Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 28, no. 3, pp. 11–30 (2024)
DOI: 10.7155/jgaa.v28i3.2969

Graph Burning in Community-based Networks

Gennaro Cordasco[1] *Luisa Gargano*[2] *Adele A. Rescigno*[2]

[1]Dipartimento di Psicologia,
Università degli Studi della Campania "Luigi Vanvitelli", Caserta, Italy
[2]Dipartimento di Informatica,
Università degli Studi di Salerno, Fisciano, Italy

Submitted: May 2023	Accepted: May 2024	Published: September 2024
Article type: Regular paper	Communicated by: C.-C. Lin, B. M.-T. Lin, G. Liotta

Abstract. Graph burning is a deterministic, discrete-time process that can be used to model how influence or contagion spreads in a graph. In the graph burning process, each node starts as dormant, and becomes informed/burned over time; when a node is burned, it remains burned until the end of the process. In each round, one can burn a new node (source of fire) in the network. Once a node is burned in round t, in round $t + 1$, each of its dormant neighbors becomes burned. The process ends when all nodes are burned; the goal is to minimize the number of rounds. We study a variation of graph burning in order to model spreading processes in community-based networks. With respect to a specific piece of information, a community is *satisfied* when this information reaches at least a prescribed number of its members. Specifically, we consider the problem of identifying a minimum length sequence of nodes that, according to a graph burning process, allows to satisfy all the communities of the network. We investigate this NP-hard problem from an approximation point of view, showing both a lower bound and a matching upper bound. We also investigate the case when the number of communities is constant and show how to solve the problem with a constant approximation factor. Moreover, we consider the problem of maximizing the number of satisfied groups, given a budget k on the number of rounds.

1 Introduction

The study of complex networks has experienced a surge of interest over the past decade because of the ubiquity of natural systems with such network structures in social sciences, transport infrastructures, biology, communications, financial markets, and more. In particular, spreading processes

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. A preliminary version of this paper appeared in [20].

E-mail addresses: gennaro.cordasco@unicampania.it (Gennaro Cordasco) lgargano@unisa.it (Luisa Gargano) arescigno@unisa.it (Adele A. Rescigno)

have recently gained a great deal of attention. There are many situations where members of a network may influence their neighbors' behavior and decisions, by swaying their opinions, suggesting what products to buy, but also influencing the behavior to take in many social problems such as public health awareness, financial inclusion, and more. A fundamental aspect to understand and controlling the spreading dynamics is the identification of spreaders that can diffuse information within the network in the least possible amount of time.

Small and marginalized groups within a larger community are those who need the most attention, information and assistance. It is important, then, to ensure that each group receives an appropriate amount of information and resources, so as to respect the diverse composition of the communities.

To address the above issue, in this paper, we consider the problem of ensuring the right amount of informed representatives for each group in the network within fast spreading processes.

In our model, the spreading process reflects the *burning process* in a graph. Bonato [5] introduced the notion of graph burning as a simplified model for the spread of memes in a network. Imagine someone trying to optimize the spread of a meme, hitting key actors in the network with the meme in a given priority sequence. To recap graph burning, nodes start off as dormant and become informed/burned over time. If a dormant node is neighboring an informed one, it becomes informed too. One can burn/inform a new node anywhere in the network in each step. We can then see the process to proceed in sequential discrete steps, where one node is selected at each time-step t as a *source* of fire and burns all of its neighbors at the next time step $t+1$. The nodes that are burned, can burn their neighbors at the next time step. That is, the information can pass from a node who have been informed in the previous time step. Furthermore, an informed node remains informed or burned throughout the process.

The analysis of the burning process enables evaluating the robustness of a network with respect to misinformation strategies (diffusion of negative behaviors) and on the other hand, it allows optimizing the impact of positive strategies.

Problems definitions. We model the network as an undirected graph $G = (V, E)$, where V is the set of individuals and the set of edges E represents the relationships among members of the network, i.e., $(u, v) \in E$ if individuals u and v can directly communicate. We denote by $n = |V|$ the number of individuals in the network. For any $u, v \in V$, we denote by $d(u, v)$ the distance between u and v in G. We denote by $N_d[v] = \{w \mid w \in V,\ 0 \leq d(v, w) \leq d\}$ the set of all nodes having distance at most d from $v \in V$; we call $N_d[v]$ the *neighborhood of radius d around v (d-neighborhood of v)*. We denote by $N[v] = N_1[v]$ the closed neighborhood of v, that is, the set composed by v and its neighbors. Note also that $N_0[v] = \{v\}$.

The Graph Burning problem [5] is defined as follows.

> Graph Burning
> **Input:** A graph $G = (V, E)$.
> **Output:** A sequence $(v_1, v_2, \ldots, v_k)$, with $v_i \in V$ for each[1] $i \in [k]$, of minimum length k such that $\bigcup_{i=1}^{k} N_{k-i}[v_i] = V$.

If $(v_1, v_2, \ldots, v_k)$ is a burning sequence for the given graph G, then we say that a source v_i, where $i \in [k]$, burns another node $u \in V$ if $u \in N_{k-i}[v_i]$, that is v_i burns only the nodes within distance $k - i$ from v_i. We notice that a node can be burned by several sources (i.e., nodes belonging to the burning sequence). Each node $u \in V$ must be either a source or burned from at least one of the sources. An example of Graph Burning is depicted in Fig. 1.

[1] For any integer a, we denote by $[a]$ the set $\{1, 2, \ldots, a\}$.

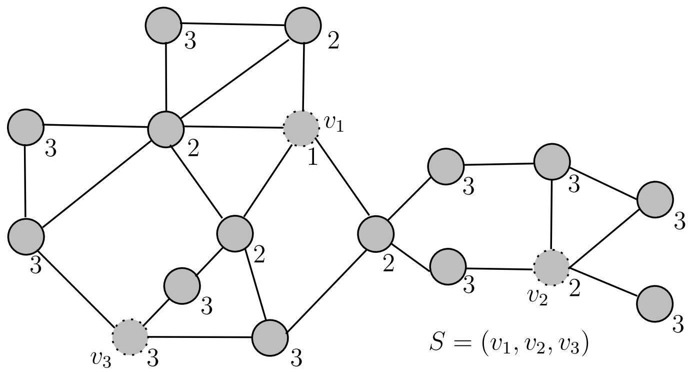

Figure 1: Graph Burning example: An example of graph $G = (V, E)$ with a burning sequence $S = (v_1, v_2, v_3)$. Source nodes are depicted with a dotted border. The number close to the node denotes the time step in which the node is burned.

In our setting, we are given a family of subsets $\{V_1, \dots, V_\omega\}$, referred to as *groups*, of the node set V and we are interested in the minimum time needed to reach at least a given number of individuals in each group. In particular, a positive integer p_j is assigned to each group V_j. The value p_j represents the minimum amount of nodes of the group V_j that have to be informed (burned) during the spreading process. It is worth mentioning that the family $\{V_1, \dots, V_\omega\}$ is not necessarily a partition of V (i.e., groups overlapping is allowed).

Burning with Groups (BG)
Input: $G = (V, E)$, a group family $\Pi = \{V_1, V_2, \dots, V_\omega\}$ (a collection of subsets of V), and a vector $R = (p_1, p_2, \dots, p_\omega)$ of requirements for each group.
Output: A sequence $(v_1, v_2, \dots, v_k)$, where $v_i \in V$, of minimum length k such that

$$\left| \left(\bigcup_{i=1}^{k} N_{k-i}[v_i] \right) \bigcap V_j \right| \geq p_j, \quad \text{for each } j \in [\omega]. \tag{1}$$

An example of Burning with Groups is depicted in Fig. 2. We notice that BG is a generalization of the Graph Burning problem. Indeed, starting from BG, the Graph Burning problem is obtained considering each node as a single group (i.e. $w = n$) and fixing all the requirements equal to 1, or alternatively, considering all the nodes in a single group (i.e., $w = 1$) and fixing its requirement to n.

In the following we assume that for each $j \in [\omega]$, $p_j \leq |V_j|$, otherwise the equation (1) is not satisfiable. Moreover, we assume that $k \leq n$, since with a burning sequence of n nodes the equation (1) is trivially satisfied.

We will also consider the following corresponding maximization problem.

Burning Max Groups (BMG)
Input: $G = (V, E)$, a group family $\Pi = \{V_1, V_2, \dots, V_\omega\}$ (a collection of subsets of V), a vector $R = (p_1, p_2, \dots, p_\omega)$ of requirements for each group, and an integer k.
Output: A sequence $(v_1, v_2, \dots, v_k)$, where $v_i \in V$, and a set of integers $J \subseteq [\omega]$ such

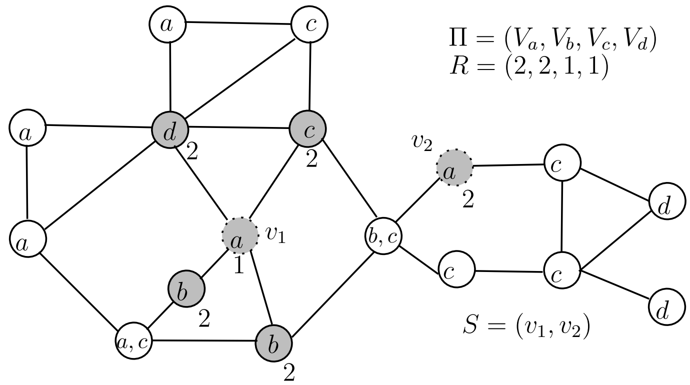

Figure 2: Burning with Groups example: An example of graph $G = (V, E)$ having four groups $\{V_a, V_b, V_c, V_d\}$ and a requirements vector $R = (2, 2, 1, 1)$. The letters inside the node indicate the groups to which the node belongs. The burning sequence $S = (v_1, v_2)$ enables to satisfy all the requirements. Burned nodes are depicted in grey, source nodes are depicted with a dotted border. The number close to the node denotes the time step in which the node is burned.

that

$$\left|\left(\bigcup_{i=1}^{k} N_{k-i}[v_i]\right) \bigcap V_j\right| \geq p_j, \quad \text{for each } j \in J \text{ and } |J| \text{ is maximized.}$$

Given a sequence $(v_1, v_2, \ldots, v_k)$, in the following we will say that the group V_j, for $j \in [\omega]$, is *satisfied* iff $\left|\left(\bigcup_{i=1}^{k} N_{k-i}[v_i]\right) \bigcap V_j\right| \geq p_j$. In words, in the BG problem, the goal is to find the sequence of the minimum length that satisfies all the groups, while in the BMG problem, given a budget of at most k sources, the goal is to maximize the number of satisfied groups.

1.1 Our results

This paper introduces and analyzes a generalization of the graph burning process [5] in order to model spreading processes in community-based networks. Then we define two optimization problems, which aim at maximizing the dissemination of information.

Both problems share the same decision problem which generalizes the decision version of the burning problem and therefore both problems are NP-hard. We investigate the Burning with Groups problem from an approximation point of view, showing both a lower bound and a matching upper bound. We also show that, when the number of communities is constant, the problem admits a constant approximation factor. Moreover, we show that the Burning Max Groups problem admits a 2-approximation when all requirements are unitary.

The preliminary version of this paper [20] considers the above problems where the groups do not overlap, that is, they are a collection of pairwise disjoint subsets of V.

2 Related works

Information spreading has been intensively studied in the context of viral marketing, which uses social networks to achieve marketing objectives through self-replicating viral processes, analogous to the spread of viruses. The goal here is to create a marketing message that can initially convince a selected set of people and then spread to the whole network in a short period of time [22] In general, an accurate information diffusion model can have many important impacts on predicting and managing emergent behaviors in lifelike complex networks such as: (i) suggesting appropriate strategies for boosting the impact of positive behaviors; (ii) measuring and evaluating the resilience and robustness of the network with respect to manipulations through information of questionable accuracy (e.g., fake news).

The problem of finding a source set of minimum size which, according to a spreading process, is able to spread a piece of information to the whole (or a fixed fraction of the) network, has its roots in the area of the spread of information in Social Networks.

The spread of viral information across a network naturally suggests many interesting optimization problems like Influence Maximization and Target Set Selection (see [8, 23] and references quoted therein). The first authors to study the spread of information in networks from an algorithmic point of view were Kempe et al. [27] who proposed two diffusion models named Linear Threshold and Independent Cascade. The Target Set Selection problem has been investigated in several papers [1, 4, 9, 12]. Subsequently, a set of papers has delved into distinct variations of the problem, presenting interesting perspectives [13, 15, 29]. Notably, the number of rounds necessary to spread a piece of information is considered in [10, 11, 24], while a more detailed diffusion model accounting for different categories of users is presented in [18, 19]. Moreover, being the problem hard in general, researchers have introduced several heuristics to address its challenges [14, 16, 17, 21, 30].

All the considered spreading processes are applied on a static snapshot of the network and ask for the identification of a set of initial adapters, which will be in charge of triggering the diffusion of the information. In a real setting, the network is dynamic, exhibiting structural changes over time. Algorithms must exhibit adaptability to accommodate these network transformations. On the other hand, observations indicate that while the network undergoes alterations, its fundamental backbone remains stable. Consequently, exploring algorithms that transcend individual interactions and operate on a broader scale becomes valuable. This approach ensures that localized changes in the network do not significantly harm the effectiveness of the algorithm.

Recently, the classical Influence Maximization problem has been revised with the aim of fetching the top influential users in social networks under a group influence perspective [25, 31, 32]. A social network is characterized by the existence of numerous communities, where nodes within each community share dense connections, while nodes across different communities have sparse connections. Indeed, social network users naturally group together based on shared interests, hobbies, or various relationships. Individuals can join multiple groups simultaneously. For example, a user might be a part of a family group, a work group, and a friends group. The assumption is that the interaction within a group is high and this favors the rapid dissemination of information inside it. Moreover, the whole group decision is based on some specific rules that determine an agreement on a concerned topic. The goal becomes reaching the desired agreement among all groups or maximizing the number of groups that reach the desired agreement. To address this objective, papers [25, 31, 32] consider the Group Influence Maximization (GIM) problem. In this problem, given a graph $G = (V, E)$, the diffusion process follows an independent cascade model. A group $U \subseteq V$ is activated if, during the diffusion process, at least $\beta|U|$ nodes in U are activated. The objective is to identify an initial set of at most k nodes that maximizes the number of influenced groups. Since the problem has been proved to be NP-hard, as it generalizes the Influence Maximization problem, three distinct heuristics are presented in [25, 31, 32].

In contrast, our approach differs as we use a distinct diffusion process (burning), which we believe is less susceptible to local network changes.

Closely related to our work is the COLORFUL K-CENTER PROBLEM studied in [3, 26].

> COLORFUL K-CENTER PROBLEM
> **Input:** A set P of n points in a metric space, and an integer k, a partition $\{P_1, P_2, \ldots, P_c\}$ of P into c color classes, and a coverage requirement $0 \le t_j \le |P_j|$ for each color class $j \in [c]$.
> **Output:** Find the smallest radius ρ such that using k balls of radius ρ, centered at points of P, we can simultaneously cover at least t_j points from each class P_j with $j \in [c]$.

Apart from the fact that the problem is defined in a metric space, the main difference with our BG problem is the fact that the number of sources/centers is fixed to k, while the radius of the neighbourhood (that is, the value to be minimized) is equal for each source, while in the BG problem the number of sources varies as well as the radius of the neighbourhood, because it depends on the position of the source in the output burning sequence.

3 Burning with Groups: The general case

3.1 Hardness results for BURNING WITH GROUPS

We show that BG cannot be approximated in polynomial time within a factor $c \log n$, for some constant $c < 1$, unless NP has quasi polynomial time (i.e., $NP \subset TIME(n^{O(\text{polylog } n)})$, where $TIME(t)$ denote the class of problems that admit a deterministic algorithm that runs in time t). To this aim, we provide an approximation preserving reduction from SET COVER.

> SET COVER (SC)
> **Input:** A universe $\mathcal{U} = \{1, 2, \ldots, n\}$ and a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$, whose union equals the universe.
> **Output:** A collection $\mathcal{C} \subseteq \mathcal{S}$ of minimum size such that $\bigcup_{C \in \mathcal{C}} C = \mathcal{U}$.

The following result has been proved in [2].

Theorem 1 *[2]* SET COVER *cannot be approximated, in polynomial time, within a factor* $(\log n)/48$, *unless NP has quasi polynomial time.*

Remark 1 *We remark that in the instance of* SET COVER *produced by the reduction in the proof of Theorem 1, the number of subsets (m) and the size of the universe (n) are polynomially related (i.e., $m \approx n^a$, for some constant $a > 0$).*

Theorem 2 B*G cannot be approximated, in polynomial time, to a factor $c \log n$ where $c < 1$ is a certain constant, unless NP has quasi polynomial time, even if all the requirements are equal to 1.*

Proof: We give an approximation preserving reduction from SET COVER.
The theorem will follow by Theorem 1 and Remark 1, since the construction below provides a graph $G = (V, E)$ having $|V| = O(n \times m)$ nodes.

To our aim, given a SC instance $\langle \mathcal{U}, \mathcal{S} \rangle$, we construct an instance $\langle G, \Pi, R \rangle$ of BG. Let $|\mathcal{U}| = n$ and $\mathcal{S} = \{C_1, \ldots, C_m\}$. We build the graph $G = (V, E)$ where V is partitioned into the disjoint sets $V_0, V_1, \ldots, V_{n+2}$ (i.e., the group family is $\Pi = \{V_0, V_1, \ldots, V_{n+2}\}$) and where all the group requirements are fixed to 1 (i.e., $R = (1, 1, \ldots, 1)$). Formally, $G = (V, E)$ is build as follows:

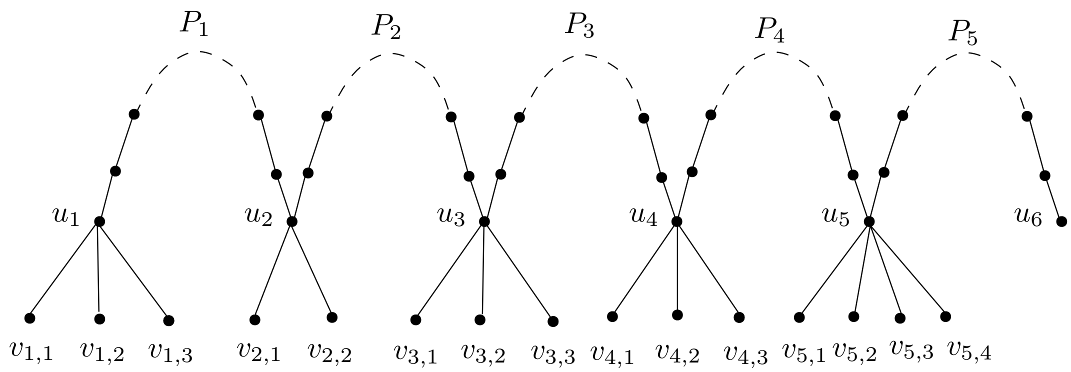

Figure 3: Representation of the graph G for the instance of SC $\langle \mathcal{U} = \{1,2,3,4,5,6,7,8,9\},\ \mathcal{S} = \{C_1 = \{1,2,3\},\ C_2 = \{3,6\}, C_3 = \{1,4,7\},\ C_4 = \{7,8,9\}, C_5 = \{2,4,5,6\}\}\rangle$.

- For any $C_i \in \mathcal{S}$, if $C_i = \{a_1, a_2, \ldots, a_r\}$ we add to G a star S_i of $r+1$ nodes $\{u_i, v_{i,1}, v_{i,2}, \ldots, v_{i,r}\}$ and edges $\{(u_i, v_{i,j}) \mid j \in [r]\}$. The center node u_i is assigned to the group V_0 and the leaf node $v_{i,j}$ is assigned to V_{a_j}, for $j \in [r]$.
- Then, we add a node u_{m+1}. We assign u_{m+1} to V_{n+1}.
- Finally, for each $i \in [m]$, we add a path P_i of length $2n+1$ connecting the node u_i to u_{i+1}; all the nodes of the these m paths are assigned to V_{n+2}.

For instance, figure 3 shows the graph G build for the SC instance $\langle \mathcal{U} = \{1,2,3,4,5,6,7,8,9\},\ \mathcal{S} = \{C_1 = \{1,2,3\},\ C_2 = \{3,6\}, C_3 = \{1,4,7\},\ C_4 = \{7,8,9\}, C_5 = \{2,4,5,6\}\}\rangle$ where $V_0 = \{u_1, u_2, u_3, u_4, u_5\}$, $V_1 = \{v_{1,1}, v_{3,1}\}$, $V_2 = \{v_{1,2}, v_{5,1}\}$, $V_3 = \{v_{1,3}, v_{2,1}\}$, $V_4 = \{v_{3,2}, v_{5,2}\}$, $V_5 = \{v_{5,3}\}$, $V_6 = \{v_{2,2}, v_{5,4}\}$, $V_7 = \{v_{4,1}\}$, $V_8 = \{v_{4,2}\}$, $V_9 = \{v_{4,3}\}$, $V_{10} = \{u_6\}$ and V_{11} comprises all the nodes in the paths P_1, P_2, P_3, P_4, P_5.

The following claim implies the correctness of the reduction.

The instance $\langle \mathcal{U}, \mathcal{S}\rangle$ of SC *admits a solution of size k if and only if the instance $\langle G, \Pi, R\rangle$ of* BG *admits a solution of size $k+1$.*

Assume that $\mathcal{C} \subseteq \mathcal{S}$ is a solution of the SC problem and $|\mathcal{C}| = k$. Let $\mathcal{C} = \{C_{b_1}, \ldots, C_{b_k}\}$. We will prove that the sequence

$$U = (u_{b_1}, u_{b_2}, \ldots, u_{b_k}, u_{m+1})$$

consisting of the centers of the stars $S_{b_1}, S_{b_2}, \ldots, S_{b_k}$ and node u_{m+1}, in this order, satisfies the requirements of all the groups. Recalling that all the requirements are set to 1, we have to prove that at least one node of each group is burned to retain the groups satisfied.
Indeed, since the center node of each star belongs to V_0 and our sequence U contains $k \geq 1$ center nodes, we have that V_0 is satisfied. We also note that since $k \geq 1$, then $N_k[u_{b_1}]$ includes at least one node of the path P_{b_1}, and since the nodes of P_{b_1} are assigned to the group V_{n+2}, we have that also group V_{n+2} is satisfied. Furthermore, since $V_{n+1} = \{u_{m+1}\}$ and node u_{m+1} is a node in U, also V_{n+1} is satisfied. Finally, the requirements of all the remaining groups $V_1, V_2, \ldots, V_n$ are satisfied because $N[u_{b_i}] \subseteq N_{k+1-i}[u_{b_i}]$, for each $i \in [k]$, and since $\mathcal{C} = \{C_{b_1}, \ldots, C_{b_k}\}$ is a solution of the SC instance we have $\left(\bigcup_{i=1}^{k} N[u_{b_i}]\right) \cap V_j \neq \emptyset$ for each $j \in [n]$.

Hence the sequence U, that has size $k+1$, is a solution of the BG problem.

Assume now that the sequence $U=(v_{\ell_1},\dots,v_{\ell_{k+1}})$ is a solution of the BG instance. We will show that we can always find a sequence of length $k+1$ that is a solution of the BG instance consisting of k centers of stars and where u_{m+1} is the last source in the sequence. This will allow us to construct a solution of size k of the SC instance. We first make considerations that will be useful in the following. By the construction of graph G we have $d(u_{m+1},u_\ell)\geq 2n+1$ for each $u_\ell\in V_0$ and any node $v\in N_k[u_{m+1}]-\{u_{m+1}\}$ is a node in the path P_m. Hence, for any $1\leq i\leq k$ it holds

$$u_{m+1}\notin N_{k+1-i}[u_\ell] \qquad \text{and} \qquad N_{k+1-i}[u_{m+1}]\subseteq (V_{n+1}\cup V_{n+2}). \tag{2}$$

Furthermore, since the length of each path P_i, for $1\leq i\leq m$, connecting u_i with u_{i+1} is $2n+1$, we have that if v is a node in P_i then exactly one between u_i and u_{i+1} is in $N_n[v]$ and if otherwise v is a leaf in the star S_i then u_i is the only center node in $N_n[v]$. Hence,

$$|N_n[v]\cap(V_0\cup V_{n+1})|=1 \text{ for each } v\notin V_0\cup V_{n+1}. \tag{3}$$

Given the sequence $U=(v_{\ell_1},\dots,v_{\ell_{k+1}})$, let v_{ℓ_i} be the source that enables to burn u_{m+1}, that is $u_{m+1}\in N_{k+1-i}[v_{\ell_i}]$, for some $1\leq i\leq k$. If $v_{\ell_i}\neq u_{m+1}$ then $i<k+1$ and since $k+1\leq n$ we have $N_{k+1-i}[v_{\ell_i}]\subseteq(V_{n+1}\cup V_{n+2})$ (i.e., v_{ℓ_i} is a node in the path P_m). Hence replacing v_{ℓ_i} with u_{m+1} in the sequence U, we obtain a sequence that is again a solution of the BG instance. Furthermore, we can assume that u_{m+1} is the last node in the sequence U. Indeed, let on the contrary u_{m+1} be the i-th node in U, for some $1\leq i\leq k$. By (3) either $v_{\ell_{k+1}}$ is a center of a star (in this case we set $u=v_{\ell_{k+1}}$), or there exists a unique center of star, that we call u, at distance at most n from $v_{\ell_{k+1}}$. Replacing $v_{\ell_{k+1}}$ with u and exchanging u with u_{m+1} in the sequence U, we obtain a sequence that is again a solution of the BG instance. This can be seen considering that $N_0[v_{\ell_{k+1}}]=\{v_{\ell_{k+1}}\}\subset N_{k+1-i}[u]$ (i.e., among the nodes burned by u, using it as i-th source, there is $v_{\ell_{k+1}}$ and some more nodes) and that one of the neighbors of u belongs to V_{n+2} (i.e., group V_{n+2} is satisfied by u when we use it as i-th source, as like, by (2), V_{n+2} was the only group, a part V_{n+1}, satisfied by u_{m+1} when it was used as i-th source). Hence, from here on we assume $U=(v_{\ell_1},\dots,v_{\ell_k},u_{m+1})$.

While $U=(v_{\ell_1},\dots,v_{\ell_k},u_{m+1})$ contains at least one node $v_{\ell_i}\notin V_0\cup V_{n+1}$, we iterate the following replacing procedure.

> Let $v_{\ell_i}\notin V_0\cup V_{n+1}$. By (3), let u be the only node in $V_0\cup V_{n+1}$ such that $u\in N_n[v_{\ell_i}]$. Then, since $k\leq n$, for each $0\leq j\leq n+2$,
>
> $$\text{if} \quad N_{k+1-i}[v_{\ell_i}]\cap V_j\neq\emptyset \quad \text{then} \quad N_{k+1-i}[u]\cap V_j\neq\emptyset.$$
>
> Hence replacing v_{ℓ_i} with u in the sequence U, we obtain a sequence U' that is again a solution of the BG instance.

After the iteration of the replacing procedure, we can assume that $U=(u_{\ell_1},\dots,u_{\ell_k},u_{m+1})$.
Finally, since each group V_j, for $1\leq j\leq n$ is satisfied, we have

$$\left|\left(\bigcup_{i=1}^{k} N_{k+1-i}[u_{\ell_i}]\right)\bigcap V_j\right|\geq 1, \qquad \text{for each } j\in[n],$$

and then

$$\left(\bigcup_{i=1}^{k} N_{k+1-i}[u_{\ell_i}]\right)\bigcap V_j\neq\emptyset, \qquad \text{for each } j\in[n]. \tag{4}$$

Since by the construction of G, $N_{k+1-i}[u_{\ell_i}] \setminus N[u_{\ell_i}] \subseteq V_{n+2}$, by (4) we have

$$\left(\bigcup_{i=1}^{k} N[u_{\ell_i}]\right) \bigcap V_j \neq \emptyset, \quad \text{for each } j \in [n].$$

Hence, let $\mathcal{C} = \{C_i \mid u_i \text{ is a node in } U, \text{ and } u_i \neq u_{m+1}\}$. We have that $\mathcal{C}$ is a solution of size k of the SC instance. □

3.2 Approximation Algorithms for Burning with Groups

We show that BG can be approximated to a factor $\log n + 1$. To this aim, we first define a novel maximization problem called Maximum Multi-Coverage Burning (MMCB) and show that it admits a 2-approximation. We will then use the 2-approximation for MMCB to obtain a $\log n + 1$ approximation for BG.

Maximum Multi-Coverage Burning (MMCB)

Input: $G = (V, E)$, a group family $\Pi = \{V_1, V_2, \ldots, V_\omega\}$ (a collection of subsets of V), a vector $L = (\ell_1, \ell_2, \ldots, \ell_\omega)$ of ω non-negative values and an integer k.

Output: A sequence $S = (v_1, v_2, \ldots, v_k)$, where $v_i \in V$, such that

$$\sum_{j=1}^{\omega} \min\left\{\left|\left(\bigcup_{i=1}^{k} N_{k-i}[v_i]\right) \bigcap V_j\right|, \ell_j\right\} \text{ is maximum.} \tag{5}$$

Let $S = (v_1, \ldots, v_h)$ be a sequence of $1 \leq h \leq k$ nodes. We define $N_{S,k} = \bigcup_{i=1}^{h} N_{k-i}[v_i]$ as the set of all the nodes burned by S by the k-th round and

$$f(S,k) = \sum_{j=1}^{\omega} \min\left\{\left|\left(\bigcup_{i=1}^{h} N_{k-i}[v_i]\right) \cap V_j\right|, \ell_j\right\} = \sum_{j=1}^{\omega} \min\left\{|N_{S,k} \cap V_j|, \ell_j\right\}. \tag{6}$$

When S is an empty sequence, we have $N_{S,k} = \emptyset$ and $f(S,k) = 0$. In the following, we will use N_S and $f(S)$, instead of $N_{S,k}$ and $f(S,k)$, whenever the value of k is clear from the context. Notice that MMCB asks for a sequence S of size k that maximizes $f(S)$.

Moreover, let $S = (v_1, \ldots, v_h)$ be a sequence of $1 \leq h \leq k$ nodes, let $i \in [k]$ and let $v \in V$ be a node, we define

$$f_i(v \mid S) = \sum_{j=1}^{\omega} \min\{|(N_{k-i}[v] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\}, \tag{7}$$

the gain that is provided by burning the neighborhood of radius $k - i$ around v, assuming that nodes in N_S are already burned.

When S is an empty sequence, we have $f_i(v \mid S) = \sum_{j=1}^{\omega} \min\{|N_{k-i}[v] \cap V_j|, \ell_j\}$.

By equation (7) we can easily observe that given two sequences S and S' such that $N_S \subseteq N_{S'}$ and a node $v \in V$ we have

$$f_i(v \mid S) \geq f_i(v \mid S') \text{ for each } i \in [k]. \tag{8}$$

The following properties will be useful to prove the approximation factor.

Algorithm 1 MMCB Algorithm(G, Π, L, k)

1: $S = ()$
2: **for** $i = 1$ to k **do**
3: $\quad v = \text{argmax}_{u \in V} f_i(u \mid S)$ $\triangleright$ v is the node providing the maximum gain.
4: $\quad S(i) = v$
5: **end for**
6: **return** S

Lemma 1 *Let $S = (v_1, v_2, \ldots, v_{h-1})$ be a sequence of $h-1$ nodes and let $S \perp v = (v_1, v_2, \ldots, v_{h-1}, v)$ be the sequence obtained by queuing v at S. We have*

$$f(S \perp v) - f(S) = f_h(v \mid S). \tag{9}$$

Proof: To prove (9) we are going to prove that for each $j \in [\omega]$,

$$\min\{|N_{S\perp v} \cap V_j|, \ell_j\} - \min\{|N_S \cap V_j|, \ell_j\} = \min\{|(N_{k-h}[v] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\}. \tag{10}$$

Observing that $|N_{S\perp v} \cap V_j| = |(N_{k-h}[v] \setminus N_S) \cap V_j| + |N_S \cap V_j|$, we have three cases to consider:

case 1: ($\ell_j \leq |N_S \cap V_j|$). In this case, the equation (10) becomes $\ell_j - \ell_j = 0$.

case 2: ($|N_S \cap V_j| < \ell_j \leq |N_{S\perp v} \cap V_j|$). In this case the equation (10) becomes

$$\ell_j - |N_S \cap V_j| = \min\{|(N_{k-h}[v] \setminus N_S) \cap V_j|, \ell_j - |N_S \cap V_j|\}.$$

Moreover, we have,

$$\ell_j \leq |N_{S\perp v} \cap V_j| = |(N_{k-h}[v] \setminus N_S) \cap V_j| + |N_S \cap V_j|$$

and consequently,

$$\min\{|(N_{k-h}[v] \setminus N_S) \cap V_j|, \ell_j - |N_S \cap V_j|\} = \ell_j - |N_S \cap V_j|.$$

case 3: ($\ell_j > |N_{S\perp v} \cap V_j|$). In this case the equation (10) becomes $|N_{S\perp v} \cap V_j| - |N_S \cap V_j| = |(N_{k-h}[v] \setminus N_S) \cap V_j|$.

□

Given the instance $\langle G, \Pi, L, k\rangle$ of MMCB, Algorithm 1 proceeds by iteratively adding nodes to the sequence S. At each iteration i, for $i \in [k]$, the node v to be added to the sequence is greedily chosen to give with its $(k-i)$-neighborhood the maximum contribution to the sum in (5), see line 3.

Let $S = (v_1, v_2, \ldots, v_k)$ be the solution provided by Algorithm 1 on the instance $\langle G, \Pi, L, k\rangle$ and let $O = (u_1, u_2, \ldots, u_k)$ be an optimal solution for the MMCB problem on the same instance $\langle G, \Pi, L, k\rangle$.

Denote by $S_i = (v_1, \ldots, v_i)$ the sequence constructed by Algorithm 1 by the end of the i-th step, for $i \in [k]$. We denote by $S_0 = ()$ the empty sequence and recall that $f(S_0) = 0$. Let v_i be the node selected at step i.

Lemma 2 *For $i \in [k]$,* $\quad f_i(v_i \mid S_{i-1}) \geq f_i(u_i \mid S)$.

Proof: If $u_i = v_a$ for some $a < i$ (i.e., u_i has already been chosen by the algorithm at step a), then $N_{k-i}[u_i] \subseteq N_{k-a}[u_i] \subseteq N_{S_{i-1}} \subseteq N_S$ and consequently $f_i(u_i \mid S) = 0$. Otherwise, when the algorithm picked v_i, the node u_i was available. However, the algorithm picked v_i because the contribution of v_i was at least equal to the contribution of u_i that is $f_i(v_i \mid S_{i-1}) \geq f_i(u_i \mid S_{i-1})$. Since $N_{S_{i-1}} \subseteq N_S$, by equation (8), the lemma follows. □

Lemma 3 $\sum_{i=1}^{k} f_i(u_i \mid S) \geq f(O) - f(S)$.

Proof: We have

$$\begin{aligned}
\sum_{i=1}^{k} f_i(u_i \mid S) &= \sum_{i=1}^{k}\sum_{j=1}^{\omega} \min\left\{|(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \\
&= \sum_{j=1}^{\omega}\sum_{i=1}^{k} \min\left\{|(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \\
&\overset{(i)}{\geq} \sum_{j=1}^{\omega} \min\left\{\sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\},
\end{aligned}$$

where (i) holds for the following arguments:

- If for each $i \in [k]$, it holds $|(N_{k-i}[u_i] \setminus N_S) \cap V_j| \leq \max\{0, \ell_j - |N_S \cap V_j|\}$ then we have

$$\begin{aligned}
&\sum_{i=1}^{k} \min\left\{|(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \\
&\quad = \sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j| \\
&\quad \geq \min\left\{\sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\}.
\end{aligned}$$

- If, otherwise there exist $a \in [k]$ such that $|(N_{k-a}[u_a] \setminus N_S) \cap V_j| > \max\{0, \ell_j - |N_S \cap V_j|\}$ then $\sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j| \geq |(N_{k-a}[u_a] \setminus N_S) \cap V_j| > \max\{0, \ell_j - |N_S \cap V_j|\}$. Hence,

$$\begin{aligned}
&\sum_{i=1}^{k} \min\left\{|(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \\
&\quad = \max\{0, \ell_j - |N_S \cap V_j|\} + \sum_{i=1, i\neq a}^{k} \min\left\{|(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \\
&\quad \geq \max\{0, \ell_j - |N_S \cap V_j|\} \\
&\quad = \min\left\{\sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\}.
\end{aligned}$$

Noticing that

$$\sum_{i=1}^{k} |(N_{k-i}[u_i] \setminus N_S) \cap V_j| \geq \left|\left(\bigcup_{i=1}^{k} N_{k-i}[u_i] \setminus N_S\right) \cap V_j\right| = |(N_O \setminus N_S) \cap V_j|$$

we get

$$\sum_{i=1}^{k} f_i(u_i \mid S) \geq \sum_{j=1}^{\omega} \min\left\{ \left|(N_O \setminus N_S) \cap V_j\right|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\}.$$

To prove the Lemma, we are going to show that for each $j \in [\omega]$,

$$\min\left\{ \left|(N_O \setminus N_S) \cap V_j\right|, \max\{0, \ell_j - |N_S \cap V_j|\}\right\} \geq \min\left\{|N_O \cap V_j|, \ell_j\right\} - \min\left\{|N_S \cap V_j|, \ell_j\right\}. \quad (11)$$

If $|N_O \cap V_j| \leq |N_S \cap V_j|$ then the right side of the inequality is at most 0 and there is nothing to prove. Now we consider $|N_O \cap V_j| > |N_S \cap V_j|$ and observing that $|(N_O \setminus N_S) \cap V_j| \geq |N_O \cap V_j| - |N_S \cap V_j|$, we have three cases to consider:

case 1: $(\ell_j \leq |N_S \cap V_j|)$. In this case the equation (11) becomes $0 \geq \ell_j - \ell_j$.

case 2: $(|N_S \cap V_j| < \ell_j \leq |N_O \cap V_j|)$. In this case the equation (11) becomes

$$\min\{|(N_O \setminus N_S) \cap V_j|, \ell_j - |N_S \cap V_j|\} \geq \ell_j - |N_S \cap V_j|.$$

Moreover, we have,

$$\ell_j \leq |N_O \cap V_j| \leq |(N_O \setminus N_S) \cap V_j| + |N_S \cap V_j|$$

and consequently,

$$\min\{|(N_O \setminus N_S) \cap V_j|, \ell_j - |N_S \cap V_j|\} = \ell_j - |N_S \cap V_j|.$$

case 3: $(\ell_j > |N_O \cap V_j|)$. In this case the equation (11) becomes

$$\min\left\{ \left|(N_O \setminus N_S) \cap V_j\right|, \ell_j - |N_S \cap V_j|\right\} \geq |N_O \cap V_j| - |N_S \cap V_j|.$$

Moreover, we have,

$$\begin{aligned}\min\left\{ \left|(N_O \setminus N_S) \cap V_j\right|, \ell_j - |N_S \cap V_j|\right\} &\geq \min\left\{|N_O \cap V_j| - |N_S \cap V_j|, \ell_j - |N_S \cap V_j|\right\} \\ &= |N_O \cap V_j| - |N_S \cap V_j|.\end{aligned}$$

□

Theorem 3 MMCB *admits a* 2*-approximation algorithm.*

Proof: Let $S = (v_1, v_2, \ldots, v_k)$ be the solution provided by Algorithm 1 on the instance $\langle G, \Pi, L, k\rangle$ and let $O = (u_1, u_2, \ldots, u_k)$ be an optimal solution for the MMCB problem on the same instance $\langle G, \Pi, L, k\rangle$.

We have that,

$$\begin{aligned}f(S) &= (f(S) - f(S_{k-1})) + (f(S_{k-1}) - f(S_{k-2})) + \ldots + (f(S_1) - f(S_0)) + f(S_0) \\ &= f_k(v_k \mid S_{k-1}) + f_{k-1}(v_{k-1} \mid S_{k-2}) + \ldots + f_1(v_1 \mid S_0) \qquad \text{by Lemma 1} \\ &= \sum_{i=1}^{k} f_i(v_i \mid S_{i-1}) \\ &\geq \sum_{i=1}^{k} f_i(u_i \mid S) \qquad \text{by Lemma 2} \\ &\geq f(O) - f(S). \qquad \text{by Lemma 3} \qquad (12)\end{aligned}$$

Inequality (12) implies $f(S) \geq f(O)/2$. □

We show now how the 2-approximation bound for MMCB can be used to obtain a $\log n + 1$ approximation for BG.

Consider an instance $\langle G, \Pi, L, k\rangle$ of MMCB. Let $\ell = \sum_{j=1}^{\omega} \ell_j$. By definition (6), we have that, for any sequence S of size k:

(i) $f(S) \leq \ell$;

(ii) $f(S) = \ell$ if and only if $|N_S \cap V_j| \geq \ell_j$, for $j \in [\omega]$.

Hence, whenever $f(S) = \ell$, the sequence S also satisfies the BG instance $\langle G, \Pi, R = L\rangle$.

Similarly, let $\langle G, \Pi, R\rangle$ be an instance of BG, let $r = \sum_{j=1}^{\omega} p_j$ and let k^* be the size of the smallest sequence S^* that satisfies all the requirements (i.e., the optimal value for the given instance). Using the same sequence S^* we get $f(S^*, k^*) = r$ for the instance $\langle G, \Pi, R, k^*\rangle$ of MMCB and we have:

(i) the optimum value for the instance $\langle G, \Pi, R, k^*\rangle$ of MMCB is r;

(ii) if the optimum value for $\langle G, \Pi, R, k\rangle$ of MMCB is r then $k \geq k^*$.

Let $\langle G, \Pi, R\rangle$ be an instance of BG. For each $k = 1, 2, \ldots, n$, we execute the Algorithm 2 and take the smallest set obtained as the solution of the problem. Theorem 4 shows that, exploiting Algorithm 2 and the above properties, one can obtain the desired approximation factor for BG.

Theorem 4 BG *can be approximated to* $\log n + 1$.

Proof: For a given k the obtained sequence $S = S^1 \perp S^2 \perp \ldots \perp S^{t_k}$ has length at most $k \times t_k$. We show now that by choosing the value k such that $k \times t_k$ is minimum, one can get the desired approximation factor.

Let k^* denote the value of an optimal solution $S_{\text{BG}} = (v_1, \ldots, v_{k^*})$ for BG instance $\langle G, \Pi, R\rangle$. Clearly $f(S_{\text{BG}}, k^*) = r$. Hence, when the Algorithm 2 is executed with $k = k^*$, we know, by Theorem 3 and the above properties, that the greedy Algorithm 1 will compute a sequence S^1 such that $f(S^1, k^*) \geq r/2$. By iterating the greedy Algorithm 1 t times, we get the sequences $S = S^1 \perp S^2 \perp \ldots \perp S^t$ (that is S is the concatenation of $S^1, S^2, \ldots, S^t$) providing a solution for the instance $\langle G, \Pi, R, k^* \times t\rangle$ of MMCB of value at least

$$f(S, k^* \times t) \geq \sum_{i=1}^{t} f(S^i, k^*) \geq r\left(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^t}\right) = r\left(1 - \frac{1}{2^t}\right).$$

Hence, for some $t \leq \log r + 1$, the algorithm will find a sequence S such that $f(S, k^* \times t) \geq r$ and the corresponding burning time is

$$|S| \leq k^* \times t \leq k^* \times (\log r + 1) \leq k^* \times (\log n + 1).$$

□

Algorithm 2 BG Algorithm$(G = (V, E), \Pi, R, k)$

$h = 1,\ \Pi^1 = \Pi,\ R^1 = R,\ V^1 = V$
$S^1 = MMCB(G, \Pi^1, R^1, k)$
while $\left(\sum_{i=1}^{h} f(S^i) < r\right)$ **do** ▷ $r = \sum_{j=1}^{\omega} p_j$
  $h = h + 1$
  $V^h = V^{h-1} \setminus N_{S^{h-1}}$ ▷ V^h is the set of nodes not burned by $S^1, S^2, \ldots, S^{h-1}$
  $\Pi^h = \{V_1^h, \ldots, V_\omega^h\}$ ▷ $V_j^h = V^h \cap V_j$, for $j \in [\omega]$
  $R^h = (p_1^h, \ldots, p_\omega^h)$ ▷ $p_j^h = \max\{0, p_j^{h-1} - |N_{S^{h-1}} \cap V_j|\}$, for $j \in [\omega]$
  $S^h = MMCB(G, \Pi^h, R^h, k)$
end while
return $S = S^1 \perp S^2 \perp \ldots \perp S^h$ ▷ the concatenation of $S^1, S^2, \ldots, S^h$

4 Burning with O(1) groups

In this section, we assume that $\omega = O(1)$ and obtain a constant approximation factor for the BG problem. Our solution goes through the following problem:

Square Domination with Groups (SDG)

Input: $G = (V, E)$, a group family $\Pi = \{V_1, V_2, \ldots, V_\omega\}$ (a collection of subsets of V), and a vector $R = (p_1, p_2, \ldots, p_\omega)$ of requirements for each group.

Output: A set $\{v_1, v_2, \ldots, v_k\} \subseteq V$ of minimum size k such that
$\left|\left(\bigcup_{i=1}^{k} N_k[v_i]\right) \bigcap V_j\right| \geq p_j$, for each $j \in [\omega]$.

SDG differs from BG because in SDG each source covers a neighbourhood of the same radius k and consequently the order of nodes in the solution does not matter, while in BG the radius of the neighbourhood depends on the position of the source in the sequence.

We notice that given any sequence S, solution for BG on $\langle G, \Pi, R\rangle$, then the set containing the nodes in S is a solution for SDG for the same instance. Moreover, from any solution $\{v_1, v_2, \ldots, v_k\}$ to SDG on $\langle G, \Pi, R\rangle$ we can get a solution for BG, of size $2k$, for the same instance as any sequence $(v_1, \ldots, v_k, u_1, \ldots, u_k)$ where $u_1, \ldots, u_k$ are k arbitrary chosen nodes. Hence, denoted by O_B and O_D the sizes of the optimal solution of the BG problem and the SDG problem, respectively, we have $O_D \leq O_B \leq 2O_D$.

In the following, we will show how to obtain, from this observation, a polynomial time algorithm for the BG problem, whose solution is upper bounded by $3O_B + \omega - 1$.

We assume that $O_D > \omega$, otherwise $O_D = O(1)$ and we are able to find the optimal solution for the SDG problem in polynomial time by simply enumerating over all subsets of V of size O_D, which results in an exact algorithm having running time $|V|^{O(O_D)} = |V|^{O(1)}$.

We are going to use a result in [3], where the authors describe a pseudo-approximation algorithm for the Colorful k-center problem (with two colors). They show how to find a solution of radius at most $2r^*$ using at most $k+1$ centers (i.e., sources), where r^* is the optimum radius for the considered problem.

Let k-SDG be the decision version of SDG for a given integer k, that is, the problem asking if there exists a subset $S \subseteq V$, with $|S| = k$ such that
$\left|\left(\bigcup_{v \in S} N_k[v]\right) \bigcap V_j\right| \geq p_j$, for each $j \in [\omega]$.

Algorithm 3 Clustering Algorithm(G, k, x, z)

$S = \emptyset, \quad V' = V$
while $V' \neq \emptyset$ and $\max_{v \in V'} z_v > 0$ **do**
  $v = \text{argmax}_{v' \in V'} z_{v'}$
  $S = S \cup \{v\}$
  $y_v = \min\{1, \sum_{u \in N_k[v]} x_u\}$
  $C_v = N_{2k}[v] \cap V'$
  $V' = V' \setminus C_v$
end while
return $S, \ \{C_v \mid v \in S\}, \ y$

We use the solution of the following natural LP relaxation for the k-SDG problem, which we name $LP_{k\text{-SDG}}$ (Fig. 4), to have a simplified LP version (Fig. 5). This will allow us to find a 2-approximation for the SDG problem.

Given a fractional solution (x, z) to $LP_{k\text{-SDG}}$, the variable x_u, for $u \in V$, represents the fraction of node u that is used as a source and z_v, for $v \in V$, represents the fraction of coverage that node v receives by the other (fractional) sources nodes (namely, nodes u at distance at most k from v with $x_u > 0$).

$LP_{k\text{-SDG}}$

$$\sum_{u \in N_k[v]} x_u \geq z_v \quad \forall v \in V$$
$$\sum_{u \in V} x_u \leq k$$
$$\sum_{u \in V_j} z_u \geq p_j \qquad \forall j \in [\omega]$$
$$x_u, z_u \in [0, 1] \qquad \forall u \in V$$

Figure 4: The natural LP relaxation for the k-SDG problem.

Following [3], we present an algorithm that, given a feasible solution (x, z) to $LP_{k\text{-SDG}}$, finds a clustering of the nodes of V and a subset S of cluster centers (nodes with $z_v > 0$), that we can use to write a simplified version of $LP_{k\text{-SDG}}$.

Let S and $\{C_v \mid v \in S\}$ be the sets returned by Algorithm 3. For any $v \in S$ and $j \in [\omega]$, let $P_{jv} = C_v \cap V_j$ be the set of nodes of the group V_j in the cluster C_v. Fix $p_{jv} = |P_{jv}|$. By using S and the values p_{jv}, for each $v \in S$ and $j \in [\omega]$, we define the linear program $LP'_{k\text{-SDG}}$ (Fig. 5).

The variable y_v in LP'_{k-SDG} represents the fraction of node v that is used as a source to cover p_{jv} nodes of the group V_j. The following Lemma shows that the vector y returned by Algorithm 3 is a feasible solution to LP'_{k-SDG}.

Lemma 4 *Given a feasible solution (x, z) to $LP_{k\text{-SDG}}$ and the sets $S \subseteq V$ and C_v, for $v \in S$ returned by Algorithm 3, the following proprieties hold:*

LP'_{k-SDG}

$$\max \sum_{v\in S} p_{1v} y_v$$

subject to:

$$\sum_{v\in S} p_{jv} y_v \geq p_j \quad \forall j = 2,\dots,\omega$$

$$\sum_{v\in S} y_v \leq k$$

$$y_v \in [0,1] \qquad \forall v \in S$$

Figure 5: A simplified LP relaxation for the k-SDG problem.

(i) The clusters C_v, for $v \in S$ are pairwise disjoint.
(ii) y is a feasible solution to $LP'_{k\text{-SDG}}$.

Proof: Since at each iteration of the while loop in Algorithm 3, each selected cluster C_v is removed from V' before choosing a new cluster among the nodes in V' at the next iteration, property (i) comes.

To prove (ii) we first show that:

$$\text{For each } u \in V \text{ there is at most one } v \in S \text{ such that } u \in N_k[v]. \tag{13}$$

Indeed, given any pair v, v' of nodes in S, by Algorithm 3, it holds $d(v, v') > 2k$. Hence, any node u such that $d(u, v) \leq k$ (i.e., $u \in N_k[v]$)) is such that $d(u, v') > k$ (i.e., $u \notin N_k[v']$)).

Then, we prove that:

$$y_v \geq z_v \text{ for each } v \in S. \tag{14}$$

Recalling that $y_v = \min\{1, \sum_{u\in N_k[v]} x_u\}$, it follows that if $y_v = 1$ then $y_v \geq z_v$ due to the fact that $z_v \in [0,1]$. If $y_v < 1$ then $y_v = \sum_{u\in N_k[v]} x_u \geq z_v$ where the last inequality is by the first constraint in $LP_{k\text{-SDG}}$.

Now, we are ready to prove that y is a feasible solution for $LP'_{k\text{-SDG}}$. To this aim, we check that the constraints in $LP'_{k\text{-SDG}}$ are satisfied. For each $j \in [\omega]$ we have

$$\begin{aligned}
\sum_{v\in S} p_{jv} y_v &= \sum_{v\in S} |P_{jv}|\, y_v \\
&= \sum_{v\in S}\sum_{u\in P_{jv}} y_v \\
&\geq \sum_{v\in S}\sum_{u\in P_{jv}} z_v && \text{by (14)} \\
&\geq \sum_{v\in S}\sum_{u\in P_{jv}} z_u && \text{by the greedy choice of } v \text{ in Alg. 3} \\
&= \sum_{u\in V_j} z_u && \text{by (i)} \\
&\geq p_j. && \text{by the third constraint in } LP_{k\text{-SDG}}
\end{aligned}$$

Finally, we check that $\sum_{v \in S} y_v \leq k$.

$$\begin{aligned} \sum_{v \in S} y_v &\leq \sum_{v \in S} \sum_{u \in N_k[v]} x_u && \text{by the setting of } y_v \text{ in Alg. 3} \\ &\leq \sum_{u \in V} x_u && \text{by (13)} \\ &\leq k. && \text{by the second constraint in } LP_{k\text{-SDG}} \end{aligned}$$

□

As $LP'_{k\text{-SDG}}$ has only ω non-trivial constraints, any extreme point will have at most ω variables attaining strictly fractional values [28]. So by the second constraint in $LP'_{k\text{-SDG}}$ at most $k - 1 + \omega$ variables of any feasible y are non-zero.

So, we choose to round up the fractional variables y_v to 1 since the coverage of each group V_i for $i \in [\omega]$ can only increase, and set $S' = \{v \in S \mid y_v > 0\}$. From the above, we get $|S'| = k - 1 + \omega$. Recalling that for each node $v \in S'$ it holds $C_v \subseteq N_{2k[v]}$ and that $k > \omega$ we have that $|S'| = k - 1 + \omega < 2k - 1$ and S' is a solution for the k-SDG problem.

Recalling by the definition of the k-SDG problem that $k \leq diam(G)$, we can repeat the above procedure for $k = 1, \cdots, diam(G)$, and stop at the smallest value of k for which a solution S' is possible. Now, we notice that, if O_D is the size of the optimal solution of the SDG problem, then $k \leq O_D$, since an optimal solution S_{O_D} for the SDG problem satisfies all the constraints of $LP_{k\text{-SDG}}$ whenever $k \geq O_D$. Hence the following theorem follows.

Theorem 5 *For $\omega = O(1)$, there exists a polynomial time algorithm that finds a 2-approximation for the* SDG *problem.*

By observing that the above strategy enables to identify a set of $k - 1 + \omega$ sources such that their $2k$-neighborhoods satisfy all the requirements, we have that one can obtain a sequence S, solution of the BG problem, by taking all the above sources in any order followed by other arbitrarily chosen $2k$ nodes. Hence, $|S| = 3k + \omega - 1$ and recalling that $k \leq O_D \leq O_B$ the following result holds.

Theorem 6 *For $\omega = O(1)$, there exists a polynomial time algorithm to find a solution for the* BG *problem whose size is upper bounded by $3O_B + \omega - 1$.*

Recalling that the Graph Burning problem is a special case of BG in which all the nodes form a single group (i.e., $\omega = 1$) with requirement n, the above result generalizes the 3-approximation for the Graph Burning [6].

5 Burning Max Groups with unitary requirements

When all the requirements are equal to one, an instance of Burning Max Groups (BMG) can be seen as an instance of the following problem [7].

> MCG
> **Input:** $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ partitioned into ℓ subsets $G_1, G_2, \ldots, G_\ell$ (where each set S_i is a subset of a given ground set $\mathcal{X}$), a global bound k and ℓ subset bounds k_i, for each G_i.
> **Output:** $H \subseteq \mathcal{S}$ such that $|H| \leq k$ and $|H \cap G_i| \leq k_i$ for $i \in [\ell]$, and $|\bigcup_{S \in H} S|$ is maximized.

Indeed given an instance $\langle G, \Pi = \{V_1, V_2, \ldots, V_\omega\}, R, k\rangle$ of the BMG problem, where all the requirements are fixed to 1 (i.e., $R = (1, \ldots, 1)$), we can set $\mathcal{X} = \{1, 2, \ldots, \omega\}$ and $\ell = k$. For each $i \in [k]$, and for each $v \in V$ let $S_{i,v} = \{j \mid N_{k-i}[v] \cap V_j \neq \emptyset\}$, we set $\mathcal{S} = \{S_{i,v} \mid v \in V, i \in [k]\}$ and for each $i \in [k]$, $G_i = \{S_{i,v} \mid v \in V\}$. Finally set $k_i = 1$, for each $i \in [k]$. Since MCG admits a 2-approximation algorithm [7], we have the following results.

Theorem 7 *There exists a polynomial time algorithm to find a* 2*-approximation for the Burning Max Groups problem with unitary requirements.*

Open Problem: Can the above result be extended to the BMG problem in the general case?

By considering each node as a separate subset, the BMG problem becomes

Max Burning (MB)
Input: $G = (V, E)$ and a integer k.
Output: A sequence of nodes $(v_1, v_2, \ldots, v_k)$, such that maximizes $\left|\bigcup_{i=1}^{k} N_{k-i}[v_i]\right|$.

Theorem 8 *There exists a polynomial time algorithm to find a* 2*-approximation for the* Max Burning *problem.*

6 Conclusion

We introduced a variation of the graph burning process in order to study the spreading of information in community-based networks, with or without overlapping communities. On top of this diffusion process, we investigated the Burning with groups problem, with the aim of maximizing the dissemination of information at a minimum cost. This NP-hard problem has been analyzed from an approximation point of view, showing both a lower bound and a matching upper bound. We also showed that when the number of communities is constant the problem admits a constant approximation factor. Eventually, we presented a preliminary analysis of the maximization version of the problem, named Burning Max Groups, showing that it admits a 2-approximation, provided that all requirements are unitary. It would be interesting to investigate whether this result can be extended to the general case.

References

[1] E. Ackerman, O. Ben-Zwi, and G. Wolfovitz. Combinatorial model and bounds for target set selection. *Theor. Comp. Sci.*, 411(44–46):4017–4022, 2010. doi:10.1016/j.tcs.2010.08.021.

[2] S. Arora and C. Lund. *Hardness of Apagesroximation*, volume Apagesroximation Algorithms for NP-Hard Problems. Ed. D. Hochbaum. PWS Publishers, 1995.

[3] S. Bandyapadhyay, T. Inamdar, S. Pai, and K. Varadarajan. A constant apagesroximation for colorful k-center. In *arXiv*, 2019. URL: https://arxiv.org/pdf/1907.08906.pdf.

[4] O. Ben-Zwi, D. Hermelin, D. Lokshtanov, and I. Newman. Treewidth governs the complexity of target set selection. *Discrete Optim.*, 8(1):87–96, 2011. doi:0.1016/j.disopt.2010.09.007.

[5] A. Bonato. A survey of graph burning. *Contributions to Discrete Mathematics*, 16:185–197, 2021. doi:10.11575/CDM.V16I1.71194.

[6] A. Bonato and S. Kamali. Apagesroximation algorithms for graph burning. In *Proc. of Theory and Apageslications of Models of Computation (TAMC 2019)*, volume 11436 of *Lecture Notes in Computer Science*, 2019. doi:10.1007/978-3-030-14812-6_6.

[7] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and apageslications. In *Proc. of RANDOM ApagesROX '2004*, volume 3122 of *Lecture Notes in Computer Science*, 2004. doi:10.1007/s10878-016-0102-0.

[8] W. Chen, L. Lakshmanan, and C. Castillo. *Information and influence propagation in social networks*, volume Synthesis Lectures on Data Management 5.4. Springer, 2013.

[9] M. Chopin, A. Nichterlein, and M. W. R. Niedermeier. Constant thresholds can make target set selection tractable. *Theory Comput. Syst.*, 55(1):61–83, 2014. doi:10.1007/s00224-013-9499-3.

[10] F. Cicalese, G. Cordasco, L. Gargano, M. Milanic, J. Peters, and U. Vaccaro. Spread of influence in weighted networks under time and budget constraints. *Theoretical Computer Science*, 586:40–58, 2015. doi:10.1016/j.tcs.2015.02.032.

[11] F. Cicalese, G. Cordasco, L. Gargano, M. Milanic, and U. Vaccaro. Latency-bounded target set selection in social networks. *Theoretical Computer Science*, 535:1–15, 2014. doi:10.1016/j.tcs.2014.02.027.

[12] A. Coja-Oghlan, U. Feige, M. Krivelevich, and D. Reichman. Contagious sets in expanders. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pages. 1953–1987*, 2015. doi:10.5555/2722129.2722260.

[13] G. Cordasco, L. Gargano, M. Lafond, L. Narayanan, A. Rescigno, U. Vaccaro, and K. Wu. Whom to befriend to influence people. *Theoretical Computer Science*, 810:26–42, 2020. doi:10.1016/j.tcs.2018.05.030.

[14] G. Cordasco, L. Gargano, M. Mecchia, A. A. Rescigno, and U. Vaccaro. Discovering small target sets in social networks: a fast and effective algorithm. *Algorithmica*, 80:1804–1833, 2018. doi:10.1007/s00453-017-0390-5.

[15] G. Cordasco, L. Gargano, J. Peters, A. Rescigno, and U. Vaccaro. Fast and frugal targeting with incentives. *Theoretical Computer Science*, 812:62–79, 2020. doi:10.1016/j.tcs.2019.07.007.

[16] G. Cordasco, L. Gargano, and A. Rescigno. Influence propagation over large scale social networks. In *Proc. of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining ASONAM'15, pages. 1531–1538*, 2015. doi:10.1145/2808797.2808888.

[17] G. Cordasco, L. Gargano, and A. Rescigno. On finding small sets that influence large networks. *Social Network Analysis and Mining*, 6(94), 2016. doi:10.1007/s13278-016-0408-z.

[18] G. Cordasco, L. Gargano, and A. Rescigno. Active influence spreading in social networks. *Theoretical Computer Science*, 764:15–29, 2019. doi:10.1016/j.tcs.2018.02.024.

[19] G. Cordasco, L. Gargano, A. Rescigno, and U. Vaccaro. Evangelism in social networks: Algorithms and complexity. *Networks*, 71(4):346–357, 2018. doi:10.1002/net.21756.

[20] G. Cordasco, L. Gargano, and A. A. Rescigno. Groups burning: Analyzing spreading processes in community-based networks. In *17th International Conference and Workshops, WALCOM'23*, volume 13973 of *LNCS*, pages 330–342. Springer, 2023. doi:10.1007/978-3-031-27051-2_28.

[21] T. Dinh, H. Zhang, D. Nguyen, and M. Thai. Cost-effective viral marketing for time-critical campaigns in large-scale social networks. *IEEE/ACM Transactions on Networking*, 22(6):62–79, 2014. doi:10.1109/TNET.2013.2290714.

[22] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages. 57–66, New York, NY, USA*, 2001.

[23] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press, 2010.

[24] L. Gargano, P. Hell, J. Peters, and U. Vaccaro. Influence diffusion in social networks under time window constraints. *Theoretical Computer Science*, 584(C):53–66, 2015. doi:10.1016/j.tcs.2015.02.015.

[25] P. Huang, L. Guo, and Y. Zhong. Efficient algorithms for maximizing group influence in social networks. *Tsinghua Science and Technology*, 27(5):832–842, 2022. doi:10.26599/TST.2021.9010062.

[26] X. Jia, K. Sheth, and O. Svensson. Fair colorful k-center clustering. In *arXiv*, 2020. URL: https://arxiv.org/abs/2007.04059.

[27] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc of Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03), pages 137–146*, 2003. doi:10.1145/956750.956769.

[28] B. Lev and A. L. Soyster. Integer programming with bounded variables via canonical separation. *The Journal of the Oper. Research Society*, 29(2):477–488, 1978.

[29] T. T. Reddy and C. P. Rangan. Variants of spreading messages. *Journal of Graph Algorithms and Apageslications*, 15(5):683–699, 2011. doi:10.7155/jgaa.00244.

[30] P. Shakarian, S. Eyre, and D. Paulo. A scalable heuristic for viral marketing under the tipagesing model. *Social Network Analysis and Mining*, 3(4):1225–1248, 2013. doi:10.1007/s13278-013-0135-7.

[31] Y. Zhong, L. Guo, and P. Huang. Maximizing group coverage in social networks. In Y. Zhang, Y. Xu, and H. Tian, editors, *Parallel and Distributed Computing, Apageslications and Technologies*, pages 274–284. Springer International Publishing, 2021.

[32] J. Zhu, S. Ghosh, and W. Wu. Group influence maximization problem in social networks. *IEEE Trans. on Computational Social Systems*, 6(6):1156–1164, 2019. doi:10.1109/TCSS.2019.2938575.