Coverage Centrality Maximization in Undirected Networks

Gianlorenzo D'Angelo
Gran Sasso Science Institute
L'Aquila, Italy
gianlorenzo.dangelo@gssi.it

Martin Olsen
Department of Business Development and Technology
Aarhus University, Denmark
martino@btech.au.dk

Lorenzo Severini
ISI Foundation
Turin, Italy
lorenzo.severini@isi.it

Abstract

Centrality metrics are among the main tools in social network analysis. Being central for a user of a network leads to several benefits to the user: central users are highly influential and play key roles within the network. Therefore, the optimization problem of increasing the centrality of a network user recently received considerable attention. Given a network and a target user v, the centrality maximization problem consists in creating k new links incident to v in such a way that the centrality of v is maximized, according to some centrality metric. Most of the algorithms proposed in the literature are based on showing that a given centrality metric is monotone and submodular with respect to link addition. However, this property does not hold for several shortest-path based centrality metrics if the links are undirected.

In this paper we study the centrality maximization problem in undirected networks for one of the most important shortest-path based centrality measures, the coverage centrality. We provide several hardness and approximation results. We first show that the problem cannot be approximated within a factor greater than $1 - 1/e$, unless $P = NP$, and, under the stronger gap-ETH hypothesis, the problem cannot be approximated within a factor better than $1/n^{o(1)}$, where n is the number of users. We then propose two greedy approximation algorithms, and show that, by suitably combining them, we can guarantee an approximation factor of $\Omega(1/\sqrt{n})$. We experimentally compare the solutions provided by our approximation algorithm with optimal solutions computed by means of an exact IP formulation. We show that our algorithm produces solutions that are very close to the optimum.

Introduction

Determining what are the most important nodes in a network is one of the main goals of network analysis (Newman 2010). Several so-called *centrality metrics* have been proposed in the literature to try to quantitatively measure the importance of a node according to network properties like: distances between nodes (e.g. closeness or harmonic centrality), number of shortest paths passing through a node (e.g. betweenness or coverage centrality), or on spectral graph properties (e.g. PageRank or information centrality).

It has been experimentally observed that nodes with high centrality values play key roles within the network. For example, closeness centrality is significantly correlated with the influence of users in a social network (Crescenzi et al. 2016; Macdonald et al. 2012), while shortest-path-based metrics are correlated with the number of passengers passing trough an airport in transportation networks (Ishakian et al. 2012; Malighetti et al. 2009). The coverage centrality of a node v is the number of distinct pairs of nodes for which a shortest path passes through v. Nodes with high coverage centrality are pivotal to the communication between many other nodes of the network.

Generally speaking, centrality metrics are positively correlated with many desirable properties of nodes, therefore, there has been a recent considerable interest on finding strategies to increase the centrality value of a given node in order to maximize the benefits for the node itself. In this paper we focus on the most used strategy which is that of modifying the network by adding a limited number of new edges incident to the node itself. In detail we study the following optimization problem: given a graph G, a node v of G, and an integer k, find k edges to be added incident to v maximising the centrality value of v in the graph G augmented with these edges. The problem is usually referred to as the *centrality maximization problem* and it can be instantiated by using different centrality metrics such as: PageRank (Avrachenkov and Litvak 2006; Olsen and Viglas 2014), eccentricity (Demaine and Zadimoghaddam 2010; Perumal, Basu, and Guan 2013), coverage centrality (Ishakian et al. 2012; Medya et al. 2018), betweenness (Bergamini et al. 2018; D'Angelo, Severini, and Velaj 2016), information centrality (Shan, Yi, and Zhang 2018), closeness and harmonic centrality (Crescenzi et al. 2016). The centrality maximization problem is in general NP-hard but in all the mentioned cases the authors were able to devise algorithms ensuring a constant approximation factor. In Table 1 we list the bounds on the approximation ratio reported in these references.

Most of these approximation algorithms are based on a fundamental result on submodular optimization due to Nemhauser et al. (Nemhauser, Wolsey, and Fisher 1978). Given a monotone submodular set function f [1] and an integer k the problem of finding a set S with $|S| \leq k$ that

[1] A set function f is is submodular if for any pair of sets $A \subseteq B$ and any element $e \notin B$, $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$.

Centrality metric	Graph type	Approximation Algorithm	Hardness of approximation
Harmonic	Undir.	$1-\frac{1}{e}$	$1-\frac{1}{15e}$
	Dir.	$1-\frac{1}{e}$	$1-\frac{1}{3e}$
Betweenness	Undir.	OPEN	$1-\frac{1}{2e}$
	Dir.	$1-\frac{1}{e}$	$1-\frac{1}{2e}$
Eccentricity	undir.	$2+\frac{1}{OPT}$	$\frac{3}{2}$
PageRank	Dir.	$\left(1-\alpha^2\right)\left(1-\frac{1}{e}\right)$	NO FPTAS
Information	Undir.	$1-\frac{1}{e}$	OPEN
Constr. coverage	Undir.	$1-\frac{1}{e}$	OPEN
Coverage	**Undir.**	$\mathbf{\Omega(1/\sqrt{n})}$	$\mathbf{1-\frac{1}{e}}$ $\mathbf{1/n^{o(1)}}$
	DAGs	$1-\frac{1}{e}$	OPEN

Table 1: Summary of approximation bounds for the centrality maximization problem. The "Constr. coverage" row refers to the version of the coverage centrality maximization problem with the additional constraint that a pair of nodes can be covered by at most one edge. The results in this paper are marked in bold, the second hardness bound is under the Gap-ETH condition.

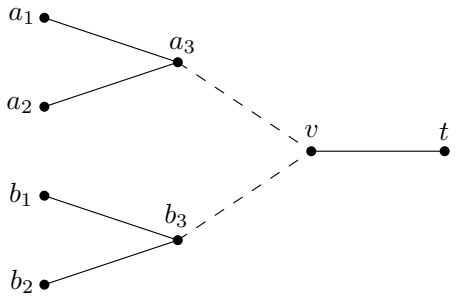

Figure 1: Adding edge $\{a_3, v\}$ increases the coverage centrality of v by 3, since nodes a_i will have a shortest path to t passing through v. Similarly, adding edge $\{b_3, v\}$ increases the coverage centrality of v by 3. However, adding both edges $\{a_3, v\}$ and $\{b_3, v\}$ will increase the coverage centrality of v by 15, since besides the shortest paths between nodes a_i and t and those between nodes b_i and t we need to take into account the 9 shortest paths between nodes a_i and nodes b_i that pass through v.

maximises $f(S)$ is NP-hard and hard to approximate within a factor greater than $1-1/e$, unless $P=NP$ (Feige 1998). However, the greedy algorithm that starts with the empty set and repeatedly adds an element that gives the maximal marginal gain of f guarantees the optimal approximation factor of $1-1/e$ (Nemhauser, Wolsey, and Fisher 1978). Many of the approximation algorithms for the centrality maximization problem are based on the fact that the value for node v of the considered centrality metric is monotone and submodular with respect to the addition of edges incident to v.

Unfortunately, not all the centrality metrics exhibit a submodular trend. Indeed, it has been shown that in *undirected graphs* some shortest-path based metrics are not submodular and, furthermore, the greedy algorithm exhibits an arbitrarily small approximation factor (D'Angelo, Severini, and Velaj 2016). For example, Figure 1 shows an undirected graph in which the increment in coverage centrality is not submodular with respect to edge addition. This is in contrast with the results for the same centrality metrics on *directed graphs*, where, e.g., betweenness and coverage centrality are monotone and submodular (Bergamini et al. 2018; Ishakian et al. 2012). Not being submodular makes things much harder and so far finding an approximation algorithm for the centrality maximization problem on shortest-paths based metrics has been left as an open problem (Bergamini et al. 2018; D'Angelo, Severini, and Velaj 2016). To overcome this issue, Medya et al. (Medya et al. 2018) consider the coverage centrality maximization problem with the additional artificial constraint that a pair of nodes can be covered by at most one edge. This constraint on the solution avoids the cases in Figure 1 and makes the objective function submodular. However, it does not consider solutions that cover pairs of nodes with pairs of edges, hence it looks for sub-optimal solutions to the general problem.

In this paper we give the first results on the general coverage centrality maximization problem in undirected graphs. In the remainder of the paper we will refer to this problem as the *Maximum Coverage Improvement* (MCI) problem. Our results can be summarized as follows (see also Table 1).

- MCI cannot be approximated within a factor greater that $1-1/e$, unless $P=NP$.
- MCI is at least as hard to approximate as the well-known Densest-k-subgraph problem and hence cannot be approximated within any constant, if the Unique Games with Small Set Expansion conjecture (Raghavendra and Steurer 2010) holds, and within $1/n^{o(1)}$, where n is the number of nodes in the graph, if the Gap Exponential Time Hypothesis holds (Manurangsi 2017).
- We propose two greedy approximation algorithms for MCI that guarantee, respectively, approximation factors of $1-e^{-\frac{(1-\epsilon)(t-1)}{k-1}}$ and $(1-\epsilon)(1-\frac{1}{e})^2\frac{k}{4n}$, where $t\geq 2$ is a constant tuning parameter and ϵ is any positive constant.
- We show that combining the two proposed algorithms we can achieve an approximation factor of $\Omega(1/\sqrt{n})$.
- We implemented the proposed algorithms and experimentally compared the solutions provided by our approximation algorithm with optimal solutions computed by means of an exact IP formulation. We experimentally show that our algorithm produces solutions that are very close to the optimum and that it is highly effective in increasing the coverage centrality of a target user.

Notation and problem statement

Let $G=(V,E)$ be an undirected graph where $|V|=n$ and $|E|=m$. For each node v, N_v denotes the set of neighbors of v, i.e. $N_v=\{u \mid \{u,v\}\in E\}$. Given two nodes s and t, we denote by d_{st} and P_{st} the distance from s to t in G and the set of nodes in any shortest path from s to t in G, respectively. For each node v, the *coverage centrality* (Yoshida

2014) of v is defined as the number of pairs (s,t) such that v is contained in a shortest path between s and t, formally,

$$c_v = |\{(s,t) \in V \times V \mid v \in P_{st}, v \neq s, v \neq t\}|.$$

In this paper, we consider graphs that are augmented by adding a set S of arcs not in E. Given a set $S \subseteq (V \times V) \setminus E$ of arcs, we denote by $G(S)$ the graph augmented by adding the arcs in S to G, i.e. $G(S) = (V, E \cup S)$. For a parameter x of G, we denote by $x(S)$ the same parameter in graph $G(S)$, e.g. the distance from s to t in $G(S)$ is denoted as $d_{st}(S)$.

The coverage centrality of a node might change if the graph is augmented with a set of arcs. In particular, adding arcs incident to some node v can increase the coverage centrality value of v. We are interested in finding a set S of arcs incident to a particular node v that maximizes $c_v(S)$. Therefore, we define the *Maximum Coverage Improvement* (MCI) problem as follows: Given an undirected graph $G = (V, E)$, a node $v \in V$, and an integer $k \in \mathbb{N}$, find a set S of arcs incident to v, $S \subseteq \{\{u, v\} \mid u \in V \setminus N_v\}$, such that $|S| \leq k$ and $c_v(S)$ is maximized.

Hardness of approximation

We first show that MCI cannot be approximated within a factor greater that $1 - 1/e$, unless $P = NP$. Then, we show that, under stronger conditions, it cannot be approximated to within a factor greater than $n^{-f(n)}$, for any $f \in o(1)$.

Constant bound

Our first hardness of approximation result is obtained by reducing the *Maximum Set Coverage* (MC) problem to MCI. The problem MC is defined as follows: Given a ground set U, a collection $F = \{S_1, S_2, \ldots, S_{|F|}\}$ of subsets of U, and an integer $k' \in \mathbb{N}$, find a sub-collection $F' \subseteq F$ such that $|F'| \leq k'$ and $s(F') = |\cup_{S_i \in F'} S_i|$ is maximized. It is known that the MC problem cannot be approximated within a factor greater than $1 - \frac{1}{e}$, unless $P = NP$ (Feige 1998).

Theorem 1. *There is no polynomial time algorithm with approximation factor greater than* $1 - \frac{1}{e}$ *for the* MCI *problem on undirected graphs, unless* $P = NP$.

Proof. Assume that we have access to a polynomial time approximation algorithm A_{MCI} for the MCI problem with approximation factor $1 - \frac{1}{e} + \epsilon$ for some positive number ϵ. We consider an instance I_{MC} of the MC problem and build the instance I_{MCI} of MCI shown in Fig. 2. The instance consists of 5 vertical layers of nodes. For each element of U of the MC instance, we have a member (filled circle) and Q copies (unfilled circles) in the M-layer to the left. In the F-layer there is a node for each set in the collection F in the MC instance. A node in the F-layer is connected to all the corresponding members and copies in the M-layer. In the third layer there is a single node z connecting all nodes in the F-layer to all nodes in the T-layer to the far right. In the fourth layer we have the node v that is connected to all nodes in the T-layer. All the nodes in the T-layer to the right form a clique. Note that not all the edges are shown in the figure. Let $\beta > 0$ be a sufficiently small positive constant satisfying

$$1 - \frac{1}{e} < \frac{1}{1+\beta}\left(1 - \frac{1}{e} + \epsilon\right) \ .$$

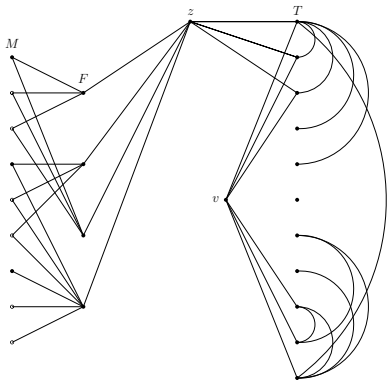

Figure 2: The I_{MCI} instance used in Theorem 1.

To improve the readability, we will not distinguish between a set and its cardinality. E.g., T can represent the set of all the T-nodes and T can also represent the number of T-nodes. Our aim is to choose relatively small Q and T such that

$$(F + M + 1)^2 \leq (\beta(Q+1) - k - F)T \ . \qquad (1)$$

We choose Q and T as follows (note that $M = U(Q+1)$):

$$Q = \left\lceil \frac{1 + k + F}{\beta} - 1 \right\rceil \qquad T = (F + M + 1)^2 \ .$$

Since β is a constant, T and M are polynomial in $|I_{\text{MC}}|$.

Now let S_{MCI} be the solution computed by A_{MCI} given the I_{MCI} instance as input. Let S_{MC} be the solution for MC corresponding to all the sets for which there is an edge between the F-node and v in S_{MCI}. Note that S_{MC} can be computed in polynomial time. Let A, A_1 and A_2 be defined as follows:

$$A = \{(s,t) \in V \times V \mid v \in P_{st}(S_{\text{MCI}})\}$$

$$A_1 = \{(s,t) \in A \mid (s,t) \in (M \times T) \cup (T \times M)\}$$

$$A_2 = A \setminus A_1 \ .$$

We now have the following identity $c_v(S_{\text{MCI}}) = A_1 + A_2$. The set A_1 consists of all the pairs of vertices with one element in M and one element in T that are covered by v in the graph corresponding to S_{MCI}. The contribution to A_1 of the edges in S_{MCI} with one element in F is $2(Q+1)T \cdot s(S_{\text{MC}})$ and the contribution of edges in S_{MCI} with one element in M is no more than $2kT$ and there might be some overlap. This allows us to establish the following upper bound on A_1:

$$A_1 \leq 2(Q+1)T \cdot s(S_{\text{MC}}) + 2kT \ .$$

The remaining pairs that might be covered 1) have an element in F and an element in T, or 2) have no elements in T: $A_2 \leq 2TF + 2(F + M + 1)^2$.

According to (1), Q and T have been chosen such that $kT + TF + (F + M + 1)^2 \leq \beta(Q+1)T$, implying $c_v(S_{\text{MCI}}) \leq 2(1+\beta)(Q+1)T \cdot s(S_{\text{MC}})$. If we add edges to v in the MCI instance corresponding to the optimal solution of the MC instance, we obtain a feasible solution for the MCI instance. For each covered element in the MC instance, we have $2(Q+1)T$ covered pairs in the MCI instance, therefore $2(Q+1)T \cdot OPT(I_{\text{MC}}) \leq OPT(I_{\text{MCI}})$.

The algorithm A_{MCI} has approximation factor $1 - \frac{1}{e} + \epsilon$:

$$\frac{c_v(S_{\text{MCI}})}{OPT(I_{\text{MCI}})} \geq 1 - \frac{1}{e} + \epsilon \ .$$

This allows us to set up the following inequality:

$$\frac{2(1+\beta)(Q+1)T \cdot s(S_{\text{MC}})}{2(Q+1)T \cdot OPT(I_{\text{MC}})} \geq 1 - \frac{1}{e} + \epsilon \ .$$

We can now establish a lower bound for the approximation factor for the solution to our MC instance:

$$\frac{s(S_{\text{MC}})}{OPT(I_{\text{MC}})} \geq \frac{1}{1+\beta}\left(1 - \frac{1}{e} + \epsilon\right) > 1 - \frac{1}{e} \ ,$$

a contradiction. □

Conditional bound

To obtain our next hardness result, we reduce the *Densest-k-Subgraph* (DkS) problem to MCI. DkS is defined as follows: Given a graph G and an integer k, find a subgraph of G induced on k vertices with the maximum number of edges.

Several conditional hardness of approximation results for DkS have been proved (see e.g. (Manurangsi 2017) and references therein). It has been shown that DkS is hard to approximate within any constant bound under the Unique Games with Small Set Expansion conjecture (Raghavendra and Steurer 2010). Recently, it has been shown that under the exponential time hypothesis (ETH) there is no polynomial-time algorithm that approximates DkS to within $n^{-1/(loglogn)^c}$, for some constant c. Moreover, under the stronger Gap-ETH assumption, the factor can be improved to $n^{-f(n)}$ for any function $f \in o(1)$ (Manurangsi 2017). The next theorem shows that there is an S-reduction (Crescenzi 1997) from DkS to MCI only adding a constant to the number of nodes. Then, all the above mentioned inapproximability results extend to the MCI problem. The current state-of-the-art algorithm for DkS guarantees a $\Omega(n^{-\frac{1}{4}-\epsilon})$ approximation (Bhaskara et al. 2010).

Theorem 2. *There is an S-reduction from* DkS *to* MCI. *The reduction transforms a* DkS *instance with n vertices into an* MCI *instance with $n+2$ vertices.*

Proof. Consider a DkS instance given by the graph $G'(V', E')$ and an integer k'. This instance is transformed into an MCI instance given by the graph $G(V, E)$, a node $v \in V$ and an integer k as follows. The set of nodes V is formed by adding a node x and the target node v for the MCI instance to V': $V = \{x, v\} \cup V'$. The node v is isolated and x is connected to all other nodes in V. In addition to the edges incident to x, the edges in the complement of the original graph G' are also added to the graph. Formally, $E = \{\{u, w\} \mid \{u, w\} \notin E'\} \cup \{\{x, u\} \mid u \in V' \cup \{v\}\}$. The value of k is not changed: $k = k'$. See Fig. 3. We also need to explain how to transform a feasible solution of the MCI instance into a feasible solution of the DkS instance. Here we simply pick the nodes that are linking to v in the feasible solution for the MCI instance (excluding x).

We now prove that the reduction is an S-reduction. We claim that the following holds: The node v is on the shortest path between s and t in the MCI instance after adding k edges if and only if 1) edges $\{v, s\}$ and $\{v, t\}$ are added, and 2) there is an edge between s and t in G'. The if-direction is clear. To prove the only-if-direction, we assume that v is on a shortest path between the nodes s and t in the MCI instance. If 1) is false, then the length of the shortest path through v is at least 3. If 2) is false, we also arrive at a contradiction.

This implies that the number of edges induced by a feasible solution of the DkS instance is precisely the coverage centrality of v in the corresponding MCI instance and vice versa. This shows that the reduction is an S-reduction. □

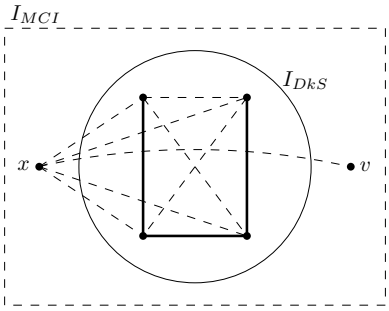

Figure 3: The reduction from DkS. In the circle, we have the original DkS instance (solid edges) that is transformed into the MCI instance indicated by the rectangle (dashed edges).

Approximation algorithm

It is easy to see that, in the undirected case, the objective function is not submodular and that there are instances of MCI (similar to that in Figure 1) for which the greedy algorithm by Nemhauser et al. exhibits an arbitrarily small approximation factor. The main problem with the greedy algorithm is that it does not take into account the shortest paths that pass through two of the added edges. In this section, we show how to overcome this limitation and we give an algorithm that guarantees a $\Omega(1/\sqrt{n})$-approximation.

The algorithm is based on a reduction to a generalization of the maximum coverage problem in which elements of a ground set are covered by pairs of "objects", instead of a single set, and we look for a bounded set of objects that maximizes the number of covered elements. We call this problem the *Maximum Coverage with Pairs problem* (MCP). Formally, MCP is defined as: Given a ground set X, a set O of objects, and an integer $k \in \mathbb{N}$, find a set $O' \subseteq O$, such that $|O'| \leq k$ and $c(O') = |\cup_{i,j \in O'} C(i,j)|$ is maximum, where $C(i,j)$ denotes the subset of X covered by pair $\{i,j\}$, for each unordered pair of objects $\{i,j\}$.

Given $O' \subseteq O$, let $C(O') = \cup_{i,j \in O'} C(i,j)$ and $c(O') = |C(O')|$. Wlog, we assume that each element in X is covered by at least a pair of objects in O and that $k \leq |O|$.

The problem MCI can be reduced to MCP as follows: for each pair (s,t) of nodes in G, we add an element (s,t) to X; for each $u \in V \setminus N(v)$, we add an object $i_u = \{u, v\}$ to O (i.e. all the edges that can be added incident to v); for each pair of objects $i_u, i_w \in O$, we set $C(i_u, i_w) = \{(s,t) \mid v \in P_{st}(\{\{u,v\},\{w,v\}\})\}$; we set $k' = k$. Any

feasible solution O' to the above instance of MCP corresponds to a feasible solution $S = O'$ for MCI. Since for each pair of nodes (s,t) in V the shortest path between s and t in $G(S)$ can only pass through at most two edges of S, then $c_v(S) = |\cup_{i_u,i_w \in O'} C(i_u,i_w)| = c(O')$. Therefore, any approximation algorithm for MCP can be used to solve MCI with the same approximation factor. We observe that MCP is a generalization of the DkS problem, which corresponds to the case in which $|C(i,j)| \leq 1$, for $i,j \in O$, and each element of X is covered by exactly one pair of objects (i.e. objects correspond to nodes and elements correspond to edges). Therefore, MCP is at least as hard to approximate as DkS.

Our algorithm exploits two procedures, called Greedy1 and Greedy2, that return two sets of objects, and selects one of these sets that gives the maximum coverage. In particular, Procedure Greedy1 returns a set that guarantees an approximation factor of $\left(1 - e^{-\frac{(1-\epsilon)(t-1)}{k-1}}\right)$, where $t \in [2,k]$ is a constant integer parameter of the procedure and ϵ is any positive constant, while Procedure Greedy2 guarantees an approximation factor of $(1-\epsilon)\frac{1}{4}\left(1-\frac{1}{e}\right)^2 \frac{k}{|O|}$ for any constant $\epsilon > 0$ (see Theorems 4 and 5). The next theorem shows the overall approximation factor. When applied to the MCI problem, it guarantees a $\Omega(1/\sqrt{n})$-approximation.

Theorem 3. *Let O^* be an optimum solution for* MCP*, let O_1 and O_2 be the solutions of Procedures* Greedy1 *and* Greedy2*, then*

$$\max\{c(O_1), c(O_2)\} \geq (1-\epsilon)\frac{1}{2}\left(1-\frac{1}{e}\right)^{3/2}\sqrt{\frac{t-1}{|O|}}c(O^*),$$

for any constant $t \geq 2$ and $\epsilon \in (0,1)$.

Proof. The value of $\max\{c(O_1), c(O_2)\}$ is at least the geometric mean of $c(O_1)$ and $c(O_2)$. Moreover, $\left(1 - e^{-\frac{(1-\epsilon)(t-1)}{k-1}}\right) \geq \left(1-\frac{1}{e}\right)\frac{(1-\epsilon)(t-1)}{k-1}$, for any $\epsilon \in (0,1)$ and $k > 1$[2]. Therefore,

$$\begin{aligned}\max\{c(O_1), c(O_2)\} &\geq \sqrt{c(O_1)\cdot c(O_2)} \\ &\geq \sqrt{\left(1-\frac{1}{e}\right)\frac{(1-\epsilon)(t-1)}{k-1}c(O^*)} \\ &\quad \cdot \sqrt{(1-\epsilon)\frac{1}{4}\left(1-\frac{1}{e}\right)^2\frac{k}{|O|}c(O^*)} \\ &\geq (1-\epsilon)\frac{1}{2}\left(1-\frac{1}{e}\right)^{3/2}\sqrt{\frac{t-1}{|O|}}c(O^*). \qquad \square\end{aligned}$$

We now introduce procedures Greedy1 and Greedy2.

Procedure Greedy1

The pseudo-code of Procedure Greedy1 is reported in Algorithm 1. For some fixed constant integer $t \in [2,k]$, the procedure greedily selects a set of objects of size t that maximizes the increment in the objective function. In particular, it starts with an empty solution and iteratively adds to it a set Z of t objects that maximizes $c(O' \cup Z) - c(O')$, where O' is the solution computed so far. The procedure stops when it has added at least $k-t$ objects to S. Eventually, if $|O'| < k$, it completes the solution by adding a further set of $k - |O'| < t$ objects (lines 5–6). Note that one or more objects of the selected set might already belong to O' (but not all of them). Hence, Algorithm 1 has at least $\left\lfloor \frac{k-t}{t} \right\rfloor$ and at most k iterations. For each iteration i of Algorithm 1, let O_i be the set O' at the end of iteration i and let O^* be an optimal solution. The next lemma is used to prove the approximation bound, the full proof can be found in (D'Angelo, Olsen, and Severini 2018).

Algorithm 1: Procedure Greedy1

```
1 O' := ∅;
2 while |O'| ≤ k − t do
3 |   Z := arg max_{Z⊆O,|Z|≤t} {C(O' ∪ Z) − C(O')};
4 |   O' := O' ∪ Z;
5 Z := arg max_{Z⊆O,|Z|≤k−|O'|} {C(O' ∪ Z) − C(O')};
6 O' := O' ∪ Z;
```

Lemma 1. *After each iteration i of Algorithm 1, the following holds*

$$c(O_i) \geq \left(1 - \left(1 - \frac{t(t-1)}{k(k-1)}\right)^i\right)c(O^*). \qquad (2)$$

Theorem 4. *If I is the number of iterations of Algorithm 1, then*

$$c(O_I) \geq \left(1 - e^{-(1-\epsilon)\frac{t-1}{k-1}}\right)c(O^*) \ ,$$

for any constant $\epsilon \in (0,1)$.

Proof. We observe that for any constant $\epsilon > 0$, there exists a constant k_0, such that for each $k \geq k_0$, $I = \left\lfloor \frac{k-t}{t} \right\rfloor \geq \frac{k}{t} - 2 \geq (1-\epsilon)\frac{k}{t}$. Note that, when k is a constant the problem can be easily solvable in polynomial time by brute force and therefore we assume that k is not a constant. Plugging I into inequality (2), we obtain:

$$\begin{aligned}c(O_I) &\geq \left(1 - \left(1 - \frac{t(t-1)}{k(k-1)}\right)^{\left\lfloor \frac{k-t}{t} \right\rfloor}\right)c(O^*). \\ &\geq \left(1 - \left(1 - \frac{t(t-1)}{k(k-1)}\right)^{(1-\epsilon)\frac{k}{t}}\right)c(O^*).\end{aligned}$$

By calculus, it can be shown that $1-x \leq e^{-x}$, which implies that $\left(1 - \frac{t(t-1)}{k(k-1)}\right)^{(1-\epsilon)\frac{k}{t}} \leq e^{-(1-\epsilon)\frac{t-1}{k-1}}$, and finally:

$$\begin{aligned}&\left(1 - \left(1 - \frac{t(t-1)}{k(k-1)}\right)^{(1-\epsilon)\frac{k}{t}}\right)c(O^*) \\ &\qquad \geq \left(1 - e^{-(1-\epsilon)\frac{t-1}{k-1}}\right)c(O^*). \qquad \square\end{aligned}$$

[2]Indeed, $1 - e^{-x} \geq (1 - e^{-1})x$, for any $x \in [0,1]$, and $\frac{(1-\epsilon)(t-1)}{k-1} \in [0,1]$ for any $\epsilon \in (0,1)$, $t \leq k$, and $k > 1$.

Algorithm 2: Procedure GREEDY2

1 Define an instance of MC made of ground set X and, for each object $o \in O$, a set equal to $N(o)$;
2 Run the greedy algorithm in (Nemhauser, Wolsey, and Fisher 1978) for MC to find a set H of size $\lceil \frac{k}{2} \rceil$ objects;
3 Define an instance of MC made of ground set $D(H) = N(H) \setminus C(H)$ and, for each object $o \in O \setminus H$, a set equal to $\cup_{i \in H} C(o,i) \setminus C(H)$;
4 Run the greedy algorithm in (Nemhauser, Wolsey, and Fisher 1978) for MC to find a set I of size $\lfloor \frac{k}{2} \rfloor$ objects;
5 **return** $H \cup I$;

Procedure GREEDY2

In order to describe Procedure GREEDY2, we need to introduce further notation. Given a set O' of objects, we denote by $N(O')$ the set of elements that the objects in O' can cover when associated with any other object in O, that is $N(O') = \cup_{o \in O', i \in O} C(o,i)$. The *degree* of O' is the cardinality of $N(O')$ and it is denoted by $d(O')$. To simplify the notation, when O' is a singleton, $O' = \{o\}$, we use $N(o)$ and $d(o)$ to denote $N(O)$ and $d(O)$, respectively. Intuitively, $N(O')$ are the elements that are covered by at least an object in O', while $C(O')$ are the elements that are covered by at least two objects in O'. In the following, we say that an element is *single* covered by O' in the former case and *double* covered by O' in the latter case. We observe that $d(O') \geq c(O')$, for any set of objects O'. Moreover, since the degree is defined as the size of the union of sets, then it is a monotone and submodular set function.

Procedure GREEDY2 is given in Algorithm 2. First, the procedure looks for a set H of $\lceil \frac{k}{2} \rceil$ objects with maximum degree. Since computing such a set is equivalent to solving an instance of MC, which is known to be NP-hard, we compute an approximation of it. In detail, the instance of MC is made of the same ground set X and, for each object $o \in O$, a set equal to $N(o)$. Any set of objects O' corresponds to a solution to this MC instance, where the number of single covered elements is equal to $d(O')$. Indeed, finding a set of $\lceil \frac{k}{2} \rceil$ objects that maximizes the degree in the MCP instance corresponds to finding a collection of sets that maximizes the single coverage of X in this MC instance. Hence, we find a set H that approximates the maximum single coverage of X, and, in particular, we exploit a greedy algorithm that guarantees an optimal approximation of $1 - 1/e$ (Nemhauser, Wolsey, and Fisher 1978). Then, the procedure selects a set of $\lfloor \frac{k}{2} \rfloor$ objects in $O \setminus H$ that maximizes the single coverage of the elements in $N(H)$ not double covered by H. In other words, these objects, *along with* objects in H, double cover the maximum fraction of $N(H)$. Again, computing such a set is equivalent to solving an instance of MC, and we find a $(1 - 1/e)$-approximation. In this case, the MC instance is made of the ground set $D(H) = N(H) \setminus C(H)$, and for each object $o \in O \setminus H$, a set equal to $\cup_{i \in H} C(o,i) \setminus C(H)$. The approximated solution found by the greedy algorithm for MC is denoted by I. Procedure GREEDY2 outputs the set of objects $H \cup I$. In the next lemma we establish a connection between the number of single and double covered elements, in particular, we show an upper bound to the optimal value $c(O^*)$ of the MCP instance as a function of $d(H)$, where H is the set of objects selected at line 2 of Algorithm 2. The full proof of the lemma can be found in (D'Angelo, Olsen, and Severini 2018).

Lemma 2. *If H is the set of objects selected at line 2 of Algorithm 2 and O^* is an optimal solution for* MCP*, then*

$$d(H) \geq \frac{1}{2}\left(1 - \frac{1}{e}\right) c(O^*).$$

Theorem 5. *If $H \cup I$ is the output of Algorithm 2 and O^* is an optimal solution for* MCP*, then*

$$c(H \cup I) \geq (1-\epsilon)\frac{1}{4}\left(1 - \frac{1}{e}\right)^2 \frac{k}{|O|} c(O^*),$$

for any constant $\epsilon \in (0,1)$.

Proof. The objects in H double cover $c(H)$ elements, hence, the number of elements that are single covered by H but not double covered by H is $d(H) - c(H)$. The set of these elements is $D(H) = N(H) \setminus C(H)$. We now show that $N(I)$ contains at least a fraction $\left(1 - \frac{1}{e}\right) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$ of these elements and hence the objects in $H \cup I$ double cover at least $(d(H) - c(H)) \left(1 - \frac{1}{e}\right) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$ of them.

Let us denote as I^* a set of $\lfloor \frac{k}{2} \rfloor$ objects in $O \setminus H$ that maximizes the single coverage of elements in $D(H)$, that is the size of $D(H) \cap N(I^*)$ is maximum for sets of $\lfloor \frac{k}{2} \rfloor$ objects. By contradiction, let us assume that the size of $D(H) \cap N(I^*)$ is smaller than $(d(H) - c(H)) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$.

Let us partition $O \setminus H$ into sets of $\lfloor \frac{k}{2} \rfloor$ objects plus a possible set of smaller size, if $|O \setminus H|$ is not divisible by $\lfloor \frac{k}{2} \rfloor$. The number of the sets in the partition is

$$\ell = \left\lceil \frac{|O \setminus H|}{\lfloor \frac{k}{2} \rfloor} \right\rceil = \left\lceil \frac{|O| - \lceil \frac{k}{2} \rceil}{\lfloor \frac{k}{2} \rfloor} \right\rceil \leq \frac{|O|}{\lfloor \frac{k}{2} \rfloor}.$$

We denote the sets of this partition as I_i, $i = 1, 2, \ldots, \ell$. Since I^* maximizes the single coverage of elements in $D(H)$, then for each I_i, the size of $D(H) \cap N(I_i)$ is smaller than $(d(H) - c(H)) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$. By submodularity we have

$$|D(H) \cap N(O \setminus H)| \leq \sum_{i=1}^{\ell} |D(H) \cap N(I_i)|$$

$$< \sum_{i=1}^{\ell} (d(H) - c(H)) \left\lfloor \frac{k}{2} \right\rfloor \frac{1}{|O|} \leq d(H) - c(H),$$

which is a contradiction because it implies that there are elements in $D(H)$ that are not covered by any object in $O \setminus H$ and hence they cannot be double covered. This proves that the size of $D(H) \cap N(I^*)$ is at least $(d(H) - c(H)) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$. Moreover, set I approximates the optimal single coverage of $D(H)$ by a factor $1 - \frac{1}{e}$ and hence the size of $D(H) \cap N(I)$ is at least $\left(1 - \frac{1}{e}\right) (d(H) - c(H)) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|}$.

It follows that the overall number of elements double covered by $H \cup I$ is at least $(d(H) - c(H)) \left(1 - \frac{1}{e}\right) \lfloor \frac{k}{2} \rfloor \frac{1}{|O|} + c(H)$. By Lemma 2, this values is at least

Network	$n = \|V\|$	$m = \|E\|$
BA	50	96
CM	50	85
karate	34	78
windsurfers	43	336
jazz	198	2742
haggle	274	2899

Table 2: Undirected networks used in the experiments.

$$\left(\frac{1}{2}\left(1-\frac{1}{e}\right)c(O^*)-c(H)\right)\left(1-\frac{1}{e}\right)\left\lfloor\frac{k}{2}\right\rfloor\frac{1}{|O|}+c(H).$$

For any constant $\epsilon > 0$ and k greater than a constant, $\left\lfloor\frac{k}{2}\right\rfloor \geq (1-\epsilon)\frac{k}{2}$, then this number is at least

$$(1-\epsilon)\left(\frac{1}{2}\left(1-\frac{1}{e}\right)c(O^*)-c(H)\right)\left(1-\frac{1}{e}\right)\frac{k}{2|O|} + c(H)$$

$$= (1-\epsilon)\frac{k}{4|O|}\left(1-\frac{1}{e}\right)^2 c(O^*) + c(H)\left(1-(1-\epsilon)\left(1-\frac{1}{e}\right)\frac{k}{2|O|}\right)$$

$$\geq (1-\epsilon)\frac{k}{4|O|}\left(1-\frac{1}{e}\right)^2 c(O^*),$$

since $c(H) \geq 0$ and $k \leq 2|O|$. □

Experimental study

In this section, we study the algorithms GREEDY1 and GREEDY2 from an experimental point of view. First, we compare the solutions of the greedy algorithms with optimal solutions computed by using an integer program formulation of MCP in order to assess the real performance in terms of solution quality (see (D'Angelo, Olsen, and Severini 2018) for the detailed implementation of the IP formulation). Then, we focus on the MCI problem and compare GREEDY1 and GREEDY2 with the natural algorithm that adds k random edges. We execute our experiments on two popular model networks, the Barabasi-Albert (BA) network (Barabasi and Albert 1999) and the Configuration Model (CM) network (Bender and Canfield 1978; Molloy and Reed 1995), and on real-world networks extracted from human activities[3]. The sizes of the networks are reported in Table 2. All our experiments have been performed on a computer equipped with an Intel Xeon E5-2643 CPU clocked at 3.4GHz and 128GB of main memory, and our programs have been implemented in C++. The results of the comparison with the optimum are reported in Figure 4. For each network, we randomly choose 10 target nodes and, for each target node v, we add k nonexistent edges incident to v for $k = 1, 2, \ldots, 10$. Then, we plot the average coverage centrality of the 10 target nodes for each k. We observe that there is little difference between the solutions of GREEDY1 and GREEDY2 and the optimal solutions, since the approximation ratio of GREEDY1 is always greater than 0.97 while the approximation ratio of GREEDY2 is always greater than 0.78. It is not possible to find the optimum on networks with thousand of edges in a reasonable time. Therefore, we compare the solutions with the natural baseline of adding k random edges incident to each target node (the RANDOM algorithm). Analogously to the previous case, we plot the average coverage centrality of the 10 target nodes for each k. The results are reported in Figure 5. We notice that also in this case GREEDY1 provides a better solution than GREEDY2. However, both algorithms perform always better than the RANDOM algorithm. On jazz, GREEDY1 with $t = 2$ needs 7.5 seconds to solve the problem for $k = 10$, GREEDY1 with $t = 3$ needs 242 seconds while and GREEDY2 requires only 4.1 seconds. Notice that GREEDY2 exhibits a better scalability than GREEDY1 as k increases since it requires the same time also for k greater than 10.

[3]http://konect.uni-koblenz.de/

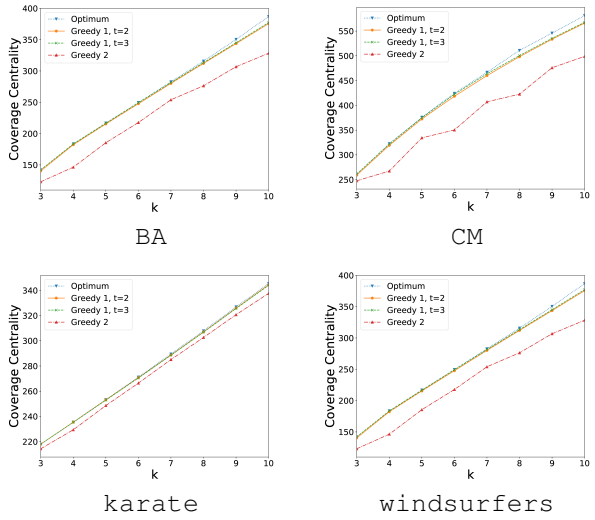

Figure 4: Average coverage centrality of target nodes as a function of the number k of inserted edges for GREEDY1 (with $t = 2, 3$), GREEDY2, and optimal solutions.

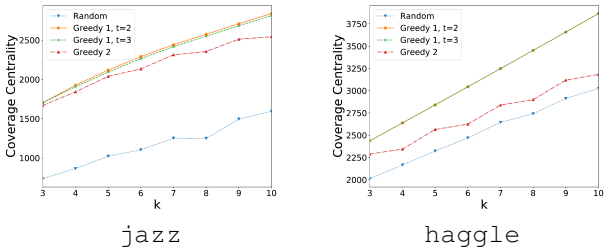

Figure 5: Average coverage centrality of target nodes as a function of the number k of inserted edges for GREEDY1 ($t = 2, 3$), GREEDY2, and RANDOM on jazz and haggle.

References

Avrachenkov, K., and Litvak, N. 2006. The effect of new links on google pagerank. *Stoc. Models* 22(2):319–331.

Barabasi, A.-L., and Albert, R. 1999. Emergence of scaling in random networks. *Science* 286(5439):509–512.

Bender, E. A., and Canfield, E. R. 1978. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A* 24(3):296–307.

Bergamini, E.; Crescenzi, P.; D'Angelo, G.; Meyerhenke, H.; Severini, L.; and Velaj, Y. 2018. Improving the betweenness centrality of a node by adding links. *ACM Journal of Experimental Algorithmics* 23:1.5:1–1.5:32.

Bhaskara, A.; Charikar, M.; Chlamtac, E.; Feige, U.; and Vijayaraghavan, A. 2010. 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, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, 201–210.

Crescenzi, P.; D'Angelo, G.; Severini, L.; and Velaj, Y. 2016. Greedily improving our own closeness centrality in a network. *ACM Transactions on Knowledge Discovery from Data* 11(1):9:1–9:32.

Crescenzi, P. 1997. A short guide to approximation preserving reductions. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, CCC '97, 262–273. IEEE Computer Society.

D'Angelo, G.; Olsen, M.; and Severini, L. 2018. Coverage Centrality Maximization in Undirected Networks. *CoRR* abs/1811.04331.

D'Angelo, G.; Severini, L.; and Velaj, Y. 2016. On the maximum betweenness improvement problem. In *Proceedings of the 16th Italian Conference on Theoretical Computer Science (ICTCS15)*, volume 322 of *Electr. Notes Theor. Comput. Sci.*, 153–168.

Demaine, E. D., and Zadimoghaddam, M. 2010. Minimizing the diameter of a network using shortcut edges. In *Proc. of the 12th Scandinavian Symp. and Work. on Algorithm Theory (SWAT)*, volume 6139 of *Lecture Notes in Computer Science*, 420–431. Springer.

Feige, U. 1998. A threshold of ln n for approximating set cover. *Journal of the ACM* 45(4).

Ishakian, V.; Erdös, D.; Terzi, E.; and Bestavros, A. 2012. A framework for the evaluation and management of network centrality. In *Proc. of the 12th SIAM Int. Conf. on Data Mining (SDM)*, 427–438. SIAM.

Macdonald, B.; Shakarian, P.; Howard, N.; and Moores, G. 2012. Spreaders in the network SIR model: An empirical study. *CoRR* abs/1208.4269.

Malighetti, P.; Martini, G.; Paleari, S.; and Redondi, R. 2009. The impacts of airport centrality in the EU network and inter-airport competition on airport efficiency. Technical Report MPRA-7673.

Manurangsi, P. 2017. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, 954–961. ACM.

Medya, S.; Silva, A.; Singh, A. K.; Basu, P.; and Swami, A. 2018. Group centrality maximization via network design. In *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018*, 126–134. SIAM.

Molloy, M., and Reed, B. 1995. A critical point for random graphs with a given degree sequence. *Random structures & algorithms* 6(2-3):161–180.

Nemhauser, G.; Wolsey, L.; and Fisher, M. 1978. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming* 14(1):265–294.

Newman, M. 2010. *Networks: An Introduction*. Oxford University Press, Inc.

Olsen, M., and Viglas, A. 2014. On the approximability of the link building problem. *Theor. Comput. Sci.* 518:96–116.

Perumal, S.; Basu, P.; and Guan, Z. 2013. Minimizing eccentricity in composite networks via constrained edge additions. In *Military Communications Conference, MILCOM 2013 IEEE*, 1894–1899.

Raghavendra, P., and Steurer, D. 2010. Graph expansion and the unique games conjecture. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC 2010, 755–764. ACM.

Shan, L.; Yi, Y.; and Zhang, Z. 2018. Improving information centrality of a node in complex networks by adding edges. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 3535–3541. International Joint Conferences on Artificial Intelligence Organization.

Yoshida, Y. 2014. Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD14, 1416–1425. ACM.