Santolini et al. *EPJ Data Science* (2021) 10:36
https://doi.org/10.1140/epjds/s13688-021-00291-w

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

Uncovering the fragility of large-scale engineering projects

Marc Santolini[1,2*], Christos Ellinas[3*] and Christos Nicolaides[4,5]

*Correspondence:
marc.santolini@cri-paris.org;
christos@nodeslinks.com
[1] Université de Paris, INSERM U1284, Center for Research and Interdisciplinarity (CRI), F-75006, Paris, France
[3] Nodes & Links Ltd, Station Road, Cambridge, CB1 2LA, UK
Full list of author information is available at the end of the article

Abstract

Engineering projects are notoriously hard to complete on-time, with project delays often theorised to propagate across interdependent activities. Here, we use a novel dataset consisting of activity networks from 14 diverse, large-scale engineering projects to uncover network properties that impact timely project completion. We provide empirical evidence of perturbation cascades, where perturbations in the delivery of a single activity can impact the delivery of up to 4 activities downstream, leading to large perturbation cascades. We further show that perturbation clustering significantly affects project overall delays. Finally, we find that poorly performing projects have their highest perturbations in high reach nodes, which can lead to largest cascades, while well performing projects have perturbations in low reach nodes, resulting in localised cascades. Altogether, these findings pave the way for a network-science framework that can materially enhance the delivery of large-scale engineering projects.

Keywords: Activity networks; Network science; Spreading processes; Cascades; Project performance

1 Introduction

Timely delivery of construction projects is notoriously challenging, with cost and duration escalations being typical across the entire industry. An influential 2003 paper captures the scale of the challenge: almost 9 out of 10 construction projects from 258 companies across 20 countries and 5 continents experienced cost overruns (average cost overrun of 28%) [1]. Follow up work focused on 44 construction projects in North America and Europe, reporting an average construction cost overrun of 45%; for a quarter of the projects cost overruns were at least 60% [2]. Considering the fact that project budgets are growing at an annual rate of 1.5%–2.5% [3], such escalations are bound to increase even further.

Poor project performance is unlikely to be the result of bad practice, since the relationship between widely recognised variables that impact performance has long been researched and acted upon (e.g., how uncertainty in the duration of project's activities impacts the overall project delivery time) [4]. To explain this disparity between theory and practice, recent work in both academia [5–10] and industry [11, 12] has proposed a new, independent variable that impacts project performance: project complexity.

Project complexity largely stems from the networked nature of the project [9, 10, 13, 14], where dependencies between a project's activities create pathways for perturbations to propagate through. In this case, a perturbation refers to the deviation of completing an activity from the expected plan, either earlier or later. Perturbation pathways can be explicitly expressed through the project's activity network, where nodes correspond to activities that need to be completed in order to complete the project. A directed link between two nodes corresponds to a functional dependency between the two activities. For example, a directed link from node i to node j indicates that activity i must be completed before activity j begins. Paths contained within the network correspond to contractually-agreed sequences of activities, reflecting the fixed nature of the network. This is due to functional constraints that underpin the associated work (e.g., a wall cannot be built before its foundation, and foundations cannot be built before the area is excavated) and legally binding costs that have been agreed prior to starting said work.

The activity network can be used to better understand the mechanisms that drive poor project performance, and eventually uncover ways to control it. For instance, the networked nature of the project activities highlights the potential for minor, local events—like a delay in completing an activity—to propagate through the activity network, delaying more downstream activities, and eventually, delaying the entire project [13]. This behaviour is qualitatively similar to propagation effects observed across a range of complex systems, where the underlying network controls the propensity of spreading events to take place [14] and consequently the system's broader fragility [15, 16] (e.g., sparse connectivity [17], node degree [18], community structure [19], centrality [20, 21] etc.). Such spreading phenomena have been extensively studied in biological systems [22, 23], where the clustering of perturbations lead to 'disease modules' underlying complex pathologies [24, 25]. Transportation networks [26] are also worth mentioning due the interplay between network structure and temporal properties, where time buffers are introduced by design in order to contain perturbations spreading (e.g., air traffic networks [27], where nodes are airports and directed links are flights, or rail systems [28], where nodes are railway stations and directed links are scheduled trips).

In the context of projects, and though theoretically plausible [13, 29, 30], there has been little empirical evidence to support the hypothesis of such cascades taking place within activity networks, beyond anecdotal observations within real-world projects [7, 31–33]. As a result, there has been limited adoption of network science tools and techniques to better understand project complexity in general, and propagation effects [16] within activity networks specifically. This lack of empirical evidence has reinforced the prominence of optimisation-based techniques in identifying activities prone to such perturbations using time based constraints (i.e., interpreting them as a form of resource constraints and expressed as a scheduling problem [4]; e.g., Critical Path Method [34], Program Evaluation Review Technique [35, 36]). Though articulate, these methods rely on linear operations [37] that forbid non-linear effects in terms of the impact that a single perturbation can have. For example, if an activity is delayed by x days, and assuming it lies on the critical path, the project will also be delayed by a maximum of x days. Alas, this linearity contrasts real-world evidence of non-linear instances, where a minor delay can have a disproportionate effect on the project [7, 31–33].

This work is a first attempt to provide empirical evidence of propagation events within an important class of sociotechnical systems—large-scale, engineering projects—and present a link between the structure of their underlying activity networks with the overall project performance. We use a novel dataset that contains fine-grained information from 14 large-scale, engineering projects. Using planned and actual activity duration, we show that large-scale perturbation cascades exist within the entire dataset. These cascades are structurally similar across projects and tend to propagate across: a perturbation in a single task can impact a large number of activities, and exert an influence downstream, up to 4 activities. We then show that the cascade size distribution follows a power-law whose exponent is a good predictor of the overall project performance (Spearman's $\rho = -0.68$, $p = 0.0089$), with extensive cascade sizes being an indicator of poor overall project performance. Finally, we show that large spreading events occur when the largest perturbations hit 'fragile' nodes with a large reach, i.e., number of downstream nodes. This paves the way for future work on implementing strategies to detect and protect such fragile nodes to minimize undesired large cascading events.

2 Results

Each of the 14 projects in our dataset (Fig. 1(a)) contains information about *a priori* (planned) and *a posteriori* (actual) activity duration (see Additional file 1, Figure S1). For each node, we define the *activity perturbation* of each node as the difference between actual and planned activity duration (measured in days). As such, perturbations correspond to deviations from the initial schedule. To quantify poor project performance, we use the positive perturbation rate or 'delay rate': that is, the proportion of activities that have endured a delay compared to the initial schedule. Assuming no knowledge about the dependencies within activities, one would expect that projects with more deliverables or higher duration would be more vulnerable to perturbations, since more things can go wrong and they are exposed to risks for longer, respectively. Contrary to this expectation, we find that project performance does not correlate significantly with the total number of activities (Figure S2a, $\rho = -0.52$, $p = 0.062$) or the cumulative baseline duration of all activities (Figure S2b, $\rho = -0.39$, $p = 0.17$). As such, the total size and total duration of a project are not informative about the overall vulnerability of the project to endure activity delays. These results prompt us to investigate whether project complexity, embedded in its activity network, can account for this unexplained variation and help predict the occurrence, magnitude, and rate of activity perturbations.

2.1 Clustering of perturbations in activity networks

Each project can be represented as a directed activity network reflecting the dependence structure of a project's activities (Fig. 1(a)). The 14 activity networks contain a very limited number of cycles (Figure S3a), which allows us to safely assume a local tree-like structure. The networks have vastly different sizes, quantified by the number of activities they are composed of (Figure S3b and Table 1), ranging from 282 to 29,080 activities. Accordingly, their global structure varies widely, and the longest path (or 'network diameter') ranges from 31 to 191 activities. Despite these differences in size, these networks exhibit shared properties: they are sparse (densities span $10^{-3} - 10^{-5}$), and while their average path lengths are similar to random expectation, they are in general more highly clustered than expected by chance (10 out of 14 networks, Figure S3c,d). As such, activity networks are 'small-world' (Table 1)—a finding in step with prior work on project networks reported in [38, 39]. Finally, their local structure, assessed through the variation of number of dependent activities or 'degree' of an activity, is strikingly similar: we observe that in 78% of the

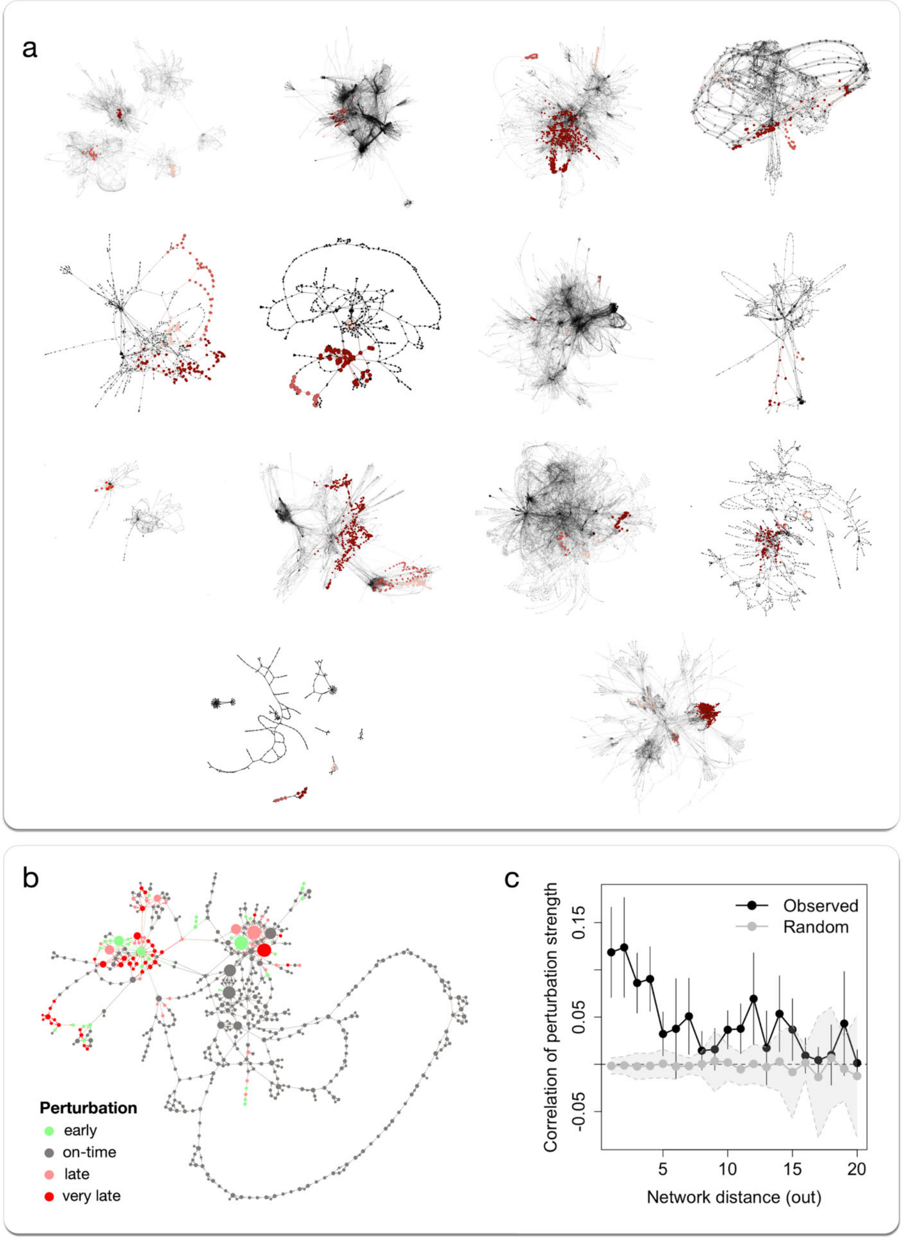

Figure 1 Perturbation clustering in activity networks. (**a**) Activity networks of all projects (project 1 top-left to project 14 bottom-right). Node size denotes out-degree. The top 3 cascades (i.e connected components of perturbed activities) in each network are shown with a red color (from dark red for the top cascade to light red for the 3rd cascade). (**b**) Activity network from Project 6. Node color indicates the type of perturbation: early for negative perturbation, on-time if there is no perturbation, late for a positive perturbation, and very late for delays larger than 30 days. We observe a clustering of perturbations within network neighborhoods. (**c**) Extent of the observed perturbations, measured by the correlation between absolute perturbation values of activities as a function of their network distance. Network distance is computed as the outgoing shortest path between two nodes in the directed network. In order to model random expectation, for each project we compute the average correlation values across 50 random controls obtained by shuffling perturbations across completed activities. The gray area corresponds to the average and 2 standard deviations of these values across the 14 projects (see Methods)

Table 1 Descriptive statistics of the 14 studied activity networks

	Number of nodes	Number of links	Density	Clustering	Average path length	Average degree	Max degree	Average reach	Max reach	Delay rate
project 1	10,734	15,524	1.4e-04	9.0e-02	16.7	2.9	106	279	2846	0.165
project 2	35,618	61,199	4.8e-05	2.7e-03	14.3	3.4	1887	869	14,848	0.235
project 3	17,160	25,790	8.8e-05	5.7e-02	28.6	3.0	231	1830	7909	0.177
project 4	2458	5525	9.2e-04	5.3e-02	12.5	4.5	137	252	2457	0.244
project 5	975	1367	1.4e-03	1.0e-01	17.6	2.8	66	75.5	335	0.359
project 6	544	776	2.6e-03	9.7e-02	32.3	2.9	39	141	430	0.305
project 7	29,080	50,101	5.9e-05	5.9e-03	20.5	3.5	1709	540	18,324	0.135
project 8	641	997	2.4e-03	2.9e-02	9.7	3.1	60	139	537	0.467
project 9	1287	2117	1.3e-03	1.2e-01	16.3	3.3	54	133	623	0.288
project 10	17,263	19,391	6.5e-05	5.4e-02	18.7	2.3	114	235	6265	0.131
project 11	13,625	25,034	1.4e-04	3.5e-02	15.9	3.7	589	541	13,501	0.201
project 12	3156	3237	3.3e-04	8.4e-02	14.7	2.1	37	46.4	451	0.215
project 13	282	292	3.7e-03	3.2e-02	9.6	2.1	17	11.1	128	0.136
project 14	15,757	22,648	9.1e-05	3.8e-02	14.6	2.9	401	198	3417	0.223

cases, the degree distributions (in-degree and out-degree) can be described with power-law distributions with exponents close to 2 (Figure S4). This exponent is stable across the 2 orders of magnitude of differences in project sizes, and is consistent with prior results in a pharmaceutical and a hospital construction projects [38].

In Fig. 1(b) we show an example of perturbations in an activity network. Perturbations are concentrated in network neighborhoods, indicative of a clustering phenomenon. To test whether perturbations are inherited, we compute for each task the proportion p_{pert} of its parent activities which have a perturbation. We observe that perturbed activities have a significantly higher p_{pert} than non-perturbed activities for 11 out of 14 projects (Figure S5). This suggests a network inheritance mechanism of perturbations, where an activity is likely to inherit a perturbation from its parents. In addition, we find that the magnitude of the perturbation also follows such an inheritance mechanism. We compute for each activity network the correlation across all activities between p_{pert} and their absolute deviation δ from baseline (Figure S6). We observe a positive and significant correlation for the same 11 projects, further supporting the premise of perturbation inheritance within the activity network.

To estimate the extent to which perturbations spread (spreading distance), we compute for each activity network n the distance cross-correlation $C_n(d)$ between the absolute values of the perturbations of activities at a distance d (see Methods). A positive $C_n(d)$ indicates a propagation effect where perturbations spread over a distance d, while $C_n(d) = 0$ corresponds to unrelated perturbations. In Fig. 1(b), we show the average cross-correlation across all activity networks, $C(d) =< C_n(d) >$. The correlation decays slowly after the first downstream task, with significant positive values up to 4 activities downstream, indicative of a clustering of perturbations in local neighborhoods. The correlation values then become comparable to those obtained when perturbations are assigned to random nodes in the network (see Methods).

These findings show that activity network structures provide pathways for perturbations to spread between activities, for up to 4 activities downstream. These perturbations can spread to downstream activities, potentially unlocking large spreading events that can impact the timely completion of the entire project.

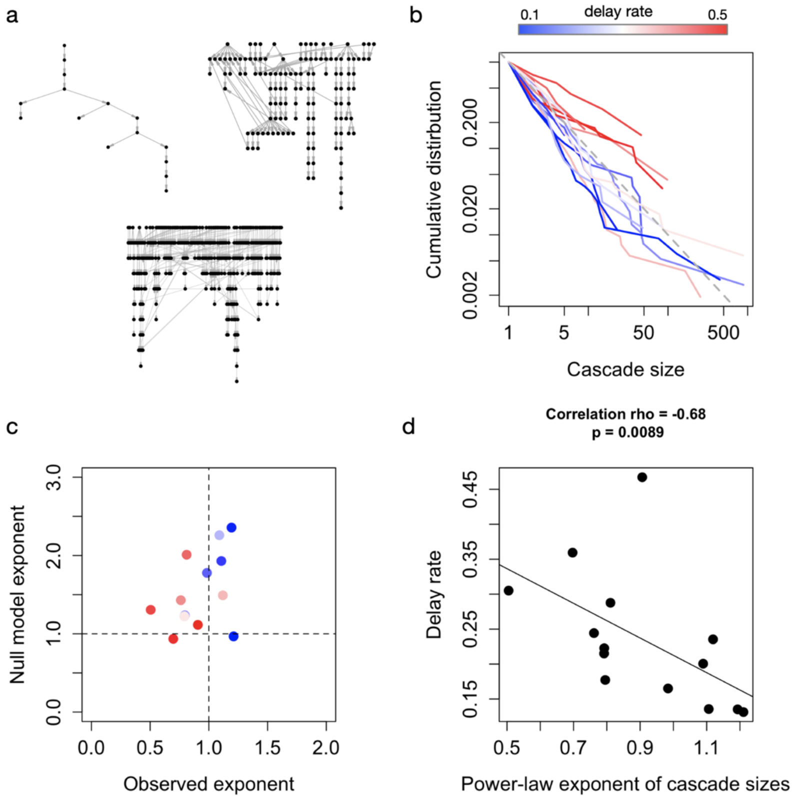

Figure 2 Structure of perturbation cascades predicts project performance. (**a**) Examples of perturbation cascades across projects with increasing tree size and complexity. (**b**) Cascade size distributions across the dataset. Color code denotes delay rate, measured by the overall proportion of delayed activities across completed activities, from blue (lowest) to red (highest). Dashed line corresponds to power-law distribution with an exponent of 1. We show distributions for the null model (shuffled perturbations) in Figure S8. (**c**) Comparison between observed power-law exponents of cascade sizes and null model exponents (see Methods). (**d**) Delay rate as a function of power-law coefficients of cascade sizes, showing a strong and statistically significant negative association ($\varrho = -0.68$, $p = 0.0089$, Spearman correlation)

2.2 The structure of real perturbation cascades

Perturbations for up to 4 activities downstream suggest the existence of clusters of perturbations, or *perturbation cascades*, in the activity networks. Cascades correspond to connected components of perturbed activities in the network. We show in Fig. 2(a) a few examples of cascades across projects, highlighting the diversity of structures and sizes. As in the case of node degree, we find that in 85% of the cases, cascade sizes can be described by a power-law distribution (see Methods and Figs. 2(b), S7). While the power-law nature of the cascade size distribution is expected if the perturbations were scattered randomly across the network (Figure S8), the exponents in observed cases are departing from random expectation (Fig. 2(c)). In accordance with the previous results showing a clustering of perturbations in local neighborhoods, the observed exponents are significantly smaller (between 0.5 and 1.2) than random expectation (between 0.9 and 2.4), indicative of larger, more extensive cascades in real-world projects.

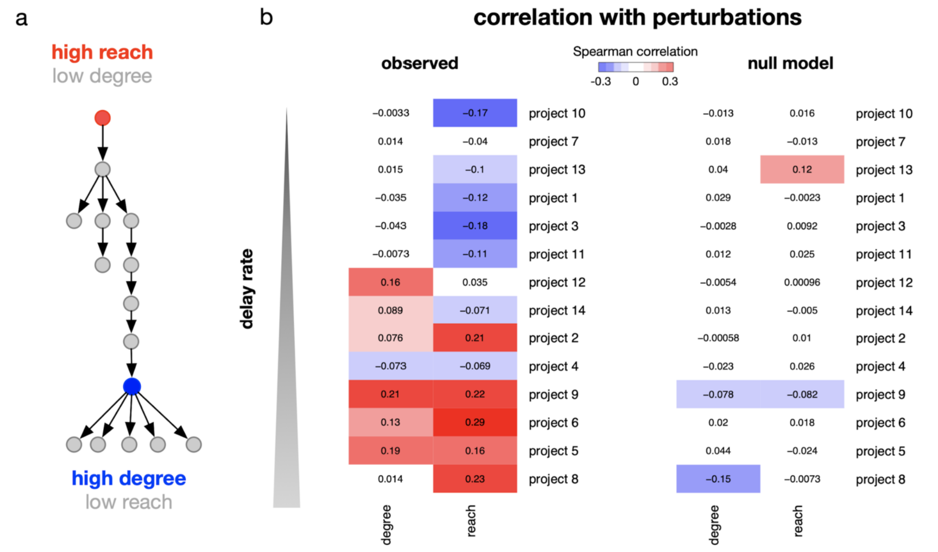

Figure 3 Node reach as a network fragility measure. (**a**) Schematics representing the two network centralities of interest. Node reach corresponds to the number of nodes downstream a given node, representing the maximum possible cascade size originating from that node, and is a global network measure. Node degree is a local network measure corresponding to the number of immediate neighbors. (**b**) Heatmaps showing the Spearman correlation between perturbation strength (absolute value of the perturbation) and two network metrics: node reach and node degree. Cell values indicate correlation values, with colors ranging from blue (lowest) to red (highest). Rows are ordered by increasing delay rate (i.e., decreasing global performance) of the project. Null model is obtained as in Fig. 1(**c**) by random shuffling of perturbation values across nodes

2.3 The structure of perturbation cascades and its impact on global performance

To further explore how the distribution of cascade sizes impacts the overall performance, we plot the delay rate as a function of the power-law exponent of cascade sizes for each project. We find strong-evidence ($\varrho = -0.68$, $p = 0.0089$, Spearman correlation) that the more localized the cascades are, the better the project performs in terms of overall delays from expectation (Fig. 2(b) and 2(d)). We used bootstrapping analysis to estimate the robustness of the significance with respect to sample size, showing that significance can be reached with at least 10 projects (see Figure S9). Finally, the result holds when controlling for the total number of perturbed nodes ($p = 0.02$, partial Spearman correlation), showing that for a similar number of perturbed nodes, projects that perform well manage to keep perturbations in local neighborhoods and avoid their spread, i.e., have a high power-law exponent, as shown in Fig. 2(c).

2.4 Global network structure underlies perturbation strength

In order to investigate the origin of these large, extensive cascades in low performing projects, we study network properties that might underlie such events: a local property, the network degree, and a global property, the number of nodes reachable downstream a given node, further coined 'node reach'. We focus on nodes for which the degree is strictly positive, meaning that they have at least one ancestor or offspring. We then ask how the degree and the reach relate to perturbation strength for each project: in particular, do large perturbations originate in nodes with specific high or low degree/reach? We show in Fig. 3 for each project the Spearman correlation between the node properties (degree and reach) and their absolute perturbation value. A positive (resp. negative) correlation means that highly perturbed nodes have a higher (resp. lower) value of the particular network property. We rank projects from best performing (lowest delay rate, top) to worst (highest delay rate, bottom). We observe that perturbations target higher degree nodes in low performing projects, while targeting both high and low degree activities in high performing projects. On the other hand, when turning to reach, we observe a positive association with perturbation strength in low performing projects, and a negative association in high performing projects.

The association of these network properties with global performance is significant only in the case of reach (Figure S10, $\rho = 0.78$, $p = 1.4\text{e-}3$; for degree we find $\rho = 0.4$, $p = 0.15$). This association remains significant when controlling for project size and number of perturbed nodes, both non-significant ($p = 3.2\text{e-}3$ for reach, linear regression).

Altogether, these results suggest that project performance is improved when large perturbations occur in nodes with small reach, limiting perturbation spread and eventually leading to more localised cascades.

3 Discussion

Managing large-scale projects is a daunting challenge, as large project sizes make it intractable for managers to harness project complexity. We showed that task perturbations occur irrespective of project size or task duration. These results validate prior insights from perturbation spreading models, where delays in project delivery were expected to be independent from project size [9]. This suggests that other factors are at play. In this work, we used a unique dataset of 14 large-scale engineering projects with activity networks and delay data to study how activity network properties relate to project performance.

The networks are small-world, making them structurally prone to the fast spreading of perturbations [9]. Here we showed that an inheritance mechanism enables large perturbations to spread up to 4 activities downstream of the root node, leading to perturbation cascades. The cascade sizes follow a power-law distribution, with smaller exponents than expected at random, indicative of larger clustering. Moreover, not all projects are equal: while some show localised, smaller cascades, others show extensive, larger cascades. We introduce an observable, the cascade distribution power-law exponent, that significantly predicts overall project performance. This exponent is predictive even when controlling for project size or number of perturbations, indicating that the clustering, and not the number, of perturbations is the source of poor project performance.

To investigate what network properties underlie larger cascades and poorer project performance, we introduced node reach as a key global network property. Poorly performing projects concentrate their largest perturbations in nodes with high reach, while well performing projects show the opposite trend, with largest perturbations in nodes with low reach.

It is interesting to contrast our results to previous insights gained from the application of complex system theory to project fragility [9, 10]. Using a perturbation spreading model, the authors showed that when the correlations between neighbors' degrees are small enough, the cascade dynamics can be related to the correlation between the in- and out-degree of a task. A positive correlation is associated with a higher project fragility, while a negative correlation favors in-time project completion. Consistently, we find that neighboring nodes show small negative degree correlations, and that there is an average positive correlation of a task in- and out-degree, indicating the structure of our studied project is more fragile (Figure S11a). When evaluating the importance of that feature across projects, we observe that the in-out degree correlations and the delay rate are moderately positively correlated at the 10% significance level (Figure S11b, $\rho = 0.45$, $p = 0.1$), calling for further future work to derive conclusive insights.

Scale-free networks, i.e networks for which the degree distribution follows a power-law, are known to be tolerant against random errors, but fragile under targeted attack towards central nodes [40]. As such, node centrality (and in particular, node out-degree, but also other correlated measures such as closeness or betweenness centrality) was hypothesized to play a critical role in predicting cascade size [9]. Consistently, we observe across 13 of the 14 projects studied a positive correlation between an activity's out-degree and the resulting cascade size (Figure S11c). When compared to degree-preserved random networks, the observed associations are moderately smaller than expected ($p = 0.13$, Mann–Whitney test), suggesting a relative protection of central nodes from perturbations.

This study exhibits the benefit of collecting a larger number of consistent activity datasets for validating associations with project performance, with the hope to uncover other contributors of project performance. For example, we showed in Figure S9 how the accumulation of a large enough number (10+) of activity networks allowed us to reach the required significance level to associate cascade size distribution and performance. Yet data volume is only one facet that can impact the accuracy of the analysis. Given that activity network data are human generated, structural errors may creep in, either in the form of missing links or redundant links. Such inconsistencies are bound to be limited, given the mission critical nature of the data, and the subsequent effort associated with project planners in generating and curating them. Larger volumes of data would help tackle this challenge, where random sampling methods could be used to contain such effects. We hope that our work will draw the attention of the community to this mission critical area of research, attracting concentrated efforts of work for exposing larger datasets that can enable such future work.

Our results pave a new way for elucidating the causal link between the structure of a project's activity network and its performance. We contribute actionable insights that can support decision makers mitigate cascades, by focusing their efforts in successfully completing high-reach nodes. From a reactive point of view, decision makers can use an activity's 'reach' to assess the priority in containing a delay when completing that activity. By doing so, decision makers can prioritise resource allocation in an effective and efficient manner. From a proactive standpoint, decision makers can provision frequent quality checks and stricter governance frameworks for activities with a high 'reach' so that to minimise the probability of delays arising in the first place. In doing so, our work partakes in the broader movement of solution-oriented social science where computational methods and big data can be used to uncover core insights for mitigating real-world challenges [41]. We believe that our contribution can stimulate a new wave of data-driven research in one of the most enduring societal challenges: why do almost all modern projects fail to be delivered on time, given that we have been delivering them for the past 80 years?

4 Methods

4.1 Data collection

Each activity network corresponds to a project schedule that was created using the Oracle Primavera P6 software—an industry standard platform used to create and manage large-scale, engineering projects (>$5 m). The corresponding data denotes functional dependencies between activities, spread across a timeline i.e., akin to a Gantt chart. In principle, the existence of cycles is forbidden, as their presence would require at least one directed link going backwards in time. As such, in the case where an existing activity needs to be reworked *after* some downstream activity has been fulfilled, one would insert a *new* downstream activity to indicate the additional work (instead of cycling back to the existing upstream activity). Yet, in reality some cycles may occur during the human annotation process underlying the generation of the data, but their existence is very limited (see Figure S3a).

We note that the schedules used herein are markedly different from the similarly purposed product development (PD) networks used in [42, 43], that indicate information flow between activities. The PD networks are usually orders of magnitude smaller (∼100 s of activities), and are created using the Dependency Structure Matrix method [44].

The combination of finely aggregated information at the activity level, the large number of activities and the minimal presence of cycles makes this data collection method ideal for the study of long-range delay propagation that we investigate in this study.

4.2 Power-law fit

In order to compute power-law fits to the degree and cascade size distributions, we use the poweRlaw package from [45], based on the method from [46]. This fitting procedure uses a Maximum Likelihood approach to estimate the exponent α of a power-law fit to the distribution:

$$p(x) = \frac{\alpha - 1}{x_{\min}} \left(\frac{x}{x_{\min}} \right)^{-\alpha}$$

The method then uses a bootstrap procedure to compute a Kolmogorov–Smirnov statistic and a corresponding p-value quantifying the confidence that the power-law fit is a plausible description of the empirical data. In the case of cascade distributions, cascade size refers to the number of downstream nodes impacted. We excluded singletons (i.e cascades of size 0) and set $x_{\min}$ equal to 1 across projects. In the manuscript, "power-law exponent" refers to the exponent of the cumulative distribution, corresponding to $\alpha - 1$ in the notation above (for $\alpha \neq 1$, which is always the case in our data).

4.3 Network distance cross-correlation

We compute for each activity network the distance cross-correlation $C(d)$ between the absolute value of a perturbation δ_i at node i and δ_j at node j for all (i,j) such that j is d steps downstream from i:

$$C(d) = \frac{< (\delta_i - \mu_i)(\delta_j - \mu_j) >}{\sigma_i \sigma_j} \quad \text{for all } d(i,j) = d,$$

where μ_i and σ_i correspond to the average and standard deviation of δ_i. A positive $C(d)$ indicates that perturbation spreads over a distance d, while $C(d) = 0$ corresponds to independent perturbations. In Fig. 1(c) we show the average and standard error of $C(d)$ across projects.

In order to obtain a random model, for each project we shuffle absolute perturbation values across all completed activities, and produce 50 randomized samples. For a project we then compute the random cross-correlation as $C_r(d) =< C_{r,i}(d) >$ where the average runs over all random samples i in $[1, 50]$. Finally we show in Fig. 1(c) the average and standard deviation of $C_r(d)$ across all projects.

4.4 Network visualisation

For network visualisations in Fig. 1(a) we use Gephi 0.9.2 with the ForceAtlas 2 layout.

4.5 Synthetic networks

For each activity network, we generated Erdös-Rényi (ER) and Barabasi–Albert (BA) random graphs of the same density as the observed network. The ER networks were generated using the erdos.renyi.game function from the R igraph package, with parameters n (number of nodes in the network), m (number of edges in the network), type = 'gnm' (use number of edges rather than edge probability), and directed = T (generate a directed graph). The BA networks were generated using the barabasi.game function of the same package, with parameters n (number of nodes), out.dist the out-degree distribution of the network, and directed = T to produce a directed graph.

In the case of Figure S11, the degree preserved networks were generated using the edge swapping method implemented in the keeping_degseq function of the igraph package. The rewiring algorithm chooses two arbitrary edges in each step ((a, b) and (c, d)) and substitutes them with (a, d) and (c, b), if they do not already exist in the graph. The algorithm does not create multiple edges. We did not allow for loop edges, and ran the algorithm for $N_{\text{iter}} = 10 * E$ iterations, where E is the number of edges in the graph.

4.6 Cycles

To find cycles in the networks, we computed all isomorphisms to a directed ring graph of size k. This was done by first defining the ring graph using the function graph.ring (k, directed = T) in the R igraph package. We then used the function graph.get.subisomorphisms.vf2(graph, ring) between the considered graph and the ring.

4.7 Statistics

All statistics, correlations and plots are computed using R version 4.0.1. Spearman correlations are used throughout this work in order to limit the effect of outliers. The partial Spearman correlation p-value is computed using the pcor.test function of the ppcor R library [47]. Throughout the manuscript, p refers to the p-value of the statistical test for the preceding quantity.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1140/epjds/s13688-021-00291-w.

Additional file 1. Supplementary information (PDF 3.7 MB)

Acknowledgements
We thank the anonymous Reviewers for constructive comments and suggestions to improve the manuscript.

Funding
Thanks to the Bettencourt Schueller Foundation long term partnership, this work was partly supported by the CRI Research Fellowship to Marc Santolini. Nodes & Links Ltd provided support in the form of salary for Christos Ellinas, but did not have any additional role in the conceptualisation of the study, analysis, decision to publish, or preparation of the manuscript. Christos Nicolaides has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 786247.

Abbreviations
Not applicable.

Availability of data and materials
The datasets used and analysed during the current study are available upon reasonable request.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
MS, CN and CE conceptualised the study, MS and CE devised the methodology, CE collected the data, MS analyzed the data. MS, CN and CE wrote the manuscript. All authors read and approved the final manuscript.

Author details
[1]Université de Paris, INSERM U1284, Center for Research and Interdisciplinarity (CRI), F-75006, Paris, France. [2]Network Science Institute and Department of Physics, Northeastern University, Boston, MA 02115, USA. [3] Nodes & Links Ltd, Station Road, Cambridge, CB1 2LA, UK. [4]School of Economics and Management, University of Cyprus, Aglantzia, 2109, Cyprus. [5]Initiative on the Digital Economy, MIT Sloan School of Management, Cambridge, MA 02142, USA.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 24 September 2020 Accepted: 22 June 2021 Published online: 08 July 2021

References
1. Flyvbjerg B, Holm MKS, Buhl SL (2003) How common and how large are cost overruns in transport infrastructure projects?. Transp Rev 23:71–88. https://doi.org/10.1080/01441640309904
2. Flyvbjerg B (2007) Cost overruns and demand shortfalls in urban rail and other infrastructure. Transp Plann Technol. 30:9–30. https://doi.org/10.1080/03081060701207938
3. Flyvbjerg B (2014) What you should know about megaprojects and why: an overview. Proj Manag J. 45:6–19. https://doi.org/10.1002/pmj.21409
4. Hartmann S, Briskorn D (2010) A survey of variants and extensions of the resource-constrained project scheduling problem. Eur J Oper Res 207:1–14
5. Kiridena S, Sense A (2016) Profiling project complexity: insights from complexity science and project management literature. Proj Manag J. https://doi.org/10.1177/875697281604700605
6. Geraldi J, Maylor H, Williams T (2011) Now, let's make it really complex (complicated): a systematic review of the complexities of projects. Int J Oper Prod Manag 31:966–990. https://doi.org/10.1108/01443571111165848
7. Mihm J, Loch C, Huchzermeier A (2003) Problem–solving oscillations in complex engineering projects. Manag Sci. 49:733–750. https://doi.org/10.1287/mnsc.49.6.733.16021
8. Pich MT, Loch CH, Meyer AD (2002) On uncertainty, ambiguity, and complexity in project management. Manag Sci. 48:1008–1023. https://doi.org/10.1287/mnsc.48.8.1008.163
9. Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. Manag Sci. 53:1127–1145. https://doi.org/10.1287/mnsc.1060.0617
10. Braha D (2016) The complexity of design networks: structure and dynamics. In: Cash P, Stanković T, Štorga M (eds) Experimental design research: approaches, perspectives, applications. Springer, Cham, pp 129–151
11. Oehmen J, Thuesen C, Ruiz PP, Geraldi J (2015) Complexity management for projects, programmes, and portfolios: an engineering systems perspective. PMI white pap
12. Navigating Complexity, A Practice Guide (2014)
13. Ellinas C (2019) The domino effect: an empirical exposition of systemic risk across project networks. Prod Oper Manag. 28:63–81. https://doi.org/10.1111/poms.12890
14. Pastor-Satorras R, Castellano C, Van Mieghem P, Vespignani A (2015) Epidemic processes in complex networks. Rev Mod Phys 87:925–979. https://doi.org/10.1103/RevModPhys.87.925
15. Helbing D (2013) Globally networked risks and how to respond. Nature 497:51–59. https://doi.org/10.1038/nature12047
16. Vespignani A (2012) Modelling dynamical processes in complex socio-technical systems. Nat Phys 8:32–39. https://doi.org/10.1038/nphys2160
17. Watts DJ (2002) A simple model of global cascades on random networks. Proc Natl Acad Sci 99:5766–5771. https://doi.org/10.1073/pnas.082090499
18. Pastor-Satorras R, Vespignani A (2001) Epidemic spreading in scale-free networks. Phys Rev Lett 86:3200–3203. https://doi.org/10.1103/PhysRevLett.86.3200
19. Liu Z, Hu B (2005) Epidemic spreading in community networks. Europhys Lett 72:315. https://doi.org/10.1209/epl/i2004-10550-5
20. Nicolaides C, Cueto-Felgueroso L, González MC, Juanes R (2012) A metric of influential spreading during contagion dynamics through the air transportation network PLoS ONE 7:e40961. https://doi.org/10.1371/journal.pone.0040961.
21. Nicolaides C, Avraam D, Cueto-Felgueroso L et al (2020) Hand-hygiene mitigation strategies against global disease spreading through the air transportation network. Risk Anal 40:723–740. https://doi.org/10.1111/risa.13438
22. Santolini M, Barabási A-L (2018) Predicting perturbation patterns from the topology of biological networks. Proc Natl Acad Sci. 115:E6375–E6383. https://doi.org/10.1073/pnas.1720589115
23. Vanunu O, Magger O, Ruppin E et al (2010) Associating genes and protein complexes with disease via network propagation PLoS Comput Biol 6:e1000641. https://doi.org/10.1371/journal.pcbi.1000641

24. Menche J, Sharma A, Kitsak M et al (2015) Disease networks. Uncovering disease-disease relationships through the incomplete interactome. Science 347:1257601. https://doi.org/10.1126/science.1257601
25. Sharma A, Kitsak M, Cho MH et al (2018) Integration of molecular interactome and targeted interaction analysis to identify a COPD disease network module. Sci Rep 8:14439. https://doi.org/10.1038/s41598-018-32173-z
26. Reggiani A, Nijkamp P, Lanzi D (2015) Transport resilience and vulnerability: the role of connectivity. Transp Res, Part A, Policy Pract 81:4–15
27. Ivanov N, Netjasov F, Jovanović R et al (2017) Air traffic flow management slot allocation to minimize propagated delay and improve airport slot adherence. Transp Res, Part A, Policy Pract. 95:183–197. https://doi.org/10.1016/j.tra.2016.11.010
28. Derrible S, Kennedy C (2010) The complexity and robustness of metro networks. Phys A, Stat Mech Appl 389:3678–3691. https://doi.org/10.1016/j.physa.2010.04.008
29. Ellinas C (2018) Modelling indirect interactions during failure spreading in a project activity network. Sci Rep 8:4373. https://doi.org/10.1038/s41598-018-22770-3
30. Guo N, Guo P, Shang J, Zhao J (2020) Project vulnerability analysis: a topological approach. J Oper Res Soc 71:1233–1242. https://doi.org/10.1080/01605682.2019.1609882
31. Sosa ME (2014) Realizing the need for rework: from task interdependence to social networks. Prod Oper Manag. 23:1312–1331. https://doi.org/10.1111/poms.12005
32. Terwiesch C, Loch CH (1999) Managing the process of engineering change orders: the case of the climate control system in automobile development. J Prod Innov Manag 16:160–172. https://doi.org/10.1111/1540-5885.1620160
33. Christoph L, Terwiesch C (1998) Communication and uncertainty in concurrent engineering Manag Sci 44:1032–1048.
34. Kelley JE, Walker MR (1959) Critical-path planning and scheduling. In: Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference. Association for Computing Machinery, New York, pp 160–173
35. Malcolm DG, Roseboom JH, Clark CE, Fazar W (1959) Application of a technique for research and development program evaluation. Oper Res 7:646–669
36. Vanhoucke M (2013) An Overview of Recent Research Results and Future Research Avenues Using Simulation Studies in Project Management. In: ISRN Comput. Math. https://www.hindawi.com/journals/isrn/2013/513549/. Accessed 2 Feb 2021
37. Elmaghraby SE (1995) Activity nets: a guided tour through some recent developments. Eur J Oper Res 82:383–408. https://doi.org/10.1016/0377-2217(94)00184-E
38. Braha D, Bar-Yam Y (2004) Topology of large-scale engineering problem-solving networks. Phys Rev E 69:016113. https://doi.org/10.1103/PhysRevE.69.016113
39. Ellinas C, Allan N, Johansson A (2016) Exploring structural patterns across evolved and designed systems: a network perspective. Syst Eng 19:179–192. https://doi.org/10.1002/sys.21350
40. Albert R, Jeong H, Barabási A-L (2000) Error and attack tolerance of complex networks. Nature 406:378–382. https://doi.org/10.1038/35019019
41. Watts DJ (2017) Should social science be more solution-oriented? Nat Hum Behav 1:1–5. https://doi.org/10.1038/s41562-016-0015
42. Yassine A, Braha D (2003) Complex concurrent engineering and the design structure matrix method. Concurr Eng 11:165–176. https://doi.org/10.1177/106329303034503
43. Braha D (2020) Patterns of ties in problem-solving networks and their dynamic properties. Sci Rep 10:18137. https://doi.org/10.1038/s41598-020-75221-3
44. Eppinger SD, Whitney DE, Smith RP, Gebala DA (1994) A model-based method for organizing tasks in product development. Res Eng Des 6:1–13. https://doi.org/10.1007/BF01588087
45. Gillespie CS (2015) Fitting heavy tailed distributions: the poweRlaw package. J Stat Softw 64:1–16. https://doi.org/10.18637/jss.v064.i02
46. Clauset A, Shalizi CR, Newman MEJ (2009) Power-law distributions in empirical data. SIAM Rev 51:661–703. https://doi.org/10.1137/070710111
47. Kim S (2015) Ppcor: an R package for a fast calculation to semi-partial correlation coefficients. Commun Stat Appl Methods 22:665–674. https://doi.org/10.5351/CSAM.2015.22.6.665