Finding low-tension communities

Esther Galbrun* Behzad Golshan† Aristides Gionis‡ Evimaria Terzi§

Abstract

Motivated by applications that arise in online social media and collaboration networks, there has been a lot of work on *community-search*. In this class of problems, the goal is to find a subgraph that satisfies a certain connectivity requirement and contains a given collection of seed nodes.

In this paper, we extend the *community-search* problem by associating each individual with a *profile*. The profile is a numeric score that quantifies the position of an individual with respect to a topic. We adopt a model where each individual starts with a *latent profile* and arrives to a *conformed profile* through a dynamic *conformation process*, which takes into account the individual's social interaction and the tendency to conform with one's social environment. In this framework, *social tension* arises from the differences between the conformed profiles of neighboring individuals as well as from the differences between individuals' conformed and latent profiles.

Given a network of individuals, their latent profiles and this conformation process, we extend the community-search problem by requiring the output subgraphs to have low social tension. From the technical point of view, we study the complexity of this problem and propose algorithms for solving it effectively. Our experimental evaluation in a number of social networks reveals the efficacy and efficiency of our methods.

1 Introduction

A large body of work in social and collaboration networks focuses on solving variants of the *community-search* problem [4, 9, 10, 11], where the high-level goal is to find a subgraph that connects a set of seed nodes and satisfies certain connectivity properties. This problem has applications primarily in online social media and collaboration networks. For example, solutions can be used to identify a set of individuals who are the most appropriate group to organize a social gathering.

Different variants of the problem impose different requirements on the structure of the solution subgraph. For example, in some settings one asks to find high-density communities [10], while in others the objective is to find small-diameter communities [9].

In contrast to the static requirements imposed by existing work, in this paper we incorporate the dynamics of social interactions in the community-search problem. We do so by associating *profiles* to the individuals in the network. The profiles of individuals model their interests, skills, preferences, opinions, and so on, with respect to different aspects or topics. For example, the profile may represent the interest of an individual in discussing about politics, or the working style of an individual — e.g. whether he is a morning person or a night owl. Since profiles may cover a number of different aspects or topics, we assume that they are represented as multi-dimensional vectors.

We further assume that the profiles of individuals change due to the social influence they receive from other individuals in the network. We model this change through what we call *conformation process*, which is a dynamic process. Motivated by existing work on models of opinion formation and social influence [2, 3, 5, 8, 6] we assume that the conformation process is a repeated-averaging process. In our work, we generalize this process from one-dimensional opinions to multi-dimensional profiles. The effect of this process is that the initial profiles of individuals, which we call *latent profiles*, get re-enforced or altered through synthesis and aggregation of different viewpoints of the network participants. Hence, the *conformed profiles* are formed through a process of repeated averaging.

Given this process, the latent profiles of individuals and a social network that represents their social interaction, *social tension* arises because of the differences between the conformed profiles of neighboring nodes and between the conformed and latent profiles of the nodes themselves. In this context, our goal is to identify communities that are not only connected, but also exhibit low social tension. We refer to this problems as the T-COMM problem. To the best of our knowledge we are the first to define and study this problem in the light of profile-conformation processes in a social network.

In terms of technical results, we show that the T-COMM problem is NP-hard and we design graph-theoretic algorithms for solving it. A key difficulty that we overcome while designing these algorithms is that we do not run the conformation process for every step of these algorithms; indeed, this would be computationally very demanding. Rather, we create effective proxies of this process that allow our algorithm to scale. Our experiments with real-world data demonstrate the efficiency and the efficacy of our algorithms.

*Inria Nancy – Grand Est, France. esther.galbrun@inria.fr

†Recruit Institute of Technology, CA, USA. behzad@recruit.ai

‡Aalto University, Finland. aristides.gionis@aalto.fi

§Boston University, MA, USA. evimaria@cs.bu.edu

While we focus here primarily on a community-search problem, we also considered a team-formation variant, where the goal is to find individuals who collectively have the skills required for a task and form a connected subgraph with some required properties.[1]

1.1 Applications. The T-Comm problem has numerous potential applications. For example, when analyzing social networks or social media, it is often useful to identify connected groups of users who have similar profiles with respect to a topic or an idea (i.e. they are in agreement). Note, however, that minimizing the social tension does not necessarily imply looking for highly homogeneous communities but, rather, favoring communities that are able to bridge opinion gaps at low social and communication costs. Groups with low social tension can be recommended as ideal in order to form a group to organize or to invite at a social event. Such problems also arise in human-resource management, when searching for groups of workers who can collaborate together in a conflict-free manner in order to successfully complete a project. Being able to identify a group of people who are not going to experience high social tension is particularly useful when considering cluster hires in universities or start-up companies, where the investment is high and the human factor risk needs to be minimized.

2 Related Work

To the best of our knowledge, we are the first to combine the problem of identifying a group of nodes from an input graph with an underlying dynamic conformation process. However, while the T-Comm problem as we define it is novel, our work is related to existing work on graph mining and opinion dynamics. We summarize the related literature below.

2.1 Community search. Given a graph and a subset of its nodes as a seed, *community-search* problems ask for a set of nodes which is a superset of the seed nodes and induces a connected subgraph in the original graph. Since this class of problems was initially introduced [4, 11], different instances of the community-search problem appeared. Each one imposes a different requirement on the graph-theoretic properties of the reported subgraph, e.g. maximizing the density [10] or minimizing the average pairwise distance [9]. Although these problems are related to the T-Comm problem we study here, the novelty of our problem comes from the fact that we associate nodes with profiles and that we assume a profile-conformation process that takes place over the network. Our objective function is directly related to this process as it aims to minimize the social tension in the reported community. As a result, the technical results we obtain for T-Comm are different from the other community-search problems.

2.2 Opinion dynamics. Starting in the 1970s, models have been built that try to capture the opinion-formation processes in groups and networks [2, 3, 5, 8, 13, 6]. For example, *voter models*, pioneered by Clifford and Sudbury [2] and Holley and Liggett [8] are stochastic models of opinion formation where at each step a node is selected at random and adopts the opinion of one of its neighbors, also selected at random. In DeGroot's *averaging model* [3], each node updates its opinion, by the weighted average of its own opinion and its neighbors' opinion at the previous time step. Friedkin and Johnsen [5, 6] introduced a model where every node has an immutable inner opinion and a changeable expressed opinion. Each node forms its expressed opinion in a repeated-averaging process involving its own inner opinion and the expressed opinions of its neighbors. Given the popularity of this model we also adopt it for modeling our profile-conformation process, and extend it to multiple dimensions.

The Friedkin and Johnsen model has been used in recent works by Bindel *et al.* [1] and by Gionis *et al.* [7]. Bindel *et al.* focus on the price of anarchy in terms of the tension achieved through local repeated averaging and global opinion coordination. Gionis *et al.* aim at identifying the set of nodes whose opinions need to be changed so that the overall positive opinion in the network is maximized. Although our work builds upon Friedkin and Johnsen's ideas to model the profiles and their conformation process, none of the abovementioned works addresses the question of identifying a subgraph with low social tension as we do.

3 Preliminaries

Throughout the paper we consider a social network $G = (V, E)$ where the nodes in V correspond to individuals and the edges in E represent the interactions between these individuals. For simplicity of exposition, we present the case of an unweighted and undirected graph, but the problem and algorithms we discuss can naturally be extended to weighted and directed graphs.

The set of neighbors of node i in G is denoted by $N_G(i)$. Given a subset of vertices $U \subseteq V$, we let $E(U)$ denote the set of edges of G *induced* by U, i.e. $E(U) = \{(i, j) \in E, i, j \in U\}$, and $G(U)$ denote the corresponding *induced subgraph* $G(U) = (U, E(U))$.

[1] An extended version of this paper including the team-formation variant as well as additional experimental results is available at `https://arxiv.org/abs/1701.05352`.

3.1 Profiles. Evidently, each individual has their own set of preferences (e.g. style, habits, biases, and opinions). We refer to this personal set of preferences as a *profile.* For now, let us assume that profiles only reflect preferences regarding a single aspect (e.g. working style), that is, profiles consist of a single attribute. We assume that profile attribute values are represented by a real number in the interval $[0, 1]$ (e.g. a value between 0 and 1 may represent an individual's preference towards a certain software tool).

The key characteristic of our model is that it captures the interaction between the user profiles and their social connections. This is done by assuming that each individual i has a *latent profile* and a *conformed profile*, denoted by x_i and f_i respectively. The latent profile of an individual represents the individual's own true preference. However, individuals may choose *not* to act in accordance with their latent profiles as they try to minimize peer pressure by conforming their preferences to those of their peers. The conformed profile represents these adjusted preferences.

For simplicity, we first describe the model for single-attribute profiles, but later on we discuss how to extend it to multi-attribute profiles. We summarize the latent and conformed profiles of all n individuals with respect to a single attribute using vectors $\mathbf{x}$ and $\mathbf{f}$ respectively.

3.2 Measuring tension. Due to the underlying social structures and mechanisms, the conformed profile f_i of a node can differ from its latent profile x_i. In such case, the node will bear an *inner tension* caused by the difference between its own latent and conformed profiles. On the other hand, the difference between the node's conformed profile and between the conformed profiles of its neighbors will cause *cross tension.* Hence, the total tension on node i is:

$$T_i(G, \mathbf{x}, \mathbf{f}) = (x_i - f_i)^2 + \sum_{j \in N_G(i)} (f_i - f_j)^2.$$

Then, the *social tension* of the network is simply the sum of the individual tensions, defined as

$$\begin{aligned} T(G, \mathbf{x}, \mathbf{f}) &= \sum_{i \in V} T_i(G, \mathbf{x}, \mathbf{f}) \\ &= \sum_{i \in V} \Big((x_i - f_i)^2 + \sum_{j \in N_G(i)} (f_i - f_j)^2 \Big), \qquad (3.1) \end{aligned}$$

which can alternatively be written as the sum of the overall inner and cross tensions

$$T(G, \mathbf{x}, \mathbf{f}) = \sum_{i \in V} (x_i - f_i)^2 + \sum_{(i,j) \in E} 2\,(f_i - f_j)^2.$$

3.3 Conformation process. But how do nodes arrive at their conformed profiles? Consider a repeated averaging process where at each step each node adjusts its conformed profile by setting it to the average of its latent profile and the conformed profile of its neighbors. Formally, denoting as $f_i(t)$ the conformed profile of node i at step t, we have:

$$f_i(t+1) = \frac{x_i + \sum_{j \in N_G(i)} f_j(t)}{1 + |N_G(i)|}. \qquad (3.2)$$

Computing the conformed profiles according to the *repeated averaging model* is equivalent to choosing f_i to minimize $T_i(G, \mathbf{x}, \mathbf{f})$. That is, if each node aims to minimize its tension, the *repeated averaging model* provides an optimal choice for the conformed profiles. In that sense, using the *repeated averaging model* yields a Nash equilibrium for the tension, not a social optimum [1].

A practical consideration is that in this model the latent profiles are assumed to be known, while the conformed profiles are the output of the conformation process. But, in practice, we have access to the conformed profiles while the latent profiles cannot be observed. This, however, does not constitute a problem for our model. One can swap the known and unknown variables of the system and solve for the latent profiles, given the conformed profiles and the original network.

The model adopted here for how conformed profiles emerge is a well-studied opinion-formation and social-influence model that has been introduced by sociologists [3, 5, 6]. In particular, the work of Friedkin *et al.* [6] validates this model by conducting a set of controlled experiments in which they observe how interactions between individuals in small groups influence their expressed opinions. The study demonstrates that the repeated-averaging model can both predict the opinions that individuals converge to, as well as the rate of convergence. Others have studied the mathematical properties of this model. For instance, it has been shown [1, 7] that the process converges in polynomial time to a fixed-point solution. In fact, the final conformed profiles can be computed by a matrix inversion [7], but actually repeating the averaging process leads to much faster computation.

3.4 Multi-attribute profiles. Our assumption so far has been that profiles reflect the preferences of individuals with respect to a single attribute. But our notion of latent and conformed profiles can be easily extended to the case of multiple attributes, leading to multi-attributes — i.e. multi-dimensional — profiles. Assume there are m aspects or topics of interest, each associated to an attribute. The latent and conformed profiles can be simply extended from a single real number to real-valued vectors of dimensionality m, where each entry corresponds to one of the attributes. We summarize the latent and conformed profiles of n individuals on m attributes using $n \times m$ matrices $\mathbf{X}$ and $\mathbf{F}$ respectively. Note that each column of these matrices, denoted by $\mathbf{x}_a$ and $\mathbf{f}_a$ for $a = 1, \dots, m$, corresponds to a single attribute. In the multi-attribute case, the conformed profiles can be computed as before by applying Equation (3.2) in a column-wise fashion. Similarly, the social tension $T(G, \mathbf{X}, \mathbf{F})$ is defined as the sum of the social tensions across all m attributes.

4 The T-Comm problem

In this section, we introduce the T-Comm problem and study its complexity.

At a high level, the T-Comm problem aims to find a connected, low-tension community that involves a chosen subset of members. Formally, this intuitive statement is captured by the following problem definition:

Problem 1. (T-Comm) *Given a network $G = (V, E)$, latent profiles $\mathbf{X}$ and a set of seed nodes $Q \subseteq V$, find $V' \subseteq V$ such that $Q \subseteq V'$, the graph G' induced by V' on G is connected and $T(G', \mathbf{X}, \mathbf{F})$ is minimized, where $\mathbf{F}$ is computed by the repeated averaging model on G'.*

Note that when defining the T-Comm problem, we assume that the social tension is computed as in Equation (3.1), $\mathbf{X}$ is the matrix containing the latent profiles of the nodes in V' and $\mathbf{F}$ contains the conformed profiles of individuals, computed using the repeated averaging model described in the previous section (see Equation (3.2)) over the subgraph induced by V'.

Given these assumptions, we can make the following observations with respect to the requirements imposed on the solution of T-Comm: as the number of edges in the resulting subgraph and the differences in the conformed profiles across these edges decrease, so does the social tension. In particular, the complete absence of edges results in no tension at all. However, the requirement that the output subgraph should be connected forbids such solutions.

From the application point of view, connectivity is important as it guarantees communication among the community members. One can see that minimizing tension and guaranteeing connectivity leads to an interesting trade-off between the density of edges and the homogeneity of the profiles of nodes in the reported subgraph. Communities should consist of individuals who share similar profiles or individuals who have divergent profiles but are needed to guarantee connectivity, these latter ones being preferably very sparsely connected with the rest of the community members.

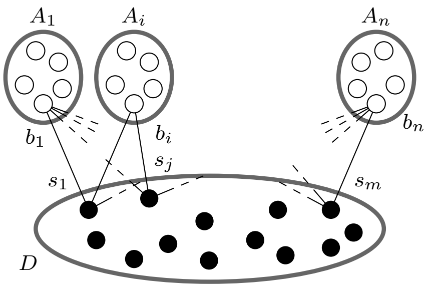

Figure 1: Schema of the constructed graph for the reduction from **X3C**. The black and white nodes have latent profiles with values 1 and 0 respectively.

With respect to the computational complexity of the T-Comm problem, we obtain the following result.

Proposition 4.1. *The T-Comm problem cannot be solved optimally in polynomial time even with a single attribute (i.e. $m = 1$) unless $P = NP$.*

Proof. (Sketch) We prove the hardness of T-Comm with a reduction from the problem of exact cover by 3-sets.

The exact cover by 3-sets problem, **X3C** for short, asks the following question. Given a universe of elements $B = \{b_1, \dots, b_p\}$, where the number of elements p is a multiple of 3, and a collection $S = \{s_1, \dots, s_q\}$ of 3-elements subsets of B, is there a collection $S' \subseteq S$ such that every element in B occurs in exactly one member of S'?

Given an instance of the **X3C** problem, we construct an instance of the T-Comm problem with one-attribute profiles (i.e. m = 1) as follows. To each element b_i in B we associate a node, called an "element-node", with latent profile 0, and to each set s_j in $\mathcal{S}$ we associate a node, called a "set-node", with latent profile 1. Each set-node is connected to the three nodes that represent the elements it contains. In addition, we make all the q set-nodes part of a larger clique D of $o + q$ nodes with latent profiles equal to 1. Also, we make each element-node b_i part of a larger clique A_i containing o nodes with latent profiles 0. Finally, we assume that all the nodes in our construction are seed nodes except for the q set-nodes. This construction is illustrated in Figure 1.

We prove that if the given instance of **X3C** has an exact 3-set cover, then the minimum tension subgraph solution of T-Comm will contain the set-nodes of this exact cover.

The main idea of the proof is based on the following observation. Selecting a large (yet polynomial in p) value for o, increases the size of all cliques in our construction (i.e. clique D and all A_i cliques). As a result, the conformed profile of all the nodes (including the set-nodes) in D will be very close to 1, and the conformed profiles of all nodes in the A_i cliques will be very close to 0. Thus, the only non-negligible source of tension will be the tension across the edges that connect element-nodes to set-nodes. Each such edge would increase the tension by almost a unit.

Note that since each b_i is a seed node, it has to be connected to the o seed nodes in D. This can be achieved only by going through a set-node. Thus, the solution to **X3C** has to pick a subset of set-nodes that cover all the element-nodes to ensure connectivity. Now, if x set-nodes are included in the subgraph then the tension would be roughly $3x$. Obviously, the solution that minimizes the tension is the solution that picks the smallest number of set-nodes. This is achieved by selecting an exact cover, if such cover exists.

5 Algorithms for T-Comm

While the objective of the T-Comm problem is to minimize the social tension in the solution graph, $T(G', \mathbf{X}, \mathbf{F})$, obtaining the conformed profiles through the repeated-averaging process is costly; thus, it is not feasible to compute the social tension on a large number of candidate subgraphs. A possible alternative is to compute the conformed profiles by applying the repeated-averaging process on the whole graph once and use these profiles when evaluating the tension in the candidates later on. However, while designing our algorithms, we observed that this is a poor choice. Intuitively, the presence of a node with a latent profile that departs greatly from its neighbors' might significantly sway their conformed profiles. During the search, this node will likely be removed from the candidate set early on, but its effect would remain.

In order to avoid such effects, but also avoid the repeated computations of social tension we use the following trick in all our algorithms: for a pair of neighboring nodes i and j we assign to edge $(i, j) \in E$ weight $w_{ij} = |x_i - x_j|$. We then use this weight as a way of quantifying the contribution of this edge in the overall social tension. Although w_{ij} is just a proxy of the edge's contribution, the trick appears to perform well in practice and leads to significant speedups. Once our algorithms obtain the set of nodes to report, we apply the repeated-averaging process on the induced subgraph in order to evaluate the social tension of the solution.

We propose two approaches for finding good candidate solutions for this problem.

5.1 Spanning-tree approach. This approach connects the seed nodes by building a spanning tree between them and is based on the 2-approximation algorithm for the Steiner tree problem [12]. It works as follows: first, it computes the shortest path between every pair of seed nodes. Next, it constructs a complete graph over the seed nodes such that the weight of the edge between two nodes corresponds to their shortest path distance in the original graph. Then, it considers the minimum-spanning tree from this complete graph, e.g. obtained with Prim's algorithm, and replaces each edge of the spanning tree with the associated original shortest path. The resulting subgraph constitutes the output of our tree-based algorithm. A sketch of this algorithm, called `CTree`, is shown in Figure 2. Note that this approach, searching for the best spanning tree, lacks any control over the induced edges that will be included in the solution. In this sense, it is an optimistic strategy.

```
Input: Network $G = (V, E)$, latent profiles $\mathbf{X}$, seed nodes $Q \subseteq V$, path length function len()
Output: Community nodes $V'$
$H \leftarrow$ complete weighted graph over nodeset $Q$, such that
        the weight of edge $(i, j)$ is $\lambda(i, j) = \min_{p \in \mathcal{P}_{ij}} \text{len}(p)$
            with $\mathcal{P}_{ij}$ the set of paths in $G$ between $i$ and $j$
$K \leftarrow$ minimum spanning tree of $H$
$V' \leftarrow$ expand $K$ by replacing edges by
                their corresponding shortest path in $G$
return $V'$
```

Figure 2: The `CTree` algorithm for solving T-Comm.

We obtain different variants of this algorithm depending on the measure used to evaluate the length of a path. Having experimented with various options, we focus on two variants, where the length of the path is either

(i) the number of edges involved, or

(ii) the sum of weights of the edges along the path.

In other words, if $p_{ij} = (i, v_1, v_2, \ldots, v_k, j)$ is a path between i and j, we have $\text{len}(p_{ij}) = k + 1$ in the first variant, and $\text{len}(p_{ij}) = w_{iv_1} + w_{v_1v_2} + \cdots + w_{v_kj}$ in the other. We denote these variants as `CTree(e)` and `CTree(s)` respectively.

The main step in our algorithm is the computation of the shortest path between all pairs of seed nodes by running the Dijkstra algorithm from each seed node in turn. The running time of our algorithm is thus $\mathcal{O}(|Q|\,(|V| + |E| \log |E|))$.

5.2 Top-down approach. The second algorithm is a top-down approach which starts with the full graph and iteratively removes nodes until it is no longer possible to continue without disconnecting the seed nodes. The pseudo-code for this algorithm, which we call `CPeel`, is given in Figure 3.

Again, we obtain different variants, this time by

```
   Input: Network $G = (V, E)$, latent profiles $\mathbf{X}$, seed
   nodes $Q \subseteq V$, node scoring function score()
   Output: Community nodes $V'$
1: $V' \leftarrow \emptyset$; $K \leftarrow V$
2: while $K \neq \emptyset$ do
3:     $v \leftarrow \arg\max_{i \in K} \text{score}(i, V' \cup K)$
4:     $K \leftarrow K \setminus \{v\}$
5:     if $Q$ is not disconnected in $G(V' \cup K)$ then
6:         $K \leftarrow \{i \in K, N_{G(V' \cup K)}(i) \neq \emptyset\}$
7:     else
8:         $V' \leftarrow V' \cup \{v\}$
9: return $V'$
```

Figure 3: The CPeel algorithm for solving T-Comm.

varying the score() function for choosing the next node to remove. We selected the following three scores:

(i) The score is a number assigned randomly to each node when initializing the algorithm, which determines the order in which the nodes are peeled. This random variant is denoted as `CPeel(r)`.

(ii) The score is the sum of the weights of the remaining edges adjacent to the node, i.e.

$$\text{score}(i, U) = \sum_{j \in N_{G(U)}(i)} w_{ij}.$$

(iii) The score is the largest weight among the remaining edges adjacent to the node, i.e.

$$\text{score}(i, U) = \max_{j \in N_{G(U)}(i)} w_{ij}.$$

The second and third scores are similar but the former uses sum where the latter takes the maximum, resulting in variants `CPeel(s)` and `CPeel(m)`, respectively. Nodes with larger scores are considered first (line 3 in Figure 3). Meanwhile, in all three variants, nodes that get isolated are pruned away (line 6). Rather than favoring good connections, this second approach removes nodes that might generate large costs, and thus follows a pessimistic strategy.

In practice, we can find a minimum connecting tree using the strategy described previously. Then, when we are about to remove a node, we check whether it belongs to the current tree, in which case we need to look for an alternative tree that does not involve this node. Only if such a tree can be found can we safely remove the node. In the worst case, we would have to recompute the spanning tree for each node, resulting in a running time $\mathcal{O}(|V| (|V| + |E| \log |E|))$.

6 Experiments

We now turn to the evaluation of the different variants of our proposed algorithms, `CTree` and `CPeel`.

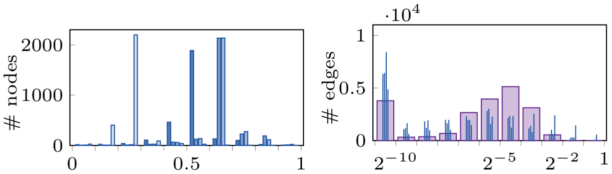

Figure 4: Distribution of latent profile values, x_i, over nodes (top) and of squared weights, w_{ij}^2, over edges (bottom) for the `ICDM` network (eigenv. conferences).

We start by presenting the datasets used in this evaluation. In our setting, each dataset consists of a network together with latent profiles for the nodes. Hence, we first introduce the networks, before explaining our approach for obtaining latent profiles.

6.1 Networks. Of our collections of networks, two consist of subgraphs extracted from the DBLP co-authorship database,[2] where vertices represent researchers, and edges represent co-authorship relations. Specifically, we extracted the ego-nets of radius 2 for some selected high-profile computer scientists. The resulting ego-nets form the collection denoted as `DBLP.E2`. We also consider the subgraphs induced by researchers who have published in the ICDM and KDD conferences, respectively, constituting the `DBLP.C` collection.

Our third collection of networks consists of subgraphs extracted from the Internet Movie Data Base[3] where vertices represent actors and edges connect actors who played together in at least one movie. Specifically, we constructed actor networks from this database by considering some well-known directors and production companies, such as Francis F. Coppola or the Warner Bros. Entertainment Inc., and extracting the network induced by their movies. The resulting networks form the collection denoted as `IMDB`.

In our problem, we are looking for connected subgraphs. If the seed nodes in T-Comm belong to different components, there is obviously no solution. Hence, in our experiments we consider only the largest component of each of the networks.

The statistics of a sample of the networks from these three collections can be found in Table 1.

6.2 Profiles. Besides links, the co-authorship and the co-acting networks contain additional information which we exploit to derive structured profiles, as follows.

In the co-authorship networks, we associate nodes, i.e. researchers, to the conferences in which they published. We then turn this information into profiles by

[2] `http://dblp.uni-trier.de/xml/`

[3] `http://www.imdb.com`

considering the eigenvectors associated to the largest eigenvalues of the incidence matrix of conferences to nodes, scaled to the unit interval. Neighboring researchers in these networks are collaborators who have published papers together. Hence, they published in some of the same conferences, and more generally share similar research interest. Intuitively, they will therefore be assigned similar profile values.

In the co-acting networks, we consider the genres of the movies each actor played in and turn this information into profiles, once again by computing the eigenvectors associated to the largest eigenvalues of the obtained incidence matrices.

The distributions of latent profiles values (x_i) over nodes and of squared weights (w_{ij}^2) over edges in the `ICDM` network are shown in Figure 4.

6.3 Evaluation measures For a solution nodeset V', our main evaluation measure is $T(G(V'), \mathbf{X}, \mathbf{F})$ — $T(V')$ for short — the social tension in the subgraph induced by V' with conformed profiles obtained by applying the repeated averaging process over that subgraph.

Two main properties contribute to a solution subgraph having a low social tension (see Equation (3.1)). On one hand, finding a subgraph with few edges results in fewer terms in the sum. On the other hand, finding a subgraph with low tension edges results in small values in the sum. Thus, we compute two auxiliary values that provide insight into the nature of the solutions obtained. Namely, for a solution V' we compute the number of edges in the solution, $|E(V')|$, and the average of the squared edge weights in the solution, $\overline{w^2}(V') = \frac{1}{|E(V')|} \sum_{(i,j) \in E(V')} w_{ij}^2$.

Solutions obtained for different seed sets are hardly comparable. Thus, to make the evaluation possible, we standardize the measured values before aggregating them. Specifically, we use the number of edges in the minimum spanning tree connecting the seed nodes, e_b, as a comparison basis (and lower bound) for the number of edges in the solution subgraphs, and divide $\overline{w^2}(V')$ by the corresponding average over the whole graph.

Given a solution V', we take the following evaluation measures, for which lower values are more desirable:

(*i*) the *standardized social tension* (main measure)
$$\tau(V') = T(V')/(2e_b \cdot \overline{w^2}(V)),$$

(*ii*) the *standardized solution size* (auxiliary measure)
$$\varepsilon(V') = |E(V')|/e_b, \text{ and}$$

(*iii*) the *standardized average edge weight* (auxiliary measure)
$$\sigma(V') = \overline{w^2}(V')/\overline{w^2}(V).$$

6.4 Generating sets of seed nodes. For each dataset, i.e. each pair of network and latent profiles, we run each algorithm with a number of different sets of seed nodes Q. Here we restrict ourselves to sets of seven and four seed nodesfor the co-authorship networks and the co-acting networks respectively, as representative scenarios for the community-search problem.

As we expect the distance between the nodes in the seed set to have an impact on the behavior of the algorithms, we want to sample seed sets across the range of possible distances and to group them based on this criterion. Thus, we generate a thousand seed sets and look at the maximum pairwise distance within each set. We then select at most 30 seed sets from the 10-33%, 33-66%, and 66-90% percentiles of the distance distribution. The resulting three groups of seed sets are denoted as $D1$, $D2$ and $D3$, from tight seed sets to more dispersed ones. Results in Figure 5 are presented aggregated according to these distance groups.

6.5 Single-attribute and multi-attribute profiles. For each latent profile construction scheme, i.e. whether derived from conferences or movie genres (see Section 6.2), we can either consider the column vectors separately, thereby obtaining several single-attribute profiles, where each node is associated to a single profile value, or consider the entire matrix at once thereby obtaining one multi-attribute profile, where each node is associated to a multi-attribute profile vector.

In our experiments, we take the first four columns of the matrix as four separate single-attribute profiles and the entire matrix as one multi-attribute profile (6 and 21 attributes for the `DBLP` and `IMDB` datasets, respectively). Results for the `C.Papadimitriou` network, for single-attribute (left) and multi-attribute (right) latent profiles, are shown in Figure 5. We observe that the algorithms that exploit the profiles of individuals outperform the other two variants in (almost) all cases.

Recall that neither `CTree(e)` nor `CPeel(r)` consider the profiles of individuals. All they can do is minimize the social tension by finding a small subgraph to connect the seed nodes. Our results show that they are indeed quite effective at minimizing the number of edges in the reported solutions, typically achieving the lowest values of $\varepsilon(V')$ (middle column in each block of Figure 5). However, `CTree(s)`, `CPeel(m)`, and `CPeel(s)`, the profile-aware variants, find solutions with lower edge weights, i.e. achieving lower values for $\sigma(V')$ (right hand side column), at the cost of including extra edges. This gives them an advantage for minimizing social tension, obtaining lower values for $\tau(V')$ (left hand side column).

This pattern clearly holds whether we consider single-attribute or multi-attribute profiles.

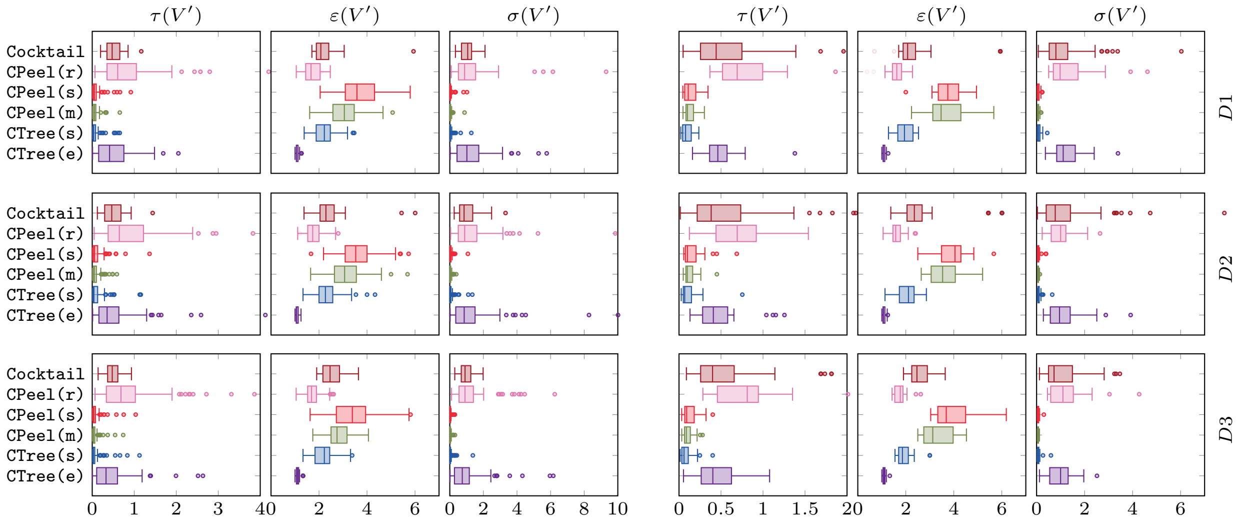

Figure 5: Results for the T-Comm problem on the C.Papadimitriou with single-attribute (left) and ten-attribute (right) latent profiles derived from conferences.

Table 1: Average running times (in seconds) of the algorithms (± std. dev.) solving the T-Comm problem on networks of varying number of vertices ($|V|$), edges ($|E|$) and average degree densities (δ).

Network	$\|V\|$	$\|E\|$	δ	CTree(e)	CTree(s)	CPeel(s)	CPeel(m)	CPeel(r)
IMDB WarnerBros 1970s	225	1 599	7.11	0.0 (±0.0)	0.1 (±0.0)	0.9 (±0.1)	0.7 (±0.1)	0.1 (±0.1)
IMDB F.F.Coppola	678	6 306	9.30	0.0 (±0.0)	0.7 (±0.1)	8.8 (±1.6)	6.4 (±1.2)	1.4 (±0.6)
DBLP.E2 E.Demaine	2 234	7 701	3.45	0.1 (±0.0)	2.8 (±0.3)	75.7 (±12.3)	60.9 (±13.0)	19.2 (±6.2)
DBLP.E2 C.Papadimitriou	2 613	9 472	3.62	0.1 (±0.0)	3.2 (±0.3)	114.6 (±20.0)	91.6 (±22.0)	31.6 (±9.6)
DBLP.C ICDM	2 795	10 280	3.68	0.2 (±0.0)	3.3 (±0.3)	163.9 (±28.1)	133.9 (±29.7)	38.9 (±11.2)
DBLP.C KDD	2 737	11 072	4.05	0.2 (±0.0)	3.5 (±0.2)	166.8 (±27.8)	136.7 (±28.1)	36.0 (±13.0)
DBLP.E2 P.Yu	4 596	13 250	2.88	0.2 (±0.0)	4.4 (±0.3)	291.3 (±56.3)	242.3 (±43.1)	68.6 (±20.5)
IMDB WarnerBros	2 111	32 166	15.24	0.3 (±0.1)	3.0 (±0.2)	139.1 (±19.7)	57.2 (±13.0)	22.8 (±9.1)
IMDB WB+Paramount+Fox	5 758	178 741	31.04	1.4 (±0.2)	15.4 (±1.0)	2192.3 (±346.6)	670.5 (±168.1)	281.6 (±101.6)

6.6 Comparison to finding dense communities. In addition to our proposed algorithms, we also obtain communities by applying the GreedyFast algorithm of Sozio and Gionis [10], denoted here as Cocktail, on the networks with the same sets of seed individuals. The aim of this algorithm is to find a subgraph that connects the seeds while maximizing the minimum degree among selected nodes. This algorithm considers neither profiles nor social tension; Nevertheless, we can compute the tension of the community that consists of the nodes returned as a solution and compare it to the communities obtained with our algorithms. Furthermore, Cocktail requires the user to set a value for the upper bound on the size of the solution. We set this value to $k = 200$, as it seems to result in reasonable runtimes while allowing the algorithm to construct a solution in most cases. The cases where the algorithm fails to return a solution are left out from our statistics.

On the C.Papadimitriou network the solutions returned by Cocktail are comparable in size to those of our CTree(s) algorithm (middle column in each block of Figure 5). The quality of edges selected in its solutions is on par with CTree(e) and CPeel(r) (right hand side column), which are also oblivious to the profiles of individuals. Expectedly, Cocktail appears poorly suited for the task of finding low-tension communities.

6.7 Impact of other factors. Further investigations (details omitted) show variations in the performance of the algorithms depending on the network structure. Indeed, in a high density network, favoring low tension paths as done by CTree(s) can result in many more edges in the induced subgraph, yielding a significantly higher social tension and actually hurting the performance.

We also looked at the impact of the distribution of profile values on the behavior of the algorithms. In order to do so, we constructed random latent profiles under different sampling distributions (e.g. uniform and exponential). We could observe variations between the different random distributions, but these differences appeared to be limited when contrasted with the gap that exists between random and eigenvector profiles. Furthermore, while the profile-aware variants generally tend to pick edges with lower tension at the cost of involving more edges, as discussed earlier, this tendency is more pronounced when handling eigenvector profiles as compared to randomly distributed profiles.

We conclude from these observations that the distribution of profile values has a limited impact compared to the presence of structure in eigenvector profiles and that the profile-aware variants are clearly suited to exploit this structure.

6.8 Running times. Indicative running times of the algorithms on networks of varying sizes and densities, with multi-attribute profiles are listed in Table 1. As expected, the tree-based algorithms are significantly faster than the top-down algorithms, up to two orders of magnitude, and scale much better.

For a comparison between multi-attribute profiles and single-attribute profiles, we look at the running times on the `ICDM` network with single-attribute profiles:

`CTree(e)`	`CTree(s)`	`CPeel(s)`	`CPeel(m)`	`CPeel(r)`
0.2 (±0.0)	3.5 (±0.5)	147 (±26.6)	107 (±24.9)	37.0 (±12.1)

We observe that, also as expected, going from single-attribute to multi-attribute profiles hardly has any impact on the running times of our algorithms.

7 Conclusions

Problems related to community search have multiple applications in online social media and collaboration networks. In this paper, we add a new modeling angle to this class of problems. The key characteristic of our model is that each node of the social network is not only characterized by its connections and its skills, but also by its profile. These profiles, which change dynamically through a conformation process, give rise to social tension in the network. Given this model, we define the T-Comm problem, where the goal is to identify a set of connected individuals that define a low-tension subgraph. Such a problem arises both in social network and social media mining as well as in human-resource management, where the goal is to find a set of workers who are not only connected, but also will have a potentially fluid collaboration. The contributions of our paper include the formal definition of this problem and the design of algorithms for solving it effectively in practice. Our experimental results with real data from social and collaboration networks highlight the characteristic behavior of the different algorithms variants and illustrate the effect of network structure and profile distribution on the algorithms' relative performance. Finally, our work enables future research combining subgraph mining with dynamic processes occurring among the nodes.

Acknowledgements. Most of the work was done while Esther Galbrun and Behzad Golshan were at Boston University. Aristides Gionis is supported by the Finnish Funding Agency for Innovation TEKES (project "Re:Know"), the Academy of Finland (project "Nestor"), and the EU H2020 Program (project "SoBigData"). This research was funded by NSF grants: IIS 1320542, IIS 1421759 and CAREER 1253393 as well as a gift from Microsoft.

References

[1] D. Bindel, J. Kleinberg, and S. Oren. How Bad is Forming Your Own Opinion? In *FOCS*, pages 57–66, 2011.

[2] P. Clifford and A. Sudbury. A Model for Spatial Conflict. *Biometrika*, 60(3):pp. 581–588, 1973.

[3] M. H. DeGroot. Reaching a Consensus. *Journal of the American Statistical Association*, 69(345):pp. 118–121, 1974.

[4] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *ACM SIGKDD*, pages 118–127, 2004.

[5] N. E. Friedkin and E. C. Johnsen. Social influence and opinions. *The Journal of Mathematical Sociology*, 15(3-4):193–206, 1990.

[6] N. E. Friedkin and E. C. Johnsen. Social influence networks and opinion change. *Advances in Group Processes*, 16:1–29, 1999.

[7] A. Gionis, E. Terzi, and P. Tsaparas. Opinion Maximization in Social Networks. In *SDM*, pages 387–395, 2013.

[8] R. A. Holley and T. M. Liggett. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. *The Annals of Probability*, 3(4):pp. 643–663, 1975.

[9] N. Ruchansky, F. Bonchi, D. García-Soriano, F. Gullo, and N. Kourtellis. The Minimum Wiener Connector Problem. In *ACM SIGMOD*, pages 1587–1602, 2015.

[10] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *ACM SIGKDD*, 2010.

[11] H. Tong and C. Faloutsos. Center-piece subgraphs: Problem definition and fast solutions. In *ACM SIGKDD*, pages 404–413, 2006.

[12] V. Vazirani. *Approximation Algorithms.* Springer, 2003.

[13] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione. Binary Opinion Dynamics with Stubborn Agents. *ACM Trans. Econ. Comput.*, 1(4):19:1–19:30, 2013.