Maximum Coverage and Maximum Connected Covering in Social Networks with Partial Topology Information

Patricio Reyes
Dept. Statistics
Universidad Carlos III de Madrid (UC3M)
Calle Madrid 126, UC3M - Estadística
28903 Getafe, Madrid, Spain
preyes@est-econ.uc3m.es

Alonso Silva
Alcatel-Lucent Bell Labs France
Centre de Villarceaux
Route de Villejust
91620 Nozay, France
alonso.silva@alcatel-lucent.com

***Abstract*—Viral marketing campaigns seek to recruit the most influential individuals to cover the largest target audience. This can be modeled as the well-studied maximum coverage problem. Another related problem, called the maximum connected cover, is when recruited nodes have to be connected. This problem ensures a strong coordination among the influential nodes which are the backbone of the marketing campaign. In this work, we are interested on both of these problems. Most of the related literature assumes knowledge about the topology of the network. Even in that case, the problem is known to be NP-hard. We analyse heuristics to these models assuming different knowledge levels about the topology of the network. We quantify the difference between these heuristics and the local and global greedy algorithms.**

I. Introduction

One of the main objectives of viral marketing campaigns is to find the most influential individuals to cover the largest target audience. This problem can be modeled as the well-studied maximum coverage problem. In the need of coordination through the marketing campaign, a more relevant objective is to seek the most influential connected individuals. Hereby the connectedness will be fundamental since the advertisement needs to spread quickly through the network. In this work, we are interested on both of these problems. Most of the related works on these topics assume knowledge about the topology of the network. Even in that case, the problem is known to be NP-hard. Recently, in [1] the authors propose a (local) greedy algorithm to the maximum connected covering problem by learning the topology of the network on-the-fly.

In this work, we present different heuristics to both of these problems with different levels of knowledge about the topology of the network. We quantify the difference between these algorithms. Obviously, different knowledge about the topology of the network will restrict us to use different heuristics with the problem at hand.

Works providing heuristics to maximize the impact of a virus marketing campaign are [2] and [3]. Other works have been interested on the spreading of information through cascades on a weighted influence graph and some internal conviction threshold of the individuals (see e.g. [4], [5], [6]). Their work is based on the submodularity of the local spreading and the bounds on the performance of the greedy algorithms for submodular functions given in [7]. Our work is different since we are not interested on the spreading of information through cascades, but on a “quick” spreading: we are interested on a one-hop spread of information only through neighbors. Closely related works are [1] and [8], where the authors assume a one-hop lookahead [1] and two-hops lookaheads [8].

The paper is structured as follows. In Section II, we formulate both, the maximum coverage problem and the maximum connected covering problem. In Section III, we describe the different levels of knowledge about the topology of the network that we consider in this work. In Section IV, we present existing heuristics and we present some new heuristics to these problems based on the different levels of knowledge of the network. In Section V, we present the simulations of the algorithms and finally we conclude in Section VI.

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

II. Problem Formulation

We consider an influence graph $G = (V, E)$ where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Each vertex of the graph represents an individual and each edge represents a relationship of mutual influence between them (e.g. friendship over a social network). An individual $i \in V$ has influence over another individual $j \in V$ if and only if $\{i, j\} \in E$. We assume that the influence graph G is an undirected graph with no self-loops. We denote by $\mathcal{N}(i)$ the set of neighbors of vertex i, i.e., $\mathcal{N}(i) = \{j \in V : \{i, j\} \in E\}$, and for a set of vertices $A \subseteq V$, we denote by $\mathcal{N}(A)$ the set of neighbors of A as $\mathcal{N}(A) = \{j \in V \setminus A : \text{exists } i \in A \text{ such that } \{i, j\} \in E\}$.

We consider that time is slotted, i.e., $t \in \mathbb{N} \cup \{0\}$. We denote by $\mathcal{R}(t)$ the set of recruited individuals at time $t \geq 0$. We will also call the observed set of nodes to $\mathcal{N}(\mathcal{R}(t))$, the set containing unrecruited neighbors of recruited nodes (see [1]).

The algorithms that we present are sequential algorithms which proceed as follows: at time t, with $0 \leq t \leq K$, the algorithm recruits a node $i \in V \setminus \mathcal{R}(t-1)$ and performs the update $\mathcal{R}(t) = \mathcal{R}(t-1) \cup \{i\}$.

The objective of the maximum coverage algorithms is to maximize the size of the network covering $\mathcal{C}(t) = \mathcal{R}(t) \cup \mathcal{N}(\mathcal{R}(t))$ and in the case of the maximum connected covering (MCC) problem this objective is subject to the additional constraint that the set $\mathcal{R}(t)$ must be connected. In this work, we are interested in convering all the network with the minimum number of recruited nodes. This is known as the minimum set cover problem (SCP). Even though there is a vast literature for the SCP, we find interesting to compare the performance of the algorithms for SCP and MCC.

The degree $d(i)$ of a node $i \in V$ is the number of neighbors of a node, i.e., $d(i) = |\mathcal{N}(i)|$ where $|\cdot|$ is the cardinality function. The observed degree $d_{obs}(i, t)$ of a node $i \in V$ at time t is the number of neighbors of i which are either recruited or observed, i.e., $d_{obs}(i, t) = |\{j \in \mathcal{R}(t) \cup \mathcal{N}(\mathcal{R}(t)) : \{i, j\} \in E\}|$. The excess degree $d_{excess}(i, t)$ of a node $i \in V$ at time t is difference between the degree and the observed degree of node i at time t, i.e., $d_{excess}(i, t) = d(i) - d_{obs}(i, t)$.

III. Information Levels

For both of the problems we are dealing with in this work, we consider different levels of information about the topology of the network.

1) List of nodes: we consider that the recruiter knows the list of nodes (the set V) so there is a knowledge about the nodes the network has and there is a possibility to recruit any node within the network. Once a node has been recruited we consider that the recruited node gives information about who are its neighbors.
2) One-hop lookahead: we consider that the recruiter knows only one node and once a node is recruited it gives information about who are its neighbors and who are their mutual neighbors (between recruited nodes). Actually, the recruiter may only need to know the quantity of neighbors, observed neighbors and mutual neighbors (between recruited nodes), in order to compute the excess degree (See the definition of 1^+-local algorithm in [9]).
3) Two-hops lookahead: we consider that the recruiter knows only one node and once a node is recruited it gives information about who are its neighbors and neighbors of neighbors and who are their neighbors and the mutual neighbors.
4) List of nodes and two-hops lookahead: The recruiter has knowledge about the list of nodes as in 1) and two-hops lookahead as in 3).
5) Full knowledge: we consider that the recruiter has full knowledge about the topology of the network. It knows the set of nodes V and the set of edges E.

We notice that in [1] the authors consider the knowledge level as in 2) since in their case, they do not have any information about the network topology and they are discovering the network while they are recruiting over the network.

IV. Algorithms

In this section, we give a brief description of the algorithms for the different scenarios (levels of information) in both problems: the set cover problem (SCP) and maximum connected covering (MCC) problem.

A. Set Cover Problem (SCP)

In the first scenario, called SCP 1, we consider that the recruiter knows the list of nodes but doesn't have any information about the topology of the graph as in III 1). Once a node is recruited and only then, we consider that the node gives the information about which nodes it is connected to. Under these characteristics, we consider Algorithm 1. Given that initially the recruiter doesn't have any information about the topology of the network, Algorithm 1 simply chooses a node at random and since then it knows to which nodes it is connected to, then it removes those nodes (since they are already covered) from the uncovered list of nodes and then again it chooses a node from within the set of remaining uncovered nodes.

For the probability distribution over a set of nodes $S \subseteq V$, we identify each node $i \in S$ with a unique integer from 1 to $|S|$. We consider a probability distribution ζ over the set of nodes $|S|$, i.e., $\zeta(i) \geq 0$ and $\sum_{i \in S} \zeta(i) = 1$. For simplicity, we consider the two following cases:

- The uniform distribution $\zeta_1(i) = 1/|S|$,
- The degree distribution $\zeta_2(i) = d(i)/\sum_{j \in S} d(j)$.

However, we notice that the probability distribution ζ is not restricted to these two choices.

Algorithm 1 SCP 1: Random

Initialize the list of uncovered nodes U with the set of all nodes $U \leftarrow V$, the list of recruited nodes R with the empty set $R \leftarrow \emptyset$, and the list of covered nodes C with the empty set $C \leftarrow \emptyset$,
$k \leftarrow 1$,
repeat
  Recruit a node $i \in U$ uniformly at random, i.e., $R \leftarrow R \cup \{i\}$,
  Remove node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e.,$U \leftarrow U \setminus (i \cup \mathcal{N}(i))$
  Add node i and its neighbors $\mathcal{N}(i)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (i \cup \mathcal{N}(i))$
  $k \leftarrow k + 1$,
until $k > K$ or $U \leftarrow \emptyset$

In the second scenario, called SCP 2, we assume that when a node is recruited it provides a two-hops lookahead information, i.e., it gives information about its neighbors and the neighbors of its neighbors as in III 3). To take advantage of this knowledge, Algorithm 2 which was originally proposed by Guha and Khuller [8], proposes to recruit the node from within a two-hop neighborhood that has the maximum number of uncovered neighbors (maximize the excess degree), and then again to choose the node from within a two-hop neighborhood of the set of recruited nodes that has the maximum number of uncovered neighbors.

In the third scenario, called SCP 3, we consider that the recruiter knows the list of nodes (as in the first scenario) and that when a node is recruited it provides a two-hop lookahead information (as in the second scenario). To take advantage of this knowledge, we propose Algorithm 3 that at every step with probability α recruits a node at random from within the set of uncovered nodes

Algorithm 2 SCP 2: Two-hops Greedy Algorithm[8]

Initialize the list of uncovered nodes U with the set of all nodes $U \leftarrow V$, the list of recruited nodes R with the empty set $R \leftarrow \emptyset$, and the list of covered nodes C with the empty set $C \leftarrow \emptyset$,
$k \leftarrow 1$,
repeat
  Recruit a node $i \in U \cap [\mathcal{N}(R) \cup \mathcal{N}(\mathcal{N}(R))]$ of maximum excess degree, i.e., $R \leftarrow R \cup \{i\}$ where i is such that $|\mathcal{N}(i) \setminus (R \cup \mathcal{N}(R))|$ is maximum restricted to the set $U \cap [\mathcal{N}(R) \cup \mathcal{N}(\mathcal{N}(R))]$,
  Remove node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (i \cup \mathcal{N}(i))$,
  Add node i and its neighbors $\mathcal{N}(i)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (i \cup \mathcal{N}(i))$
  $k \leftarrow k + 1$
until $k > K$ or $U \leftarrow \emptyset$

and with probability $(1-\alpha)$ recruits the node from within a two-hop neighborhood that has the maximum number of uncovered neighbors (See section V for details about the actual version considered in the simulations). It is clear that the appeal from this version of the algorithm is that it is a probabilistic combination from both previous scenarios. This type of algorithms has been discussed in [9] as a solution to address the greedy issues.

For the probability distribution over a set of nodes $S \subseteq V$, we consider ζ as in the first scenario. We consider α to be a variable to be chosen $0 \leq \alpha \leq 1$. According to [9], we decided to set $\alpha = 0.5$.

The fourth scenario is the full knowledge scenario as in III 5) where the recruiter knows the topology of the network (the list of nodes, the list of neighbors of the nodes, the list of neighbors of the neighbors of the nodes, etc).

B. Maximum Connected Coverage (MCC) problem

Now, we present the algorithms for the maximum connected coverage problem (MCC). In this problem (unlike SCP) the resulting set of recruited nodes must be connected. In the first scenario, called MCC 1, we consider that the recruiter knows a node, denoted node $i \in V$, and then, when a node is recruited, it gives a one-hop lookahead as in III 2). In Algorithm 4, we propose a random selection over the observed set of nodes (the neighbors of the recruited nodes which are not themselves already recruited) i.e., $P = \mathcal{N}(R) \setminus R$. We notice that this scenario is different from a random

Algorithm 3 SCP 3: THG + Random α

Initialize the list of uncovered nodes U with the set of all nodes $U \leftarrow V$, the list of recruited nodes R with the empty set $R \leftarrow \emptyset$, and the list of covered nodes C with the empty set $C \leftarrow \emptyset$,
$k \leftarrow 1$,
repeat
  Draw a Bernoulli random variable X with parameter α
  if $X = 1$ **then**
    Recruit a node $j \in U$ at random (according to ζ) from the set U, i.e., $R \leftarrow R \cup \{j\}$
    Remove node j and its neighbors $\mathcal{N}(j)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (j \cup \mathcal{N}(j))$
    Add node j and its neighbors $\mathcal{N}(j)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (j \cup \mathcal{N}(j))$
  else
    Recruit a node $i \in U \cap [\mathcal{N}(R) \cup \mathcal{N}(\mathcal{N}(R))]$ of maximum excess degree, i.e., $R \leftarrow R \cup \{i\}$ where i is such that $|\mathcal{N}(i) \setminus (R \cup \mathcal{N}(R))|$ is maximum restricted to the set $U \cap [\mathcal{N}(R) \cup \mathcal{N}(\mathcal{N}(R))]$,
    Remove node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (i \cup \mathcal{N}(i))$
    Add node i and its neighbors $\mathcal{N}(i)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (i \cup \mathcal{N}(i))$
  end if
  $k \leftarrow k + 1$
until $k > K$ or $U \leftarrow \emptyset$

Algorithm 4 MCC 1: Random Neighbor

Initialize the list of uncovered nodes U with the set of all nodes $U \leftarrow V$, the list of recruited nodes R with the empty set $R \leftarrow \emptyset$, and the list of covered nodes R with the empty set $R \leftarrow \emptyset$,
Recruit a node $i \in U$ at random (according to ζ), i.e., $R \leftarrow R \cup \{i\}$,
Remove node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (i \cup \mathcal{N}(i))$,
Add node i and its neighbors $\mathcal{N}(i)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (i \cup \mathcal{N}(i))$,
Initialize the list of candidates to be recruited with the set of neighbors of i, i.e., $P \leftarrow \mathcal{N}(i)$,
$k \leftarrow 2$
repeat
  Recruit a node $j \in P$ uniformly at random from the set P, i.e., $R \leftarrow R \cup \{j\}$ with $j \in P$,
  Remove node j from the list of candidates to be recruited, i.e., $P \leftarrow P \setminus \{j\}$,
  Remove the node j and its neighbors $\mathcal{N}(j)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (j \cup \mathcal{N}(j))$,
  Add node j and its neighbors $\mathcal{N}(j)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (j \cup \mathcal{N}(j))$,
  Add the unrecruited neighbors of j to the list of candidates to be recruited, i.e., $P \leftarrow P \cup (\mathcal{N}(j) \cap U)$,
  $k \leftarrow k + 1$
until $k > K$ or $U \leftarrow \emptyset$

walk since we are choosing among the whole set P and not only the neighbors of the newly recruited node.

In the second scenario, called MCC 2, we also consider that the recruiter knows a node, denoted node $i \in V$, and then, when a node is recruited, it gives the list of neighbors of the recruited nodes. In Algorithm 5, which was originally proposed by [1], the algorithm greedily recrutes the node in P which maximizes the excess degree.

V. Simulations

We performed simulations of the previously described algorithms in Erdös-Rényi graphs $G(N, p_N)$ where $N \in \{50, 100, 150, 200, 250\}$ is the number of nodes in the graph and p_N is the probability of two nodes being connected. We chose $p_N = 2\ln(N)/N$ to ensure connectivity. We generate 10 graphs per each size and model (Erdös-Rényi or Barabasi-Albert). Moreover, each one of the algorithms is run 3 time on each graph, starting from 3 different nodes (the first recruited node). Therefore, the figures show the mean of the number of recruited nodes (over all the graph instances with same size) needed to cover the whole graph. The bands in the figures correspond to the confidence interval at 95%. We notice that this corresponds to the case when there is no restriction over K but only on the number of uncovered nodes.

For the case of algortihm 3, we decided to use the deterministic version for $\alpha = 0.5$ proposed in [9]: The algorithm alternates between recruiting at random and recruiting via excess-degree.

It is difficult to compare different knowledge levels and levels of randomness. The first observation we make from Figure 1 is that recruiting nodes at random SCP 1 performs roughly 40% worse than a two-hops lookahead and a greedy algorithm SCP 3

Algorithm 5 MCC 2: Online Myopic MCC [1]

Initialize the list of uncovered nodes U with the set of all nodes $U \leftarrow V$, the list of recruited nodes R with the empty set $R \leftarrow \emptyset$, and the list of covered nodes R with the empty set $R \leftarrow \emptyset$,
Recruit node $i \in U$, i.e., $R \leftarrow R \cup \{i\}$,
Remove node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e., $U \leftarrow U \setminus (i \cup \mathcal{N}(i))$,
Add node i and its neighbors $\mathcal{N}(i)$ to the list of covered nodes, i.e., $C \leftarrow C \cup (i \cup \mathcal{N}(i))$,
$k \leftarrow 2$
repeat
  Recruit a node $i \in U$ that maximizes the excess degree, i.e., $R \leftarrow R \cup \{i\}$, where $i \in U$ is such that $|\mathcal{N}(i) \setminus (R \cup \mathcal{N}(R))|$ is maximum,
  Activate a node $i \in U$ that maximizes the excess degree, i.e., $R = R \cup \{i\}$, where $i \in U$ is such that $|\mathcal{N}(R) \cap \mathcal{N}(i)|$ is maximum,
  Activate one of the nodes $i \in U$ of maximum excess degree, i.e., $R = R \cup \{i\}$ where i is such that $d_i - d_i^{obs} = \max_{k \in \{1,\ldots,n\}} d_k - d_k^{obs}$ where d^{obs} is the observed degree.
  Remove the node i and its neighbors $\mathcal{N}(i)$ from the list of uncovered nodes, i.e., $U = U \setminus (i \cup \mathcal{N}(i))$
  $k \leftarrow k + 1$
until $k > K$ or $U = \{\emptyset\}$

(((SCP 1 − SCP 3)/SCP 3) × 100). The second and surprising observation we found was that in Erdös-Rényi graphs the greedy approach works better than the mixed approach (algorithm SCP 3 which combines the greedy approach and the random choice). The reason why we were expecting to have a different behavior is because the algorithm may start in a bad initial location and through a greedy approach it may take a while before finding good nodes to recruit. In fact, SCP 3 performs worse than the greedy approach SCP 2. We believe that the performance of SCP 3 may improve by modifying the parameter α which we took as $\alpha = 1/2$.

We performed simulations of the previously described algorithms also in Barabasi-Albert graphs. The chosen Barabasi-Albert graphs were undirected graphs generated as follows. We started with a single vertex. At each time step, we added one vertex and the new vertex connects two edges to the old vertices. The probability that an old vertex is chosen is proportional to its degree.

In Figure 3, we notice that random algorithm SCP 1 needs more than twice the number of nodes recruited with SCP 3. The mixed approach with combines the greedy approach and the random choice SCP 2 performs worse than the greedy approach SCP 3.

Likewise, in Figure 4, we see that choosing uniformly at random performs very poorly compared to the greedy one-hop lookahead algorithm.

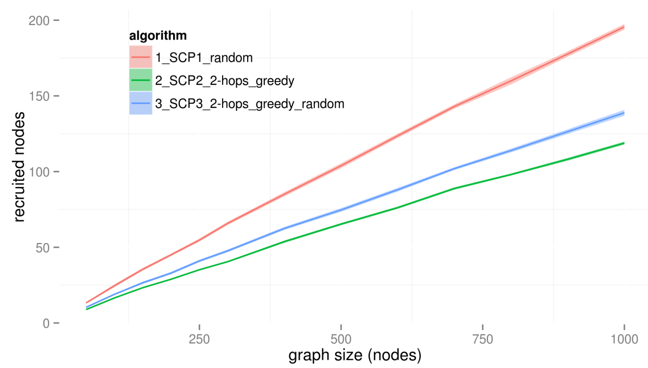

Fig. 1. SCP Algorithms in Erdös-Rényi connected graphs

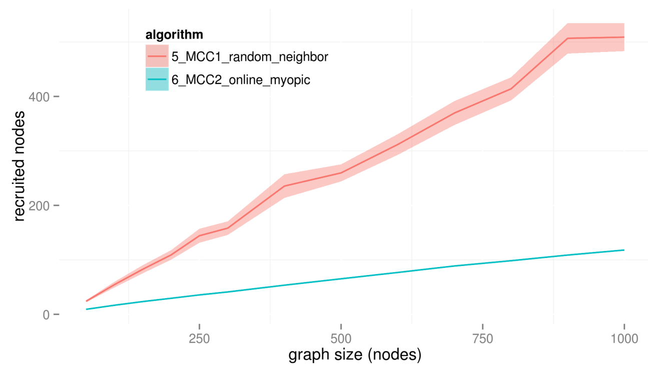

Fig. 2. MCC Algorithms in Erdös-Rényi connected graphs

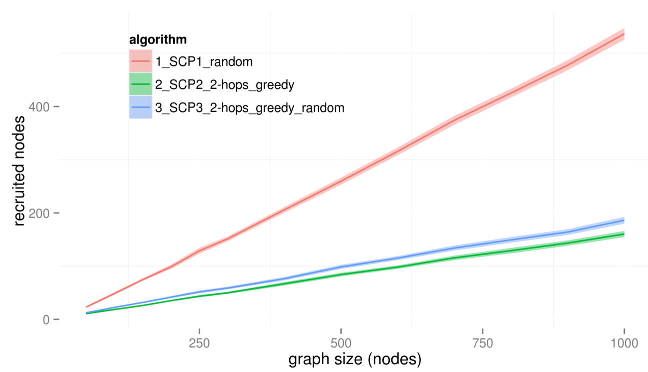

Fig. 3. SCP Algorithms in Barabasi-Albert graphs

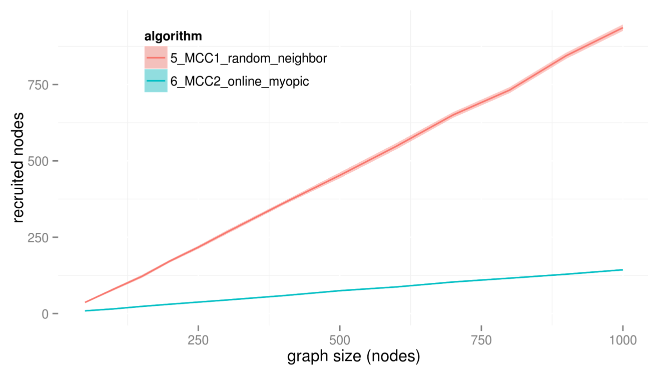

Fig. 4. MCC Algorithms in Barabasi-Albert graphs

VI. Conclusions and Future Directions

In this work, we were interested on two different problems: the maximum coverage problem (or set cover for the complete network) and the maximum connected covering problem. The motivation of our work is viral marketing campaigns on social networks. Our perspective was to analyze both problems from the knowledge we may have of the topology of the network. We presented some existing and new heuristics to both of these problems. We quantified how different levels of information have an effect on the type of algorithm we choose and this translates into a better or worse performance depending on the knowledge we have on the topology of the network.

According to our simulations, we see that the number of recruited nodes scales linearly with the graph size for all the algortihms considered. Unlike expected (see [9]), the simulations show that deterministic greedy algorithms perform better than those in which nodes are chosen at random. There is no need to address greedy isues with 2-hops local information in the graphs considered.

There are many interesting future directions to this work. Just to name a few, one direction is to provide theoretical bounds to the new heuristics and to consider digraphs instead of undirected graphs. Another is to study how changes on the topology of the network can affect the problem at hand.

Acknowledgments

The work of A. Silva was partially done in the context of the ADR "Network Science" of the Joint Alcatel-Lucent Inria Lab. The work of A. Silva was partially carried out at LINCS (www.lincs.fr).

References

[1] K. Avrachenkov, P. Basu, G. Neglia, B. F. Ribeiro, and D. Towsley, "Online myopic network covering," *CoRR*, vol. abs/1212.5035, 2012.

[2] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 57–66. [Online]. Available: http://doi.acm.org/10.1145/502512.502525

[3] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/775047.775057

[4] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-03)*. New York: ACM Press, Aug. 24–27 2003, pp. 137–146.

[5] ——, "Influential nodes in a diffusion model for social networks," in *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 2005.

[6] Mossel and Roch, "On the submodularity of influence in social networks," in *STOC: ACM Symposium on Theory of Computing (STOC)*, 2007.

[7] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978. [Online]. Available: http://dx.doi.org/10.1007/BF01588971

[8] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, 1996.

[9] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, "The Power of Local Information in Social Networks," in *Internet and Network Economics. 8th International Workshop, WINE 2012, Liverpool, UK, December 10-12, 2012. Proceedings*, ser. Lecture Notes in Computer Science, P. W. Goldberg, Ed., vol. 7695. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 406–419.