HAL
open science

Top-k overlapping densest subgraphs

Esther Galbrun, Aristides Gionis, Nikolaj Tatti

To cite this version:

Esther Galbrun, Aristides Gionis, Nikolaj Tatti. Top-k overlapping densest subgraphs. Data Mining and Knowledge Discovery, 2016, 30 (5), pp.1134 - 1165. 10.1007/s10618-016-0464-z . hal-01399184

HAL Id: hal-01399184

https://hal.science/hal-01399184

Submitted on 25 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Top-k overlapping densest subgraphs

Esther Galbrun · Aristides Gionis · Nikolaj Tatti

the date of receipt and acceptance should be inserted later

Abstract Finding dense subgraphs is an important problem in graph mining and has many practical applications. At the same time, while large real-world networks are known to have many communities that are not well-separated, the majority of the existing work focuses on the problem of finding a *single* densest subgraph. Hence, it is natural to consider the question of finding the *top-k densest subgraphs*. One major challenge in addressing this question is how to handle overlaps: eliminating overlaps completely is one option, but this may lead to extracting subgraphs not as dense as it would be possible by allowing a limited amount of overlap. Furthermore, overlaps are desirable as in most real-world graphs there are vertices that belong to more than one community, and thus, to more than one densest subgraph. In this paper we study the problem of finding *top-k overlapping densest subgraphs*, and we present a new approach that improves over the existing techniques, both in theory and practice. First, we reformulate the problem definition in a way that we are able to obtain an algorithm with *constant-factor approximation guarantee*. Our approach relies on using techniques for solving the *max-sum diversification* problem, which however, we need to extend in order to make them applicable to our setting. Second, we evaluate our algorithm on a collection of benchmark datasets and show that it convincingly outperforms the previous methods, both in terms of quality and efficiency.

Esther Galbrun
Inria Nancy – Grand Est, France
E-mail: esther.galbrun@inria.fr

Aristides Gionis · Nikolaj Tatti
Helsinki Institute for Information Technology (HIIT) and
Department of Information and Computer Science, Aalto University, Finland
E-mail: aristides.gionis@aalto.fi

Nikolaj Tatti
E-mail: nikolaj.tatti@aalto.fi

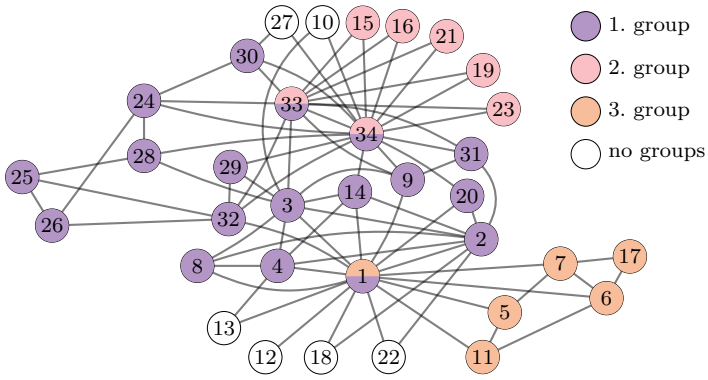

Fig. 1 Densest overlapping subgraphs on Zachary karate club dataset [44]. $k = 3$, $\beta = 2$.

1 Introduction

Finding dense subgraphs is a fundamental graph-mining problem, and has applications in a variety of domains, ranging from finding communities in social networks [25,33], to detecting regulatory motifs in DNA [15], to identifying real-time stories in news [3].

The problem of finding dense subgraphs has been studied extensively in theoretical computer science [2,8,13,24], and recently, due to the relevance of the problem in real-world applications, it has attracted considerable attention in the data-mining community [5,34–36,33]. In a domain where most interesting problems are **NP**-hard, much of the recent work has leveraged the fact that under a specific definition of density, the *average-degree density*, finding the densest subgraph is a polynomially-time solvable task [19]. Furthermore, there is a linear-time greedy algorithm that provides a factor-2 approximation guarantee [8].

The exact polynomial algorithm [19] and its fast approximation counterpart [8], apply only to the problem of finding the *single* densest subgraph. On the other hand, in most applications of interest we would like to find the top-k densest subgraphs in the input graph. Given an efficient algorithm for finding the single densest subgraph, there is a straightforward way to extend it in order to obtain a set of k dense subgraphs. This is a simple iterative method, in which we first find the densest subgraph, remove all vertices contained in that densest subgraph, and iterate, until k subgraphs are found or only an empty graph is left.

This natural heuristic has two drawbacks: First it produces a solution in which all discovered subgraphs are disjoint. Such disjoint subgraphs are often not desirable, as real-world networks are known to have not well-separated communities and hubs that may belong to more than one community [26], and hence, may participate in more than one densest subgraph. Second, when searching for the top-k densest subgraphs, we would like to maximize a global objective function, such as the sum of the densities over all k subgraphs and, as

shown by Balalau et al. [5], enforcing disjointness may lead to solutions that have very low total density, compared to solutions that allow some limited overlap among the discovered subgraphs.

From the above discussion it follows that it is beneficial to equip a top-k densest-subgraph discovery algorithm with the ability to find overlapping subgraphs. This is precisely the problem on which we focus in this paper. The challenge is to control the amount of overlap among the top-k subgraphs; otherwise one can find the densest subgraph and produce $k-1$ slight variations of it by adding or removing a very small number of vertices — so that the $k-1$ almost-copies are also very dense.

A simple example that demonstrates the concept of finding the top-k densest subgraphs with overlap is shown in Figure 1. The example is the famous Zachary karate club dataset [44], and the result is an actual execution of our algorithm, with $k = 3$. The importance of allowing overlap between dense subgraphs is immediate: subgraphs 1 and 2 overlap on vertices 33 and 34. Had those vertices been assigned only to subgraph 1, which was discovered first, subgraph 2 would fall apart.

Our paper follows up on the recent work of Balalau et al. [5]: finding top-k densest subgraphs with overlap. The main difference of the two papers is that here we use a distance function to measure overlap of subgraphs and penalize for overlap as part of our objective, while Balalau et al. are enforcing a hard constraint. Our approach allows to obtain a major improvement over the results of Balalau et al., both in theory and in practice. On the theoretical side, we provide an algorithm with worst-case approximation guarantee, while the method of Balalau et al. offers guarantees only for certain input cases. On the empirical side, our method outperforms the previous one in practically all datasets we experimented, with respect to all measures. In addition, in terms of computation time, our method is more scalable.

From the technical point of view, our approach is inspired by the results of Borodin et al. [7] for the *max-sum diversification* problem. In particular, Borodin et al. designed a greedy algorithm in order to find, in a ground set U, a set of elements S that maximizes a function of the form $f(S) + \lambda \sum_{\{x,y\}:x,y \in S} d(x,y)$, subject to $|S| = k$. Here f is a submodular function and d is a distance function. To apply this framework, we set S to be the set of k subgraphs we are searching for, f to capture the total density of all these subgraphs, and d to capture the distance between them — the larger the overlap the smaller the distance.

Yet, the greedy algorithm of Borodin et al. cannot be applied directly; there are a number of challenges that need to be addressed. The most important is that, for our problem, the iterative step of the greedy algorithm results in an **NP**-hard problem. This is a major difficulty, as the basic scheme of the algorithm of Borodin et al. assumes that the next best item in the greedy iteration can be obtained easily and exactly. To overcome this difficulty, we design an approximation algorithm for that subproblem. Then, we show that an approximation algorithm for the greedy step yields an approximation guarantee for the overall top-k densest-subgraph problem.

We apply our algorithm on a large collection of real-world networks. Our results show that our approach exhibits an excellent trade-off between density and overlap and outperforms the previous method. We also note that density and overlap are not comparable measures, and balancing the two is achieved via the parameter λ. As demonstrated in our experiments, executing the algorithm for a sequence of values of λ creates a density vs. overlap "profile" plot, which reveals the trade-off between the two measures, and allows the user to select meaningful values for the parameter λ.

The contributions of our work can be summarized as follows:

- We consider the problem of finding top-k overlapping densest subgraphs on a given graph. Our problem formulation relies on a new objective that provides a soft constraint to control the overlap between subgraphs.
- For the proposed objective function we present a greedy algorithm. This is the first algorithm with an *approximation guarantee* for the problem of finding top-k overlapping densest subgraphs.
- We evaluate the proposed algorithm against state-of-the art methods on a wide range of real-world datasets. Our study shows that our algorithm, in addition to being theoretically superior, it also outpeforms its main competitors in practice, in both quality of results and efficiency.

The rest of the paper is organized as follows. We start by reviewing the related work in Section 2. In Section 3 we define our problem, and in Section 4 we present an overview of background techniques that our approach relies upon. Our method is presented in Section 5, while the experimental evaluation of our algorithm and the comparison with state-of-the-art methods is provided in Section 6. Finally, Section 7 is a short conclusion. For convenience of presentation the proofs of the main claims are given in the Appendix.

2 Related work

Dense-subgraph discovery. As already discussed, the problem of finding dense subgraphs, and its variations, have been extensively studied in the theoretical computer science and graph mining communities. The complexity of the problem depends on the exact formulation. The quintessential dense graph is the clique, but finding large cliques is a very hard problem [22]. On the other hand, if density is defined as the average degree of the subgraph, the problem becomes polynomial. This was observed early on by Goldberg, who gave an algorithm based on a transformation to the minimum-cut problem [19]. For the same problem, Asahiro et al. [4] and Charikar [8] provided a greedy linear-time factor-2 approximation algorithm, making the problem tractable for very large datasets.

The previous algorithms do not put any constraint on the size of the densest subgraph. Requiring the subgraph to be of a certain size makes the problem **NP**-hard. Feige et al. show that, for a fixed k, the problem of asking for the dense subgraph of size *exactly* k can be approximated within a factor of

$\mathcal{O}(|V|^{\alpha})$, for $\alpha < \frac{1}{3}$ [13]. The problems of asking for a densest subgraph of size *at most* k or *at least* k are also **NP**-hard, and the latter problem can be approximated within a constant factor [2,24].

Other notions of density have been considered. Tsourakakis et al. provide algorithms for the problem of finding α-quasicliques and evidence that such subgraphs are, in practice, "more dense than the densest subgraph" [36]. Recently, Tsourakakis showed that finding triangle-dense subgraphs is also a polynomial problem, and gave a faster approximation algorithm [35] similar to Charikar's approach [8].

All of the above papers focus on the problem of finding a single densest subgraph. Seeking to find the top-k densest subgraphs is a much less studied problem. Tsourakakis et al. [36] address the question, but they only considered the naïve algorithm outlined in the introduction—remove and repeat. Balalau et al. [5] are the first to study the problem formally, and demonstrate that the naïve remove-and-repeat algorithm performs poorly. Their problem formulation is different than ours, as they impose a hard constraint on overlap. Their algorithm is also greedy, like ours, but there is no theoretical guarantee. On the other hand, our formulation allows to prove an approximation guarantee for the problem of top-k overlapping densest subgraphs. Furthermore, our experimental evaluation shows that our algorithm finds denser subgraphs for the same amount of overlap.

Finally, extracting top-k communities that can be described by conjunctions of labels was suggested by Galbrun et al. [16]. In this setup, the communities can overlap but an edge can be assigned to only one community. In addition, a community should be described with a label set. However, the algorithm can be easily modified such that it operates without labels and we use this modified algorithm as one of our baselines.

Community detection. Community detection is one of the most well-studied problems in data mining. The majority of the works deal with the problem of partitioning a graph into disjoint communities. A number of different methodologies have been applied, such as hierarchical approaches [18], methods based on modularity maximization [6,11,18,39], graph theory [14], random walks [31, 37,45], label propagation [37], and spectral graph partitioning [23,28,38]. A popular suite of graph-partitioning algorithms, which is accompanied by high-quality software, includes the Metis algorithm [23]. The Metis algorithm is one of the baselines against which we benchmark our approach in our experimental evaluation.

A considerable amount of work has also been devoted into the problem of finding *overlapping communities*. Different methods have been proposed to address this problem, relying on clique percolation [29], extensions to the modularity-based approaches [9,20,30], label propagation [21,41], analysis of ego-networks [12], game theory [10], non-negative matrix factorization [43] or edge clustering [1]. A comprehensive survey on the topic of overlapping community detection has been compiled by Xie et al. [40].

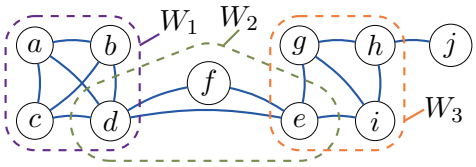

Fig. 2 Toy example

Our problem formulation is different than most existing work in community detection, as we focus solely in subgraph density, while typically community-detection approaches consider a combination of high density inside the communities and small cuts across communities. In fact, there has been experimental evidence that most real-world networks do not exhibit such structure of dense and relatively isolated communities [26]. To contrast our approach with methods for overlapping community detection, we experimentally compare our algorithm with the state-of-the-art method of Ahn et al. (Links) [1].

3 Preliminaries and problem definition

Throughout the paper we consider a graph $G = (V, E)$, where V is a set of vertices and E is a set of undirected edges. Given a subset of vertices $W \subseteq V$, we write

$$E(W) = \{(u, v) \in E \mid u, v \in W\},$$

for the edges of G that have both end-points in W, and $G(W) = (W, E(W))$ for the subgraph induced by W.

As we are interested in finding dense subgraphs of G, we need to adopt a notion of density.

Definition 1 The *density* of a graph $G = (V, E)$ is defined as $\mathrm{dens}(G) = |E|/|V|$.

Note that $\mathrm{dens}(G)$ is half of the average degree of G. If the graph $G = (V, E)$ is known from the context and we are given a subset of vertices $W \subseteq V$, we define the density of W by $\mathrm{dens}(W) = \mathrm{dens}(G(W))$, that is, the density of the induced subgraph $G(W)$. Additionally, we refer to the set $X \subseteq V$ that maximizes $\mathrm{dens}(X)$ as the *densest subgraph* of G.

Next we define the density of a collection of subsets of vertices.

Definition 2 Given a graph $G = (V, E)$ and a collection of subsets of vertices $\mathcal{W} = \{W_1, \ldots, W_k\}$, $W_i \subseteq V$, we define the density of the collection $\mathcal{W}$ to be the sum of densities of the individual subsets, that is,

$$\mathrm{dens}(\mathcal{W}) = \sum_{i=1}^{k} \mathrm{dens}(W_i)\,. \tag{1}$$

Example 1 Consider the graph given in Figure 2 with three subgraphs $\mathcal{W} = \{W_1, W_2, W_3\}$. The collective density of $\mathcal{W}$ is then equal

$$\text{dens}(\mathcal{W}) = \text{dens}(W_1) + \text{dens}(W_2) + \text{dens}(W_3) = \frac{6}{4} + \frac{3}{3} + \frac{5}{4} = \frac{15}{4}.$$

Our goal is to find a set of subgraphs $\mathcal{W}$ with high density $\text{dens}(\mathcal{W})$. At the same time we want to allow overlaps among the subgraphs of $\mathcal{W}$. Allowing overlaps without any restriction is problematic though, as the obvious solution is to find the densest subgraph of G and repeat it k times. Therefore we need a way to control the amount of overlap among the subgraphs of $\mathcal{W}$. To control overlaps we use a *distance function* defined over subsets of vertices. Let $d : 2^V \times 2^V \to \mathbb{R}_+$ denote such a distance function. We are then asking to find a set of subgraphs $\mathcal{W}$ so that the overall density $\text{dens}(\mathcal{W})$ is high, and the subgraphs in $\mathcal{W}$ are far apart, according to the distance function d.

More precisely, the problem definition that we work with in this paper is the following.

Problem 1 (Dense-Overlapping-Subgraphs) Given a graph $G = (V, E)$, a distance function d over subsets of vertices, a parameter λ, and an integer k, find k subgraphs $\mathcal{W} = \{W_1, \ldots, W_k\}$, $W_i \subseteq V$, which maximize a combined *reward function*

$$r(\mathcal{W}) = \text{dens}(\mathcal{W}) + \lambda \sum_{i=1}^{k} \sum_{j=i+1}^{k} d(W_i, W_j)\,.$$

A few remarks on our problem definition. First note that the parameter λ is necessary as the two terms that compose the reward function, density and distance, are quantitatively and qualitatively different. Even if one normalizes them to take values in the same range, say, between 0 and 1, they would still not be directly comparable. In fact, the parameter λ provides a weight between density and distance. A small value of λ places the emphasis on density and gives solutions with high overlap, but as λ increases the subgraphs of the optimal solution need to be more distant, at the expense of the density term. In our experimental evaluation we illustrate the effect of the parameter λ by drawing a "profile" of the solution space. By displaying the evolution of the two terms of the reward function as λ varies, such plots also allow to select meaningful values for this parameter.

Second, observe that we are not asking to assign every vertex of V in a subgraph of $\mathcal{W}$. The fraction of vertices assigned to at least one subgraph is controlled by both parameters λ and k. The higher the value of λ, the less overlap between the subgraphs, and thus, the higher the coverage. Similarly, the higher the value of k, the more subgraphs are returned, and thus, coverage increases.

Finally, we restrict the family of functions used to define the distance between subgraphs. As is common, we work with *metric* and *relaxed-metric* functions.

Definition 3 Assume a function d mapping pairs of objects to a non-negative real number. If there is a constant $c \geq 1$ such that (i) $d(x, y) = d(y, x)$, (ii) $d(x, x) = 0$, and (iii)

$$d(x, y) \leq c(d(x, z) + d(z, y)),$$

we say that the function d is a *c-relaxed metric*. If (iii) holds for $c = 1$, the function d is a *metric*.

The distance function we use to measure distance between subgraphs in this paper is a metric. However, our approach applies more generally to any c-relaxed metric. We formulate our results in their generality, so that the dependency on the parameter c of a c-relaxed metric becomes explicit.

4 Overview of background methods

In this section we provide a brief overview of some of the fundamental concepts and algorithmic techniques that our approach relies upon.

The densest-subgraph problem. We first discuss the problem of finding the densest subgraph, according to the average-degree density function, given in Definition 1. This problem of finding the densest-subgraph can be solved in polynomial time. An elegant solution that involves a mapping to a series of minimum-cut problems was given by Goldberg [19]. However, since the fastest algorithm to solve the minimum-cut problem runs in $\mathcal{O}(|V||E|)$ time, this approach is not scalable to very large graphs. On the other hand, there exists a linear-time algorithm that provides a factor-2 approximation to the densest-subgraph problem [4,8]. This is a greedy algorithm, which starts with the input graph, and iteratively removes the vertex with the lowest degree, until left with an empty graph. Among all subgraphs considered during this vertex-removal process, the algorithm returns the densest. Hereinafter, we refer to this greedy algorithm as the Charikar algorithm.

Max-sum diversification. Our approach is inspired by the results of Borodin et al. [7] for the *max-sum diversification* problem, which, in their general form, extend the classic Nemhauser et al. approximation results [27] for the problem of submodular function maximization. A brief description of the problem setting and the methods of Borodin et al. follows.

Let U be a ground set, let $d : U \times U \to \mathbb{R}_+$ be a *metric distance function* over pairs of U, and $f : 2^U \to \mathbb{R}_+$ be a *monotone submodular function* over subsets of U. Recall that a set function $f : 2^U \to \mathbb{R}$ is *submodular* if for all $S \subseteq T \subseteq U$ and $u \notin T$ we have $f(T \cup \{u\}) - f(T) \leq f(S \cup \{u\}) - f(S)$ [32]. Also the function f is *monotonically increasing* if for all $S \subseteq T \subseteq U$ $f(T) \geq f(S)$ holds. For the rest of the paper, we write *monotone* to mean monotonically increasing.

The max-sum diversification problem is defined as follows.

Problem 2 (Max-Sum-Diversification) Let U be a set of items, $f : 2^U \to \mathbb{R}_+$ a monotone submodular function on subsets of U, and $d : U \times U \to \mathbb{R}_+$ a distance function on pairs of U. Let k be an integer and $\lambda \geq 0$. The goal is to find a subset $S \subseteq U$ that maximizes

$$f(S) + \lambda \sum_{\{x,y\}:x,y\in S} d(x,y)\,, \quad \text{subject to } |S| = k.$$

Here, λ is a parameter that specifies the desired trade-off between the two objectives $f(\cdot)$ and $d(\cdot,\cdot)$. Borodin et al. [7] showed that under the conditions specified above, there is a simple linear-time factor-2 approximation greedy algorithm for the max-sum diversification problem. The greedy algorithm works as follows. For any subset $S \subseteq U$ and any element $x \notin S$ the *marginal gain* is defined as

$$\phi(x;S) = \frac{1}{2}\left(f(S \cup \{x\}) - f(S)\right) + \lambda \sum_{y\in S} d(x,y)\,. \tag{2}$$

The greedy starts with the empty set $S = \emptyset$ and proceeds iteratively. In each iteration it adds to S the element x which is currently not in S and maximizes the marginal gain $\phi(x;S)$. Note that due to the 1/2-factor in the first term, $\phi(x;S)$ is not equal to the actual gain of the score upon adding x to S. The algorithm stops after k iterations, when the size of S reaches k. As already mentioned, this simple greedy algorithm provides a factor-2 approximation to the optimal solution of the max-sum diversification problem.

5 The proposed method

As suggested in the previous section, our method for solving Problem 1 and finding dense overlapping subgraphs relies on the techniques developed for the max-sum diversification problem. First note that, in the Dense-Overlapping-Subgraphs problem, the role of function f is taken by the density function dens. The density of a collection of subgraphs is a simple summation (Definition 2) therefore it is clearly monotone and submodular.

The other components needed for applying the method of Borodin et al. [7] in our setting are not as simple. A number of challenges need to be addressed:

(1) While in the max-sum diversification problem we are searching for a set of elements in a ground set U, in the Dense-Overlapping-Subgraphs problem we are searching for a *collection of subsets* of vertices V. Thus, $U = \mathcal{P}(V)$ (the powerset of V) and the solution space is $\binom{\mathcal{P}(V)}{k}$.
(2) Adapting the max-sum diversification framework discussed in the previous section requires selecting the item in U that maximizes the marginal gain ϕ in each iteration of the greedy algorithm. Since U is now the powerset of V, the marginal gain computation becomes $\phi(W, \mathcal{W})$, where W is a subset of vertices, and $\mathcal{W}$ is the set of subgraphs previously selected by the greedy algorithm. Finding the subgraph W that maximizes the marginal gain now becomes a difficult computational problem. Thus, we need to

extend the framework to deal with the possibility that we are only able to find a subgraph that maximizes the marginal gain *approximately*.

(3) We need to specify the distance function $d(\cdot,\cdot)$ that will be used to measure overlap between subsets of vertices. There are many natural choices, but not all are applicable to the general max-sum diversification framework.

Challenges (2) and (3) are intimately connected. The problem of finding the set of vertices that maximizes the marginal gain ϕ in each iteration of the greedy algorithm crucially depends on the distance function d chosen for measuring distance between subgraphs. Part of our contribution is to show how to incorporate a distance function, which, on the one hand, is natural and intuitive, and, on the other hand, makes the marginal gain-maximization problem tractable. We present our solution in Section 5.2.

But before, we discuss how to extend the greedy algorithm in order to deal with approximations of the marginal gain-maximization problem and with c-relaxed metrics.

5.1 Extending the greedy algorithm

Consider the Max-Sum-Diversification problem, defined in Problem 2. To simplify notation, if X and Y are subsets of U we define

$$d(X,Y) = \sum_{x\in X}\sum_{y\in Y} d(x,y) \text{ and } d(X) = \frac{1}{2}\sum_{x,y\in X} d(x,y)\,,$$

so, the objective function in Problem 2 can be written as

$$r(S) = f(S) + \lambda d(S)\,.$$

The algorithm of Borodin et al. [7], discussed in the previous section, is referred to as Greedy. As already mentioned, Greedy proceeds iteratively: in each iteration it needs to find the item x^* of U that maximizes the marginal gain with respect to the solution set S found so far. The marginal gain is computed according to Equation (2). The problem is formalized below.

Problem 3 (Gain) Given a ground set U, a monotone, non-negative, and submodular function f, a c-relaxed metric d, and a set $S \subseteq U$, find $x \in U \setminus S$ maximizing

$$\phi(x;S) = \frac{1}{2}\left(f(S\cup\{x\}) - f(S)\right) + \lambda d(\{x\}\,,S)\,.$$

Now consider the case where the Gain problem cannot be solved optimally, but instead an approximation algorithm is available. In particular, assume that we have access to an oracle that solves Gain with an approximation guarantee of α. In this case, the Greedy algorithm still provides an approximation guarantee for the Max-Sum-Diversification problem. The quality of approximation of Greedy depends on α as well as the c parameter of the c-relaxed metric d. We prove the following proposition in Appendix.

Proposition 1 *Given a non-negative, monotone and submodular function f, a c-relaxed metric d, and an oracle solving* GAIN *with an approximation guarantee of α, the* Greedy *algorithm yields an approximation guarantee of $\alpha/(2c)$.*

5.2 Greedy discovery of dense subgraphs

We now present the final ingredients of our algorithm: (i) defining a metric between subsets of vertices; and (ii) formulating the problem of maximizing marginal gain, establishing its complexity, and giving an efficient algorithm for solving it. We first define the distance measure between subgraphs.

Definition 4 Assume a set of vertices V. We define the distance between two subsets $X, Y \subseteq V$ as

$$D(X,Y) = \begin{cases} 2 - \frac{|X \cap Y|^2}{|X||Y|} & \text{if } X \neq Y, \\ 0 & \text{otherwise.} \end{cases}$$

The distance function D resembles very closely the *cosine distance* ($1 -$ cosine similarity), but unlike the cosine distance, the distance function D is a metric.

Proposition 2 *The distance function D is a metric.*

Proof Clearly, D is symmetric and $D(X,X) = 0$. To prove the triangle inequality consider $X, Y, Z \subseteq V$. It is easy to see that the triangle inequality holds if at least two of the three sets are identical. So, assume that X, Y, and Z are all distinct. Note that $0 \leq 1 - \frac{|X \cap Y|^2}{|X||Y|} \leq 1$. This gives

$$D(X,Y) \leq 2 \leq D(X,Z) + D(Z,Y)\,,$$

which is the desired inequality. □

Example 2 Let us consider again the graph given in Figure 2 with $\mathcal{W} = \{W_1, W_2, W_3\}$. We already know that $\text{dens}(\mathcal{W}) = 15/4$. Let us adopt D as our metric. Then we have

$$D(W_1, W_2) = D(W_2, W_3) = 2 - \frac{1}{3 \times 4} = \frac{23}{12}$$

and $D(W_1, W_3) = 2$. We can now compute the reward of $\mathcal{W}$, $r(\mathcal{W}) = \frac{15}{4} + \lambda\left(2 + 2 \times \frac{23}{12}\right)$.

Next we formulate the problem of selecting the set of vertices that maximizes the marginal gain, in the iterative step of the greedy process, given a set of subgraphs selected so far. The following problem statement specializes Problem 3 in the context of the DENSE-OVERLAPPING-SUBGRAPHS problem. We denote the marginal gain with $\chi(\cdot;\cdot)$, instead of $\phi(\cdot;\cdot)$, to emphasize that it is a function of a vertex set given a collection of vertex sets.

Problem 4 (Dense-Subgraph) Given a graph $G = (V, E)$ and a collection of vertex sets $\mathcal{W} = \{W_1, \ldots, W_k\}$, $W_i \subseteq V$, find a set of vertices $U \subseteq V$ and $U \notin \mathcal{W}$ maximizing the marginal gain

$$\chi(U; \mathcal{W}) = \frac{1}{2}\mathrm{dens}(U) + \lambda D(\{U\}, \mathcal{W}) = \frac{1}{2}\mathrm{dens}(U) + \lambda \sum_{W \in \mathcal{W}} D(U, W)\,.$$

Example 3 Let us consider again the graph given in Figure 2. Set $\mathcal{W}' = \{W_1, W_2\}$ and consider adding W_3. In this case, the gain is equal to

$$\chi(W_3; \mathcal{W}') = \frac{1}{2}\mathrm{dens}(W_3) + \lambda(D(W_3, W_1) + D(W_3, W_2)) = \frac{1}{2} \times \frac{5}{4} + \lambda\left(2 + \frac{23}{12}\right).$$

We should stress that $\chi(W_3; \mathcal{W}')$ is not equal to the difference in rewards $r(\mathcal{W}' \cup \{W_3\}) - r(\mathcal{W}')$. This is due to the 1/2-factor in the density term.

In the definition of the Dense-Subgraph problem, dens is the density function given in Definition 1, while D is the subgraph distance function defined above. Had the term $\lambda D(\cdot, \cdot)$ been absent from the objective, the Dense-Subgraph problem would have been equivalent with the *densest-subgraph problem*, discussed in Section 4. In that case, the problem could be solved in polynomial time with Goldberg's algorithm, or approximated efficiently with Charikar's algorithm.

However, as we show next, adding the $\lambda D(\cdot, \cdot)$ term in the objective changes the complexity of the problem. In other words, in contrast to the densest-subgraph problem, Dense-Subgraph is not solvable in polynomial time.

Proposition 3 Dense-Subgraph *is* **NP**-*hard.*

Proof We consider the decision version of the problem. The problem is obviously in **NP**. To prove the completeness, we will use Regular-Clique. An instance of Regular-Clique consists of a regular graph (all vertices have the same degree) and an integer c. We are then asked to decide whether the graph contains a clique of size at least c [17].

Assume that we are given a d-regular graph $G = (V, E)$ with $n = |V|$ and $m = |E|$, and an integer $c > 2$. Define $\mathcal{W}$ as follows: for each $(x, y) \notin E$, add a set $\{x, y\}$ to $\mathcal{W}$. Finally, set $\lambda = m$.

Let U be a subgraph and let $p = \binom{|U|}{2} - |E(U)|$ be equal to the number of non-edges in U. A straightforward calculation shows that the gain $\chi(U; \mathcal{W})$ can be written as

$$\chi(U; \mathcal{W}) = \frac{|E(U)|}{2|U|} + 2\lambda|\mathcal{W}| - \lambda(n - d)\frac{1}{2} - \frac{\lambda p}{|U|}.$$

If we assume that U is a singleton set $\{u\}$, then the gain is equal to $\chi(\{u\}; \mathcal{W}) = \alpha = 2\lambda|W| - \lambda(n - d)/2$.

Let U be the subgraph with the optimal gain. Assume that $p \geq 1$. Then, since $|E(U)| \leq m = \lambda$ the gain is at most $\alpha - m/(2|U|)$, which is less than the gain of a singleton. Hence, $p = 0$ and the gain is equal to $(|U| - 1)/4 + \alpha$.

Algorithm 1: DOS; Algorithm for finding top-k overlapping densest subgraphs (problem Dense-Overlapping-Subgraphs)

Input: $G = (V, E), \lambda, k$
Output: set of subgraphs $\mathcal{W}$ s.t. $|\mathcal{W}| = k$ and maximizing $r(\mathcal{W})$
$\mathcal{W} \leftarrow \emptyset$;
foreach $i = 1, \ldots, k$ **do** $\mathcal{W} \leftarrow \mathcal{W} \cup \mathsf{Peel}(G, \mathcal{W}, \lambda)$;
return $\mathcal{W}$;

Hence, U will be the largest clique. It follows that the graph G contains a clique of size c if and only if Dense-Subgraph has a solution for which the gain is at least $(c-1)/4 + \alpha$. □

Despite this hardness result, it is still possible to devise an approximation algorithm for the Dense-Subgraph problem. Our algorithm, named Peel, is a variant of the Charikar algorithm for the densest-subgraph problem. Peel, similar to Charikar, starts with the whole graph and proceeds iteratively, removing one vertex in each step. Peel stops when there is no vertex left, and it returns the set of vertices that maximizes the gain function, selected among all vertex sets produced during the execution of the algorithm.

Peel has two main differences when compared to Charikar. First, instead of removing the minimum-degree vertex in each iteration, Peel removes the vertex that minimizes the following *adjusted degree* expression

$$\deg(v; V_i) - 4\lambda \sum_{W_j \ni v} \frac{|V_i \cap W_j|}{|W_j|}.$$

Here V_i stands for the set of vertices that constitute the candidate in the current iteration—after removing some vertices in earlier iterations. The intuition for using this *adjusted degree* is to lower the gain associated to vertices that belong to subgraphs selected in earlier steps of the greedy process. Indeed, we want to favor high-degree vertices but we want to penalize such vertices that are contained in previously selected subgraphs and thereby generate overlap with (i.e. reduce the distance to) the current subgraph. A further difficulty is that we do not know the current subgraph (since we are currently searching for it!) so we use as a proxy the set of vertices still contained in the candidate at that step (V_i). Despite making this seemingly crude approximation, as we will see shortly, the Peel algorithm provides an approximation guarantee to the Dense-Subgraph problem.

The second difference between Peel and Charikar is the following: it is possible that Peel returns a subgraph that has been selected previously. This could happen if the value of the parameter λ is small compared to dense subgraphs that may be present in the input graph. When Peel returns a previously-selected subgraph U, it is sufficient to *modify* U: we can either add one vertex, remove one vertex, or just replace U with a trivial subgraph of size 3; among all these options we select the best solution according to our marginal gain objective χ. A detailed description of this process is given in Algorithm 3.

Algorithm 2: Peel; finds a dense subgraph U of the graph G, overlapping with a collection of previously discovered subgraphs $\mathcal{W}$.

Input: $G = (V, E), \mathcal{W}, \lambda$
Output: U maximizing $\chi(U; \mathcal{W})$

$V_n \leftarrow V$;
foreach $i = n, \ldots, 2$ **do**
  $v \leftarrow \arg\min_v \left\{ \deg(v; V_i) - 4\lambda \sum_{W_j \ni v} \frac{|V_i \cap W_j|}{|W_j|} \right\}$;
  $V_{i-1} \leftarrow V_i \setminus \{v\}$;
foreach $i = 1, \ldots, n$ **do**
  if $V_i \in \mathcal{W}$ **then** $V_i \leftarrow \mathsf{Modify}(V_i, G, \mathcal{W}, \lambda)$;
return $\arg\max_{V_j} \{\chi(V_j; \mathcal{W})\}$;

Algorithm 3: Modify; modifies U if $U \in \mathcal{W}$

Input: $U, G, \mathcal{W}, \lambda$
Output: modified U

$X \leftarrow \{U \cup \{x\} \mid x \notin U,\ U \cup \{x\} \notin \mathcal{W}\}$;
$Y \leftarrow \{U \setminus \{y\} \mid y \in U,\ U \setminus \{y\} \notin \mathcal{W}\}$;
if $X = \emptyset$ **and** $\mathrm{dens}(U) \leq 5/3$ **then**
  $U \leftarrow \{$a wedge of size 3 not in $\mathcal{W}\}$;
else
  $U \leftarrow \arg\max_{C \in X \cup Y} \{\chi(C; \mathcal{W})\}$;
return U;

For the quality of approximation of Peel, which is detailed in Algorithm 2, we can show the following result, which is proved in Appendix.

Proposition 4 *Assume that we are given a graph $G = (V, E)$, a collection of previously discovered vertex sets $\mathcal{W}$ and $\lambda > 0$. Assume that $|\mathcal{W}| < |V|$ and G contains more than $|\mathcal{W}|$ wedges, i.e. connected subgraphs of size 3. Then* Peel *yields* $2/10$ *approximation for* Dense-Subgraph.

The approximation guarantee of $2/10$ is rather pessimistic due to pathological cases, and we can obtain a better ratio if we consider these cases separately. In particular, if Peel does not call Modify, then the approximation ratio is $1/2$. If $X \neq \emptyset$ during Modify, then the approximation ratio is at least $\frac{|U|}{2(|U|+1)}$, otherwise the ratio is at least $2/10$.

We note that the main function of Modify is to allow us to prove a *worst-case* approximation guarantee; i.e., for *all* possible values of λ. In practice, if Modify is called for a certain value of λ, the user should perceive this as a signal that λ is too small (as overlaps are not penalized enough) and should increase it.

5.3 The DOS algorithm

Finally, we consider the specialization of the generic Greedy algorithm studied above for the problem at hand in this paper. To obtain an algorithm for the Dense-Overlapping-Subgraphs problem, we instantiate f and d with functions whose domains consist of collections of vertices. Specifically, we let $f = \text{dens}$ and $d = D$. The resulting overall algorithm for the Dense-Overlapping-Subgraphs problem, named DOS, is shown as Algorithm 1. By combining Propositions 1 and 4 we obtain the following result.

Theorem 1 DOS *is an* $\frac{1}{10}$*-approximation algorithm for the* Dense-Overlapping-Subgraphs *problem.*

Computational complexity: DOS consists of k successive runs of Peel, iteratively removing the vertex with minimum *adjusted degree*. To perform this operation fast, vertices are stored into separate worker queues according to the previous subgraphs they belong to. Since such vertices incur the same distance penalty, we keep the queues sorted by degree. There are at most $t = \min\{2^k, |V|\}$ such queues. In practice, the number is much smaller. To find the next vertex, we compare the top vertex of every worker queue. The search can be done in $\mathcal{O}(t)$ time. Upon deletion of a vertex, updating a single adjacent vertex in a worker queue can be done in constant time,[1] while updating the distance penalties can be done in $\mathcal{O}(k)$ time. If we are forced to call Modify, then finding the best subgraph in X and Y can be done in $\mathcal{O}(|E| + |V|k)$ time. In a rare pathological case we are forced to return a wedge. We can do this by first computing k wedges before invoking DOS. If we ever need a wedge, we simply select a wedge from this list that is not yet in $\mathcal{W}$. Consequently, we can execute DOS in $\mathcal{O}(k(|E| + |V|(t + k)))$ time.

6 Experiments

In this section we report on the experimental evaluation of our proposed algorithm DOS. Our python implementation of the DOS algorithm is publicly available online.[2]

First, we explore the behavior of the algorithms in a controlled setting, using synthetic networks with planted ground-truth subgraphs. Then, we evaluate the algorithms on real-world networks for which the ground-truth is not accessible. In these two parts, we use different evaluation measures, as required by the distinct goals of the two scenarios.

Algorithms. In both parts, our primary baseline is the MAR algorithm recently proposed by Balalau et al. [5]. Similarly to our approach, their goal is to

[1] Here we use the fact that edges are not weighted, and consequently the queue can be implemented as an array of linked lists of vertices.

[2] `http://research.ics.aalto.fi/dmg/dos_code.tgz`

extract dense overlapping subgraphs while controlling the Jaccard coefficient between the discovered subgraphs.

Both methods MAR and DOS allow/require to adjust the overlap permitted between subgraphs in the solution. In MAR, this is done by fixing a strict maximum threshold for the Jaccard coefficient between two subgraphs, while in DOS, by setting the value of the parameter λ, which balances density and distance.

For DOS, note that the density of subgraphs vary from dataset to dataset, while the distance function D is always below 2. Consequently, in practice it is easier to set the value of λ relative to the density of the first subgraph discovered by the algorithm. Note that when extracting the first subgraph, λ has no effect since the distance part of the score given in Problem 4 is 0. Thus, when running the algorithm we provide a value β and we set $\lambda = \beta \mathrm{dens}(W_1)$.

For MAR, we denote the Jaccard threshold by σ. Note that larger values of σ result in increased overlap with MAR but larger values of β reduce overlap with DOS.

6.1 Finding planted subgraphs

In the first part of our experiments, we consider synthetic networks and evaluate how well the algorithms are able to recover planted ground-truth subgraphs.

Datasets. Our synthetic networks are generated as follows: each network consists of a backbone of five subsets of vertices. We generate networks both with and without overlaps. In networks without overlap, the subgraphs simply consists of five disjoint subsets of 30 vertices. In configurations with overlap, the subgraphs are arranged in a circle, they are assigned 20 vertices of their own plus 10 vertices shared with the subgraph on one side and 10 shared vertices with the other side.

We consider two families of networks depending on whether connections are generated using the Erdős-Rényi model with fixed densities between 0.6 and 0.9, or using the Barabási-Albert preferential attachment model.

This way, we obtain four types of noise-free networks. We also consider noisy variants, that is, networks where noise is added on top of the backbone. Specifically, we double the number of vertices in the network by adding new vertices that do not belong to any of the subsets, and the connections are flipped with the probability of 0.01.

For each of the eight configurations, we generate ten synthetic networks, each containing five planted subgraphs.[3]

Statistics. We adopt an approach similar to [42] and compare the sets of subgraphs detected by the algorithms to the planted ground-truth using the following measures.

[3] The synthetic networks used in our experiments are available at `http://research.ics.aalto.fi/dmg/dos_synth.tgz`

Table 1 Performance of DOS and MAR on synthetic data. *For each dataset, we compare the detected subgraphs to the ground truth using the Normalized Mutual Information (NMI), F_1 scores ($F_1[t/d]$ and $F_1[d/t]$) and Omega index (Ω).*

method	param.	NMI	$F_1[t/d]$	$F_1[d/t]$	Ω	NMI	$F_1[t/d]$	$F_1[d/t]$	Ω
		Erdős-Rényi				Barabási-Albert			
		No overlap, noise-free							
DOS	2.0	0.94	0.95	0.93	0.95	0.96	0.87	0.86	0.91
	1.0	0.94	0.96	0.96	0.95	0.65	0.75	0.79	0.73
	0.5	0.93	0.97	0.97	0.93	0.70	0.64	0.76	0.80
MAR	0.1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.92**	**0.92**	**0.95**
	0.5	**1.00**	**1.00**	**1.00**	**1.00**	0.81	0.54	0.81	0.88
	0.75	0.92	0.82	0.97	0.94	0.79	0.54	0.85	0.86
		No overlap, noisy							
DOS	2.0	0.84	0.83	0.72	0.87	0.72	0.71	0.69	0.93
	1.0	0.69	0.87	0.81	0.62	0.66	0.65	0.70	0.95
	0.5	0.81	0.91	0.90	0.95	0.57	0.61	0.68	0.93
MAR	0.1	**0.90**	**0.93**	0.88	0.98	**0.98**	**0.90**	**0.89**	**0.98**
	0.5	0.88	**0.93**	**0.92**	**0.98**	0.57	0.67	0.76	0.93
	0.75	0.74	0.83	0.87	0.94	0.45	0.57	0.73	0.92
		With overlap, noise-free							
DOS	2.0	0.27	0.72	0.72	0.41	**0.34**	0.71	0.69	**0.75**
	1.0	0.27	0.82	0.79	0.20	0.33	**0.72**	**0.73**	0.75
	0.5	0.25	**0.87**	**0.83**	0.22	0.24	0.70	0.72	0.66
MAR	0.1	0.19	0.56	0.51	**0.51**	0.24	0.56	0.56	0.68
	0.5	**0.28**	0.66	0.70	0.44	0.12	0.58	0.60	0.68
	0.75	0.22	0.71	0.72	0.33	0.04	0.53	0.56	0.63
		With overlap, noisy							
DOS	2.0	0.26	0.58	0.54	0.45	**0.34**	0.67	0.64	**0.93**
	1.0	**0.28**	0.74	0.72	0.79	0.31	**0.68**	**0.69**	0.92
	0.5	0.26	**0.87**	**0.84**	0.79	0.21	0.65	0.67	0.90
MAR	0.1	0.16	0.53	0.27	**0.84**	0.16	0.49	0.37	0.87
	0.5	0.12	0.56	0.51	0.83	0.11	0.56	0.58	0.91
	0.75	0.04	0.53	0.51	0.79	0.03	0.51	0.54	0.89

NMI: the Normalized Mutual Information between the ground-truth and the detected subgraphs,

$F_1[t/d]$: the average F_1-score of the best-matching ground-truth subgraph to each detected subgraph ("*t*ruth to *d*etected"),

$F_1[d/t]$: the average F_1-score of the best-matching detected subgraph to each ground-truth subgraph ("*d*etected to *t*ruth"),

Ω: the Omega index, i.e., the fraction of vertex pairs that share the same number of communities in the solution and the ground-truth.

Table 2 Datasets statistics. *For each dataset, we indicate how many graphs it consists of (#G), the number of edges (|E|) and vertices (|V|), as well as the density (*dens(G)*) across all graphs in the collection.*

Set	#G	|E|			|V|			dens(G)		
		min	max	avg	min	max	avg	min	max	avg
DBLP.E2	13	1 427	13 251	6 435.77	721	4 598	2 197.38	1.98	3.69	2.84
DBLP.C	2	10 689	11 208	10 948.50	2 891	3 140	3 015.50	3.40	3.88	3.64
G+.S	41	204	41 123	13 146.22	35	1 842	479.34	3.75	68.52	23.48
G+.L	22	42 810	176 691	96 124.32	1 007	3 799	1 840.68	15.85	103.28	55.05
FB	10	146	30 025	8 508.70	52	1 034	408.90	2.81	40.19	14.31
BKGW	12	555	218 434	46 152.67	358	46 942	11 785.25	1.55	5.38	3.29
XL	4	1 992 636	11 095 298	5 099 402.25	281 903	1 696 415	997 230.25	2.63	7.07	5.29

Each of these four measures takes value between 0 and 1, with values closer to 1 indicating highest similarity between the detected subgraphs and the planted ground-truth.

Results. The averages of these statistics over ten datasets for each configuration are shown in Table 1. We observe that the MAR algorithm performs better in the absence of overlap and noise, and is able to recover the ground truth perfectly in the case of the Erdős-Rényi graphs. This result can be explained by the fact that MAR handles overlaps with a hard constraint. However, our method is clearly superior when the backbone communities overlap, and also more resistant to noise, that is, for graphs that resemble better real-world application scenarios.

Further experiments, in a setup where the five planted subgraphs have different sizes, show that the difference between the two algorithms is less pronounced. In particular, in the absence of overlap and noise, the improved performance of DOS gets closer to the relatively stable performance of MAR, while in the presence of overlap and noise the performance of MAR is improved compared to the balanced case and gets closer to the performance of DOS, which remains relatively stable.

6.2 Comparison on real-world networks

We now turn to an empirical evaluation of our proposed algorithm on real-world datasets.

Datasets. We use the following networks.

Co-authorship networks: The first dataset is the DBLP network,[4] where vertices represent researchers, and edges represent co-authorship relations.

From this co-authorship network we extract smaller instances. First, we consider some ego-net graphs. We start with 13 high-profile computer scien-

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

tists,[5] and consider their ego-nets of radius 2. We collectively refer to this collection of ego-nets as `DBLP.E2`. Second, we consider the subgraphs induced by researchers who have published in the ICDM and KDD conferences, respectively, giving rise to two networks that form the `DBLP.C` dataset.

The remaining collections consist of networks distributed via the Stanford network analysis project.[6]

Social circles: Ego-nets collected from Google+ users who have shared their circles are divided into two collections. The first one, `G+.S`, contains relatively small ego-nets, having fewer than 42 000 edges, while the second, `G+.L`, contains ego-nets with larger numbers of edges. The `FB` collection contains 10 ego-nets representing friend lists from Facebook.

Location-based social networks: We also consider a collection of location-based networks from Brightkite and Gowalla, two websites that allow there users to share their location. We extract smaller instances by applying the following procedure: we assign each user to their most frequent location and divide the network into 8 broad geographic areas.[7] We denote this collection of 16 networks as `BKGW`.

Large graphs: Finally, denoted by `XL`, we consider a number of large graphs that were used by Balalau et al. [5]: web graphs from Stanford and Google, the YouTube social network, and the internet topology graph from Skitter.

Statistics on all datasets are provided in Table 2.

Statistics. Again we compare our algorithm to its main competitor, the MAR algorithm. We apply both algorithms on every dataset to extract the top-k overlapping subgraphs, while varying the overlap parameters β and σ. For each run we obtain a set of subgraphs $\mathcal{W} = \{W_1, \ldots, W_k\}$. In some cases, MAR returns fewer than k subgraphs, therefore $|\mathcal{W}|$ can be smaller than k.

For each solution $\mathcal{W}$, we compute its *coverage*, that is, the ratio of vertices that belong to at least one subgraph $C(\mathcal{W}) = \left|\cup_{i=1}^{k} W_i\right| / |V|$, the *average size* of the subgraphs $|W|$ and the average *vertex multiplicity* over covered vertices, where the multiplicity of vertex v, denoted as $\mu(v)$, is the number of subgraphs it belongs to.

In addition, for each solution we compute (i) the average *density* $\mathrm{dens}(W_i)$ over all subgraphs, (ii) the average *distance* $D(W_i, W_j)$ over all pairs of subgraphs, (iii) the average *Jaccard distance*

$$J(W_i, W_j) = 1 - \frac{|W_i \cap W_j|}{|W_i \cup W_j|}$$

[5] Namely, S. Abiteboul, E. Demaine, M. Ester, C. Faloutsos, J. Han, G. Karypis, J. Kleinberg, H. Mannila, K. Mehlhorn, C. Papadimitriou, B. Shneiderman, G. Weikum and P. Yu.

[6] `http://snap.stanford.edu`

[7] Namely, Oceania, Latin-America, the USA, Europe, the Middle-East and East Asia

over all pairs of subgraphs, and (iv) the *modularity*

$$Q(\mathcal{W}) = \frac{1}{2|E|} \sum_{W \in \mathcal{W}} \sum_{u,v \in W} \left(A_{uv} - \frac{\delta(u)\delta(v)}{2|E|} \right) \frac{1}{\mu(u)\mu(v)},$$

where $\mu(v)$ is the number of subgraphs that vertex v belongs to, $\delta(v)$ its degree, and A the graph adjacency matrix.

Results. Tables 3–6 provide summaries of our experimental comparison for DBLP and G+ datasets. We report the number of subgraphs ($|\mathcal{W}|$), average density ($\overline{\text{dens}}$), average distance ($\overline{D}$), average Jaccard distance ($\overline{J}$), modularity(Q), coverage (C), average subgraph size ($\overline{|W|}$) and average vertex multiplicity ($\overline{\mu(v)}$) averaged across all networks in the collection. Results for the remaining collections are similar, and are thus omitted.

The profile of solutions obtained on five networks from different collections, for a wide range of values of β and σ are shown in Figure 3. Each row corresponds to the profile of one network: the ego-net of C. Papadimitriou from DBLP.E2, the KDD network from DBLP.C, the 1183. . . 6467 ego-net from G+.S, the 1684 ego-net from FB, and the Brightkite network of Latin-America from BKGW. The columns show the average distance ($\overline{D}$), average Jaccard distance ($\overline{J}$) and coverage (C) of solutions plotted against average density ($\overline{\text{dens}}$).

For reference, we added three points to the profiles, representing the solutions obtained respectively with three baseline methods which do not allow to adjust the overlap tolerance. Namely, we considered

Links: The algorithm of Ahn et al. [1] discovers overlapping subgraphs by performing a hierarchical clustering on the edges rather than the vertices of the input graph. We consider the top 20 densest subgraphs returned by this method.

Metis: We apply the popular spectral graph-partitioning algorithm by Karypis and Kumar [23] to obtain a complete partition of the graph vertices into k disjoint sets.

Dense This algorithm, used as a baseline by Galbrun et al. [16], extracts dense subgraphs in the same iterative process similar to the DOS algorithm but edges are allowed to contribute to at most one subgraph and are assigned where they benefit most.

At one end of the range of solutions returned by DOS lies the solution that corresponds to setting $\lambda = 0$, i.e., focusing entirely on density. This solution consists of copies of the densest subgraph. At the other end, for large values of λ, lies the solution that does not tolerate any overlap.

Expectedly, we observe that the density of the subgraphs returned increases as the distance decreases, that is, as more overlap is allowed, both algorithms can find denser subgraphs. However, we note that DOS exploits the overlap allowance better, returning subgraphs with greater average densities than MAR for similar values of distance D and Jaccard distance. Additionally, we see that DOS tends to return larger subgraphs and to achieve better coverage scores.

Table 3 Results for `DBLP.E2`, $k = 20$. *For each algorithm/parameter pair we report the number of subgraphs ($|\mathcal{W}|$), average density ($\overline{\text{dens}}$), average distance ($\overline{D}$), average Jaccard distance ($\overline{J}$), modularity(Q), coverage (C), average subgraph size ($\overline{|W|}$) and average vertex multiplicity ($\overline{\mu(v)}$) averaged across all networks in the collection.*

method	param.	$\lvert\mathcal{W}\rvert$	$\overline{\text{dens}}$	$\overline{D}$	$\overline{J}$	Q	$\overline{C}$	$\overline{\lvert W\rvert}$	$\overline{\mu(v)}$
DOS	1.0	20	5.09	2.00	0.99	0.098	0.71	96	1.33
	0.1	20	7.59	1.93	0.90	0.034	0.50	168	3.08
	0.01	20	9.72	1.50	0.49	0.008	0.24	232	9.52
	0.001	20	10.01	1.06	0.06	0.003	0.11	200	15.91
MAR	0.25	20	4.64	2.00	0.98	0.142	0.63	81	1.22
	0.5	20	4.97	1.96	0.93	0.111	0.66	118	1.66
	0.75	20	6.28	1.85	0.81	0.066	0.49	140	2.77
	0.95	20	8.82	1.55	0.52	0.017	0.21	156	6.92

Table 4 Results for `DBLP.C`, $k = 20$. *Columns as in Table 3.*

method	param.	$\lvert\mathcal{W}\rvert$	$\overline{\text{dens}}$	$\overline{D}$	$\overline{J}$	Q	$\overline{C}$	$\overline{\lvert W\rvert}$	$\overline{\mu(v)}$
DOS	1.0	20	6.45	2.00	0.99	0.117	0.62	117	1.27
	0.1	20	10.92	1.92	0.88	0.049	0.49	216	2.94
	0.01	20	14.97	1.40	0.39	0.005	0.17	301	11.97
	0.001	20	15.39	1.03	0.03	0.003	0.11	299	18.11
MAR	0.25	20	5.84	2.00	0.99	0.115	0.40	74	1.23
	0.5	20	6.69	1.95	0.92	0.098	0.45	114	1.71
	0.75	20	8.70	1.84	0.79	0.054	0.34	147	2.85
	0.95	20	13.17	1.38	0.36	0.011	0.15	206	9.46

Table 5 Results for `G+.S`, $k = 20$. *Columns as in Table 3.*

method	param.	$\lvert\mathcal{W}\rvert$	$\overline{\text{dens}}$	$\overline{D}$	$\overline{J}$	Q	$\overline{C}$	$\overline{\lvert W\rvert}$	$\overline{\mu(v)}$
DOS	1.0	20	11.08	2.00	0.99	0.036	0.98	37	1.71
	0.1	20	42.81	1.83	0.78	0.010	0.96	130	6.38
	0.01	20	67.41	1.22	0.21	0.002	0.55	172	14.67
	0.001	20	68.60	1.04	0.04	0.001	0.43	168	18.53
MAR	0.25	14	13.11	1.98	0.96	0.059	0.91	36	1.29
	0.5	17	16.45	1.91	0.87	0.038	0.90	42	1.91
	0.75	19	27.42	1.75	0.71	0.020	0.79	62	3.55
	0.95	20	55.74	1.34	0.32	0.005	0.53	121	10.81

Table 6 Results for `G+.L`, $k = 20$. *Columns as in Table 3.*

method	param.	$\lvert\mathcal{W}\rvert$	$\overline{\text{dens}}$	$\overline{D}$	$\overline{J}$	Q	$\overline{C}$	$\overline{\lvert W\rvert}$	$\overline{\mu(v)}$
DOS	1.0	20	30.58	2.00	0.99	0.035	0.91	124	1.55
	0.1	20	111.37	1.85	0.81	0.010	0.90	406	5.27
	0.01	20	175.83	1.25	0.24	0.001	0.45	545	14.10
	0.001	20	178.91	1.04	0.04	0.001	0.33	531	18.89
MAR	0.25	20	25.50	1.98	0.96	0.051	0.84	96	1.31
	0.5	20	40.77	1.90	0.86	0.032	0.76	128	1.93
	0.75	20	72.69	1.73	0.69	0.015	0.62	190	3.57
	0.95	20	147.08	1.31	0.30	0.003	0.40	380	11.34

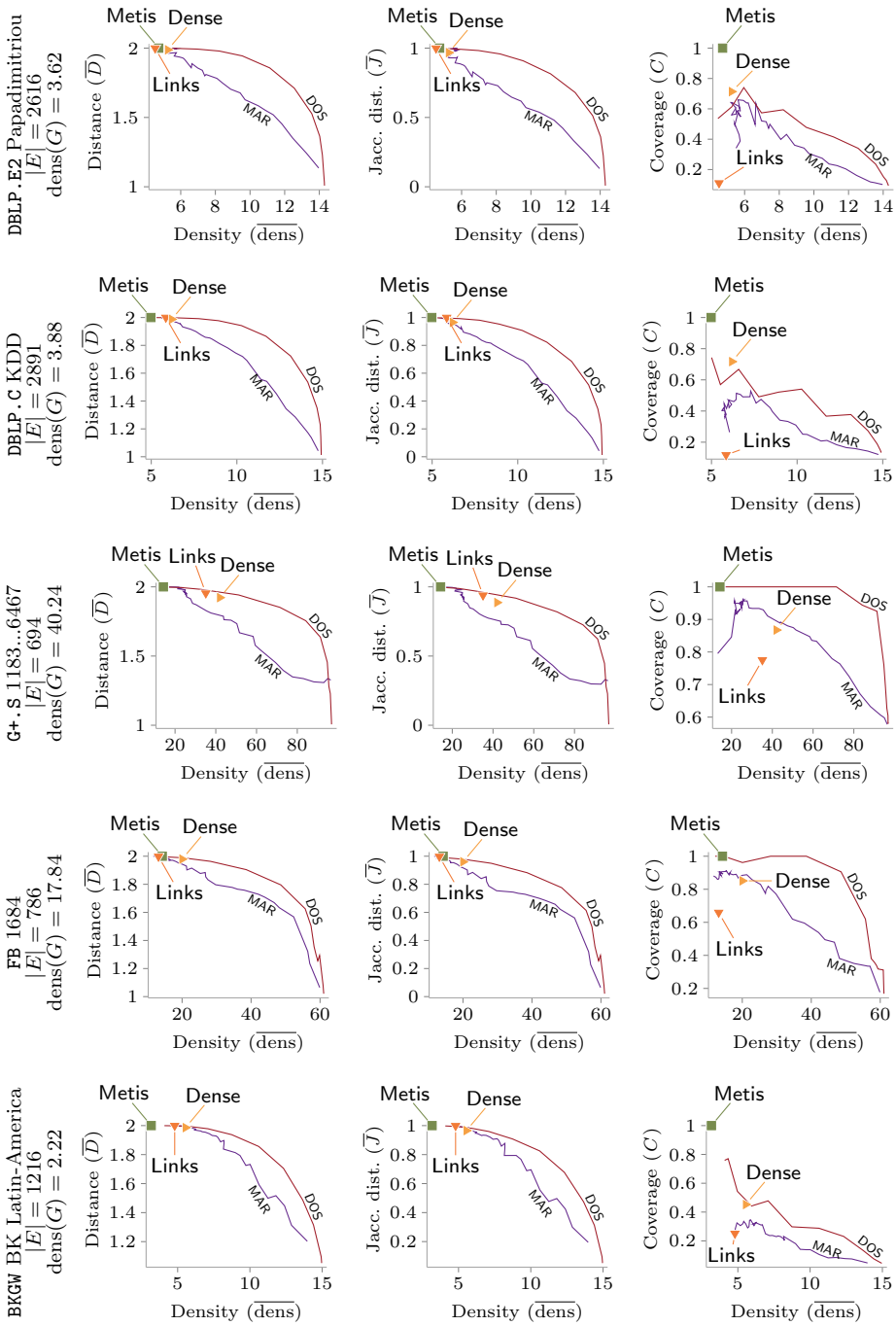

Fig. 3 Solution profiles for five networks. *Each row corresponds to the profile of one network, as indicated on the left.*

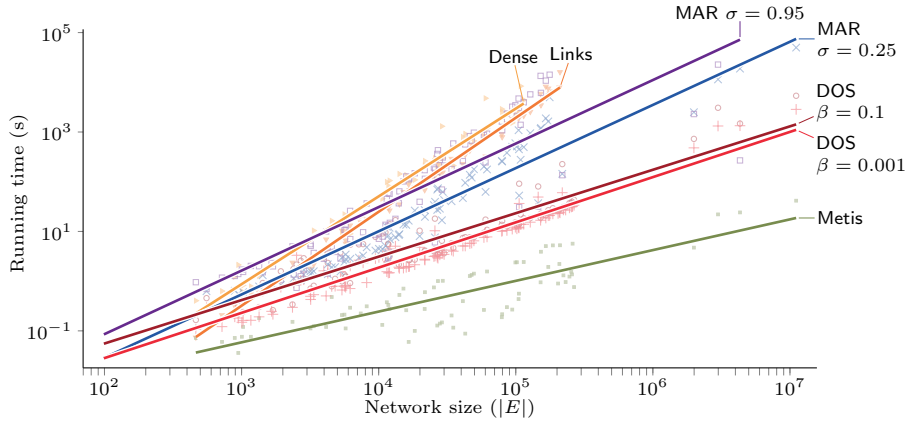

Fig. 4 Running times of DOS and MAR with different values of their overlap parameters, as well as the three baselines, on networks of increasing sizes sampled from the seven datasets.

6.3 Running times

Finally, we evaluate the scalability of our method. Figure 4 shows running times for DOS and MAR, as a function of the size of the network, measured by the number of edges, for different values of their overlap parameters.

The same figure also provide running times for the baseline methods and we can observe that both MAR and DOS scale better than Links and Dense but not as well as Metis.

On average, our algorithm is faster than MAR. An added advantage is the better stability of the running time across values of the overlap parameter, with the extreme value $\sigma = 0.95$ resulting in much larger running time for MAR. We should also note here that Balalau et al. [5] offer a heuristic that is faster than MAR, but at the expense of quality.

7 Concluding remarks

We studied the problem of discovering dense and overlapping subgraphs. Our approach optimizes density as well as the diversity of the obtained collection. Our solution, inspired by the work of Borodin et al. [7] and Charikar [8], is an efficient greedy algorithm with an approximation guarantee of 1/10. Our method improves significantly the previous work on the problem by Balalau et al. [5]. Not only do we provide an approximation guarantee, while the baseline is a heuristic, but it also yields practical improvements, both in terms of quality and efficiency. Our experiments, especially Figure 3, demonstrate that we obtain denser and more diverse subgraphs.

Our approach has two parameters: the number of subgraphs k and the parameter λ controlling the relative importance of density and overlap. Since our method is greedy, the problem of selecting k is somewhat alleviated as the

first k subgraphs will remain constant if we increase k. Selecting λ is more difficult as the relation between the density and the diversity terms is not obvious. We approach this problem by computing profiles as in Figure 3 by varying λ and selecting values that provides a good trade-off between density and overlap.

This work opens several new directions for future work. We have shown that a subproblem, Dense-Subgraph, is **NP**-hard, however, we did not establish the hardness of the main problem Dense-Overlapping-Subgraphs. We conjecture that it is also **NP**-hard. Another open question is to improve the approximation guarantee, as well as to study what other types of density functions and overlap distances can be used in our framework.

References

1. Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.
2. Reid Andersen and Kumar Chellapilla. Finding dense subgraphs with size bounds. In *Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph (WAW)*, pages 25–37, 2009.
3. Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *Proceedings of the Very Large Data Bases Endowment (VLDB)*, 5(6):574–585, 2012.
4. Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 136–148, 1996.
5. Oana Denisa Balalau, Francesco Bonchi, TH Chan, Francesco Gullo, and Mauro Sozio. Finding subgraphs with maximum total density and limited overlap. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 379–388, 2015.
6. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.
7. Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 155–166, 2012.
8. Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 84–95, 2000.
9. Mingming Chen, Konstantin Kuzmin, and Boleslaw Szymanski. Extension of modularity density for overlapping community structure. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 856–863, 2014.
10. Wei Chen, Zhenming Liu, Xiaorui Sun, and Yajun Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Mining and Knowledge Discovery*, 21(2):224–240, 2010.
11. Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, page 066111, 2004.
12. Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. DEMON: a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 615–623, 2012.
13. Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

14. Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 150–160, 2000.
15. Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14):150–157, 2006.
16. Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. Overlapping community detection in labeled graphs. *Data Mining and Knowledge Discovery*, 28(5-6):1586–1610, 2014.
17. Michael Garey and David Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. WH Freeman & Co., 1979.
18. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826, 2002.
19. Andrew V Goldberg. Finding a maximum density subgraph. Technical report, University of California, Berkley, 1984.
20. Steve Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 2007 European Conference on Principles and Practice of Knowledge Discovery in Databases, Part I (ECML/PKDD)*, pages 91–102, 2007.
21. Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10), 2010.
22. Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–636, 1996.
23. George Karypis and Vipin Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC)*, pages 1–13. IEEE Computer Society, 1998.
24. Samir Khuller and Barna Saha. On finding dense subgraphs. In *Automata, Languages and Programming*, pages 597–608, 2009.
25. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16):1481–1493, 1999.
26. Jure Leskovec, Kevin Lang, Anirban Dasgupta, and Michael Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.
27. George Nemhauser, Laurence Wolsey, and Marshall Fisher. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294, 1978.
28. Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856, 2001.
29. Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.
30. J. Pinney and D. Westhead. Betweenness-based decomposition methods for social and biological networks. In *Interdisciplinary Statistics and Bioinformatics*, pages 87–90, 2006.
31. Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms Applications*, 10(2):284–293, 2006.
32. Alexander Schrijver. *Combinatorial optimization*. Springer, 2003.
33. Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 939–948, 2010.
34. Nikolaj Tatti and Aristides Gionis. Density-friendly graph decomposition. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 1089–1099, 2015.
35. Charalampos Tsourakakis. The k-clique densest subgraph problem. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 1122–1132, 2015.
36. Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. Denser than the densest subgraph: Extracting optimal quasi-cliques with

quality guarantees. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 104–112, 2013.
37. Stijn van Dongen. *Graph Clustering by Flow Simulation.* PhD thesis, University of Utrecht, 2000.
38. Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
39. Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 76–84, 2005.
40. Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4):43, 2013.
41. Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *International Conference on Data Mining Workshops (ICDMW)*, 2011.
42. Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM)*, pages 1170–1175, 2012.
43. Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a non-negative matrix factorization approach. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining, (WSDM)*, pages 587–596, 2013.
44. Wayne Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, pages 452–473, 1977.
45. Haijun Zhou and Reinhard Lipowsky. Network Brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities. In *Computational Science (ICCS)*, volume 3038, pages 1062–1069, 2004.

A Proof of Proposition 1

Let us first define $h(x;Y) = [f(x \cup Y) - f(Y)]/2$ and

$$g(x;Y) = h(x;Y) + d(Y \cup x) - d(Y) = h(x;Y) + d(x,Y)\,.$$

For proving the proposition, we will need Lemma 1.

Lemma 1 *Let d be a c-relaxed metric. Let X and Y be two disjoint sets. Then*

$$c(|X|-1)d(X,Y) \geq |Y|d(X)\,.$$

Proof Let $y \in Y$ and $x, z \in X$. By definition,

$$c(d(x,y) + d(z,y)) \geq d(x,z)\,.$$

For a given $x \in X$, there are exactly $|X|-1$ pairs (x,z) such that $x \neq z \in X$. Consequently, summing over all $x, z \in X$ such that $x \neq z$ gives us

$$2c(|X|-1)d(X,y) \geq 2d(X)\,.$$

Summing over $y \in Y$ proves the lemma. □

Proof (Proof of Proposition 1) Let $G_1 \subset \cdots \subset G_k$ be the sets during Greedy. Fix $1 \leq i \leq k$. Then G_i is the current solution after i-th iteration of Greedy.

Let O be the optimal solution. Write $A = O \cap G_i$, $C = O \setminus A$, and $B = G_i \setminus A$. Lemma 1 implies that

$$c(|A|-1)d(A,C) \geq |C|d(A)\,,$$

which in turn implies

$$\begin{aligned}|C|i(d(A)+d(A,C)) &\le ci(|A|-1)d(A,C)+|C|id(A,C)\\ &= ci(|A|-1+|C|)d(A,C)\\ &= ci(k-1)d(A,C)\,.\end{aligned}$$

Moreover, Lemma 1 implies that

$$\begin{aligned}c(|C|-1)d(B,C) &\ge |B|d(C)\\ c(|C|-1)d(A,C) &\ge |A|d(C)\,,\end{aligned}$$

which, together with $|C| = |B| + k - i$, implies

$$\begin{aligned}|C|id(C) &= (k-i)id(C)+|B|id(C)\\ &= (k-i)(|A|+|B|)d(C)+|B|id(C)\\ &= (k-i)|A|d(C)+|B|kd(C)\\ &\le c(k-i)(|C|-1)d(A,C)+ck(|C|-1)d(B,C)\\ &\le c(k-i)(k-1)d(A,C)+ck(k-1)d(B,C)\,.\end{aligned}$$

Combining these two inequalities leads us to

$$\begin{aligned}|C|id(O) &= |C|id(A)+|C|id(C)+|C|id(A,C)\\ &\le ck(k-1)(d(A,C)+d(B,C))\\ &= ck(k-1)d(G_i,C)\,.\end{aligned}$$

Submodularity and monotonicity imply

$$\begin{aligned}\sum_{v\in C} g(v;G_i) &= \sum_{v\in C}\left[h(v;G_i)+d(\{v\},G_i)\right]\\ &= \Big(\sum_{v\in C} h(v;G_i)\Big)+d(C,G_i)\\ &\ge \frac{1}{2}\left[f(O)-f(G_i)\right]+\frac{i|C|}{ck(k-1)}d(O)\\ &\ge \frac{1}{2}\left[f(O)-f(G_k)\right]+\frac{i|C|}{ck(k-1)}d(O)\,.\end{aligned}$$

Let u_i be the item added at the $i+1$th step, $G_{i+1} = \{u_i\} \cup G_i$. Then, since $g(u_i;G_i) \ge \alpha g(v;G_i)$ for any $v \in C$,

$$g(u_i;G_i) \ge \frac{\alpha}{2k}\left[f(O)-f(G_k)\right]+\frac{i\alpha}{ck(k-1)}d(O)\,.$$

Summing over i gives us

$$\frac{1}{2}f(G_k)+d(G_k) = \sum_{i=0}^{k-1} g(u_i;G_i) \ge \frac{\alpha}{2}\left[f(O)-f(G_k)\right]+\frac{\alpha}{2c}d(O)\,.$$

Since $\alpha \le 1$ and $c \ge 1$, we have

$$r(G_k) = f(G_k)+d(G_k) \ge \frac{\alpha}{2}f(O)+\frac{\alpha}{2c}d(O) \ge \frac{\alpha}{2c}r(O)\,,$$

which completes the proof. □

B Proof of Proposition 4

To prove the proposition we need to first show that Modify does not decrease the gain of a set significantly.

Lemma 2 *Assume a graph $G = (V, E)$. Assume a collection of k distinct subgraphs $\mathcal{W}$ of G, and let $U \in \mathcal{W}$. Assume that $k < |V|$ and G contains more than k wedges, i.e. connected subgraphs of size 3. Let $M = \mathsf{Modify}(U, G, \mathcal{W}, \lambda)$. Then $\chi(V; \mathcal{W}) \geq 2/5 \times (\chi(U, \mathcal{W}) + \lambda)$.*

Proof Write $r = |U|$ and $\alpha = \frac{r}{r+1}$. We will split the proof in two cases. *Case 1:* Assume that X, as given in Algorithm 3, is not empty. Select $B \in X$. We will show that

$$\text{dens}(B) \geq \alpha \text{dens}(U) \quad \text{and} \quad D(B, W) \geq \alpha(D(U, W) + I[U = W]),$$

for any $W \in \mathcal{W}$, where $I[U = W] = 1$ if $U = W$, and 0 otherwise. This automatically guarantees that

$$\chi(B; \mathcal{W}, \lambda) \geq \alpha(\chi(U; \mathcal{W}, \lambda) + \lambda),$$

proving the result since $\alpha \geq 1/2$ and the gain of M is at least as good as the gain of B.

To prove the first inequality, note that

$$\text{dens}(B) = \frac{|E(B)|}{r+1} \geq \frac{|E(U)|}{r+1} = \alpha \frac{|E(U)|}{r} = \alpha \text{dens}(U)\,.$$

To prove the second inequality fix $W \in \mathcal{W}$, and let $p = |W|$, $q = |W \cap U|$. Define

$$\Delta = D(U, W) + I[U = W] = 2 - \frac{q^2}{rp} = \frac{2rp - q^2}{rp},$$

Let v be the only vertex in $B \setminus U$. If $v \notin W$, then $D(B, W) \geq \Delta$. Hence, we can assume that $v \in W$. This leads to

$$\begin{aligned} D(B, W) &= 2 - \frac{|B \cap W|^2}{|B||W|} \\ &= 2 - \frac{(1+q)^2}{(1+r)p} = \frac{2p(1+r) - (1+q)^2}{(1+r)p}. \end{aligned}$$

Let us define β as the fraction of the numerators,

$$\beta = \frac{2p(1+r) - (1+q)^2}{2rp - q^2}.$$

We wish to show that $\beta \geq 1$. Since $p \geq q + 1$,

$$\begin{aligned} \beta &= \frac{2p(1+r) - (1+q)^2}{2rp - q^2} = \frac{2rp - q^2 + 2p - 2q - 1}{2rp - q^2} \\ &\geq \frac{2rp - q^2 + 2(q+1) - 2q - 1}{2rp - q^2} = \frac{2rp - q^2 + 1}{2rp - q^2} \geq 1. \end{aligned}$$

The ratio of distances is now

$$\frac{D(B, W)}{\Delta} = \beta \frac{r}{r+1} \geq \frac{r}{r+1} = \alpha.$$

This proves the first case.

Case 2: Assume that $X = \emptyset$. Then we must have $Y \neq \emptyset$ and $r \geq 2$, as otherwise $|\mathcal{W}| \geq |V|$, which violates the assumption of the lemma.

Assume that $\text{dens}(U) \geq 5/3$. Let $B \in Y$. Removing a single item of U decreases the density by 1, at most. This gives us

$$\frac{\text{dens}(B)}{\text{dens}(U)} \geq \frac{\text{dens}(U)-1}{\text{dens}(U)} \geq \frac{5/3-1}{5/3} = \frac{2}{5}.$$

To bound the distance term, fix $W \in \mathcal{W}$, and let $p = |W|$, $q = |W \cap U|$. Let v be the only vertex in $U \setminus V$. Define $\Delta = D(U,W) + I[U = W]$. If $v \in W$, then we can easily show that $D(V,W) \geq \Delta$. Hence, assume that $v \notin W$. This implies that $q \leq \min p, r-1$, or $q^2 \leq p(r-1)$. As before, we can express the distance term as

$$\Delta = 2 - \frac{q^2}{rp} = \frac{2rp - q^2}{rp},$$

and

$$D(B,W) = 2 - \frac{|B \cap W|^2}{|B||W|} = 2 - \frac{q^2}{(r-1)p} = \frac{2p(r-1)-q^2}{(r-1)p}.$$

The ratio is then

$$\begin{aligned}\frac{D(B,W)}{\Delta} &= \frac{2p(r-1)-q^2}{2rp-q^2}\frac{r}{r-1} \\ &\geq \frac{2p(r-1)-p(r-1)}{2rp-p(r-1)}\frac{r}{r-1} = \frac{p(r-1)}{rp+p}\frac{r}{r-1} = \frac{r}{r+1} \geq 1/2,\end{aligned}$$

where the first inequality follows from the fact that the ratio is decreasing as function of q.

Assume that $\text{dens}(U) < 5/3$. By assumption there is a wedge B outside $\mathcal{W}$. Since $\text{dens}(B) \geq 2/3$, we have $\text{dens}(B)/\text{dens}(U) \geq 2/5$. The distance terms decrease by a factor of $1/2$, since

$$D(U,W) \leq 2 = 2 \times 1 \leq 2D(B,W).$$

Combining the inequalities proves that

$$\chi(B;\mathcal{W},\lambda) \geq \frac{2}{5}\chi(U;\mathcal{W},\lambda)$$

which proves the lemma. □

Proof (Proof of Proposition 4) To prove the proposition, we will first form a new graph H, and show that the *density* of a subgraph in H is closely related to the *gain*. This then allows us to prove the statement.

Let us first construct the graph H: given a vertex v let us define

$$s(v) = -\sum_{v \in W_j} \frac{2\lambda}{|W_j|}.$$

Let $H = (V, E', c)$ be a fully connected weighted graph with self-loops where the weight of an edge $c(v,w)$ is

$$c(v,w) = I[(v,w) \in E] - \sum_{j|v,w \in W_j} \frac{4\lambda}{|W_j|},$$

for $v \neq w$, and $c(v,v) = s(v)$.

Next, we connect the gain of set of vertices U (w.r.t. G) with the weighted density of U in H. Given an arbitrary set of vertices U, we will write $c(U)$ to mean the total weight of edges in H. Each $c(v,w)$, for $v \in w$, participates in $\deg_H(v;U)$ and $\deg_H(w;U)$, and each $c(v,v) = s(v)$ participates (once) in $\deg_H(v;U)$. This leads to

$$2c(U) = \sum_{v \in U} \deg_H(v;U) + s(v).$$

We can express the (weighted) degree of a vertex in H as

$$\begin{aligned}\deg_H(v;U) &= s(v) + \sum_{\substack{w\in U\\ w\neq v}} c(v,w) = \deg_G(v;U) - \sum_{j|v\in W_j}\frac{2\lambda}{|W_j|} - \sum_{\substack{w\in U\\ w\neq v}}\sum_{j|v,w\in W_j}\frac{4\lambda}{|W_j|}\\ &= \deg_G(v;U) - \lambda\sum_{j|v\in W_j}\frac{4|U\cap W_j|-2}{|W_j|}.\end{aligned} \tag{3}$$

Write $k = |\mathcal{W}|$. These equalities lead to the following identity,

$$\begin{aligned}\mathrm{dens}(U;H) + 4\lambda k &= \frac{1}{|U|}c(U) + 4\lambda k\\ &= 4\lambda k + \frac{1}{2|U|}\sum_{v\in U}\deg_H(v;U) + s(v)\\ &= 4\lambda k + \mathrm{dens}(U;G) - \frac{1}{2|U|}\sum_{v\in U}\lambda\sum_{j|v\in W_j}\frac{4|U\cap W_j|}{|W_j|}\\ &= \mathrm{dens}(U;G) - 2\lambda\sum_{j=1}^{k}2 - \frac{|U\cap W_j|^2}{|U||W_j|}\\ &= 2\chi(U;\mathcal{W},\lambda) + \epsilon(U,\mathcal{W}),\end{aligned} \tag{4}$$

where $\epsilon(U,\mathcal{W})$ is a correction term, equal to 2λ if $U \in \mathcal{W}$, and 0 otherwise.

Let O be the densest subgraph in H. Next we show that during the for-loop Peel finds a graph whose density clost to $\mathrm{dens}(O;H)$. Let o be the first vertex in O deleted by Peel. We must have

$$\deg_H(o;O) \geq \mathrm{dens}(O;H)\,,$$

as otherwise we can delete o from O and obtain a better solution. Let $R = V_i$ be the graph at the moment when o is about to be removed. Let us compare $\deg_H(o;O)$ and $\deg_H(o;R)$. We can lower-bound of the second term of the right-hand side in Equation (3) by $-4k\lambda - s(v)$. Since $O \subseteq R$, this gives us

$$\begin{aligned}\deg_H(o;O) &\leq \deg_G(o;O) \leq \deg_G(o;R)\\ &\leq \deg_H(o;R) + s(o) + 4k\lambda.\end{aligned}$$

To upper-bound the first two terms, note that by definition of Peel, the vertex o has the smallest $\deg_H(o;R) + s(o)$ among all the vertices in R. Hence,

$$\deg_H(o;R) + s(o) \leq \sum_{v\in R}\frac{\deg_H(v;R)+s(v)}{|R|} = 2\frac{c(R)}{|R|} = 2\mathrm{dens}(R;H)\,.$$

To complete the proof, let O' be the graph outside $\mathcal{W}$, maximizing the gain. Due to Eq. 4, we have

$$\begin{aligned}2\chi\big(O';\mathcal{W},\lambda\big) &= \mathrm{dens}\big(O';H\big) + 4k\lambda \leq \mathrm{dens}(O;H) + 4k\lambda\\ &\leq 2\mathrm{dens}(R;H) + 8k\lambda = 2(\mathrm{dens}(R;H) + 4k\lambda)\\ &= 4\chi(R;\mathcal{W},\lambda) + 2\epsilon(R;\mathcal{W}).\end{aligned}$$

Let S be the set returned by Peel.

If $R \notin \mathcal{W}$, then $\epsilon(R;\mathcal{W}) = 0$. Moreover, R is not modified, and is one of the graphs that is tested for gain. Consequently, $\chi(S;\mathcal{W}) \geq \chi(R;\mathcal{W})$, proving the statement.

If $R \in \mathcal{W}$, then it is modified by Modify to, say, R'. Lemma 2 implies that $5/2 \times \chi(R',\mathcal{W}) \geq \chi(R;\mathcal{W}) + \epsilon(R;\mathcal{W})$. Since, $\chi(S;\mathcal{W}) \geq \chi(R';\mathcal{W})$, this completes the proof. □