

Article

An Exploratory Study on Optimal Iterative Design Schedules with the Consideration of Design Quality and Resource Constraints

Sou-Sen Leu *, Theresia Daisy Nattali Suparman and Cathy Chang-Wei Hung

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; daisy7nattali@yahoo.com (T.D.N.S.); hungcathy@mail.ntust.edu.tw (C.C.-W.H.)

* Correspondence: leuss@mail.ntust.edu.tw

Abstract: The classical dependency structure matrix (DSM) can effectively deal with iterative schedules that are highly coupled and interdependent, such as the design process and the concurrent process. Classical DSM generally follows the assumption that the least iteration occurs to achieve the shortest completion time. Nevertheless, the assumption may not hold because tasks ought to be re-visited several times if the design qualities do not meet the requirements. This research proposed a novel iterative scheduling model that combines the classical DSM concept with quality equations. The quality equations were used to determine the number of tasks that ought to be re-visited for fulfilling quality requirements during the iterative design process. Moreover, resources for concurrent activities are generally limited in the real world. Resource allocation should be incorporated in scheduling to avoid the waste and shortage of resources on a design project. This research proposed a new iterative scheduling model based on the classical DSM to optimize the iterative activities' structure in terms of minimizing completion time with the consideration of design quality under resource constraints. A practical design schedule was introduced to demonstrate the applicability of the proposed DSM algorithm.

Keywords: design structure matrix; iterative schedule; quality equation; resource constraint; genetic algorithm

Citation: Leu, S.-S.; Suparman, T.D.N.; Hung, C.C.-W. An Exploratory Study on Optimal Iterative Design Schedules with the Consideration of Design Quality and Resource Constraints. *Sustainability* **2021**, *13*, 4584. <https://doi.org/10.3390/su13084584>

Academic Editor:
Edmundas Kazimieras Zavadskas

Received: 10 March 2021

Accepted: 12 April 2021

Published: 20 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

1. Introduction

The design process is recognized as a complex system. Since there are several interdependent and highly coupled activities during the process, iterative schedules occur. Over the years, several scheduling methods have been developed in the industries. Gantt Charts is a graphical way of showing task durations and task schedule. Network techniques (like PERT and CPM) represent a project (set of activities) as a network using the graph theory. These scheduling algorithms fail to deal with the iterative schedules. To overcome the limitations of the traditional scheduling algorithms, the dependency structure matrix (DSM) is proposed to deal with iterative tasks. Based on the concept of DSM, research has been developed to manage iterative process to reduce completion time by rearranging the sequence of the activities (known as sequencing). Previous research applied fewer feedback dependencies concepts in sequencing to achieve minimum completion time. The concept mainly focuses on the number of feedback dependencies, since it is believed that possibility that the tasks have to be iterated is proportional to the number of feedback dependencies. The fewer feedback dependencies, the smaller the possibility that some tasks will be reworked, resulting in a shorter completion time.

Classical DSM generally follows the assumption that the least iteration occurs to achieve the shortest completion time. Nevertheless, the assumption may not hold because tasks ought to be re-visited several times if the design qualities do not meet the

requirements. In coupled tasks, this may result in several design iterations. Moreover, the other assumption underlying classical DSM is that resources are unlimited. Given the constraints of resource availability, the scheduler must take account of the trade-off between available resources and activity durations from time to time. This calls for resource-constrained scheduling. In this study, quality equations are employed in the classical DSM model to determine the number of tasks that ought to be re-visited. The trade-off between the number of feedback dependencies and the fulfillment of the design requirements under resource constraints could offer a better solution to the practical implementation.

This paper is organized as follows. First, literature studies of iterative scheduling algorithms and their limitations are reviewed. Second, the DSM, mainly activity-based DSM, is presented. Next, a new DSM algorithm combining the quality equations and resource constraints is presented. Finally, a building design process at an engineering consultant company is analyzed as a real case study to demonstrate the application of the new method.

2. Literature Review

Iterative scheduling may occur during the design process and the concurrent projects. DSM is a useful tool to perform both the analysis and the management of iterative schedules. DSM was first proposed by Steward to denote a generic matrix-based model for project information flow analysis [1,2]. A primary goal in basic DSM analysis is to minimize the number of feedbacks and their scope by restructuring or re-architecting the process. Since then, several DSM software systems have been proposed for practical implementation, such as DeMaid/GA and ADePT [3,4]. However, most classical DSM-based models do not: (1) obtain an optimum activity sequence; (2) tolerate uncertainty of activity time and cost; (3) consider resource constraints; or (4) consider the number of task iterations (i.e., the number of tasks to be re-visited during the design process).

Some research has been conducted to overcome some limitations of classical DSM. Smith and Eppinger [5] refined the DSM concept to include the stochastic nature of task iteration by adding probabilities to the off-diagonal entries, and deterministic task completion times to the diagonal entries of the matrix. Hisham and Bao [6] presented a simulation-based optimization framework that determines the optimal sequence of activities execution within a product development project to minimize the project's total iterative time given stochastic activity durations. Hejducki and Rogalska [7] proposed a time coupling method to determine the optimal schedule under resource constraints. Although the research mentioned above overcome some limitations of classical DSM, the number of tasks that need revisiting based on feedback dependency is generally assumed to be fixed or ignored. Additionally, the issue of resource constraints is generally not included in these models.

This research proposes a new concept, quality equations, to solve the problem. Quality equations are proposed to determine the iteration number for each task along with the design process. Furthermore, physical resource constraints are also considered in the model to determine the optimal iterative project schedules. These considerations are generally essential to the practical concurrent process.

3. Methodologies


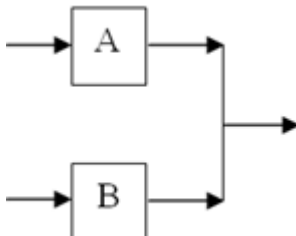
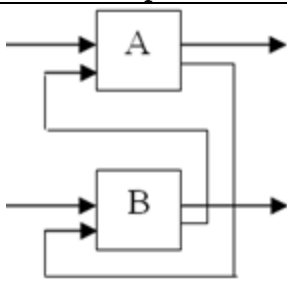

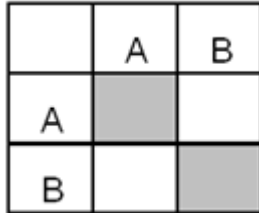

To address the aforementioned objectives, a modified DSM is proposed to analyze iterative project schedules by considering the number of task iterations and resource availability. DSM is first used to highlight the overall relationships among coupled activities. Quality equations are then applied to capture the logic of the number of the tasks that ought to be re-visited based on the design qualities of the tasks. The equations are used to determine the iteration number for each task based on the design quality and its improvement. Finally, a genetic algorithm (GA) is used to explore the optimum solution under resource constraints.

3.1. Design Structure Matrix (DSM)

DSM is a matrix which represents forward and backward relationships of activities in an iterative project. The basic representation of the DSM is a square matrix with each row and column indicating one activity of a project [1,8–10]. There are four types of DSM: (1) component-based or architecture DSM; (2) team-based or organization DSM; (3) activity-based or schedule DSM; and (4) parameter-based DSM. The first two applications are classified as static DSM, and the other two are known as time-based DSM. Static DSM is usually analyzed with clustering algorithms. Time-based DSM uses sequencing algorithms instead. Our study is one kind of activity-based DSM.

Activity-based DSM is used to describe the input/output relationship between activities and highlights iteration and coupled activities during the design process. There are three basic building blocks for describing the relationship between system activities: parallel (or concurrent), sequential (or dependent), and coupled (or interdependent), as depicted in Table 1. Figure 1 depicts an example of the basic DSM for seven activities. If there is no dependency, nothing (a zero) is recorded in the entry. The activities' relationships have to be read along the columns as “provide information to” and along the rows as “need information from”. The marks in the sub-diagonal matrix show feedforward information, while in those in the super-diagonal matrix indicate feedback information, the potential for iteration, and rework in the process. Based on the concept of sequencing, the arrangement of the rows and columns in DSM can be re-arranged without losing structural information. For example, the orderings of activities in Figure 1a can be randomly rearranged while keeping the same dependencies. Figure 1b shows the rearranged DSM. As shown in Figure 1b, the ordering of activities is rearranged to hopefully minimize the number of feedbacks.

Table 1. Relationships between activities in DSM.

Relationship	Sequential	Parallel	Coupled
Graph Representation			
DSM Representation			

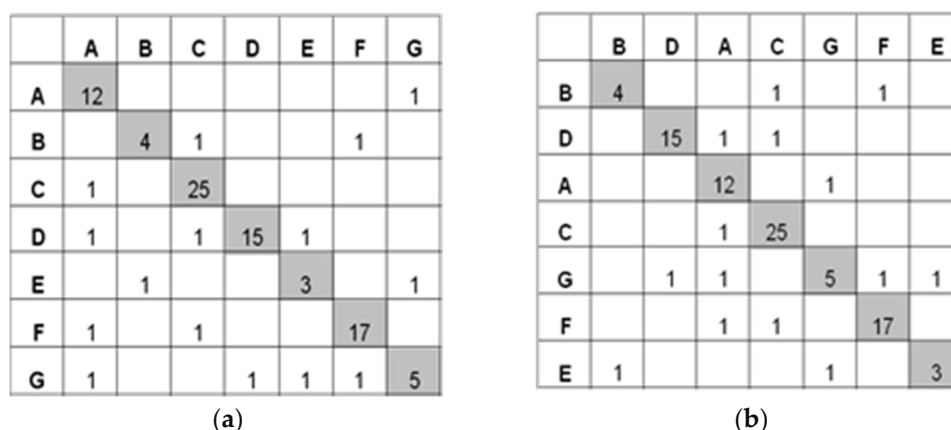


Figure 1. DSM representation before and after arrangement. (a) Before arrangement. (b) After arrangement.

3.2. Quality Equations

Quality equations are used to determine the number of tasks that need revisiting during the design process and to ensure the required quality. Each task is able to jump out from the coupling block if its output quality has fulfilled the requirement. Once the design converges to the acceptable multi-attribute performance level, the iteration could jump out from the coupling block. By adding these quality equations into DSM, this not only ensures the quality fulfillment of the task, but also decreases the risk of project duration delay due to poor work outputs. The design of the quality equations follows the logic of multiple attribute decision making (MADM). In practice, by collecting and plotting performance scores in terms of iteration steps based on historical performance data, the quality equation can be established. There are two main constituents in the equation: performance rating and quality improvement [11].

3.2.1. Performance Rating

Performance rating is used as the measurement of the quality level for each task. Performance refers to the degree of accomplishment of the tasks. It reflects how well the design fulfills the requirements, such as customer's demand and industrial standards. Performance rating can be gained using some MADM methodologies, such as the analytical hierarchy process (AHP). Using MADM methods, a design project manager can check the level of performance of each design task. Instead of focusing on sort of methodologies to gain performance rating value, this research mainly concentrates on how to use the performance rating as a tool to determine the number of iterations for each task.

3.2.2. Quality Improvement

It is reasonable to assume that an improvement in design performance occurs as the task is repeated a number of times. The increase in the number of trials or iterations generally leads to the improvement of the design quality. When more iterations occur, a higher quality level is achieved. Once the quality rating value goes beyond the satisfactory level, the iteration is able to jump out from the coupling block. This research uses the concept of quality improvement to determine the number of iterations needed for each task to achieve the requirements (required quality level). In concept, the quality equation is expressed as a curve in which the performance score on the Y axis is in accordance with iteration step on the X axis. The difference between the values in the X axis for the required quality and performance rate in Y will become the number of iterations needed for each activity to fulfill the requirement. Figure 2 depicts an example of the quality equation for an activity. Originally, the activity has a quality level of 2 and the requirement is to achieve the quality level of 4. The activity has to be iterated at least 4 times to achieve the required quality, which is yielded from 10 minus 6 on the X axis.

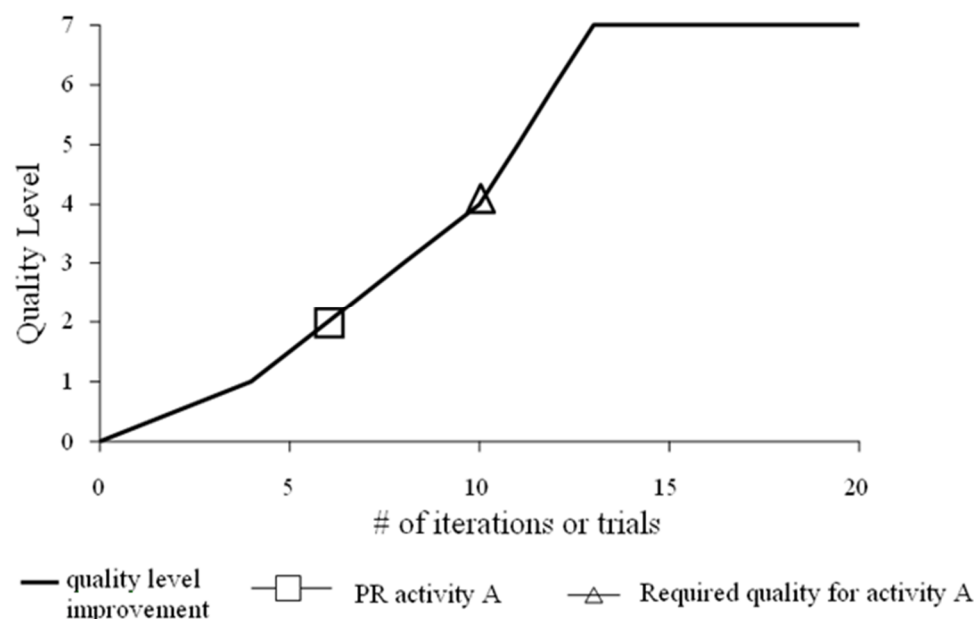


Figure 2. Concept of quality equation.

3.3. Resource Constraints

Resources for project activities are always limited in the real world. The unlimited resources assumption in the classical DSM technique may not be feasible. To avoid the waste and shortage of resources in a project, scheduling must take resource allocation into consideration based on the project characteristics. In a design project, engineers and project manager are the main manpower resources. Each design task has its own type and number of required manpower. The number and the type of the manpower are generally more than one and limited in a design project. In this study, all of the engineer's time must be devoted to a task until it is completed (i.e., no preemption is allowed in this study).

3.4. Genetic Algorithm for DSM

Genetic algorithms (GA) have been popularly used in many areas: scheduling and sequencing, transportation, reliability optimization, constrained or unconstrained optimization, artificial intelligence (AI), and many others [12]. Significant research has been done in the optimization of construction scheduling using genetic algorithms [13–15]. The improved DSM in this study aims to determine the scheduling order of the individual activities with the consideration of design quality and resource constraints during the iterative design process. A GA is used to re-organize and optimize the task flow for the iterative design project to achieve the minimum project duration [16]. There are four main GA steps: initialization, evaluation, selection, and crossover and mutation. Each step is further explained as follows.

3.4.1. Initialization

The first step in GA is to generate the initial population. A population (N) is a set of chromosomes, and it serves as a group of candidates or potential solutions. This research used integer form to represent the chromosome in GA. Figure 3a depicts a DSM example with five activities and Figure 3b describes its corresponding GA chromosome. A chromosome (also known as a string) in the population represents a possible sequence of activities and each gene (also known as a character) in the chromosome stands for an activity ID (or name). Each gene in a chromosome is selected randomly and only can be picked exactly once. Precedence relationship does not have to be considered when choosing

sequence of the gene, since in DSM, feedback dependency is allowed in estimating the project completion time.

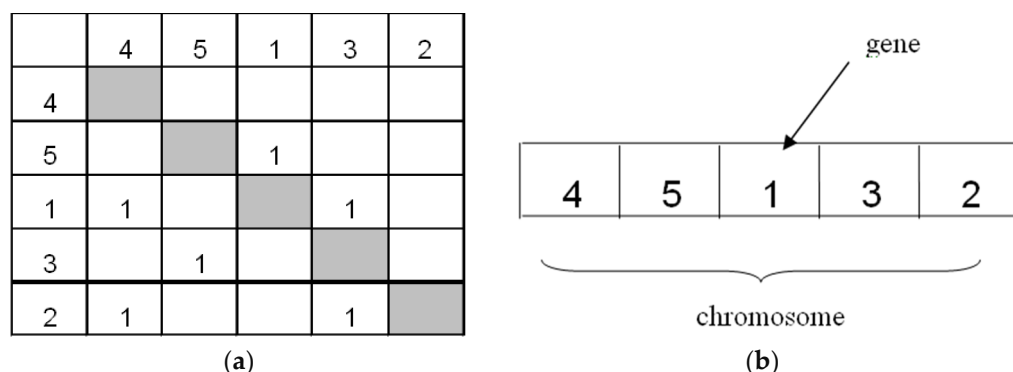


Figure 3. DSM and chromosome representation. (a) DSM data. (b) GA chromosome.

3.4.2. Evaluation (Fitness Function Calculation)

After a set of chromosomes is generated for potential solutions, each of them is assessed based on the fitness function defined. The fitness function shown below is to evaluate the quality (or fitness) of every chromosome in the population by calculating the fitness value. An example of time calculation for iterative design flow is depicted in Figure 4. The iteration for every feedback is assumed to occur once it causes the process to repeat the tasks in the feedback loop. The time for repeated tasks should be included as parts of the project duration. Task 2 is not included in the time calculation because Task 2 and Task 3 can be undertaken concurrently, so that the shorter of these two tasks (i.e., Task 2) is dropped from the total time calculation. Based on the method proposed by Mazur et al. [17], the total time period is $T1 + T3 + (T1 + T3) + T4 + (T1 + T3 + T4) + T5$.

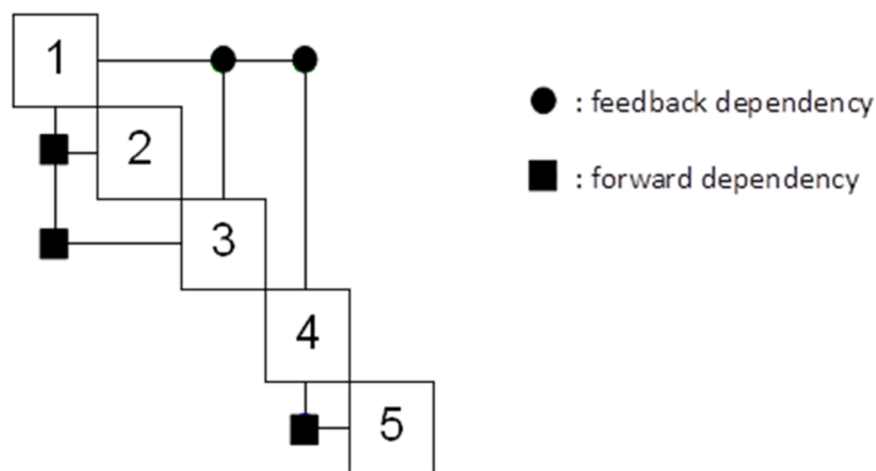


Figure 4. Example of DSM task flow.

3.4.3. Selection

There are many kinds of selection methods and generally the concepts of roulette wheel and elitism are used. The roulette wheel simulates the behavior of a roulette wheel, where each chromosome has a slot on the wheel. The size of each slot is proportional to the fitness of the corresponding chromosome. To select an individual, a random number is generated and the individual whose segment spans the random number is selected. In the roulette wheel selection, the fittest individual occupies the largest interval, whereas the least fit individual takes correspondingly smaller intervals.

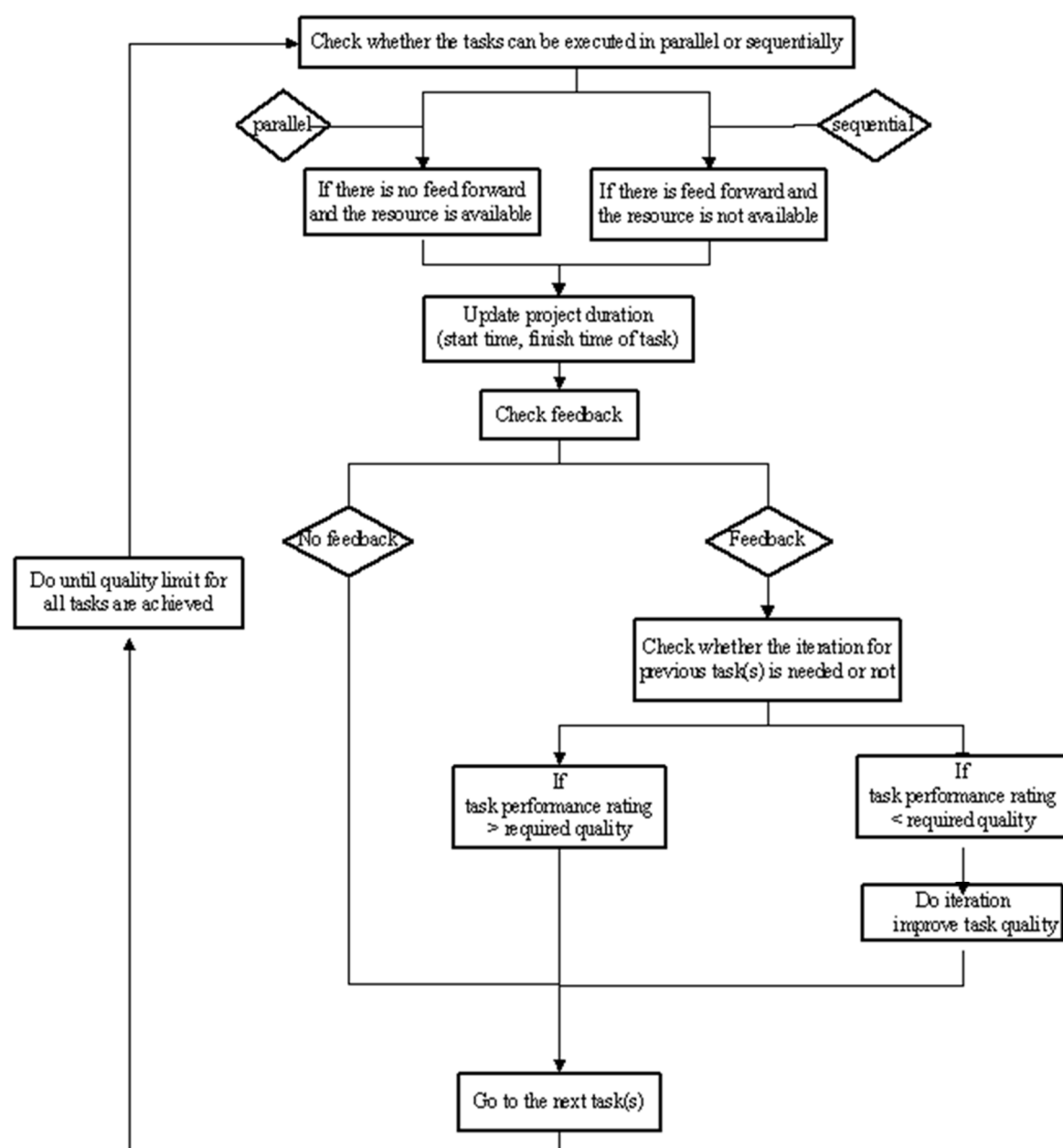
The second selection method is elitism, which is an extension of the selection process that guarantees the survival of the fittest chromosome. Using elitism, the simple solution is only copied from the fittest chromosome(s) from one generation to the next. The elitism operator has proved to increase the speed of convergence of the GA because it ensures that the best solution found in each generation is retained and guarantees that the final result will not become worse.

3.4.4. Crossover and Mutation

A special crossover method, a partially mapped crossover, is undertaken in this study. It passes ordering and value information from the parent to the offspring. A portion of one parent's string is mapped onto a portion of the other parent string and the remaining information is exchanged. For mutation, a swapping algorithm was used in this research. It only swaps two different numbers and exchanges both of them. Mutation is an auxiliary function of GA to prevent the genetic search from trapping at local optima. In the mutation operation, an appropriate mutation rate (generally around 0.01) is defined so that there is a low disruptive effect on the fitness of the overall population.

4. Case Study and Sensitivity Analysis

The overall flowchart for DSM fitness function calculation in this study is depicted in Figure 5. The GA search algorithm consists of two loops. The inner loop is developed to deal with iterative scheduling with the consideration of quality equations. The outer loop is for the optimal solution exploration under rare resource constraints. The model was first verified against Mazur et al. [17]. DSM data were adopted from Mazur et al. [17] and the functions of quality equations and resource constraints in our model were temporarily terminated. Comparing the project durations from both models, their values are identical (i.e., 19,824 days). This reveals that the logic and the codes of our model are accurate.



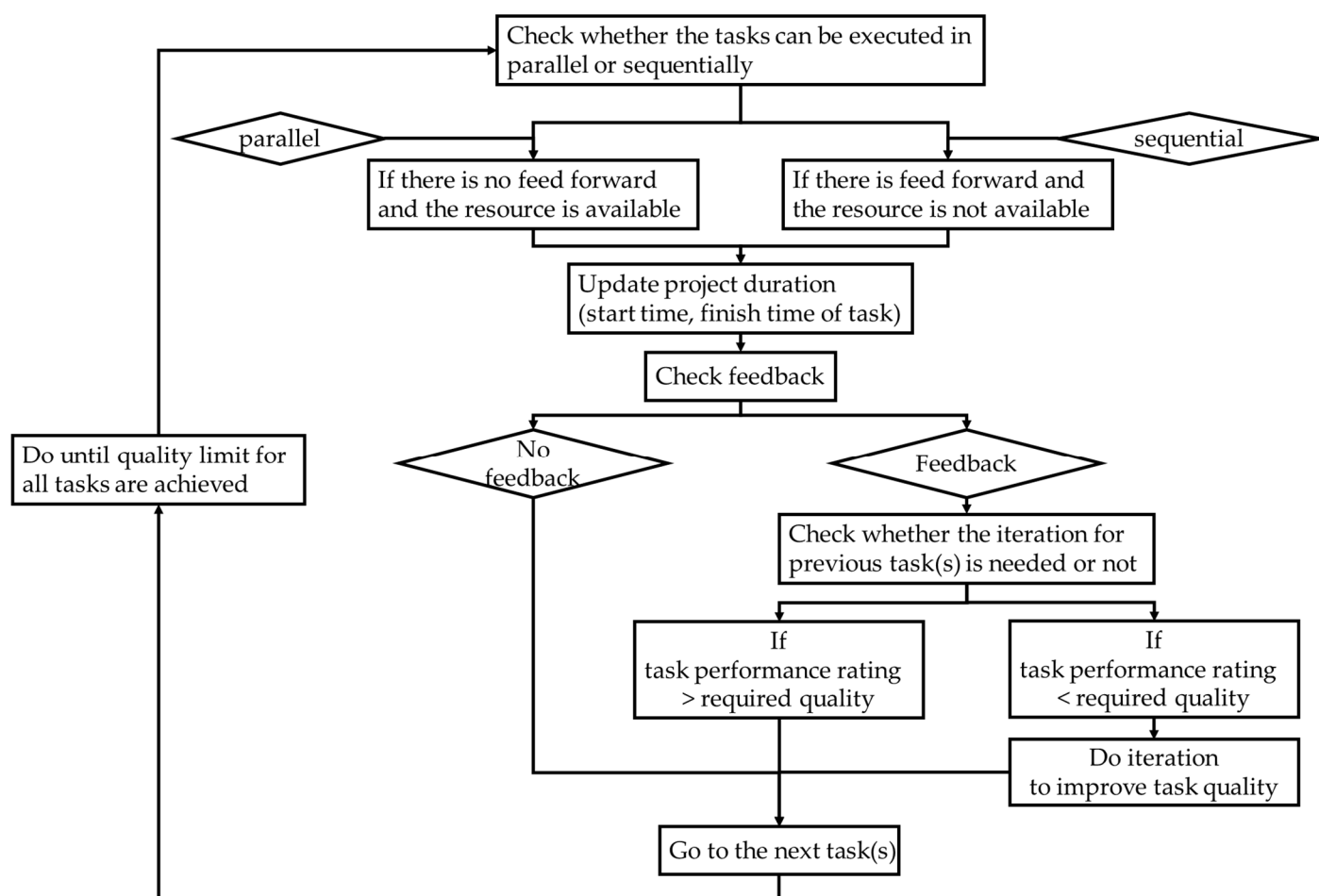


Figure 5. Flowchart for DSM fitness function calculation.

Furthermore, a new set of DSM data used as the case study was taken from an engineering consultant company in Taiwan. The company was a building design project with a 1.5-year construction duration and the design fee was around USD \$430,000. There are 45 activities which are part of the design process. The descriptions of the activity durations and their relationships are shown in Figure 6. The values in the diagonal line indicate the duration for each design task (in days). Thirteen kinds of engineer are required as resources in the project. Their demands and the upper limits are listed in Table 2. For ease in the simulation, performance ratings and the quality level improvement curves used in this case study were previously assumed and are depicted in Table 3. The quality equations used in this case study are assumed to be linear. In the practical implementation, such quality equations can be acquired based on real exercise and data plotting.

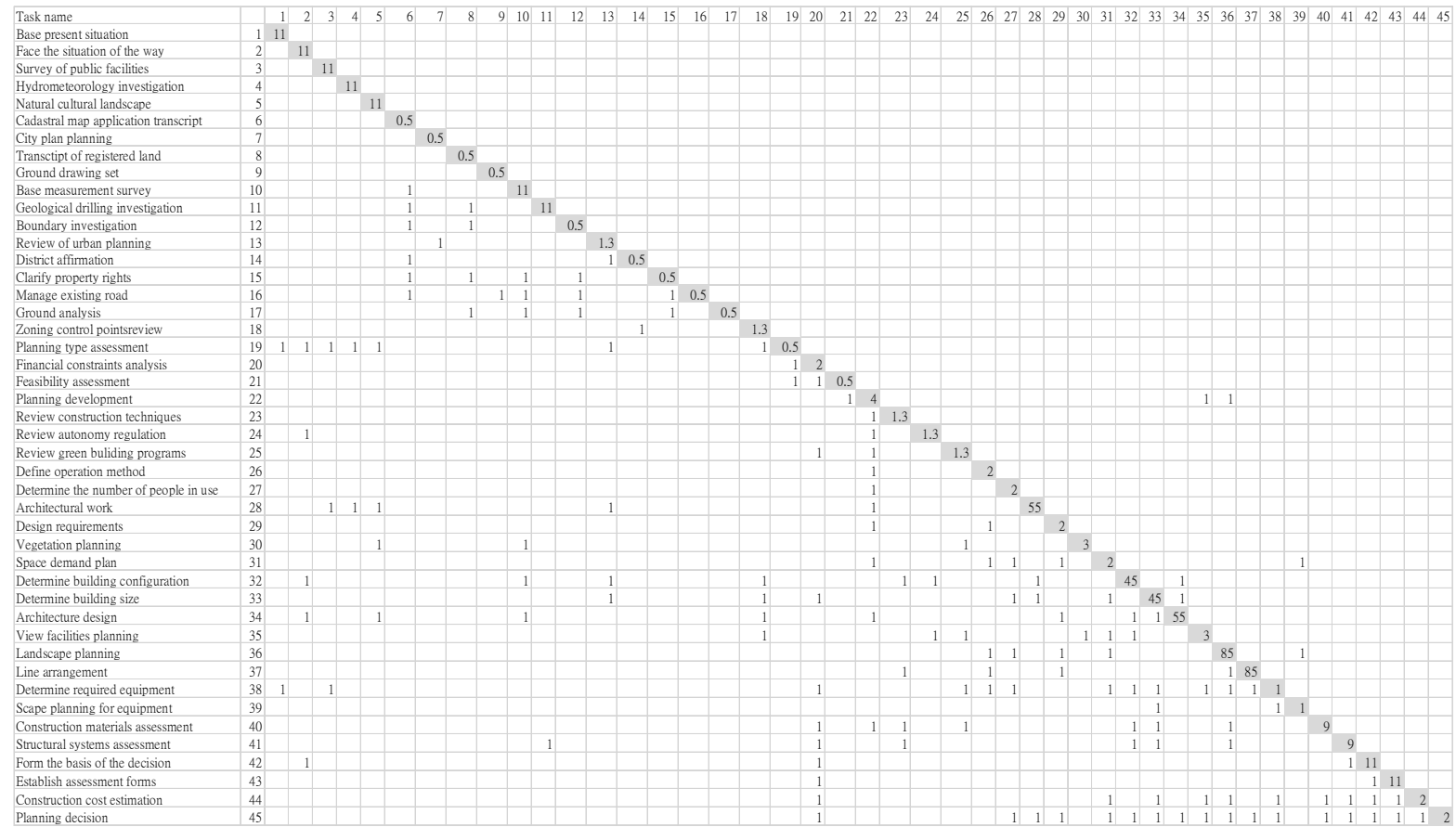


Figure 6. Activity durations and their relationships of test case.

Table 2. Engineers used in the design project and their upper limit of test case.

Task Number	Task Name	Resource Number												
		1	2	3	4	5	6	7	8	9	10	11	12	13
1	Base present situation			1							2	2		
2	Face the situation of the way			1							2	2		
3	Survey of public facilities			1							2	2		
4	Hydrometeorology investigation			1							2	2		
5	Natural cultural landscape			1							2	2		
6	Cadastral map application transcript			1										
7	City plan planning			1										
8	Transcript of registered land			1										
9	Ground drawing set			1										
10	Base measurement survey			1							2	2		
11	Geological drilling investigation			1							2	2		
12	Boundary investigation			1										
13	Review of urban planning	1	1											
14	District affirmation			1										
15	Clarify property rights		1											
16	Manage existing road		1											
17	Ground analysis		1											
18	Zoning control points review	1	1											
19	Planning type assessment	1												
20	Financial constraints analysis		1											
21	Feasible assessment	1												
22	Planning development	1	1											1
23	Review construction techniques	1	1											
24	Review autonomy regulation	1	1											
25	Review green building programs	1	1											
26	Define operation method		1											
27	Determine the number of people in use		1											
28	Architecture work	1	1		1	1								
29	Design requirements		1											
30	Vegetation planning									2				
31	Space demand plan		1											
32	Determine building configuration	1	1		1									
33	Determine building size	1	1		1									
34	Architecture design	1	1		1	1								
35	View facilities planning									2				
36	Landscape Planning	1	1		1									
37	Line arrangement	1	1		1									
38	Determine required equipment							1						
39	Space planning for equipment							1						
40	Construction materials assessment						2				1			
41	Structural systems assessment						2				1			
42	Form the basis of the decision						2				1			1
43	Establish assessment forms						2				1			1
44	Construction cost estimation	1	1											
45	Planning decision	1	1											
	Upper limit of resources	1	1	1	1	1	2	1	0	2	2	2	0	1

Rem: resource 1: project manager; 2: ad hoc engineer; 3: administrative support; 4: mapping engineer; 5: model making engineer; 6: structure design engineer; 7: mechanical engineer; 8: electrical engineer; 9: landscape engineer; 10: industrial engineer; 11: geographic engineer; 12: lighting design engineer; 13: CM.

Table 3. Quality level and performance rating of test case.

Type of Engineer	Performance Rating (PR)
Project Manager	5
Ad hoc designer	4
Administrative Support	2
Mapping engineer	3
Model making engineer	2
Structural design engineer	3
Mechanical engineer	4
Electrical engineer	3
Landscape engineer	4
Industrial engineer	2
Geographic engineer	2
Lighting design engineer	2
CM	3
Quality level	Accuracy (the Correctness of Work Duties Performance)
1	Make frequent errors
2	Careless, make few errors
3	Usually accurate, only small number of mistakes
4	Require little supervision, exact and precise most of the time
5	Require absolute minimum of supervision, almost always accurate

The GA convergence is generally reached before the 100th iteration. The basic reason is that the elitism method is used in the selection process. The survival of the fittest chromosomes in a generation are kept into to the next generation. There are two optimum sequences that have the same project duration:

	1	2	3	4	5	6	7	8	11	10	12	18	13	14	15	16	19
I.	22	23	24	25	26	28	27	29	30	31	32	33	36	37			
	35	34	38	40	39	41	42	43	44	45							
	1	2	3	4	5	6	7	8	11	10	12	18	13	14	15	16	17
II.	19	9	20	21	22	23	24	25	26	28	27	29	30	31	32	33	
	36	37	35	34	38	40	39	41	42	43	44	45					

The minimum project duration is 185.4 days and there are five feedback dependencies. It is still 2.3 days faster than the original DSM/DSM data (187.7 days), although there are more feedback dependencies compared to the original DSM. More feedback dependencies do not always increase project duration. The duration may occasionally be shortened due to less forward dependencies, and hence more activities can be done in parallel. Moreover, accelerating project duration by moving the dependency (forward to feedback) becomes more reliable if the iteration number for each feedback dependency can be determined.

A small iterative schedule with three activities is illustrated below as an explanation example (as shown in Figure 7). In Figure 7a, all the dependencies are forward dependencies, while in Figure 7b,c, they are all feedback dependencies. The total duration is 18 days when all the dependencies are forward dependencies because activities B and C can only be executed after activity A is finished. Comparatively, if the relationships have some feedback dependencies, the total duration may be shorter or longer. It becomes shorter if no iteration occurs and it turns out to be longer if activities B and C have to be iterated once or more. If activities B and C are required to be iterated at least once based on the quality equation, then the sequence of activity in Figure 7a is preferred with a shorter duration. On the other hand, if activities B and C do not have to be iterated, the sequence of activity in Figure 7b provides a better outcome.

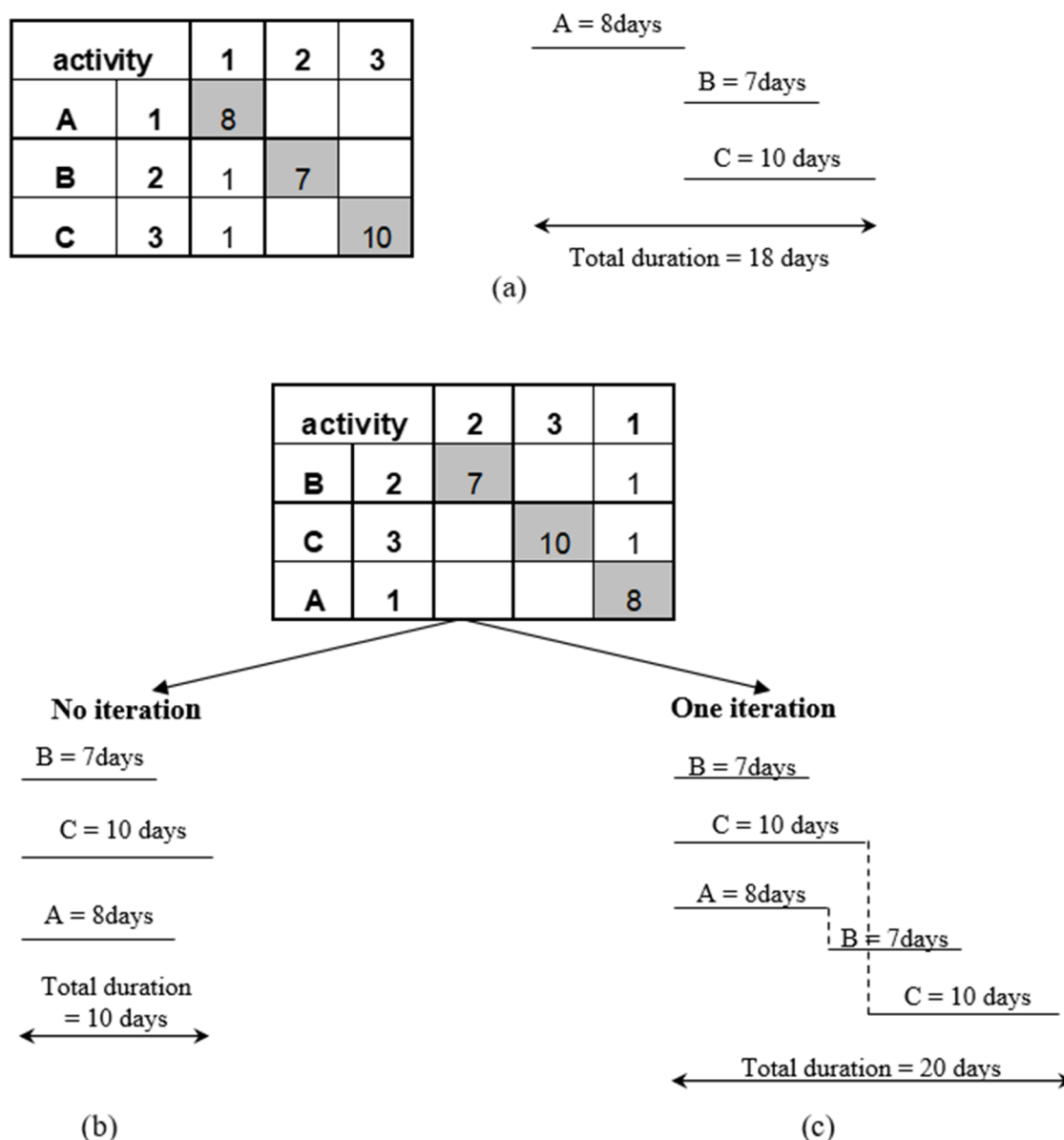


Figure 7. Discussion of time calculation under different DSM scenarios. (a) Activities with all forward dependences. (b) Activities with all backward. (c) Activities with all backward dependence.

4.1. Sensitivity Analysis and Discussion

There are four sensitive factors in the proposed DSM: activity dependencies, activity duration, resource constraints, and quality equations. Generally, activity dependencies and activity duration are defined and fixed based on the design project characteristics. The sensitivity analysis is conducted based on the variables of resource constraints and quality equations.

4.1.1. Quality Equations

Quality equations have two major parameters: performance rating and quality improvement. The sensitivity analysis was conducted to identify their impact. Table 4 summarizes the results of the sensitivity test. In Scenario 1, the equation is assumed to be linear as the baseline. In Scenario 2, the performance rating values of all tasks are adjusted to have a maximum value of 5; i.e., no quality improvement is needed since the qualities of all tasks have already achieved the highest level. In Scenario 3, the quality improvement

curve is assumed to have a higher curvature, i.e., less iteration is needed for quality improvement. As no quality improvement in Scenario 2 is needed, the project duration is much less than in Scenario 1 (129 days compared to 185.4 days). Meanwhile, one iteration for quality improvement in Scenario 3 is required; as a result, the project durations are still less than Scenario 1 (140.9 days compared to 185.4 days). This means that performance rating and quality improvement iteration greatly influence the DSM sequences to get the optimal values. It is observed that more feedback dependencies tend to occur by raising performance rating values and reducing quality improvement iterations. As previously discussed, if more forward dependencies are switched into feedback dependencies, the related activities can be done in parallel. That causes the project duration to be shorter (coupled activities are an exception).

Table 4. Project durations under different quality equations.

Scenario	Resource Constraint	Performance Rating	Quality Improvement	Result
	# Of Resource			(Days)
1	13	different (range: 1–5)	0.5 * iteration + PR	185.4
2	13	uniform: 5	-	129
3	13	different (range: 1–5)	iteration + PR	140.9

*. (times).

4.1.2. Resource Constraints

Resources for design activities are limited in the real world. To avoid the waste and shortage of resources on a design project, scheduling must include resource allocation. In general, the construction project regards the amount of resources as the resource constraints. Nevertheless, in the design phase, the resource types and amount are limited and may not be critical. The resource types and their time availability, especially rare resources, play an important role in the completion of design tasks. For example, outsourcing experts, who usually have high credibility, cannot be able to stand idle. Additionally, engineers are often appointed by several projects simultaneously in real practice. They are not always available to a single design project. The scheduler has to match the sequence of the design project with the available time of the rare resources. The task under wait will proceed right after the due rare resources have finished their tasks on hand.

In the tested design case, the mapping engineer is regarded as the rare resource. Their stay period and arrival time are two major parameters. Rare resources usually stay within the limited design period. Practically, the shorter the stay time of a rare resource is, the more the rarer resources are needed. This study mainly focused on the arrival time plan with a fixed rare resource. Different arrival time is set as a control parameter to conduct sensitivity analysis. The GA search algorithm consists of two loops. The inner loop is developed to deal with iterative scheduling, with the consideration of quality equations. The outer loop is for the optimal solution exploration under rare resource constraints; i.e., the different arrival time of the mapping engineer in our case. Table 5 depicts the result of sensitivity analysis. The basic findings are concluded as follows. First of all, the optimal search algorithm tends to rearrange related activities into the chain block to achieve minimum project duration; nevertheless, that could cause the increment of feedback. In our case, activities 31, 37, 36, 33, 32, 34, and 28 need the support of the mapping engineer. The final project duration is 186.4 and the best arrival time of the mapping engineer is at 100th. The sequence of design activities is listed below.

	1	2	3	4	5	6	7	8	11	10	2	18	13	14	15	16	19
III.	17	9	20	21	22	23	24	25	26	27	29	30	31	37			
	36	33	32	34	28	35	38	40	39	41	42	43	44	45			

Secondly, as shown in Table 5, it is found that the different arrival time of rare resources (the mapping engineer in our case) affects the project completion time and also the stay time period of rare resources. The basic reason is that by moving expert arrival time, the sequence in the system will change accordingly, and it will also alter dependencies between activities. Because of that, the rare resource may need to be re-invited to the project several times. This means that the stay time period of the rare resource could increase to fulfill the project requirements (quality requirements, in our case). Moreover, project durations for some scenarios are less than the original DSM and defined by a two-loop GA algorithm, but some are longer. This result is highly affected by the feedback dependency. As mentioned before, feedback dependency can extend or reduce project duration. The different arrival time largely changes feedback dependency.

Table 5. Summary of project durations under different resource constraints.

Scenario	Arrival Time of Rare Resource	Project Duration (Days)
Benchmark	100	186.4
1	0	162
2	20	183.5
3	40	194.9
4	140	238.9

5. Conclusions and Further Directions

Classical DSM mainly focuses on the number of feedback dependencies. It is believed that the fewer the feedback dependencies, the shorter the completion time. Nevertheless, the assumption may not hold because tasks ought to be re-visited several times if the required qualities do not meet the requirements. Moreover, resources for iterative design projects are generally limited in practice. This paper proposes a novel DSM algorithm combining the classical DSM concepts with quality equations. The quality equations were used to determine the number of tasks that ought to be re-visited during the iterative process for fulfilling the design requirements. It was proven that quality equations greatly influence the sequence of activities and the project duration of iterative schedules. Finally, resource constraints were considered to determine their effects. Resource constraints (rare resources, in our study) strongly influence the stay time period of rare resources and project durations.

While this study is exploratory in nature, further research is needed, especially in establishing procedures to develop practical quality equations. Furthermore, efficiently solving iterative schedules for rare resources is required. The case of rare resources may happen in the iterative schedules, due to some project resources that have fixed arrival times and limited durations in the project. Finally, risks frequently occur during projects, especially in complex concurrent projects. It is necessary to explore the logic and the effects of risk factors on DSM.

Author Contributions: S.-S.L. designed and supervised the research; T.D.N.S. ran the simulation and wrote the manuscript draft; C.C.-W.H. edited and proofread the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Steward, D.V. The design structure system: A method for managing the design of complex systems. *IEEE Trans. Eng. Manag.* **1981**, *28*, 71–74, doi:10.1109/TEM.1981.6448589.
2. Steward, D.V. *Systems Analysis and Management: Structure, Strategy, and Design*; Petrocelli Books: New York, NY, USA, 1981.
3. Rogers, J.L.; McCulley, C.M.; Bloebaum, C.L. Integrating a genetic algorithm into a knowledge-based system for ordering complex design processes. In *Artificial Intelligence in Design'96*; Springer: Dordrecht, The Netherlands, 1996; pp. 119–133.
4. Austin, S.; Baldwin, A.; Li, B.; Waskett, P. Analytical design planning technique (ADePT): A dependency structure matrix tool to schedule the building design process. *Constr. Manag. Econ.* **2000**, *18*, 173–182, doi:10.1080/014461900370807.
5. Smith, R.P.; Eppinger, S.D. A predictive model of sequential iteration in engineering design. *Manag. Sci.* **1997**, *43*, 1104–1120, doi:10.1287/mnsc.43.8.1104.
6. Hisham, M.E.; Bao, H.P. A simulation-based optimization framework for product development cycle time reduction. *IEEE Trans. Eng. Manag.* **2006**, *53*, 69–85, doi:10.1109/TEM.2005.861805.
7. Hejducki, Z.; Rogalska, M. The application of time coupling methods in the engineering of construction projects. *Tech. Trans.* **2017**, *9*, 767–776, doi:10.4467/2353737XCT.17.148.7160.
8. Eppinger, S.D.; Whitney, D.E.; Yassine, A.A. The Design Structure Matrix—DSM Home Page. Available online: <http://www.dsmweb.org/> (accessed on 20/08/2018).
9. 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, doi:10.1109/17.946528.
10. Maheswari, J.U.; Varghese, K.; Sridharan, T. Application of dependency structure matrix for activity sequencing in concurrent engineering projects. *J. Constr. Eng. Manag.* **2006**, *132*, 482–490, doi:10.1061/(ASCE)0733-9364(2006)132:5(482).
11. Pekar, D.G. *Modeling Design Time and Quality Tradeoffs*. Master's Thesis, MIT, Boston, MA, USA, 1992.
12. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Longman Publishing Co., Inc., 75 Arlington Street, Suite 300 Boston, MA, USA, 1989.
13. Chan, W.T.; Chua, K.H.; Kannan, G. Construction resource scheduling with genetic algorithms. *J. Constr. Eng. Manag.* **1996**, *122*, 125–132, doi:10.1061/(ASCE)0733-9364(1996)122:2(125).
14. Leu, S.S.; Yang, C.H. A genetic-algorithm-based resource-constrained construction scheduling system. *Constr. Manag. Econ.* **1999**, *17*, 767–776, doi:10.1080/014461999371105.
15. Leu, S.S.; Yang, C.H. GA-based multicriteria optimal model for construction scheduling. *J. Constr. Eng. Manag.* **1999**, *125*, 420–427, doi:10.1061/(ASCE)0733-9364(1999)125:6(420).
16. Zhuang, M.; Yassine, A.A. Task scheduling of parallel development projects using genetic algorithms. In *Proceedings of the DETC '04 ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DETC2004/DAC-57159, Salt Lake City, UT, USA, September 28–October 2, 2004.
17. Mazur, L.M.; Chen, S.J.; Sasiadek, M.; Kielec, R.; Semik, W. Project task coordination using design structure matrix and genetic algorithm. In *Proceedings of the Portland International Conference on Management of Engineering and Technology*; Portland, OR, USA, 25–29 August 2019.