

Article

# A Critical Design Structure Method for Project Schedule Development under Rework Risks

Guofeng Ma, Jianyao Jia \*, Tiancheng Zhu \* and Shan Jiang

School of Economics and Management, Tongji University, Shanghai 200092, China; 06125@tongji.edu.cn (G.M.); 1830353@tongji.edu.cn (S.J.)

\* Correspondence: jianyaoj@tongji.edu.cn (J.J.); zhutiancheng1992@gmail.com (T.Z.)

Received: 18 November 2019; Accepted: 14 December 2019; Published: 16 December 2019



**Abstract:** In order to overcome the difficulty in quantifying rework by traditional project schedule management tools, this study proposes an innovative method, namely improved Critical Chain Design Structure Matrix (ICCDSM). From the perspective of information flow, the authors firstly make assumptions on activity parameters and interactions between activities. After that, a genetic algorithm is employed to reorder the activity sequence, and a banding algorithm with consideration of resource constraints is used to identify concurrent activities. Then potential criticality is proposed to measure the importance of each activity, and the rework impact area is implicated to indicate potential rework windows. Next, two methods for calculating project buffer are employed. A simulation methodology is used to verify the proposed method. The simulation results illustrate that the ICCDSM method is capable of quantifying and visualizing rework and its impact, decreases iterations, and improves the completion probability. In this vein, this study provides a novel framework for rework management, which offers some insights for researchers and managers.

**Keywords:** schedule management; rework; uncertainty; Design Structure Matrix; critical chain

## 1. Introduction

Project rework is defined as an unnecessary workload to re-execute processes or activities that have not been executed correctly for the first time [1]. Increasing communication and collaboration among participants, which results from project complexity [2], is the main cause of rework. In most projects, rework leads to changes, damage, defects, errors, and other failures, which may finally cause cost overrun and schedule delay [1,3,4].

However, traditional schedule management tools, including the PERT/CPM network, et al. [5] mostly treat rework as uncertainty and fuzziness [6]. These tools all lack a quantifying measure for project rework [7], which brings a difficulty to its effective management. To be specific, rework is an important cause of project schedule uncertainty. If project managers could have a glance at the rework period, the resource required for rework will be prepared in advance. In this vein, project managers can only manage project rework passively. As such, it is necessary to explore a new schedule management tool to quantify project rework for control and deepen our understanding of project rework.

The Design Structure Matrix (DSM) is a network modeling tool used to represent the elements comprising a system and their interactions [8]. The DSM takes the form of an  $N \times N$  matrix and adopts graphical or digital symbols to express the interactions of elements in a cell [9]. Therefore, DSM has natural advantages in dealing with coupled tasks. Among the four main types of DSM [10], process DSM has attracted the most attention of scholars [11]. A complete process DSM consists of at least the following three parts: (1) a structure decomposing the process hierarchically into a single activity; (2) input-output relationships between activities; and (3) other constraints between activities. The expression by process DSM will solve the first two parts, which is the defect of other

schedule management tools. For the third part, advanced object-oriented modeling and database referencing techniques will be employed [8]. For example, the DSM can't indicate mutual resource dependencies between activities, while other process modeling methods such as the Critical Chain Project Management (CCPM) can deal with this issue.

Although the Critical Path Method (CPM) has been widely used to develop and manage project schedules [12], the CPM is not always effective as demands for resources grow [13]. In view of the limitations of CPM in dealing with uncertainties, CCPM solves the bottleneck of "flow" by setting a buffer, which aggregates most of the uncertainties. In addition, the theory of constraints (TOC) [14,15] is applied in project management to deal with resource constraints. Nowadays, CCPM has become the most popular schedule management technique [16]. However, the most current buffer sizing approaches in CCPM assumes that each activity is independent [17]. In other words, most research ignores interactions between activities, which hinders project rework research with CCPM.

Ma et al. [18] first combined CCPM with DSM by the Max-plus algorithm as the Critical Chain Design Structure Matrix (CCDSM) framework to study project rework. They incorporated a genetic algorithm into the CCDSM framework to re-sequence activities. Moreover, Ma et al. [18] also introduced a new rework buffer to represent the impact of rework instances quantitatively. However, they didn't incorporate rework into the activity execution process. In other words, rework may happen when an activity is completed; as a result, the later activities will be all affected as well as the resource requirements. Besides, resource constraints were also ignored in their research [18]. Such shortcomings greatly reduced the practicability of their proposed method. Therefore, an ICCDSM model is conducted in this study, which realizes the combination of CCPM and DSM without the help of max-plus algebra. The proposed method uses a genetic algorithm to sequence activities and employs a banding algorithm to improve concurrency, quantifies the importance and impact area for each activity with the introduction of potential criticality, and visualizes an optimal schedule. An example case consisting of 19 activities is used to verify the method. In summary, the proposed method in this study enables planning and optimizing schedules, as well as identifying critical activities under rework risks.

In the next section, a literature review on project rework management is presented. The third section introduces the proposed method in detail. Then, an example case is used to validate this method in the fourth section. Finally, conclusions are offered in the last section.

## 2. Literature Review

### 2.1. Project Rework

Previous research on project rework focuses mainly on soft research and quantitative measures. In soft research, a great number of scholars have conducted in-depth research on the identification of factors causing rework [4,19,20], the evaluation of rework impact [21–23], and the measures to reduce rework [24–26]. As for quantitative measure, previous research tends to treat rework as uncertainty and deals with fuzziness because of the shortcomings of traditional schedule management tools. In the early 21st century, R&D enterprises began to explore methods to increase the efficiency and predictability of the development process. How to manage rework and how to decrease iterations are the two issues that are dealt with most often. Browning [27] studied the sources of project schedule uncertainty by system dynamics. He found that the three main causes of increasing uncertainty are the number of unintentional iterations, intentional iterations, and its scope. What's more, the number of unintentional iterations increases with the rise of performance uncertainty and design mistakes. The number of intentional iterations mounts up with the rise of performance uncertainty, the iteration productivity, and the degree of activity coupling. On this basis, scholars led by Eppinger began to employ DSM into quantitative rework research. Browning and Eppinger [28] firstly quantified rework by Rework Probability (RP) and Rework Impact (RI), i.e., the risk of rework originates from the probability of occurrence and consequences. Following this framework, Cho and Eppinger [29]

subsequently standardized the expression of RP and RI in the DSM technique [30], which is also the theoretical basis of this study.

## 2.2. Rework Management with CCPM

CCPM aggregates and manages most of the uncertainties in a project by setting buffers. Rework is often included in the uncertainties. There are three main buffers in CCPM, resource buffer (RB), feeding buffer (FB), and project buffer (PB). The three are distinguished by their location and function [12]. Resource buffers, which are placed on the critical chain, ensure the required resource availability when needed [31]. Feeding buffers are added at the end of a non-critical chain [32] to ensure critical activities will not wait for any sub-critical activities. A project buffer, at the end of the schedule, protects against unpredictable failures, accidents, and delays [33].

A considerable amount of research has been done on how to calculate buffer size. Goldratt [14] employed a cut and paste method (C&PM) to calculate project buffer size. However, Bie et al. [17], Herroelen and Leus [34] believed that the 50% expected duration of the critical chain used for project buffer would lead to a long buffer, which was further verified by Ashtiani et al. [35]. Therefore, the root square error method (RSEM) based on central limit theory is usually used to replace the C&PM [36,37], especially in large-scale projects [38]. However, the premise of RSEM is that the duration distribution of each activity is independent. In fact, activities in a project interact with each other. For example, underutilized resources can be adapted to help overutilized resources [39]. Besides, any mismatch between activities comprising a complete process may lead to rework [40]. This makes the calculated project buffer smaller than the theoretical value. Besides, RSEM takes the 50% expected duration as the planned duration, which makes the project buffer size difficult to satisfy practical needs with different risks. Some scholars put forward different schemes to overcome the shortcomings of RSEM. Tukel et al. [36] employed another two new methods to calculate the feeding buffer size, namely the adaptive procedure with resource tightness (APRT) and the adaptive procedure with density (APD). The APRT integrates resource tightness, and the APD employs the complexity of network density. The simulation results show that the project buffer calculated by the two new methods are more accurate than those calculated by RSEM and C&PM.

Long and Ohsato [6] simulated the project uncertainty with fuzzy numbers and calculated the buffer size using a genetic algorithm (GA) and trapezoidal fuzzy theory. The innovation of this method is that expert judgment will be used to judge whether the specific environment of a project is conducive to dealing with specific problems in the absence of historical data. Chu [41] and Zhao et al. [42] introduced a method, based on the consideration of resource tightness, network complexity, and the impact of the managers' risk preference, to ensure appropriate buffer size, and the irrelevance of the number of critical activities. Shi et al. [43] conducted a method for buffer size calculation based on resource tightness, network complexity, risk preference, the cost factor, and resource sustainability. To overcome the shortcoming of taking 50% duration as the project buffer in C&PM, Zhang et al. [44] adopted a new approach to calculate buffer size by considering the activities' information flow instead of simple proportional cutting. Based on project risk, dynamic manual deviation monitoring, and model control, Zhang et al. [45] proposed the effort buffer to improve the accuracy and realize dynamic management in IT projects. In addition, Zhang et al. [46] measured the influence of complex resource tightness on project buffer size in the DSM framework, triggering further research on buffer size.

To sum up, on the one hand, most of the existing research on scheduling and buffer settings relies on the assumption of independence of each activity and ignores their interactions. On the other hand, it is impossible to measure rework accurately due to aggregate uncertainty management in CCPM, which further makes it difficult to conduct effective management on rework.

## 2.3. DSM Application in Rework Management

The common representation for DSM places activity inputs in the rows and activity outputs in the columns (used in this paper). Then marks above the diagonal represent iterations or cycles. To be

specific, an activity begins with inputs from other activities, while the elements above the diagonal indicate that activity will not begin with all of its inputs. In project activities, many of the inputs are information, and thus, can be assumed [8]. Therefore, to minimize such assumptions, most process DSM research focuses on reducing the number of elements in the upper triangle, i.e., decreasing the number of feedbacks. Tarjan [47] identified the coupled activity blocks efficiently with the employment of a depth-first search algorithm. Once the coupled blocks are identified, the sequence of activities in blocks will be rearranged with the tearing algorithm [48], eigenvalue analysis [49], analytic hierarchy process [50,51], clustering algorithm [52], and the genetic algorithm [18,53]. Equilibrium and optimization often involve activity overlap, concurrency, as well as the number of cyclic rework and cost or duration parameters. Some other research paid attention to the impact of rework on project duration. Smith and Eppinger [54] developed a model to predict project duration under different activity logic relationships. Browning and Eppinger [28] firstly proposed the discrete Monte Carlo simulation model based on DSM to predict project duration and cost. According to his research, moderate overlap and rework are helpful to improve schedule performance. Therefore, the least feedback in the process DSM will not necessarily mean the shortest duration. Since then, most DSM scheduling simulation algorithms have been developed on this framework. Cho and Eppinger [30] introduced resource constraints to this framework, and Yassine [55] optimized this framework by considering schedule robustness. Meier [56] also designed a multi-objective genetic algorithm to optimize and explore the time-cost equilibrium problem.

However, although rework has been quantitatively and qualitatively expressed, DSM-based simulation produces uncertain results, which results in the difficulty in identifying the critical chain.

### 3. Proposed Method

ICCDSM is the combination of DSM and CCPM, which adopts their advantages. Specifically, ICCDSM follows the assumption of rework, the scheduling mechanism, and the concept of criticality in the DSM framework. Besides, the critical chain and buffer in CCPM are introduced to the framework of ICCDSM, in which the assumption of rework and criticality are modified. Figure 1 plots the research process.

#### 3.1. Assumptions

##### 3.1.1. Interactions between Activities

From the perspective of information flow, the inputs and outputs of activities are information and therefore, for better or worse, can be assumed [8]. Under such assumptions, rework uncertainty from interactions between activities originates mainly from the uncertainty of information transmission.

As for information transmission between activities, in the sequential and concurrent model, when activity is executed to a certain stage, enough information is generated to judge if activity B will be carried out (i.e., overlapping) [18,57]. Information feedback will be inevitably generated during the execution of activity B, which may lead to partial or total re-execution of activity A, as shown in Figure 2a. For the convenience of modeling, we assume that activity may not begin until all of its upstream activities have been finished (i.e., no overlapping) [57], as shown in Figure 2b.

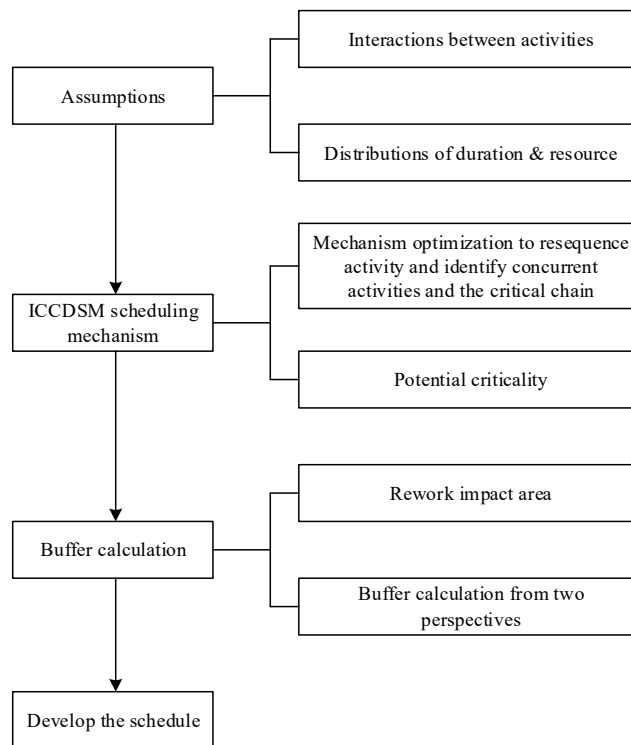


Figure 1. The research process.

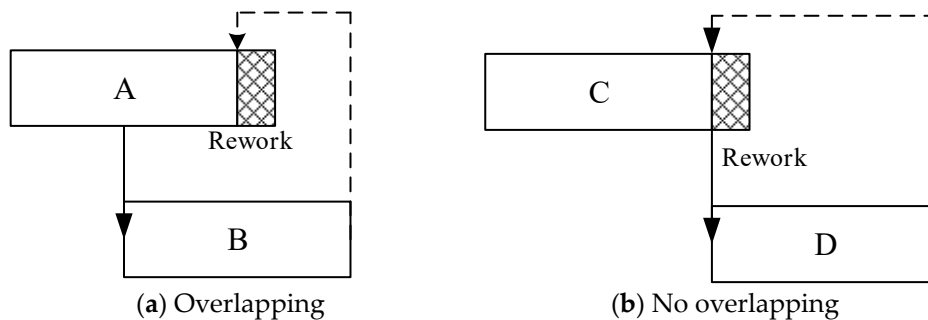


Figure 2. Information interaction between activities.

Based on the above qualitative analysis, the assumptions are as follow:

(1) Rework caused by information interactions occurs by a certain probability [28], i.e., rework probability  $(RP(i, j, r) \in [0, 1](i, j = 1, \dots, n; r = 1, 2, \dots))$ : the probability of the  $r$ -th rework of activity  $i$  caused by activity  $j$ ).  $RP(i, j, 1)$  is shown in Figure 3;

(2) The probability of rework is reduced by 50% after each rework [30].  $RP(i, j, r + 1) = 0.5 \times RP(i, j, r), r = 1, 2, \dots$ ;

(3) The percentage of the activity that must be reworked is used to measure the impact for each input on each activity [28], i.e., rework impact  $(RI(i, j)(i, j = 1, \dots, n))$  as shown in Figure 3b;

(4) It takes less effort to rework an activity compared to the first time [28]. Because activity participants will learn from the first experience and make adjustments. We model the learning curve for each activity, i.e.,  $LC_i$  (activity  $i$  takes a certain percentage of the original duration and resource, referred mainly to the experts' opinion).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1			0.1	0.2	0.3	0.2													
2	0.4		0.2		0.1	0.1	0.2												
3				0.2			0.1	0.3											
4			0.3				0.4	0.2						0.3	0.2				
5	0.6	0.3	0.5	0.5		0.4		0.1			0.1			0.4					
6		0.3	0.2	0.5	0.5		0.3	0.3		0.3	0.2			0.2					
7	0.2	0.5	0.5	0.2				0.2	0.2							0.1			
8			0.4	0.2	0.6		0.3		0.3	0.2									
9			0.2	0.3	0.3		0.4	0.2											
10		0.3	0.3	0.4	0.5	0.3	0.5	0.5				0.2				0.1	0.3		
11	0.4	0.5	0.2		0.6	0.3		0.3	0.3	0.5		0.2			0.4	0.1	0.2		
12	0.3	0.5	0.5	0.3	0.4	0.4		0.3	0.3		0.4		0.4	0.3	0.3	0.2	0.3		
13				0.2							0.3	0.4		0.1	0.1	0.1	0.3		
14				0.3							0.6	0.3	0.5		0.1	0.2	0.1		
15	0.5	0.3	0.2	0.5	0.4	0.5	0.6	0.3	0.3	0.2	0.5	0.6	0.3	0.5		0.2	0.2		
16	0.4	0.4	0.6	0.2	0.4	0.4	0.5	0.6	0.4	0.5	0.6	0.5	0.4	0.3	0.3		0.1		
17	0.3	0.6	0.5	0.3	0.6	0.5	0.4	0.5	0.3	0.6	0.4	0.5	0.3	0.2	0.4	0.3		0.2	
18	0.3	0.5		0.3	0.3	0.3	0.5	0.5	0.2		0.6	0.4	0.3	0.4	0.4	0.3	0.6		
19															0.4	0.5	0.2	0.4	

(a) RP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1			0.2	0.3	0.2	0.1													
2	0.3		0.3		0.4	0.3	0.1												
3				0.1			0.2	0.3											
4			0.7				0.4	0.5							0.6	0.3			
5	0.6	0.4	0.3	0.4		0.3		0.3				0.3			0.1				
6		0.2	0.3	0.2	0.1		0.6	0.5		0.3	0.2			0.4					
7	0.3	0.1	0.3	0.5				0.3	0.2								0.1		
8			0.3	0.2	0.6		0.7		0.3	0.4									
9			0.2	0.3	0.2		0.4	0.5											
10		0.3	0.4	0.3	0.5	0.6	0.7	0.5				0.3				0.5	0.3		
11	0.4	0.6	0.4		0.5	0.3		0.4	0.3	0.2		0.2			0.3	0.2	0.2		
12	0.1	0.2	0.2	0.2	0.3	0.3		0.6	0.5		0.4		0.2	0.1	0.4	0.5	0.2		
13				0.3							0.4	0.5		0.1	0.3	0.3	0.1		
14				0.1							0.3	0.2	0.7		0.1	0.1	0.2		
15	0.1	0.2	0.2	0.1	0.2	0.2	0.1	0.2	0.1	0.2	0.3	0.1	0.1	0.4		0.1	0.1		
16	0.1	0.2	0.2	0.3	0.4	0.2	0.2	0.2	0.2	0.1	0.3	0.1	0.6	0.4	0.5		0.2		
17	0.3	0.1	0.3	0.1	0.3	0.3	0.1	0.1	0.3	0.1	0.1	0.2	0.1	0.2	0.1	0.5		0.2	
18	0.2	0.6		0.4	0.3	0.2	0.4	0.3	0.3		0.1	0.1	0.3	0.1	0.1	0.1	0.4		
19															0.6	0.7	0.5	0.6	

(b) RI

Figure 3. RP & RI matrix.

Therefore, the expected duration of the  $r$ -th rework of activity  $i$  can be calculated according to Equation (1):

$$E(D_{Ri})_r = (D_i - \sum_{k=r-1} (D_{Ri})_k) \times RP(i, j, r) \times RI(i, j) \times LC_i \quad (1)$$

where  $D_i$  is the initial duration of activity  $i$ ,  $(D_{Ri})_k$  is the  $k$ -th rework duration of activity  $i$ ,  $E(D_{Ri})_r$  is the  $r$ -th expected rework duration of activity  $i$ .

### 3.1.2. Distributions of Duration and Resources

Many probability distribution functions are used to model uncertainties of activity duration, resource, and cost, such as Beta distribution [58], triangle distribution [28,30]. In order to simplify the model and improve code efficiency, this study simplifies the positive skewness to triangular distribution. Then the most optimistic value (MOV), the most likely value (MLV), and the most pessimistic value (MPV) are chosen to conduct the model, as shown in Figure 4. Besides, it is assumed that the resource and cost of the activities are linearly distributed over the duration to simplify subsequent modeling.

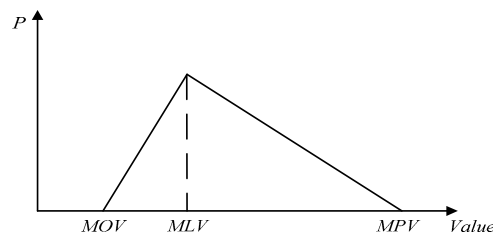


Figure 4. Skewed triangle distribution.

The Equation of the probability distribution function is:

$$\int_{MOV}^{MLV} \left( \frac{P(MLV)}{MLV - MOV} x - \frac{P(MLV) \times MOV}{MLV - MOV} \right) dx + \int_{MLV}^{MPV} \left( \frac{P(MLV)}{MLV - MPV} x - \frac{P(MLV) \times MPV}{MLV - MPV} \right) dx = 1 \quad (2)$$

The case is mainly referred to as the modular real estate development process from Eppinger's research [59]. Some supplement information is refined from experts' opinion. Parameters concerning duration, resource, and learning curve of 19 activities are detailed in Table 1. The resource may refer to personnel, material, and machinery. For simplification, here we use the unit resource for measurement. The order of activities in Table 1 is the original sequence in which project activities are executed. Information on upstream or downstream activities and on rework probabilities are reflected in the above RP matrix.

## 3.2. CCDSM Scheduling Mechanism

### 3.2.1. ICCDSM Scheduling Mechanism

Feedback above the diagonal indicates that activity will begin with assumed outputs of activities that have not yet been executed. This inevitably brings risks. Moreover, a mark's distance from the diagonal indicates the scope of the feedback. Long feedbacks in the upper right corner of the DSM probably cause a cascade of rework. Adjusting the activity sequence is effective in decreasing iterations and shorten the distance of information feedbacks. Therefore, following previous research [18,60], a genetic algorithm is used to reorder the activity sequence. Banding algorithm in the DSM framework is employed to identify independent elements in the DSM to improve concurrency and reduce the duration.

**Table 1.** Parameters of the 19 activities.

No.	Activity Name	Duration (Day)			Resource (Unit)			LC (%)
		MOV	MLV	MPV	MOV	MLV	MPV	
1	Perform preliminary market analysis	3	5	10	4.16	7.81	20.28	0.5
2	Evaluate market ability options	3	5	10	3.97	7.75	20.36	0.6
3	Engage feasibility consultants	2	3	6	3.84	5.30	12.49	0.3
4	Evaluate planning and zoning	10	13	20	13.33	20.32	41.69	0.5
5	Perform massing study	15	18	27	19.68	28.11	55.73	0.6
6	Develop conceptual design	30	35	50	40.15	53.00	100.3	0.8
7	Identify external stakeholders	5	8	15	7.35	13.34	30.16	0.4
8	Identify permits and approvals	10	13	20	13.63	20.17	41.92	0.7
9	Complete phase 1 ESA	10	13	20	13.73	19.67	40.17	0.3
10	Evaluate consultants and contractors	3	5	10	4.27	7.74	21.56	0.6
11	Obtain rough construction costs	3	5	10	4.97	8.16	21.33	0.5
12	Determine highest and best use	3	5	9	4.68	7.88	18.39	0.5
13	Identify debt options	2	3	5	2.87	4.63	10.25	0.4
14	Identify equity options	2	3	5	3.83	5.54	10.03	0.5
15	Update financial underwriting	3	5	10	3.95	8.42	21.58	0.5
16	Reevaluate organization strategy	1	2	4	2.51	4.40	8.88	0.6
17	Estimate schedule	3	4	7	4.34	7.04	14.69	0.6
18	Gain control of site and/or client	3	4	7	4.83	7.43	15.03	0.6
19	Review and approve	3	4	7	4.07	6.31	15.22	0.7

### Feedforward Optimization

Sequencing is a common method in process architecture DSM models to reorder the rows and columns to minimize iterations (the number of marks above the diagonal). In other words, re-sequencing activities involves arranging activities with as many interactions as possible below the diagonal. If we cannot move elements from the upper to the lower triangle, these elements will be arranged as close as possible to the diagonal. The sequencing of DSM belongs to the NP-hard problem, which can be solved with GA [53,61,62]. Therefore, a GA is employed to re-sequence activities in DSM with expected feedback length as objective function [18,59], that is, to minimize the sum of the product of upper triangle feedback distance and rework probability. The optimization objective function applied in the genetic algorithm is:

$$ED = \sum_{i=1}^N \sum_{j=i+1}^N (RP(i, j) \times (j - i)) \quad (3)$$



where  $RP(i, j)$  is the first rework probability of activity  $i$  caused by activity  $j$ .

#### Identification of Concurrent Activities and the Critical Chain

Banding algorithm is a graphical technique proposed by Grose [63] (1994) to identify potential sets of concurrent activities in a process. The banding algorithm ignores feedback elements in the upper triangle and binds one or more independent adjacent activities into “bands”. In the beginning, identify the first unfinished activity in the sequenced order. Next, search activities executed without outputs from identified activities (the first unfinished activity or other activities already in the band). Then, check if the resources available fill the requirements (activities identified in the first two steps). Only the first several activities within the resource constraints will be placed in the same band. Finally, repeat the above steps until all activities are placed in bands. Activities in the same band can be executed at the same time. In this framework, the critical chain can be identified. Equation (4) presents the mathematical expression:

$$CC = \{A_{L_{ki}} \mid \max(D_{A_i}), A_i \in L_k, k = 1, \dots, m, i = 1, 2, 3, \dots\} \quad (4)$$

where  $m$  is the number of bands in the DSM,  $L_k$  is the  $k$ -th band,  $\max(D_{A_i})$  is the longest duration of all activities in a band,  $A_{L_{ki}}$  is the activity with the longest duration of all activities in the  $k$ -th band.

Therefore, the length of the critical chain can be calculated with Equation (5):

$$T_{CC} = \sum CC = \sum_{k=1}^m D(A_{L_{ki}}) \quad (5)$$

where  $D(A_{L_{ki}})$  is the duration of the critical activity.

#### 3.2.2. Potential Criticality

In practice, almost all projects' structures are stochastic networks, which makes each activity have the possibility to become a critical activity in theory. Therefore, the critical chain identified by CPM or CCPM and used as critical activities is greatly suspicious for application. From a more detailed level—activity, this study extends the interface criticality to potential criticality. The importance (or impact area) of each activity is quantified from two aspects of duration and resource.

Interface criticality was first proposed by Browning and Eppinger [28] to measure the criticality in the interface from the perspective of information interaction. In fact, the definition of interface criticality by Browning and Eppinger [28] refers to the first rework time [18]. Although we may have a glance at the overlaps between activities, the potential impact of the activity is still not clear. Referring to Ma et al. [18], we define the potential criticality, i.e., the impact from upstream activities and the impact on downstream activities to measure the activity impact. Specifically, we place activity inputs in rows and activity outputs in columns. For each activity  $i \in [1, n]$ ,  $n \in N^*$ , we define the activity with input to activity  $i$  as the upstream activity of activity  $i$ , and the activity with input from activity  $i$  as the downstream activity of activity  $i$ . An example is shown in Figure 4, where activity 2 receives information from activity 1 and 3, so activity 1 and 3 are upstream activities of activity 2. Activity 2 also sends information to activity 3, so activity 3 is a downstream activity of activity 2 as well. The impact area of activity 2 is shown in Figure 5.

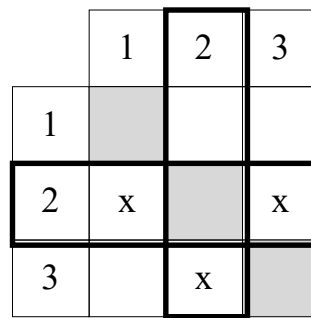


Figure 5. The impact area of activity 2.

From the perspective of project schedule, the impact of upstream activities on activity  $i$  is defined as:

$$Impact_{S_{j \rightarrow i}} = \sum_{j=1, j \neq i}^n RP(i, j) \times RI(i, j) \times LC_i \times D_i \tag{6}$$

The impact of activity  $i$  on downstream activities is defined as:

$$Impact_{S_{i \rightarrow k}} = \sum_{k=1, k \neq i}^n RP(k, i) \times RI(k, i) \times LC_k \times D_k \tag{7}$$

Therefore, the impact of activity  $i$  in the project, i.e., the potential criticality of the schedule, is defined as:

$$Impact_{S_i} = Impact_{S_{j \rightarrow i}} + Impact_{S_{i \rightarrow k}} \tag{8}$$

Similarly, the mathematical expression of the potential impact criticality of the resource can be defined as:

$$\begin{aligned} Impact_{Re_i} &= Impact_{Re_{j \rightarrow i}} + Impact_{Re_{i \rightarrow k}} \\ &= \sum_{j=1, j \neq i}^n RP(i, j) \times RI(i, j) \times LC_i \times R_i \\ &\quad + \sum_{k=1, k \neq i}^n RP(k, i) \times RI(k, i) \times LC_k \times R_k \end{aligned} \tag{9}$$

### 3.3. Buffer Calculation in CCDSM

#### 3.3.1. Rework Impact Area

Similar to the definition of potential criticality, rework impact refers to potential rework arising from information input of activity  $i$ 's downstream activities. Therefore, the rework impact area measures the duration of potential rework occurrence after finishing the activity  $i$  in the first time. The mathematical expression of the rework effect area is shown in Equation (10).

$$RIA_i = LFT_i - FFT_i \tag{10}$$

where  $FFT_i$  is the time point at which activity  $i$  is first affected by rework,  $LFT_i$  is the final completion time (including rework workload) of activity  $i$ .

In a sense, the rework impact area is independent of the initial activity, which will not be limited by the critical chain and other schedule parameters. Rework impact area also intuitively displays rework impact for each activity. Because of the dependency, the scheduling algorithm and the identification algorithm for the critical chain will not affect rework uncertainty. Feedforward identification of uncertain events will cause the NP-hard problem, and intelligent algorithms will

greatly lower efficiency. Therefore, the identification of the critical chain and the optimization of scheduling algorithms are based on the schedule of the first execution and do not involve rework.

### 3.3.2. Buffer Calculation

Feeding buffers are added at the end of a non-critical chain to ensure critical activities will not wait for any sub-critical activities. Therefore, the identification of critical chain and non-critical chain are necessary before setting feeding buffers. However, no matter if the activity is on the critical chain, rework may occur because assumed inputs have been modified with real outputs of other activities. In other words, the completion of any activity may cause rework, which makes traditional activity parameters (such as the earliest start time and the earliest end time) invalid. In this situation, the critical chain and the non-critical chain are difficult to differentiate. Then the calculation of feeding buffer is no longer the first thing due to rework uncertainty. Although this study has employed a banding algorithm to identify the critical chain, rework impact area with similar function and broader coverage is also introduced. Therefore, this chapter will focus mainly on project buffer instead of feeding buffer.

#### The Perspective of Completion Probability

As a part of project duration, project buffer occupies time, and resources. However, the project buffer also provides sufficient time to meet the project delivery standard with higher completion probability.

From the perspective of the completion probability, the project buffer is defined as:

$$PB_{completepro} = D_{x\%} - D_{MLV} \quad (11)$$

where  $D_{x\%}$  is the duration with  $x\%$  completion probability,  $D_{MLV}$  is the duration simulated with each activity's most likely the duration.

Project buffer may be consumed over the project life cycle; therefore, to estimate potential resource consumption during the project buffer and for calculation convenience, a traditional prediction technique, simple average (i.e., average resource consumption in the project life cycle), is used to allocate resource for project buffer.

$$Re_{PB_t} = \frac{1}{dur} \sum_{i=1}^{dur} Re_i \quad (12)$$

where  $t$  is the start time when project buffer will be consumed (In the simulation model,  $t$  is commonly referred to as the end time without the project buffer),  $Re_{PB_t}$  is the daily resource consumption during the project buffer,  $dur$  is the project duration without the project buffer,  $Re_i$  is the resource consumption in the  $i$ -th day.

Project buffer defined in the perspective of completion probability conforms to the assumption of "sacrificing" a certain period in exchange for a higher completion probability. The results are purely rational, posteriori, and simulation-based; therefore, the project buffer size is dependent ofn the project schedule and scheduling algorithm, making it easy for practical applications.

#### The Perspective of Potential Criticality

The adaptive procedure with density (APD) is a method to calculate buffer sizes [36]. The most important assumption of the APD method is that as the number of precedence relationships increases, it is more likely that delays will occur, and therefore, the buffer size should also increase. In this study, the higher an activity's potential criticality or reworks impact, the more delays will occur. A certain correlation exists between the potential criticality of duration and rework impact because of similar definition forms. In order not to reuse the indicators, project buffer size is calculated based on the potential criticality of duration.

Inspired by the APD method, the project buffer size is calculated with Equations (13) and (14).

$$K_i = 1 + Impact_{S_i} / \sum_{i=1}^N Impact_{S_i} \tag{13}$$

$$PB = \sqrt{\sum_{i=1}^N K_i^2 \times \sigma_i^2} \tag{14}$$

where  $Impact_{S_i}$  is the potential duration criticality of the activity  $i$ ,  $\sigma_i^2$  is the duration variance of activity  $i$ ,  $N$  is the number of activities.

### 4. Simulation and Results

#### 4.1. Parameters and Steps

According to the above discussion, initial data includes RP matrix, RI matrix, skewed triangular probability distribution parameters of duration and resource, and learning curve parameters. Several variables are introduced for simulation implementation, as shown in Table 2. The steps of the ICCDSM simulation algorithm are shown in Table 3.

**Table 2.** Intermediate variables for simulation.

Name	Definition
$r$	Number of simulation iterations
$r_{max}$	Maximum number of simulation iteration
$Re$	Sum of resource for the entire project
$S$	Duration for the entire project
$t$	Current time
$W_n$	A unit vector with $n$ elements initialized by 1, indicating the remaining workload. $W_n \in [0, 1]$
$WN_n$	A Boolean vector (A value of 1 stands for activities being executed, and 0 for finished activities and activities that still have to be started)

**Table 3.** Steps of ICCDSM simulation algorithm.

Step	Content
1	Initialize parameters when $t = 0$ ;
2	Re-sequence activities in the DSM;
3	Sample the duration and resource for each activity randomly.
4	For the current state: 4.1 Determine which activity is firstly executed: search the upstream activity $i$ ( $W(i) > 0, WN(i) = 1$ ); 4.2 Find concurrent activities: banding algorithm considering resource constraints (daily available resource is 3.2 unit) is employed to find the activity $k$ remaining to be done and dependent of the unfinished upstream activities, then $WN(k) = 1, k \in (i, n]$ ; 4.3 Advance time and calculate $S_t$ and $Re_t$ until the completion of the activities identified in step 4.1 and 4.2; 4.4 Calculate rework: firstly, search potential rework in column $j$ , if random number $P_{rew} < RP(i, j)$ ( $P_{rew}$ is subject to uniform distribution in $[0,1]$ ), rework will occur and $W(i) = \min((W(i) + RI(i, j) \times LC_i), 1)$ . Resource requirements will also be linearly adjusted according to work that need to be done as well as activity duration. If $P_{rew} > RP(i, j)$ , no rework will occur. Similarly, the search for potential rework in column $i$ .
5	The process advances $t$ , in which step 4 will be executed repeatedly until the completion of all activities.
6	Output $S$ and $Re$ ;
7	$r$ plus 1. Simulation won't end until $r = r_{max}$ .

This algorithm is programmed in MATLAB R2018a. When dealing with large amounts of data, the concurrent computing of MATLAB is more efficient than traditional programming platforms for DSM, such as VBA (Visual Basic for Applications).

#### 4.2. Results

In this section, the proposed method is validated by an example with the following steps.

Step 1: Input the above data into the ICCDSM model.

Step 2: Re-sequence the activities with genetic algorithm. In the genetic algorithm settings, the initial population size is 100; the number of generations is 200; the crossover probability is 0.95; the mutation probability is 0.08. The new sequence is as follows:

[3 1 2 7 4 8 9 5 6 10 11 15 12 13 16 14 17 18 19]

Compared to the un-optimized sequence, this result shows improvements in some indicators, as detailed in Table 4.

**Table 4.** Comparison of different sequences.

	Number of Marks above the Diagonal	Total Distance	Expected Distance (Objective Function)
Un-optimized sequence	52	193	42
Reordered sequence	49	166	36.8

Step 3: Identify the critical chain with the banding algorithm. The critical chain based on MLV and resource constraints is as follows:

[3 1 2 7 4 8 5 6 10 11 15 12 13 16 14 17 18 19]

In fact, only activity 9 and 5 are in the same band, indicating the two activities can be executed concurrently. This concurrency will reduce project duration. On the one hand, many activities can not begin with other activities because the case project is strongly coupled. On the other hand, resource constraints, to some degree, also reduce the concurrence.

Step 4: Potential criticality of duration and resource can be calculated by Equations (8) and (9), as seen in Figures 6 and 7.

The two images indicate that the potential criticality of duration and resources of activities 5, 6, and 8 are higher than others. This indicates that the three activities not only are more likely to cause rework for other activities but also have a high probability of rework caused by other activities. So, the three activities require more attention.

Step 5: Simulate the duration and resource consumption. The results of 1000 Monte Carlo simulations show that 50%, 80%, and 90% expected duration are 426 days, 480 days and 509 days respectively, and 50%, 80% and 90% expected resource consumption are 773 units, 882 units, and 956 units respectively.

Step 6: Calculate MLV-based rework impact area.

According to Equation (10), we can calculate the rework impact area for each activity. Here we take the calculation of the rework impact area for activity 3 as an example. Activity 3 was first completed in the third day. After that, other activities were executed with optimized sequence. However, activity 3 would be reworked for the first-time on the 48-th day (calculated by stochastic rework network). Until the 316-st day, all workload of activity 3 was completed. Therefore, the rework impact area is as follows:

$$RIA_{New\_seq} = [268 \ 221 \ 344 \ 319 \ 350 \ 350 \ 320 \ 226 \ 312 \ 155 \ 261 \ 251 \ 209 \ 84 \ 177 \ 5 \ 130 \ 5 \ 0]$$

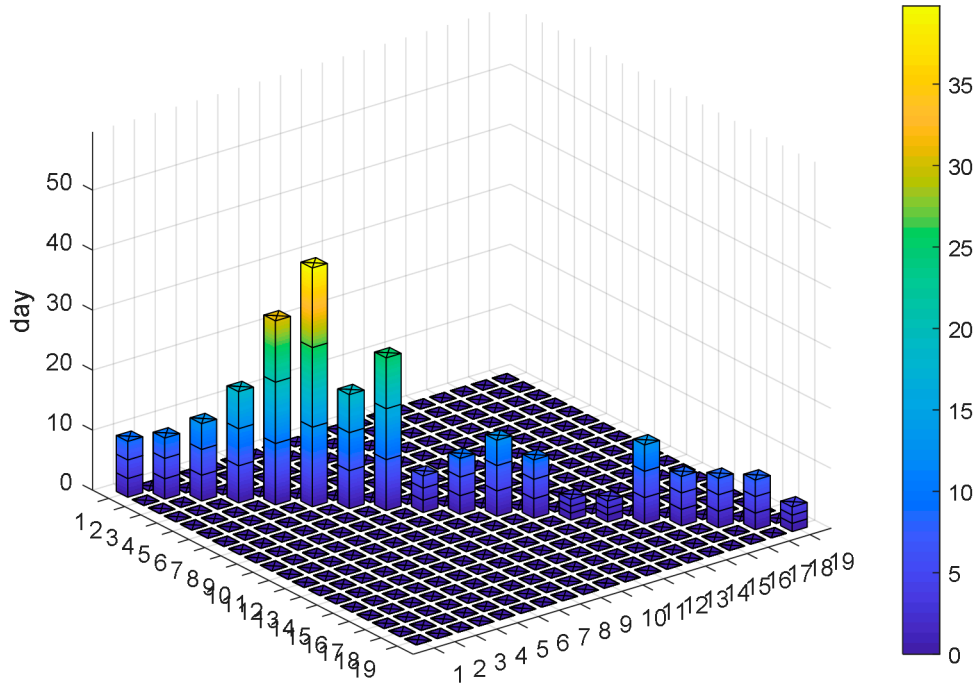


Figure 6. Potential criticality of duration.

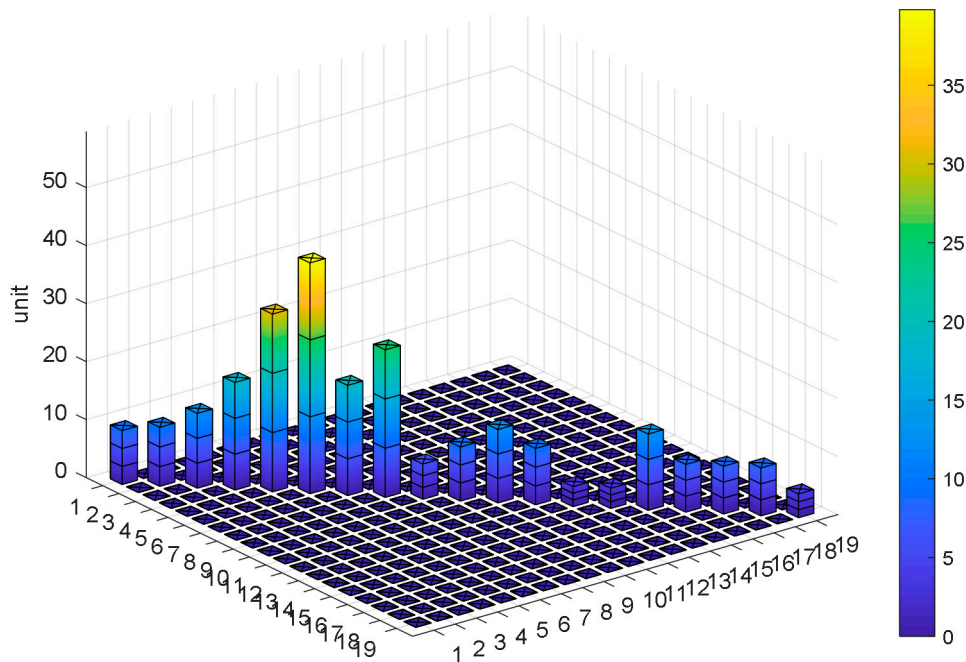


Figure 7. Potential criticality of resource.

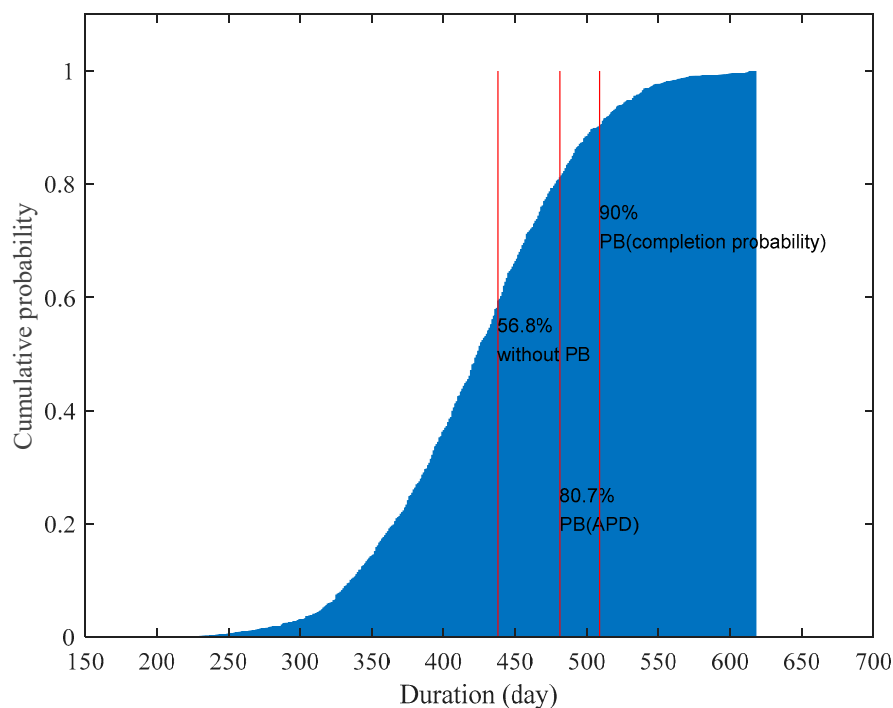
The results indicate that almost all of the rework impact area is very long. The main reason is a strong coupling between activities. Therefore, the rework impact area also reflects the complexity of a project.

Step 7: Calculate the project buffer and completion probability.

Two types of project buffer sizes are 72 days and 43 days, respectively, calculated by Equations (11)–(14). To be specific, the MLV based project duration is 438 days. The completion time with 90% probability is 510 days. Therefore, from the perspective of completion probability, the project

buffer size is 72 days. In step 5, the potential duration criticality of each activity has been calculated. Then the  $K_i$  defined in the third section can be obtained. Next, the duration variance of 1000 simulations is recorded. Consequently, from the perspective of potential criticality, the project buffer size is 43 days.

Figure 8 depicts the completion probability with the method proposed in this study. Even if we don't consider the project buffer, the completion probability is still 56.8%, indicating that incorporating rework into the project execution process may solve many uncertainties. Two types of project buffer will increase the completion probability to 80.7% and 90%, respectively. A comparison to previous research is detailed in Figure 9. The most likely duration calculated by CCPM and CCDSM is 164 days and 170 days, respectively. However, it is impossible to complete the project with such duration. Moreover, we also calculate the duration with no consideration on second rework and resource constraints, respectively. The completion probability is only 0.1%, almost equaling to zero, if we pay no attention to the impact of the second rework. Besides, ignoring the impact of resource constraints results in a more optimistic completion probability (an increase of 27% approximately). To sum, the ICCDSM method improves the reliability and robustness of the project schedule.



**Figure 8.** Completion probability of the proposed method.

Step 8: Develop the schedule and resource consumption.

The resource consumption in project buffer adopts the life cycle average resource occupancy. Here we develop a schedule and resource consumption based on MLV and completion probability, as shown in Figure 10.

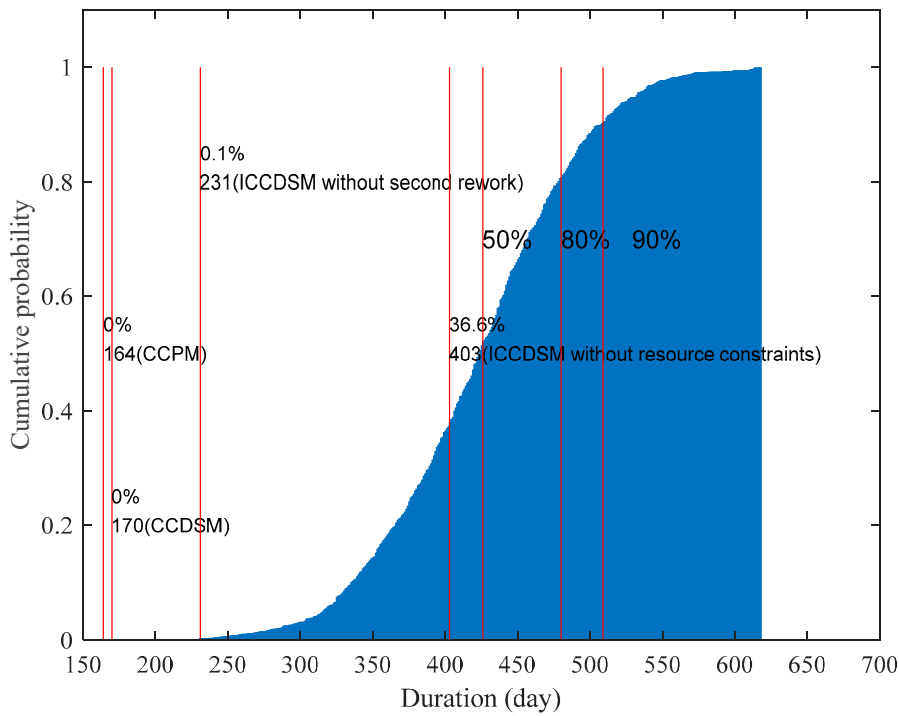


Figure 9. Results comparison with previous research.

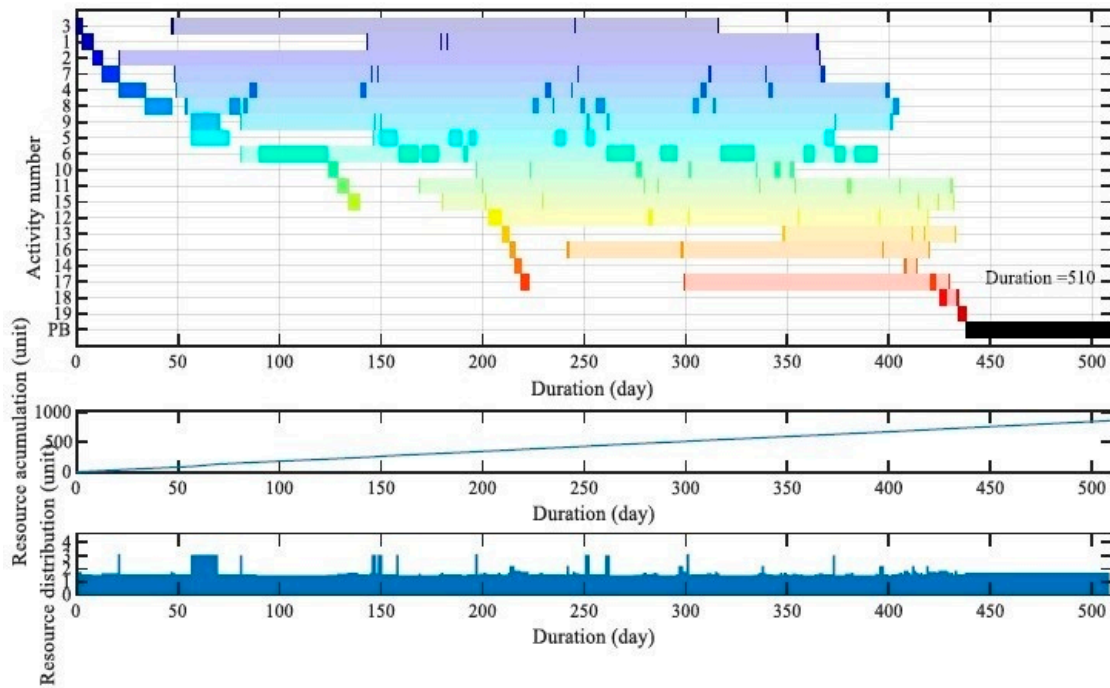


Figure 10. The schedule and resource consumption based on MLV and completion probability. Notes: The color presented in this Figure is to represent different activity processes.

5. Conclusions

This study develops an ICCDSM model for project schedule development and scheduling mechanism under rework risks. Firstly, we make assumptions on rework uncertainties. To deal with large-scale rework, the corresponding mechanism is established, in which rework probability is decreased by 50% instead of remaining as a constant in the previous DSM framework. Secondly, DSM is combined with CCPM without the help of max-plus algebra. ICCDSM retains buffer and



critical chain from CCPM. A genetic algorithm is employed to re-sequence activities, and a banding algorithm is used to identify concurrent activities. Thirdly, potential criticality is proposed to measure the importance of the activity and its impact area, which provides the basis for project controlling. Then, the rework impact area measures potential rework duration. From the perspectives of completion probability and potential criticality of duration, two types of project buffer are calculated. At last, we apply the ICCDSM algorithm in an example. The results indicate that the proposed ICCDSM model can help make and visualize a stochastic network schedule under rework risks and visualize the schedule. On this basis, the project schedule can be optimized from the perspective of completion probability or minimum duration.

This study contributes to the literature on the project rework management, the literature on project schedule management. First, we incorporate rework into the project dynamic execution process. Although previous research [18] tried to quantify rework (including second rework), they adopted a static view to calculate rework. In fact, the rework may happen as long as the activity receives information from upstream activities. From the dynamic perspective, the rework impact area is proposed and visualized to indicate potential rework windows, which provides guidance for project managers to perform project control in time. Second, resource constraints are the focus of the CCPM, which was ignored in the previous CCDSM framework. The consideration of resource constraints in this research improves the practicality of the proposed framework. Third, from the perspective of information flow, potential criticality measures the importance of each activity, which is a significant supplement to the critical activity. Specifically, potential criticality implies the impact of upstream activities and the impact on downstream activities. The higher the potential criticality of activity, the more attention it should receive. Finally, the schedule developed by the proposed method has a higher completion probability than the previous methods, offering a novel framework for schedule management.

There exist some shortcomings in this study. The work presented in this paper is a methodology work rather than empirical work. Although the method was tested using an example project from the literature under some assumptions to show its utility, more extensive applications in different project settings need to be undertaken to demonstrate the applicability and validity of the method further. Moreover, as a new schedule management tool, the proposed ICCDSM model focuses more on designing the underlying structure. Further research on algorithm optimization and multi-object optimization should be conducted in the future. What's more, the proposed method is a supplement instead of an alternative to other schedule management tools. Besides, the authors pay less attention to resource constraints (single resource instead of multi resources), which reduces the practicability of this method.

**Author Contributions:** Conceptualization—G.M. and T.Z.; Methodology—T.Z. and J.J.; Formal Analysis—T.Z. and J.J.; Writing—Original Draft Preparation—J.J. and T.Z.; Writing—Review and Editing—J.J. and S.J.; Supervision—G.M.; Funding Acquisition—G.M.

**Funding:** This research was funded by the National Natural Science Foundation of China (grant number 71671128).

**Acknowledgments:** The authors wish to acknowledge the support from the responders of the questionnaires.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Love, P.E.D. Influence of project type and procurement method on rework costs in building construction projects. *J. Constr. Eng. Manag.* **2002**, *128*, 18–29. [[CrossRef](#)]
2. Love, P.E.D.; Li, H. Overcoming the problems associated with quality certification. *Constr. Manag. Econ.* **2000**, *18*, 139–149. [[CrossRef](#)]
3. Hwang, B.G.; Thomas, S.R.; Haas, C.T.; Caldas, C.H. Measuring the impact of rework on construction cost performance. *J. Constr. Eng. Manag.* **2009**, *135*, 187–198. [[CrossRef](#)]

4. Hwang, B.G.; Zhao, X.; Goh, K.J. Investigating the client-related rework in building projects: The case of Singapore. *Int. J. Proj. Manag.* **2014**, *32*, 698–708. [[CrossRef](#)]
5. Schonberger, R.J. Why projects are 'always' late: A rationale based on manual simulation of a PERT/CPM network. *Interfaces* **1981**, *11*, 66–70. [[CrossRef](#)]
6. Long, L.D.; Ohsato, A. Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. *Int. J. Proj. Manag.* **2008**, *26*, 688–698. [[CrossRef](#)]
7. Ma, G.; Gu, L.; Li, N. Scenario-based proactive robust optimization for critical-chain project scheduling. *J. Constr. Eng. Manag.* **2015**, *141*, 04015030. [[CrossRef](#)]
8. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*; MIT Press: Cambridge, MA, USA, 2012.
9. Eppinger, S.D.; Whitney, D.E.; Smith, R.P.; Gebala, D.A. A model-based method for organizing tasks in product development. *Res. Eng. Des.* **1994**, *6*, 1–13. [[CrossRef](#)]
10. Browning, T.R. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Trans. Eng. Manag.* **2001**, *48*, 292–306. [[CrossRef](#)]
11. Browning, T.R. Design structure matrix extensions and innovations: A survey and new opportunities. *IEEE Trans. Eng. Manag.* **2016**, *63*, 27–52. [[CrossRef](#)]
12. Ma, G.; Wang, A.; Li, N.; Ai, Q. Improved critical chain project management framework for scheduling construction projects. *J. Constr. Eng. Manag.* **2014**, *140*, 04014055. [[CrossRef](#)]
13. Hall, N.G. Project management: Recent developments and research opportunities. *J. Syst. Sci. Syst. Eng.* **2012**, *21*, 129–143. [[CrossRef](#)]
14. Goldratt, E.M. *Critical Chain: A Business Novel*; North River Press: Great Barrington, MA, USA, 2017.
15. Ikeziri, L.M.; de Souza, F.B.; Gupta, M.C.; de Camargo Fiorini, P. Theory of constraints: Review and bibliometric analysis. *Int. J. Prod. Res.* **2019**, *57*, 5068–5102. [[CrossRef](#)]
16. PMI. *PMI's Pulse of the Profession: Capturing the Value of Project Management*; PMI: Newtown Square, PA, USA, 2015.
17. Bie, L.; Cui, N.; Zhang, X. Buffer sizing approach with dependence assumption between activities in critical chain scheduling. *Int. J. Prod. Res.* **2012**, *50*, 7343–7356. [[CrossRef](#)]
18. Ma, G.; Hao, K.; Xiao, Y.; Zhu, T. Critical Chain Design Structure Matrix Method for Construction Project Scheduling under Rework Scenarios. *Math. Probl. Eng.* **2019**, *2019*, 1595628. [[CrossRef](#)]
19. Love, P.E.D.; Edwards, D.J.; Watson, H.; Davis, P. Rework in civil infrastructure projects: Determination of cost predictors. *J. Constr. Eng. Manag.* **2010**, *136*, 275–282. [[CrossRef](#)]
20. Ye, G.; Jin, Z.; Xia, B.; Skitmore, M. Analyzing causes for reworks in construction projects in china. *J. Manag. Eng.* **2015**, *31*, 04014097. [[CrossRef](#)]
21. Arashpour, M.; Wakefield, R.; Blismas, N.; Lee, E.W.M. Analysis of disruptions caused by construction field rework on productivity in residential projects. *J. Constr. Eng. Manag.* **2014**, *140*, 04013053. [[CrossRef](#)]
22. Li, Y.; Taylor, T.R.B. Modeling the impact of design rework on transportation infrastructure construction project performance. *J. Constr. Eng. Manag.* **2014**, *140*, 04014044. [[CrossRef](#)]
23. Oyewobi, L.O.; Falemu, O.A.; Ibironke, O.T. The impact of rework and organizational culture on project delivery. *J. Eng. Des. Technol.* **2016**, *14*, 214–237.
24. Love, P.E.D.; Smith, J. Benchmarking, benchaction, and benchlearning: Rework mitigation in projects. *J. Manag. Eng.* **2003**, *19*, 147–159. [[CrossRef](#)]
25. Love, P.E.D.; Irani, Z.; Edwards, D.J. A rework reduction model for construction projects. *IEEE Trans. Eng. Manag.* **2004**, *51*, 426–440. [[CrossRef](#)]
26. Love, P.E.D.; Edwards, D.J.; Irani, Z.; Goh, Y.M. Dynamics of rework in complex offshore hydrocarbon projects. *J. Constr. Eng. Manag.* **2011**, *137*, 1060–1070. [[CrossRef](#)]
27. Browning, T.R. Sources of schedule risk in complex system development. *Syst. Eng.* **1999**, *2*, 129–142. [[CrossRef](#)]
28. Browning, T.R.; Eppinger, S.D. Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Trans. Eng. Manag.* **2002**, *49*, 428–442. [[CrossRef](#)]
29. Cho, S.H.; Eppinger, S. Product development process modeling using advanced simulation. In Proceedings of the DETC'01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, PA, USA, 9–12 September 2001.

30. Cho, S.H.; Eppinger, S.D. A simulation-based process model for managing complex design projects. *IEEE Trans. Eng. Manag.* **2005**, *52*, 316–328. [[CrossRef](#)]
31. Moradi, H.; Shadrokh, S. A robust scheduling for the multi-mode project scheduling problem with a given deadline under uncertainty of activity duration. *Int. J. Prod. Res.* **2019**, *57*, 3138–3167. [[CrossRef](#)]
32. Tian, M.; Liu, R.J.; Zhang, G.J. Solving the resource constrained multi-project scheduling problem with an improved critical chain method. *J. Oper. Res. Soc.* **2019**. [[CrossRef](#)]
33. Zhang, J.; Song, X.; Díaz, E. Critical chain project buffer sizing based on resource constraints. *Int. J. Prod. Res.* **2017**, *55*, 671–683. [[CrossRef](#)]
34. Herroelen, W.; Leus, R. On the merits and pitfalls of critical chain scheduling. *J. Oper. Manag.* **2001**, *19*, 559–577. [[CrossRef](#)]
35. Ashtiani, B.; Jalali, G.R.; Aryanezhad, M.B.; Makui, A. A new approach for buffer sizing in critical chain scheduling. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management 2007, Singapore, 2–4 December 2007; pp. 1037–1041.
36. Tukul, O.I.; Rom, W.O.; Eksioğlu, S.D. An investigation of buffer sizing techniques in critical chain scheduling. *Eur. J. Oper. Res.* **2006**, *172*, 401–416. [[CrossRef](#)]
37. Leach, L.P. *Critical Chain Project Management*; Artech House: Norwood, MA, USA, 2014.
38. Newbold, R.C. *Project Management in the Fast Lane: Applying the Theory of Constraints*; CRC Press: Boca Raton, FL, USA, 1998.
39. Arashpour, M.; Kamat, V.; Bai, Y.; Wakefield, R.; Abbasi, B. Optimization modeling of multi-skilled resources in prefabrication: Theorizing cost analysis of process integration in off-site construction. *Autom. Constr.* **2018**, *95*, 1–9. [[CrossRef](#)]
40. Arashpour, M.; Heidarpour, A.; Nezhad, A.A.; Hosseini, Z.; Chileshe, N.; Hosseini, R. Performance-based control of variability and tolerance in off-site manufacture and assembly: Optimization of penalty on poor production quality. *Constr. Manag. Econ.* **2019**. Forthcoming. [[CrossRef](#)]
41. Chu, C.C. Buffer sizing and critical chain project management. *Comput. Integr. Manuf. Syst.* **2008**, *14*, 1029–1035.
42. Zhao, Z.Y.; You, W.Y.; Zuo, J. Application of innovative critical chain method for project planning and control under resource constraints and uncertainty. *J. Constr. Eng. Manag.* **2010**, *136*, 1056–1060. [[CrossRef](#)]
43. Shi, Q.; Wang, Y.T.; Gong, T. An improved approach for project buffer sizing and evaluation. *Syst. Eng. Theory Pract.* **2012**, *32*, 1739–1746.
44. Zhang, J.; Song, X.; Chen, H.; Shi, R. Optimisation of critical chain sequencing based on activities' information flow interactions. *Int. J. Prod. Res.* **2015**, *53*, 6231–6241. [[CrossRef](#)]
45. Zhang, J.; Shi, R.; Diaz, E. Dynamic monitoring and control of software project effort based on an effort buffer. *J. Oper. Res. Soc.* **2015**, *66*, 1555–1565. [[CrossRef](#)]
46. Zhang, J.; Song, X.; Díaz, E. Project buffer sizing of a critical chain based on comprehensive resource tightness. *Eur. J. Oper. Res.* **2016**, *248*, 174–182. [[CrossRef](#)]
47. Tarjan, R. Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1972**, *1*, 146–160. [[CrossRef](#)]
48. English, K.; Bloebaum, C.L.; Miller, E. Development of multiple cycle coupling suspension in the optimization of complex systems. *Struct. Multidiscip. Optim.* **2001**, *22*, 268–283. [[CrossRef](#)]
49. David, M. Organising, valuing and improving the engineering design process. *J. Eng. Des.* **2013**, *24*, 524–545. [[CrossRef](#)]
50. Aleisa, E.; Lin, L. Sequencing of design tasks based on the degree of permissible concurrency. *Int. J. Appl. Manag. Sci.* **2009**, *1*, 367–387. [[CrossRef](#)]
51. Xu, C.; Li, A.; Liu, X. Tearing method of coupled activity set within concurrent design process. *J. Tongji Univ.* **2010**, *38*, 427–431.
52. Chen, S.J.G.; Huang, E. A systematic approach for supply chain improvement using design structure matrix. *J. Intell. Manuf.* **2007**, *18*, 285–299. [[CrossRef](#)]
53. Meier, C.; Yassine, A.A.; Browning, T.R. Design process sequencing with competent genetic algorithms. *J. Mech. Des.* **2007**, *129*, 566–585. [[CrossRef](#)]
54. Smith, R.P.; Eppinger, S.D. A predictive model of sequential iteration in engineering design. *Manag. Sci.* **1997**, *43*, 1104–1120. [[CrossRef](#)]
55. Yassine, A.A. Investigating product development process reliability and robustness using simulation. *J. Eng. Des.* **2007**, *18*, 545–561. [[CrossRef](#)]

56. Meier, C. Time-Cost Tradeoffs in Product Development Processes. Ph.D. Thesis, Technische Universität München, Munich, Germany, 2011.
57. Maheswari, J.U.; Varghese, K. Project Scheduling using Dependency Structure Matrix. *Int. J. Proj. Manag.* **2005**, *23*, 223–230. [[CrossRef](#)]
58. Keefer, D.L.; Verdini, W.A. Better estimation of PERT activity time parameters. *Manag. Sci.* **1993**, *39*, 1086–1091. [[CrossRef](#)]
59. Eppinger, S.D.; Bonelli, S.; Gonzalez, A.M. Managing Iterations in the Modular Real Estate Development Process. In Proceedings of the 15th International Dependency and Structure Modelling Conference, Melbourne, Australia, 29–30 August 2013; pp. 37–44.
60. Gebala, D.A.; Eppinger, S.D. Methods for analyzing design procedures. *Proc. AMSE Third Int. Conf. Design Theory Methodol.* **1991**, *31*, 227–233.
61. Sheng, H.T.; Wei, F.J. The research and application of genetic-based design structure matrix optimization algorithm. *Chin. J. Manag. Sci.* **2007**, *15*, 98–103.
62. Yang, Q.; Lv, J. Project optimization and simulation based on DSM rework risk evaluation matrix. *Syst. Eng. Theory Pract.* **2010**, *30*, 1065–1071.
63. Grose, D.L. Reengineering the Aircraft Design Process. In Proceedings of the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, Panama, 7–9 September 1994; pp. 7–9.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).