoted as $X, Y \in O^{m \times m}$, whose elements are nonnegative numbers or negative infinity, these operations are defined as follows:

$$\begin{aligned} [X \oplus Y]_{ij} &= \max([X]_{ij}, [Y]_{ij}) \\ [X \wedge Y]_{ij} &= \min([X]_{ij}, [Y]_{ij}) \\ [X \otimes Y]_{ij} &= \bigoplus_{k=1}^m ([X]_{ik} + [Y]_{kj}) \\ [X \odot Y]_{ij} &= \bigwedge_{k=1}^m (-[X]_{ik} + [Y]_{kj}) \end{aligned} \quad (3)$$

where $X \otimes Y$ could be further simplified as XY , and $X^n = \underbrace{XX \dots X}_n$.

To apply the max-plus algorithm to schedule management, three new matrices need to be developed to describe relevant project information. These matrices include (1) activity duration matrix, denoted as D , which is developed by filling the diagonal elements with each activity's duration and the nondiagonal elements with negative infinity; (2) activity deviation matrix, denoted as F , which is developed based on the rule that $[F]_{ij}$ represents the deviation of the start time of activity j and activity i ; and (3) activity relationship matrix, denoted as R , which is developed based on the rule that if there is information delivered from activity j to activity i , then the value of $[R]_{ij}$ is 0; otherwise the value is negative infinity.

According to the max-plus algorithm [32], we have the following.

The early finish time for each activity is calculated as

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

where $(F)^* = e \oplus F \oplus \dots \oplus (F)^{l-1}$. The (i, j) -th element of matrix $(F)^*$ means the largest deviation of the start time of activity j and activity i , if activities i and j are on one or more paths at the same time; otherwise, it is e . For simplicity, the elements of vector x_0 are set as e , which means the beginning of activities in the project will not be affected by other construction projects.

The length of critical path, denoted as the maximum of early finish time for all activities, is calculated as

$$TT = \max[x_E^+] \quad (5)$$

The early start time is calculated as the difference between early finish time for each activity and its duration according to

$$x_E^- = D \odot x_E^+ \quad (6)$$

The late start time of activity i is calculated as the difference between length of critical path and the sum of activity i 's duration as well as subsequent critical activities' durations according to

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

The total float is calculated as

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

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

It needs to be noted that when determining floats, the max-plus algorithm is run in simple matrix form which significantly improves the computational efficiency and is more applicable to large-scale projects [33]. The definitions of parameters in the above equations are summarized in the nomenclature in Table 1.

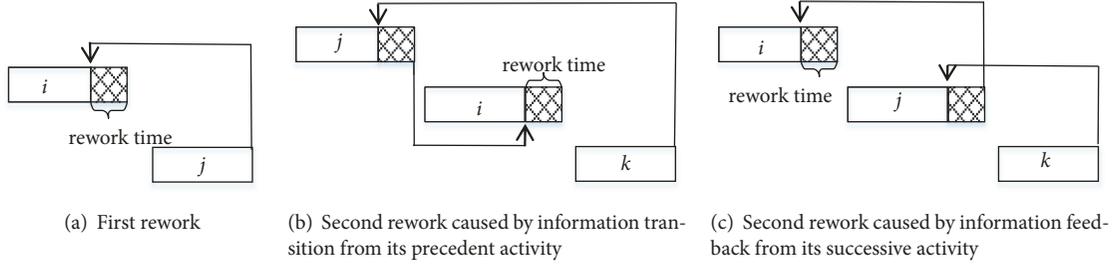


FIGURE 4: First rework and second rework.

3.2. Determining Project Rework Buffer. It is widely believed that information uncertainty is the main cause of rework [34]. Such uncertain information may transmit from upstream activities to downstream activities or feedback in the opposite direction, creating information flow. Information flow interactions lead to rework risks in construction projects and consequently bring about more rework time. There are two main situations that rework exists.

(1) First rework: as depicted in Figure 4(a), i is a precedent activity of j , and they perform sequentially according to relationships. After it is completed, the performance information of j will be generated and transmitted to i , which may result in rework of i . The first rework time, shown as the shadow area in Figure 4(a), can be calculated by [27]

$$[RT_1]_{ij} = RP_{ij} \cdot RI_{ij} \cdot LC_i \cdot D_{ii} \quad (9)$$

Then the total first rework time of i caused by all its downstream activities can be obtained by

$$[RT_1]_i = \sum_{j=i+1}^n RP_{ij} \cdot RI_{ij} \cdot LC_i \cdot D_{ii} \quad (10)$$

where n represents the number of activities in the schedule.

(2) Second rework: as depicted in Figures 4(b) and 4(c), after completing the rework of j caused by k , j transmits or feeds back some revised information to i , which may cause rework of i . Such two forms are defined as the second rework in the paper. The second rework times, shown as the shadow areas in Figures 4(b) and 4(c), can be calculated by

$$[RT_2]_{ijk} = RP_{ij} \cdot RP_{jk} \cdot RI_{ij} \cdot LC_i \cdot D_{ii} \quad (11)$$

Then the total second rework time of i caused by all its downstream activities can be obtained by

$$[RT_2]_i = \sum_{j=1}^{k-1} \sum_{k=i+1}^n RP_{jk} \cdot RP_{ij} \cdot RI_{ij} \cdot LC_i \cdot D_{ii} \quad (12)$$

To proactively account for the uncertainty caused by interactivity relationships and the associated rework risks, a rework buffer (RwB) is introduced in the proposed CCDSM. Conceptually, RwB assesses and compresses the duration uncertainty caused by rework and is placed after the completion of rework activities in project schedules. RwB is designed to warn project participants from how much workload may

increase for each rework activity, so as to prepare the project participants for sufficient time and resources.

The rework buffer of i can be obtained by

$$RwB_i = [RT_1]_i + [RT_2]_i \quad (13)$$

For the last critical activity θ in the schedule, considering all reworks it causes other activities to generate will take place after its completion and have an impact on project completion time; therefore, its rework buffer is defined as the sum of the total rework time it causes other activities to generate and its total rework time caused by other activities.

The total rework time it causes all other activities to generate can be calculated by

$$[RT_3]_{\theta} = \sum_{i=1}^n RP_{i\theta} \cdot RI_{i\theta} \cdot LC_i \cdot D_{ii} + \sum_{i=1}^n \sum_{j=1}^n RP_{ij} \cdot RP_{j\theta} \cdot RI_{ij} \cdot LC_i \cdot D_{ii} \quad (14)$$

The rework buffer of θ then can be obtained by

$$RwB_{\theta} = [RT_1]_{\theta} + [RT_2]_{\theta} + [RT_3]_{\theta} \quad (15)$$

3.3. Determining Project Buffers and Feeding Buffers in Max-Plus Representation

3.3.1. Determining Project Buffers. Previous studies on the CCPM-max-plus representation adopted the C&PM method for calculating project buffers, shown as (16), and feeding buffers [33].

$$PB = \frac{\max [(D \otimes w \otimes R)^* (D \otimes w) x_0]}{3} \quad (16)$$

where (i, j) -th element of matrix $(D \otimes w \otimes R)^*$ means the cumulative time of duration of activity α , which is on the critical chain and range from $i + 1$ to j ; (i, j) -th element of matrix $D \otimes w$ satisfies that if activity i is on the critical chain, then (i, j) -th element is duration of activity I ; otherwise it is ε .

However, the C&PM method lacks sound mathematical foundation and overestimates project durations, resulting in a waste of time and resources. Alternatively, the C&PM is replaced by the RSEM, which is based on the large number law and central limit theorem, in the CCDSM to calculate the PB:

$$PB = \sqrt{\sum_{i \in CP} SD_i} \quad (17)$$

where SD_i is the variance of duration of activity i on the critical path.

To calculate the PB in max-plus representation, a matrix denoted as T_p is introduced. For the element in row i and column $i - 1$ in $[T_p]_{ij}$, if activity i is critical activity, it will be filled with the variance of duration of activity i ; else, it will be zero. For other elements in $[T_p]_{ij}$, it is filled with zero. The PB then can be represented in max-plus algebra as

$$PB = \sqrt{C_0^T \otimes T_p^* \otimes C_0} \quad (18)$$

where $C_0^T = \underbrace{(e, e, \dots, e)}_{n+1}$.

3.3.2. Determining Feeding Buffers. In order to insert feeding buffers in place to protect 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 $[\text{diag}()]_{ij} = \{x_i : \text{if } i = j, \varepsilon : \text{else}\}$, and a new vector can be obtained by the operator $\#$ as $\{[x]_i\}^\# = \{\varepsilon : \text{if } [x]_i \neq \varepsilon, e : \text{if } [x]_i = \varepsilon\}$.

$$FB_0 = \sqrt{\text{diag} \left\{ \left[R_{\beta\alpha} \otimes \text{diag}(v'^\#) \otimes \text{diag}(SD) \right]^* \otimes C_1 \right\} \otimes \text{diag} \left[\text{diag}(SD) \otimes \text{diag}(v') \otimes C_1 \right] \otimes C_1} \quad (20)$$

Additionally, the calculation of feeding buffer should be adjusted using the following equation to meet the constraint that the size of feeding buffer cannot exceed total float:

$$[FB]_\gamma = \min \left([FB_0]_\gamma, [TF]_\gamma \right) \quad (21)$$

3.4. Implementation of the CCDSM with Max-Plus Linear Expression. The max-plus-based implementation of the CCDSM is summarized in the following seven steps. It needs to be noted that, as the max-plus algorithm cannot solve nonlinear conversions such as multiplication, the input data for the project rework problem including matrices A , P , S , RP , and RI needs to be preprocessed:

Step 1 (preprocess A). Extract all diagonal elements of matrix A and form a diagonal matrix D .

Step 2 (preprocess P and S). Combine matrices P and S to generate matrix F according to the following equation:

$$[F]_{ij} = [P]_{ij} \times [P]_{jj} - [S]_{ij} \times [S]_{ii} \quad (22)$$

Extract all nonnull elements of matrix F and use them to form matrix R . Then update R , by replacing all nonnull elements with e and all null elements with ε .

Step 3 (preprocess RP and RI). Based on (9)-(15), merge matrices RP and RI to generate the vector RwB , whose i -th element is the corresponding rework buffer of activity i .

In preparation of the location and size of the feeding buffer, another two vectors are introduced, denoted as w and v . Matrix w is developed based on the rule that for a diagonal element; if the corresponding activity is on the critical path, then the element value is 0; otherwise, the element value is ε . Matrix v is developed based on the rule that, for a diagonal element, if the corresponding activity is on the noncritical path, then the value is 0; otherwise, the value is ε . 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:

$$[v']_\gamma = [R_{\beta\alpha}^T \otimes w]_\gamma \quad (19)$$

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

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

Step 4 (calculate the length of critical path). Based on (4)-(8), the length of critical path TT and other parameters, including the total float time TF and critical path are determined.

Step 5 (calculate matrices w and v). According to the critical path determined in Step 4, matrices w and v can be generated. Meanwhile, the activity duration variance matrix SD can be generated based on data collected from the project to calculate the buffers size;

Step 6 (calculate buffer sizes). Based on (17)-(21), project buffer PB and feeding buffers FB s are calculated with RSEM and represented in max-plus algebra.

Step 7 (generate the schedule). As the principle of CCPM, the feeding buffers should be placed on the noncritical chains prior to the joints of the critical chain and noncritical chains, and the project buffer should be placed at the end of the schedule to protect the whole project process. Then, the estimated project time PT can be calculated based on

$$PT = TT + PB + RwB_0 \quad (23)$$

4. Case Study

In this section, the proposed CCDSM method is implemented in a case project to demonstrate its feasibility and effectiveness in addressing rework risks in project schedule

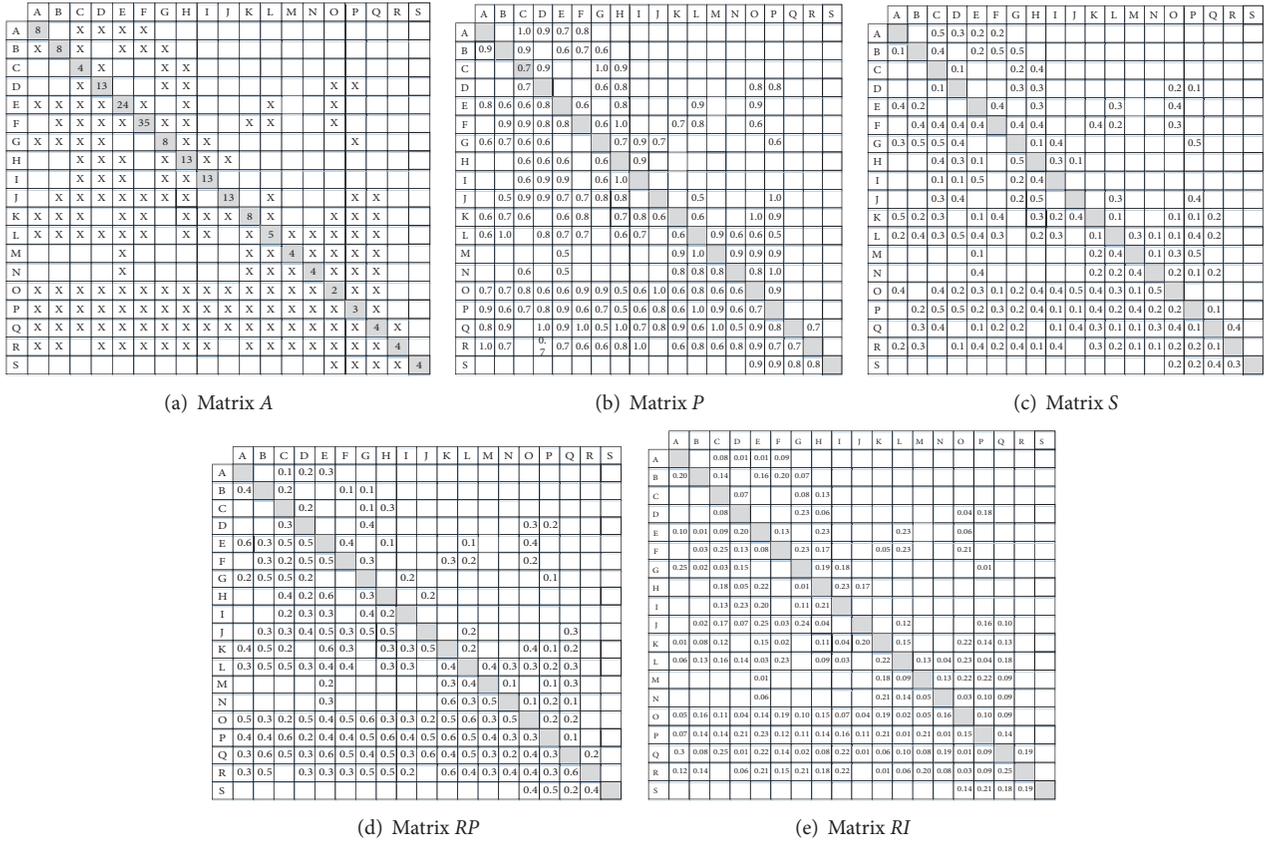


FIGURE 5: Project information in the case project.

management. The settings of the case project are first presented, followed by descriptions of implementation of the CCDSM in the case project. The schedule generated with the CCDSM method is assessed in detail and compared with the schedules generated with traditional CCPM- and DES-based methods.

4.1. Case Project Settings. The case used in this study was derived from a modular real estate development project firstly introduced in [35] and further described in [36]. The matrix A of the case is shown in Figure 5(a). This case consisted of 19 major activities and 183 interactivity relationships, including 65 rework relationships represented by elements above the diagonal of matrix A . It is assumed that rework can propagate up to twice to avoid infinite loop in the computation. The parameters for each activity and the interactivity relationships were derived based on literature and empirical evidence, and are shown in Figures 5(b), 5(c), 5(d), and 5(e) and summarized in Table 2. Large-scale rework relationships and complex connections between activities were observed in the case project, which was representative of typical construction projects in reality.

4.2. Implementation of the CCDSM Method. The proposed CCDSM method was implemented in the case project following the steps explained in the last section and the results

were shown below. Noticeably, for better understanding of scheduling process, the implementation process of the CCDSM method is reorganized as below.

Step 1. Matrices D and R were derived from matrix A , as shown in Figure 6. The duration of each activity was presented in the diagonal of the matrix D .

Step 2. Matrix F (see Figure 7) was derived from matrices P and S according to (22). This step calculated the activity deviation matrix based on the predecessor time matrix and the successor time matrix.

Step 3. Optimize activity sequence and update matrices D , R , RP , RI , and F according to new activity sequence. To reduce project rework and obtain near-optimal project completion time, the genetic algorithm (GA) was applied to optimize activity sequence. The GA is a metaheuristic method that searches for optimal solutions using processes similar to those in natural selections and genetics [37]. In the paper, minimization of total length of rework path first proposed by Gebala and Eppinger [38] was used as the objective function to calculate the optimal sequence of activities:

$$f = \sum_{i=1}^n \sum_{j=i+1}^n (j-i) \cdot w(i, j) \quad (24)$$

TABLE 2: Activity parameters in the case project.

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

	F	C	A	I	J	R	K	P	D	M	O	B	H	E	N	L	G	Q	S
F	30																		
C		3																	
A			5																
I				10															
J					10														
R						3													
K							5												
P								2											
D									10										
M										2									
O											1								
B												5							
H													10						
E														20					
N															3				
L																3			
G																	5		
Q																		3	
S																			3

(a) Matrix D

	F	C	A	I	J	R	K	P	D	M	O	B	H	E	N	L	G	Q	S
F																			
C																			
A	e	e																	
I		e																	
J	e	e																	
R	e		e	e															
K	e	e	e	e	e														
P	e	e	e	e	e	e													
D		e					e												
M							e	e											
O	e	e	e	e	e	e	e	e	e										
B	e	e	e																
H		e		e	e			e											
E	e	e	e					e	e	e	e								
N							e	e	e	e			e						
L	e	e	e	e			e	e	e	e	e	e	e	e	e				
G		e	e	e				e	e			e	e						
Q	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e		
S						e	e		e										e

(b) Matrix R (element ε is omitted)

FIGURE 6: Matrix D & R.

where $w(i, j)$ represents rework probability of activity i caused by activity j .

The parameter settings were selected as follows: population size set as 50, number of generations set as 150, crossover probability set as 0.95, and mutation probability set as 0.08. The GA process reported an optimal schedule as [P, Q, H, E, R, B, C, K, N, L, M, F, D, S, A, G, J, I, O]. The GA convergence process is shown in Figure 8.

Step 4. RwB was generated from matrices RP and RI based on (9)-(15). The goal of this step was to factor in and calculate the rework time of the case project. RwB was [3, 4, 6, 10, 5, 1, 1, 3, 1, 2, 1, 15, 1, 1, 1, 1, 1, 9].

Step 5. To generate reliable project schedules, the most likely duration of each activity, which has been widely used and accepted by project teams in prior research [39], was selected

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S			
A			1	10	15	28																
B	7		1		13	23	2															
C				13			8	11														
D			2				1	7							-1	2						
E	-3	1	3	10		15		4				-1			-8							
F		-6	11	-4	5		12	-1			-8	-1			-9							
G	3	2		6				9	9											-1		
H			-3	4	14				8	9												
I			2	11	16		3	8														
J		5	-1	8	18	28	4	5								-1				-1	3	
K	2	5	2		15	28		7	9	5										2	3	2
L	4	6	1	10	16	27		7	9		5		3	2	1							2
M						13					7	4		4	1	1	4					
N					12						6	4	2		1	3	3					
O	5	6	2	8	16	35	7	7	8	12	4	4	3	2		3	4					
P	8	5	2	11	23	22	6	6	9	10	4	5	3	2	1					2		
Q	7	7	2	14	23	37	4	13	9	10	6	3	4	1		2				2		
R	7	5		11	17	23	4	10	11		4	4	2	3	1	2	3					
S															1	2	2	3				

FIGURE 7: Matrix F (element ε is omitted).

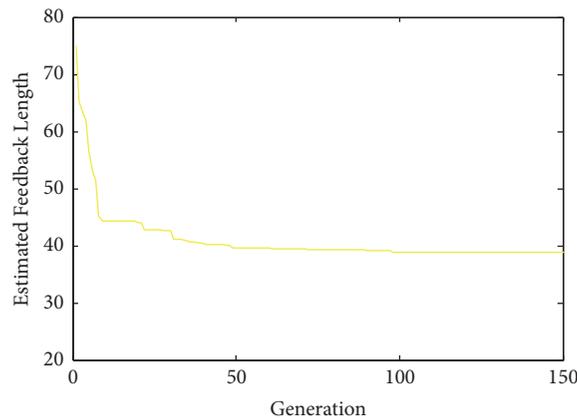


FIGURE 8: Convergence curve in the GA-based optimization.

to calculate the estimated project duration. The critical chain was reported as $[H, E, B, K, N, L, F, A, G, J, O]$.

Step 6. The activity duration variance matrix SD was generated with 1,000 Monte Carlo simulations of the activities' duration that follows a beta distribution.

Step 7. Buffers were calculated based on (17)-(21). The project buffer was 36 days, and the feeding buffer was $[3, 4, 0, 0, 38, 0, 4, 0, 0, 0, 6, 0, 10, 38, 0, 0, 0, 0, 2, 0]$.

Step 8. The estimated project duration was calculated as 102 days, based on (23). The project buffer was placed at the end of the schedule, and the feeding buffer was placed on the noncritical chains prior to the joints of the critical chain and noncritical chains, following the principle of CCPM.

It was worth mentioning that overlapping was observed among various activities, including critical activities, in the

above schedule, which illustrated real-world high-frequency interactions among activities in the project.

To assess the performance of the CCDSM method and the quality of the project schedule it generated, the execution of case project was simulated 1000 times using Monte Carlo approach, where the sampling of actual activity durations followed beta distribution. The simulated actual project durations were depicted in Figure 9. It can be seen in the figure that under rework assumption, the process schedule of 102 days had a completion probability of 98.1%, indicating that the proposed CCDSM method was able to provide satisfactory assurance of on-time project completion under varying scheduling scenarios.

4.3. Comparison of the CCDSM Method with Traditional CCPM and DES Methods. This subsection further compares the performance of the CCDSM method with traditional

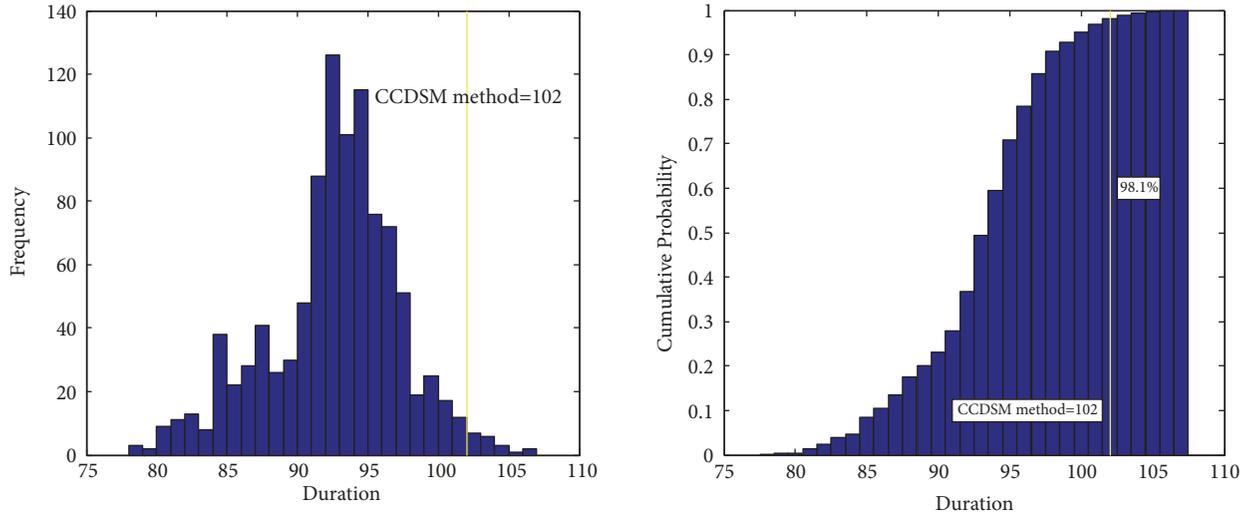


FIGURE 9: Simulated actual project durations and on-time project completion probability of the CCDSM-based schedule.

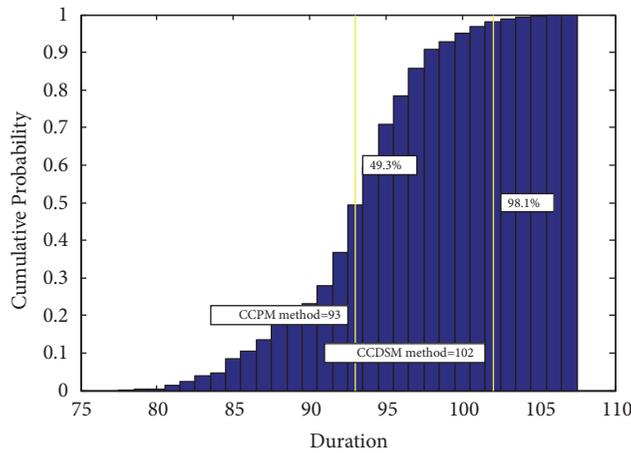


FIGURE 10: Comparison of on-time project completion probabilities between the CCDSM- and CCPM-based schedules.

CCPM and DES methods to assess its effectiveness in addressing rework risks.

A CCPM-based schedule was generated with traditional buffers including project buffer and feeding buffer and without rework buffer. The results indicated that the estimated project durations of the CCPM-based schedule were 93 days, including project buffers of 36 days. When the estimated project duration of CCPM-based schedule was superimposed in the Monte Carlo-simulated project durations, as depicted in Figure 10, it can be seen that the on-time completion probability was 49.3%. The results suggested that the CCDSM-based schedule, which took into consideration rework relationships and had a duration of 102 days, significantly outperformed the CCPM-based schedule in ensuring on-time completion of the case project that was faced with typical rework risks.

Furthermore, a DES-based schedule was generated by implementing DES algorithm, in which activity durations following a beta distribution were generated at random to compute the project duration. When the DES algorithm was

used to generate project schedules under rework scenarios, a probabilistic judgment is performed firstly on whether or not each activity causes rework, and then an iterative simulation process began until all activities were finished. Specifically, the DES algorithm proposed by Browning and Eppinger [27] was adopted in this study, with the following minor modifications to make it more applicable to the case project and comparable to the CCDSM method: (1) the optimized activity sequence in Section 4.2 was employed in the DES process; (2) activity i did not started until STS logic tie with its each predecessor activity j ; that is $[F]_{ij}$, was satisfied; (3) rework probability of activity i would decrease by 50% each time it reworked [28].

To compare the CCDSM-based schedule with the DES-based schedule, the execution of the case project using the DES-based schedule was simulated 1,000 times using Monte Carlo approach, and the simulated actual project durations are depicted in Figure 11. The results showed that the estimated project duration of the DES-based schedule was 115 days, which was 13 days longer than that of the

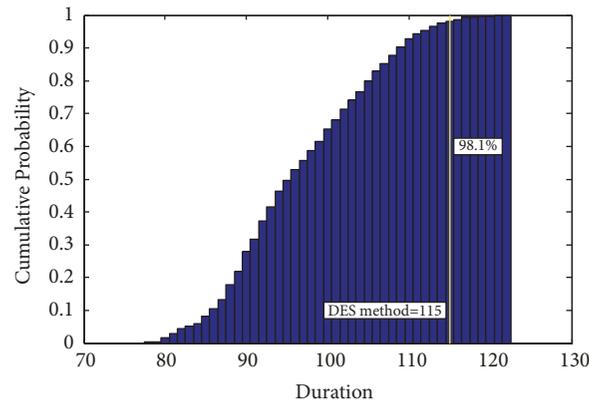


FIGURE 11: On-time project completion probability of the DES-based schedule.

CCDSM-based schedule, with the same on-time completion probability of 98.1%. In other words, the CCDSM method outperformed the DES method by ensuring the same on-time completion probability with a shorter project schedule. Moreover, the CCDSM method was also advantageous over the DES-based method in that it enabled effective control of project schedules during project implementation, which could be done by monitoring consumption of time buffers, leading to reduction in project overrun risks.

5. Conclusions

Construction projects are constantly challenged by rework risks, which have largely remained unaddressed by existing project scheduling. This paper proposed the CCDSM method for developing project schedules that are adaptive to rework instances during project execution. This method models large-scale rework relationships among activities with the introduction of a new rework buffer in traditional CCPM method. The method allows analysis and visualization of the schedules and utilizes max-plus algorithm to transform complex logic relationships into simple matrix operations, reducing the computational load of generating process schedules. A case study was conducted to demonstrate the implementation of the CCDSM method and assess its effectiveness in addressing rework risks. The results showed that CCDSM-based schedule outperformed the CCPM-based schedule in ensuring on-time completion of the case project that was faced with typical rework risks. In addition, in comparison to the DES method, the CCDSM performed better by providing a smaller project duration with the same probability of on-time project completion.

The CCDSM method contributes to construction management in the following ways. First, the CCDSM method absorbs risks of rework and various other uncertainties to generate the reliable project schedule, enabling managers to predict project durations more accurately before projects start. Based on accurate prediction of project durations, the negotiation of contracts and supply of resources can be conducted in a more reasonable manner, avoiding costs overruns, schedule delays, and even project failures. Second,

based on generated project schedule with time buffers, effective monitoring and control of construction progress can be realized. Overall speaking, the proposed CCDSM method provides a promising solution to mitigate rework risks in construction projects and protect projects from undesirable delays caused by possible rework instances.

Meanwhile, the CCDSM bears several limitations that would benefit from future improvement. First, only the STS logic tie between activities is tested in this study. Techniques for integrating multiple logic ties including STS, STF, FTS, and FTF deserve further attention. Second, resource conflicts and leveraging are considered beyond the scope of this study, but it is a critical challenge that should be addressed in the context of CCDSM-based scheduling in future research. Third, the parameters required to implement the CCDSM method are hard to obtain for a specific project. To employ CCPM or max-plus algorithm for project scheduling, it is necessary to obtain the estimated activity durations under various completion rates and the logic tie between activities. Similarly, the rework relative parameters including RP , RI , and LC are required for implementation of DSM-based method. As an integration method of CCPM, max-plus algorithm, and DSM, the implementation of CCDSM requires all the above parameters to be obtained. Lastly, methods for dynamic updating of buffer sizes during project execution are needed in order to better prepare project teams for changing project environment and dynamic project progress.

Data Availability

All data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) [Grant no. #71671128].

References

- [1] Y. Li and T. R. Taylor, "Modeling the impact of design rework on transportation infrastructure construction project performance," *Journal of Construction Engineering and Management*, vol. 140, no. 9, p. 04014044, 2014.
- [2] B. 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.
- [3] E. K. Simpeh, R. Ndiokubwayo, P. E. Love, and W. D. Thwala, "A rework probability model: a quantitative assessment of rework occurrence in construction projects," *International Journal of Construction Management*, vol. 15, no. 2, pp. 109–116, 2015.
- [4] P. W. Morris, J. K. Pinto, and S. Jonas, *The Oxford Handbook of Project Management*, Oxford University Press, Oxford, UK, 2012.
- [5] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*, MIT Press Books, 2012.
- [6] 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.
- [7] A. Kulkarni, D. K. Yadav, and H. Nikraz, "Aircraft maintenance checks using critical chain project path," *Aircraft Engineering Aerospace Technology*, vol. 89, no. 6, pp. 879–892, 2017.
- [8] J. Trojanowska and E. Dostatni, "Application of the theory of constraints for project management," *Management and Production Engineering Review*, vol. 8, no. 3, pp. 87–95, 2017.
- [9] M. Ghaari and M. Emsley, "Buffer sizing in CCPM portfolios with different resource capacities," *International Journal of Information Technology Project Management*, vol. 8, no. 3, pp. 40–51, 2017.
- [10] E. M. Goldratt, *Critical Chain: A Business Novel*, MA: North River Press, Great Barrington, United States, 1997.
- [11] S. Van de Vonder, E. Demeulemeester, and W. Herroelen, "Proactive heuristic procedures for robust project scheduling: An experimental analysis," *European Journal of Operational Research*, vol. 189, no. 3, pp. 723–733, 2008.
- [12] O. Lambrechts, E. Demeulemeester, and W. Herroelen, "Time slack-based techniques for robust project scheduling subject to resource uncertainty," *Annals of Operations Research*, vol. 186, no. 1, pp. 443–464, 2011.
- [13] D. Trietsch and K. R. Baker, "PERT 21: Fitting PERT/CPM for use in the 21st century," *International Journal of Project Management*, vol. 30, no. 4, pp. 490–502, 2012.
- [14] 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, no. 12, p. 04014055, 2014.
- [15] R. C. Newbold, *Project Management in The Fast Lane: Applying The Theory of Constraints*, CRC Press, Boca Raton, Florida, London, 1998.
- [16] 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.
- [17] O. Icmeli and S. S. Erenguc, "A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows," *Management Science*, vol. 42, no. 10, pp. 1395–1408, 1996.
- [18] 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.
- [19] J. Zhang, X. Song, and E. Díaz, "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.
- [20] F. A. Rivera and A. Duran, "Critical clouds and critical sets in resource-constrained projects," *International Journal of Project Management*, vol. 22, no. 6, pp. 489–497, 2004.
- [21] G. Ma, L. Gu, and N. Li, "Scenario-based proactive robust optimization for critical-chain project scheduling," *Journal of Construction Engineering and Management*, vol. 141, no. 10, p. 04015030, 2015.
- [22] H. Goto, "Forward-compatible framework with critical-chain project management using a max-plus linear representation," *OPSEARCH*, vol. 54, no. 1, pp. 201–216, 2017.
- [23] T. Salama, A. Salah, and O. Moselhi, "Integration of linear scheduling method and the critical chain project management," *Canadian Journal of Civil Engineering*, vol. 45, no. 1, pp. 30–40, 2018.
- [24] C. W. Hu, X. D. Chen, and L. H. Wu, "The setting method of project buffer in critical chain management of mould manufacturing project based reworking," *Advanced Materials Research*, vol. 317–319, pp. 418–422, 2011.
- [25] D. V. Steward, *Systems Analysis and Management: Structure, Strategy and Design*, Irwin/McGraw-Hill, Boston, 1981.
- [26] T. R. Browning, *Modeling and analyzing cost, schedule, and performance in complex system product development [Ph.D. thesis]*, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1998.
- [27] 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.
- [28] 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.
- [29] V. Lévardy and T. R. Browning, "An adaptive process model to support product development project management," *IEEE Transactions on Engineering Management*, vol. 56, no. 4, pp. 600–620, 2009.
- [30] J. U. Maheswari and K. Varghese, "Project scheduling using dependency structure matrix," *International Journal of Project Management*, vol. 23, no. 3, pp. 223–230, 2005.
- [31] C. G. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems*, Springer, New York, NY, USA, 2nd edition, 2008.
- [32] S. Yoshida, H. Takahashi, and H. Goto, "Modified max-plus linear representation for inserting time buffers," in *Proceedings of the IEEE International Conference on Industrial Engineering & Engineering Management*, IEEE, 2010.
- [33] S. Yoshida, H. Takahashi, and H. Goto, "Resolution of time and worker conflicts for a single project in a max-plus linear representation," *Industrial Engineering & Management Systems*, vol. 10, no. 4, pp. 279–287, 2011.
- [34] S. Muralidharan, "Factors affecting rework in construction project," *International Journal of Engineering Sciences & Research Technology*, vol. 1, no. 5, pp. 578–584, 2016.
- [35] B. B. E. Bulloch and J. Sullivan, *Application of the Design Structure Matrix (DSM) to the real estate development process*.

- [Master, thesis], Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2009.
- [36] S. D. Eppinger, S. Bonelli, and A. M. Gonzalez, "Managing iterations in the modular real estate development process," *Reducing Risk in Innovation: Proceedings of the 15th International DSM Conference Melbourne, Australia, 29-30 August 2013*, pp. 37–44, 2013.
- [37] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer Science & Business Media, 2000.
- [38] D. A. Gebala and S. D. Eppinger, "Methods for analyzing design procedures," in *Proceedings of the AMSE Third International Conference On Design Theory and Methodology*, vol. 31, pp. 227–233, Miami, Florida, 1991.
- [39] Ö. Hazır, M. Haouari, and E. Erel, "Robust scheduling and robustness measures for the discrete time/cost trade-off problem," *European Journal of Operational Research*, vol. 207, no. 2, pp. 633–643, 2010.

