Computers & Industrial Engineering 127 (2019) 1–7

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Modeling and analysis of cascading failures in projects: A complex network approach

Ning Guo[a,*], Peng Guo[a], Haiyong Dong[b], Jing Zhao[a], Qingye Han[a]

[a] School of Management, Northwestern Polytechnical University, Xi'an 710072, PR China
[b] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, PR China

ARTICLE INFO

Keywords:
Project
Cascading failures
Complex networks
Risk management

ABSTRACT

Due to the complex integration of the tasks in a project, the failure of a single task can trigger extremely large-scale failures and destroy a considerable part of the overall project. To investigate cascading failures in projects, in this paper, the project is first abstracted as a weighted directed network composed of tasks and task interactions, after which a cascade model that takes account of the project's self-protection mechanism is developed to examine a failure propagation process originates from a single task failure. The model is then applied to examine cascading failures in a project under three types of single task failures with varying parameters. The experiment results demonstrate that the method is very effective in predicting cascading failures, evaluating the impact of cascading failures on the project, and identifying large cascades. The insights gained from the simulation results have implications for managers in the implementation of protective measures to mitigate cascading failure risk and avert project catastrophes.

1. Introduction

A project is a scope of work in which human, material, and financial resources are organized under significant constraints (Chapman & Ward, 2003). Projects are fundamental to modern society, with more than one-fifth of global GDP being generated from projects (Turner, Huemann, Anbari, & Bredillet, 2010). To ensure successful project implementation in today's competitive environment, significant research has addressed the problems associated with the triple project constraints: cost, time and quality (Mohammadipour & Sadjadi, 2016; Qazi, Quigley, Dickson, & Kirytopoulos, 2016; Tabrizi & Ghaderi, 2016), with many scholars having focused on project risk management (Muriana & Vizzini, 2017; Thamhain, 2013; Zeng, An, & Smith, 2007). The failure of a single task can trigger a cascade of subsequent failures, which can result in whole project collapse (Ellinas, 2018a; Ellinas, Allan, & Johansson, 2016). Although many exogenous risk sources have investigated in project risk research, the risk of cascading failure resulting from endogenous task failures has received less research attention. As efficient and effective project management requires the appropriate management of all sources of uncertainty (Chapman & Ward, 2003), it is essential to understand cascading failures so as to be able to deal with the potential negative effects. This study, therefore, explores cascading failures in project in order to give guidance to project managers on effective preventive measures to reduce cascading failure risk.

1.1. Literature review

A cascading failure is when one of the elements fails in a system of interdependent elements, which then causes a cascade of failures of other elements because of the elemental interconnections (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006; Crucitti, Latora, & Marchiori, 2004; Mirzasoleiman, Babaei, Jalili, & Safari, 2011; Motter & Lai, 2002). A lot of cascading failure research has developed in the power grid, transportation, and social-economic system disciplines (Hasan & Ukkusuri, 2011; Huang, Vodenska, Havlin, & Stanley, 2013; Kinney, Crucitti, Albert, & Latora, 2005; Koç, Warnier, Kooij, & Brazier, 2013; Van Eeten, Nieuwenhuijs, Luiijf, Klaver, & Cruz, 2011; Wang & Rong, 2011); however, there has been little focus on cascading failures in project.

As project collapse often results from cascading failures induced by a single task failure (Ellinas, Allan, Durugbo, & Johansson, 2015), recent research has been focused on exploring cascading failures in projects. For example, Ellinas (2018a) developed a computational model to describe cascading failures in projects and found that a modest local failure could trigger large-scale project failures, Ellinas (2018b) also studied the impact of indirect interactions during cascading failure

* Corresponding author.
E-mail addresses: nerine@mail.nwpu.edu.cn (N. Guo), guopeng@nwpu.edu.cn (P. Guo), donghaiyong@mail.nwpu.edu.cn (H. Dong), jzhao@nwpu.edu.cn (J. Zhao), hanqy11@mail.nwpu.edu.cn (Q. Han).

https://doi.org/10.1016/j.cie.2018.11.051
Received 23 February 2018; Received in revised form 28 September 2018; Accepted 25 November 2018
Available online 26 November 2018
0360-8352/

propagation, and Wang, Yang, Zhang, and Song (2018) explored mitigation strategies for schedule risks by restraining cascading failures in an organization network. Existing studies have made significant contributions to project systemic risks originating from cascading failures. However, while most previous research has indicated that large cascades may occur if some vital nodes fail (Fang, Yang, & Yan, 2014; Motter & Lai, 2002; Wang & Rong, 2011; Wu, Peng, Wang, Chan, & Wong, 2008), studies focusing on discussing the role of different tasks in cascading failures have been somewhat limited. Understanding the significant performance degradation caused by various task failures and identifying the specific tasks that can trigger large-scale failures could assist in developing protective measures to mitigate cascading failure risk. Therefore, this paper explores the impact of several types of task failures on cascading failures in projects.

An initial step in studying of cascading failures is the mathematical presentation of the failure propagation process (Motter & Yang, 2017). The three most common cascade models are branching process models (Dobson, 2012; Harris, 2002), percolation models (Granovetter, 1978; Watts, 2002), and flow redistribution models (Crucitti et al., 2004; Motter & Lai, 2002). Flow redistribution models can be used to describe cascading failures in flow networks (Motter & Yang, 2017; Zeng & Xiao, 2014). Node failures in flow networks alter the flow balance and result in network load redistributions that can initiate a cascade of overload failures (Crucitti et al., 2004; Motter & Lai, 2002). The project workflow travels from the upstream tasks to the downstream tasks, with the workflow from a task being delivered to its successor neighbor tasks; that is, project networks are flow networks. In this paper, the failure of task refers to a task completion quality that is below the completion criteria. Moreover, a task failure can impact its downstream tasks and thus its immediate downstream tasks need to bear an extra load from the failed task. According to the above project's characteristics, a flow redistribution model is tailored to analyze cascading failures in projects.

1.2. Focuses of this study

This paper uses a complex network approach to explore the role of different tasks in project cascading failures and to identify which types of task failures can induce the large cascade. To achieve this goal, the project is first abstracted as a weighted directed network to reveal its topological structure and the specific features, after which a cascade model is adapted to investigate the project failure propagation. A comparison study of three task attack strategies is then conducted to investigate the effects on cascading failures. Based on the simulation results, preventive measures are then provided to prevent cascading failures, effectively protect different task types, and improve project robustness. Overall, this paper lays the foundation for the project cascading failure research and enriches project risk management studies.

The central contributions of this paper are as follows. First, the project failure propagation process is modelled using an extended flow redistribution model that takes account of the project's self-protection mechanism to mimic the catastrophic project failure propagation process, analyze cascading failures resulting from endogenous task failures, and evaluate the impact of task failures on the overall project. Second, the effects of three types of task failures on cascading failures are examined, from which it was found that different task failures have widely varied effects on cascading failures. This work improves the understanding on the significant performance degradation caused by various task failures and identifies the specific tasks that can trigger large-scale failures. Third, corresponding protective measures are proposed to mitigate cascading failure risk caused by various task failures. A single task failure can significantly affect many downstream tasks because of cascading failures. In this paper, several different types of task failures are examined and thus commensurate practical preventative measures can be proposed to assist managers reduce cascading failure risk.

The remainder of this paper is organized as follows. In Section 2, the project is first abstracted as a weighted directed network, after which a cascade model is developed to replicate the project failure propagation process. Section 3 introduces the three types of task failures in this paper. Then, different types of task failures and the consequent cascading failures are explored through simulations in Section 4. Section 5 discusses the practical implications, research limitations, and possible future research directions, and Section 6 gives the final conclusions.

2. Modeling cascading failures in project

Cascade models have usually been built based on networks, that is, a real-world complex system is first abstracted as a complex network to expose the hidden laws behind the system from a global structural point of view (Boccaletti et al., 2006), after which a cascade model is used to capture the failure propagation in the network (Motter & Yang, 2017). In order to study cascading failures in projects, a project is first abstracted as a network in which the nodes represent tasks and the links represent the task interactions. A cascade model is then developed to replicate the project failure propagation process.

2.1. Network-based description of project

To build a suitable project network, the network model properties must be in line with the structure of real-world projects. Projects are a collection of tasks/activities that generally run in sequential order, with the upstream tasks providing the foundation for the downstream tasks; that is, a task can only be started once the predecessor tasks directly connect to this task have been completed. To capture these essential features, the project is abstracted as a network using a set of nodes that represent the tasks and a set of edges that express the sequential task order and the relationships between the tasks (nodes). Therefore, the project network is $G = (V, E)$, where $V = (v_1, v_2 \ldots, v_N$ is the node set and E is the edge set. As some edges have stronger connections than others, a weighted adjacency matrix W is constructed to express the weights of the edges, where $W = [w_{ij}]$ is an $N \times N$ asymmetric matrix, and N is the total number of nodes in the project network. As tasks that take longer tend to have a greater impact than shorter tasks (Ellinas et al., 2015; Ellinas et al., 2016), it is assumed that the node weight is relevant with the task duration. The edge between the nodes with greater weights tend to have a more significant influence than the edge between the nodes with smaller weights; therefore, from Park, Lai, and Ye (2004), the weight of an edge l_{ij} that connects v_i to v_j is assumed to be:

$$w_{ij} = \begin{cases} (w_i + w_j)/2, & if \quad v_i \quad directly \quad connects \quad to \quad v_j \\ 0, & otherwise \end{cases} \tag{1}$$

where w_i and w_j are the weights for v_i and v_j, $w_i = t_i$, $w_j = t_j$, and t_i and t_j are the durations for task i and task j.

2.2. Cascading failures model

As the interdependencies between the tasks give rise to multiple possible failure propagation channels, the failure of a task can easily affect its associated successor tasks and initiate a cascade of task failures that can threaten the project's operation. To replicate this catastrophic spreading process, the flow redistribution model is adapted.

First, an initial node load assignment and node capacity are introduced. Each component is assigned an initial load (Kim & Dobson, 2010) that characterizes the transport network dynamics (Goh, Kahng, & Kim, 2001). Depending on the scenario, the load is estimated based on the node betweenness centrality (Crucitti et al., 2004), the node degree centrality (Wang, 2013) or the node out-degree centrality (Tang, Jing, He, & Stanley, 2016). As the execution of each task in the workflow is related to both its immediate upstream tasks and its immediate

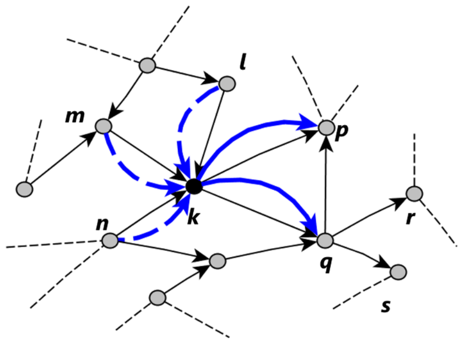

Fig. 1. Load adjustment rule after v_k fails.

downstream tasks, the initial load $L_n(0)$ for v_n is defined as a function of the degree for v_n; that is,

$$L_n(0) = (k_n)^{\alpha} \tag{2}$$

where α is the tunable parameter applied to adjust the strength of the initial load, k_n is the sum of the edge weights adjacent to v_n that denotes the degree for v_n, and $k_n = \sum_{j=1}^{N} w_{nj} + \sum_{i=1}^{N} w_{in}$ (Costa, Rodrigues, Travieso, & Boas, 2005).

Node capacity indicates the maximum load that a node can handle and it is strictly limited due to the cost (Zeng & Xiao, 2014). The capacity C_n for v_n is assumed to be linearly correlated with the initial load for v_n (Crucitti et al., 2004); that is;

$$C_n = (1 + \beta)L_n(0) \tag{3}$$

where β is the adjustable parameter, different values of β decides the tolerance of the node against disturbances, and C_n implies that v_n can handle a greater load than the initial load, which can assist in resisting disturbances.

As shown in Fig. 1, assuming $w_m = 2$, $w_n = 4$, $w_l = 4$, $w_k = 6$, $w_p = 8$, $w_q = 2$, $\alpha = 1$, and $\beta = 1$. According to Eq. (1), it can obtain $w_{mk} = 4$, w_{nk}=5, $w_{lk} = 5$, $w_{kp} = 7$, and $w_{kq} = 4$. According to the definition of degree, it can obtain $k_k = \sum_{j=1}^{N} w_{kj} + \sum_{i=1}^{N} w_{ik} = w_{kp} + w_{kq} + w_{mk} + w_{nk} + w_{lk} = 25$. Based on Eqs. (2) and (3), it can obtain $L_k(0) = (k_k)^{\alpha} = (k_k)^1 = 25$ and $C_n = (1 + \beta)L_k(0) = 50$.

Owing to the project's self-protection mechanisms, the task failures do not always affect its successor tasks because it can be restored. The project's self-protection mechanisms act as defenses against possible damage and prevent the project from being affected by task failures. Therefore, in this paper, a flow redistribution model that takes account of the project's self-protection mechanisms is developed to describe cascading failures in projects. As tasks in projects are sequentially performed, a failing task can only be restored by its immediate upstream tasks by sharing part of the capacity to reconstruct the task. In this paper, if the task has immediate upstream tasks, it is assumed that this reconstruction automatically occurs after task failure and the only way to restore a failing task is put an equal amount of load into its recreation. Fig. 1 shows the load adjustment rule after v_k fails.

As shown in Fig. 1, if v_k dissatisfies the completion criteria, its three neighbor predecessor nodes (v_m, v_n *and* v_l) first restore v_k. The load $L_k(0)$ to be shared by v_m, v_n and v_l is then defined as ΔL_{mk}, ΔL_{nk}, and ΔL_{lk}, which are proportional to the respective weights for l_{mk}, l_{lk} and l_{nk}; that is,

$$\begin{cases} \Delta L_{mk} = L_k(0)\frac{w_{mk}}{\sum_{b\in\Gamma_k} w_{bk}} \\ \Delta L_{lk} = L_k(0)\frac{w_{lk}}{\sum_{b\in\Gamma_k} w_{bk}} \\ \Delta L_{nk} = L_k(0)\frac{w_{nk}}{\sum_{b\in\Gamma_k} w_{bk}} \end{cases} \tag{4}$$

where Γ_k is the set of predecessor neighbor nodes directly connecting to v_k.

Generally, there are two possible situations that may arise after node restoration: (1) the failed node is restored; that is, its immediate upstream nodes are able to carry the load of the failed node; or (2) the failed node is not repaired. More precisely, if v_k is able to be recreated by the immediate upstream nodes, the inequalities

$$\begin{cases} L_m(0) + \Delta L_{km} < C_m \\ L_l(0) + \Delta L_{kl} < C_l \\ L_n(0) + \Delta L_{kn} < C_n \end{cases} \tag{5}$$

should be satisfied, and if the above inequalities are satisfied, v_k can be restored and does not influence its successor nodes. If any of the above inequalities are not satisfied, v_k is not reconstituted, which means that v_k inevitably influences its successor tasks and the immediate downstream tasks p and q need to bear greater loads to ensure the project workflow, which may trigger further load redistributions and result in cascading failures.

A failure can lead to further load redistributions; the load of the failed node is reallocated to other related nodes according to a redistribution rule, which may result in some nodes exceed their capacity and thus failed. As shown in Fig. 1, $L_k(0)$ is redistributed to v_p and v_q, which are defined as ΔL_{kp} and ΔL_{kq}, and which are proportional to the respective weights l_{kp} and l_{kq}; that is;

$$\begin{cases} \Delta L_{kp} = L_k(0)\frac{w_{kp}}{\sum_{b^{'}\in\Gamma_{k}^{'}} w_{kb^{'}}} \\ \Delta L_{kq} = L_k(0)\frac{w_{kq}}{\sum_{b^{'}\in\Gamma_{k}^{'}} w_{kb^{'}}} \end{cases} \tag{6}$$

where $\Gamma_{k}^{'}$ is the set of successor neighbor nodes directly connecting from v_k. If v_p and v_q are able to handle $L_k(0)$, the project proceeds smoothly; that is, if $L_p(1)$ and $L_q(1)$ are below capacity, the inequalities $L_p(1) < C_p$ and $L_q(1) < C_q$ should be satisfied; where $L_p(1) = L_p(0) + \Delta L_{kp}$ and $L_q(1) = L_q(0) + \Delta L_{kq}$. Replacing the parameters in the aforementioned inequalities with their definitions in Eqs. (2), (3) and (6), and after being appropriately simplified, we get;

$$\begin{cases} \left(\frac{\sum_{j=1}^{n} w_{kj}}{\sum_{j=1}^{n} w_{pj}}\right)^{\alpha} \frac{w_{kp}}{w_{kp} + w_{kq}} < \beta \\ \left(\frac{\sum_{j=1}^{n} w_{kj}}{\sum_{j=1}^{n} w_{qj}}\right)^{\alpha} \frac{w_{kq}}{w_{kp} + w_{kq}} < \beta \end{cases} \tag{7}$$

According to Eq. (7), the cascading failure propagation is related to parameters β and α. Therefore, the impact of cascading failures in a project is examined by varying parameters β and α in the numerical experiments.

Load redistribution increases the probability that the successor neighbor nodes exceed their capacity; if the carried load of one node exceeds its capacity, the node will fail. If the load redistribution of v_k causes some successor neighbor nodes to exceed their capacity; for instance, if v_q is unable to sustain the load from $L_k(0)$; this may trigger further failures in the successor neighbor nodes (v_r and v_s) through load redistribution. The cascading process iterates until there are no overloaded nodes in the project network. The cascading process may end after a few steps; however, it can also spread, thereby causing failures in a considerable part of the network or even the entire network (Koç et al., 2013; Motter & Lai, 2002).

Cascading failures can cause significant performance degradation. Several metrics such as a decrease in average network efficiency (Crucitti et al., 2004), normalized avalanche size (Mirzasoleiman et al., 2011) and the relative size of the largest connected component (Motter & Lai, 2002) have been proposed to calculate the damage caused by cascading failures. In this paper, two metrics are applied to quantify the impact of cascading failures on a project. One is, the first of which is the normalized avalanche size (CF_1), which is calculated by avalanche size (Mirzasoleiman et al., 2011):

$$CF_1 = \frac{\sum_{i\in A} N_{S_i}}{N_A(N-1)} \tag{8}$$

where N_{S_i} denotes the number of broken nodes induced by a v_i failure, S_i is the set of nodes that breakdown because of the v_i failure, $A$is the failed nodes, and N_A is the number of nodes that fail. Different tasks have different impacts on the project, with longer tasks generally having a much greater impact on the project (Ellinas et al., 2015, 2016); therefore, the second metric (CF_2) that takes account of the weight of failed nodes and is derived from the first metric; that is;

$$CF_2 = \frac{\sum_{i\in A}\sum_{j\in S_i} w_j}{N_A \sum_{n\in V} w_n} \tag{9}$$

where V is the set of all nodes in the project network. CF_2 is determined by the sum of the weights of the failed nodes divided by the weights of all nodes.

3. Attack strategies

Cascades are usually induced by small initial failures (Crucitti et al., 2004). Most previous research has indicated that large cascades may occur if some vital nodes fail (Motter & Lai, 2002; Wang & Rong, 2011; Wu et al., 2008). To investigate cascading failures in projects under different tasks failures, three types of failures are compared. As centrality measures are the most fundamental and frequently used measures of network structure (Newman, 2008), the three failure types in the proposed cascade model are based on the three common measures of node centrality: betweenness, degree, and closeness. Studying these types of failures can contribute to understanding the possible associations between the structural properties and cascading failures, and can provide a scientific reference for the development of effective protection measures.

(1) Attack on the node with a maximum betweenness (MB). Betweenness factors "connectivity" and demonstrates the importance of the node in the network. In this failure type, the nodes in order of descending betweenness in the project network are selected; if there is more than one node with maximum betweenness, one is randomly selected.
(2) Attack on the node with a maximum degree (MD). Degree denotes the number of node adjacencies. As the initial load of the node is a function of the node's degree in this paper, this type of failure equates with the highest load failures, which has been widely studied (i.e. Crucitti et al., 2004; Wang & Rong, 2011). This failure strategy chooses the nodes in a descending order of degree; if the maximum degree corresponds to more than one node, the node that is going to fail is chosen randomly.
(3) Attack on the node with a maximum closeness (MC). Closeness is defined as the reciprocal of the sum of the shortest path distances from this node to all other $N-1$ nodes. The rule of this failure strategy is to continually choose the node in descending order of closeness; if the maximum closeness corresponds to more than one node, one is randomly chosen.

4. Numerical experiments

In this section, simulation experiments are used to examine cascading failures in a construction project under the three failure types. The construction project aims to complete a 13 million USD Commercial Office in 743 days, and was originally presented in Ellinas et al. (2015). As both the task interactions and task durations are known, the project can be abstracted as a weighted directed network based on its topological structure and specific features. As shown in Fig. 2, the project was first abstracted as a network with 817 nodes and 806 weighted directed edges, in which the nodes symbolize the tasks and the edges represent the relationships between the tasks.

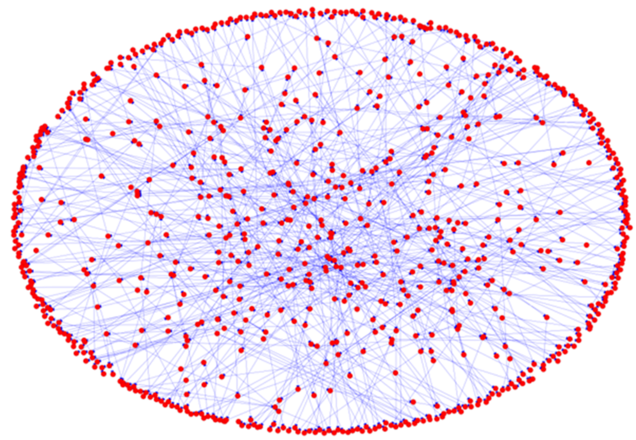

Fig. 2. A construction project network with NetworkX spring layout.

For the numerical experiments, only the cascading failures induced by single-node failures are investigated. For the three types of nodes failures, 10 nodes are chosen as the failed nodes for each failure type, after which the effect of the cascading failure is quantified using Eqs. (8) and (9). Further, as the cascading failure propagation is related to parameters β and α, the simulations are conducted with β changing from 0.1 to 1.5 at a step size of 0.1, and α respectively taking the values 0.4, 0.7, 1.0, and 1.3. The simulation was performed using Python and NetworkX, the results for which are presented in Figs. 3–6.

From Fig. 3, there are the following observations: (1) with an increase in the adjustable parameter β, the project's capacity to cope with the cascading failures increases, and both CF_1 and CF_2 reduce gradually; β_c is the lowest value for β to avert a cascading failure. Note that $\beta_c = 1.3$ for the MB, and $\beta_c = 1.0$ for the MC; (2) the drop rates of CF_1 and CF_2 vary markedly when β changes across the three failure types; for instance, the CF_1 of the MC dramatically decreases as β increases from 0.1 to 0.4, and the CF_2 of MD remains the same when $\beta = [0.8, 1.5]$; (3) on average, the CF_1 of MB and MC is slightly higher than the CF_2 of MB and MC, and the CF_1 of MD is significantly lower than the CF_2 of MD.

From Fig. 4, there are the following observations: (1) The$\beta_c = 1.5$ for MB, and$\beta_c = 1.1$ for MC; however, even when the capacity of the load is two and a half times the initial load, the CF_1 and CF_2 of MD does not reach zero, which indicates that its predecessor neighbor nodes are unable to handle the failure of the MD node; (2) The largest cascades are more probably to be induced by MB when $\beta \leq 0.2$, an MD failure

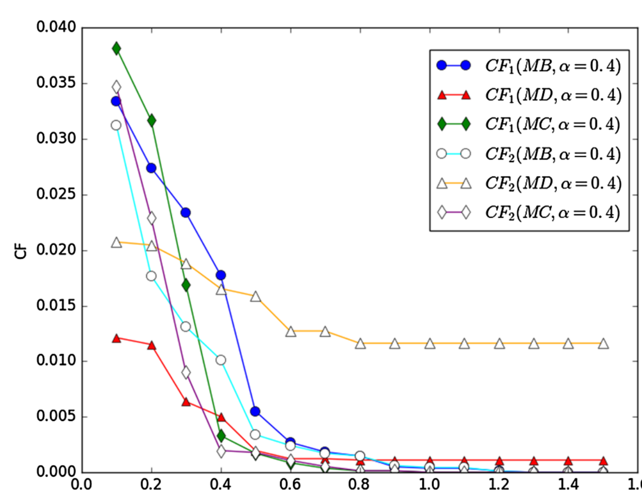

Fig. 3. Impact of cascading failures on a project when $\alpha = 0.4$.

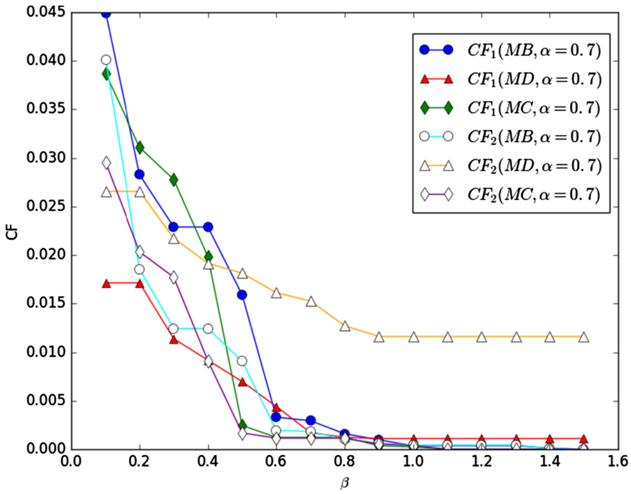

Fig. 4. Impact of the cascading project failures on a project when $\alpha = 0.7$.

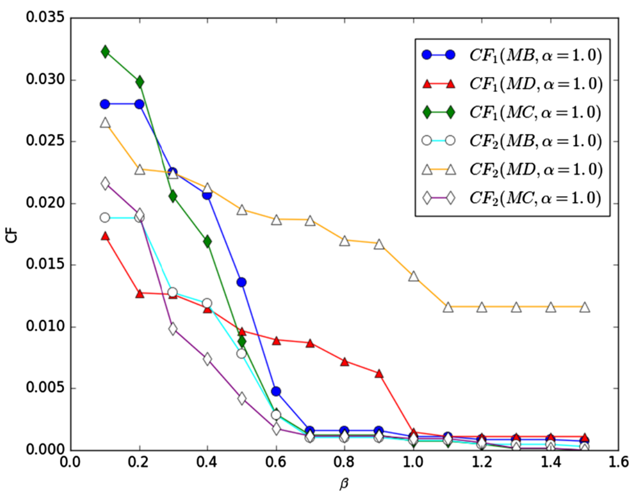

Fig. 5. Impact of the cascading failures when $\alpha = 1.0$.

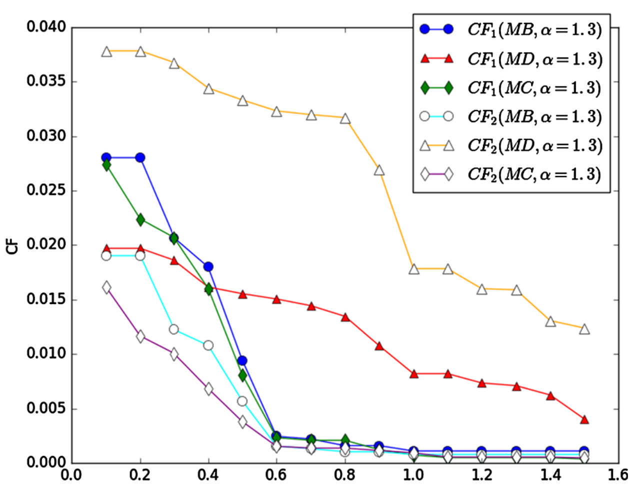

Fig. 6. Impact of the cascading failures on a project when $\alpha = 1.3$.

causes the most damage when $0.9 \leq \beta \leq 1.5$; (3) on average, the difference between the CF_1 and CF_2 of MB and MC is smaller than the variation between the CF_1 and CF_2 of MD.

From Fig. 5, there are the following observations: (1) $\beta_c = 1.5$ for MC; nevertheless, the cascading failures resulted from the MB and MD attack cannot be avoided even if $\beta = 1.5$. The CF of MB and MC show similar tendencies; they first sharply decline then fall more gently. However, a rising β has little effect on preventing MD-induced cascading failures; (2) The damage degree order is MC > *MB* > *MD* when $\beta \leq 0.2$, whereas the largest cascades are more possibly to be initiated by MD when $0.6 \leq \beta \leq 1.5$; (3) With an increase in β, the difference between the CF_1 and CF_2 of MB and MC becomes increasingly smaller; however, there is little variation between the CF_1 and CF_2 of MD.

From Fig. 6, there are the following observations: (1) Even when the load capacity is 2.5 times the initial load, the three failure types can trigger cascading failures, and any increase in β has a negligible effect on restraining MD-induced cascading failures; (2) The largest cascades are induced by MD when $0.5 \leq \beta \leq 1.5$, and the largest cascades size are more likely to be triggered by MB when $\beta \leq 0.4$; (3) Overall, the difference between the CF_1 and CF_2 in these three failure types have similar tendencies, with the difference becoming increasingly smaller.

The CF_1 and CF_2 of MB, MD and MC were compared for four cases; $\alpha = 0.4$, $\alpha = 0.7$, $\alpha = 1.0$, and $\alpha = 1.3$; from which it was found:

(1) With an increase in task capacity over a certain range, the abilities of the MB, MD, and MC tasks to mitigate cascading failures can be significantly improved.
(2) There may be a threshold (β_c) for task capacity, if the task capacity is higher than the threshold, the failed tasks can be restored and cascading failures avoided.
(3) As the task load rises, the strengthened task capacity may gradually lose its effectiveness in mitigating cascading failures; that is, as the task load increases, it becomes increasingly difficult to reduce cascading failures by strengthening task capacity.
(4) The CF_1 and CF_2 for these three failure types vary significantly; the CF_2 of the MD is generally much higher than the CF_1 of the MD, and the CF_2 of the MB and MC is somewhat smaller than the CF_1 of the MB and MC; that is, the broken tasks under an MD attack often carry a larger weight, which amplifies the impact of MD failure; however, for MB and MC, the reverse occurs.

5. Discussion

In this section, three alternative protective countermeasures for mitigating cascading failure risk are presented based on the experimental results. Several limitations and possible future work are also discussed.

5.1. Practical implications

The above results indicated that a single task failure was sufficient to cause cascading failures in many downstream tasks. Therefore, it is vital to develop preventative measures to mitigate the impact of such cascading failures. The simulation results suggested several preventative measures for managers to better cope with cascading failure risk and improve project performance.

(1) Improve task performance by establishing proper task capacity. Each task is assigned a certain capacity to fulfill the project workflow. To better withstand possible disturbances in the project, the allocated task capacity is normally greater than the needed task load. The experimental results indicate that the destruction caused by cascading failures varies depending on task capacity; therefore, if task capacity is set to be within a certain range, this could reduce the impact of task failure. Managers could then restrain cascading failure risk within a certain range by setting an appropriate capacity for each task. However, task capacity is limited by the cost; that is, the higher the capacity the higher the cost; therefore, assigning a proper value to each task can not only assist in resisting cascading failures in project but can also optimize the costs of maintaining the capacity.

(2) Conduct different protective measures for different types of task failures. The simulation results suggested that different types of task failures have noticeably different effects on the ability to initiate cascading failures. Therefore, managers can have targeted approaches to suppress cascading failures: (1) the failure of tasks with higher betweenness or higher closeness is more possibly to trigger the largest cascade size when task capacity is relatively low; in turn, the protection of these tasks should be prioritized; (2) for tasks with higher betweenness or higher closeness, there is a sharp decline in cascading failures with an increase in task capacity; thus adjusting the task capacity within this range could efficiently lower the impact of these two types of task failures on the project and avert cascade-failure-induced disasters; (3) for tasks with a higher degree, it is difficult to reduce cascading failures by enhancing their capacity; therefore, multiple methods should be developed to mitigate the risk of cascading failure that originating from this type of task failure, such as immediately isolating the failed task from the other tasks, preparing alternatives for the possible failed tasks in advance, and avoiding a node (task) with a large number of node adjacencies in the design phase.

(3) Comprehensively evaluate the impact of cascading failures on the project. This paper examined two evaluation metrics; the first metric was the normalized avalanche size, and the second metric takes account of both the significance of the tasks and the avalanche size. Although these two metrics are scientific and exhibit similar tendencies, the results of the two metrics varied significantly; for instance, the failure of tasks with a higher degree tended to affect a smaller number of other tasks; however, when the significance of the failed tasks was considered, the possible impact was found to be amplified. Therefore, to obtain relatively objective evaluation results, managers may need to evaluate the impact of cascading failures from multiple aspects.

5.2. Limitations and future work

To better comprehend the cascading failures in projects, several issues deserve further study. First, in this paper, the focus was on cascading failures induced by a single project task failure. However, multiple task failures may occur concurrently, which may significantly affect overall project performance. Due to the high interactions between tasks, the failure propagation triggered by multiple failed tasks may differ from single task failure. Therefore, a better understanding of the cascading failures induced by multiple task failures is needed to further enrich project systemic risk research.

Second, several assumptions were made when constructing the cascade model. Although these assumptions were reasonable, they inevitably limited the possibilities. For example, it was assumed that all task capacities were linearly correlated with their respective loads and other probabilities were not considered, which indicated that there is no unified cascade model. To capture the project failure propagation in specific conditions, the cascade model needs to be modified in accordance with each scenario. Therefore, another productive direction for studying cascading failures in projects would be to tailor cascade models to concrete contexts.

Although a few researchers have already started conducting analyses of cascading failures in project, cascading failures in programs have been less studied. As programs are becoming more common in new capital formation and are vulnerable to large-scale failures, a vital future research direction is exploring cascading failures in programs. Studying cascading failures in programs can also contribute to reducing cascading failure risk and improving program performance. A program contains two or more individual projects and there is a close cooperation between these projects (Rijke et al., 2014). Therefore, the cascading failure process in programs is different from the failure propagation process in project as they include both intra-network cascading and inter-network cascading.

6. Conclusions

Because the high coupling among tasks and multiple failure propagation paths from one task to another task, a single task failure can induce a cascade of downstream task failures and thus give rise to significant project failures. Although various exogenous risk sources have been investigated in past project risk studies, the cascading failure risk resulting from trivial endogenous disturbances is still a relatively immature research field. To ensure smooth project implementation, this paper investigated cascading failures triggered by endogenous project task failures. A cascade model was tailored to examine cascading failures induced by a single task failure. Under three types of single task failures with varying parameters, numerical experiments were then conducted to quantify the impact of cascading failures on the projects. The simulation results suggested several preventative measures for managers to better cope with cascading failure risk.

This paper makes three primary contributions. First, the self-amplification process associated with cascading failures in projects was modeled using an extended flow redistribution model, which allowed for a replication of the failure propagation process that originated from a single task failure, and an assessment of the influence of cascading failures on the project operations. Second, the cascading failures under different types of single task failures were compared, with the simulation results indicating that the different task failure types had noticeably different capacities to trigger cascading failures. Therefore, this study adds to the understanding of the possible correlations between structural properties and cascading failures and the types of tasks that may induce large cascades. Third, targeted protection measures were proposed to restrain and/or mitigate cascading failures risk. As single task failures may cause failures in many downstream tasks due to cascading failures, these proposed preventative measures for different task failure types can give guidance to managers to mitigate cascading failure risk, avoid project catastrophes, and improve project performance.

Acknowledgements

We are particularly grateful to Dr. Christos Ellinas for providing the dataset. We thank the four anonymous referees for their insightful comments. This work was supported by the National Natural Science Foundation of China [grant numbers 71672145, 71802003, 71702149, 71402142] and the Provincial Natural Sciences Basic Research Plan in Shaanxi, China [grant numbers 2017JQ7011, 2018JM7002].

References

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics Reports, 424*(4), 175–308.
Chapman, C., & Ward, S. (2003). *Project risk management: Processes, techniques, and insights.* Wiley.
Costa, L. F., Rodrigues, F. A., Travieso, G., & Boas, P. R. V. (2005). Characterization of complex networks: A survey of measurements. *Advances in Physics,* 167–242.
Crucitti, P., Latora, V., & Marchiori, M. (2004). Model for cascading failures in complex networks. *Physical Review E Statistical Nonlinear & Soft Matter Physics, 69*(4 Pt 2), 045104.
Dobson, I. (2012). Estimating the propagation and extent of cascading line outages from utility data with a branching process. *IEEE Transactions on Power Systems, 27*(4), 2146–2155.
Ellinas, C. (2018b). Modelling indirect interactions during failure spreading in a project activity network. *Scientific Reports,* 8.
Ellinas, C. (2018a). The domino effect: An empirical exposition of systemic risk across project networks. *Production and Operations Management*.
Ellinas, C., Allan, N., Durugbo, C., & Johansson, A. (2015). How robust is your project? From local failures to global catastrophes: A complex networks approach to project systemic risk. *PLoS One, 10*(11), e0142469.
Ellinas, C., Allan, N., & Johansson, A. (2016). Project systemic risk: Application examples of a network model. *International Journal of Production Economics, 182*, 50–62.
Fang, X., Yang, Q., & Yan, W. (2014). Modeling and analysis of cascading failure in directed complex networks. *Safety Science, 65*, 1–9.
Goh, K.-I., Kahng, B., & Kim, D. (2001). Universal behavior of load distribution in scale-free networks. *Physical Review Letters, 87*(27), 278701.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology, 83*(6), 1420–1443.

Harris, T. E. (2002). *The theory of branching processes.* Courier Corporation.

Hasan, S., & Ukkusuri, S. V. (2011). A threshold model of social contagion process for evacuation decision making. *Transportation Research Part B, 45*(10), 1590–1605.

Huang, X., Vodenska, I., Havlin, S., & Stanley, H. E. (2013). Cascading failures in bi-partite graphs: Model for systemic risk propagation. *Scientific Reports, 3*, 1219.

Kim, J., & Dobson, I. (2010). Approximating a loading-dependent cascading failure model with a branching process. *IEEE Transactions on Reliability, 59*(4), 691–699.

Kinney, R., Crucitti, P., Albert, R., & Latora, V. (2005). Modeling cascading failures in the North American power grid. *The European Physical Journal B-Condensed Matter and Complex Systems, 46*(1), 101–107.

Koç, Y., Warnier, M., Kooij, R. E., & Brazier, F. M. (2013). An entropy-based metric to quantify the robustness of power grids against cascading failures. *Safety Science, 59*, 126–134.

Mirzasoleiman, B., Babaei, M., Jalili, M., & Safari, M. (2011). Cascaded failures in weighted networks. *Physical Review E, 84*(4), 046114.

Mohammadipour, F., & Sadjadi, S. J. (2016). Project cost–quality–risk tradeoff analysis in a time-constrained problem. *Computers & Industrial Engineering, 95*, 111–121.

Motter, A.E. & Yang, Y. (2017). The unfolding and control of network cascades. arXiv preprint arXiv:1701.00578.

Motter, A. E., & Lai, Y.-C. (2002). Cascade-based attacks on complex networks. *Physical Review E, 66*(6), 065102.

Muriana, C., & Vizzini, G. (2017). Project risk management: A deterministic quantitative technique for assessment and mitigation. *International Journal of Project Management, 35*(3), 320–340.

Newman, M. E. (2008). The mathematics of networks. *The New Palgrave Encyclopedia of Economics, 2*(2008), 1–12.

Park, K., Lai, Y. C., & Ye, N. (2004). Characterization of weighted complex networks. *Physical Review E Statistical Nonlinear & Soft Matter Physics, 70*(2), 026109.

Qazi, A., Quigley, J., Dickson, A., & Kirytopoulos, K. (2016). Project complexity and risk management (ProCRiM): Towards modelling project complexity driven risk paths in construction projects. *International Journal of Project Management, 34*(7), 1183–1198.

Rijke, J., van Herk, S., Zevenbergen, C., Ashley, R., Hertogh, M., & ten Heuvelhof, E. (2014). Adaptive programme management through a balanced performance/strategy oriented focus. *International Journal of Project Management, 32*(7), 1197–1209.

Tabrizi, B. H., & Ghaderi, S. F. (2016). A robust bi-objective model for concurrent planning of project scheduling and material procurement. *Computers & Industrial Engineering, 98*, 11–29.

Tang, L., Jing, K., He, J., & Stanley, H. E. (2016). Complex interdependent supply chain networks: Cascading failure and robustness. *Physica A: Statistical Mechanics and its Applications, 443*, 58–69.

Thamhain, H. (2013). Managing risks in complex projects. *Project Management Journal, 44*(2), 20–35.

Turner, R. J., Huemann, M., Anbari, F. T., & Bredillet, C. N. (2010). Perspectives on projects. *Construction Management & Economics, 30*(30), 416–420.

Van Eeten, M., Nieuwenhuijs, A., Luiijf, E., Klaver, M., & Cruz, E. (2011). The state and the threat of cascading failure across critical infrastructures: The implications of empirical evidence from media incident reports. *Public Administration, 89*(2), 381–400.

Wang, J. (2013). Mitigation strategies on scale-free networks against cascading failures. *Physica A: Statistical Mechanics and its Applications, 392*(9), 2257–2264.

Wang, J.-W., & Rong, L.-L. (2011). Robustness of the western United States power grid under edge attack strategies due to cascading failures. *Safety Science, 49*(6), 807–812.

Wang, J., Yang, N., Zhang, Y., & Song, Y. (2018). Development of the mitigation strategy against the schedule risks of the R&D project through controlling the cascading failure of the R&D network. *Physica A Statistical Mechanics & Its Applications,* 508.

Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences of the United States of America, 99*(9), 5766.

Wu, Z.-X., Peng, G., Wang, W.-X., Chan, S., & Wong, E. W.-M. (2008). Cascading failure spreading on weighted heterogeneous networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(05), P05013.

Zeng, J., An, M., & Smith, N. J. (2007). Application of a fuzzy based decision making methodology to construction project risk assessment. *International Journal of Project Management, 25*(6), 589–600.

Zeng, Y., & Xiao, R. (2014). Modelling of cluster supply network with cascading failure spread and its vulnerability analysis. *International Journal of Production Research, 52*(23), 6938–6953.