# LETTER • OPEN ACCESS

# Neglecting complex network structures underestimates delays in a large-capital project

To cite this article: C Ellinas et al 2023 J. Phys. Complex. 4 02LT01

View the article online for updates and enhancements.

# You may also like

- <u>Transfer operators on graphs: spectral</u> <u>clustering and beyond</u> Stefan Klus and Maia Trower
- <u>Hyper-diffusion on multiplex networks</u> Reza Ghorbanchian, Vito Latora and Ginestra Bianconi
- <u>A controlled transfer entropy approach to</u> detect asymmetric interactions in heterogeneous systems

Rishita Das and Maurizio Porfiri

# Journal of Physics: Complexity

# LETTER

#### **OPEN ACCESS**

CrossMark

RECEIVED 12 July 2022

REVISED

16 March 2023

ACCEPTED FOR PUBLICATION 20 April 2023

PUBLISHED

9 May 2023

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Neglecting complex network structures underestimates delays in a large-capital project

C Ellinas<sup>1,\*</sup>, D Avraam<sup>2</sup> and C Nicolaides<sup>3,4</sup>

<sup>1</sup> Nodes & Links, Station Road, Cambridge CB1 2LA, United Kingdom

Population Health Sciences Institute, Newcastle University, Newcastle, United Kingdom

<sup>3</sup> School of Economics and Management, University of Cyprus, Nicosia, Cyprus

- Initiative on the Digital Economy, Massachusetts Institute of Technology, Cambridge, MA, United States of America
- \* Author to whom any correspondence should be addressed.

E-mail: christos@nodeslinks.com

Keywords: complex networks, spreading, delay propagation, project delay

#### Abstract

Completing large-scale projects on time is a daunting challenge, partly due to the intricate network of dependencies between a project's activities. To support this challenge, existing theory focuses on predicting whether a delay in completing a single activity is likely to spread and impact downstream activities. Using fine-grained information from 68 546 activities and 84 934 pairs, associated with the delivery of a \$1.86Bn infrastructure project, we show that the core mechanism that underpins existing theory underestimates delay propagation. To elucidate the mechanisms that drive delay, we generated null models that destroyed the structural and temporal correlations of the original project activity network. By doing so, we argue that this underestimation is the result of neglecting endogenous structural features within the project's activity network. Formulating a new mechanism that utilizes both temporal and structural features may help improve our capacity to predict how delays spread within projects.

# 1. Introduction

Successful delivery of project-based operations is central to the global economy [1]. World Bank data indicates that more than 22% of the world's gross domestic product—equivalent to approximately \$48 trillion—relies on the successful completion of projects. These projects are composed of thousands of activities, planned over a well-defined timeline. The intricacy of the interdependencies between these activities challenges the ability of decision makers to deliver large scale projects on time [2, 3]. As a result, these projects regularly experience major cost escalations. For example, a survey of 258 infrastructure projects taking place across 20 countries shows an average cost escalation of 28% [4]. The fiscal implications of such escalations grows further due to the projected average annual growth of 1.5%–2.5% in project value [5].

Delay in completing a single activity can impact subsequent, downstream activities due to the functional dependencies that exist between them. As a result, a seemingly insignificant delay in the completion of a parent activity can trigger a cascade of delays to its downstream child activities [6]. The time between the completion date of the parent activity and the start date of its child activity is the perceived centerpiece in predicting whether a delay in completing the parent will impact the child activity. This time period is known as the 'free-float'.

Given sufficient 'free-float', existing theory postulates that a delay in the completion date of the parent activity will be absorbed without impacting the start date of its child activity. In other words, a delay in the parent activity will impact its child only if the 'free-floa' between the parent and the child is smaller than the delay in completing the parent. This 'free-float' mechanism is central to most delay spread prediction models, for example Critical Path Method, Program Evaluation and Review Technique [7]. This mechanism is also similar to the ones used to model delay spreads in other systems, namely railway [8] and airport networks [9].





We use project data to test the accuracy of the predictions that stem from existing theory [10]. We argue that this 'free-float' mechanism neglects endogenous structural aspects of the project's activity network, and as a result systematically underestimates the probability of a delay to spread. We provide empirical evidence of this using a large, novel dataset that contains fine grained information of 68 546 unique activities and 84 934 unique pairs that underpin the completion of a \$1.86Bn infrastructure project. Each activity is characterized by a planned and an actual start and completion date, which determines whether it has been 'delayed' or completed 'on-time'. We consider an activity to be 'delayed' ('on-time') if the actual completion date is later (earlier or the same) than its planned completion date. By knowing which activity is 'on-time' or 'delayed, we can spot the origin of a delay, and track its potential spread to downstream activities, see figure 1 [7].

#### 2. Materials and methods

#### 2.1. Network sample

Our original dataset is composed of 315 268 unique activity and 546 380 unique directed links. We created a subset of 74 731 activities and 111 550 links, for which we have complete information on their planned and actual start/completion dates. From this subset, we focused on the links that reflect functional dependencies between activity pairs i.e. where the output of a parent is a necessary input to start the child activity. This is to exclude links that may signal partial dependencies which could distort the trueness of the dependency structure (e.g. parent activity can be completed in part, for its child activity to also start). As a result, we ended up with our final dataset, which is composed of 68 546 unique activities and 84 934 unique links. We used this information to generate the project's activity network.

This activity network is a directed, acyclic graph (DAG). This is the result of the temporal ordering enforced by the nodes, where a link from a child node cannot link back to an activity that has a start date earlier than that child's completion date. The network forms a single weakly connected component, and has a heavy-tail degree distribution, see figure 2. The mean out-degree is 1.52 whilst its max out-degree is 63. The mean in-degree is 1.52 and the max in-degree is 50. The resulting activity network is also sparse, with an edge density of  $6.2 \times 10^{-6}$ , and has an average path length of 16.7. These statistics are consistent with those reported in [6].

For our analysis, we focused on single-parent/single-child pairs in this activity network. In doing so, we ensure that our results are not affected by compounding spreading dynamics that underpin more complicated network structures. In the case of the actual and expected fractions of delayed parents, this is composed of 53 519 unique activities and 35 380 unique links, with 15 829 delayed parents involved in 8860 unique links.

#### 2.2. Expected delayed spread

Focusing on the subset of pairs where a parent is delayed (N = 8860), we evaluated the expectation of a child also being delayed by comparing the actual completion date of the parent activity with the planned start date of its child. If the actual completion date of the parent is later than the planned start date of the child, the 'free-floa' between them has been used and therefore, we expect that the child activity would also be delayed.





Figure 2. Probability that the degree is larger than, or equal to, a specified value. Both axes are in log scale, where a straight line is indicative of a heavy tail.



**Figure 3.** Fraction of delayed child activities, given that their parent has been delayed, as a function of the planned 'free-float'. Red line corresponds to the actual fraction of delayed child activities, where the actual completion date for each child is larger than its planned completion date. Black line corresponds to the expected fraction of delayed child activities, whose planned start date is earlier than the actual completion date of their delayed parent. This fraction corresponds to the outcome that we would expect using the 'free-float' mechanism. Both planned and actual dates used are found within the data. Dotted grey lines correspond to the average  $(\pm 2\sigma)$  fraction of delayed childs under Null Model 1 and 2.

Otherwise, if free-float still exists between this pair, the child activity should not be delayed. In other words, if the parent has been delayed by more days than the free float between the parent and the child, we *expect* the child to be also delayed. The outcome of this computation corresponds to figure 3, expected delays line. Note that we use the term 'expected' to signal that if actual delays were identical to the expected delays (in terms of probability of them happening and their magnitude), then the 'free-float' mechanism can accurately predict how delays propagate.



#### 2.3. Null models

With Null Model 1, we destroyed both structural and 'free-float' correlations by reshuffling all links within the activity network, in accordance with the configuration model. We first recorded the in/out degree of each activity, and represent those as half-links [11]. We then randomly selected a pair of activities and connected them, starting from the end of the project and moving progressively backwards in time. Starting from the project end, we pick the activity with the latest start date—this is our candidate parent. We then find all the activities that have a start date that is larger, or equal, to that of the completion date of that candidate parent. All activities that satisfy this condition form a set of candidate childs. We randomly pick an activity from the set of candidate childs, and create a link from the candidate parent to this child—we now have a new parent/child pair. We remove the two nodes from the pool of nodes, and repeat until no nodes are left.

This process ensures ensure that for each newly formed parent/child pair satisfied the requirement of the parent having a planned completion date *before* the planned start date of its child activity. If this was not the case, we would end up with pairs where the child could have a start date that is earlier than the completion date of its parent, which goes against the definition of a link in an activity network.

We repeated the process until all half-links are paired up. In a few cases, we missed a minor amount of links (worst case contains 0.09% less links than the original) due to the lack of pair combinations where a child's start date is after the parents completion date. Depending on the order in which half-links are paired, the resulting network of each reshuffle is different whilst preserving the in/out degree of each activity. With Null Model 2, we maintained the structure of the activity network and destroy the 'free-float' correlations by randomly reassigning free floats across all pairs. As with Null Model 1, the stochastic nature of the switch results in different temporal correlations for each reshuffled network.

We created 100 reshuffled networks for each Null Model. For each reshuffled network, and once the reshuffled process was completed, we compared the actual completion date of a delayed parent with the planned start date of its child. If the actual completion date was larger than the planned start date of the child, then we would mark the child as delayed. We repeated this process across all parent-child pairs to get the fraction of delayed childs. We repeated this process all 100 instances and averaged the fraction of delayed childs for each Null Model, as shown in figure 3 (standard deviation across 100 instances for each of the Null Models is also shown in figure 3).

#### 3. Results

We first confirmed that a delay in a parent activity can spread to its child activity, and that the probability of doing so reduces as 'free-float' increases. This observation is in step with existing theory, where 'free-float' is the determining quantity in evaluating whether a delay in completing a parent activity can impact its downstream activities.

We found that this negative correlation between 'free-float' and probability of delay spreading vanishes once structural and 'free-float' correlations are destroyed. This means that the importance of the 'free-float' mechanism relies on a combination of temporal and structural correlations within the underlying activity network. We employed two null models that allowed us to selectively destroy these correlations [12]. With Null Model 1, we destroyed both structural and 'free-float' correlations by randomly rewiring all activities (N = 68546) according to the network configuration model [11]. This corresponds to the bulk of tasks having a similar number of dependencies (in reality, some tasks have many more dependencies than the average). With Null Model 2, we randomly reshuffled the 'free-float' entries across all activity pairs (N = 84934). This corresponds to tasks being planned without any consideration to operational and financial constraints (in reality, 'free-float' is often used to smoothen spending across the project's life cycle).

We next focused on the 'free-float' mechanism by which delay is currently thought to be predicated on, and tested whether this mechanism can be used to effectively predict delay spreading. We compare the actual delay spreading patterns with the expected delay spreading patterns (figure 3 across all single-parent/single-child unique pairs ( $N = 35\,380$ ). We focus on single-parent/single-child pairs to ensure that our results are independent from compounding spreading dynamics that may underpin more complicated structures [13, 14] (i.e. in such cases, the state of an activity may depend on the state of more than one parent and/or child activities).

We find that the delay expectation from using the 'free-float' mechanism consistently underestimates the probability of a delay spreading from a parent to a child activity. This means that it is possible for a delay to propagate from a delayed parent to its child even when there is still some 'free-float' between them—a possibility that is unaccounted for within existing theory. This may be due to the incomplete formulation of the 'free-float' mechanism in current theory. Specifically, we noted that the relationship between 'free-float' and probability of a delay to spread is a function of both temporal and structural features (figure 3; Null Models 1 and 2). Yet the 'free-float' mechanism utilizes temporal features only. Formulating a new



mechanism that characterizes delay spread by taking into account both temporal and structural features could improve the accuracy of predicting whether a delay can spread within any project, leading to significant cost savings.

### 4. Conclusion

Our results indicate that the probability of a delay spreading within the project is related to the length of the 'free-float', in step with existing theory. We showed that this relationship may be due to a combination of the structural and temporal features that underpin activity networks. We show that existing delay prediction methods which currently rely on the 'free-float' mechanism systematically underestimate the actual delay spread. This may be due to the existing 'free float' mechanism focusing solely on temporal features, despite the evidence above that suggests that the importance of 'free-float' is due to a combination of both structural and temporal features.

### Data availability statement

Data is secured under an NDA. Subsets and/or meta data can be made available to other parties (though may require an NDA). The data that support the findings of this study are available upon reasonable request from the authors.

### Acknowledgment

We thank Marc Santolini and Konstantinos Sakellariou for useful comments.

### Funding

This work was partly supported by European Union's Horizon 2020 and the Cyprus Research & Innovation Foundation under the SEED program (Grant Agreement 0719(B)/0124). Nodes & Links Ltd provided support in the form of salary for CE, but did not have any additional role in the conceptualization of the study, analysis, decision to publish, or preparation of the manuscript. CN received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 786247

# ORCID iD

C Ellinas b https://orcid.org/0000-0001-5146-6554

#### References

- [1] Jensen A, Thuesen C and Geraldi J 2016 Proj. Manage. J. 47 21-34
- [2] Jacobs M and Swink M 2011 J. Oper. Manage. 29 677-91
- [3] Sosa M, Mihm J and Browning T 2011 J. Mech. Des. 133 101008
- [4] Flyvbjerg B, Holm M and BUHL L 2004 Transp. Rev. 24 3-18
- [5] Flyvbjerg B 2014 Proj. Manage. J. 45 6-19
- [6] Santolini M, Ellinas C and Nicolaides C 2021 EPJ Data Sci. 10 10-36
- [7] Elmaghraby S 1995 Eur. J. Oper. Res. 82 383-408
- [8] Monechi B, Gravino P, Clemente D and Servedio V 2018 EPJ Data Sci. 7 35
- [9] Fleurquin P, Ramasco J and Eguiluz V 2013 Sci. Rep. 3 1159
- [10] Watts D 2017 Nat. Hum. Behav. 1 1-5
- [11] Molloy M and Reed B 1995 Random Struct. Algorithms 6 161-80
- [12] Karsai M, Kivelä M, Pan R K, Kaski K, Kertész J, Barabási A-L and Saramäki J 2011 Phys. Rev. E 83 025102
- [13] Centola D, Eguíluz V and Macy M 2007 Cascade dynamics of complex propagation Phys. A: Stat. Mech. 374 449-56
- [14] Ferraz G, Rodrigues F and Moreno Y 2018 Phys. Rep. 756 1–59