Weighted alpha-rate dominating sets in social networks

Danica Vukadinović Greetham*, Anush Poghosyan* and Nathaniel Charlton*†
*Centre for the Mathematics of Human Behaviour
Department of Mathematics and Statistics
University of Reading, UK
Email:d.v.greetham,a.poghosyan, n.a.charlton@reading.ac.uk
†CountingLab Ltd
Reading, UK

***Abstract*—We are looking into variants of a domination set problem in social networks. While randomised algorithms for solving the minimum weighted domination set problem and the minimum alpha and alpha-rate domination problem on simple graphs are already present in the literature, we propose here a randomised algorithm for the minimum weighted alpha-rate domination set problem which is, to the best of our knowledge, the first such algorithm. A theoretical approximation bound based on a simple randomised rounding technique is given. The algorithm is implemented in Python and applied to a UK Twitter mentions networks using a measure of individuals' influence (klout) as weights. We argue that the weights of vertices could be interpreted as the costs of getting those individuals on board for a campaign or a behaviour change intervention. The minimum weighted alpha-rate dominating set problem can therefore be seen as finding a set that minimises the total cost and each individual in a network has at least alpha percentage of its neighbours in the chosen set. We also test our algorithm on generated graphs with several thousand vertices and edges. Our results on this real-life Twitter networks and generated graphs show that the implementation is reasonably efficient and thus can be used for real-life applications when creating social network based interventions, designing social media campaigns and potentially improving users' social media experience.**

I. Introduction

The social media advertising industry in UK is growing annually in double figures. Social media platforms provide unique opportunities in comparison with other communication channels to monitor, respond, amplify, and lead consumer behavior. Looking to the wider socio-economic horizon, social media is also slowly but steadily becoming an important channel to run policy information and education campaigns on a mass scale, and an exclusive channel to get the attention of some socio-demographic groups, especially in the younger population, who are not reading newspapers or watching traditional television. All this opens interesting opportunities for social network based behaviour change interventions [1].

In the health-related behaviour change context, for an intervention to work at the individual level, it is often of the utmost importance for a support network to exist (see e.g. [2]). In this way an individual is surrounded with social support. Also, a support network needs to have a major influence on the individual, as possible negative influences also come from her/his social network (for example in interventions aimed at addictive behaviours).

For these reasons, one often needs to find a set of nodes/individuals such that all other or indeed all individuals are connected to that set. In graph theory such a set is called a dominating set and the problem of finding a dominating set of minimal cardinality is NP-complete [3]. The notion was generalised introducing k-domination where each node needs to have at least k neighbours in the dominating set, and α domination where $0 < \alpha \leq 1$, where each node not in the dominating set needs to have at least $\alpha * 100$ percent of its neighbours in the dominating set [4], and α-rate domination [5] where each node (including ones in the dominating set) needs to have at least $\alpha * 100$ percent of its neighbours in the dominating set. Again, finding minimum cardinalities of α and α-rate dominating sets is NP-complete.

Here, we introduce the α-rate dominating set problems on weighted networks. Why weighted networks? It might be that the "best" candidates (from a structural perspective) for dominating sets are not feasible for a behaviour intervention for various reasons: they cannot be a part of the intervention because they do not have the desired attributes, or they do not have time to invest in the intervention. We want to represent this by assigning to each node a cost to be part of the intervention. Thus, our goal is to find the most cost effective set from which we can control or dominate the network. Note that here we do not model negative influences that come from a social network, but just require at least $\alpha * 100$ percent of neighbours to be in the support network.

In the next section we give preliminaries and formally define the problem. In Section III an overview of the previous work is given. In Section IV a theoretical upper bound on weighted α-rate dominating set is given which leads to a simple randomised rounding algorithm using a linear programming formulation of the problem. In Section V we analyse the results obtained from the algorithm's application to a Twitter network and generated graphs and compare them for the non-weighted case with the existing algorithm in [6] for α-rate domination. We conclude in Section VI.

II. Preliminaries

In this section we introduce the notation and definitions that we use throughout this paper.

A simple, undirected graph G is an ordered pair G = $(V,\ E)$ where V is a set, elements of which are called vertices or nodes, and E is a set of unordered pairs of distinct vertices called edges. We assume that no self-loops or multiple edges are allowed. If G is a graph of order n, then $V(G) = \{v_1, v_2, ..., v_n\}$ is the set of vertices in G, d_v denotes the degree of v, and $\bar{d_v} = d_v + 1$. Let $N(v)$ denote the *neighbourhood* of a vertex v. Also, let $N(V) = \cup_{v \in V} N(v)$ and $N[V] = N(V) \cup V$. Then $\bar{d_v} = |N[v]|$. Denote by $\delta(G)$ and $\Delta(G)$ the *minimum* and *maximum* degrees of vertices of G, respectively. Put $\delta = \delta(G)$ and $\Delta = \Delta(G)$.

A set D is called a *dominating set* if every vertex not in D is adjacent to one or more vertices in D. The minimum cardinality of a dominating set of G is called the *domination number* $\gamma(G)$.

Let α be a real number satisfying $0 < \alpha \leq 1$. A set $X \subseteq V(G)$ is called an *α-dominating set* of G if $|N(v) \cap X| \geq \alpha d_v$ for every vertex $v \in V(G) \setminus X$, i.e. v is adjacent to at least $\lceil \alpha d_v \rceil$ vertices of X. The minimum cardinality of an α-dominating set of G is called the *α-domination number* $\gamma_\alpha(G)$. It is easy to see that $\gamma(G) \leq \gamma_\alpha(G)$, and $\gamma(G) = \gamma_\alpha(G)$ if α is sufficiently close to 0. A set $X \subseteq V(G)$ is considered an *α-rate dominating set* of G if for any vertex $v \in V(G)$, $|N[v] \cap X| \geq \alpha d_v$. The minimum cardinality of an α-rate dominating set of G is called the *α-rate domination number* $\gamma_{\times\alpha}(G)$. It is easy to see that $\gamma_\alpha(G) \leq \gamma_{\times\alpha}(G)$.

Now we consider vertex-weighted graphs. These are finite and undirected graphs, with no self-loops or multiple edges, in which each vertex has been assigned a weight. Let w_v be the weight (cost) of each vertex v of graph G. Let $\gamma_w(G)$ denote a minimum weight of a dominating set X of G and let $\gamma_{\times\alpha,w}$ denote a minimum weight of an α-rate dominating set D. Finding an α-rate dominating set D of G such that $\sum_{v \in D} w_v$ is minimised is the main problem studied in this paper.

III. Previous work

Variants of domination have been studied extensively and have various applications for real life problems. A smaller number of studies in domination parameters consider weighted graphs in particular.

The minimum weighted dominating set problem is one of the classic NP-hard optimisation problems in graph theory. Zou et al. [7] studied the minimum-weighted dominating set and the minimum-weighted connected dominating set problems on a node-weighted unit disk graph and devised approximation algorithms for these problems with performance ratios of $5 + \varepsilon$ and $4 + \varepsilon$ respectively. In [8] Polynomial Time Approximation Scheme (PTAS) was generalised for the weighted case in polynomial growth bounded graphs with a bounded degree constraint. A variant of the weighted dominating set problem — the weighted minimum independent k-domination (WMIkD) problem — was studied by Yen in [9]; an algorithm linear in the number of vertices of the input graph for the WMIkD problem on trees was presented.

Discussing a more general domination set problem [10], where the direct connections are replaced with shortest paths corresponding to some measure f defined on the vertices of a graph, the authors give an approximation algorithm for the vertex-weighted version. Using randomised rounding they prove the approximation ratio of $O(\log \Delta)$ for their randomised algorithm, where Δ is the maximum cardinality of the sets of vertices that can be dominated by any single vertex, or in our case the maximum degree of the vertices in the graph.

In [11], the maximum spanning star forest problem is discussed, which is the complement problem of domination set. A 0.71-approximation algorithm for this problem is given, and for the vertex-weighted case a 0.64-approximation algorithm is presented.

The α-domination problem was introduced by Dunbar et al. in [4]. Introduced by Zverovich et al. [5] the concept of α-rate domination can be considered as a particular case of an α dominating set in the same graph. Note that both the α and α-rate domination problems are known to be NP-complete. Thus it is of importance to determine bounds for α and α-rate domination numbers and various similar parameters. In [6] and [5] the authors explicitly provide new upper bounds and randomised algorithms for finding the α and α-rate domination sets in terms of a parameter α and graph vertex degrees on undirected simple finite graphs by using probabilistic constructions. Their algorithm is bounded by:

$$\gamma_{\times\alpha}(G) \leq \left(1 - \frac{\widehat{\delta}}{(1+\widehat{\delta})^{1+1/\widehat{\delta}}\ \widetilde{d}_\alpha^{\,1/\widehat{\delta}}}\right) n, \tag{1}$$

where $\widetilde{d}_\alpha$ is a closed α-degree of G and $\widehat{\delta} = \lfloor \delta(1-\alpha) \rfloor + 1$.

Studies of the propagation of influence in the context of social networks carried out by Wang et al. in [12] resulted in introducing new variants of domination such as the positive influence dominating set (PIDS) and total positive influence dominating set (TPIDS). From the definitions given in [12] it is easy to see that PIDS and TPIDS problems are equivalent to α-dominating and α-rate dominating set problems respectively for a special case when $\alpha = 1/2$. Wang et al. proved that both these problems are NP-hard. Thus, it is important to study approximability of the problems. In their work Dinh et al. [13] generalise PIDS and TPIDS by allowing any $0 < \alpha < 1$ and show that both problems can be approximated within a factor $\ln \Delta + O(1)$ and present a linear time exact algorithm for trees.

IV. Randomised rounding algorithm

In this section we present an approximation algorithm by randomised rounding for constructing a minimum weighted α-rate dominating set problem. The algorithm is based on the probabilistic method [14] and the techniques used by Chen at al. [10] for simple domination with measure functions (where adjacency may be replaced with limited length paths) on weighted graphs.

Let us assume that for every vertex v_i, $1 \le i \le n$ the variable x_i has the following meaning: $x_i = 1$ if v_i is contained in the α-rate dominating set and $x_i = 0$ otherwise. We consider the following linear programming relaxation LP of an integer program IP:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} \quad & \sum_{v_j \in N[v_i]} x_j \ge \lceil \alpha \bar{d_{v_i}} \rceil, \quad \forall v_i \in V \\ & 0 \le x_i \le 1, \quad \forall 1 \le i \le n. \end{aligned}$$

As we know LP is polynomial-time solvable and we can compute an optimal solution $\{\widehat{x_i}\}_{1 \le i \le n}$. If we denote with IP_{OPT} an optimal solution of the corresponding integer program IP we have that

$$IP_{OPT} \ge \sum_{i=1}^{n} w_i \widehat{x_i}. \tag{2}$$

We obtain a candidate IP solution $\{x_i\}_{1 \le i \le n}$ by using randomised rounding, setting $x_i = 1$ with probability $\widehat{x_i}$ and 0 otherwise. Let D be the set of vertices that are assigned ones after rounding, i.e. $D = \{v_i | x_i = 1, 1 \le i \le n\}$.

In the next step we estimate the probability that D is a feasible solution for IP. For any vertex $v \in V$, with d_v neighbours, let $k = \lceil \alpha \bar{d_{v_i}} \rceil$. We know that $\sum_{v_i \in N[v]} \widehat{x_i} \ge k$, and $\forall x_i, 0 \le \widehat{x_i} \le 1$. Now, the probability that v_i is α-rate dominated is equal to

$$Pr(v_i \text{ is } \alpha\text{-rate dom.}) = 1 - (Pr(v_i \text{ is not } \alpha\text{-rate dom.})).$$

We can look at the number of neighbours of v (including v itself) which are in D as the sum of $\bar{d_v}$ independent trials, random processes, where the success probability of each trial i is equal to $\widehat{x_i}$. Thus this sum, $|N[v] \cap D|$, follows Poisson's binomial distribution [15] with parameters $\widehat{x_1}, \ldots, \widehat{x_{\bar{d_v}}}$. Let $S = \{1, 2, \ldots \bar{d_v}\}$, and $\mathcal{F}_k = \{A | A \subseteq S, |A| = k\}$ denote all subsets of S with exactly k members where we are going over all possible combinations. Then $|\mathcal{F}_k| = \binom{\bar{d_v}}{k}$. Let A^C denote the complementary set, i.e. $S \setminus A$.

$$Pr(v_i \text{ is not } \alpha\text{-rate dom.}) = \sum_{l=0}^{k-1} \sum_{A \in \mathcal{F}_l} (\prod_{i \in A} \widehat{x_i})(\prod_{j \in A^c} (1 - \widehat{x_j})). \tag{3}$$

Theorem 1:

$$Pr(v_i \text{ is not } \alpha\text{-rate dom.}) < \frac{1}{2}. \tag{4}$$

Proof:

Let the random variable X be the number of neighbours that vertex v has in D. Then X follows Poisson's binomial distribution with parameters $\widehat{x_1}, \ldots, \widehat{x_{\bar{d_v}}}$:

$$Pr(X = l) = \sum_{A \in \mathcal{F}_l} (\prod_{i \in A} \widehat{x_i})(\prod_{j \in A^c} (1 - \widehat{x_j})).$$

Showing our goal (4) is equivalent to showing

$$\frac{1}{2} \le \sum_{l=k}^{\bar{d_v}} \sum_{A \in \mathcal{F}_l} (\prod_{i \in A} \widehat{x_i})(\prod_{j \in A^c} (1 - \widehat{x_j})) = Pr(k \le X). \tag{5}$$

So we are looking for a minimum of the right hand side of (5) subject to $\sum_{i=1}^{\bar{d_v}} x_i \ge k$ (this minimum must exist by continuity and compactness). Clearly the minimum will be found when $\sum_{i=1}^{\bar{d_v}} x_i = k$; increasing one of the x_is without changing the others will clearly only increase the RHS. (Intuitively, increasing the probability of success in one of the trials, while leaving the others unchanged, can only increase the probability of getting at least k successes.) Keep in mind that $0 \le \widehat{x_i} \le 1$ for all $\widehat{x_i}$. So we may assume that

$$\sum_{v_i \in [N(v)]} \widehat{x_i} = k. \tag{6}$$

Now we can use the result from [16], Theorem 5, that shows that the tail distribution function of Poisson's binomial distribution attains its minimum in the binomial distribution, i.e. when all probabilities are equal. The theorem states that for two integers b, and c such that $0 \le b \le np \le c \le n$, the probability $P(b \le X \le c)$ reaches its minimum where all the probabilities $p_1 = \ldots = p_n = p$, unless $b = 0$ and $c = n$. Here the p_is are the probabilities (or parameters) of Poisson's binomial distribution, and n and p are the parameters of the related binomial distribution. We apply that theorem taking the two integers b and c to be our k and $\bar{d_v}$ respectively. We have that p, the equal probability is $\frac{k}{\bar{d_v}}$ from (6), whence np equals our k. The theorem gives us

$$\sum_{l=k}^{\bar{d_v}} \binom{\bar{d_v}}{l} p^l (1-p)^{\bar{d_v} - l} \quad \le \quad Pr(k \le X).$$

Thus, we will be done if we can show that

$$\sum_{l=k}^{\bar{d_v}} \binom{\bar{d_v}}{l} p^l (1-p)^{\bar{d_v} - l} \tag{7}$$

is at least $\frac{1}{2}$. Let Y be a random variable of binomial distribution with $\bar{d_v}$ trials each of probability p. Then observe that in fact $Pr(Y \ge k)$ is equal to (7) above. The median of Y is bounded by $\lfloor \bar{d_v} p \rfloor$ and $\lceil \bar{d_v} p \rceil$ [17], but $\bar{d_v} p$ is exactly the integer k, so k is the unique median of Y. It follows from the defining property of medians that $Pr(Y \ge k) \ge \frac{1}{2}$, and thus $Pr(Y < k) < \frac{1}{2}$ and the proof is complete. ■

Hence, the probability is lower bounded by $\frac{1}{2}$, and the feasibility follows. Let A_i denote the event that vertex v_i is α-rate dominated and let $B = \cap_{i=1}^{n} A_i$ be the event that all vertices are dominated. We use the amplification approach (repeating randomised rounding $t = O(\log_2 \Delta)$ times) as found in [10] which results in $Pr([x_i = 1]) = 1 - (1 - \hat{x}_i)^t$. We obtain that the expected value of the solution resulting from randomised rounding, given that event B happens, (i.e.

that the solution is feasible) is

$$\begin{aligned} E\left[\sum_{i=1}^{n} w_i x_i | B\right] &= \sum_{i=1}^{n} w_i Pr([x_i=1]|B) \\ &= \sum_{i=1}^{n} w_i \frac{Pr(B|[x_i=1])}{Pr[B]} Pr(x_i=1) \\ &\leq \sum_{i=1}^{n} w_i \frac{1}{\prod_{j\in N(v_i)} Pr(A_j)} (1-(1-\hat{x_i})^t) \\ &\leq \frac{1}{(1-2^{-t})^{\Delta}} \sum_{i=1}^{n} w_i (1-(1-t\hat{x_i})) \\ &\leq \frac{t}{(1-2^{-t})^{\Delta}} \sum_{i=1}^{n} w_i \hat{x_i} \\ &\leq O(\log_2 \Delta \cdot OPT). \end{aligned}$$

Hence, there exists a particular solution that is within $O(\log_2 \Delta)$ ratio to the optimal solution. Note that $C = \frac{1}{(1-\frac{1}{\Delta})^{\Delta}}$ decreases monotonically down to e with increasing Δ and assuming that $\Delta \geq 2$, the maximum is achieved for $\Delta = 2$, $C = 4$. A simple randomised rounding algorithm AlgRR follows immediately, by first solving LP and then rounding the solutions to zero or one. This process is repeated $\lceil \log_2 \Delta \rceil$ times. All vertices with ones then create with high probability an α-rate domination set with the sum of the weights within $O(\log_2 \Delta)$ factor of the optimal solution. We implemented AlgRR in Python.

V. Twitter UK mentions network

The Twitter data-set was collected on our behalf by Datasift, a certified Twitter partner, which allowed us to access the full Twitter firehose rather than being rate-limited. The data-set consists of all UK based[1] Twitter users that sent tweets with at least one mention between 8 Dec 2011 and 4 Jan 2012 (28 days in total). Mentions are messages that include an @ followed by a username and are used to address people. Thus, if person A posts a tweet containing "@B" that means A is addressing the tweet to B specifically. Mentions are not private messages and can be read by anyone who searches for them. A tweet can be addressed to several users simultaneously by using @ repetitively.

A. Data

We preprocessed the data, removing empty mentions and self-addressing which left us with $3,614,705$ time-stamped arcs (individual mentions) from a total of $819,081$ distinct usernames, or nodes. We then removed all users who did not tweet but just received messages, as we did not have a weight measure for them. There were approximately 50k nodes that appeared both as tweeters and receivers. We aggregated the data on a weekly basis and kept only two-directional arcs (thus if person A mentioned B and person B mentioned A at least once during a week there is a bi-directional edge between A and B in the weekly graph). For simplicity, we treated those bi-directional edges as undirected. This left us with 4 undirected weekly graphs with around 5k nodes in each and around 3.5k edges on average. For each vertex we retrieved its *Klout* score and used it as the weight. The Klout score measures an individual's influence based on her/his social media activity[2] It is a single number that represents the aggregation of multiple pieces of data about individuals' social media activity, based on a score model which is not publicly available [18]. The descriptive statistics of the Twitter mentions weekly graphs are given in Table I below. As the 4 mentions graphs are quite sparse, we experimented in addition with random graphs with similar number of nodes and greater number of edges (denoted *rnd-d*, where d is for dense). We created those random graphs using the `dense_gnm_random_graph` method of NetworkX [19] which picks a graph randomly out of the set of all graphs with n nodes and m edges. Additionally, we used the NetworkX method `powerlaw_cluster_graph` to create graphs with similar number of vertices and edges as random but that also satisfy preferential attachment and high average clustering (we used 0.8 for probability of triangles)[20]. These graphs are denoted with *pref-d*. The weights were created by picking uniformly a random number from 1 to 71. The details are given in Table II .

B. Results

In this section we investigate how our randomised rounding algorithm AlgRR performs on some real and created networks. We also compare it with the existing α-rate domination algorithm for simple (non-weighted) graphs from [6] (denoted here as AlgA). We have run both algorithms on the 4 weekly Twitter graphs, random graphs and preferential graphs described above. As the algorithms are randomised, we have run both algorithms 100 times taking averages. Results are presented in Tables III and IV below.

The results on Figs 1 and 2 show that for dense networks (networks denoted with *pref-d* and *rnd-d*) the algorithm AlgRR outperforms algorithm AlgA significantly and not only in terms of total weights (which would be expected, as algorithm AlgA optimises the size of α-rate dominating set, while algorithm AlgRR optimises the weights) but also on the sizes of α-rate dominating sets. According to the theoretical bounds for algorithm AlgA the probability with which each candidate vertex for the α-rate dominating set is selected gets close to 1 for dense networks *pref-d*, thus resulting in selecting all the nodes of the network. However on sparse networks such as *twitt1-4* algorithm AlgA slightly outperforms algorithm AlgRR.

Since the algorithm AlgRR is based on LP-relaxation technique it runs in polynomial time. Our results show that the fastest run for algorithm is 6.99 ms on *twit3* network where

[1] All Twitter users appearing in our data-set had selected the UK as their location.

[2] In Twitter, Klout focuses on retweets of a user's tweets, their username mentions by other users, their list memberships on other users' curated lists, a number of followers and a number and frequency of replies i.e. how engaged they are.

TABLE I
TWITTER MENTIONS NETWORK STATISTICS, ME DENOTES NUMBER OF MULTI-EDGES IN THE ORIGINAL GRAPHS, CC NUMBER OF CONNECTED COMPONENTS IN THE UNDIRECTED GRAPHS, K IS A KLOUT NUMBER.

Graph	ME	V	E	CC	δ	Δ	δ_{avg}	K_{min}	K_{max}	K_{avg}
twitt1	244829	5775	3716	2174	1	16	1.2869	10	71	33
twitt2	236104	5537	3537	2094	1	19	1.2776	10	71	34
twitt3	226707	5279	3434	1957	1	15	1.3010	10	71	34
twitt4	244362	5597	3599	2093	1	16	1.2860	10	71	33

TABLE II
EXAMPLES OF CREATED RANDOM GRAPHS STATISTICS.

name	V	E	CC	δ	Δ	δ_{avg}	K_{min}	K_{max}	K_{avg}
rnd-d	5000	50000	1	4	41	20	1	71	36
pref-d	5000	49835	1	9	615	19.93	1	71	36

TABLE III
ALPHA-RATE DOMINATION SETS' SIZES (#), WEIGHTS(W) AND RUNNING TIMES(T) FOR ALGA, FOR DIFFERENT GRAPHS AND $\alpha = 0.25, 0.5, 0.75$ RESPECTIVELY.

Graph	Avg#	AvgW	Min#	Max#	MaxW	MinW	AvgT(ms)
pref-d0.25	5000	193419	5000	5000	193419	193419	12.71
pref-d0.5	5000	194267	5000	5000	194267	194267	12.87
pref-d0.75	5000	188938	5000	5000	188938	188938	13.09
rnd-d0.25	4730	182675	4689	4768	184149	180909	12.11
rnd-d0.5	4991	191934	4985	4998	192222	191573	12.35
rnd-d0.75	4998	194278	4994	5000	194343	194082	12.58
twitt10.25	4328	146960	4259	4408	149577	144171	3.21
twitt10.5	5291	179701	5222	5334	181377	176986	4.21
twitt10.75	5056	171733	4993	5113	173600	169609	3.80
twitt20.25	4612	157207	4539	4670	159495	154516	3.22
twitt20.5	5453	185872	5436	5475	186780	185343	3.77
twitt20.75	5259	179254	5217	5297	180618	177888	3.58
twitt30.25	3960	135557	3879	4031	138291	132624	2.70
twitt30.5	4897	167738	4854	4946	169551	165730	3.60
twitt30.75	4753	162770	4697	4794	164233	160732	3.24
twitt40.25	4195	142143	4119	4289	145122	139323	3.24
twitt40.5	5127	173809	5069	5183	175550	171776	3.64
twitt40.75	5036	170727	4979	5092	172826	168874	3.53

$\alpha = 0.25$, and the longest time it takes to run is 1180.02 ms on dense network (*rnd-d*) where $\alpha = 0.75$.

The analysis of AlgA has shown a similar spread of solutions for different runs, and were relatively stable. For algorithm AlgRR the results in Table IV show significant difference between the minimum and maximum cardinalities of α-rate dominating sets for dense networks *pref-d* and *rnd-d*. This indicates that the values of variables in the solutions obtained by LP relaxation are spread out over $(0, 1)$ interval (i.e. are fractional). We have verified the spread and consistency by performing additional 200 runs for algorithm AlgRR where $\alpha = 0.25$ for *pref-d* and *rnd-d* networks and recorded the average number of variables in the solution obtaining values from $(0, 0.5]$ and (0.5,1) intervals as well as the average number of variables with values equal to 1 (see Table V).

Our results also show that by increasing the number of runs for algorithm AlgRR up to 200 on *pref-d* and *rnd-d* networks, the difference between minimum and maximum cardinalities of α-rate dominating sets does not change significantly compared with the results obtained from 100 runs. Thus it can be concluded that more runs are unlikely to achieve better results.

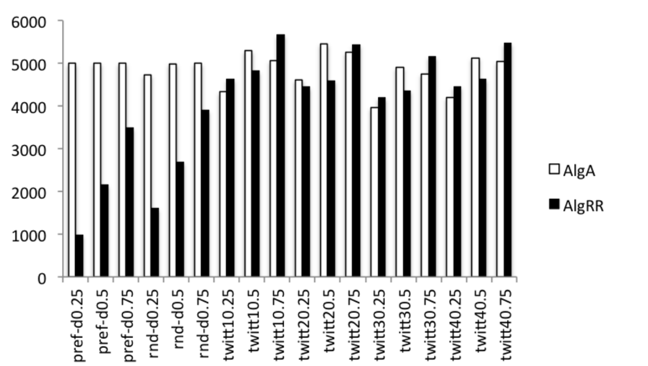

Fig. 1. Comparison of average sizes of alpha-rate domination sets solutions for AlgA and AlgRR

Figure 3 gives sizes and weights for 100 runs for AlgRR on *rnd-d* networks where $\alpha = 0.5$. Similar plots were obtained for all graphs and for both algorithms.

TABLE IV
ALPHA-RATE DOMINATION SETS' SIZES, WEIGHTS AND RUNNING TIMES FOR ALGRR, FOR DIFFERENT GRAPHS AND $\alpha = 0.25, 0.5, 0.75$ RESPECTIVELY.

Graph	Avg#	AvgW	Min#	Max#	MaxW	MinW	AvgT(ms)
pref-d0.25	979	16647	578	1512	30845	7124	65.70
pref-d0.5	2158	51541	1498	2777	75523	28143	139.24
pref-d0.75	3489	109794	2661	4279	149581	71461	256.96
rnd-d0.25	1604	28367	960	2420	56342	10869	341.29
rnd-d0.5	2688	68953	1843	3664	115395	34074	870.87
rnd-d0.75	3901	130250	2747	4770	180502	70105	1180.02
twitt10.25	4628	153367	4607	4655	154355	152560	8.79
twitt10.5	4817	159901	4792	4837	160658	158939	9.11
twitt10.75	5665	191648	5665	5665	191648	191648	9.29
twitt20.25	4443	148065	4413	4469	148980	147007	7.77
twitt20.5	4595	153249	4566	4625	154311	152196	8.06
twitt20.75	5431	184481	5431	5431	184481	184481	8.13
twitt30.25	4191	140094	4166	4220	141094	139172	6.99
twitt30.5	4360	145765	4321	4391	146869	144362	7.13
twitt30.75	5159	175854	5159	5159	175854	175854	7.26
twitt40.25	4459	147854	4447	4471	148289	147384	7.80
twitt40.5	4637	153552	4624	4651	154067	153058	8.12
twitt40.75	5468	184486	5468	5468	184486	184486	8.16

TABLE V
SPREAD OF VARIABLES' VALUES IN THE SOLUTIONS OBTAINED BY LP RELAXATION IN ALGRR

name	α	Avg# Var $\in (0, 0.5]$	Avg# Var $\in (0.5, 1]$	Avg# Var $= 1$
rnd-d	0.25	808	744	838
pref-d	0.25	503	477	494

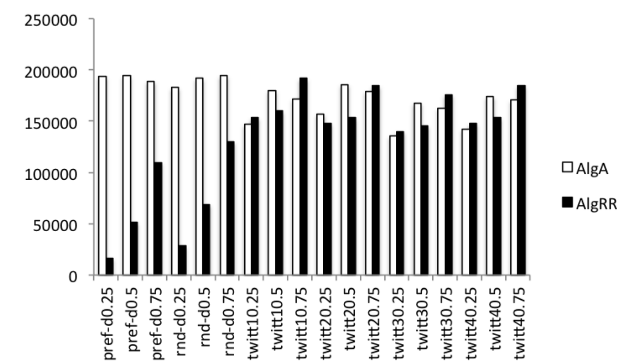

Fig. 2. Comparison of average sum of weights of alpha-rate domination sets solutions for AlgA and AlgRR

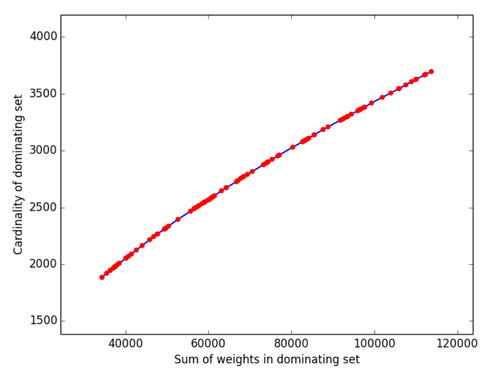

Fig. 3. Variation in size and weight for 100 runs of AlgRR, for *rnd-d*, $\alpha = 0.5$

VI. CONCLUSION

We have explored how to pick optimal sets of individuals for interventions in social networks. If each person in a network has been assigned a cost, the aim was to find a group of people having the minimum total cost so that each individual in network has at least $\alpha * 100$ percent of its neighbourhood in this designated group. We presented a randomised algorithm for finding approximations of minimum weight α rate domination set in graphs. We proved that this algorithm's output is within $O(\log_2 \Delta)$ ratio of the optimal solution. We are not aware of any other existing algorithms that work for α-rate domination on vertex-weighted networks. We have shown on the real-life networks that the algorithm compares beneficially with the existing algorithm for the non-weighted version and for the denser generated networks produces in most cases not just sets with smaller sums of weights but also significantly smaller sets. This is compensated by longer running time, as a linear program needs to be solved. Thus, although we were able to run our algorithm in reasonable time on graphs with around 50k edges, it would be interesting to look at the different solutions scalable for very large networks.

Acknowledgments.: This work is partially funded by the RCUK Digital Economy programme via EPSRC grant EP/G065802/1 'The Horizon Hub'. We would like to thank Datasift for the provision of the data analysed.

REFERENCES

[1] T. Valente, "Network interventions," *Science*, vol. 337, no. 6090, 2012.

[2] C. Greaves, P. Reddy, and K. Sheppard, "Supporting behaviour change for diabetes prevention." in *Diabetes Prevention in Practice.*, P. Schwarz, P. Reddy, C. Greaves, J. Dunbar, and S. J., Eds. Dresden: Tumaini Institute for Prevention Management, 2010, pp. 19–29.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[4] R. L. J.E. Dunbar, D.G. Hoffman and L. Markus, "α-domination," *Discrete Math.*, vol. 211, pp. 11–26, 2000.

[5] A. Gagarin, A. Poghosyan, and V. E. Zverovich, "Upper bounds for alpha-domination parameters," *Graphs and Combinatorics*, vol. 25, no. 4, pp. 513–520, 2009.

[6] A. Gagarin, A. Poghosyan, and V. Zverovich, "Randomized algorithms and upper bounds for multiple domination in graphs and networks," *Discrete Applied Mathematics*, vol. 161, no. 4-5, pp. 604 – 611, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X11002423

[7] F. Zou, Y. Wang, X.-H. Xu, X. Li, H. Du, P. Wan, and W. Wu, "New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs," *Theoretical Computer Science*, vol. 412, no. 3, pp. 198 – 208, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397509004162

[8] Z. Wang, W. Wang, J. Kim, B. M. Thuraisingham, and W. Wu, "Ptas for the minimum weighted dominating set in growth bounded graphs," *J. Global Optimization*, vol. 54, no. 3, pp. 641–648, 2012.

[9] W. C.-K. Yen, "Algorithmic results of independent k-domination on weighted graphs." *Chiang Mai Journal of Science*, vol. 38, pp. 58–70, 2011.

[10] N. Chen, J. Meng, J. Rong, and H. Zhu, "Approximation for dominating set problem with measure functions." *Computing and Informatics*, vol. 23, pp. 37–49, 2004.

[11] N. Chen, R. Engelberg, C. T. Nguyen, P. Raghavendra, A. Rudra, and G. Singh, "Improved approximation algorithms for the spanning star forest problem," in *Proc. APPROX 2007, LNCS*, 2007, pp. 44–58.

[12] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, and S. Shan, "On positive influence dominating sets in social networks," *Theoretical Computer Science*, vol. 412, no. 3, pp. 265 – 269, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397509007221

[13] T. N. Dinh, Y. Shen, D. T. Nguyen, and M. T. Thai, "On the approximability of positive influence dominating set in social networks," *Journal of Combinatorial Optimization*, pp. 1–17, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10878-012-9530-7

[14] N. Alon and J. Spencer, *The Probabilistic Method*. John Wiley, 1992.

[15] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *In SRDS*, 2003, pp. 25–34.

[16] W. Hoeffding, "On the distribution of the number of successes in independent trials," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 713–721, 1956.

[17] R. Kaas and J. Buhrman, "Mean, median and mode in binomial distributions," *Statistica Neerlandica*, vol. 34, no. 1, pp. 13–18, 1980. [Online]. Available: http://dx.doi.org/10.1111/j.1467-9574.1980.tb00681.x

[18] Klout, "http://klout.com/corp/klout_score".

[19] A. Hagberg, D. Schult, and P. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference (SciPy2008), Passadena, CA USA*, August 2008, pp. 11–15.

[20] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Phys. Rev. E*, vol. 65, p. 026107, 2002.