Network Analysis of Collaborative Design Networks: A Case Study of OpenIDEO

Mark Fuge[*]
Kevin Tee[†]
Alice Agogino[‡]

Berkeley Institute of Design
University of California
Berkeley, CA 94709

Nathan Maton

IDEO
San Francisco, CA 94105

Abstract

This paper presents a large-scale empirical study of OpenIDEO, an online collaborative design community. Using network analysis techniques, we describe the properties of this collaborative design network and discuss how it differs from common models of network formation seen in other social or technological networks. One major finding is that in OpenIDEO's social network the highly connected members talk more to less connected members than each other—a behavior not commonly found in other social and collaborative networks. We discuss how some of the interventions and incentives inherent in OpenIDEO's platform might cause this unique structure, and what advantages and disadvantages this structure has for coordinating distributed design teams. Specifically, its core-periphery structure is robust to network changes, but is at risk of decreasing design exploration ability if the core becomes too heavily clustered or loses efficiency. We discuss possible interventions that can prevent this outcome: encouraging core members to collaborate with periphery nodes, and increasing the diversity of the user population.

1 The Rise of Distributed Design Communities

To solve increasingly complex design problems, companies are beginning to look outside of their existing talent pool to absorb and build off of ideas from distributed individuals or groups. This practice is called different names by different groups, including Open Innovation, Crowdsourcing, and Crowd Design, among others. It is practiced by a range of organizations, from large global corporations (*e.g.*, P&G's Connect+Develop program[1]) all the way down to small decentralized groups of individuals (the Open Source Software movement). Internet technologies enable regular people to cooperatively design better products, permitting a powerful new kind of product development process.

To increase the effectiveness of these distributed teams, it would be helpful to understand how they act differently than traditional groups, and how existing design and management practices need to be adapted to this new setting. This paper contributes to that understanding through the use of network analysis techniques. By comparing a real-world design network with prior models of collaborative networks, this paper presents two main contributions:

(1) An empirical network analysis of OpenIDEO, an online design innovation network, which can act as a test bed for models of design networks.
(2) A summary of key differences between observed behavior and existing network models, with discussion on the implications for directing design practice.

Specifically, we explore the role of community structure in OpenIDEO, explaining how some of its common network properties predispose OpenIDEO to certain advantages and disadvantages when facilitating idea generation and collaboration. We find that OpenIDEO's social interactions center around a core of users who communicate more frequently with members on the periphery than among themselves (an uncommon disassortative core-periphery social structure). This structure is more robust to network changes than standard social networks—a good thing for open innovation platforms in which participation is voluntary. However, the central core structure also represents a risk to potential idea generation effectiveness: high clustering within the core could precipitate design fixation on a small number of concepts as a result of complex contagion (repeated exposure to the same stimuli from multiple people) [1]. We discuss several possible interventions that can prevent this effect, such as promoting collaboration between core and periphery members and increasing diversity of participants.

This paper provides a brief introduction of current network models and reviews previous studies of similar networks (*e.g.*, Open Source Software networks, Co-Authorship networks, *etc.*). It then describes the OpenIDEO dataset and the network qualities we studied, and presents our empirical

[*]Address all correspondence to this author `mark.fuge@berkeley.edu` – Dept. Mechanical Engineering
[†]`kevintee@berkeley.edu` – Dept. Computer Science
[‡]`agogino@berkeley.edu` – Dept. Mechanical Engineering

[1]http://www.pgconnectdevelop.com/

results. Finally, it discusses the implications of our results on design network models and management strategies for distributed design teams.

2 Prior Research on Network Structures

Despite the growing trend to use distributed design communities to crowd-source design tasks, there has been limited empirical study of the network properties of product or service design communities themselves. This is due, in part, to the lack of non-proprietary data, as well as the relatively recent emergence of online design communities compared to communities in different fields (*e.g.*, software or social communication). A notable exception is Stephen *et al.* [2], who studied network effects of small, experimentally-controlled design teams. This paper extends that line of work by studying large in-situ teams. To our knowledge, this paper is the first to empirically study the collaboration practices of a distributed product or service design community that operates outside of a single corporate entity.

This paper builds on prior work in three areas: 1) network analysis techniques, 2) existing empirical studies of collaboration networks, and 3) existing theoretical models of design networks. These three aspects provide the background necessary to discuss the main results of our study, which center around the network properties of OpenIDEO and its effects on ideation.

2.1 Background on Network Analysis Techniques

Network Analysis is a class of mathematical techniques that can be used to study particular types of complex phenomena. Its primary assumption is that a phenomenon can be reasonably modeled as a mathematical graph consisting of nodes (or *vertices*) connected to each other by links (or *edges*). For example, in a social network such as Facebook, a node might be a person, a link might be the strength of a relationship, and the phenomena of interest might be how a viral video propagates among people over time. By representing phenomena as graphs, network analysis can adapt measures from graph theory in order to explain or predict certain behaviors, ranging from disease transmission to co-authorship to protein interactions. A full summary of these techniques is beyond the scope of the paper, but interested readers are directed to [3,4].

The most critical assumption in any network analysis study comes from how the network nodes and links are defined. However, once the nodes and links have been defined, one can compare several graph properties, both on a global (whole network) level, and at a local (node-centric) level that provide insight into the behavior of the network. For example, graph properties can be used to predict qualities like the social power of individuals, weak-points in information flow within networks, or the likelihood of co-authorship between researchers.

Our goal in this paper is to investigate several network properties that have implications on the ability to share and build off of information present in a design collaboration network. We first define some commonly used terms from Network Analysis that help clarify our later explanations:

Link (also called an edge) is a connection between two nodes on the graph. It can have a direction as well as a weight (*e.g.*, A sent B ten emails).

Size is the total number of nodes in a graph.

Diameter is the length of the shortest path between the two farthest nodes in the graph. It provides a sense of how spread out the graph is and provides one measure of the resistance to the flow of information.

Connected Component is a subset of the nodes in a graph that can be reached by following links between them. For example, if two nodes are connected to each other, but not to any other nodes, then they form their own connected component. Real-world networks tend to have one large connected component which contains most of the nodes (*e.g.*, the center component of Fig. 1d contains around 95% of the nodes), followed by some smaller components with only a few nodes each (*e.g.*, the single nodes on the outside of Fig. 1d).

Density is the ratio between the number of links that exist between nodes and the maximum number of possible links that could exist (*i.e.*, a complete graph). In large real-world networks, the density is typically low.

Clustering Coefficient is a measure of how tightly connected nodes are in a graph, specifically measuring the ratio of number of triangles between a node and any two neighbors and the number of possible triangles (*e.g.*, how many of your friends are also friends with each other) [3]. Node-wise clustering coefficients can be averaged to characterize how clustered a graph is as a whole [3].

Centralization refers to how well the graph is centered around a single focal point on a scale from zero to one. High centralization would imply a deeply hierarchical structure, such as a star graph (Fig 1a), while low centralization would imply that all nodes are equally central, such as a cycle graph (Fig 1c).

Efficiency measures how easily and quickly information is transferred across a network. It is inversely related to the average shortest path length required to go between all pairs of nodes on the graph; if efficiency is high, all nodes are within a few links of one another, and if efficiency is zero then no node can communicate with any other node.

Degree of a node measures the number of incoming and outgoing links to that node. For example, in Fig. 1(c), node one has a degree of four because it connects to four other nodes.

Degree Distribution refers to the fact that different nodes have different degrees. The distribution of these degrees follows different patterns depending on the type of network structure. In many real-world networks, this distribution is *power-law distributed* (or *scale-free*), which means that it exhibits a relatively linear plot when plotted log-log scaled, as in Fig. 3. This corresponds to many nodes having only a few links, and only a few nodes having many links.

Assortativity or assortative mixing, is the propensity for nodes in a network to create links with similar nodes, and to avoid creating links with dissimilar nodes. For example, engineers might be more likely to be friends with other engineers than with dentists, and vice versa. *Degree assortativity*, means that nodes with high degree (those who communicate with many people) are more likely to communicate with other nodes with high degree, instead of nodes with low degree (those who communicate infrequently). Social networks are known for being positively degree assortative.

k-Clique is a set of k nodes that are all connected to one another (*i.e.*, they form a complete sub-graph). For example, if A knows B and C, and B also knows C, then A, B, and C are a 3-clique. We study these cliques in section 3.3 to address community structure in OpenIDEO.

We return to several of the above network properties in section 3, when we address how each of them determines the advantages and disadvantages of the OpenIDEO network for idea generation and collaboration.

2.2 Empirical Findings from Other Fields

We review prior studies of three types of networks: 1) Open Source Software, 2) research co-authorship, and 3) social communication. We pick these three since they each have elements you would expect to find in a collaborative design network, and therefore serve as a meaningful basis on which to benchmark OpenIDEO.

Open source software is similar to open design networks in that the members are typically decentralized, can choose which projects they want to work on, and are creating some artifact that will be used by people. This type of network study is the closest existing example to the work that we are presenting, though it is different in both the kind of project as well as the specific mechanisms of collaboration. In most studies of Open Source Software, the node unit of analysis is a particular developer and a link exists between developers if they have worked on the same project together [5]. These networks display high assortativity and often generate many smaller communities, particularly around programming languages. They possess standard power-law distributed degree distributions that are typical of many social networks [6].

Research co-authorship is another type of network where there is formal interaction and the goal is to generate new ideas in collaboration with others. It differs from OpenIDEO in that the barriers to collaboration in OpenIDEOs case are smaller than for research co-authorship, and the online social interactions in OpenIDEO are traceable in a way that is not feasible in research networks. Co-authorship networks are also positively assortative, tend to form communities within the larger network, and have a low average clustering coefficient [3].

Social networks, such as social media or email networks, are similar to online design networks in that a traceable process of social communication occurs between participants, and because OpenIDEO's user population is situated in a social community. Social networks tend to be highly positively assortative with multiple smaller communities of people interacting together [3,7,8]. They also tend to possess power-law distributed degree distributions, though there are notable exceptions (*e.g.*, Facebook [8]).

2.3 Theoretical Models for Design Networks

While there has been limited empirical work on actual design networks, prior research has proposed different *theoretical* models for how design networks might operate. The vast majority of available theoretical models for collaborative design networks are either simulation studies using agents with predefined collaboration rules, or are optimization studies which fit common theoretical network models, such as Preferential Attachment models [3, Ch. 14.1], to data from almost exclusively open source software communities.

Agent-based simulations typically define a computational model of a product, and then create a series of software agents who can choose what portion of the product to work on. These simulations then track the product and communication between the agents, building a simulated collaboration network that can then be analyzed for structural properties [9–11]. The typical applications for this line of work are in identifying potential strategies for managing complex system design, under the assumption that the agents behave similarly to real people.

In contrast, network optimization studies attempt to take real network data and fit theoretical network models to that data [12]. The key assumption behind this line of work is that if real networks obey certain properties, such as power-law distributions, one should be able to determine those parameters by matching the theoretical model to the data. Upon doing so, insights are often gained about why the network does or does not conform to theoretical expectations.

3 Analysis of OpenIDEO Design Network

We chose OpenIDEO from among other possible online design collaboration platforms (*e.g.*, Napkin Labs, frogMob, VehicleForge, *etc.*) due to the breadth of project types, the large user community, and availablity of collaboration metadata (such as explicit links between concept ideas). To understand both the general properties of the OpenIDEO network as well as the properties of any sub-communities, we divided our analysis into three parts: 1) Structural measures that address information flow within the network; 2) Community measures that address the network's robustness and community structure; and 3) Effects of certain members that address the specific role of possible OpenIDEO interventions.

3.1 Data Collection and Pre-processing

We collected data from 22 OpenIDEO challenges that were complete at the time of writing[2]. Most design challenges start with a question, for example "How might we

[2] http://www.openideo.com/open

restore vibrancy in cities and regions facing economic decline?" Each challenge consists of a series of sequential phases: Inspiration, Concepting, Applause, Evaluation, Selection of Winners, and Realization. During Inspiration, contributers can submit "insights, examples, stories, or comments"[3] designed to provoke possible solutions from the community. During Concepting, contributors submit concepts designed to solve the challenge. Inspirations and concepts are typically a few paragraphs long with accompanying figures.

In either stage, contributors can link to other people's inspirations or concepts by clicking a "build off of this idea" button in the web interface. This creates an explicit link that we use to model interrelations between the submissions: we construct a graph (using the NetworkX library [13]) where nodes represent a submission in the particular design challenge, and we add a directed link from node A to B if concept B builds upon concept A (*e.g.*, Fig. 1). We refer to this as the *concept graph*. We created separate concept graphs for each of the 22 challenges.

During all stages of the challenge, people may post comments on other people's concepts as well as reply to comments on their own concepts. To model this social interaction, we construct a separate graph (the social graph) where each node represents an OpenIDEO user, and a weighted directed link is added from user A to user B if user A comments on user B's concept for that particular design challenge or if user A replies to a comment given by user B. Every additional interaction from A to B adds an additional unit of weight to the link between A and B. There are separate social graphs for each of the 22 challenges. There are particular users, who we refer to as OpenIDEO community managers, that have specific roles on the platform: they help facilitate the challenge by reaching out to many concepts and commenting on them to promote interaction. There are usually two of these managers per challenge, one who is a member of IDEO staff and another who is an active member of the larger community.

3.2 Structural Measures

The measures in this section are designed primarily to address the ease with which information flows through each network. Greater size and diameter imply that information has farther to travel, while greater clustering, centralization, efficiency, and density imply greater ease of information transfer.

Comparing the concept and social graphs across challenges reveals certain key structural similarities and differences: Despite similar network sizes, the two networks have drastically different link structures, diameters, densities, and average clustering. Figure 1b and d illustrate representative concept and social graphs, respectively, positioned using a Fruchterman–Reingold force directed layout algorithm; immediate inspection reveals the tightly clustered core-periphery structure of the social graph (Fig. 1d) as well as the sparser, more community clustered concept graph (Fig. 1b). To make the differences between these structures clearer, Fig. 2 highlights several key similarities and differences:

Size (Figure 2a): within each challenge, both the concept and social graphs have approximately between 200-800 nodes, with two concept graphs reaching into the low 1,000s—these sizes are fairly small compared to social internet communication networks, but large compared to the size of typical collaborative design groups. Particularly for the concept graphs, this size indicates that it would be time-prohibitive for any individual to actually read through all available ideas in a challenge.

Diameter (Figure 2b): Since both types of networks in OpenIDEO have disconnected components (thus infinite graph diameters), it is more reasonable to measure the diameter of the largest connected component of the graph. For that case, OpenIDEO's social graph has a significantly smaller component diameter than that of the concept graph, despite their being of roughly equal size. Part of the reason for this smaller diameter is the fairly efficient center of the social graph (Fig. 1d) which bridges many nodes, decreasing the distance information needs to travel and making communication and feedback easier to transmit.

It is notable, though not unexpected, that both the concept and social graph have disconnected components. This indicates that there are concepts that are never being built off of and users who are not participating in the social community, both of which are losses of potential information.

Density (Figure 2c): The social graph is about four times as dense as the concept graph—most people who help out with the challenge interact at least once, whereas, on average, 42% of the inspirations or concepts that get submitted are never built off of (or at least tagged as such on the website). The concept graph's low density is possibly due to the shear number of available concepts or the effort required to build off of an idea.

Average Clustering Coefficient (Figure 2d): As expected, the sparsely connected and spread out concept graph has low average clustering, while the social graph has higher clustering, roughly comparable to other social networks.

Centralization (Figure 2e): Figure 2e demonstrates that both the concept graphs and social graphs are decentralized, with the concept graphs having significantly less centralization. Both of these results match what you would expect from an open innovation platform: many users should have access to different parts of the graph in order to have exposure to diverse groups of ideas and people. Part of the increased centralization in the social graph comes from the presence of the OpenIDEO community managers who are well connected to many members of the social community—a point we explore more in section 3.4.

[3] http://www.openideo.com/how-it-works/full.html

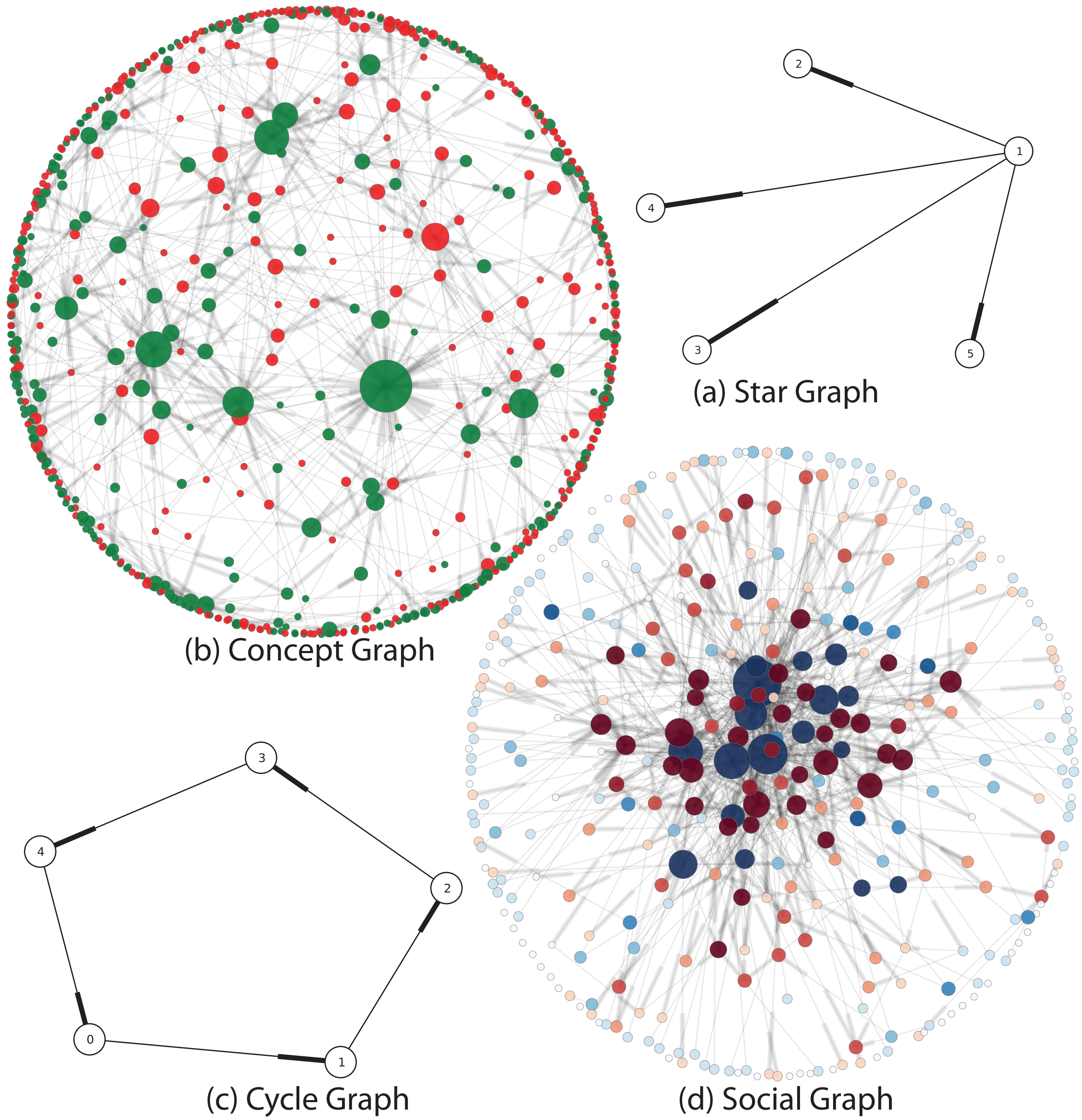
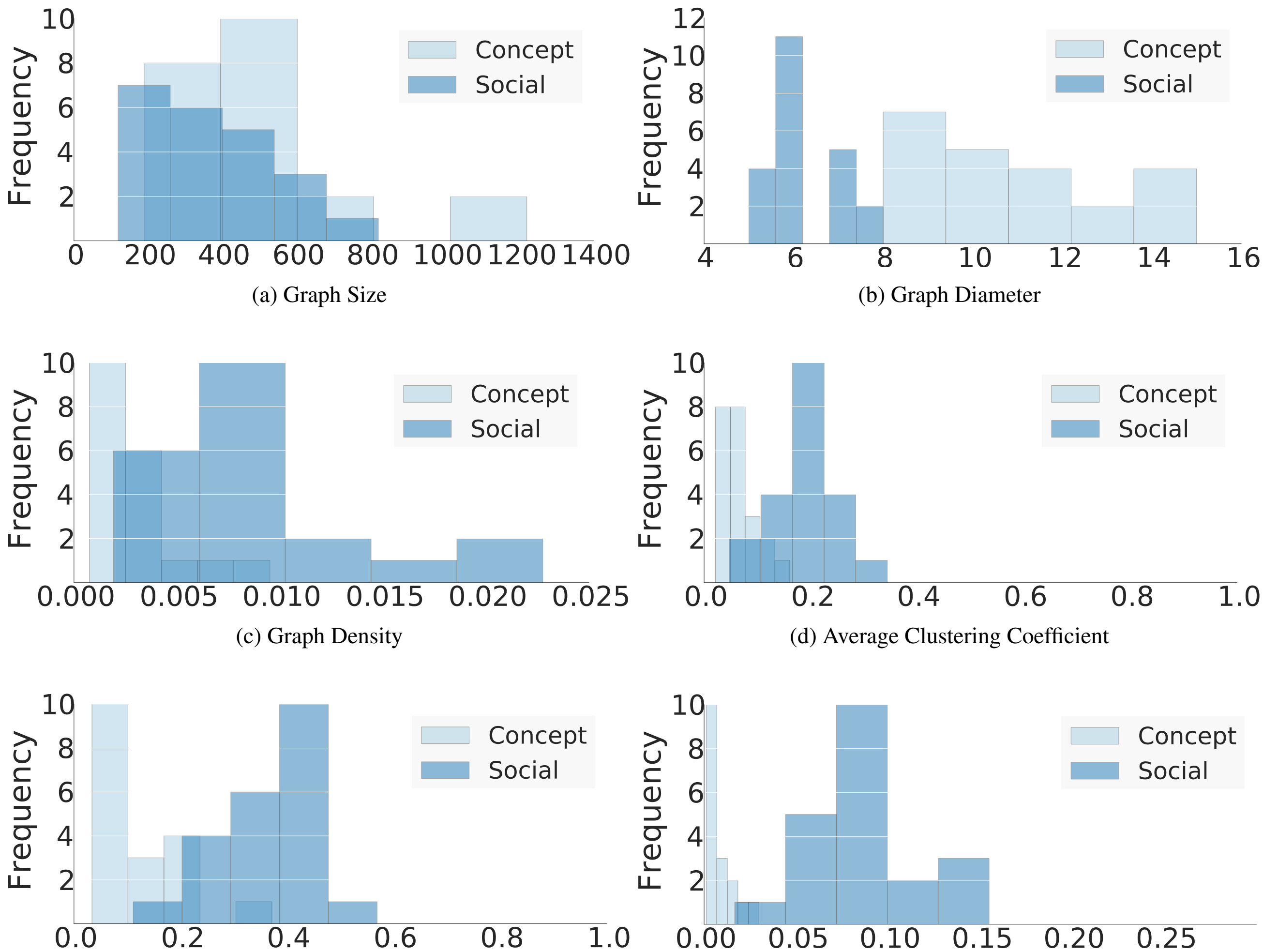


Fig. 1: Directed links are represented by a thicker segment indicating the direction (*e.g.*, in (c) 2 points to 3, and 3 points to 4). (b) is a concept graph, where red nodes represent inspirations and the green nodes represent concepts. (d) is a social graph, where redder nodes indicate more comments are received than given, whereas bluer indicates the opposite. In both cases, the size of the nodes represent the degree (number of incoming and outgoing links) of the node.

Efficiency (Figure 2f): Figure 2f illustrates the global network efficiency of both the concept and social graph. The low efficiencies in the graphs come with both advantages and disadvantages: on the one hand, lower efficiencies mean higher network redundancy and robustness at a given density—the concept graph has both low density and low efficiency, so it doesn't gain the redundancy benefit, while the social graph's central core structure does. However, at a given density, higher efficiencies create better information transfer across the network and also correlate to lower clustering—this is a useful structure when groups of people have to collaboratively solve uncertain problems together without getting stuck [1]. We will return to these ramifications in section 4 once we discuss the role that community structure plays in these two types of networks.



(a) Graph Size

(b) Graph Diameter

(c) Graph Density

(d) Average Clustering Coefficient

(e) Degree Centralization

(f) Global Efficiency

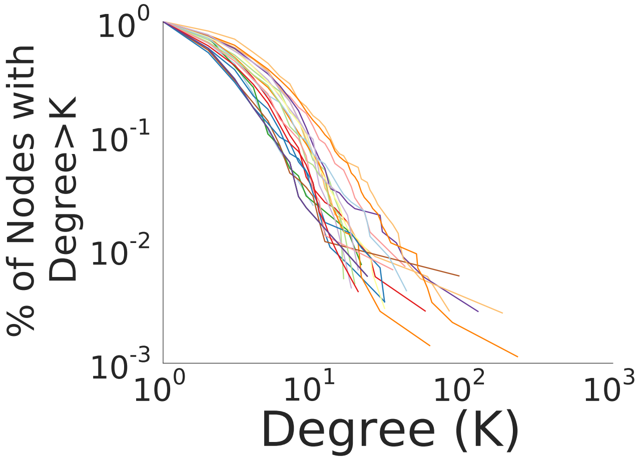
Fig. 2: The concept graphs have higher diameter(2b) and lower density (2c) than the social graphs, despite roughly equivalent network sizes(2a). This is possible due to small levels of clustering within the concept graph, and the fact that the social graph has certain mechanisms built in that reduce the graph diameter (see section 4). The concept graph exhibits low centralization(2e) and low global efficiency(2f), while the social graph exhibits medium centralization and low efficiency. In both cases, higher efficiency would be more advantageous in order to ease transfer of ideas and feedback, respectively. Figure 1 provides some visual intuition behind these results.

3.3 Community Measures

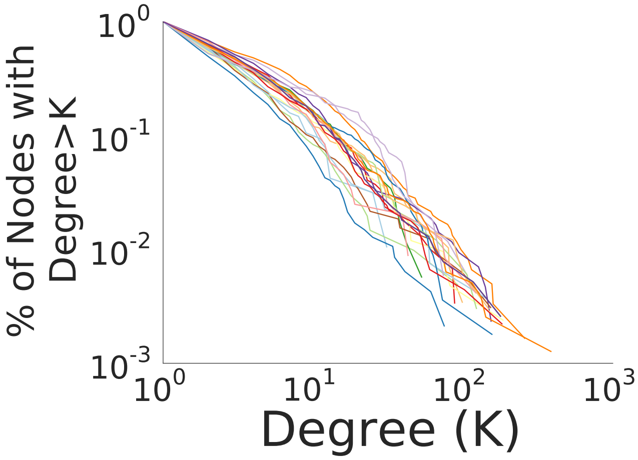
To understand the type of community structures inherent in the OpenIDEO network, we conducted three types of analysis: (1) degree distribution, (2) assortativity, and (3) community detection using the k-clique percolation method [14]. The results were unexpectedly different than other networks of their type: the social graph is highly disassortative with only a single, large core structure, while the concept graph has many small communities. For the social graph, this unique structure gives it higher robustness under node removal than standard social networks, and its disassortativity likely helps it maintain that structure—both of these are advantageous properties for an open innovation network where participation is voluntary.

Degree Distribution (Figure 3): Both the concept and social graphs appear power-law distributed, due to the linear nature of the degree distributions in Fig. 3. In terms of robustness, power-law distributed networks are robust (*i.e.*, do not change much) under random node removal (*i.e.*, random people leaving the network), but are particularly susceptible to targeted node removal (*i.e.*, removing the highest degree or most important individuals) [3]. However, as we demonstrate below, OpenIDEO possesses a core-periphery structure that mitigates this robustness concern [15, 16]; even removing several of OpenIDEO's highest degree members (the OpenIDEO community managers) does not significantly alter the network properties.

Assortativity (Figure 4): One possible reason for the social graph's robust core-periphery structure lies in the network's lack of assortativity. Figure 4 compares the assortativity of the OpenIDEO concept and social graphs, where assortativity ranges from 1 (completely assortative) to -1 (com-



(a) Concept Graphs



(b) Social Graphs

Fig. 3: Degree complementary cumulative distribution functions (CCDF) for the largest connected component of different types of Open IDEO networks. Each line corresponds to a different challenge. Both types of networks are generally power-law distributed.

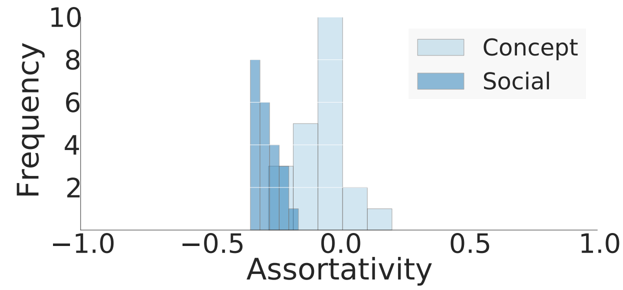


Fig. 4: Unlike most social networks, the OpenIDEO social graph appears negatively assortative (disassortative) by degree, rather than positively assortative. This means that members with high degree (lots of communication) talk more with those with low degree, rather than with others of high degree. This style of communication is highly atypical of most social networks. It reduces the diameter of the network and increases the fraction of the members in the largest graph component. The concept graph appears neither assortative nor disassortative.

pletely disassortative).

Unlike other social networks, the OpenIDEO social graph is actually *disassortative*, meaning that those members who communicate frequently are actually communicating more often with infrequent members of the group rather than frequent members, and vice versa. Indeed the directed links in Fig. 1d display a balance between outsiders commenting on concepts generated by members within the core, as well as core members reaching out to those on the periphery.

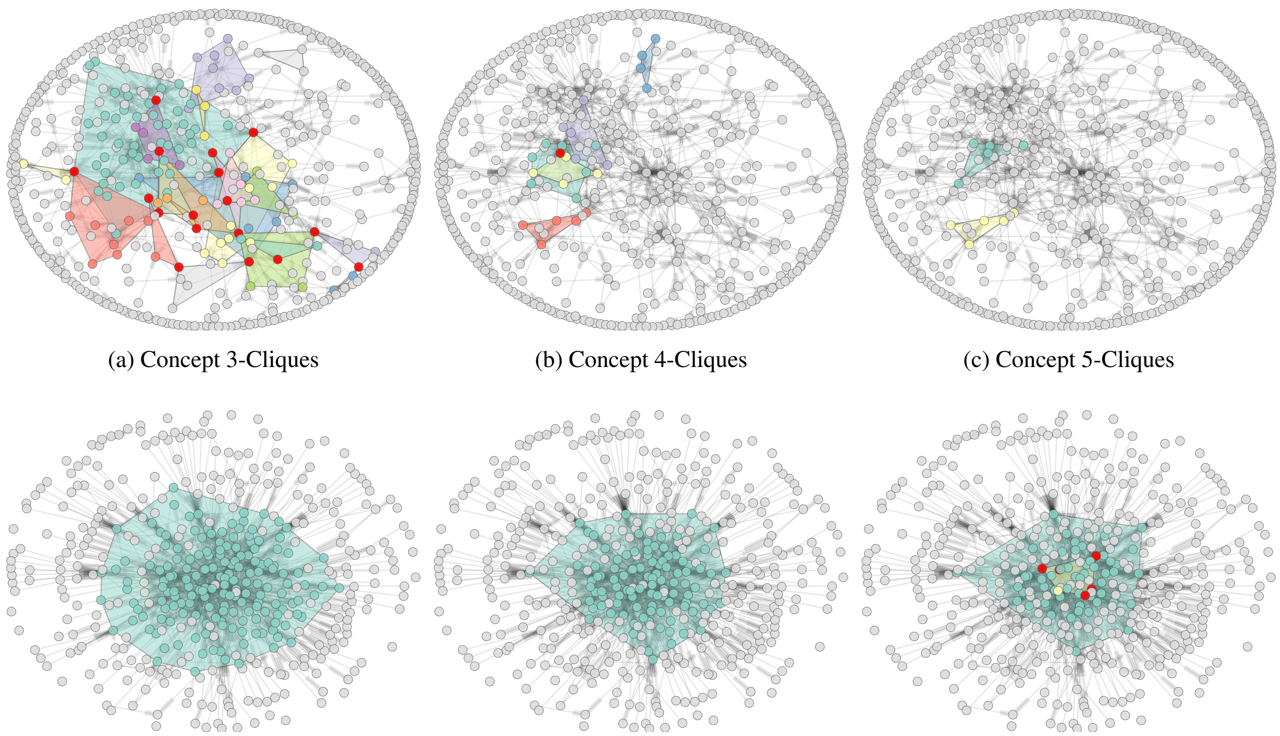
We hypothesize that this is one of the reasons for the disassortative, core-periphery structure seen in the social graph. Other possible reasons include: OpenIDEO's reputation system, which awards "collaboration points" for commenting with other people's concepts; community managers who reach out to less active users; specific stages of the design process for commenting, viewing, and evaluating the work of others; and soft incentives from IDEO that reward active users through possible job opportunities within the larger company.

While these features undoubtedly improve participation, collaboration, and disassortativity, they may not be present in other design networks. We encourage this behavior as a means to increase network efficiency, decrease clustering, and improve idea generation—a recommendation we return to in section 4.

K-Clique Percolation (Figures 5&6): To uncover any possible community structures, we used the Clique Percolation Method [17] to detect communities of different sizes and overlap. This approach constructs k-cliques and then merges k-cliques together if they share k-1 nodes in common, identifying larger communities. For example, a 2-clique would be any 2 connected node pair, and a 2-clique community would merge any pairs which shared at least 1 node in common—this special case would be the same as finding the connected components of the graph. By increasing k, you can uncover increasingly connected communities within the graph.

Figure 5 compares the number of k-clique communities for each type of graph as k is increased. The concept graph contains many 3 and 4-clique communities, but none larger than 5. The social graph contains, on average, 1-2 communities, but becomes a well-connected central community as k increases.

To characterize what these communities look like, Fig. 6 plots a representative example from challenge 10 that compares the identified k-clique communities as k is increased. In the concept graph, as k increases we see several mostly non-overlapping communities form throughout different parts of the graph—this demonstrates patches of interrelation between small collections of different concepts. In contrast, the social graph starts with a large, central commu-



(a) Concept 3-Cliques (b) Concept 4-Cliques (c) Concept 5-Cliques

(d) Social 3-Cliques (e) Social 4-Cliques (f) Social 5-Cliques

Fig. 6: Visualizing the communities created using the k-Clique Percolation Method, for different values of k in both the social and concept graphs [14]. This uses the networks from challenge 10 as a representative example. Colored sub-graphs represent nodes within a given community, and red nodes represent nodes in multiple communities. For the concept graphs(6a-6c), multiple, non-overlapping communities are present at different community scales(k=[3,5]). However, for the social graphs there is generally only a single core community—any additional communities tend to be heavily overlapping (*e.g.*, the red nodes in 6f)

nity incorporating most of the network core. As k increases, the core remains, decreasing somewhat in size. Any new communities that form have substantial overlap with the existing central core, rather than forming on a different portion of the graph—this is again consistent with the notion of the social graph maintaining a core-periphery structure.

We conducted an additional verification of these results using a different method proposed by Sarkar and Dong [7], however, the conclusions were the same so we omitted the results in the interest of space. Interested readers are welcome to download supplemental results of additional analyses.[4]

3.4 Effect of OpenIDEO Community Managers

One hypothesis for some of the observed behavior is that the OpenIDEO community managers could be purposefully acting within the network to produce these structures, and that removing them from the graph would better resemble a standard social network model. To test this hypothesis, we removed those users, and any of their links, from the social graphs across all challenges and re-ran all of our above analyses. Almost all of our results remain unchanged.

Figure 7 compares the two most substantive changes: (7a) demonstrates that removing the OpenIDEO community managers increases the assortativity of the social graph, though it still remains significantly disassortative; (7b) demonstrates that the centralization of the network decreases substantially. Given that the role of the OpenIDEO community managers is to reach out and involve different members, it is not surprising that their actions change both assortativity and centralization. What *is* surprising is that, even devoid of the community managers' comments, OpenIDEO's social graph remains disassortative and still somewhat centralized. We discuss possible cause and implications of this next.

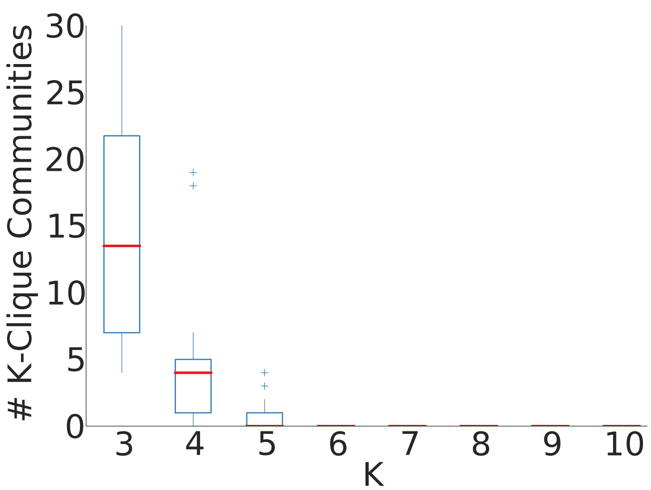
4 Implications for Design Practice and Research

Our interpretations of these results fall into two categories: implications for operators of design collaboration networks, and implications for theoretical models of design networks.

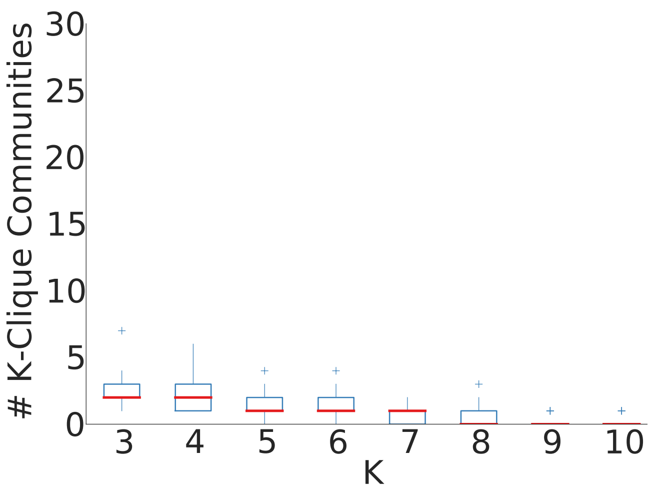
4.1 Implications for Operators of Design Collaboration Networks

Low efficiency and high diameter reduce information flow in the concept graphs. Since concept nodes represent ideas and links represent information flow in the form of building off of ideas, the way concept graphs are evolving into distributed, low efficiency networks leads to a couple of

[4]http://www.markfuge.com/openideo

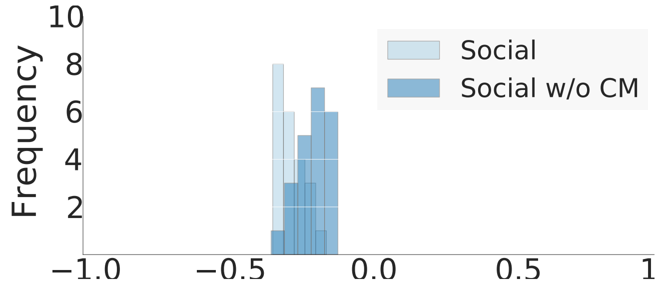


(a) Concept Graphs

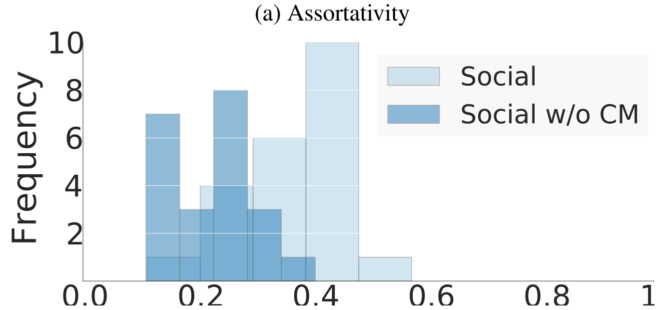


(b) Social Graphs

Fig. 5: Boxplots of the number of communities detected using the k-Clique Percolation Method, for different values of k in both the concept(5a) and social graphs(5b) [14]. The concept graphs have a high number of small communities, while the social graphs have only a few communities that are significantly more connected. This reinforces the visual data in Fig. 6.



(a) Assortativity



(b) Degree Centralization

Fig. 7: Removing OpenIDEO community managers from the social graph (“Social w/o CM”), we see some noticeable, but small changes: the centralization of the network decreases and the assortativity increases. The general behaviors we described above are unlikely to be caused exclusively by existing OpenIDEO community managers.

possible conjectures: (1) the vast majority of concepts lack useful information, and thus are not worth building off of; (2) it is difficult to find and connect disparate concepts, leading to only minor local clustering and limited global structure; or (3) the time frame or format of concept submission is such that it does not provide sufficient time to review, connect, and cycle through iterations of concepts on the network.

Addressing (1) is outside of our scope, but (2) and (3) could be addressed by employing many of the techniques we have used in our network analysis: locating ideas from distance parts of the graph to present to participants as possible idea “mash-ups” or using community detection techniques to identify or create common idea groupings.

Incentivizing core-periphery social structures increases robustness, centralization, and efficiency. While the core-periphery social structure was different than expected, it carries with it several advantages and trade-offs that help make the design network more robust and stable:

(1) Core-periphery networks are more robust to random or targeted node loss than other power-law distributed network types of similar efficiency [3, 15, 16]. This is good since open innovation networks are reliant on voluntary participation by individual nodes, any of which could stop participating at any moment.
(2) The core-periphery structure is conducive to high centralization and network efficiency, which helps transfer information among collaborators.
(3) The disassortative mixing creates an inclusive environment for periphery users to get involved and move towards the core.

As a proactive strategy for strengthening design networks, we recommend incentivizing disassortative behavior by asking high-degree core members to comment or collaborate with periphery members more regularly (a practice currently employed by the OpenIDEO community managers).

However, a highly clustered central core may harm ideation potential. The primary concern with highly clustered core networks is that, when used to communicate ideas or concepts, it may impede idea generation. Highly clustered, inefficient networks facilitate forms of complex contagion, or multiple repeated exposure, that can cause people to prematurely cease exploring ideas [1]. Essentially, if all your neighbors are exploring similar ideas, you are more likely to produce something similar to that idea—fixating on it in place of exploring other options. In a highly clustered network this effect feeds on itself since many people have common neighbors, creating false confidence about the strength of an idea and premature fixation on a portion of the design space.

Therein lies the double-edged sword of using a core-periphery structure as a base for a design network: it can enable a robust, self-sustaining, efficient collaboration network, but if the network becomes too clustered it can result in premature design fixation and lack of exploration ("groupthink," essentially). We recommend the following countervailing measures that maintain structure and increase the efficiency of the network while reducing clustering:

(1) Encourage core nodes to collaborate with periphery nodes—this increases disassortativity and efficiency, and lowers clustering.
(2) Expand the diversity of the contributors—this improves the overall diversity and quality of sets of ideas being discussed, regardless of network structure.
(3) Encourage the idea generation practice of first doing individual ideation before viewing the ideas of others—this limits initial exposure to potentially fixating ideas, after which members can take better advantage of the core-periphery network regardless of its efficiency or clustering.
(4) Encourage building off of ideas from different parts of the concept graph—this creates better efficiency within the concept graph and has the potential to combine distinct features from different parts or "idea communities" within the concept graph.

4.2 Implications for Theoretical Models of Design Networks

Consider explicitly modeling disassortativity in collaboration networks. The disassortative nature of the collaboration social graph is non-standard in current social collaboration models, and does not appear in datasets from nearby domains like Open Source Software. We would encourage those working on theoretical or simulation models of design team collaboration to consider including disassortativity characteristics as part of their modeling strategy.

We need better understanding and models of core-periphery structures. Research in core-periphery structures is still an active area of research [15,16]—there is much to be gained by collaborating with other researchers working in network analysis. As an example, we presented an initial exploration of the role of the community managers in section 3.4—much more work could be done to explore the possibilities for network interventions in design collaborations. A natural extension of this work would be exploring the structural effect of pairing new periphery members with existing core members or recommending concepts from different parts of the concept graph.

5 Conclusion

This paper presented an empirical study of OpenIDEO, a real-world design innovation network. Through the use of network analysis techniques, we found that OpenIDEO's social graph is disassortative and lacks the multiple community structure found in typical social networks. Instead, the graph contains a moderately centralized core-periphery structure that is robust to network attacks. This could be caused by multiple factors, including: size, presence of community member leads, and collaboration incentives, though further study would be necessary to determine causal relationships.

While the efficiency and robustness benefits of the social graphs' structure are advantageous, there is the possibility for design fixation through complex contagion if the core network becomes too clustered. We discussed possible counter-strategies including increased community involvement with periphery nodes and increased participant diversity. We also addressed how these structures might impact theoretical models of design networks, specifically the need to model disassortative collaboration behavior and core-periphery structures.

Moreover, this work raises several new questions for future investigation: At what point do transitions occur between single and multi community network structures? How do these design structures change over time, as people begin to develop reputations within the community? How do you balance network efficiency with the desire to help members exploit the good ideas of others? By using network analysis techniques to better understand their structure and operation, this paper helps further the potential of online communities, by providing insight into what types of behavior make them sustainable and effective.

6 Acknowledgments

We would like to thank the OpenIDEO team and community members, without whose efforts this work would not have been possible. We would also like to thank the anonymous reviewers for their help improving the manuscript. This work was supported by NSF CMMI-1334361 and the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program.

References

[1] Mason, W., and Watts, D. J., 2012. "Collaborative learning in networks". *Proceedings of the National Academy of Sciences,* **109**(3), Jan., pp. 764–769.
[2] Stephen, A. T., Zubcsek, P. P., and Goldenberg, J., 2013. "Product Ideation in Social Networks". *Social Science Research Network Working Paper Series*, Mar.

[3] Newman, M., 2010. *Networks: An Introduction*. Oxford University Press, Mar.
[4] Scott, J., 2011. "Social network analysis: developments, advances, and prospects". *Social Network Analysis and Mining,* **1**(1), Jan., pp. 21–26.
[5] Madey, G., Freeh, V., and Tynan, R., 2002. "The Open Source Software development phenomenon: An analysis based on social network theory". In *Proceedings of the Eighth Americas Conference on Information Systems*. pp. 1806–1815.
[6] Xu, J., Gao, Y., Christley, S., and Madey, G., 2005. "A Topological Analysis of the Open Souce Software Development Community". In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Vol. 7, IEEE, p. 198a.
[7] Sarkar, S., and Dong, A., 2011. "Community detection in graphs using singular value decomposition". *Physical Review E,* **83**(4), Apr., pp. 046114+.
[8] Ugander, J., Karrer, B., Backstrom, L., and Marlow, C., 2011. The Anatomy of the Facebook Social Graph, Nov.
[9] Zhang, S., Hu, Y., Zhang, X., and Li, Y., 2013. "Simulation study on collaborative behaviors in mass collaborative product development". In Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on, IEEE, pp. 375–379.
[10] Panchal, J. H., 2009. "Co-evolution of products and communities in mass-collaborative product development-a computational exploration". In International Conference on Engineering Design (ICED'09).
[11] Le, Q., 2012. "Analysis and Modeling of the Product Structure and Community Structure in Open Source Processes". PhD thesis, Washington State University, Aug.
[12] Sha, Z., and Panchal, J., 2013. "Estimating the node-level behaviors in complex networks from structural datasets". In 2013 ASME Computers and Information in Engineering (CIE) Conference, ASME.
[13] Hagberg, A. A., Schult, D. A., and Swart, P. J., 2008. "Exploring network structure, dynamics, and function using NetworkX". In Proceedings of the 7th Python in Science Conference (SciPy2008), pp. 11–15.
[14] Palla, G., Derenyi, I., Farkas, I., and Vicsek, T., 2005. "Uncovering the overlapping community structure of complex networks in nature and society". *Nature,* **435**(7043), June, pp. 814–818.
[15] Csermely, P., London, A., Wu, L.-Y., and Uzzi, B., 2013. Structure and dynamics of core-periphery networks, Sept.
[16] Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J., 2013. Core-Periphery structure in networks, Apr.
[17] Fortunato, S., 2010. "Community detection in graphs". *Physics Reports,* **486**(3-5), Feb., pp. 75–174.