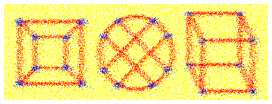
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 18, no. 1, pp. 1–34 (2014)
DOI: 10.7155/jgaa.00313

The Complexity of the Simultaneous Cluster Problem

Zhentao Li[1] *Manikandan Narayanan*[2] *Adrian Vetta*[3]

[1]Laboratoire de l'Informatique du Parallélisme,
École Normale Supérieure de Lyon.
[2]Present address: National Institute of Allergy and Infectious Diseases,
National Institutes of Health, Bethesda, MD.
[3]Department of Mathematics and Statistics, and
School of Computer Science, McGill University.

Submitted:	Reviewed:	Accepted:	Final:	Published:
January 2013	December 2013	December 2013	January 2014	January 2014

Article type: Regular paper
Communicated by: I.G. Tollis

E-mail addresses: zhentao.li@ens-lyon.fr (Zhentao Li) manikandan.narayanan@nih.gov (Manikandan Narayanan) vetta@math.mcgill.ca (Adrian Vetta)

Abstract

We study clustering over multiple graphs - each encoding a distinct set of similarity relationships (edges) over the *same* set of objects (nodes) - where the aim is to identify clusters that are supported across the collection of graphs. This problem of *simultaneous clustering* is readily motivated by the recent deluge of datasets in several domains (including the biological sciences, social sciences, and marketing), where the same objects are repeatedly measured in different conditions, populations or time points. Whilst there has been a vast amount of heuristic work on practical simultaneous clustering problems, little is known on the theoretical side – we present theoretical results that help explain why such heuristics typically come without quantitative guarantees. We give algorithmic and complexity results for simultaneous clustering using two standard measures on clustering quality: density and connectivity. Specifically, we focus on the basic problem of finding a *single* cluster (rather than an entire clustering) that is simultaneously of high quality in every graph. When the quality of a cluster is its minimum density over all graphs, we show the problem is not approximable within a factor of $2^{\log^{1-\varepsilon} n}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$. Furthermore, this problem appears very difficult even when there are just two graphs; the resulting problem is approximately as hard as the problem of finding a dense subgraph on at most k vertices. When cluster quality is a fixed connectivity requirement between terminals within the cluster, there are two natural optimization problems: a maximization version (find a good quality cluster with as many terminals as possible) and a minimization version (find a good quality cluster that is as small as possible). We show that the maximization problem is tractable in polynomial time for any fixed connectivity requirement k. On the other hand the minimization problem is hard to approximate within a factor of $2^{\log^{1-\varepsilon} n}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$. The number of graphs in our reduction depends on n. If instead the number of graphs is fixed, we show there is an $\varepsilon > 0$ for which the minimization problem is not approximable within $g^{1/2-\varepsilon}$ for any fixed number g of graphs unless $NP = ZPP$. These hardness results for the minimization problem hold even in the simple cases where the connectivity requirement is one and there are either just two terminal nodes or every node is a terminal node. We remark that our results extend to case where more robust variants of the quality measure are used.

1 Introduction

The problem of clustering – partitioning a set of objects into similar groups based upon a graph of similarity relationships defined over the objects – is ubiquitous. Applications abound in data mining, with clustering being a primary choice for exploratory data analysis in various domains such as biology [16], medicine [44], marketing [25], and social network analysis [45]. Our interest in clustering derives from the recent, rapid accumulation of datasets in such domains, where measurements are taken on the *same* set of objects repeatedly under different experimental conditions, time points, or populations. This yields a collection

of graphs defined over the same set of objects (nodes) but with different sets of relations (edges) amongst them. This, in turn, calls for a new paradigm of clustering that jointly analyses multiple graphs to identify common signals and conserved clusters.

This paradigm is very relevant in the biological sciences for instance, where the replication of a discovery (for example, functional similarity of a set of genes) is often sought across multiple, independent datasets to minimize spurious findings caused by noise/artifacts in individual datasets and to exploit the complementarity of the datasets [30, 21, 39]. With advances in high-throughput instruments, there is a deluge of molecular data on the same biological system generated using different experimental backgrounds, perturbation techniques and technological platforms.

Each dataset comes with its own set of biases and artifacts due to these differences, and calls for methods that integrate diverse datasets more carefully than simply concatenating or combining them into one dataset or similarity graph prior to clustering. Machine-learning methods could be used to carefully integrate multiple datasets into one similarity function, but they typically rely heavily on domain knowledge in the form of training data and model assumptions [22]. We are interested in a problem abstraction that naturally extends single-graph clustering to multiple graphs and is suitable for the exploratory or "unsupervised" setting where there is no training data.

Our goal, therefore, is to obtain a clustering that is *good* over a collection of graphs, $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$ that share the same set of nodes. We dub this problem *simultaneous clustering*. Of course, in order to assess whether a clustering is good we must specify a measure of quality. For example, in this paper we use perhaps the two most natural and widely-studied attributes associated with a cluster, namely *density* and *connectivity*. Thus, a clustering will be good if it induces dense or highly connected clusters in each of the graphs G_i, even though the actual edge sets induced may vary widely between the graphs. In Section 2 we will see how these two measures arise in biological studies aimed at discovering sets of functionally coherent genes and complexes/scaffolds of interacting proteins. First, though, we formalise the problem and state our results.

1.1 Our Results.

We are given a collection of graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$, where $G_i = (V, E_i)$ for each $1 \leq i \leq t$, and a quality measure. A *clustering* is a partition of V into subsets $S_1, S_2, \ldots, S_\ell$; each S_i is called a *cluster*. We restrict our attention to the fundamental problem of finding a single cluster $S \subseteq V$ that is good, that is, has at least a specified quality q^* in the subgraph $G_i[S]$ it induces in each graph G_i. We call this the *simultaneous cluster problem* and show that it is polynomially tractable in a few cases but is typically very hard.

`Simultaneous Cluster Problem.`
Input: Graphs $G_i = (V, E_i)$, where $1 \leq i \leq t$, and a quality threshold q^*.
Objective: A cluster $S \subseteq V$ such that the quality of $G_i[S]$ is at least q^* for all i.

As stated the two quality measures we will consider are density and (terminal) connectivity.

• We define the density of a cluster S in a collection of graphs to be

$$\text{den}(S; \mathcal{G}) = \min_{G_i \in \mathcal{G}} \text{den}_i(S) = \min_i \frac{|E_i[S]|}{|S|}$$

where $E_i[S]$ is the set of edges in the graph $G_i[S]$ induced by the vertex set S.

• Given a set of terminals $T \subseteq V$, we define the (terminal) connectivity of a cluster S in a collection of graphs to be

$$\kappa(S; \mathcal{G}) = \min_{G_i \in \mathcal{G}} \kappa_i(S)$$

where $\kappa_i(S)$ is the minimum pairwise connectivity between terminals $T \cap S$ in $G_i[S]$.

For the density measure, our first result shows that there is major difference in hardness when we move from a single graph to just two graphs. Specifically the densest subgraph problem is polynomial time solvable with one graph (see Chapter 4 of [29], [34] and [19]), but for two graphs we prove the following for densest simultaneous cluster:

Theorem 1. *If we can solve* Densest Simultaneous Subgraph *on two graphs in polynomial time then we can solve* Densest k-Subgraph *in polynomial time.*

Theorem 2. *If we can approximate* Densest Simultaneous Subgraph *on two graphs within a factor of* α *then we can approximate* Densest k-Subgraph *within a factor of* $4\alpha^2$.

Here Densest k-Subgraph refers to the problem of finding the densest subgraph on at most k vertices given an input graph G and a number k. This problem can be approximated to within a factor of $O(n^{\frac{1}{4}+\varepsilon})$, due to a recent breakthrough result of [9]. Our result is of interest because it is widely believed [10, 3, 17, 18] that the hardness of Densest k-Subgraph is also close to this upper bound – indeed, Bhaskara et al. [10] present $O(n^{\Omega(1)})$ lower bounds for lift and project methods based upon the Sherali-Adama and the Lassere hierarchies. If so, whilst a size restriction is clearly vital with regards to complexity in the case of a single graph, it is redundant in the case of two graphs - there the problem is very hard even when no size restrictions are given.

To complement this result, we show that the problem does have large inapproximability bounds when the number of graphs gets large.

Theorem 3. Densest Simultaneous Subgraph *is not approximable within* $2^{\log^{1-\varepsilon} n}$ *for any* $\varepsilon > 0$*, unless* $NP \subseteq DTIME(n^{\text{polylog}n})$.

In fact, this hardness result also applies to the problem of finding a minimum cardinality subset that has non-zero density in each graph, i.e. $\text{den}(S;\mathcal{G}) > 0$. That is, the simple problem of finding the smallest cluster that induces at least one edge in many graphs is very hard to approximate. So if in an application the functionality (quality) of a cluster S is defined to simply depend upon whether or not at least two nodes in that cluster can interact then, from an approximation viewpoint, we are already in trouble! This helps explain why heuristics for many clustering problems with more complex quality measures, e.g. in bioinformatics, typically come without quantitative guarantees.

For the terminal connectivity measure, we fix the desired connectivity k for determining whether a cluster is good and study two natural optimization criteria. We first present good news for finding a good cluster with as many terminals as possible.

Theorem 4. *For a fixed connectivity requirement k, there is a polynomial time algorithm for* Maximum Simultaneous k-Connected Steiner Cluster.

As connectivity is a monotonic property with regards to the addition of non-terminal nodes, this maximization criteria could produce large clusters that contain extraneous nodes in some scenarios. So we also study the problem of finding a good cluster with as few nodes as possible. We show this is hard to approximate even in the extreme cases of just two terminals $\{s,t\}$ or all nodes being terminals, even when the connectivity requirement $k = 1$.

Theorem 5. Simultaneous s-t Path *is not approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

Theorem 6. Minimum Simultaneous Connected Steiner Cluster *is not approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

In fact, we obtain inapproximability results that scale with the number of input graphs.

Theorem 7. Simultaneous s-t Path *is not $g^{1/2-\varepsilon}$-approximable for some $\varepsilon > 0$ where g is the number of graphs unless $NP = ZPP$.*

Theorem 8. Minimum Simultaneous Connected Steiner Cluster *is not $g^{1/2-\varepsilon}$-approximable for some $\varepsilon > 0$ where g is the number of graphs unless $NP = ZPP$.*

These hardness results for clustering many graphs also extend to robust variants of the problems where the optimal solution is only required to satisfy the quality (density or connectivity) constraint in a c fraction of the g input graphs. This follows readily as an algorithm for the robust variant ($c < 1$) can be used to solve an instance of the exact variant ($c = 1$) by adding $(g/c - g)$ empty graphs. For example, this would mean that maximizing the median density ($c = \frac{1}{2}$) of a subgraph in the input graphs is at least as hard as maximizing the minimum density of a subgraph in the original graphs. When clustering many graphs, if we let the optimal solution satisfy the quality constraint in all graphs

as in the original problem definitions, but relax the approximation algorithm to return a solution that satisfies the constraint in only a c fraction of the input graphs, this c-relaxed approximate solution is still hard to find.

Theorem 9. Densest Simultaneous Subgraph *is not c-relaxed approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$ and constant $c > \frac{2}{3}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

Theorem 10. Simultaneous s-t Path *is not c-relaxed approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$ and constant $c > \frac{1}{2}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

Theorem 11. Minimum Simultaneous Connected Steiner Cluster *is not c-relaxed approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$ and constant $c > \frac{4}{5}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

We prove our results for the density and connectivity measures in Sections 4 and 5, respectively. Before doing so, in Section 2, we describe in detail how the problem of simultaneous clustering arises naturally in bioinformatics, and discuss the techniques and heuristics currently used for such problems. We then compare, in Section 3, our problem to previous work in stochastic optimization where there are multiple inputs (or scenarios).

2 Simultaneous Clustering in Bioinformatics

The major motivation underlying this work is the abundance in bioinformatics of simultaneous clustering problems based upon connectivity and, especially, density quality measures. So, in this section we give a detailed and slightly technical overview of why such problems arise and give a guide to some of the research that has been carried out in this area. This provides context for our research but a reader solely interested in the theoretical aspects of the underlying combinatorial problem may chose to proceed to the next section.

Interactions between genes, proteins and other molecules form the basis of most cellular processes, and large-scale measurements of such interactions are now routine in the life sciences [23]. For instance, it is possible to monitor the activity or expression patterns of thousands of genes in an organism across many replicates, and currently more than 22,000 such expression datasets from different studies are available in a public resource called GEO [8]. An expression dataset can be used to build a coexpression network, whose nodes are monitored genes and whose edges are gene pairs with similar activity patterns. If the activity patterns are measured in a sufficient number of systematically perturbed replicates, the edges in a coexpression network correspond to functionally related gene pairs. This idea is central to a large number of bioinformatic studies that discover new (or characterize known) biological processes by systematically identifying densely connected clusters in the coexpression network [48, 16]. A similar approach is widely used to identify connected scaffolds or dense complexes of physically interacting proteins from a genome-wide network of protein-protein interactions [36].

The joint analysis of multiple biological graphs is becoming increasingly important for two major reasons. The first reason is statistical - each dataset is a noisy measurement of the true functional relation of genes, hence discoveries (functionally coherent genes/protein clusters) supported by independent coexpression or protein interaction networks are more robust against artifacts in individual datasets [39, 30]. The other reason is biological - interesting insights into the evolution and regulation of biological systems are sometimes possible only by integrating diverse datasets obtained from different species, cell types or conditions [37, 27].

Several techniques and heuristics are employed to address the related problems above. A common strategy is to frame the problem of finding protein complexes in a single protein network [36] or finding evolutionarily conserved complexes in multi-species networks [37, 27] as locating heavy subgraphs in a single weighted "alignment graph". The node and edge weights of this alignment graph aggregates the features of each input network using a biologically-motivated scoring scheme or Bayesian model. A node in the alignment graph for instance could represent a gene in the input networks for genes exhibiting one-to-one evolutionary relationship in multiple species and a gene family for genes in one species that are related to multiple genes in other species. A heuristic that starts with seed nodes and greedily adds or removes nodes to these seeds is then used to optimize the score of the induced subgraph of the alignment graph. When certain criteria based on the connectivity and monotonic local similarity between proteins in different species were used to define evolutionary conservation, a provably efficient algorithm based on a recursive approach was possible for finding conserved protein complexes [31].

The problem of finding connected subnetworks in one network (protein network) that is dense or high-scoring in another network (coexpression network) has been addressed using greedy heuristics too [43]. Spectral techniques found use in a related problem of finding a clustering that maximizes the connectedness of each cluster and minimizes the weight of edges lost between the clusters in all input biological networks [32]. Different notions of terminal connectivity were explored to find protein interactions that optimally explain the differential activity of a set of genes and thereby expand our current knowledge of proteins/genes involved in certain biological processes [46]. Algorithms for finding k-cliques (for small k) have been used as subroutines to uncover the structure and evolution of overlapping clusters in biological and social networks [33, 49]. Recently, a study used simulated annealing to detect disease-specific genes that clustered in hundreds of coexpression networks [30]. So it is not exactly a steiner problem but there are some similarities.

Clearly, the exact models, heuristics and algorithms used in the multi-graph methods above are driven mainly by biological considerations. As stated, our aim in this paper is to provide a computational treatment of the underlying simultaneous clustering problem. In particular, whilst we show that good algorithms are possible with some quality measures, our main contribution is to give an explanation for why quantitative guarantees have been elusive in previous works.

3 Related Work

Our work bears some relation to the field of stochastic optimization which encompasses optimization problems that are robust to uncertainty in the input data. The uncertainty is modeled by a probability distribution over possible realizations (scenarios) of the input data, and the objective function involves minimizing the expected cost (or maximizing the expected profit) of the algorithm [12]. The framework also includes other robustness measures such as minimizing the maximum cost across all (or a large fraction of) scenarios [42, 41] or permitting the cost in each scenario to be worse by a factor of p than the optimal cost in that scenario [2, 38, 1].

Given this generic definition, the simultaneous clustering problem could be considered as a stochastic optimization problem where the graphs with different edge weights are the different scenarios and we seek a set of common clusters that are robust (of good quality) in all input scenarios (graphs). Existing works on approximation algorithms or complexity results of stochastic optimization problems focus either on problems not closely related to clustering such as covering problems or finance-related problems, or on facility location problems that differ in several ways from the clustering model considered in this work.

For the simultaneous clustering problem, our objective is to minimize the maximum cost across all scenarios (the so-called min-max objective). Complexity results have been obtained for non-clustering problems with this objective. Strong NP-hardness is known for the shortest path problem [47], the assignment problem (bipartite matching) and the knapsack problem [26]. Set cover with min-max objective is known to be hard to approximate (as hard as Densest k-Subgraph) [4]. These results are for the cases where the number of scenarios is also given as input. Weak NP-hardness results are also known when the number of scenarios is fixed [2]. Our inapproximability results for simultaneous clustering with the density measure apply when there are only two scenarios, also reducing from Densest k-Subgraph (our reduction differs from the one given for set cover). To our knowledge, the closest work to ours is for the min-max version of the k-centre problem [11]. There the problem is studied with different scenarios in order, for example, to account for the congestion effects of rush hours. They gave a simple but elegant 3-approximation algorithm for the case of two scenarios but show the problem is inapproximable for three scenarios. As well as the quality measure, their work differs from ours in one important aspect. Whilst the single time-interval version of the k-centre problem can be viewed as a clustering (around centres) problem on one graph, the min-max variant is *not* a clustering problem because nodes can be serviced by different centres in different scenarios. Indeed, it is easy to show that the simultaneous clustering version of the k-centre problem has a factor 2-approximation for any number of graphs, as it reduces to the single graph case. There is also a rich body of work on other stochastic uncapacitated facility location (SUFL) problems where the objective is to find an optimal set of facilities to robustly serve a set of clients. The uncertainty could be in the demands of the clients (eg., which clients need service), the client locations and hence their distances to the

facilities or other input parameters, and are modeled using single/multiple stage stochastic models [40, 4, 38]. These problems typically differ from ours in many respects: in the choice of measure and objective function, in that they cease to be clustering problems in the multiple scenario case, and in that they only use a single distance metric between the clients across all scenarios (eg. [4])[1].

4 The Density Measure

To begin our study into the simultaneous cluster problem, we consider the density measure.

`Densest Simultaneous Subgraph Problem`
Input. Graphs $G_i = (V, E_i)$ for $1 \leq i \leq t$.
Objective. A set $S \subseteq V$ maximizing $\min_{1 \leq i \leq t} \text{den}_i(S)$.
(Here den_i is the density of the graph induced by S in G_i.)

For the "non-simultaneous" case of a single graph, that is $t = 1$, Densest Simultaneous Subgraph is equivalent to the densest subgraph problem and so is solvable in polynomial time [29, 34, 19]. For the simultaneous case, in this section, we consider the complexity of the cases $t = 2$ and t large. We reduce the two graphs problem to the single graph problem where the solution is restricted to have at most k vertices, a problem widely believed to be difficult to approximate. We reduce the case where t is large to LabelCover-Max and consequently, show this problem is inapproximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.

4.1 Clustering Two Graphs

So let's consider the simultaneous cluster problem with exactly two graphs under the density measure. As noted, finding the densest subgraph in a single graph is easy. This is certainly not the case with two graphs. Specifically, here we show that finding a vertex set that simultaneously induces dense subgraphs in two graphs is approximately as hard as finding a densest subgraph on at most k vertices in a single graph:

`Densest` k`-Subgraph`
Input. A graph $G = (V, E)$ and a number k.
Objective. An induced subgraph H^* of maximum density containing at most k vertices.

To obtain this hardness result, we begin by showing how a polynomial time algorithm for Densest Simultaneous Subgraph in two graphs would lead to a polynomial time algorithm for Densest k-Subgraph. Then, we adapt those

[1] Recall how the simultaneous clustering problem seeks common clusters of nodes across all graphs (so invariant cluster composition across scenarios), with each graph with different edge weights implying a different distance metric.

techniques to show how inapproximability bounds (whatever they may be!) are also roughly maintained between these two problems.

Theorem 1. *If we can solve* Densest Simultaneous Subgraph *on two graphs in polynomial time then we can solve* Densest k-Subgraph *in polynomial time.*

Proof. Note that, for a fixed n, there are at most n^3 possible different density values. Therefore, we can assume that the optimal density d is fixed; that is, we know d whenever needed.

Now, given an instance (G, k) of Densest k-Subgraph we reduce it to an instance of Densest Simultaneous Subgraph on two graphs, G_1 and G_2. We actually build G_1 and G_2 out of two graphs, G_1' and G_2', on disjoint vertex sets by taking their disjoint union. So edges in G_1 have both endpoints in G_1' and edges in G_2 have both endpoints in G_2'. We use the notation $n_i = |V(G_i')|$ and $d_i = \text{den}(G_i')$ for $i = 1, 2$. Obtaining the first graph G_1' is simple; we just set $G_1' = G$. Obtaining G_2' is a little more complex. We desire G_2' to have the following two properties:

(I) It is a minimum cardinality graph with exactly dk edges.

(II) All of its proper subgraphs are strictly less dense.

Observe that if G_2' satisfies Property (I) then it must have density $d_2 = \frac{dk}{n_2}$. Furthermore, since G_2' contains as few vertices as possible,

$$\binom{n_2 - 1}{2} \; < \; dk \; = \; n_2 d_2 \; \leq \; \binom{n_2}{2}$$

and thus, dividing by $n_2/2$, we obtain

$$\frac{(n_2 - 2)(n_2 - 1)}{n_2} \; < \; 2d_2 \; \leq \; n_2 - 1$$

Now G_2' contains $r \geq 0$ edges less than the complete graph on n_2 vertices, K_{n_2}. It must be the case that $r \leq n_2 - 2$, otherwise the clique $K_{n_2 - 1}$ has at least as many edges as G_2'. So, we can construct G_2' by removing r edges from K_{n_2}. We need to choose these edges judiciously, in order for Property (II) to hold. Towards this goal let $P = \{e_1, e_2, \ldots, e_{n_2 - 1}\}$ form a Hamiltonian path in K_{n_2}. Let M_1 consist of the odd indexed edges in P, and let M_2 be the even edges. Then to build G_2' we remove the r edges by first deleting edges of M_1 and then deleting edges of M_2 in reverse order.

Suppose that we are required to remove edges from M_2, that is, $r > \frac{1}{2}n$. Then the maximum degree, $\Delta(G_2')$, is $n_2 - 2$ and the minimum degree, $\delta(G_2')$, is $n_2 - 3$. If not, the maximum and minimum degrees are bounded by $n_2 - 1$ and $n_2 - 2$, respectively. We may now show that G_2' does satisfy Property (II).

Claim 1. *Every proper subgraph of G_2' is less dense than G_2'.*

Proof. To show every proper subgraph H of G'_2 has lower density, we consider three cases. Two cases are simple. If H has n_2 vertices then, as it is a proper subgraph of G'_2, it has fewer edges so is less dense. If H has at most $n_2 - 2$ vertices then the maximum degree $\Delta(H)$ is at most $n_2 - 3$. Consequently, the average degree in H is at most $n_2 - 3$. However, G'_2 has average degree strictly greater than $n_2 - 3$, as, by construction, it always has a vertex of degree at least $n_2 - 2$. So H is less dense. So consider the case where H has $n_2 - 1$ vertices. Then

$$\text{den}(G'_2) \;=\; \frac{n_2 \cdot \Delta(G'_2) - 2r}{2n_2} \;=\; \frac{1}{2}\Delta(G'_2) - \frac{r}{n_2}$$

Furthermore

$$\begin{aligned} \text{den}(H) &\leq \frac{|E(G'_2)| - 2\delta(G'_2)}{2(n_2 - 1)} \\ &= \frac{n_2 \cdot \Delta(G'_2) - 2r - 2\delta(G'_2)}{2(n_2 - 1)} \\ &\leq \frac{n_2 \cdot \Delta(G'_2) - 2r - 2(\Delta(G'_2) - 1)}{2(n_2 - 1)} \end{aligned}$$

and thus

$$\text{den}(H) \leq \frac{1}{2}\Delta(G'_2) - \frac{(\Delta(G'_2) + 2r - 2)}{2(n_2 - 1)} \leq \frac{1}{2}\Delta(G'_2) - \frac{r}{n_2} = \text{den}(G'_2)$$

The last inequality holds provided $n_2(\Delta(G'_2) - 2) + 2r \geq 0$, that is if $\Delta(G'_2) \geq 2$. This is true if $n_2 \geq 4$. □

We may now complete the description of our Densest Simultaneous Subgraph instance (G_1, G_2). Given G'_1 and G'_2, as above, set $V(G_i) = V(G'_1) \cup V(G'_2)$ and $E(G_i) = E(G'_i)$, for $i = 1, 2$. Now suppose there is a subgraph H^* of cardinality k and density d in $G = G'_1$. Then the value of solution $H^* \cup V(G'_2)$ in our instance of Densest Simultaneous Subgraph is

$$D^* \;=\; \min\left(\text{den}_1(H^* \cup V'_2), \text{den}_2(H^* \cup V'_2)\right) \;=\; \min\left(\frac{kd}{n_2 + k}, \frac{n_2 d_2}{n_2 + k}\right)$$

Note that since $dk = n_2 d_2$, the two terms inside the min are the same. Since we assumed we know the optimal density d in G, the optimal solution to our instance of Densest Simultaneous Subgraph has value at least D^*. (Algorithmically, if the optimum is less than D^*, we can stop our search for this value of d and claim that the optimal density for G is lower than d.)

It remains to show that an optimal solution $H_1 \cup H_2$ of value at least D^*, where $H_1 \subseteq G'_1$ and $H_2 \subseteq G'_2$, produces a subgraph of density at least d with at most k vertices in G. We let $k = \beta n_2$, $|H_1| = \tau_1 k$, and $|H_2| = \tau_2 n_2$. As n_2 is the cardinality of the smallest graph with dk edges, it must be the case that $k \geq n_2$ (since the desired subgraph H^* in G has k vertices and dk edges). So $k = \beta n_2$ for some $\beta \geq 1$.

We now have several cases to consider.

(a) $|H_1| \geq k$

(i) If $H_2 = G_2'$ then $\text{den}_2(H_1 \cup G_2') \leq \frac{n_2 d_2}{n_2+k} = D^*$ and equality holds only when $|V(H_1)| = k$. In that case, we can return H_1 as it then has size k and density d in $G_1 = G$.

(ii) If $|H_2| = t \leq n_2 - 1$ then, by Claim 1,

$$\text{den}_2(H_1 \cup H_2) < \frac{td_2}{k+t}.$$

Now $\frac{td_2}{k+t}$ is increasing in t, so is maximized when $t = n_2 - 1$. We then have

$$\text{den}_2(H_1 \cup H_2) \;\leq\; \frac{(n_2-1)d_2}{k+n_2-1} \;<\; \frac{n_2 d_2}{k+n_2} \;=\; D^*$$

Here the strict inequality follows by simple algebra. So $\text{den}_2(H_1 \cup H_2) < D^*$ which is a contradiction.

(b) $|H_1| < k$

Suppose $\text{den}_1(H_1 \cup H_2) > \frac{dk}{n_2+k} = D^*$. Then

$$\frac{dk}{n_2(\beta+1)} \;<\; \frac{|E(H_1)|}{n_2(\tau_1\beta+\tau_2)} \;<\; \frac{\tau_1 dk}{n_2(\tau_1\beta+\tau_2)}$$

The second inequality follows as $|E(H_1)| < d|H_1| = d\tau_1 k$. Thus, $(\tau_1\beta + \tau_2) < \tau_1(\beta+1)$. It follows that $\tau_2 < \tau_1$. In particular, we have $\tau_2 < 1$ because $\tau_1 < 1$. Therefore $|V(H_2)| = \tau_2 n_2 \leq n_2 - 1$. We now show that $\text{den}_2(H_1 \cup H_2) < D^*$.

$$\begin{aligned}
D^* &= \frac{n_2 d_2}{n_2(\beta+1)} \\
&= \frac{\tau_2 n_2 d_2}{\tau_2 n_2(\beta+1)} \\
&= \frac{|H_2| \cdot d_2}{\tau_2 n_2(\beta+1)} \\
&> \frac{|H_2| \cdot \text{den}(H_2)}{\tau_2 n_2(\beta+1)} \qquad \text{[By Claim 1]} \\
&> \frac{|E(H_2)|}{\tau_2 n_2(\beta+1)} \\
&> \frac{|E(H_2)|}{\tau_1\beta n_2 + \tau_2 n_2} \\
&= \frac{|E(H_2)|}{|H_1| + |H_2|} \\
&= \text{den}_2(H_1 \cup H_2)
\end{aligned}$$

This contradicts the optimality of $H_1 \cup H_2$ and the result follows. □

Lemma 1. *If we can approximate* Densest Simultaneous Subgraph *on two graphs within a factor of α then we can approximate* Densest k-Subgraph *within a factor 2α with a solution of size at most $(2\alpha - 1)k$.*

Proof. Again, we assume the optimal density d is known when needed. Given an instance (G, k) of Densest k-Subgraph we reduce it to an instance (G_1, G_2) of Densest Simultaneous Subgraph as before. Again, if there is a subgraph H in $G = G'_1$ of cardinality k and density d then the value of solution $H \cup V(G'_2)$ in our instance of Densest 2-Simultaneous Subgraph is

$$D^* \;=\; \min\left(\mathrm{den}_1(H \cup V'_2), \mathrm{den}_2(H \cup V'_2)\right) \;=\; \min\left(\frac{kd}{n_2 + k}, \frac{n_2 d_2}{n_2 + k}\right)$$

Note that since $dk = n_2 d_2$, the two terms inside the min are the same. Moreover, as n_2 is the cardinality of the smallest graph with dk edges, it must be the case that $k \geq n_2$. So $k = \beta n_2$ for some $\beta \geq 1$. Now take a solution $H_1 \cup H_2$ output by the approximation algorithm, where $H_1 \subseteq V(G) = V(G'_1)$ and $H_2 \subseteq V(G'_2)$. Then

$$\min\left(\mathrm{den}_1(H_1 \cup H_2), \mathrm{den}_2(H_1 \cup H_2)\right) \;\geq\; \frac{1}{\alpha} D^*$$

We will show that H_1 is an approximate solution to the instance of Densest k-Subgraph. Again, assume that $|H_1| = \tau_1 k$, and $|H_2| = \tau_2 n_2$.

We now have several cases to consider.

(a) $\tau_1 > 2\alpha - 1$

(i) If $H_2 = G'_2$ then

$$\begin{aligned}
\mathrm{den}_2(H_1 \cup G'_2) &= \frac{|E(G'_2)|}{|H_1| + n_2} \\
&< \frac{|E(G'_2)|}{(2\alpha - 1)k + n_2} \\
&\leq \frac{|E(G'_2)|}{\alpha(k + n_2)} && \text{[By algebra, as } \beta \geq 1\text{]} \\
&= \frac{1}{\alpha} \cdot \frac{n_2 d_2}{k + n_2}
\end{aligned}$$

Thus we obtain a contradiction.

(ii) If $|H_2| = t \leq n_2 - 1$ then

$$\begin{aligned}
\mathrm{den}_2(H_1 \cup H_2) &= \frac{|E(H_2)|}{|H_1| + |H_2|} \\
&\leq \frac{\binom{t}{2}}{|H_1| + t} \\
&\leq \frac{\binom{n_2-1}{2}}{|H_1| + (n_2 - 1)} && \text{[As } \tfrac{\binom{t}{2}}{x+t} \text{ is increasing in } t\text{]} \\
&\leq \frac{\binom{n_2-1}{2}}{((2\alpha-1)k + 1) + (n_2 - 1)} \\
&= \frac{\frac{1}{2}(n_2 - 1)(n_2 - 2)}{(2\alpha - 1)k + n_2} \\
&< \frac{d_2(n_2 - 1)}{(2\alpha - 1)k + n_2} \\
&\leq \frac{d_2 n_2}{\alpha(k + n_2)} && \text{[By algebra, as } \beta \geq 1\text{]} \\
&= \frac{1}{\alpha} \cdot \frac{n_2 d_2}{k + n_2}
\end{aligned}$$

This is a contradiction. So H_1 is at most a factor $2\alpha - 1$ larger than H.

(b) $\tau_1 \leq 2\alpha - 1$

$$\mathrm{den}_1(H_1 \cup H_2) = \frac{|E(H_1)|}{|H_1| + |H_2|} < \frac{|E(H_1)|}{|H_1|}$$

and

$$\mathrm{den}_1(H_1 \cup H_2) \geq \frac{1}{\alpha} D^* = \frac{1}{\alpha} \cdot \frac{dk}{k + n_2} \geq \frac{1}{2\alpha} d$$

Again, the last inequality arises as $\beta \geq 1$. Hence

$$2\alpha \frac{|E(H_1)|}{|H_1|} > d$$

So H_1 contains at most $(2\alpha - 1)k$ vertices and has a density within a factor 2α of the densest subgraph on k vertices in G_1 so we can return H_1 as an approximate solution. □

Lemma 2. *Let G be a graph on k_1 vertices and density d_1 then for any $k_2 = \gamma k_1$, there exists a subgraph of G on k_2 vertices with density γd_1.*

Proof. Randomly choose a subset V_2 of size k_2 with each subset equally likely. Then each edge appears with probability

$$\frac{\binom{k_1-2}{k_2-2}}{\binom{k_1}{k_2}} \geq \left(\frac{k_2}{k_1}\right)^2$$

in the subgraph H induced by V_2. Since the total number of edges in G is d_1k_1, it follows that the expected number of edges in H is $\frac{d_1k_2^2}{k_1}$. Thus, there exists a subgraph with this many edges and density $d_1\frac{k_2}{k_1} = \gamma d_1$. □

4.2 Clustering Many Graphs

Our hardness result for two graphs is compelling but, given the current state of knowledge, it still remains possible that there are constant factor approximation algorithms for Densest Simultaneous Subgraph in two graphs. For the case of many graphs, however, we are able to obtain much stronger inapproximability results. Specifically, we give a reduction from LabelCover; this is one of the six canonical inapproximable problems described by Arora and Lund [5]. We will need its maximization version.

`LabelCover-Max Problem:`
Input: A d-regular bipartite graph $G = (A \cup B, E)$, an integer N and a partial function $\Pi_e : [N] \to [N]$ for each $e \in E$.
Objective: Label $\ell(v)$ each vertex $v \in G$ to maximize the number of covered edges. [An edge $e = (u, v)$ is *covered* if and only if $\Pi_e(\ell(u)) = \ell(v)$.]

The following *gap-preserving reduction* for LabelCover-Max is known, and follows from the PCP Theorem [6, 7] and Raz's Parallel Repetition Theorem [35].

Theorem 12. *For any $\varepsilon > 0$, an instance of* Sat *can be transformed in quasi-polynomial time into a d-regular instance of* LabelCover-Max *such that*

- *if the original instance of* Sat *is satisfiable then the instance of* LabelCover-Max *has a solution of value* 1*,*
- *if the original instance of* Sat *is not satisfiable then all solutions to the instance of* LabelCover-Max *has value at most $2^{-\log^{1-\varepsilon} n}$.* □

[The value *of a solution is the ratio of edges covered compared to $|E|$, the number of edges.]*

Consequently, the inapproximability bounds for LabelCover-Max are very large.

Corollary 1. LabelCover-Max *is not approximable to within a factor $2^{\log^{1-\varepsilon} n}$, for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.* □

We show that an approximation algorithm for Densest Simultaneous Subgraph leads to an approximation algorithm for LabelCover-Max with the following guarantees.

Theorem 13. *If* Densest Simultaneous Subgraph *is α-approximable then* LabelCover-Max *is $72\alpha^2$-approximable.*

Proof. Take an instance (G, N, Π) of LabelCover-Max. We build an instance of Densest Simultaneous Subgraph on a collection of graphs $\mathcal{H}$ as follows. There is one graph $H_e \in \mathcal{H}$ for each edge of G. Each graph contains the same vertex set: there is a vertex (u, i) in H_e for each pair $u \in V(G), i \in [N]$. The edge sets of the graphs, however, are disjoint. For an edge $e = (u, v) \in G$, there is an edge in H_e between (u, i) and (v, j) if and only if $\Pi_{(u,v)}(i) = j$. Thus, if $|A| = q = |B|$ then $\mathcal{H}$ contains qd graphs and each such graph H_e is a bipartite graph with $2qN$ vertices.

We now add an extra graph and extra vertices so that later in the proof, we are guaranteed solutions of size s have density at most $1/s$. We add two isolated vertices $\hat{u}, \hat{v}$ to the vertex set (of each graph in $\mathcal{H}$) and add a new graph $\hat{H}$ containing only one edge $(\hat{u}, \hat{v})$.

Note that we may partition the vertices of $H - \{\hat{u}, \hat{v}\}$ into sets $\{W_1, W_2, \ldots, W_{2q}\}$ where $W_v = \{(v, i) : i \in [N]\}$. Clearly any optimal solution S^* to the instance of Densest Simultaneous Subgraph must use at least one vertex from each of these sets. Otherwise there is at least one (in fact, at least d) graph H_e within which no edges are induced and, thus, the minimum density is zero. Furthermore, $\hat{u}$ and $\hat{v}$ are both in S^* or the density of S^* in $\hat{H}$ is zero. So the optimal solution S^* has cardinality at least $2q + 2$.

Observe that if an edge $((u, i), (v, j))$ is induced by S^* in $H_{u,v}$ then the corresponding edge in LabelCover-Max is covered, provided we set $\ell(u) = i$ and $\ell(v) = j$. For our hardness result, we may assume that all the edges in the LabelCover-Max instance can be covered. Thus, we may assume that the solution S^* induces a density D^* of at least $\frac{1}{2q+2}$ in each graph.

By our hypothesis, we can approximate D^* to within an α factor. Thus we obtain a solution S with density at least $\frac{1}{2\alpha q + 2\alpha}$. By the construction of $\hat{H}$, S has size at most $2\alpha q + 2\alpha < 3\alpha q$. We now use S to build a solution to the instance of LabelCover-Max.

Let $X = \{v \in G : |W_v \cap S| > 6\alpha\}$. Now $|X| < \frac{1}{2}q$, otherwise, $|S| > \frac{1}{2}q \cdot 6\alpha = 3\alpha q$. Furthermore, as G is d-regular the vertices in X cover at most half of the dq edges of G; thus the vertices in $\bar{X} = (A \cup B) \setminus X$ cover at least half of the edges.

Take the set $S' = \{(v, i) \in S : v \in \bar{X}\}$. From S', we build a random labelling by selecting a random node (v, i) in $S' \cap W_v$, for each vertex $v \in \bar{X}$. We then set $\ell(v) = i$. Because $|W_v \cap S| \leq 6\alpha$ for all $v \in \bar{X}$, any edge induced by $\bar{X}$ is covered by this labelling with probability at least $\frac{1}{36\alpha^2}$. Thus, this labelling covers at least $\frac{1}{36\alpha^2} \cdot \frac{1}{2} dq = \frac{1}{72\alpha^2} \cdot dq$ edges, as desired. □

By derandomizing this reduction, we obtain the following hardness result.

Theorem 3. Densest Simultaneous Subgraph *is not approximable within* $2^{\log^{1-\varepsilon} n}$ *for any* $\varepsilon > 0$*, unless* $NP \subseteq DTIME(n^{\text{polylog}n})$.

Proof. So we need to alter the proof of Theorem 13 so that random choices are not used to recover the solution. To do so, instead of sampling from the approximate solution S, we will essentially compute the expected value of picking each vertex (u, i) for each i, choosing the vertex maximizing this expectation

and repeat this process for each u (but conditioning on choices already made in our computation).

Formally, we let $v_1, v_2, \ldots, v_{2q}$ be the vertices of G (ordered arbitrarily). Recall that our proof of Theorem 13 selects a label L_i for v_i uniformly at random amongst all labels ℓ with $(v_i, \ell) \in S' \cap W_{v_i}$ (it does this for all i from 1 to $2q$). This defines $2q$ independent variables $L_1, \ldots, L_{2q}$. We now see how to deterministically assign values to $L_1, \ldots, L_{2q}$ so that the number of edges covered by this assignment is at least the expected number of edges covered by assigning values randomly. Let Covered$(\ell_1, \ldots, \ell_{2q})$ denote the number of covered edges given labels ℓ_i to v_i.

For each i from 1 to $2q$, proceed as follows. For each $(v_i, \ell) \in S' \cap W_{v_i}$ compute

$$e(i, \ell) = E[\text{Covered}(L_1, \ldots, L_{2q}) | L_j = \ell_j \text{ for } j = 1, 2, \ldots, i-1 \text{ and } L_i = \ell]$$

and pick ℓ_i so that $e(i, \ell_i) = \max_\ell e(i, \ell)$. It is easy to see that this algorithm produces a solution at least $E[\text{Covered}(L_1, \ldots, L_{2q})]$ since for each i, by our choice of ℓ_i,

$$\begin{aligned} &E[\text{Covered}(L_1, \ldots, L_{2q}) | L_j = \ell_j \text{ for } j = 1, \ldots, i] \geq \\ &E[\text{Covered}(L_1, \ldots, L_{2q}) | L_j = \ell_j \text{ for } j = 1, \ldots, i-1]. \end{aligned}$$

Thus, by induction,

$$\begin{aligned} \text{Covered}(\ell_1, \ldots, \ell_{2q}) &= E[\text{Covered}(L_1, \ldots, L_{2q}) | L_j = \ell_j \text{ for } j = 1, \ldots, 2q] \\ &\geq E[\text{Covered}(L_1, \ldots, L_{2q})] \end{aligned}$$

as required. □

Furthermore, these results extend to the robust variations discussed in the Introduction. This follows via standard techniques, so we defer the corresponding proof (along with a formal definition of *robustness*) to the Appendix.

4.3 Non-zero density

To conclude our discussion on the density measure, we remark that clearly the most basic structure we can possibly search for is a *single* edge. But an induced subgraph that contains a single edge has non-zero density and vice versa. This leads to the following cluster problem.

> `Non-Zero Density Problem`
> *Input:* Graphs $G_i = (V, E_i)$, where $1 \leq i \leq t$.
> *Objective:* A minimum cardinality set $S \subseteq V$ with non-zero density in each $G_i[S]$.

It can then be seen that our hardness proof also applies to Non-Zero Density. Thus this very basic problem is extremely hard to approximate!

Corollary 2. Non-Zero Density *is not approximable within* $2^{\frac{1}{3}\log^{1-\varepsilon} n}$ *for any* $\varepsilon > 0$*, unless* $NP \subseteq DTIME(n^{\text{polylog}n})$. □

Of course, if it is very hard to search for a single edge then it is not surprising that quantitative guarantees for practical simultaneous clustering problems are rare.

5 The Connectivity Measure

Now let's consider the simultaneous cluster problem using our second quality measure, namely graph connectivity. Our vertex set is partitioned into two: a subset $T \subseteq V$ of *terminals* and a set $V \setminus T$ of *steiner* vertices. A cluster $S \subseteq V$ is then considered good if every pair of terminals in S is simultaneously connected (or k-connected) with respect to each graph. As described in Section 2, notions of terminal connectivity have been applied to expand our current knowledge of genes involved in certain biological processes by treating the known genes in these processes as terminal nodes. Some applications have also treated all nodes (genes) as terminals to detect clusters of functionally coherent genes from biological networks where connectivity implies functional similarity.

Once the desired connectivity k is specified, there are two natural optimization criteria. The first is a maximization criterion, we may desire a good cluster that contains as many terminals as possible. Since connectivity is a monotonic property with regards to the steiner nodes, it can never hurt to add additional steiner vertices to such a cluster. Consequently, this maximization criterion is likely to produce very large clusters. Therefore, the second natural criterion is to minimize the cardinality of a good cluster.

In this section we present both good news and bad. The simultaneous cluster problem is tractable in polynomial time with respect to the maximization measure, but is very hard to approximate with the minimization measure.

5.1 Terminal Maximization

Consider then our maximization problem.

`Maximum Simultaneous` k`-Connected Steiner Cluster`
Input. Graphs $G_i = (V, E_i)$ for $1 \leq i \leq t$ and a set $T \subseteq V$ of terminals.
Objective. Find a cluster $S \subseteq V$ maximizing $|S \cap T|$, such that the terminals in S are k-connected in each induced subgraph $G_i[S]$.

For a fixed connectivity requirement k, this problem is polynomial time solvable. We remark that this is the case for both vertex-connectivity and edge-connectivity requirements. We show how to solve the vertex-connectivity version using the following recursive approach (the approach for edge-connectivity is similar). We are given a collection $\mathcal{G}$ of graphs with vertex set V and terminal set T. If every pair of terminals are k-connected in every graph $G_i \in \mathcal{G}$ then we simply output the cluster $S = V$.

If not, by Menger's Theorem, we can find terminals t_1 and t_2 that are separated by a vertex-cut W (with cardinality less than k) in some graph G_j. So, assume $t_1 \in V_1$ and $t_2 \in V_2$, where $V_1 \cup V_2 = V \setminus W$, and let $T_1 = T \cap (V_1 \cup W)$ and $T_2 = T \cap (V_2 \cup W)$. Observe that T_1 and T_2 need not be disjoint but $|T_1 \cap T_2|$ must be less than k.

We now recurse on the subproblems $\mathcal{G}^1$ and $\mathcal{G}^2$. Here $\mathcal{G}^1$ contains graphs $G_i^1 = G_i[V - (T - T_1)]$, for all $1 \leq i \leq t$, and has terminal set T_1. Similarly, $\mathcal{G}^2$ contains the graphs $G_i^2 = G_i[V - (T - T_2)]$ and has terminal set T_2. Note that each subproblem contains all the steiner nodes.

Finally, when the algorithm terminates on every subproblem we simply output the best cluster obtained amongst all the subproblems. Let's see that this algorithm gives a polynomial time algorithm.

Theorem 4. *For a fixed connectivity requirement k, there is a polynomial time algorithm for* Maximum Simultaneous k-Connected Steiner Cluster.

Proof. First we need to show that the algorithm gives an optimal solution. The terminals in the cluster output by each subproblem are k-connected, otherwise the algorithm would have found a new vertex-cut to recurse on. So the clusters are feasible solutions. Suppose the optimal solution set of terminals T^* is not output. Then consider the first time at which two terminals $t_1, t_2 \in T^*$ are separated by the algorithm. At this point, let the subproblem consist of the terminals $\hat{T}$ and all the steiner nodes, and let W be the vertex-cut separating t_1 and t_2. But $T^* \subseteq \hat{T}$. So W must also separate t_1 and t_2 in the graph induced by T^* and all the steiner nodes. This is a contradiction as, by definition, the cluster consisting of T^* and all the steiner nodes k-connects all the terminals in T^*.

Second we need to show how to implement the algorithm in polynomial time. We do this in two stages. In Stage I, we only run the method until each subproblem contains at most $k+1$ terminals. In Stage II, we solve each of these subproblems by brute force, that is, for every subset of the terminals in the subproblem, we check if those terminals are k-connected using all the steiner nodes, in every graph.

To analyse the running time for Stage I, we show that at most $|T| - k$ subproblems can be examined in this stage. To search for the vertex-cut, we only need to run $k|T|$ max-flow algorithms to check all the terminal pairs. Each flow algorithm takes time $O(km)$ as we can stop if the flow between a pair exceeds k. We must do this on each of the t graphs, so this search takes time $O(|T| \cdot k^2 mt)$. If there are at most $|T| - k$ subproblems in Stage I then the total run time for the stage is $O(|T|^2 \cdot k^2 mt)$.

We show by induction that the number of subproblems in Stage I is indeed at most $|T| - k$. For the base case, if $|T| = k + 1$ then we stop immediately. Consequently, there is only one subproblem to consider. So consider the case where $|T| > k + 1$. Suppose $|T|$ is split into T_1 and T_2 by the vertex-cut. By induction, the number of subproblems considered for T_i is at most $|T_i| - k$, for $i = \{1, 2\}$. Moreover, we know that $|T_1 \cap T_2| \leq k - 1$. Thus the total number of

subproblems considered for T is at most

$$1 + |T_1| - k + |T_2| - k \ \leq \ 1 + (|T| + k - 1) - k - k \ = \ |T| - k$$

as desired.

Now consider the running time for Stage II. When $|T| \leq k + 1$ we can simply use brute force. For every subset of the terminals, check whether those terminals are k-connected using all the steiner nodes, in every graph. By the method above this takes time $O(|T| \cdot k^2 mt) = O(k^3 mt)$. There are 2^{k+1} subsets and $|T| - k$ subproblems so the run time Stage II is at most $O(2^k \cdot |T| \cdot k^3 mt)$.

Thus the total run time of the algorithm is polynomial for any fixed k. □

□

5.2 Cluster Minimization

On the other hand, if we wish to minimize the number of vertices, the problem becomes hard again, even for the simplest connectivity requirement $k = 1$. Interestingly, it remains very difficult even in the two extremes cases where (i) there are only two terminals, and (ii) every vertex is a terminal.

Let's begin with the case of exactly two terminals, say $T = \{s, t\}$. Then our minimization problem is:

`Simultaneous s-t Path`:
Input: Graphs $G_i = (V, E_i)$ for $1 \leq i \leq t$, and special vertices s and t.
Objective: A minimum cardinality cluster $S \subseteq V$ inducing an $s-t$ path in each $G_i[S]$.

Theorem 14. *If* Simultaneous s-t Path *is α-approximable then* LabelCover-Max *is $\frac{1}{72\alpha^2}$-approximable.*

Proof. Take an instance (G, N, Π) of LabelCover-Max where G has bipartition (A_G, B_G). We build an instance of Simultaneous s-t Path by first building an instance $\mathcal{H}$ of Densest Simultaneous Subgraph as in the proof of Theorem 13 but without the vertices $\hat{u}$ and $\hat{v}$. Note that each graph $H_e \in \mathcal{H}$ is bipartite with bipartitions $A_H = \{(u, i) | u \in A_G, i \in [N]\}$ and $B_H = \{(v, i) | v \in B_G, i \in [N]\}$. We now build a graph F_e from each graph H_e by

- adding a source s with edges from s to every vertex of A_H, and
- adding a sink t with edges from every vertex of B_H to t.

Let $\mathcal{F}$ be the collection of graphs F_e built from each graph $H_e \in \mathcal{H}$. Now, a solution S to our instance $\mathcal{F}$ of Simultaneous s-t Path with $\{\hat{u}, \hat{v}\}$ added and $\{s, t\}$ removed is a solution to the instance $\mathcal{H}$ of Densest Simultaneous Subgraph of the same value. □

Now consider the other extreme, where all vertices are terminals. So in any solution cluster, every pair of vertices in that cluster must be connected in each induced graph $G_i[S]$.

Minimum Simultaneous Connected Steiner Cluster
Input. Graphs $G_i = (V, E_i)$ for $1 \leq i \leq t$.
Objective. A minimum cardinality cluster $S \subseteq V$ (of size at least 2) such that every pair of vertices in S is connected in each induced graph $G_i[S]$.

Theorem 15. *If* Minimum Simultaneous Connected Steiner Cluster *is α-approximable then* LabelCover-Max *is $\frac{1}{72\alpha^2}$-approximable.*

Proof. Take an instance (G, N, Π) of LabelCover-Max where G has bipartition (A, B). We build an instance of Minimum Simultaneous Connected Steiner Cluster by first building an instance $\mathcal{H}$ of Densest Simultaneous Subgraph as in the proof of Theorem 13 but without the vertices $\hat{u}$ and $\hat{v}$. Note that each graph $H_e \in \mathcal{H}$ is bipartite with bipartitions $A_H = \{(u, i) | u \in A, i \in [N]\}, B_H = \{(v, i) | v \in B, i \in [N]\}$.

We build a graph F_e from each graph H_e by

- adding a vertex s with edges between s and every vertex of A_H, and
- adding a vertex t with edges between t and every vertex of B_H.

Let $\mathcal{F}$ be the collection of (a) graphs F_e built from each graph $H_e \in \mathcal{H}$, and (b) further graphs F_s, F_t, and F_v (for each $v \in A_G \cup B_G$) built over the same set of nodes. F_s has edges between s and every other vertex, and F_t has edges between t and every other vertex. For each $v \in A_G \cup B_G$, F_v has edges between every vertex of (v, i) (for all values of the label i i.e., for all $i \in [N]$) and every other vertex. No other edges are present in these graphs.

Now any solution must contain s, t, a vertex (u, i) for each vertex $u \in A_G$ and a vertex (v, j) for each vertex $v \in B_G$. Otherwise, the subgraph is not connected in F_s, F_t, F_u for some $u \in A_G$ or F_v for some $v \in B_G$.

Again, any solution S to our instance $\mathcal{F}$ of Minimum Simultaneous Connected Steiner Cluster with $\{\hat{u}, \hat{v}\}$ added and $\{s, t\}$ removed is a solution to the instance $\mathcal{H}$ of Densest Simultaneous Subgraph of the same value. □

Thus we obtain the hardness results of Theorem 5 and Theorem 6.

Theorem 5. Simultaneous s-t Path *is not approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.* □

Theorem 6. Minimum Simultaneous Connected Steiner Cluster *is not approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.* □

Similar hardness results also extend to the problem of finding an approximate solution that is required to satisfy connectivity constraint in only a c fraction of the input graphs. Again, these robustness results are deferred to the Appendix.

5.3 Lower Bounds for a Fixed Number of Graphs

Polynomial lower bounds can be obtained in terms of the number g of input graphs. Clearly for Simultaneous s-t Path, we can also obtain a g-approximation given g input graphs by simply taking the union of all solutions in each individual graph.

In this section, we show for any $\varepsilon > 0$ both Simultaneous s-t Path and Minimum Simultaneous Connected Steiner Cluster are $g^{1/2-\varepsilon}$-inapproximable unless $NP = ZPP$. We use a similar approach to other k^{ε} complexity results for problems with a fixed parameter k [13, 15]. Again, by the PCP theorem and Raz's parallel repetition theorem we have:

Theorem 16. *[35, 6] There exists a constant $\gamma > 0$ (independent of ℓ) such that the* LabelCover-Max *problem obtained from instances of* Max-3Sat(5) *with ℓ repetitions cannot be approximated within a factor of $2^{\gamma^{\ell}}$. (For constant ℓ, this holds if $P \neq NP$. For $\ell = \text{polylog}(n)$, this holds under the assumption $NP \not\subseteq DTIME(\text{polylog}(n))$.)* □

Here, Max-3Sat(5) simply refers to Max-Sat instances where there are 3 variables in each clause and every variable appears in 5 clauses. Since instances of LabelCover-Max obtained from Max-3Sat(5) with ℓ-repetitions are $(3^{\ell}, 5^{\ell})$-regular, we obtain the following corollary.

Corollary 3. *There exists a constant $\gamma' > 0$ (independent of ℓ) such that the d-regular* LabelCover-Max *problem cannot be approximated within a factor of $d^{\gamma'}$. (For constant d, this holds if $P \neq NP$. For $d = n^{\alpha}$, this holds under the assumption $NP \not\subseteq DTIME(\text{polylog}(n))$.)* □

Thus, it suffices to build a $g = d^{\beta}$ (for some constant $\beta > 0$) instance of our problem of interest from a d-regular instance of LabelCover-Max to obtain $g^{\varepsilon} = d^{\gamma'/\beta}$ inapproximability for our problem.

To improve this to an $g^{1/2-\varepsilon}$-inapproximability result, we use Goldreich and Sudan's [20] random sampling technique that reduces the degree of the instance of LabelCover-Max needed. This allows us to improve the bound from Corollary 3 to $g^{1/2-\varepsilon}$ under the assumption that $NP \neq ZPP$.

Theorem 17. *[14, 28] For any $\varepsilon > 0$, it is hard to approximate instances of* LabelCover-Max *where vertices have degrees between $d/4$ and d within a factor of $d^{1/2-\varepsilon}$, unless $NP = ZPP$.*

Thus, it suffices to build a $g = d^{\beta}$ (for a fixed $\beta > 0$) instance of our problem of interest from a d-regular instance of LabelCover-Max to obtain $d^{1/2-\varepsilon} = g^{1/(2\beta)-\varepsilon'}$-inapproximability for our problem.

We are now ready to prove Theorems 7 and 8. In our case, the number of graphs is linear in the degree of the input graph to LabelCover-Max (i.e., $\beta = 1$).

Theorem 7. Simultaneous s-t Path *is not $g^{1/2-\varepsilon}$-approximable for any $\varepsilon > 0$ where g is the number of graphs unless $NP = ZPP$.*

Proof. We reduce our problem from LabelCover-Max and construct an instance of Simultaneous s-t Path whose number of graphs is linear in the degree of the graph in LabelCover-Max. Take an instance (G, N, Π) of LabelCover-Max where G has bipartition (A, B). We build an instance of Simultaneous s-t Path by first building an instance $\mathcal{H}$ of Densest Simultaneous Subgraph as in the proof of Theorem 13 but without the vertices $\hat{u}$ and $\hat{v}$ (note although G is not regular, this construction is still well defined). We then build a new instance $\mathcal{F}$ of Simultaneous s-t Path using d graphs from $\mathcal{H}$.

Note that each graph $H_e \in \mathcal{H}$ is bipartite with bipartitions $A_H = \{(u, i) | u \in A, i \in [N]\}, B_H = \{(v, i) | v \in B, i \in [N]\}$. Since G is bipartite and of maximum degree d, we can partition its edges into d matchings $M_1, \ldots, M_d$. Let $u_{i,1}v_{i,1}, \ldots, u_{i,q}v_{i,q}$ be the edges of M_i. We construct F_i by taking the union of all (edges in) $H_e, e \in M_i$ and adding a source s and a sink t and the following edges C_i, S_i and T_i.

$$\begin{aligned} C_i &= \bigcup_{j=1}^{q-1} \bigcup_{\ell_1, \ell_2 \in [N]} (v_{i,j}, \ell_1)(u_{i,j+1}, \ell_2) \\ S_i &= \bigcup_{\ell \in [N]} s(u_{i,1}, \ell) \\ T_i &= \bigcup_{\ell \in [N]} (v_{i,q}, \ell)t \end{aligned}$$

Note that every st-path in F_i uses at least one edges from each $H_e, e \in M_i$ (since each $E(H_e)$ is an st-cut in F_i). Furthermore, we can obtain an st-path by choosing (any) one edge from each $H_e, e \in M_i$ and the appropriate edges in C_i, S_i and T_i.

Therefore, there is an st-path in all F_i if and only if S induces an edge in each $H_e \in \mathcal{H}$. The result now follows from Theorem 17. □ □

Theorem 8. Minimum Simultaneous Connected Steiner Cluster *is not $g^{1/2-\varepsilon}$-approximable for any $\varepsilon > 0$ where g is the number of graphs unless $NP = ZPP$.*

Proof. Take the instance of Simultaneous s-t Path from Theorem 7 above and add a graph F_s which is a star centered at s and add a graph F_t which is a star centered at t.

Now, every solution S to our instance of Minimum Simultaneous Connected Steiner Cluster contains s and t (or one of $F_s[S]$ or $F_t[S]$ is disconnected). Therefore, all solutions to our instance of Minimum Simultaneous Connected Steiner Cluster is also a solution to the original instance of Simultaneous s-t Path.

To complete the proof, we show that any feasible solution S to Simultaneous s-t Path corresponds to a feasible solution $S^* = S \cup \{s, t\}$ to Minimum Simultaneous Connected Steiner Cluster. $F_s[S^*]$ is connected since

$s \in S^*$ and every other vertex has an edge to s. $F_t[S^*]$ is connected since $t \in S^*$ and every other vertex has an edge to t.

Since S is a solution to SIMULTANEOUS S-T PATH, for any i, there is an st path
$P = \{s, (u_{i,1}, \ell_1), (v_{i,1}, \ell_1), (u_{i,2}, \ell_2), (v_{i,2}, \ell_2), \ldots, (u_{i,q}, \ell_q), (v_{i,q}, \ell_q), t\}$ in $F_i[S]$ and since M_i is a perfect matching, this path P contains a vertex (x, j) for each $x \in A \cup B$. We now show that every other vertex of F_i has an edge to P, thus proving $F_i[S^*]$ is connected.

Indeed, for any vertex $(u_{i,j}, k) \in A_H$ with $u_i \in A$ and $u_{i,j}v_{i,j} \in M_i$, either

- $j = 1$ and $s(u_{i,j}, k) \in S_i$ so $(u_{i,j}, k)$ is adjacent to P, or
- $j > 1$ and $(v_{i,j-1}, \ell_{j-1})(u_{i,j}, k) \in C_i$ so again $(u_{i,j}, k)$ is adjacent to P.

We use a symmetric proof for any vertex $(v_{i,j}, k) \in B_H$ with $v_i \in B$ and $u_{i,j}v_{i,j} \in M$. Either

- $j = q$ and $(v_{i,j}, k)t \in S_i$ so $(v_{i,j}, k)$ is adjacent to P, or
- $j < q$ and $(v_{i,j}, k)(u_{i,j+1}, \ell_{j+1}) \in C_i$ so again $(v_{i,j}, k)$ is adjacent to P.

Thus, all the $F_i[S^*]$ are connected and the theorem follows by Theorem 17. □

6 Conclusion and Directions

We have presented algorithmic and complexity results for the problem of finding clusters supported by multiple graphs, where each graph represents distinct set of similarity relationships (edges) over the same set of objects (nodes). While we obtain tractable algorithms for certain measures of cluster quality, we show that the problem is typically hard to approximate even when we relax many of the requirements, such as relaxing the problem from many graphs to just two graphs for the density measure, connectivity among many terminals to just two terminals, or quality constraints of a solution to be met in only a fraction of the input graphs.

The implications of our results are two-fold. First, our results explain why guarantees on the clustering quality or running time have been elusive in the vast amount of previous empirical and heuristic works on simultaneous clustering of datasets arising in scientific and commercial domains. Second, our work suggests alternate problem abstractions may also be suitable for quantitative study.

For example, we could consider a new model where the input graphs have correlated edge weights, since the hardness of most problems we consider stem from allowing the graphs to have arbitrary edge weights. Assuming the similarity function of different input graphs to be correlated for *all* edges is not realistic though, especially in the biological sciences where the datasets are very noisy, incomplete and heterogeneous (due to factors like the different types of cellular responses each input network captures, highly incomplete nature of

networks assembled from small-scale biological studies, bias or batch effect or technology-dependent artifacts affecting networks inferred from large-scale biological studies, etc. [24]). However, we could reasonably assume that those edges present in the optimal solution or subgraph have correlated edge weights across the input graphs. Introducing this assumption may make the problem tractable by allowing us to exclude edges that are not correlated in the graphs before searching for the optimal solution.

Comparative analysis of clustering structures between multiple networks is another pressing problem in data integration. Given a separation of the input networks into two classes A and B (say diseased vs. healthy), can we find subgraphs that cluster well in most of the class A networks and poorly in most of the class B networks?

Acknowledgements

We are very grateful to Bundit Laekhanukit for useful comments and suggestions, and to Samir Khuller for introducing us to the min-max k-centre problem.

References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max and min-max regret assignment problems. *Operations Research Letters*, 33(6):634–640, 2005. doi:10.1016/j.orl.2004.12.002.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, 197(2):427–438, 2009. doi:10.1016/j.ejor.2008.09.012.

[3] R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In *Algorithms and Models for the Web-Graph*, pages 25–37. Springer, 2009. doi:10.1007/978-3-540-95995-3_3.

[4] B. Anthony, V. Goyal, A. Gupta, and V. Nagarajan. A plant location guide for the unsure: Approximation algorithms for min-max location problems. *Mathematics of Operations Research*, 35:79–101, 2010. doi:10.1287/moor.1090.0428.

[5] S. Arora and C. Lund. Hardness of approximations. In *Approximation Algorithms for NP-Hard Problems*, pages 399–446. PWS Publishing Company, 1996.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

[7] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

[8] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, et al. NCBI GEO: archive for functional genomics data sets–10 years on. *Nucleic acids research*, 39(suppl 1):D1005–D1010, 2011. doi:10.1093/nar/gkq1184.

[9] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 201–210. ACM, 2010. doi:10.1145/1806689.1806719.

[10] A. Bhaskara, M. Charikar, A. Vijayaraghavan, V. Guruswami, and Y. Zhou. Polynomial integrality gaps for strong SDP relaxations of densest k-subgraph. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 388–405. SIAM, 2012.

[11] R. Bhatia, S. Guha, S. Khuller, and Y. J. Sussmann. Facility location with dynamic distance functions. *Journal of combinatorial optimization*, 2(3):199–217, 1998.

[12] J. R. Birge. State-of-the-art-survey – stochastic programming: Computation and applications. *INFORMS journal on computing*, 9(2):111–133, 1997. doi:10.1287/ijoc.9.2.111.

[13] T. Chakraborty, J. Chuzhoy, and S. Khanna. Network design for vertex connectivity. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 167–176. ACM, 2008. doi:10.1145/1374376.1374403.

[14] M. Dinitz, G. Kortsarz, and R. Raz. Label cover instances with large girth and the hardness of approximating basic k-spanner. In *Automata, Languages, and Programming*, pages 290–301. Springer, 2012. doi:10.1007/978-3-642-31594-7_25.

[15] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 595–601. ACM, 2003. doi:10.1137/S0097539704443057.

[16] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, 1998. doi:10.1073/pnas.95.25.14863.

[17] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001. doi:10.1007/s004530010050.

[18] U. Feige and M. Seltser. On the densest k-subgraph problem. Technical report, The Weizmann Institute, Rehovot, 1997.

[19] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989. doi:10.1137/0218003.

[20] O. Goldreich and M. Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM (JACM)*, 53(4):558–655, 2006. doi:10.1145/1162349.1162351.

[21] K. C. Gunsalus, H. Ge, A. J. Schetter, D. S. Goldberg, J.-D. J. Han, T. Hao, G. F. Berriz, N. Bertin, J. Huang, L.-S. Chuang, N. Li, R. Mani, A. A. Hyman, B. Sonnichsen, C. J. Echeverri, F. P. Roth, M. Vidal, and F. Piano. Predictive models of molecular machines involved in caenorhabditis elegans early embryogenesis. *Nature*, 436(7052):861–865, 2005. doi:10.1038/nature03876.

[22] C. Huttenhower, M. Hibbs, C. Myers, and O. G. Troyanskaya. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897, 2006. doi:10.1093/bioinformatics/btl492.

[23] C. Huttenhower and O. Hofmann. A quick guide to large-scale genomic data mining. *PLoS computational biology*, 6(5):e1000779, 2010. doi:10.1371/journal.pcbi.1000779.

[24] D. Hwang, A. G. Rust, S. Ramsey, J. J. Smith, D. M. Leslie, A. D. Weston, P. de Atauri, J. D. Aitchison, L. Hood, A. F. Siegel, et al. A data integration methodology for systems biology. *PNAS*, 102(48), 2005. doi:10.1073/pnas.0508647102.

[25] D. J. Ketchen and C. L. Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458, 1996. doi:10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G.

[26] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Kluwer Academic Pub, 1997.

[27] M. Koyutürk, A. Grama, and W. Szpankowski. Pairwise local alignment of protein interaction networks guided by models of evolution. In *RECOMB*, pages 48–65, 2005. doi:10.1007/11415770_4.

[28] B. Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. 2012. arXiv:1212.0752.

[29] E. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, 1976.

[30] M. R. Mehan, J. Nunez-Iglesias, M. Kalakrishnan, M. S. Waterman, and X. J. Zhou. An integrative network approach to map the transcriptome to the phenome. *Journal of Computational Biology*, 16(8):1023–1034, 2009. PMID: 19630539. doi:10.1089/cmb.2009.0037.

[31] M. Narayanan and R. M. Karp. Comparing protein interaction networks via a graph match-and-split algorithm. *Journal of Computational Biology*, 14(7):892–907, 2007. doi:10.1089/cmb.2007.0025.

[32] M. Narayanan, A. Vetta, E. E. Schadt, and J. Zhu. Simultaneous clustering of multiple gene expression and physical interaction datasets. *PLoS Comput Biol*, 6(4):e1000742, 04 2010. doi:10.1371/journal.pcbi.1000742.

[33] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007. doi:10.1038/nature05670.

[34] J. Picard and M. Queyranne. A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. *Networks*, 12(2):141–159, 1982. doi:10.1002/net.3230120206.

[35] R. Raz. A parallel repetition theorem. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 447–456. ACM, 1995. doi:10.1137/S0097539795280895.

[36] R. Sharan, T. Ideker, B. P. Kelley, R. Shamir, and R. M. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. In *RECOMB*, pages 282–289, 2004. doi:10.1145/974614.974652.

[37] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1974–1979, 2005. doi:10.1073/pnas.0409522102.

[38] L. V. Snyder and M. S. Daskin. Stochastic p -robust location problems. *IIE Transactions*, 38:971–985, 2006.

[39] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255, 2003. doi:10.1126/science.1087447.

[40] C. Swamy. *Approximation Algorithms for Clustering Problems (Chapter 4)*. PhD thesis, Dept. of Computer Science, Cornell University, 2004.

[41] C. Swamy. Risk-averse stochastic optimization: Probabilistically-constrained models and algorithms for black-box distributions. In *SODA*, pages 1627–1646, 2011.

[42] C. Swamy and D. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37, 2006. doi:10.1145/1122480.1122493.

[43] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology*, 1(1):8, 2007.

[44] L. J. van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002. doi:10.1038/415530a.

[45] S. Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[46] N. Yosef, L. Ungar, E. Zalckvar, A. Kimchi, M. Kupiec, E. Ruppin, and R. Sharan. Toward accurate reconstruction of functional protein networks. *Molecular systems biology*, 5(1), 2009. doi:10.1038/msb.2009.3.

[47] G. Yu and J. Yang. On the robust shortest path problem. *Computers and Operations Research*, 25(6):457–468, 1998. doi:10.1016/S0305-0548(97)00085-3.

[48] B. Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1):Article 17, 2005. doi:10.2202/1544-6115.1128.

[49] J. Zhu, B. Zhang, E. N. Smith, B. Drees, R. B. Brem, L. Kruglyak, R. E. Bumgarner, and E. E. Schadt. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nat Genet*, 40(7):854–861, 2008. doi:10.1038/ng.167.

Appendix: Robustness Optimization

Even if we permit the approximate solution to satisfy the quality (density or connectivity) constraint in only a c fraction of the input graphs, the problems discussed in Sections 4 and 5 remain very hard. In this section, we formalize this concept of *robustness* for each problem and prove that they remain difficult.

The Density Measure

We now formalize the notion of robustness for Densest Simultaneous Subgraph. We say that a solution S to an instance of Densest Simultaneous Subgraph on g graphs $G_1, \ldots, G_g$ is *c-relaxed α-approximate* if there exists cg graphs $G_{i_1}, \ldots, G_{i_{cg}}$ for which $\text{den}_{G_{i_j}}(S) \geq D^*/\alpha$ where D^* is the value of the optimal (unrelaxed) solution to this instance of Densest Simultaneous Subgraph. Furthermore, if we can find a c-relaxed α-approximate solution to instances of Densest Simultaneous Subgraph in polynomial time, we say that Densest Simultaneous Subgraph is *c-relaxed α-approximable* (or *c-relaxed approximable within α*).

We now prove this "robust" variant of Theorem 13.

Theorem 9. Densest Simultaneous Subgraph *is not c-relaxed approximable within*
$2^{\log^{1-\varepsilon} n}$ *for any $\varepsilon > 0$ and constant $c > \frac{2}{3}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

Proof. Take an instance (G, N, Π) of LabelCover-Max and build an instance $\mathcal{H}$ of Densest Simultaneous Subgraph as in the proof of Theorem 13. But instead of adding a single $\hat{H}$ graph to the dq graphs H_e in $\mathcal{H}$, we now add $\hat{c} = \frac{(1-c)dq+1}{c}$ graphs $\hat{H}_1, \ldots, \hat{H}_{\hat{c}}$ each containing only one edge $(\hat{u}, \hat{v})$.

For our hardness result, we may assume that all the edges in the LabelCover-Max instance can be covered, and thus we may assume that any optimal solution S^* to the instance of Densest Simultaneous Subgraph induces a density D^* of at least $\frac{1}{2q+2}$ in each graph as in the proof of Theorem 13.

By our hypothesis, we can approximate D^* to within an $\alpha = 2^{\log^{1-\varepsilon} n}$ factor in a c fraction of the graphs. Thus we obtain a solution S with density at least $\frac{1}{2\alpha q+2\alpha}$ in at least a c fraction of the graphs. The total number of graphs is $\frac{dq+1}{c}$ and S contains both $\hat{u}$ and $\hat{v}$ (or it has density zero in more than $(1-c)$ fraction of all graphs). Similarly, S induces at least one edge in at least $c' = \frac{(dq+1)-\hat{c}}{dq} = (2 - \frac{1}{c} - \frac{1-c}{cdq})$ fraction of the dq graphs H_e in $\mathcal{H}$. Finally, by the construction of $\hat{H}$, S has size at most $2\alpha q + 2\alpha < 3\alpha q$. We now use S to build a solution to the instance of LabelCover-Max.

Let $X = \{v \in G : |W_v \cap S| > 6\alpha\}$, where $W_v = \{(v, i) : i \in [N]\}$. Now $|X| < \frac{1}{2}q$, otherwise, $|S| > \frac{1}{2}q \cdot 6\alpha = 3\alpha q$. Furthermore, as G is d-regular the vertices in X cover at most half of the dq edges of G; thus the vertices in $\bar{X} = (A \cup B) \setminus X$ cover at least $(c' - \frac{1}{2})$ fraction of the edges.

Take the set $S' = \{(v, i) \in S : v \in \bar{X}\}$. From S', we build a random labelling by selecting a random node (v, i) in $S' \cap W_v$, for each vertex $v \in \bar{X}$. We then

set $\ell(v) = i$. Because $|W_v \cap S| \leq 6\alpha$ for all $v \in \bar{X}$, any edge induced by $\bar{X}$ is covered by this labelling with probability at least $\frac{1}{36\alpha^2}$. Thus, this labelling covers at least

$$\frac{1}{36\alpha^2} \cdot \left(c' - \frac{1}{2}\right) \cdot dq = \frac{1}{36\alpha^2} \cdot \left(\frac{3}{2} - \frac{1}{c} - \frac{1-c}{cdq}\right) \cdot dq$$

edges. Since c is a constant (greater than $\frac{2}{3}$), for dq large, the term

$$\frac{1}{36\alpha^2} \cdot \left(\frac{3}{2} - \frac{1}{c} - \frac{1-c}{cdq}\right)$$

is bounded below by $\frac{1}{c''\alpha^2}$ for some c''. Since the approximation ration α for LabelCover-Max is an increasing function of $n = dq$, $\frac{1}{c''\alpha^2}$ is bounded below by $\frac{1}{\alpha^3}$ and the result follows. Similar to the proof of Theorem 3, this reduction can be derandomized in a straightforward manner. □

The Connectivity Measure

We now formalize the notion of robustness for Simultaneous s-t Path and Minimum Simultaneous Connected Steiner Cluster. We say that the problem Simultaneous s-t Path or Minimum Simultaneous Connected Steiner Cluster is c-relaxed approximable within a factor α if there exists a polynomial time algorithm which finds a solution S that induces connectivity between all pairs of terminals in at least a c fraction of the input graphs in the problem instance and whose size is within a factor α of the size of the optimal (unrelaxed) solution of the instance.

Theorem 10. Simultaneous s-t Path *is not c-relaxed approximable within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$ and constant $c > \frac{1}{2}$, unless $NP \subseteq DTIME(n^{\text{polylog}n})$.*

Proof. Take an instance (G, N, Π) of LabelCover-Max and build an instance $\mathcal{F}$ of Simultaneous s-t Path as in the proof of Theorem 14. Note that the F_e graphs in this collection $\mathcal{F}$ contain the vertices s, t and other vertices that partitions into sets $\{W_1, W_2, \ldots, W_{2q}\}$ with $W_v = \{(v, i) : v \in A \cup B, i \in [N]\}$. Clearly any optimal solution S^* to the instance of Simultaneous s-t Path uses at least one vertex from each of the sets W_v, otherwise some graph $F_e \in \mathcal{F}$ has no $s - t$ path induced by S^*.

Also observe that if an edge $((u, i), (v, j))$ is induced by S^* in $F_{u,v}$ then the corresponding edge in LabelCover-Max is covered, provided we set $\ell(u) = i$ and $\ell(v) = j$. For our hardness result, we may assume that all the edges in the input LabelCover-Max instance can be covered. Thus, we may assume the solution S^* is of size at most $2q + 2$.

Let $\alpha = 2^{\log^{1-\varepsilon} n}$. By our hypothesis, we can obtain a solution S of size at most $2\alpha q + 2\alpha < 3\alpha q$ that induces an $s - t$ path in at least a c fraction of the graphs F_e in $\mathcal{F}$. We now use S to build a solution to the instance of LabelCover-Max.

Let $X = \{v \in G : |W_v \cap S| > 6\alpha\}$. Now $|X| < \frac{1}{2}q$, otherwise, $|S| > \frac{1}{2}q \cdot 6\alpha = 3\alpha q$. Furthermore, as G is d-regular the vertices in X cover at most half of the dq edges of G; thus the vertices in $\bar{X} = (A \cup B) \setminus X$ cover at least $(c - \frac{1}{2})$ fraction of the edges.

Take the set $S' = \{(v, i) \in S : v \in \bar{X}\}$. From S', we build a random labelling by selecting a random node (v, i) in $S' \cap W_v$, for each vertex $v \in \bar{X}$. We then set $\ell(v) = i$. Because $|W_v \cap S| \leq 6\alpha$ for all $v \in \bar{X}$, any edge induced by $\bar{X}$ is covered by this labelling with probability at least $\frac{1}{36\alpha^2}$. Thus, this labelling covers at least

$$\frac{1}{36\alpha^2} \cdot (c - \frac{1}{2}) \cdot dq$$

edges, which is bounded below by $\frac{1}{\alpha^3}$ for $c > 1/2$ and large enough dq, as desired. Again, this process can be derandomized. □

Theorem 11. Minimum Simultaneous Connected Steiner Cluster *is not c-relaxed approximable within* $2^{\log^{1-\varepsilon} n}$ *for any* $\varepsilon > 0$ *and constant* $c > \frac{4}{5}$, *unless* $NP \subseteq DTIME(n^{\text{polylog}n})$.

Proof. Take an instance (G, N, Π) of LabelCover-Max where G has bipartition (A, B). We build an instance of Minimum Simultaneous Connected Steiner Cluster by first building an instance $\mathcal{H}$ of Densest Simultaneous Subgraph as in the proof of Theorem 13 but without the vertices $\hat{u}$ and $\hat{v}$. Note that each graph $H_e \in \mathcal{H}$ is bipartite with bipartitions $A_H = \{(u, i) | u \in A, i \in [N]\}, B_H = \{(v, i) | v \in B, i \in [N]\}$.

We build a graph F_e from each graph H_e by

- adding a vertex s with edges between s and every vertex of A_H, and
- adding a vertex t with edges between t and every vertex of B_H.

Let $\hat{c} = \frac{(1-c)dq+1}{3c-2}$ (we picked $\hat{c}$ so $\frac{\hat{c}}{dq+3\hat{c}} > (1-c)$). Let $\mathcal{F}$ be the collection of (a) a set $\mathcal{F}$ of graphs F_e built from each graph $H_e \in \mathcal{H}$, and (b) a set $\hat{\mathcal{F}}$ of graphs $F_{s,1}, \ldots, F_{s,\hat{c}}$, $F_{t,1}, \ldots, F_{t,\hat{c}}$, and $P_1, \ldots, P_{\hat{c}}$ built over the same set of nodes. $F_{s,1}, \ldots, F_{s,\hat{c}}$ are $\hat{c}$ copies of the same graph F_s which has edges between s and every other vertex. $F_{t,1}, \ldots, F_{t,\hat{c}}$ are $\hat{c}$ copies of the same graph F_t which has edges between t and every other vertex. $P_1, \ldots, P_{\hat{c}}$ are $\hat{c}$ copies of the same graph P which we now describe. Let $A = \{a_1, \ldots, a_q\}$, $B = \{b_1, \ldots, b_q\}$. P has

- all edges between s and (a_1, ℓ) for all $\ell \in [N]$,
- all edges between (a_j, ℓ_1) and (a_{j+1}, ℓ_2) for all $j \in \{1, \ldots, q-1\}$ and $\ell_1, \ell_2 \in [N]$,
- all edges between (a_q, ℓ_1) and (b_1, ℓ_2) for all $\ell_1, \ell_2 \in [N]$,
- all edges between (b_j, ℓ_1) and (b_{j+1}, ℓ_2) for all $j \in \{1, \ldots, q-1\}$ and $\ell_1, \ell_2 \in [N]$,
- all edges between (b_q, ℓ) and t for all $\ell \in [N]$,

No other edges are present in these graphs.

Now any c-relaxed solution must contain s, t, a vertex (u, i) for each vertex $u \in A_G$ and a vertex (v, j) for each vertex $v \in B_G$. Otherwise, the subgraph is not connected in F_s, F_t or P and thus all $\hat{c}$ copies of these graphs.

For our hardness result, we may assume that all the edges in the LabelCover-Max instance can be covered, and thus we may assume that any optimal solution S^* to the instance of Minimum Simultaneous Connected Steiner Cluster induces a density D^* of at least $\frac{1}{2q+2}$ in each graph as in the proof of Theorem 13.

By our hypothesis, we can approximate D^* to within an $\alpha = 2^{\log^{1-\varepsilon} n}$ factor in a c fraction of the graphs. Thus we obtain a solution S with density at least $\frac{1}{2\alpha q+2\alpha}$ in at least a c fraction of the graphs. The total number of graphs is $dq + 3\hat{c} = \frac{dq+1}{3c-2}$ and recall S contains s, t, a vertex (u, i) for some $i \in [N]$ for each vertex $u \in A_G$ and a vertex (v, i) for some $j \in [N]$ for each vertex $v \in B_G$. Furthermore, S induces at least one edge in at least $\frac{c(dq+1)}{3c-2} - 3\hat{c} = \frac{(4c-3)dq+c-3}{3c-2}$ graphs $F_e \in \mathcal{F}$ and thus a $c' = \frac{(4c-3)dq+c-3}{(3c-2)dq}$ fraction of these graphs. Finally, by the construction of $\hat{F}$, S has size at most $2\alpha q + 2\alpha < 3\alpha q$. We now use S to build a solution to the instance of LabelCover-Max.

Let $X = \{v \in G : |W_v \cap S| > 6\alpha\}$, where $W_v = \{(v, i) : i \in [N]\}$. Now $|X| < \frac{1}{2}q$, otherwise, $|S| > \frac{1}{2}q \cdot 6\alpha = 3\alpha q$. Furthermore, as G is d-regular the vertices in X cover at most half of the dq edges of G; thus the vertices in $\bar{X} = (A \cup B) \setminus X$ cover at least $(c' - \frac{1}{2})$ fraction of the edges.

Take the set $S' = \{(v, i) \in S : v \in \bar{X}\}$. From S', we build a random labelling by selecting a random node (v, i) in $S' \cap W_v$, for each vertex $v \in \bar{X}$. We then set $\ell(v) = i$. Because $|W_v \cap S| \leq 6\alpha$ for all $v \in \bar{X}$, any edge induced by $\bar{X}$ is covered by this labelling with probability at least $\frac{1}{36\alpha^2}$. Thus, this labelling covers at least $\frac{1}{36\alpha^2} \cdot (c' - \frac{1}{2}) \cdot dq$ edges.

Since c is a constant (greater than $\frac{4}{5}$), for dq large, the term

$$\frac{1}{36\alpha^2} \cdot (c' - \frac{1}{2}) = \frac{1}{36\alpha^2} \cdot \left(\frac{4c-3}{3c-2} - \frac{3-c}{3c-2} \frac{1}{dq} - \frac{1}{2} \right)$$

is bounded below by $\frac{1}{c''\alpha^2}$ for some c'' (since $\frac{4c-3}{3c-2} - \frac{1}{2} > 0$ when $c > \frac{4}{5}$). Since the approximation ration α for LabelCover-Max is an increasing function of $n = dq$, $\frac{1}{c''\alpha^2}$ is bounded below by $\frac{1}{\alpha^3}$. Again, this process can be derandomized and the result follows. □