Graph-Theoretic Approach for Increasing Participation in Networks With Assorted Resources

Waseem Abbas, Aron Laszka, Mudassir Shabbir, and Xenofon Koutsoukos, *Fellow, IEEE*

***Abstract*—In many cooperative networks, individuals participate actively as long as they recognize a sufficient value in participation, which depends not only on the number, but also on the attributes of other participating members. In this paper, we present a generalized model of individuals' participation in such networks, and a strategy to maximize the number of participating individuals. Unlike most of the existing literature, our model incorporates both the network structure and the heterogeneity of individuals in terms of their attributes and resources. We consider that each individual possesses a subset of available resources (attributes), which it shares with neighbors as long as neighbors reciprocate and provide the missing resources to the individual. However, individual leaves the network if it cannot find all the resources in its neighborhood. To model this phenomenon, we introduce a graph-theoretic notion of the (r, s)-core, which is the sub-network consisting of only those individuals who can access all the resources by collaborating with their neighbors. Since disengagement of an individual could initiate a cascading withdrawal of more individuals from the network, one of our main goals is to prevent this unraveling and maximize the number of participating individuals. For this purpose, we utilize the notion of anchors—individuals that continue to participate (due to incentives) even if they cannot find all of the resources in their neighborhood. By introducing only a few anchors, we can significantly increase the number of participating individuals, which in our model corresponds to increasing the size of the (r, s)-core. We formulate and thoroughly analyze the anchors' selection problem by classifying the cases in which the problem is polynomial-time solvable, NP-complete, and inapproximable. Further, we provide greedy and metaheuristic search algorithms to compute a set of anchors and evaluate our results on various networks. Our results are applicable to a large number of cooperative networking applications, including participatory sensing in which users develop an elaborate knowledge of their environment through sharing measurements.**

***Index Terms*—Network structure, graph algorithms, k-core, (r, s)-core, social networks, participatory sensing.**

I. Introduction

In the broad domain of cooperative networks, it is crucial to understand how and when individuals cooperate with each other and actively participate in a network activity. At what point individuals decide to disengage themselves from the network, and how can we maximize the participation of individuals? It turns out that mutual benefit is a key to sustainable coordination among individuals in cooperative networks. Studies in social networks, behavioral economics, and sociobiology also reveal that individuals are more likely to share (their resources, information, etc.) with others in a society if others reciprocate (e.g., see [1]–[5]). This sharing induces a positive network effect and allows individuals to utilize a broad spectrum of resources available within the network. On the other hand, individuals are more likely to disengage from the network if they do not find a sufficient value in participation, or if they fail to receive the desired resources or information from the network. This behavior describes a group participation mechanism, in which individuals collaborate and contribute in a group as long as they receive a sufficient reward in terms of access to the overall group resources.

As an example, consider the phenomenon of participatory sensing, that enables users to share measurements of their environment [6]–[8], such as traffic and parking situations [9], [10], waiting times at businesses, weather information, or disaster scenarios [11]. The goal is to allow users to develop a knowledge of their environment through sharing. This knowledge is significantly more elaborate than what individuals could develop on their own, relying only on their limited sensing capabilities. A key feature of participatory sensing is that an individual's benefit from the application depends strongly on what measurements are shared by the other users. Consequently, an individual's eagerness to participate depends on the group of users already participating, which results in a cascading effect: as the size of the user base grows, other individuals become more eager to join. Since the success of a participatory-sensing application is very often measured in terms of the number of its users, finding innovative ways to maximize the number of individuals joining the application is crucial.

Manuscript received July 9, 2018; revised November 20, 2018; accepted December 16, 2018. Date of publication January 16, 2019; date of current version September 2, 2020. This work was supported in part by the National Science Foundation under Grant CNS-1238959 and in part by the National Institute of Standards and Technology under Grant 70NANB18H198. Recommended for acceptance by H. Liu. *(Corresponding author: Waseem Abbas.)*

W. Abbas is with the Electrical Engineering Department, Information Technology University, Lahore 54000 Pakistan (e-mail: w.abbas@itu.edu.pk).

A. Laszka is with the Department of Computer Science, University of Houston, Houston, TX 77004 USA (e-mail: alaszka@uh.edu).

M. Shabbir is with Computer Science Department, Information Technology University, Lahore 54000 Pakistan, and is with the Computer Science Department, Information Technology University, Lahore, Pakistan (e-mail: mudassir@rutgers.edu).

X. Koutsoukos is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37240 USA (e-mail: xenofon.koutsoukos@vanderbilt.edu).

Digital Object Identifier 10.1109/TNSE.2019.2893584

In this paper, we model the participation of individuals in a network, each of which has a certain type of resources (capabilities, information, attributes), as a graph-theoretic problem. An individual shares (personal) resources with friends in a network, and in return expects them to reciprocate by providing the resources not possessed by the individual. If friends in the network fail to cater the missing resources, the individual disengages from the network, thus reducing the size of the overall network. As a result of a disassociation of the individual from the network, one of the friends that might be depending on the individual for a particular resource might also leave the network due to the unavailability of the desired resources. This phenomenon could lead to a cascading effect involving a subsequent disengagement of many more individuals. At the end, we are left with a small network in which every individual, as a result of collaboration with friends, has an access to all types of resources available within the network. We introduce the notion of (r, s)-*core of the network* to model this participation strategy. If r is the total number of various resources available in the network, and each individual possesses at most s of them, then the (r, s)-core represents the sub-network in which every individual finds all r resources between himself and friends (neighbors).

As we observe in participatory sensing, it is desired to maximize the number of active users. In other words, we need to modify or design the network so as to maximize the size of its (r, s)-core. A key factor here is to prevent the unraveling of the network by averting the cascading removal of individuals. To achieve this objective, we utilize the notion of *anchor nodes* inspired by the work of Bhawalkar et al. [12]. In our work, anchors are the individuals that continue to participate in the network even if they do not find the desired resources in their neighborhood, i.e., whose participation in the network does not depend on the attributes of their peers. A small subset of individuals in a network can be made anchors by incentivizing them, for example, by providing them with rewards for their participation. We show that by incentivizing a small but strategically selected set of individuals can lead to a very large number of users following them and joining the network, thus, resulting in a significantly larger (r, s)-core. Finding such a small set of anchors that maximize the (r, s)-core of the network is a computationally challenging problem. We analyze the complexity of this problem in detail, and present heuristics to find anchors that significantly increase the size of the (r, s)-core of the network.

We summarize our main contributions below:

- We introduce the notion of (r, s)-core to model the participation of individuals in a network. Our model is general in the sense that it incorporates the participation mechanism in which individual's decision to participate in the network depends not only on the number of friends (neighbors), but also on their types and attributes (resources, capabilities).
- To maximize the number of participants in a network and increase the size of (r, s)-core, we propose to incentivize few individuals called anchors. We show that the size of (r, s)-core is significantly increased by having few anchors in a network that are strategically selected. We formulate the problem of finding a given number of anchors that maximize the size of the (r, s)-core, and provide an integer linear program (ILP) for the anchor selection problem.
- We analyze the complexity of the anchors selection problem in detail. We show that in arbitrary graphs the problem can be solved in polynomial time in the special case of $r = s + 1$ (Theorem 5.1). For any other value of r, the problem is NP-complete (Theorem 5.2). In fact, we show strong inapproximability results for general graphs and $r \geq s + 4$ (Theorem 5.3). This detailed complexity analysis of the anchors selection problem is one of the main highlights of this work.
- We propose a greedy heuristic and a metaheuristic search algorithm based on simulated annealing to find a given number of anchors to increase the size of the (r, s)-core of the network.
- Finally, we evaluate our results numerically on a number of networks, and use simulations to demonstrate the efficacy of our approach.

We presented preliminary ideas in a brief note in [13]. The current paper provides a thorough description of the problem along with the technical details, which are not available in [13]. In fact, majority of the results here are new, including ILP formulation of the anchors selection problem, and the main results such as Theorems 5.1 and 5.3 (that classify the cases in which the anchors selection problem is polynomial time solvable, and is inapproximable). Moreover, simulated annealing based heuristic along with detailed numerical evaluation of the results are also new. We also discuss a number of extensions and further directions here.

The rest of the paper is organized as follows: Section II gives an overview of the related work. Section III introduces the network model, and formally defines the (r, s)-core of the network. Section IV presents the idea of anchors to maximize the size of the core. It presents the anchor selection problem, and also provides an ILP formulation of the problem. Section V analyzes the complexity of the anchors selection problem in detail, showing that the problem is computationally challenging in general. Section VI proposes heuristics to select a given number of anchors to maximize the core, and Section VII provides a generalization of the (r, s)-core. Section VIII presents a numerical evaluation of our approach. Section IX provides various ideas and directions to further extend this work.

II. Related Work

To describe the norms and mechanisms that explain coordination in societies, prosocial behaviors, and reciprocity for the mutual benefit of the network members, various theories and ideas have been put forward in the social sciences, behavioral sciences and psychology literature. A good account of the dilemmas of social cooperation and an overview of the human cooperation mechanisms is presented in [3], [14]. It is reported that reciprocity plays a key role in defining human cooperation. Based on various aspects including when and with whom to cooperate, different notions of reciprocity have been specified,

such as direct reciprocity [15], [16], indirect reciprocity [1], and generalized reciprocity [2], [17]. Individuals help others in a social set up as long as they also receive help from the other members of the society. Another related phenomenon is of social cohesion [18], in which individual's decision to participate in a group and retain its membership, has a significant impact on the overall cohesiveness of the group.

Various mathematical models, for instance as in [12], [19]–[22], characterize the processes by which groups get together, grow by engaging new members, or diminish as a result of defection of members. In a recent paper, a mathematical model of the evolution of a social network based on the notion of social capital is presented in [23]. Threshold models of collective behavior (e.g., [24], [25]) hold a central position in this domain. The basic premise is that individual's behavior and decision to stay or leave a network depends on a certain number of other individuals (threshold) engaged in a similar behavior. The k-core of the network, introduced in [26], is one such widely studied and applied model. It asserts that an individual continues to participate in a network if at least k of the individual's friends are also participating. In graph-theoretic terms, it means that a node with degree less thank k is removed from the graph, and k-core is the maximal subgraph in which each node has degree at least k. k-core and its extensions, including defining a distinct threshold k for each node [27], [28], k-core in weighted graphs [29], directed graphs [30], and various others have been extensively studied in the context of social networks (e.g., [31]–[33]), as well as other networks (e.g., see [34] and the references therein).

Disengagement of one member from the network could initiate a cascaded withdrawal of others. To prevent this unraveling in social networks, the idea of anchor nodes has been presented in [12] in the context of k-core. The authors showed that the size of the k-core can be increased through anchors – nodes that remain engaged even if they have less than k friends. Our work is related to [12] in this direction, as we also utilize the notion of anchors to maximize the size of (r, s)-core. However, a major limitation of the k-core based participation model is that it assumes participation to be determined solely by the network structure, ignoring the heterogeneity of users and their attributes. In many realistic scenarios, such as in participatory sensing, the heterogeneity of users plays a key role. For example, users may take measurements at different geographical locations, and they may have devices with different sensing capabilities. To develop a more complete knowledge of the environment, these heterogeneous measurements must be combined. As a consequence, a users' benefit from a participatory-sensing application depends not only on the number of participating peers, but also on the heterogeneous nature of the measurements shared by them. To account for heterogeneity, we introduce the concept of (r, s)-core, which incorporates the attributes (resources, capabilities, information) of users also.

Finally, the anchors selection problem is closely related to finding the most influential nodes in social networks that could instigate the cascading removal of nodes as studied in [35]. Based on a network model, there are many variants and extensions of the problem (e.g., [36]–[39]) along with detailed complexity results [40]–[43]. We note that similar to the notion of anchors, the idea of having some 'special nodes' that are more resilient to structural changes in the network, have been used previously in other contexts, for instance to increase the connectivity and structural robustness of the network [44], [45]. In this paper, we study the anchors selection problem to maximize the (r, s)-core of the network, which is an elaborate model of users participation in a network of individuals with assorted resources and attributes. We analyze the complexity of the problem in detail and outline heuristics to compute anchors.

III. Network Model and (r, s)-Core

We model the network by a simple, undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which the node set $\mathcal{V}$ represents the set of individuals, and the edge set $\mathcal{E}$ represents connections between them. Any two nodes $x, y \in \mathcal{V}$ are *adjacent* in $\mathcal{G}$ if an edge exists between them, for instance, if the corresponding individuals share measurements with each other. A set of nodes that are adjacent to $x \in \mathcal{V}$ is called the *neighborhood* of x, and is denoted by $N(x)$. Similarly, we define the *closed neighborhood* of x as $\mathcal{N}[x] = \mathcal{N}(x) \cup \{x\}$. Each node of the graph has attributes, which model the specific set of resources, sensing capabilities, measurements, or information contained by the corresponding individual. A node shares its attributes with its neighbors, and make them accessible to the neighbors. We model these attributes by a *label set* containing r distinct labels, that is $\mathcal{R} = \{0, 1, 2, \ldots, r-1\}$, and assign a subset of $s \leq r$ labels to each node in $\mathcal{G}$, depending on the attributes of the node. As a result, we have a (node) labeled graph that captures the sharing phenomenon between individuals with different attributes. We denote labels assigned to node x by $\ell(x)$, i.e.,

$$\ell: \quad \mathcal{V} \longrightarrow [\mathcal{R}]_s. \tag{1}$$

Here, $[\mathcal{R}]_s$ is the set of all subsets of $\mathcal{R}$ having exactly s elements.

Node Participation Rule – A node (individual) participates in the network on the *reciprocity* principle, that is, it shares its resources with neighbors in the network, and in return expects to receive the missing set of resources from the neighbors. In particular, a node *participates* (or engages) in the network as long as its neighbors provide all the labels that are missing from the node's own label set. Formally, a node x participates in the network as long as the following condition is satisfied:

$$\bigcup_{y \in \mathcal{N}[x]} \ell(y) = \mathcal{R}. \tag{2}$$

This engagement rule models networking phenomena in which a node continues to participate in the network and shares its information as long as it acquires all the missing information from the neighbors. If the condition in (2) is not satisfied for a node, then the node simply leaves the network. We note here that there could be other variants and extensions of the above participation rule depending on the networking

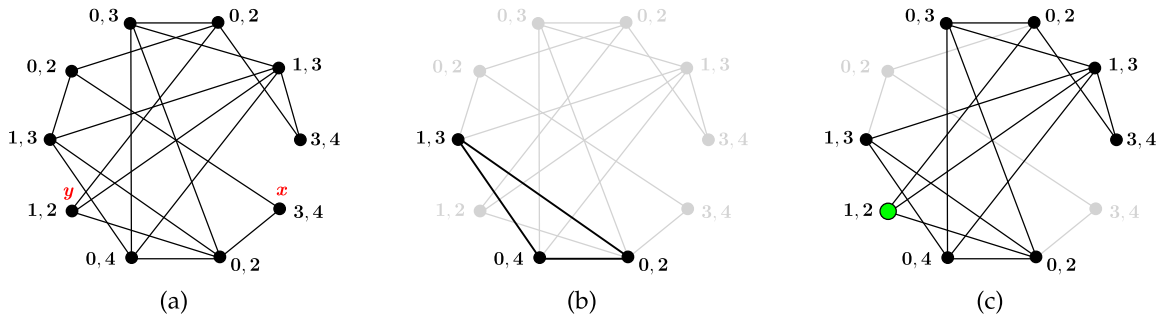


Fig. 1. (a) Example network. (b) (5,2)-core of the example network. (c) (5,2)-core with an anchor node (marked green).

applications. We further discuss these variations and generalizations in Section IX.

A. (r, s)-*Core of the Network*

Consider a network in which each node has a subset of s labels from the set of r labels, and nodes participate as per engagement rule (2). In such a setup, a node leaving the network can have a cascading effect as it may further cause its neighbors to depart. For instance, consider a node x with a label $a \in \ell(x)$, and let $y \in \mathcal{N}(x)$ be such that x is the only node in $\mathcal{N}[y]$ with the label a. Then, node x leaving the network will also result in node y leaving. Thus, the removal of a node from a network may cause a cascading effect, or *unraveling*, due to which nodes that initially satisfy the condition (2) may also get removed from the network. In the end, we are left with a subnetwork consisting of nodes all of which satisfy the participation rule (2). We call this remaining network as the (r, s)-*core* of the original network. More precisely, we define (r, s)-core as follows:

(r, s)-***Core*** Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a set of r labels, denoted by $\mathcal{R}$, and an assignment $\ell : \mathcal{V} \to [\mathcal{R}]_s$ (that is, assigning s labels from $\mathcal{R}$ to each node $v \in \mathcal{V}$), the *(r, s)-core of $\mathcal{G}$*, denoted by $\tilde{\mathcal{G}}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$, is the maximal subgraph in which every node satisfies

$$\bigcup_{y \in \left(\mathcal{N}[x] \cap \tilde{\mathcal{V}}\right)} \ell(y) = \mathcal{R}, \quad \forall x \in \tilde{\mathcal{V}}. \tag{3}$$

Note that it follows from the definition readily that every labeled graph has a unique (r, s)-core.

Example – As an example, consider the network shown in Fig. 1a. This network has ten nodes, each of which has $s = 2$ labels from the set $\mathcal{R} = \{0, 1, 2, 3, 4\}$. Initially, there are two nodes x and y that do not satisfy the condition (2). As a result, they leave the network, which leads to further node removals. In the end, only three nodes remain, as shown in Fig. 1b. Each of these nodes have all five labels between itself and its neighbors; thus, constituting the (5,2)-core of the network.

IV. Anchored (r, s)-Core Problem

The (r, s)-core represents individuals that actively participate in the networking application, for instance, individuals in a participatory-sensing application that share measurements with their peers. To increase the participation of individuals, we desire to increase the size of the (r, s)-core. Thus, from a design perspective, the following question arises.

For a given network $\mathcal{G}$, label set $\mathcal{R}$, and a positive integer s, how can we modify or design our network so as to maximize the size of its (r, s)-core?

One way to achieve this objective is to find an *optimal assignment* of labels to nodes, that is, a labeling ℓ defined in (1) that maximizes the size of the (r, s)-core. However, in some situations, the labeling ℓ is fixed, that is, the labels assigned to nodes are given and cannot be changed. For example, in participatory sensing, users can have devices with fixed sensing capabilities that cannot be easily changed. A new approach is needed in these situations to increase participation. In this paper, we explore the idea of significantly increasing participation by incentivizing a few selected individuals to participate regardless of their peers' attributes.

The departure of a node from the network may lead to the departure of its neighbors, thereby causing a cascading phenomenon. The size of the (r, s)-core of the network can be larger if we can prevent this cascading effect. By ensuring the participation of few individuals, even if they do not satisfy the participation rule, we can prevent the unraveling of the network. We call such individuals as *anchors*.

Anchors Anchors are the nodes that never leave the network and participate irrespective of the labels assigned to them or to their neighbors. In other words, they continue to participate even if they do not satisfy the rule in (2).

Individuals can be made anchors by offering them rewards for their participation when condition (2) is not satisfied. We now define the (r, s)-*core with anchors* as follows:

(r, s)-***Core with Anchors*** Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a set of r labels, denoted by $\mathcal{R}$, an assignment $\ell : \mathcal{V} \to [\mathcal{R}]_s$ that assigns s labels from $\mathcal{R}$ to each node $v \in \mathcal{V}$, and a set of anchor nodes $\mathcal{A} \subseteq \mathcal{V}$, the (r, s)-*core with anchors* $\mathcal{A}$ is the maximal subgraph $\tilde{\mathcal{G}}_{\mathcal{A}}(\tilde{\mathcal{V}}_{\mathcal{A}}, \tilde{\mathcal{E}}_{\mathcal{A}})$ consisting of all anchor nodes as well as non-anchor nodes satisfying

$$\bigcup_{y \in \mathcal{N}[x] \cap \tilde{\mathcal{V}}_A} \ell(y) = \mathcal{R}, \quad \forall x \in \tilde{\mathcal{V}}_{\mathcal{A}} \setminus \mathcal{A}. \tag{4}$$

Note that it follows from the definition readily that for every labeled graph and set of anchor nodes, the (r, s)-core with anchors exists uniquely. An example of (r, s)-core with an

anchor node is shown in Fig. 1c. Without any anchor node, the maximum subgraph in which each node has all five labels between itself and its neighbors consists of only three nodes, that is, the size of the (5,2)-core is three, as shown in Fig. 1b. However, with only one anchor node, we obtain a subgraph consisting of eight nodes in which each non-anchor node has a complete set of five labels between itself and its neighbors. Thus, we see a significant improvement in terms of users participation even with a single anchor.

Next, we study the problem of maximizing the size of (r, s)-core by selecting an appropriate set of anchor nodes $\mathcal{A}$. Formally, we state the problem as:

(r, s)-Core with Anchors Maximization Problem (CAMP) Given a node labeled graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which each node is assigned a subset of s labels from a set of r labels, and a budget b, then the (r, s)-Core with Anchors Maximization Problem is to find a subset $\mathcal{A} \subseteq \mathcal{V}$ such that the number of nodes in the anchored (r, s)-core is maximized over all sets $\mathcal{A} \subseteq \mathcal{V}$ of size $|\mathcal{A}| \leq b$.

In Section V, we analyze the complexity of this problem in detail. First, we outline an integer linear program to solve the CAMP.

A. An ILP for the (r, s)-Core With Anchors Maximization Problem

An ILP formulation of the CAMP is as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_{u \in \mathcal{V}} x_u, \\
\text{subject to} \quad & \sum_{u \in \mathcal{V}} z_u \leq b; \\
& \sum_{v \in \mathcal{N}[u]} x_v * w(v, c) + z_u \geq x_u; \ \forall u \in \mathcal{V}, c \in \mathcal{R} \\
& x_u, z_u \in \{0, 1\}; \ \forall u \in \mathcal{V} \\
& w(c, v) \in \{0, 1\}; \forall v \in \mathcal{V}, c \in \mathcal{R}.
\end{aligned}$$

The variable z_u represents whether the node u is anchored or not, and the variable x_u represents whether the node u is included in the (r, s)-core or not. The goal is maximize the number of nodes in the core. The first constraint shows that the number of anchors is bounded by the budget b. The second constraint makes sure that for each node $u \in \mathcal{V}$ and for each label $c \in \mathcal{R}$, if $x_u = 1$ (that is u is in the core), then u is either an anchor node, or u must have c in its closed neighborhood. Here, $w(v, c)$ is an indicator variable for whether the label c is in the label set of node v, that is $w(v, c) = 1$ if $c \in \ell(v)$ and 0 otherwise. Note that if node u is anchor ($z_u = 1$), then the corresponding $x_u = 1$ and u is included in the core.

V. Complexity Results

In this section, we discuss in detail the complexity of the (r, s)-core with anchors maximization problem. In particular, we show that for any positive integer s, the problem can be solved in polynomial time for a special case of $r = s + 1$ (Theorem 5.1). For all other $r > s + 1$, the problem is NP-complete (Theorem 5.2). In fact, for $r \geq s + 4$, we prove a strong inapproximability result (Theorem 5.3). The main results of this section are stated below.

Theorem 5.1: For any positive integer s, the $(s + 1, s)$-core with anchors maximization problem can be solved in $\mathcal{O}(N + sE)$, where N and E are the number of nodes and edges in $\mathcal{G}(\mathcal{V}, \mathcal{E})$ respectively.

Theorem 5.2: Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a positive integer s, a set of $r = s + j$ labels where $j > 1$, a labeling ℓ, a number of anchor nodes b, and a threshold core size δ, determining if there exists a set $\mathcal{A}$ of at most b anchor nodes that results in an anchored (r, s)-core whose cardinality is at least δ is an NP-complete problem.

Theorem 5.3: For any positive integer s and $j \geq 4$, it is NP-complete to approximate the $(s + j, s)$-core with anchors maximization problem within any factor $\mathcal{O}(N^{1-\epsilon})$ for $\epsilon > 0$, where N is the number of nodes in the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

We prove the above results in Sections V-A, V-B, and V-C respectively.

A. Maximization of the Anchored $(s + 1, s)$-Core

Whenever the total number of labels is only one more than the number of labels assigned to each node, then every node misses at most one label in its closed neighborhood. An important observation in this special case is outlined below.

Lemma 5.4: For any positive integer s and a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, let $\tilde{\mathcal{V}}$ be the set of nodes in the $(s + 1, s)$-core of $\mathcal{G}$ with no anchors, and $\tilde{\mathcal{V}}_{\mathcal{A}}$ be the set of nodes in the $(s + 1, s)$-core of $\mathcal{G}$ with one anchor node $\mathcal{A} = \{x\}$, where $x \in \mathcal{V} \setminus \tilde{\mathcal{V}}$, then $\tilde{\mathcal{V}}_{\mathcal{A}} = \tilde{\mathcal{V}} \cup \{x\}$.

Proof: For the sake of contradiction, we assume that by making x an anchor node, another node $y \in \mathcal{V} \setminus \tilde{\mathcal{V}}$ is included in $\tilde{\mathcal{V}}_{\mathcal{A}}$. It implies that $y \in \mathcal{N}(x)$, and $\ell(x) \cap \ell(y) \neq \ell(y)$, where $\ell(x)$ and $\ell(y)$ are the labels assigned to x and y respectively. This is only possible when there exists a label $i \in \ell(x)$ such that $i \notin \ell(y)$, and a label $j \in \ell(y)$ such that $j \notin \ell(x)$. Since both x and y already have s labels each, and miss only a single label, their labels must complement each others, that is, $\{0, 1, \ldots, s\} \setminus \ell(x) \in \ell(y)$, and $\{0, 1, \ldots, s\} \setminus \ell(y) \in \ell(x)$. It means that both x and y have $(s + 1)$ labels in their closed neighborhoods and are included in $\tilde{\mathcal{V}}$ which constitutes the $(s + 1, s)$-core with no anchors, which is a contradiction. ■

1) Proof of Theorem 5.1: From Lemma 5.4, we get that the size of the $(s + 1, s)$-core with anchors $\mathcal{A}$ is simply $|\tilde{\mathcal{V}} \cup \mathcal{A}|$, where $\tilde{\mathcal{V}}$ is the set of nodes in the $(s + 1, s)$-core (with no anchors). Thus, to prove Theorem 5.1, all we need to show is that $(s + 1, s)$-core of a labeled graph can be computed in $\mathcal{O}(N + sE)$, which is indeed the case using the algorithm outlined below.

Let (u, v) be a *monochromatic* edge, that is an edge between u and v such that $\ell(u) = \ell(v)$. Then we observe that $\bigcup_{i \in \mathcal{N}[v]} \ell(i) = \bigcup_{i \in (\mathcal{N}[v] \setminus \{u\})} \ell(i)$, and similarly $\bigcup_{i \in \mathcal{N}[u]} \ell(i) = \bigcup_{i \in (\mathcal{N}[u] \setminus \{v\})} \ell(i)$. Hence, removing edge (u, v) doesn't affect the $(s + 1, s)$-core of given $\mathcal{G}$. Once we have removed all monochromatic edges, all vertices with at least one neighbor will be included in the $(s + 1, s)$-core. Checking whether an edge is

monochromatic or not takes $s.E$ time, while enumerating all vertices with at least one neighbor can be achieved in N time. This completes the proof of the theorem.

B. Hardness of Anchored $(s+j,s)$-Core for $j>1$

The membership in NP is obvious because the size of the anchored $(s+j,s)$-core can be verified in polynomial time. In the following, we show that for any $j>1$, the $(s+j,s)$-core with anchors maximization problem is computationally hard using a reduction from a well-known NP-hard problem, the Set Cover Problem.

Set Cover Problem (SCP). Given a base set $\mathcal{U}=\{1,2,\ldots,m\}$, a collection $\mathcal{F}=\{S_1,S_2,\ldots,S_n:\ S_i\subseteq\mathcal{U}\}$ where $\bigcup_{S_i\in\mathcal{F}}=\mathcal{U}$, and a number σ; determine if there exists a subcollection $\mathcal{F}'\subseteq\mathcal{F}$ of at most σ subsets such that every element of $\mathcal{U}$ is contained by at least one subset in $\mathcal{F}'$.

Proof of Theorem 5.2: Given an instance of the Set Cover Problem, we construct an instance of the anchored $(s+j,s)$-core problem for $j>1$ as follows:

- $\mathcal{R}=\{0,1,\ldots,s,s+1,\ldots,s+j-1\}$, and $b=\sigma$;
- for every $S_i\in\mathcal{F}$, there is a node w_i with labels $\{0,1,\ldots,s-1\}$;
- for every $u\in\mathcal{U}$, there is a node v_u with a set of s labels $\{1,2,\ldots,s\}$;
- there exists a clique of $|\mathcal{R}|=s+j$ nodes, denoted by K_{s+j}, where each node $o_i\in K_{s+j}$ is assigned a set of labels $\ell(o_i)=\{i-1\}\cup Q_i$, where Q_i is a subset of $s-1$ arbitrarily picked labels from the set $\mathcal{R}\setminus\{i-1\}$.
- for every $u\in\mathcal{U}$ and $S_i\in\mathcal{F}$, the corresponding nodes v_u and w_i are adjacent if and only if $u\in S_i$;
- every v_u is adjacent to all the clique nodes in the set $\{o_{s+2},o_{s+3},\ldots,o_{s+j}\}$. Note that as a result of these connections, each v_u has a set of labels $\{1,2,\ldots,s\}\cup\{s+1,s+2,\ldots,s+j-1\}$ available in its closed neighborhood.
- $\delta=b+|\mathcal{U}|+|\mathcal{R}|$.

The above reduction can be clearly carried out in time that is polynomial in the size of the Set Cover Problem instance. Hence, it remains to show that the Set Cover Problem has a solution if and only if the anchored $(s+j,s)$-core problem does.

First, if there exists a set cover $\mathcal{F}'$ containing σ subsets of $\mathcal{U}$, then the anchor set $\mathcal{A}$ consisting of nodes corresponding to those subsets in $\mathcal{F}'$ is a solution to the anchored $(s+j,s)$-core problem. To see this, consider that every node v_u (corresponding to $u\in\mathcal{U}$) is adjacent to at least one node in $w_i\in\mathcal{A}$, which provides label 0 to each v_u in its neighborhood. Moreover, we note that $\ell(v_u)=\{1,\ldots,s\}$, $\forall v_u$, and labels in $\{s+1,\cdots,s+j-1\}$ are available in the closed neighborhood of each v_u by its connections with the nodes of in K_{s+j}. Thus, the anchored $(s+j,s)$-core includes all v_u, $\forall u\in\mathcal{U}$. Since all nodes in the clique K_{s+j} have all $\mathcal{R}$ labels in their closed neighborhoods, they are also in the anchored $(s+j,s)$-core, which implies that the size of the anchored $(s+j,s)$-core is $b+|\mathcal{U}|+|\mathcal{R}|$ (after including b anchor nodes in $\mathcal{A}$).

Second, the other direction (that is, proving that any solution $\mathcal{A}$ to the anchored $(s+j,s)$-core problem is also a set cover) follows from a similar argument.

C. Inapproximability of the Anchored $(s+j,s)$-Core for $j\geq 4$

To prove Theorem 5.3, first we show that it is NP-hard to approximate the problem of maximizing the anchored (5,1)-core. Then, using this result, we show that it is always possible to get an instance of maximizing the anchored $(s+j,s)$-core for $j\geq 4$ from an instance of maximizing the anchored (5,1)-core, thus, implying Theorem 5.3.

Proof of Theorem 5.3: The membership in NP is trivial to see as given a subset $\mathcal{A}$, we can always confirm the size of the anchored $(s+j,s)$-core by iteratively removing the nodes that are not in the anchored $(s+j,s)$-core in polynomial time. Next, we state and prove an important result required to prove Theorem 5.3.

Theorem 5.5: It is NP-Hard to approximate the (5,1)-core with anchors maximization problem within any factor $O(N^{1-\epsilon})$ for $\epsilon>0$.

Proof: We show this by providing a gap-based reduction from the *Set Cover problem*. An instance of the set cover problem consists of a set $\mathcal{U}=\{1,2,3,\ldots,m\}$ and a family of sets $\mathcal{F}=\{S_1,S_2,\ldots,S_n:S_i\subseteq\mathcal{U}\}$ such that $\bigcup_{S_i\in\mathcal{F}}S_i=\mathcal{U}$. The problem is to decide whether there exists a subfamily $\mathcal{F}'\subseteq\mathcal{F}$ of b sets such that $\bigcup_{S_i\in\mathcal{F}'}S_i=\mathcal{U}$. In the following, we describe a construction of a graph Γ as an instance of the anchored (5,1)-core problem given a set $\mathcal{U}$, a family $\mathcal{F}$, and set cover budget b. We assume that m is a multiple of 5 – otherwise, we can add upto four extra elements to $\mathcal{U}$ and to all of S_i without any consequence. The graph Γ will have three main parts:

- a grid with $m\times M$ nodes where $M>>m$,
- n complete tertiary (3-ary) trees, and
- a sink.

■

These parts will be connected with some auxiliary edges as detailed below. For an ease of presentation, we assume that the five labels are $\{0,1,2,3,4\}$.

The Grid:

For each $i\in\mathcal{U}$ construct a path $v_{i,1},v_{i,2},\ldots v_{i,M}$ where M should be thought of a "large" integer to be fixed later. Add edges between nodes $v_{i,j},v_{i+1,j}$ for all $1\leq i\leq m-1$ and $1\leq j\leq M$. Also add edges between nodes $v_{1,j},v_{m,j}$ for $1\leq j\leq M$. We assign label 0 to the node $v_{1,1}$, and then recursively label the rest of the grid according to the following rule: *if a node $v_{i,j}$ is labeled with the label k then $v_{i,j+1}$ is labeled with $k+1 \bmod 5$, and $v_{i+1,j}$ is labeled with $k+2 \bmod 5$.* It is easy to see that as a result of this labeling, each node (except the nodes at the boundary of the grid, that is $v_{i,j}$ where $i\in\{1,\ldots,m\}$ and $j\in\{1,M\}$) has all five labels in its closed neighborhood since $\{k,k-1,k-2,k+1,k+2\} \bmod 5=\{0,1,2,3,4\}$. Note that we constructed and labeled the nodes in the grid in such a way that

- if nodes in the first row $v_{1,1},v_{2,1}\ldots,v_{m,1}$ and the last row $v_{1,M},v_{2,M}\ldots,v_{m,M}$ are ensured to be in the core

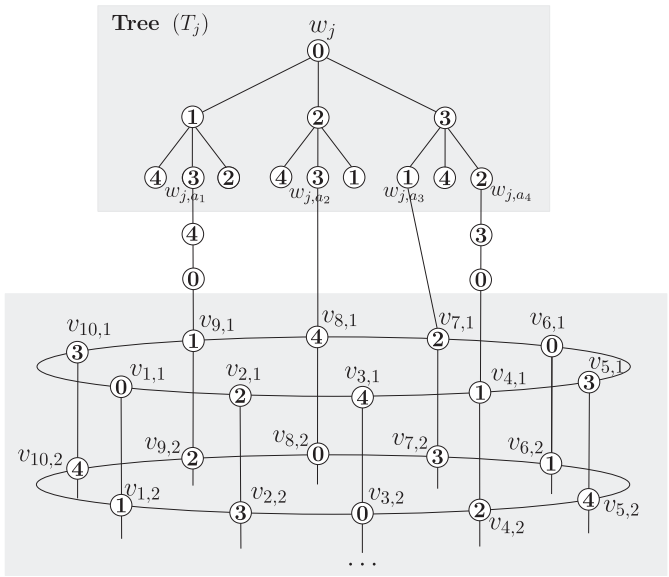


Fig. 2. An illustration of tree and its connection with the grid. Consider a set cover instance with $\mathcal{U} = \{1, 2, \cdots, 10\}$, and $S_j \subset \mathcal{F}$, where $S_j = \{4, 7, 8, 9\}$. In Γ, we have a tree T_j corresponding to S_j with a root node w_j. The four leaf nodes $\{w_{j,a_1}, w_{j,a_2}, w_{j,a_3}, w_{j,a_4}, \}$ (arbitrarily picked) correspond to four elements of S_j. Numbers inside circles represent labels assigned to the corresponding nodes.

then all the nodes in the grid $v_{i,j}$, $\forall 1 \leq i \leq m, 1 \leq j \leq M$ are also in the core.

- Moreover, if there exists a node $v_{i,j}$ in the grid that is not in the core of Γ, then none of the nodes in the grid is in the core.

The Tree: For each set S_i, we construct a tertiary (3-ary) tree T_i with height $\lceil \log_3 |S_i| \rceil$ - that is minimum height required so that the number of leaf nodes is at least the size of the set S_i. Let w_i be the root node of the tree T_i. Let $S_i = \{a_1, a_2, \ldots a_k\}$, then chose k arbitrary leaves of T_i and call them $w_{i,a_1}, w_{i,a_2}, \ldots, w_{i,a_k}$. The remaining leaves of T_i (if there are any) are of no interest to this reduction but we will make sure that they are in the core by linking them to the *sink* (to be discussed). We label each root w_i with the label 0 and its children with labels 1,2,3 in an arbitrary order. Rest of the tree is recursively labeled according to the following rule: *if a node is labeled with the label x and its parent node is labeled y, then label the three children of x with the labels $\{0, 1, 2, 3, 4\} \setminus \{x, y\}$ in arbitrary order.* We observe that this labeling ensures the following:

Observation 1:

- The end nodes of every edge have different labels.
- If the root and the leaf nodes of a tree T_j are in the core then all the nodes in the tree T_j are in the core.
- Further, if there exists a node $x \in T_j$ that is not in the core then none of the nodes in the tree are in the core (unless T_j contains anchored nodes).

An illustration is given in Fig. 2.

Connecting the Grid and Trees:

For each member $i \in S_j$, we add a path of length 1 or 3 between the leaf node $w_{j,i}$ in the tree T_j, and $v_{i,1}$ in the first row of the grid. The length of the path is determined by the labels of the two nodes. Note that in the grid, $v_{i,1}$ has three uniquely labeled neighbors and it has all labels in its closed neighborhood except some label x. If $w_{j,i}$ is labeled x then we simply add an edge between the two nodes. Otherwise we will add two intermediary nodes making a path of length 3. The neighbor of $v_{i,1}$ on this path is labeled x, and the neighbor of $w_{j,i}$ is labeled with a label y such that $y \neq x$ and is different from the labels of both $w_{j,i}$ and $v_{i,1}$. We illustrate this construction in an example in Fig. 2.

The Sink:

The sink of this construction is a clique of size 5, denoted by K_5. All nodes in this clique are labeled with unique labels to make sure that all of them are in the core. Each node u that either

(i) lies on the bottom row of the grid, that is $v_{i,M}, \forall i \in \{1, \ldots, m\}$, or
(ii) is a leaf node of a tree T_j, or
(iii) is an intermediary node on a path from a leaf node of a T_j to the grid node $v_{i,1}$,

may not see some of the labels in its closed neighborhood, and thus may not be included in the core. To make sure that any such u is in the core if all its neighbors are in the core, we add edges between u and nodes in the sink such that u has all five labels in its closed neighborhood. While doing this we also make sure that every such u finds each label in its closed neighborhood exactly once.

A detailed illustration of such a construction and Γ is given in Fig. 3. Next, we use this construction to present the following results.

Claim 5.6: If the set cover instance is a *YES* instance then the size of the (5,1)-core of the graph Γ with b anchors is at least $m \times M$.

Proof: Observe that the degree of every node $v_{i,j}$ in the grid, where $j \neq 1$, is exactly four. Moreover, four neighbors of each such $v_{i,j}$ are uniquely labeled with labels different from the label of $v_{i,j}$. Similarly all tree nodes, except the root nodes, have degrees exactly four and have all five labels in their respective closed neighborhoods. Thus, we have only two types of nodes that may have degree not equal to four. They include,

(i) the root node w_j of tree T_j, where $j \in \{1, \ldots, n\}$, and
(ii) node $v_{i,1}$ in the first row of the grid, where $i \in \{1, \ldots, m\}$.

Since root nodes of all trees have exactly three neighbors, they are not in the (5,1)-core of Γ. However, since for every $i \in \mathcal{U}$ there exists some $S_k \in \mathcal{F}$ such that $i \in S_k$ in the set cover instance, we note that $v_{i,1}$, $\forall i \in \{1, \ldots, m\}$, has a degree at least four in Γ and is included in the (5,1)-core of Γ. As the set cover instance is a yes instance, we have $\mathcal{F}' \subset \mathcal{F}$, consisting of b subsets of $\mathcal{U}$ such that their union is $\mathcal{U}$. Let $\mathcal{F} \setminus \mathcal{F}' = \{S_{j_1}, S_{j_2}, \ldots, S_{j_{n-b}}\}$, and the corresponding trees in Γ are $\{T_{j_1}, T_{j_2}, \ldots, T_{j_{n-b}}\}$. Next, we obtain a graph Γ' from Γ by removing all the nodes in trees $\{T_{j_1}, T_{j_2}, \cdots, T_{j_{n-b}}\}$. Now, in Γ', we have b trees and we can ensure that their roots are included in the anchored (5,1)-core by making them the anchor nodes. It is clear that all the non-anchor nodes in Γ' have all five labels in their closed neighborhoods, and are thus in the anchored (5,1)-core. Γ' contains a grid of $m \times M$ nodes

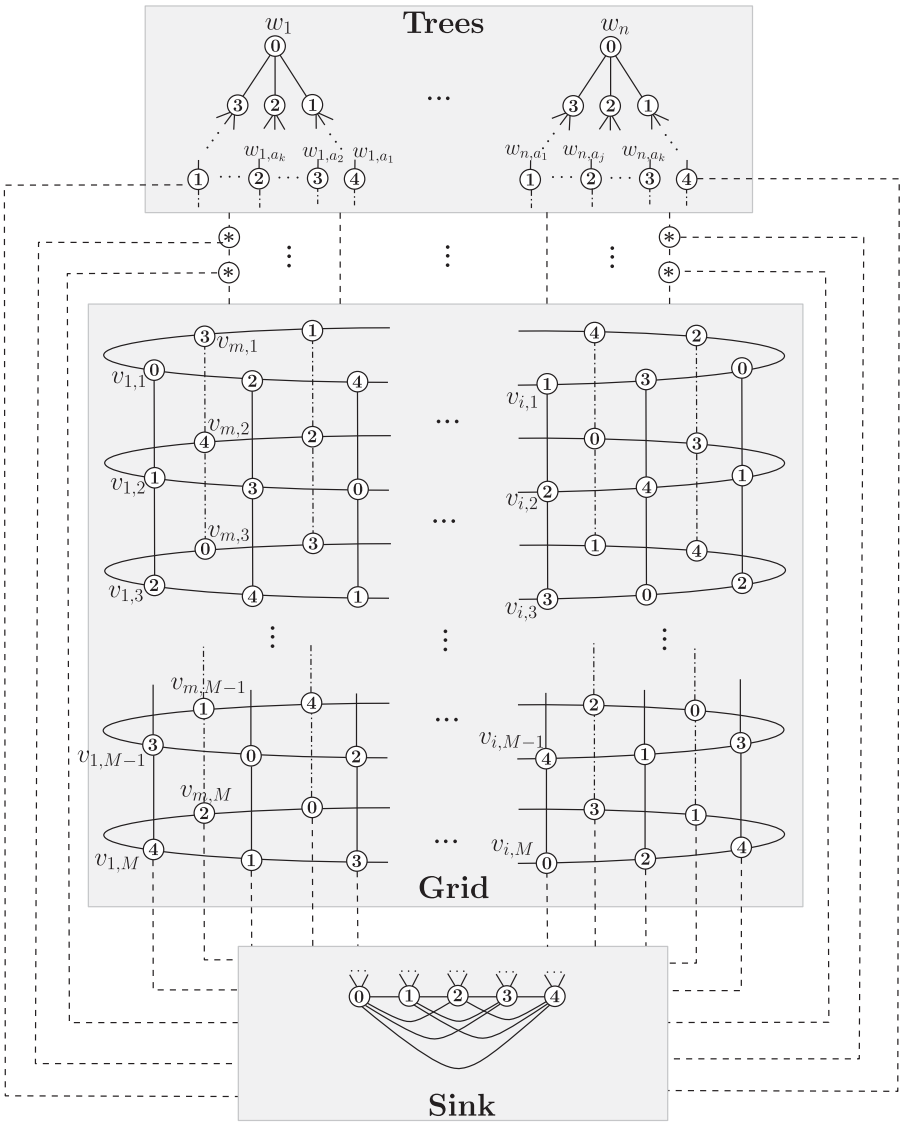


Fig. 3. An illustration of the construction of Γ. Labels assigned to nodes are represented inside the corresponding circles. $(*)$ means that appropriate labels (as defined in the construction) are assigned to such nodes.

in the anchored (5,1)-core, and as Γ' is a subgraph of Γ, the anchored (5,1)-core of Γ has at least the prescribed size. This is true because the core of a graph is a superset of the core of any of its subgraphs.

Claim 5.7: If the set cover instance is a *NO* instance then the size of the (5,1)-core of Γ with b anchors is at most $7m^2$.

Proof. In this case, there are at least $n - b$ trees that do not have any anchor node, implying that there is at least one node in each such tree that is not in the anchored (5,1)-core. Due to Observation 1, all of the nodes in the corresponding trees are also not included in the anchored (5,1)-core. Since the set cover instance is a NO instance, the set $\mathcal{U}$ can't be covered with b sets, and we have at most b trees in the anchored (5,1)-core. Also at least one node in the first row of the grid in Γ is not in the anchored (5,1)-core as $b < m$ without loss of generality. Using an earlier fact that if there exists a node in the grid that is not in the core, then none of the nodes in the grid is in the core, we conclude that the anchored (5,1)-core of Γ does not include any node in the grid except any anchored nodes. Since there are at most $6m$ nodes in each tree, we can't have more than $6m^2$ nodes in the trees that are in the anchored (5,1)-core. Furthermore, with a budget of b anchors, we can ensure at most b^2 nodes from the grid to be in the anchored (5,1)-core. Hence, the size of the anchored (5,1)-core is bounded[1] from above by $7m^2$. ■

We can now set M appropriately to prove the Theorem 5.5. For examples if we set $M = m^{3/\epsilon \log_m 7c}$ then we can't distinguish between an instance with the core size at least N and an instance with the core size at most $\frac{N^\epsilon}{c}$ where N is the number of nodes in Γ. Moreover size of the graph Γ is polynomial in the size of the set cover instance. Therefore, this reduction rules out the existence of a polynomial time algorithm that can approximate the anchored (5,1)-core problem to within a factor of $c.N^{1-\epsilon}$ unless $P = NP$.

[1] Although this bound is good enough for our purpose here, with a more careful analysis, readers may obtain a tighter bound of cm^2 with $c < 1$.

Now for $j \geq 4$, we can always get an instance of anchored $(s+j, s)$-core problem from an instance of the anchored (5,1)-core as follows:

- From the labeled graph Γ (used in the Proof of Theorem 5.5), we obtain a new labeled graph Γ' by keeping the nodes in trees and grid exactly the same. If $\ell(x)_\Gamma$ and $\ell(x)_{\Gamma'}$ are the labels assigned to x in Γ and Γ' respectively, then $\ell(x)_{\Gamma'} = \ell(x)_\Gamma \cup \{5, 6, \ldots, s+3\}$, for every node x in the grid and trees.
- In Γ', we replace the sink with a clique of $(s+j)$ nodes, denoted by K_{s+j}. Each node o_i in the clique is assigned a label set $\ell(o_i)_{\Gamma'} = \{i-1\} \cup Q_i$, where Q_i is a subset of $s-1$ arbitrarily picked labels from the set $\mathcal{R} \setminus (\{i-1\} \cup \{0,1,2,3,4\})$.
- In Γ', for every node x in trees and the gird, we add an edge between x and clique node o_u, $\forall u \in \{s+5, s+6, \ldots, s+j\}$. Note that as a result of these edges, each x gets a set of labels $\{s+4, s+5, \ldots, s+j-1\}$ in its neighborhood.
- Note that as a result of this construction, a (tree or grid) node in Γ' has all $(s+j)$ labels in its closed neighborhood if and only if the corresponding node in Γ has all 5 labels, that is $\{0,1,2,3,4\}$, in its closed neighborhood.

It is easy to see that Claims 5.6 and 5.7 hold for our construction Γ' as well. We conclude that an $O(N^{1-\epsilon})$ approximation algorithm for anchored $(s+j, s)$-core problem for $j \geq 4$ would imply a polynomial time algorithm for Set Cover problem. This is a contradiction unless $P = NP$.

VI. Heuristics for the (r, s)-Core With Anchors Problem

Since the problem of finding a given number of anchors maximizing the (r, s)-core is computationally hard, in fact, is inapproximable in most cases, we present two heuristic algorithms: first based on a simple *greedy heuristic*, and second using a local search based strategy using *simulated annealing*.

First, we note that for a given labeled $\mathcal{G}(\mathcal{V}, \mathcal{E})$, r, s, and anchor nodes $\mathcal{A}$, the anchored (r, s)-core is unique, and we can compute it by iteratively removing non-anchor nodes from $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that do not satisfy the condition (2). We repeat this until we are left with the subgraph $\tilde{\mathcal{G}}_\mathcal{A}(\tilde{\mathcal{V}}_\mathcal{A}, \tilde{\mathcal{E}}_\mathcal{A})$, which is the (r, s)-core with anchors $\mathcal{A}$. We denote this simple scheme by:

$$\tilde{\mathcal{G}}_\mathcal{A}(\tilde{\mathcal{V}}_\mathcal{A}, \tilde{\mathcal{E}}_\mathcal{A}) \leftarrow \texttt{rs\_core\_A}(\mathcal{G}, \ell, \mathcal{A}, r, s). \tag{5}$$

A. Greedy Heuristic

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a labeled graph with $\mathcal{R}$ being the set of labels, s be the number of labels assigned to each node, and b be the number of anchors that need to be selected to maximize the anchored (r, s)-core. In a greedy approach, as outlined in Algorithm 1, we begin with an empty set of anchors $\mathcal{A}$, compute (r, s)-core with no anchor node, and then iteratively add nodes to $\mathcal{A}$ one-by-one. In each iteration, we include a node $v' \in (\mathcal{V} \setminus \tilde{\mathcal{V}}_\mathcal{A})$ in $\mathcal{A}$ that maximizes the size of the resulting (r, s)-core with $\mathcal{A}$. Here, $\tilde{\mathcal{V}}_\mathcal{A}$ is the set of nodes in the anchored (r, s)-core with $\mathcal{A}$. We repeat this step until $|\mathcal{A}| = b$.

Algorithm 1: Greedy Selection of Anchors.

1: **Given:** $\mathcal{G}(\mathcal{V}, \mathcal{E}), \ell, \mathcal{R}, s, b$
2: **Initialization:** $\mathcal{A} = \emptyset$.
3: $\tilde{\mathcal{G}}_\mathcal{A}(\tilde{\mathcal{V}}_\mathcal{A}, \tilde{\mathcal{E}}_\mathcal{A}) \leftarrow \texttt{rs\_core\_A}(\mathcal{G}, \ell, \mathcal{A}, r, s)$
4: **while** $|\mathcal{A}| \leq b$ **do**
5: **for** all $v \in (\mathcal{V} \setminus \tilde{\mathcal{V}}_\mathcal{A})$ **do**
6: $\tilde{\mathcal{G}}_\mathcal{A}(\tilde{\mathcal{V}}_\mathcal{A}, \tilde{\mathcal{E}}_\mathcal{A}) \leftarrow \texttt{rs\_core\_A}(\mathcal{G}, \ell, \mathcal{A} \cup \{v\}, r, s)$
7: $f(v) \leftarrow |\tilde{\mathcal{V}}_\mathcal{A}|$
8: **end for**
9: $v' \leftarrow \arg\max_{v \in (\mathcal{V} \setminus \tilde{\mathcal{V}}_\mathcal{A})} f(v)$
10: $\mathcal{A} \leftarrow \mathcal{A} \cup \{v'\}$
11: $\tilde{\mathcal{G}}_\mathcal{A}(\tilde{\mathcal{V}}_\mathcal{A}, \tilde{\mathcal{E}}_\mathcal{A}) \leftarrow \texttt{rs\_core\_A}(\mathcal{G}, \ell, \mathcal{A}, r, s)$
12: **end while**
13: **Return:** $\mathcal{A}$

Note that if a node is in the (r, s)-core with anchors $\mathcal{A}$, then it is also in the (r, s)-core with anchors $\mathcal{A}' \supset \mathcal{A}$. Thus, when we add v to $\mathcal{A}$ in each iteration, for an efficient implementation, we only need to check the nodes in $\mathcal{V} \setminus \big(\tilde{\mathcal{V}}_\mathcal{A} \cup \{v\}\big)$, instead of $\mathcal{V}$, for the inclusion in the (r, s)-core with anchors. The greedy algorithm takes $\mathcal{O}(b\rho|\mathcal{V}|)$ time, where b is the number of anchors and ρ is the time complexity of computing (r, s)-core for a fixed set of anchors. For our implementation ρ is on the order of $|\mathcal{E}|^2$, but this cost can be reduced to $O(|\mathcal{E}|\log r|\mathcal{V}|)$ by a clever use of appropriate data structure. However, we observe that this does not lead to any significant improvement in running time on practical instances. A numerical evaluation of the algorithm for various networks is given in Section VIII-A.

B. Metaheuristic Search Algorithm

Next, we present an algorithm based on a metaheuristic approach, in particular using *simulated annealing*, to select a given number of anchors to maximize the size of the anchored (r, s)-core. We first compute the (r, s)-core with no trusted node, and then randomly select a subset of b nodes that are not in the core as anchors (line 5). In each iteration, we *perturb* our solution $\mathcal{A}$, that is compute a new set of anchors $\mathcal{A}'$ by randomly selecting a node from the current solution $\mathcal{A}$, and replacing it with a randomly selected node in $\mathcal{V} \setminus \tilde{\mathcal{V}}_\mathcal{A}$ (line 9). If the size of the (r, s)-core with $\mathcal{A}'$ is improved as compared to the anchors in $\mathcal{A}$, then $\mathcal{A}'$ becomes our current solution. In case, the perturbed solution $\mathcal{A}'$ is inferior to $\mathcal{A}$, we replace $\mathcal{A}$ with $\mathcal{A}'$ only with a small probability that is a function of the difference between the sizes of (r, s)-cores with $\mathcal{A}$ and $\mathcal{A}'$, as well as a *temperature* parameter that decreases exponentially with the number of iterations. Note that these occasional replacements, in which anchors resulting in a smaller sized cores become current solutions, prevent the search from getting stuck at local minima. The scheme is outlined in Algorithm 2, and takes $\mathcal{O}(k\rho)$ time, where k is the number of iterations and ρ is the time complexity of computing (r, s)-core for a fixed set of anchors. Unlike the greedy algorithm, the running time here does not depend on the anchors' budget b.

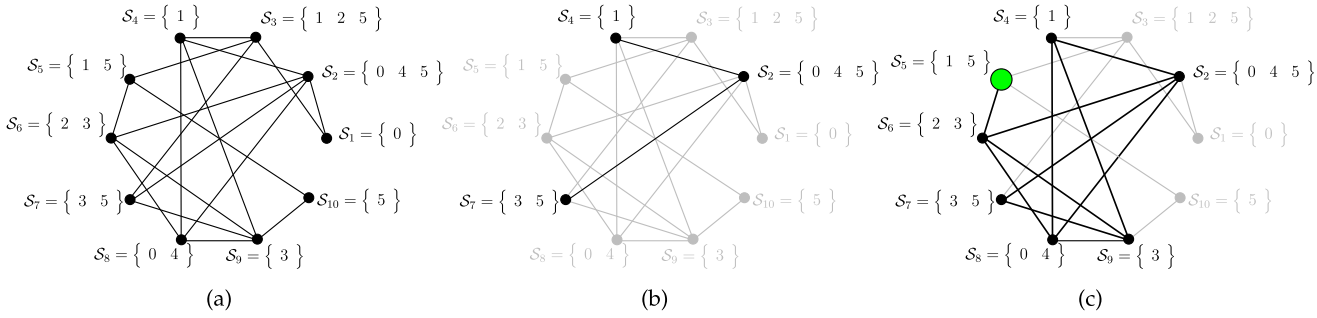


Fig. 4. An example of generalized $(\mathcal{R},\mathcal{S})$-core. (a) For each node u, $\mathcal{S}_u$ is shown. The corresponding $\mathcal{R}_u$'s are: $\mathcal{R}_1=\{1,3,5\}$, $\mathcal{R}_2=\{1,4,5,6\}$, $\mathcal{R}_3=\{1,2,3,4\}$, $\mathcal{R}_4=\{1,4,5\}$, $\mathcal{R}_5=\{1,2,3,4,5,6\}$, $\mathcal{R}_6=\{1,2,3,4,6\}$, $\mathcal{R}_7=\{3,5,6\}$, $\mathcal{R}_8=\{1,2,4,5,6\}$, $\mathcal{R}_9=\{1,2,3,4,5\}$, and $\mathcal{R}_{10}=\{1,2,4,5\}$. (b) The generalized $(\mathcal{R},\mathcal{S})$-core consists of only three nodes. (c) The generalized $(\mathcal{R},\mathcal{S})$-core with node 5 as anchor consists of seven nodes.

Algorithm 2: Simulated Annealing for Anchor Selection.

Input: $\mathcal{G}(\mathcal{V},\mathcal{E}),\ell,r,s,b,$ iterations.
Output: $\mathcal{A}$, $\tilde{\mathcal{G}}_{\mathcal{A}}(\tilde{\mathcal{V}}_{\mathcal{A}},\tilde{\mathcal{E}}_{\mathcal{A}})$
Initialize: $c\leftarrow 1, T_0, \beta$
$\tilde{\mathcal{G}}_{\mathcal{A}}(\tilde{\mathcal{V}}_{\mathcal{A}},\tilde{\mathcal{E}}_{\mathcal{A}}) \leftarrow$ `rs_core_A`$(\mathcal{G},\ell,\emptyset,r,s)$
$\mathcal{A}\leftarrow$ `Random_Selection`$(\mathcal{V}\setminus\tilde{\mathcal{V}}_{\mathcal{A}},b)$
$\tilde{\mathcal{G}}_{\mathcal{A}}(\tilde{\mathcal{V}}_{\mathcal{A}},\tilde{\mathcal{E}}_{\mathcal{A}}) \leftarrow$ `rs_core_A`$(\mathcal{G},\ell,\mathcal{A},r,s)$
$\mathcal{P}\leftarrow|\tilde{\mathcal{V}}_{\mathcal{A}}|$
while $c\leq$ iterations **do**
  $\mathcal{A}'\leftarrow$ `Perturb`$(\mathcal{A},\mathcal{V}\setminus\tilde{\mathcal{V}}_{\mathcal{A}})$
  $\tilde{\mathcal{G}}_{\mathcal{A}'}(\tilde{\mathcal{V}}_{\mathcal{A}'},\tilde{\mathcal{E}}'_{\mathcal{A}}) \leftarrow$ `rs_core_A`$(\mathcal{G},\ell,\mathcal{A}',r,s)$
  $\mathcal{P}'\leftarrow|\tilde{\mathcal{V}}_{\mathcal{A}'}|$
  $p\leftarrow e^{-(\mathcal{P}-\mathcal{P}')/T}$
  if $(\mathcal{P}'>\mathcal{P})\ \vee\ (\texttt{rand}(0,1)\leq p)$ **then**
    $\mathcal{A}\leftarrow\mathcal{A}', \mathcal{P}\leftarrow\mathcal{P}'$
    $\tilde{\mathcal{V}}_{\mathcal{A}}\leftarrow\tilde{\mathcal{V}}_{\mathcal{A}'}$
  end if
  $T\leftarrow T_0\cdot e^{-\beta c}$
  $c\leftarrow c+1$
end while
return: $\mathcal{A}$, $\tilde{\mathcal{G}}_{\mathcal{A}}(\tilde{\mathcal{V}}_{\mathcal{A}},\tilde{\mathcal{E}}_{\mathcal{A}})$

VII. Generalized $(\mathcal{R},\mathcal{S})$-Core

So far, in the context of (r,s)-core, we have assumed that each node is assigned a subset of s labels and would remain to be a part of the network as long as it has an access to all r labels in its closed neighborhood. We can easily generalize this framework by allowing nodes to have different number of labels, that is resources/attributes. Thus, instead of having the same number of labels s by two nodes u and v, we can have a different number of labels for nodes u and v. Similarly, it is also possible that a node u does not need to see all r labels in its closed neighborhood to continue participating in the network. In fact, u participates in the network as long as a subset of labels $\mathcal{R}_u\subseteq\mathcal{R}$ is available in its closed neighborhood. Thus, each node has its own specific condition, in terms of the subset of labels that need to be available in its closed neighborhood, to participate in the network. We can model this by the notion of *generalized* $(\mathcal{R},\mathcal{S})$*-core*.

Generalized $(\mathcal{R},\mathcal{S})$***-core*** Let $\mathcal{G}(\mathcal{V},\mathcal{E}$ be a graph and $\mathcal{R}$ be a set of r labels. Each node $u\in\mathcal{V}$ is assigned a subset of labels $S_u\subseteq\mathcal{R}$, and u needs to have $\mathcal{R}_u\subseteq\mathcal{R}$ labels in its closed neighborhood. Then, the generalized $(\mathcal{R},\mathcal{S})$-core, denoted by $\bar{\mathcal{G}}(\bar{\mathcal{V}},\bar{\mathcal{E}})$ is the maximal subgraph in which every node satisfies

$$\bigcup\nolimits_{v\in(\mathcal{N}[u]\cap\bar{\mathcal{V}})}\mathcal{S}_v=\mathcal{R}_u,\quad\forall u\in\bar{\mathcal{V}}. \tag{6}$$

Here, $\boldsymbol{\mathcal{R}}=\{\mathcal{R}_u\}$ and $\boldsymbol{\mathcal{S}}=\{\mathcal{S}_u\}$. An example of generalized $(\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{S}})$-core is shown in Fig. 4.

As with the (r,s)-core, anchor nodes can significantly improve the size of generalized $(\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{S}})$-core (as illustrated in Fig. 4c). We note that the greedy and simulated annealing based heuristics in Section VI are directly applicable in selecting anchor nodes to maximize the size of generalized $(\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{S}})$-core, as we demonstrate in Section VIII-D.

VIII. Numerical Evaluation

In this section, we evaluate our results on various types of networks including *Erdös-Rényi (ER)* networks, *Barabási-Albert (BA)* networks, and a real-world social network of *Facebook (FB)* users [46], [47].

A. Networks and Related Parameters

ER networks are generated by creating an edge between any two nodes with a specified probability p. BA networks are generated using a preferential attachment mechanism, in which nodes are added to an existing network one-by-one. Each new node is connected to m existing nodes which are chosen with probabilities proportional to the degrees of nodes (i.e., number of their neighbors). The details of networks used are given below.[2]

- [ER–1000] An ER graph with 1000 nodes and $p=0.006$. Average degree of a node is 6.
- [ER–3000] An ER graph with 3000 nodes, $p=0.0027$, and average node degree of 8.
- [BA–1000] A Barabási-Albert graph with with 1000 nodes and $m=3$. The average node degree is 6.

[2] The adjacency matrices of all the graphs and the assignment of labels used in various instances are available in [48].

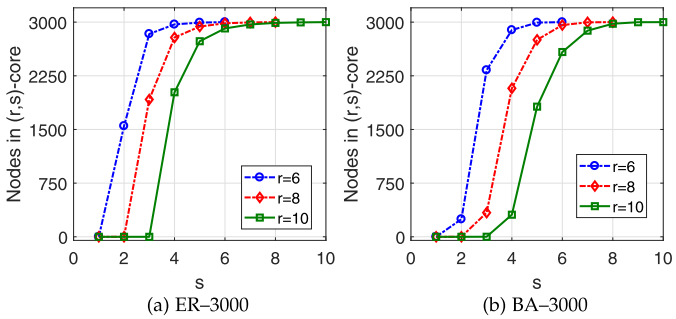


Fig. 5. Size of the anchored (r, s)-core as a function of s for various values of r.

– [BA–3000] A Barabási-Albert graph with with 3000 nodes and $m = 3$. The average node degree is 6.
– [FB–4039] A real-world social network of 4,039 Facebook users, 88,234 edges, and average node degree of 43. More details about the network are in [46], [47].

B. (r, s)-Cores With and Without Anchors

First, for a given r, we illustrate the size of (r, s)-core (with no anchor nodes) as a function of s in Fig. 5. As expected, the size of (r, s)-core increases as the number of labels assigned to each node increases. To compute the (r, s)-core, we randomly assign s labels (using a uniform distribution) from a set of r-labels to each node. Every point in the plots in Fig. 5 is an average of 25 such randomly generated instances.

Second, we illustrate the significance of anchor nodes in improving the size of the (r, s)-core. We also compare the selection of anchor nodes computed by solving ILP, greedy approach as in Algorithm 1, and simulated annealing as in Algorithm 2. For each of the above network, we plot the size of the anchored (r, s)-core as a function of the number of anchors. In particular, we select anchors in the network ER–1000 to maximize the anchored (5,2)-core and the anchored (6,2)-core (Fig. 6a–6b); and in ER-3000 to maximize the anchored (6,2)-core and the anchored (7,2)-core (Fig. 6c–6d). Similarly, in the case of BA–1000 network, we compute and compare the anchors' selection in terms of the sizes of the anchored (5,2)-core and the anchored (6,2)-core (Fig. 7a–7b); and in the case of BA–3000, the anchored (7,3)-core and the anchored (8,3)-core (Fig. 7c–7d).

From the plots in Figs. 6 and 7, we observe that the sizes of the (r, s)-cores are significantly increased by having a small number of anchor nodes, as compared to the baseline cases, in which there are no anchor nodes. For instance, in the case of ER-3000 network, with no anchor nodes the (6,2)-core and (7,2)-core contains 47 and 0 percent of the overall nodes respectively. However, by selecting only 6 percent of the nodes as anchors, the anchored (6,2)- and the anchored (7,2)-cores contain about 80 and 47 percent of the overall nodes respectively, which is indeed a significant improvement. Similarly, in the case of BA-3000 network, the (7,3)-core and the (8,3)-core (with no anchors) contain about 48 and 14 percent of the overall nodes respectively. However, by selecting only 6 percent of the nodes as anchors, the anchored (7,3)- and the anchored (8,3)-cores contain about 67 and 48 percent of the overall nodes, which is again a huge improvement in the sizes of the cores.

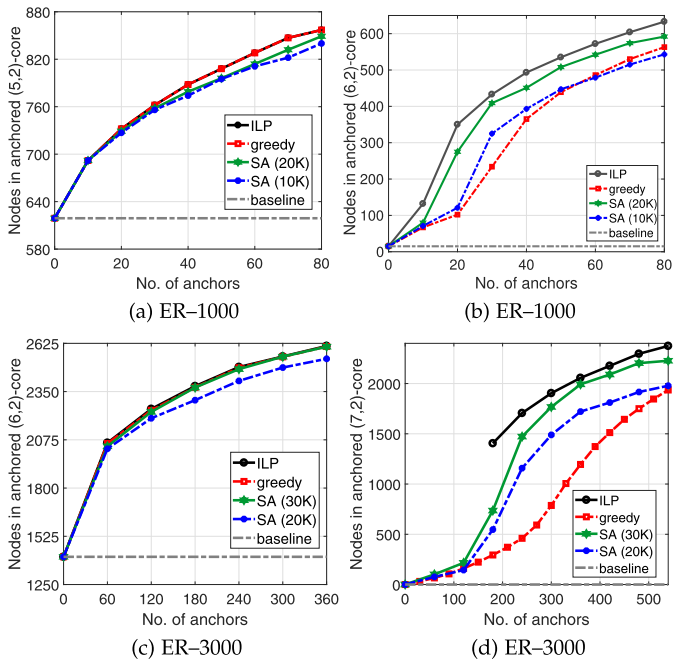


Fig. 6. Nodes in anchored (r, s)-core as a function of the number of anchors selected by solving ILP, greedy, and simulated annealing heuristics. The baseline case indicates the size of (r, s)-core with no anchor nodes. For ER–1000, we plot the sizes of (5,2)- and (6,2)-cores with anchors, whereas, for ER–3000, we plot the sizes of (6,2)- and (7,2)-cores with anchors.

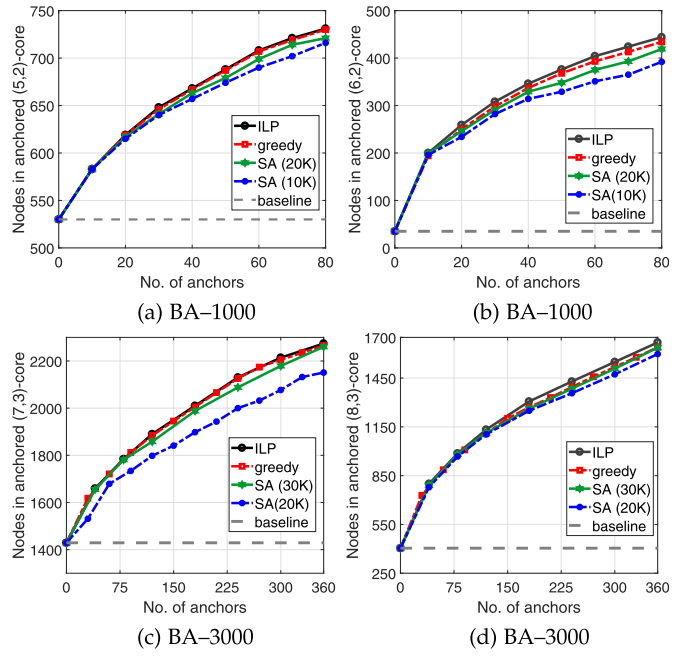
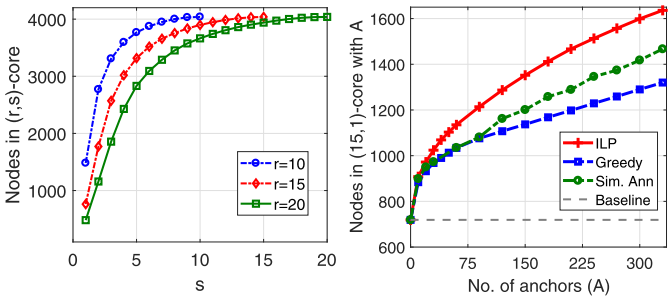


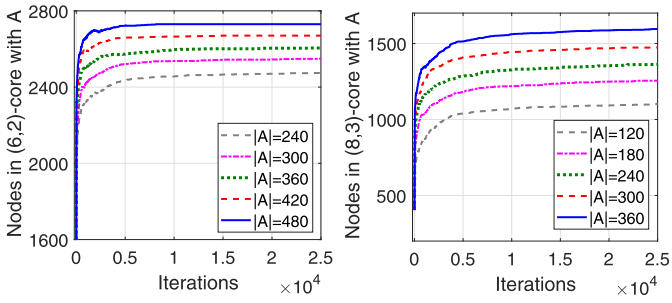
Fig. 7. For BA–1000 and BA–3000 networks, the plots of nodes in (r, s)-core with anchors as a function of the number of anchors selected through an ILP solution, greedy and simulated annealing heuristics. The baseline case indicates the size of (r, s)-core with no anchors.

Moreover, from the plots in Figs. 6 and 7, we observe that simulated annealing gives a solution that is close to ILP solution. In fact, if we perform a large number of iterations, solution by simulated annealing heuristic is very close to the optimal. For our simulations, we use ten, twenty, and thirty



(a) FB–4039 (b) FB–4039

Fig. 8. (a) Size of the (r, s)-core as a function of s for various $r = 10, 15$, and 20. (b) Size of the (15,1)-core with anchors as a function of the number of anchors. The baseline indicates the size of (15,1)-core with no anchors.



(a) ER–3000 (b) BA–3000

Fig. 9. For ER–3000 and BA–3000 networks, sizes of the anchored (r, s)-cores as functions of the number of iterations in simulated annealing for various values of $|\mathcal{A}|$.

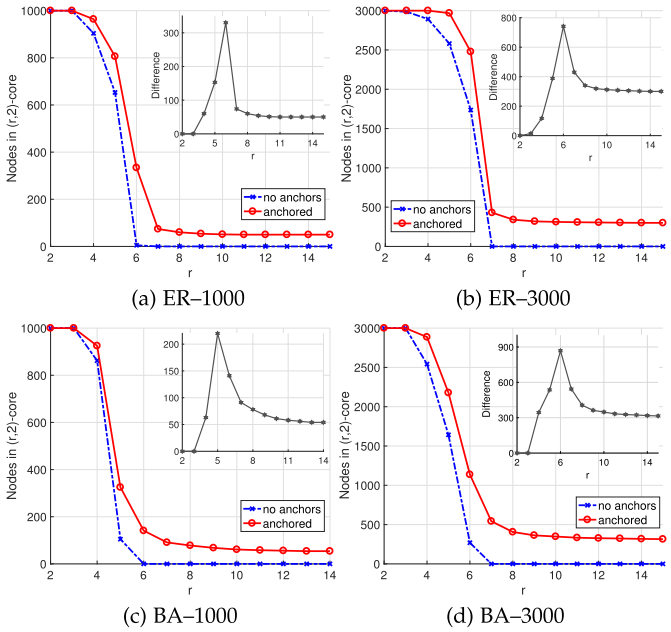
thousand iterations, (as mentioned in the plots), and choose $T_0 = 0.5$ and $\beta = \frac{20}{\text{iterations}}$. On the other hand greedy, also performs well and gives solutions that are not far from optimal solutions. We note that for all the instances, we initially assign s unique labels to nodes randomly (using a uniform distribution) from a set of r labels.

In Fig. 8, we illustrate similar results for the FB-4039 network. Fig. 8a shows the size of (r, s)-core as a function of s for various values of r. In Fig. 8b, we plot the size of (15,1)-core with anchors as a function of number of anchors selected by solving ILP, greedy heuristic and simulated annealing. With no anchors, the (15,1)-core has 719 nodes, that is about 18 percent of the overall nodes. By having only 4 percent of nodes as anchors (using ILP), the size of the (15,1)-core with anchors is 1376 nodes, which is about 34 percent of the overall nodes, and is a significant improvement with such a small number of anchor nodes. Moreover, we observe that simulated annealing performed with 20,000 iterations outperforms the greedy heuristic.

Finally, in Fig. 9, we plot the the sizes of (r, s)-cores with anchors as a function of number of iterations in the simulated annealing heuristic. In particular, for the ER-3000 network, we plot the size of anchored (6,2)-core as a function of iterations in Fig. 9a, and for the BA-3000 network, we plot the size of anchored (8,3)-core as a function of the number of iterations in Fig. 9b. For various values of $|\mathcal{A}|$, we see a rapid increase in the size of cores initially, and then the plots begin to flatten after about 10,000 iterations suggesting the computation of a near-optimal solution.

C. (r, s)-Core With and Without Anchors for Varying r

Fig. 10 illustrates the role of anchors in improving the size of (r, s)-core with varying r values. We fix $s = 2$, and first plot the sizes of (r, s)-cores without anchors as a function of r. Then we select 50 anchors in ER-1000 and BA-1000 networks, and 300 anchors in each of the ER-3000 and BA-3000 networks, and plot the sizes of (r, s)-cores as a function of r. We observe that the difference in the sizes of cores with and without anchors is small whenever r and s are either too close or too far from each other. When r and s are almost the same, it means nodes have almost all the labels by themselves and do not exceedingly depend on neighbors to



(a) ER–1000 (b) ER–3000

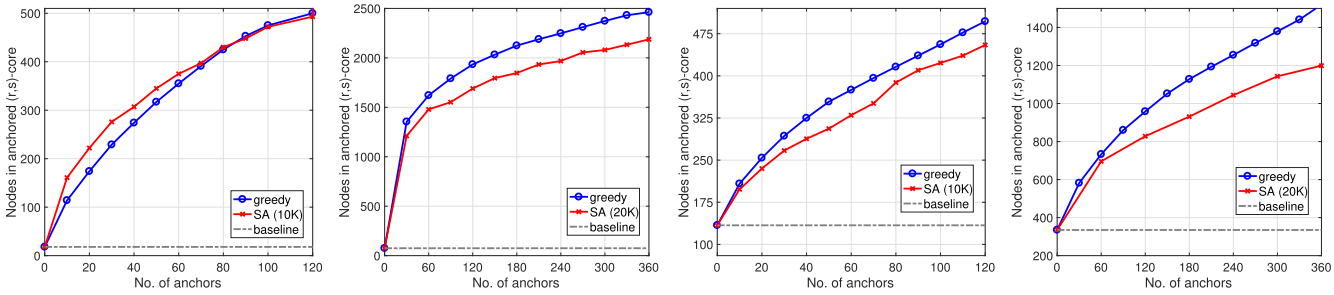
(c) BA–1000 (d) BA–3000

Fig. 10. Nodes in $(r, 2)$-core as a function of r with and without anchors. The subplot within each plot illustrate the difference in the sizes of $(r, 2)$-cores with and without anchors as a function of r.

acquire all the desired labels. On the other hand, if r and s are too far off, it means that nodes depend on neighbors excessively for the desired labels. In such a situation, if there are not a sufficient number of neighbors and labels in the neighborhood of a node, it will drop out unless it is an anchor node. In the worst case, only the anchors continue to participate as we also note this behavior in plots in Fig. 10. Thus, maximum benefit of anchors in improving the size of (r, s)-cores is noted for certain values of r/s depending on the underlying network topology. In our plots, anchors are maximally beneficial for $r/s = 3$ in case of ER-1000, ER-3000, and BA-3000 networks, and $r/s = 2.5$ in case of BA-1000 network.

D. Generalized $(\boldsymbol{r}, \boldsymbol{s})$-Core

In Fig. 11, we illustrate the significance of anchors in improving the sizes of generalized $(\mathcal{R}, \mathcal{S})$-cores. For each of



(a) ER–1000 (b) ER–3000 (c) BA–1000 (d) BA–3000

Fig. 11. Nodes in generalized $(\mathcal{R}, \mathcal{S})$-core with anchors as a function of the number of anchors selected using greedy and simulated annealing. The baseline indicates nodes in $(\mathcal{R}, \mathcal{S})$-core without anchors.

TABLE I
ER-1000

	$\lvert\mathcal{R}_i\rvert = 4$	$\lvert\mathcal{R}_i\rvert = 5$	$\lvert\mathcal{R}_i\rvert = 6$	$\lvert\mathcal{R}_i\rvert = 7$	$\lvert\mathcal{R}_i\rvert = 8$
$\lvert\mathcal{S}_i\rvert = 2$	69	63	73	68	61
$\lvert\mathcal{S}_i\rvert = 3$	62	76	63	63	70
$\lvert\mathcal{S}_i\rvert = 4$	69	61	64	69	69

TABLE II
ER-3000

	$\lvert\mathcal{R}_i\rvert = 4$	$\lvert\mathcal{R}_i\rvert = 5$	$\lvert\mathcal{R}_i\rvert = 6$	$\lvert\mathcal{R}_i\rvert = 7$	$\lvert\mathcal{R}_i\rvert = 8$
$\lvert\mathcal{S}_i\rvert = 2$	193	216	197	199	195
$\lvert\mathcal{S}_i\rvert = 3$	212	200	206	189	193
$\lvert\mathcal{S}_i\rvert = 4$	195	184	197	212	212

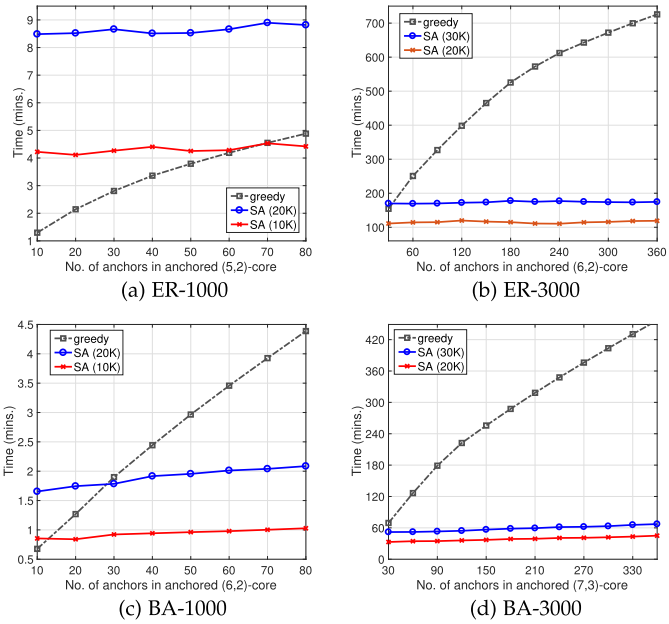
TABLE III
BA-1000

	$\lvert\mathcal{R}_i\rvert = 4$	$\lvert\mathcal{R}_i\rvert = 5$	$\lvert\mathcal{R}_i\rvert = 6$	$\lvert\mathcal{R}_i\rvert = 7$	$\lvert\mathcal{R}_i\rvert = 8$
$\lvert\mathcal{S}_i\rvert = 2$	68	77	63	60	66
$\lvert\mathcal{S}_i\rvert = 3$	63	62	71	78	60
$\lvert\mathcal{S}_i\rvert = 4$	69	61	66	62	74

TABLE IV
BA-3000

	$\lvert\mathcal{R}_i\rvert = 4$	$\lvert\mathcal{R}_i\rvert = 5$	$\lvert\mathcal{R}_i\rvert = 6$	$\lvert\mathcal{R}_i\rvert = 7$	$r = 8$
$\lvert\mathcal{S}_i\rvert = 2$	201	194	192	195	218
$\lvert\mathcal{S}_i\rvert = 3$	194	216	197	189	204
$\lvert\mathcal{S}_i\rvert = 4$	205	190	211	216	178

the ER-1000, ER-3000, BA-1000, and BA-3000 networks, we randomly assign two to four labels to each node from a set of ten labels, that is $2 \leq |\mathcal{S}_i| \leq 4, \forall i$. Moreover, each node requires a set of four to eight labels ($4 \leq |\mathcal{R}_i| \leq 8$), again selected randomly, in its closed neighborhood to continue participating in the network. The exact $\mathcal{S}$ and $\mathcal{R}$ that we use in Fig. 11 are available in [48]. However, a summary of $\mathcal{S}$ and $\mathcal{R}$ used are given in Tables I–IV, in which the $(|\mathcal{S}_i|, |\mathcal{R}_i|)$th entry indicates the number of nodes that are assigned $|\mathcal{S}_i|$ labels and that require $|\mathcal{R}_i|$ labels in their closed neighborhoods. For instance, in Table I, the number of nodes that are assigned $|\mathcal{S}_i| = 2$ labels, and which require $|\mathcal{R}_i| = 8$ labels in their closed neighborhoods for participating in the network is 61. All the plots in Fig. 11 illustrate that anchors (selected by either greedy or simulated annealing) clearly improve the sizes of generalized $(\mathcal{R}, \mathcal{S})$-cores as compared to the baseline cases, in which there are no anchors. Next, we give a numerical evaluation of the running times of heuristics.



(a) ER-1000 (b) ER-3000 (c) BA-1000 (d) BA-3000

Fig. 12. Computation time of greedy and simulated annealing (using ten and twenty thousand iterations) as a function of the number of anchors.

E. Running Timing Plots

In Fig. 12, we plot the running times[3] of greedy and simulated annealing heuristics as a function of the number of anchors in the case of anchored (5,2)-core in ER-1000, anchored (6,2)-core in ER-3000, anchored (6,2)-core in BA-1000, and anchored (7,3)-core in BA-3000 networks. We observe in all the plots that the running time of greedy increases as the anchors increase. However, for a fixed number of iterations, the running time of simulated annealing remains almost constant and does not change with the number of anchors. In fact, the running time increases only if the number of iterations increases. Moreover, as compared to greedy, simulated annealing scales well with an increase in the network

[3] We performed simulations on an Intel Core i7 machine with a 3.6 GHz processor, and 16 GB of RAM.

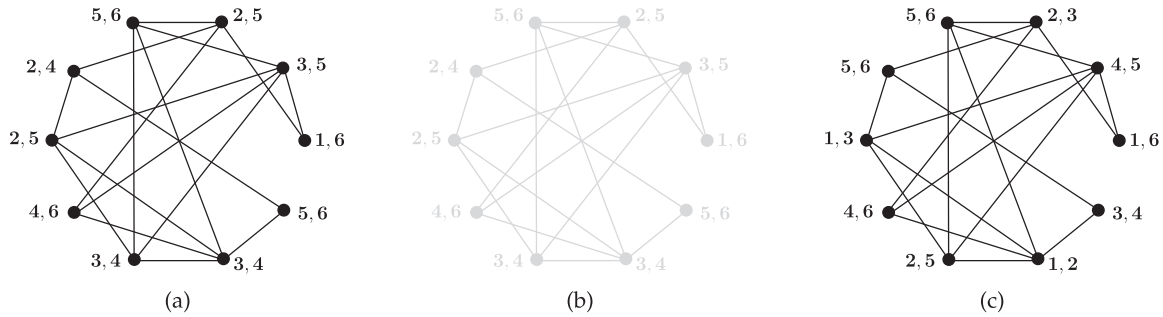


Fig. 13. Increasing the size of (r, s)-core through node relabeling. (a) An initially labeled graph. (b) (6,2)-core with the initial labels. The (6,2)-core is an empty graph in this case. (c) (6,2)-core with the new labels. All nodes are included in the core.

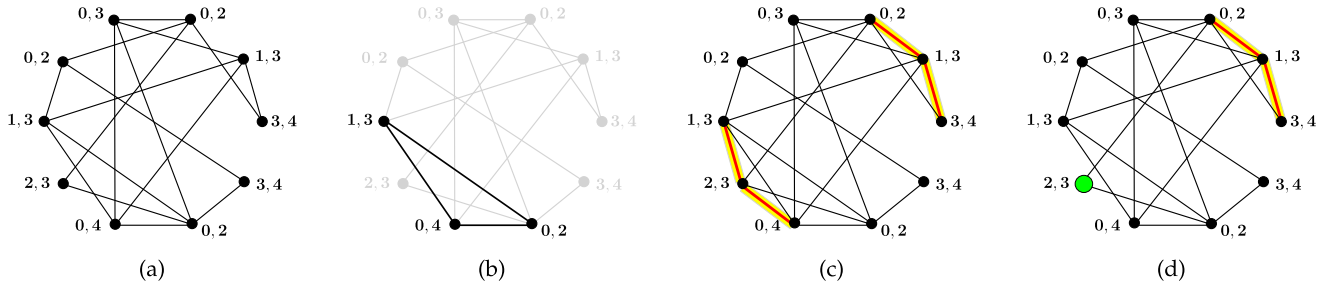


Fig. 14. (a) A network with ten nodes and five labels $\mathcal{R} = \{0, 1, 2, 3, 4\}$. (b) (5,2)-core of the network. (c) The (5,2)-core after adding four extra edges (highlighted). (d) The (5,2)-core after adding two extra edges and a single anchor (highlighted).

size. It not only takes significantly lesser time than greedy, but also produces comparable results in terms of the sizes of the anchored (r, s)-cores.

IX. Further Directions and Conclusions

The notion of (r, s)-core and the anchors based maximization of (r, s)-core can be extended in many different directions. Some of the further directions are described below.

Node Relabeling to Maximize (r, s)*-Core*. The size of the (r, s)-core depends on both the structure of the network and the assignment ℓ of labels to nodes. So far, we have seen the significance of anchor nodes to increase the size of (r, s)-cores, while assuming that nodes' labels are fixed. However, if we are allowed to change the labels of nodes, we get an alternate way – *node relabeling approach* – to further increase the size of (r, s)-core. In practice, re-assignment of labels can be achieved by incentivizing users to change their attributes (e.g., sensing capabilities, resources). An example of this node relabeling approach is presented in Fig. 13. With the initial labels, the (6,2)-core is an empty graph. However, if we change the labels of nodes appropriately, the (6,2)-core is the whole graph. Thus, the *problem is to find an optimal assignment of labels, that is* ℓ*, to maximize the* (r, s)*-core*. For instance, using a result (Theorem 1) in [49], we establish that for any positive integer s, we can always find a labeling of a graph $\mathcal{G}$ such that its $(\lfloor \frac{5s}{2} \rfloor, s)$-core is the graph $\mathcal{G}$ itself under certain conditions.

- the minimum degree of $\mathcal{G}$ is at least two,
- a star graph with six leaf nodes , denoted by $K_{1,6}$ is not an induced subgraph of $\mathcal{G}$, and
- $\mathcal{G} \neq \{$ $\}$.

Along these lines, another interesting direction is to determine the (structural) conditions on the network for a given r and s, which if satisfied would guarantee the existence of labeling ℓ through which the (r, s)-core of the network consists of the whole network. Moreover, in case the node labels are pre-defined, and changing the labels of all (or most) of the nodes is not feasible, we can formulate a *budgeted node relabeling* problem, in which a fixed number of nodes (or labels) can be changed with the objective of maximizing the size of the (r, s)-core.

Connectivity Augmentation to Maximize (r, s)*-Core*: Another way to improve the size of (r, s)-core is by the strategic addition of edges. For instance, one can ask about the minimum number of edges that should be added to the labeled network such that the resulting (r, s)-core consists of the whole network. The problem is related to the *connectivity augmentation* problem (e.g., [50]), in which the goal is to determine the minimum edge set which if added to the existing graph induces the desired connectivity or structural robustness. In Fig. 14, we illustrate the effect of strategically adding more edges in improving the size of (r, s)-core, in which the size of original (5,2)-core is three. However, by adding four extra edges (as highlighted in Fig. 14c), the (5,2)-core consists of the whole network. In fact, we can also employ a *combination* of these approaches simultaneously, for instance, the anchors and and extra edges. We illustrate this synergistic approach in Fig. 14d,

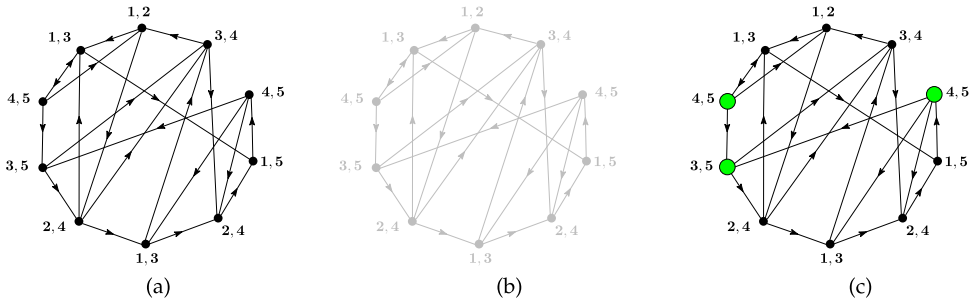


Fig. 15. (a) A directed network with ten nodes. (b) An empty (5,2)-core. (c) Anchored (5,2)-core consisting of all ten node.

in which we have one anchor ad two extra edges that result in the (5,2)-core consisting of the whole network.

Further Generalizations of (r, s)*-Core*: We discussed a generalization of (r, s)-core in Section VII, in which nodes contain different number of resources (labels). Moreover, nodes have their own personalized requirements in terms of labels that need to be available in their neighborhoods for continuous participation. However, we assumed that a node shares its resources uniformly with all of its neighbors. Furthermore, neighbors on which a node depends for resources and neighbors with whom it shares its resources are identical. By considering *directed graphs*, we can distinguish between two types of neighbors. The out-neighbors of a node u in a directed graph can represent the set of nodes with whom u shares its resources, whereas, the in-neighbor can correspond to the individuals that share their resources with u. Now, u continues to participate and share resources with out-neighbors as along as it receives a desired set of resources from its in-neighbors. As in the case of undirected networks, we can introduce anchors to maximize the number of participating individuals.

As an example, consider a directed network of ten nodes in Fig. 15, in which each node is assigned two distinct labels ($s = 2$) from a set of five labels $\mathcal{R} = \{1, 2, 3, 4, 5\}$. A node continues to be a part of the network if it finds all five labels in $\mathcal{R}$ between itself and its in-neighbors, otherwise, it drops out from the network. At the end, the sub-network that remains is the (r, s)-core, where $r = 5$ and $s = 2$. We observe that the (5,2)-core is empty (Fig. 15b). However, if we make three nodes (green circled) as anchors, (5,2)-core consists of all ten nodes, as each non-anchor node finds a complete set of five labels between itself and its in-neighbors.

X. Conclusions

In cooperative networks, reciprocity and mutual benefit are significant in attaining sustainable cooperation among its members. Individuals participate in a network as long as they recognize a sufficient value in such a participation, which depends both on the number and types of peers also participating, which may be different in terms of their attributes, capabilities, resources, information etc. To model such an engagement among heterogeneous nodes within a network, we introduced the notion of (r, s)-core. We considered that each node had a subset of resources that are available within the overall network. A node shared its resources with neighbors as long as it acquired the missing resources from them also. (r, s)-core of the network was the sub-network in which each node had access to all the resources available within the network. We observed that (r, s)-core of the network could be small due to the cascading withdrawal of individuals. To maximize the size of the (r, s)-core, we utilized the idea of anchors – individuals that continued to participate irrespective of the attributes of their neighbors. We showed that by introducing few anchors that are placed strategically within the network, the size of the (r, s)-core can be significantly increased. We analyzed the complexity of the anchors selection problem in detail showing that it is a computationally challenging problem. In fact, we classified the cases in which the problem is polynomial-time solvable, NP-complete, and inapproximable. We also presented heuristics to select anchors, and showed that the anchors selected using the proposed heuristics significantly improved the sizes of the (r, s)-cores with anchors.

Acknowledgments

We thank the anonymous reviewers for their detailed comments and feedback that improved the quality of the paper.

References

[1] M. A. Nowak and K. Sigmund, "Evolution of indirect reciprocity by image scoring," *Nature*, vol. 393, no. 6685, 1998, Art. no. 573.
[2] T. Pfeiffer, C. Rutte, T. Killingback, M. Taborsky, and S. Bonhoeffer , "Evolution of cooperation by generalized reciprocity," *Proc. Roy. Society London B: Biol. Sci.*, vol. 272, no. 1568, pp. 1115–1120, 2005.
[3] D. G. Rand and M. A. Nowak, "Human cooperation," *Trends Cognitive Sci.*, vol. 17, no. 8, pp. 413–425, 2013.
[4] B. Simpson and R. Willer, "Beyond altruism: Sociological foundations of cooperation and prosocial behavior," *Annu. Rev. Sociol.*, vol. 41, pp. 43–63, 2015.
[5] J. Surma, "Social exchange in online social networks. the reciprocity phenomenon on facebook," *Comput. Commun.*, vol. 73, pp. 342–346, 2016.
[6] D. Wang *et al.*, "Using humans as sensors: An estimation-theoretic perspective," in *Proc. 13th Int. Symp. Inf. Process. Sensor Netw.*, 2014, pp. 35–46.
[7] C. C. Aggarwal and T. Abdelzaher, "Social sensing," in *Managing and Mining Sensor Data*. New York, NY, USA: Springer, 2013, pp. 237–297.
[8] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 337–350.
[9] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "PoolView: stream privacy for grassroots participatory sensing," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 281–294.

[10] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A participatory sensing fuel-efficient maps application," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Serv.*, 2010, pp. 151–164.
[11] S. Gu *et al.*, "Data extrapolation in social sensing for disaster response," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst.*, 2014, pp. 119–126.
[12] K. Bhawalkar, J. Kleinberg, K. Lewi, T. Roughgarden, and A. Sharma, "Preventing unraveling in social networks: The anchored k-core problem," *SIAM J. Discrete Math.*, vol. 29, no. 3, pp. 1452–1475, 2015.
[13] W. Abbas, A. Laszka, and X. Koutsoukos, "Graph-theoretic approach for increasing participation in social sensing," in *Proc. 2nd Int. Workshop Social Sensing*, 2017, pp. 43–48.
[14] M. A. Nowak, "Five rules for the evolution of cooperation," *Sci.*, vol. 314, no. 5805, pp. 1560–1563, 2006.
[15] R. L. Trivers, "The evolution of reciprocal altruism," *Quart. Rev. Biol.*, vol. 46, no. 1, pp. 35–57, 1971.
[16] M. Van Veelen, J. García, D. G. Rand, and M. A. Nowak, "Direct reciprocity in structured populations," *Proc. Nat. Acad. Sci.*, vol. 109, no. 25, pp. 9929–9934, 2012.
[17] G. S. van Doorn and M. Taborsky, "The evolution of generalized reciprocity on social interaction networks," *Evolution*, vol. 66, no. 3, pp. 651–664, 2012.
[18] N. E. Friedkin, "Social cohesion," *Annu. Rev. Sociology*, vol. 30, pp. 409–425, 2004.
[19] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: Membership, growth, and evolution," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 44–54.
[20] S. Wu, A. Das Sarma, A. Fabrikant, S. Lattanzi, and A. Tomkins, "Arrival and departure dynamics in social networks," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 233–242.
[21] F. D. Malliaros and M. Vazirgiannis, "To stay or not to stay: Modeling engagement dynamics in social graphs," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 469–478.
[22] N. E. Friedkin, *A Structural Theory of Social Influence*, vol. 13. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[23] A. M. Alaa, K. Ahuja, and M. van der Schaar, "A micro-foundation of social capital in evolving social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 1, pp. 14–31, Jan.–Mar. 2018.
[24] M. Granovetter, "Threshold models of collective behavior," *Amer. J. Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.
[25] J. S. Juul and M. A. Porter, "Synergistic effects in threshold models on networks," *Chaos: An Interdisciplinary J. Nonlinear Sci.*, vol. 28, no. 1, 2018, Art. no. 013115.
[26] S. B. Seidman, "Network structure and minimum degree," *Social Netw.*, vol. 5, no. 3, pp. 269–287, 1983.
[27] G. J. Baxter, S. N. Dorogovtsev, A. V. Goltsev, and J. F. Mendes, "Heterogeneous k-core versus bootstrap percolation on complex networks," *Phys. Rev. E*, vol. 83, no. 5, 2011, Art. no. 051134.
[28] D. Cellai, A. Lawlor, K. A. Dawson, and J. P. Gleeson, "Tricritical point in heterogeneous k-core percolation," *Phys. Rev. Lett.*, vol. 107, no. 17, 2011, Art. no. 175703.
[29] M. Eidsaa and E. Almaas, "S-core network decomposition: A generalization of k-core analysis to weighted networks," *Phys. Rev. E*, vol. 88, no. 6, 2013, Art. no. 062819.
[30] J.-H. Zhao, "Generalized k-core pruning process on directed networks," *J. Statist. Mech.: Theory Exp.*, vol. 2017, no. 6, 2017, Art. no. 063407.
[31] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "*k*-core organization of complex networks," *Phys. Rev. Lett.*, vol. 96, Feb. 2006, Art. no. 040601.
[32] A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes, "*k*-core (bootstrap) percolation on complex networks: Critical phenomena and nonlocal effects," *Phys. Rev. E*, vol. 73, May 2006.
[33] D. Garcia, P. Mavrodiev, and F. Schweitzer, "Social resilience in online communities: The autopsy of friendster," in *Proc. 1st ACM Conf. Online Social Netw.*, 2013, pp. 39–50.
[34] N. K. Panduranga, J. Gao, X. Yuan, H. E. Stanley, and S. Havlin, "Generalized model for k-core percolation and interdependent networks," *Phys. Rev. E*, vol. 96, no. 3, 2017, Art. no. 032317.
[35] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.
[36] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 199–208.
[37] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. 10th IEEE Int. Conf. Data Mining*, 2010, pp. 88–97.
[38] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, 2007, Art. no. 5.
[39] O. AskariSichani and M. Jalili, "Influence maximization of informed agents in social networks," *Appl. Math. Comput.*, vol. 254, pp. 229–239, 2015.
[40] D. Kempe, J. Kleinberg, and É. Tardos, "Influential nodes in a diffusion model for social networks," in *Proc. Int. Colloq. Automata Lang. Program.*, 2005, pp. 1127–1138.
[41] N. Chen, "On the approximability of influence in social networks," *SIAM J. Discrete Math.*, vol. 23, no. 3, pp. 1400–1415, 2009.
[42] E. Mossel and S. Roch, "Submodularity of influence in social networks: From local to global," *SIAM J. Comput.*, vol. 39, no. 6, pp. 2176–2188, 2010.
[43] R. Chitnis, F. V. Fomin, and P. A. Golovach, "Parameterized complexity of the anchored k-core problem for directed graphs," *Inf. Comput.*, vol. 247, pp. 11–22, 2016.
[44] M. Dziubiński and S. Goyal, "Network design and defence," *Games Econ. Behav.*, vol. 79, pp. 30–43, 2013.
[45] W. Abbas, A. Laszka, and X. Koutsoukos, "Improving network connectivity and robustness using trusted nodes with application to resilient consensus," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 2036–2048, Dec. 2017.
[46] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 539–547.
[47] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," Jun. 2014. [Online]. Available: http://snap.stanford.edu/data
[48] 2018. [Online]. Available: http://aronlaszka.com/data/abbas_social.zip
[49] W. Abbas, M. Egerstedt, C.-H. Liu, R. Thomas, and P. Whalen, "Deploying robots with two sensors in $K_{1,6}$-free graphs," *J. Graph Theory*, vol. 82, no. 3, pp. 236–252, 2016.
[50] T. Watanabe and A. Nakamura, "Edge-connectivity augmentation problems," *J. Comput. Syst. Sci.*, vol. 35, no. 1, pp. 96–144, 1987.

Waseem Abbas received the M.Sc. and Ph.D. degrees from Georgia Institute of Technology, Atlanta, GA, USA, both in electrical and computer engineering, in 2010 and 2013, respectively. He is currently an Assistant Professor with the Electrical Engineering Department, Information Technology University, Lahore, Pakistan. Previously, he was a Postdoctoral Research Scholar with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA. His research interests include network control systems, graph-theoretic methods in complex networks, social networks, and resilience of cyber-physical systems. He was a Fulbright scholar from 2009 to 2013.

Aron Laszka received the Ph.D. degree (summa cum laude) in computer science from the Budapest University of Technology and Economics, Budapest, Hungary, in 2014. He is currently an Assistant Professor with the Department of Computer Science, University of Houston, Houston, TX, USA. Previously, he was a Research Assistant Professor with the Department of Electrical Engineering and Computer Science, Vanderbilt University, and a Postdoctoral Scholar with the University of California, Berkeley, between 2015 and 2016. In 2013, he was a Visiting Research Scholar with the Pennsylvania State University. His research interests broadly revolve around the security and resilience of cyber-physical systems and the Internet of Things, the economics of security, and game-theoretic modeling of security problems.

Mudassir Shabbir received the Ph.D. degree from the Division of Computer Science, Rutgers University, New Brunswick, NJ, USA, in 2014. He is currently an Assistant Professor with the Department of Computer Science, Information Technology University, Lahore, Pakistan. Previously, he was with Lahore University of Management Sciences, Pakistan, Los Alamos National Labs, NM, Bloomberg L.P. New York, and Rutgers University. His main area of research is algorithmic and discrete geometry, and has developed new methods for the characterization and computation of succinct representations of large data sets with applications in nonparametric statistical analysis. He also works in Combinatorics and Extremal Graph Theory. He was Rutgers Honors Fellow for 2011–2012.

Xenofon Koutsoukos received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2000. He is currently a Professor with the Department of Electrical Engineering and Computer Science and a Senior Research Scientist with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He was a Member of Research Staff with the Xerox Palo Alto Research Center (PARC) (2000–2002), working in the embedded collaborative computing area. His research work is in the area of cyber-physical systems with emphasis on formal methgods, data-driven methods, distributed algorithms, security and resilience, diagnosis and fault tolerance, and adaptive resource management. He has authored or coauthored more than 250 journal and conference papers and he is co-inventor of four U.S. patents. He was the recipient of the NSF Career Award in 2004, the Excellence in Teaching Award in 2009 from the Vanderbilt University School of Engineering, and the 2011 NASA Aeronautics Research Mission Directorate (ARMD) Associate Administrator (AA) Award in Technology and Innovation. He was named a Fellow of the IEEE for the class of 2018.