The Graph Burning Problem under Constrained Diffusion

Enrico Iurlano[0000−0001−7528−0834] and
Günther R. Raidl[0000−0002−3293−177X] and
Marko Djukanović[0000−0003−1358−3789]

Abstract Relying on a simplified model of fire spread, the Graph Burning Problem is an NP-hard combinatorial optimization problem that yields a social contagion metric. It concerns a discrete-time process on a simple undirected graph, with a diffusion phase of fire in each timestep towards the entire neighborhoods of "burned" vertices and a phase in which a next not-yet-burned vertex is made burned. The aim is to find the minimum number of timesteps so that the complete graph gets burned. The applicability of the problem becomes more relevant in practice when one takes into account the fact that in the context of wildfire, virus, or information spreading, diffusion might realistically be prevented by implemented (local) countermeasures or some given limitations/obstacles. Therefore, this paper proposes the Constrained Diffusion Graph Burning Problem where thresholds per vertex are considered that specify the maximum number of neighbors the vertex can ignite. Additionally, the burned vertices are permitted to diffuse fire only immediately after their status changed to burned, but not to a later timestep. Two mixed integer linear programming formulations, both relying on a flow approach are proposed to tackle small to medium-sized instances. In contrast, a greedy heuristic is proposed for solving large problem instances.

Enrico Iurlano
Algorithms and Complexity Group, TU Wien, Favoritenstraße 9-11/192-01, Vienna, 1040, Austria
e-mail: eiurlano@ac.tuwien.ac.at

Günther R. Raidl
Algorithms and Complexity Group, TU Wien, Favoritenstraße 9-11/192-01, Vienna, 1040, Austria
e-mail: raidl@ac.tuwien.ac.at

Marko Djukanović
Faculty of Natural Sciences and Mathematics, University of Banja Luka, Mladena Stojanovića 2, Banja Luka, 78000, Bosnia and Herzegovina e-mail: marko.djukanovic@pmf.unibl.org

This document is a preprint. The contribution was presented at the Optimization and Wildfire Conference (O&W), Luso, Portugal, 3rd of October 2024.

1 Introduction

In 2014, Bonato et al. [3] proposed the discrete-time process of *graph burning* as a model of social contagion. This process relies on a highly simplified model of fire spread and originally was used to provide insights into the achievability of rapid information dissemination in social networks under certain temporal assumptions on the release of information. Due to similarities, e.g., to the *Target Set Selection Problem* [14] or the *Least Cost Influence Maximization Problem* (LCIMP) [12], the *Graph Burning Problem* (GBP) additionally provides an interesting alternative model for viral marketing or opinion-making. For a formal description of the graph burning process, we consider a simple and undirected input graph $G = (V, E)$ whose vertices $v \in V$ carry a time-dependent binary status indicating either *unburned* or *burned* [3]. Assuming (discrete) timesteps indexed by $t \in \mathbb{N} \cup \{0\}$, consider the following process.

- Initially, all vertices are unburned.
- In timestep $t = 0$ the status of a single vertex is changed to burned.
- In each timestep $t \geq 1$, the neighbors of all vertices that have the status burned in timestep $t - 1$ become burned as well. Note that some of them may already be burned.
- Within the same timestep t, an unburned vertex is picked and made burned (if such a one exists).

The process is iterated and stops when a timestep is reached after which all the vertices of the graph are burned. Each such processes is called an admissible burning process. The *burning number* $b(G)$ of G is defined as the minimum number of timesteps over all admissible burning processes, i.e., $b(G) := \min\{T+1 : \text{there is an admissible burning process on } G \text{ terminating with index } T\}$. Returning to the perspective of a social contagion model, a burned vertex could represent—within the setting of opinion-making or viral marketing—the scenario in which an individual has been convinced to adopt a certain opinion either by a direct influence of some external actor or by propagation from a neighboring individual that is already affected by that opinion. A small burning number for a social network reveals that it is possible to influence all individuals in a short amount of time.

Several combinatorial aspects of the graph burning number have been examined, e.g., it is known that its calculation is NP-hard even on trees of maximum degree three [1] or on caterpillars of maximum degree three [13, 16]; several other graph classes are mentioned in the literature in this context, see [2] for a partial overview. The central open problem in the field is stated in [3] claiming that for any graph $G = (V, E)$ the bound $b(G) \leq \lceil |V|^{1/2} \rceil$ holds; partial progress towards this so-called *burning number conjecture* has been recently obtained; in [18] it is proved that this conjecture holds asymptotically, and in [17] it is shown that the conjecture holds for any tree without vertices of degree two. The survey of Bonato [2] describes several further interesting computational findings on $b(G)$, in particular, established $b(G)$-numbers respectively approximation algorithms [4]; for general graphs, a 3-approximation algorithm is presented together with a 2-approximation algorithm

specifically designed for trees. A randomized 2.314-approximation algorithm for computing the burning number of a general graph is presented in [15]. In [9], an approximation algorithm termed "burning farthest-first" is proposed with a tighter approximation ratio of $3 - 2/b(G)$. It relies on the idea to iteratively incorporate locally-best vertices into an initially empty set of candidate vertices. The chosen vertex lies here among the distance maximizers to the set of burning vertices. A similar approach has been designed earlier in [7] but from a purely heuristic perspective. Several further heuristics for this problem are proposed in [19, 11] based on the eigencentrality as a measure for attractivity of a next vertex to be ignited. More recently in [8], based on the relationship between the GBP and the so-called *Clustered Maximum Coverage Problem*, the authors propose a greedy heuristic that effectively solves instances with up to hundreds of thousands of vertices. An integer linear programming formulation and a constraint programming approach are proposed in [10]. Recently, in [5] considerable progress has been achieved towards solving large solution instances optimally by a row generation approach.

The main contribution of this paper is the introduction and solution of a generalization of GBP where vertices can ignite a certain maximum number of neighbors leading to an in general larger search space of feasible solutions. We come up with two Mixed Integer Linear Program (MILP) formulations and evaluate their computational advantages and limitations by running experiments on instances from the literature commonly used for the classical GBP. Moreover, we propose a degree- and eigencentrality-based greedy-heuristic and compare it with the MILP.

Notation and preliminaries. If not stated otherwise, graph $G = (V, E)$ is simple and undirected; the *order* of a graph G refers to $|V|$, the number of vertices. By $N(v)$, for $v \in V$, we denote the set of vertices that are adjacent to v, i.e., $N(v) := \{w \in V : \{v, w\} \in E\}$. Moreover, let us denote by $\hat{G}$ the digraph resulting from bidirecting each edge of G, i.e., $\hat{G} := (V, \hat{E})$ with $\hat{E} := \{(u, v), (v, u) \in V \times V : \{u, v\} \in E\}$. In case some graph $G = (V, E)$ is a directed graph (digraph), the *in-degree* and *out-degree* of v refer to the quantities $|\{u \in V : (u, v) \in E\}|$ and $|\{w \in V : (v, w) \in E\}|$ respectively. The number of edges on a shortest path between v and w is denoted by $\mathrm{dist}(v, w)$. An *arborescence* is a directed acyclic graph $A = (V, E)$ with at most one vertex of in-degree zero—called the *root*—having in-degree one for all remaining vertices. Denote by $\mathrm{leaves}(A)$, the set of all leaves, i.e., vertices with out-degree zero, by $\mathrm{depth}(v)$ the number of edges on the unique directed path from the root of A to a vertex v, and by $\mathrm{height}(r)$ the maximum depth over all vertices contained in the arborescence with root r. Sequences (or unions) of pairwise vertex-disjoint arborescences are called *forests of arborescences*. Denote by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ the Gaussian *floor* and *ceil* function, respectively.

The remainder of the paper is organized as follows. In Section 2 we motivate our generalization of graph burning. Section 3 provides two MILP formulation to solve the introduced problem. Section 4 proposes a heuristic approach whose performance is in Section 5 compared to the MILP approaches. Conclusions and open questions are given in Section 6.

2 Constraints on the diffusion process

The classical GBP despite being practically motivated, assumes for its diffusion process that all neighbors of burned vertices get also burned in the subsequent timestep. For many scenarios this assumption may seem too unrealistic: In general, people can successfully influence only a certain maximum number of other people from their circle of contacts (e.g., due to individually different resources, skills, or ambition for persuasion). Therefore, two constraints on the diffusion process shall be imposed to make the model more realistic and more flexibly usable. We propose to limit, by a vertex-dependent parameter threshold, the number of neighbors that can be ignited by a burned vertex.

Moreover, we want to incorporate the assumption that people become active in influencing other people just in the time period immediately after having been influenced themselves; we motivate this by referring to [20] which affirms a reduction of the spread of social contagion related to the decay of novelty. In the language of fire diffusion, the latter model is explained as follows: Vertices represent regions and a comparable behavior of the fire spread arises when fire can be contained to spread to at most a certain maximum number of neighboring regions, e.g., by having a specific resource of firefighter units locally available. Moreover, in this context, we assume that an already burned region cannot propagate fire at a later time point.

For a formalization, let us consider the graph $G = (V, E)$ with imposed thresholds $\theta_v \in \mathbb{N} \cup \{0\}$, $v \in V$. Apart from the aforementioned time-dependent status burned or unburned for each vertex, we implicitly track a second time-dependent, irreversible binary status of a vertex, called the *expiration* status, which prevents vertices having this status activated from diffusing their status burned to their neighbors. Assume that we are given a maximum time index $T \in \mathbb{N} \cup \{0\}$. Initially, for $t = 0$, burn a single vertex. Afterwards, execute the following actions for $t = 1, \ldots, T$:

(i) *Constrained diffusion from unexpired vertices:* For each vertex v that has the status burned in timestep $t - 1$ but not in timestep $t - 2$, burn at most θ_v of its—according to the timestep $t - 1$—unburned neighbors. (By convention, in the (-1)-th timestep all vertices are unburned.)
(ii) *Burn:* Pick at most one currently unburned vertex and make it burned.

We call an execution of the latter procedure a *T-terminating θ-burning process (with constrained diffusion)* and denote the set of all such processes as $\mathcal{B}_{\theta,T}(G)$. For $P \in \mathcal{B}_{\theta,T}(G)$, the number of vertices with the status burned after termination of both phases in the final timestep T is called the *penetration* of P, adopting the terminology of the LCIMP [12]. If T^* is the minimum number that ensures a process $P \in \mathcal{B}_{\theta,T^*}(G)$ of penetration $|V|$ exists, we define $b_\theta(G) := T^* + 1$ and call the latter *(constrained diffusion) θ-burning number of G*. The problem aiming for this optimal value T^* is henceforth called the *Constraint Diffusion Graph Burning Problem* (CDGBP, or θ-GBP). If we want to provide more information on the burning process, we denote it as $\mathcal{B}_{\theta,T}(G, (s_0, \ldots, s_T), y)$ where s_i is the chosen

vertex, henceforth called *seed*[1] during the *burn*-phase in timestep i. Moreover, for each directed edge $(u, v) \in \hat{E}$ we define

$$y_{u,v} := \begin{cases} 1 & \text{if } u \text{ was responsible to burn } v \text{ in some timestep } t \in \{1, \dots, T\}, \\ 0 & \text{otherwise.} \end{cases}$$

The subsequent Proposition 1, in particular, gives a preparatory viewpoint for the later stated MILP (1)–(17) by directly lifting the considerations concerning $b(\cdot)$ in the proof of [3, Theorem 5] to θ-burning. Moreover, a straight-forward observation is given in the subsequent Proposition 2.

Proposition 1 *Let $T \in \mathbb{N} \cup \{0\}$ and $X \subseteq V$ for a given graph $G = (V, E)$ with thresholds θ. Then, the following assertions are equivalent.*

(i) There is a θ-burning process $\mathcal{B}_{\theta,T}(G, (s_0, \dots, s_T), y)$ whose set of burned vertices after termination is X.

(ii) In G there is a subforest consisting of arborescences $T_{s_0}, T_{s_1}, \dots, T_{s_T}$ with respective roots $s_0, \dots, s_T$ and heights $h_0, \dots, h_T$ such that, firstly, the set of all vertices contained in some arborescence corresponds to X; secondly, $h_i \leq T - i$ for $i = 0, \dots, T$; and thirdly, each $v \in X$ is a member of a single arborescence $T_{s(v)}$ where the out-degree of v inside $T_{s(v)}$ is at most θ_v.

Proof. (i) $\implies$ (ii): Let $P \in \mathcal{B}_{\theta,T}(G, (s_0, \dots, s_T), y)$ and X be the set of vertices burned after termination of P. During the process, a vertex v might have become burned because one vertex (or even multiple vertices) "chose" to make their neighbor v burned. In the following, consider the the digraph $\hat{G} := (V, \hat{E})$, i.e., the bidirected version of G. Note that if $s \in \{s_0, \dots, s_T\}$ is a seed, we have $y_{u,s} = 0$ for each $u \in N(s)$, whereas $y_{s,u}$ can in fact attain the value 1 for $u \in N(s)$. If $\hat{E}|_y := \{(u, v) \in \hat{E} : y_{u,v} = 1\}$, then $\hat{G}|_y := (V, \hat{E}|_y)$ contains no directed cycles (not even of length two): In fact, the existence of a vertex having an incoming and an outgoing edge contradicts the property that only unburned vertices can be turned to burned by a burned neighbor. Facing a directed acyclic graph, we can therefore locate a set of sources (i.e., vertices of in-degree zero) that here must coincide with $\{s_0, \dots, s_T\}$.

We run now the following procedure which returns an updated edge-status $y'_{u,v} \in \{0, 1\}$ satisfying for all $(u, v) \in \hat{E}$ $y'_{u,v} \leq y_{u,v}$: Initially set $y'_{u,v} := y_{u,v}$. Then, iteratively, for $d = 1, \dots, T$, execute the following: For each vertex $v \in V$ being reachable from a seed s_i via a directed length-d path do the following: Pick the seed s_{i^*} with the minimum index i^* permitting such a directed path $(s_{i^*}, u_1, u_2, \dots, u_{d-1}, v)$. Then, overwrite $y'_{u,v} := 0$ for all $u \in N(v) \setminus \{u_{d-1}\}$. After termination of this procedure, we obtain that $\hat{G}|_{y'} := (V, \hat{E}|_{y'})$ is a forest of arborescences $(A_{s_0}, \dots, A_{s_T})$ whose roots are given by the seeds of the process. As all finally-burned vertices in X got burned in timestep T the latest, the distance from s_i to any vertex in A_{s_i} can be at most $T - i$. Moreover, y meets the capacity constraints of θ for each vertex $v \in V$, i.e., $\sum_{w \in N(v)} y_{v,w} \leq \theta_v$, and hence y', as a thinned-out version of y, does so as well.

[1] Potentially a seed reflects the empty vertex selection ε and this will manifest itself in our models in that direct source-sink flows are present.

(ii) $\implies$ (i): An admissible burning process can be reconstructed by simply choosing the sequence of seeds $(s_0, \ldots, s_T)$ and in iteration i, for each burned vertex v, choosing as next vertices to be burned those being the children of v when v is seen as a member of its associated arborescence. Here the threshold-respecting out-degrees of the vertices in the arborescence will automatically imply compatibility with the thresholds for an admissible θ-burning processes. The number of timesteps needed for this burning process is bounded from above by T, the height-restriction on the arborescence rooted in s_0. By construction, all $v \in X$ are therefore burned at most after termination of timestep T. □

Proposition 2 *For $G = (V, E)$ and $\theta_v := \deg(v)$, $v \in V$, we have $b_\theta(G) = b(G)$.*

Given $G = (V, E)$, θ and T, finding the maximum penetration over all $P \in \mathcal{B}_{\theta,T}(G)$ may clearly be seen as a special case of the problem asking, for a pre-specified *arbitrary* parameter vector of heights $H := (h_1, \ldots, h_Q)$, $Q \leq |V|$, to find a sequence of Q arborescences maintaining these heights, respecting the thresholds θ and yielding maximum penetration. For the 1-tuple $H = (h)$ we call it the problem of finding a *Maximum Coverage Arborescence with max-Degree θ and max-Height h*, abbreviated MCADH(G, θ, h).

3 Mixed integer linear programming formulations

In the following, given the maximum time-index T, we address the problem of finding a sequence of seeds $(s_0, \ldots, s_T)$ for a θ-burning process leading to maximum penetration in a given graph $G = (V, E)$ with threshold parameter θ. The idea is to make use of Proposition 1 and to find a collection of pairwise vertex-disjoint arborescences $A^0, \ldots, A^T$ of the graph meeting all height and threshold constraints and maximizing the number of vertices comprised by the collection.

For the following we consider $\hat{G}_* = (V \cup \{p, q\}, \hat{E}_q^p)$, a transformed version of $\hat{G}$, whose vertex set includes an auxiliary new source vertex p respectively a new sink vertex q and whose edge set $\hat{E}_q^p := \hat{E} \cup \{(p, v), (v, q) : v \in V\} \cup \{(p, q)\}$ models extended connectivity. In view of a MILP, we introduce variables $y_{u,v}^t \in \{0, 1\}$ for $t = 0, \ldots, T$ and $(u, v) \in \hat{E}_q^p$ modeling for $(u, v) \in \hat{E}$ the presence of the edge (u, v) within the arborescence A^t and modeling auxiliary information for the remaining edges in $\hat{E}_q^p$.

Moreover, for $t = 0, \ldots, T$, we declare the continuous variables $\varphi_{u,v}^t$, $(u, v) \in \hat{E}_q^p$, for modeling a flow, associated to the t-indexed commodity, which starts at the source p, spreads through the arborescence A^t (on the activation edges selected by y^t), augments in volume after every transition, and finally enters the sink q. We use this commodity flow approach to formulate constraints controlling the height of A^t. For doing so, we use the additional notation $\hat{E}_q := \hat{E} \cup \{(v, q) : v \in V\}$; moreover, note that in the following $\deg(\cdot)$ and $N(\cdot)$ apply to the original, undirected graph $G = (V, E)$ without any additional vertices.

$$\max \sum_{t=0}^{T} \sum_{(u,v)\in\hat{E}} y^t_{u,v} + \sum_{t=0}^{T} \sum_{v\in V} y^t_{p,v} \tag{1}$$

$$\text{s.t.} \sum_{v\in V} y^t_{p,v} + y^t_{p,q} = 1 \qquad \forall t \tag{2}$$

$$\sum_{v\in N(w)} y^t_{v,w} + y^t_{p,w} \leq 1 \qquad \forall t, \forall w \in V \tag{3}$$

$$y^t_{v,w} \leq y^t_{p,v} + \sum_{u\in N(v)\setminus\{w\}} y^t_{u,v} \qquad \forall t, \forall (v,w) \in \hat{E}_q \tag{4}$$

$$y^t_{u,v} \leq \sum_{w\in N(v)\setminus\{u\}} y^t_{v,w} + y^t_{v,q} \qquad \forall t, \forall (u,v) \in \hat{E} \tag{5}$$

$$y^t_{v,q} \leq 1 - y^t_{v,w} \qquad \forall t, \forall (v,w) \in \hat{E} \tag{6}$$

$$\sum_{w\in N(v)} y^t_{v,w} \leq \theta_v \qquad \forall t, \forall v \in V \tag{7}$$

$$y^t_{v,w} + y^t_{w,v} \leq 1 \qquad \forall t, \forall \{v,w\} \in E \tag{8}$$

$$\varphi^t_{p,v} = y^t_{p,v} \qquad \forall t, \forall v \in V \cup \{q\} \tag{9}$$

$$0 \leq \varphi^t_{v,w} \leq T - t + 1 \qquad \forall t, \forall (v,w) \in \hat{E} \tag{10}$$

$$0 \leq \varphi^t_{v,q} \leq T - t + 2 \qquad \forall t, \forall v \in V \tag{11}$$

$$\varphi^t_{v,w} \leq (T - t + 2) y^t_{v,w} \qquad \forall t, \forall (v,w) \in \hat{E}^p_q \tag{12}$$

$$(T - t + 2)(y^t_{v,w} - 1) + \varphi^t_{u,v} + 1 \leq \varphi^t_{v,w} \qquad \forall t, \forall (u,v,w) \in P_3, \tag{13}$$

$$\text{where } P_3 := \left\{(u',v',w') \in \mathcal{K} : u' \neq w', (u',v') \in \hat{E}^p, (v',w') \in \hat{E}_q\right\} \text{ with } \mathcal{K} := (V \cup \{p\}) \times V \times (V \cup \{q\})$$

$$\sum_{t=0}^{T} y^t_{v,w} \leq 1 \qquad \forall (v,w) \in \hat{E}^p_q \tag{14}$$

$$\sum_{v\in N(w)} \sum_{t=0}^{T} y^t_{v,w} + \sum_{t=0}^{T} y^t_{p,w} \leq 1 \qquad \forall w \in V \tag{15}$$

$$y^t_{v,w} \in \{0,1\} \qquad \forall t, \forall (v,w) \in \hat{E}^p_q \tag{16}$$

$$\varphi^t_{v,w} \in \mathbb{Q} \qquad \forall t, \forall (v,w) \in \hat{E}^p_q \tag{17}$$

Here, "$\forall t$," quantifies over all $t \in \{0, \ldots, T\}$ corresponding to different commodities. Constraints (2)–(17) are individually defined for each commodity t, and they enforce the subsequently stated properties, together describing an intermediate "quasi-arborescence" $A^t_{\text{itmd}} := (V, \hat{E}^p_q)|_{y^t}$ indicated by the y^t-values. Constraint (2) assures that the source points via a uniquely activated edge (p, v) to exactly one vertex in $v \in V \cup \{q\}$; this vertex v is supposed to become the root of A^t_{itmd}. In (3) the uniqueness of predecessors in A^t_{itmd} is enforced (except for q). The absence of cycles will be implied by the augmenting flow passing through the graph. The fact that a non-source vertex not receiving activation from some incoming edge cannot

provide an outgoing activation is reflected by (4). Constraint (5) makes sure that every non-sink vertex receiving incoming activation must also forward this activation to at least one successor. Constraint (6) enforces that whenever a vertex activates a non-sink successor, then it cannot simultaneously activate the sink. Constraints (7)–(8) guarantee that the threshold constraints are fulfilled and that activation is never bidirectional (asymmetry).

In (9)–(13) the management of the flow through the activated edges is addressed. Constraints (9)–(11) establish that the flow leaving the source precisely agrees with the edge-activation status respectively that the flow may not exceed the prescribed upper bound on the height of the arborescence A^t. The fact that no flow can pass through an inactivated edge is reflected in (12), whereas (13) implies that vertices in V that receive incoming activation and incoming flow f transport at least flow $f + 1$ on each of its edges giving outgoing activation.

Finally, constraints (14) respectively (15) enforce a non-mixing respectively a non-incident behavior of the different commodities. More precisely, (14) ensures that if an edge is activated according to some commodity t, then it remains non-activated for all other commodities. Similarly, (15) guarantees that if a vertex has an incoming activating edge according to a commodity, then it has no incoming activating edges according to all other commodities.

After removing from A^t_{itmd} the single edge leaving p as well as all edges leading to q, we obtain a proper arborescence inside $\hat{E}$ of height at most $T - t$. The flow is thus responsible for tracking the depth of each node and therefore limiting the height of the arborescence; cannot exceed a limit of $T - t$ as it never exceeds the value of $T - t + 2$ for A^t_{itmd} which is two units higher.

The objective function (1) uses the fact that the number of vertices in arborescence A^t is the count of edges plus one and is stated in such a manner that also corner cases, yielding height two or one for A^t_{itmd}, are handled correctly. A supportive illustration for (1)–(17) is given in Fig. 1.

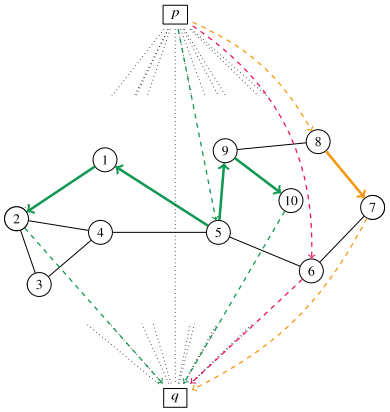

Fig. 1 The MILP (1)–(17) with an own commodity (i.e., an own color in the illustration) per arborescence. $T = 2$ and thresholds $\theta_v := \lfloor \deg(v)/2 \rfloor$ for $v \in \{1, \ldots, 10\}$ are assumed.

We now adapt our previous formulation towards an alternative one that only requires a single "universal" commodity implicitly carrying information on $T+1$ commodities. We rely here on single collections of binary activation variables y, respectively flow variables φ. This adaptation comes at the price that for each $w \in V$ additional binary variables ℓ_w^t, $t \in \{0, \dots, T\}$, are used. They indicate which commodity t is actually passing through w, whenever $y_{v,w}$ activates a w-incident edge $(v, w) \in \hat{E}$.

We again use an auxiliary graph

$$\hat{G}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)} := (V \cup \{p^0, \dots, p^T\} \cup \{q^0, \dots, q^T\}, \hat{E}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)}),$$

this time having $T+1$ sources $p^0, \dots, p^T$ respectively sinks $q^0, \dots, q^T$ and directed edges in

$$\hat{E}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)} := \hat{E}_{(q^0,\dots,q^T)} \cup (\{p^0, \dots, p^T\} \times V) \cup \{(p^0, q^0), \dots, (p^T, q^T)\},$$

where

$$\hat{E}_{(q^0,\dots,q^T)} := \hat{E} \cup (V \times \{q^0, \dots, q^T\}).$$

In this setting our alternative model reads as follows.

$$\max \sum_{(u,v)\in\hat{E}} y_{u,v} + \sum_{t=0}^{T} \sum_{v\in V} y_{p^t,v} \tag{18}$$

$$\text{s.t.} \sum_{v\in V} y_{p^t,v} + \sum_{z=0}^{T} y_{p^t,q^z} = 1 \qquad \forall t, \tag{19}$$

$$\sum_{v\in N(w)} y_{v,w} + y_{p^t,w} \le 1 \qquad \forall t, \forall w \in V \tag{20}$$

$$y_{v,w} \le \sum_{t=0}^{T} y_{p^t,v} + \sum_{u\in N(v)\setminus\{w\}} y_{u,v} \qquad \forall (v,w) \in \hat{E}_{(q^0,\dots,q^T)} \tag{21}$$

$$y_{u,v} \le \sum_{w\in N(v)\setminus\{u\}} y_{v,w} + \sum_{t=0}^{T} y_{v,q^t} \qquad \forall (u,v) \in \hat{E} \tag{22}$$

$$y_{v,q^t} \le 1 - y_{v,w} \qquad \forall t, \forall (v,w) \in \hat{E} \tag{23}$$

$$\sum_{w\in N(v)} y_{v,w} \le \theta_v \qquad \forall v \in V \tag{24}$$

$$y_{v,w} + y_{w,v} \le 1 \qquad \forall \{v,w\} \in E \tag{25}$$

$$\varphi_{p^t,v} = y_{p^t,v} \qquad \forall t, \forall v \in V \cup \{q^0, \dots, q^T\} \tag{26}$$

$$0 \le \varphi_{u,v} \le T+1 \qquad \forall (u,v) \in \hat{E} \tag{27}$$

$$0 \le \varphi_{v,q^t} \le T-t+2 \qquad \forall t, \forall v \in V \tag{28}$$

$$\varphi_{v,w} \le (T+2) y_{v,w} \qquad \forall (v,w) \in \hat{E}_{(q^0,\dots,q^T)} \tag{29}$$

$$(T+2)(y_{v,w}-1)+\varphi_{u,v}+1\le\varphi_{v,w} \qquad \forall t,\forall(u,v,w)\in P_3, \tag{30}$$

$$\text{where } P_3:=\left\{(u',v',w')\in\mathcal{K}: u'\neq w', (u',v'),(v',w')\in \hat{E}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)}\right\}$$

$$\text{with } \mathcal{K}:=(V\cup\{p^0,\dots,p^T\})\times V\times(V\cup\{q^0,\dots,q^T\}),$$

$$y_{p^t,v}\le\ell^t_v \qquad \forall t,\forall v\in V \tag{31}$$

$$\sum_{t=0}^{T}\ell^t_v\le 1 \qquad \forall v\in V \tag{32}$$

$$(\ell^t_v-1)+y_{v,w}\le\ell^t_w \qquad \forall t,\forall(v,w)\in\hat{E} \tag{33}$$

$$y_{v,q^t}\le\ell^t_v \qquad \forall t,\forall v\in V \tag{34}$$

$$y_{v,w}\in\{0,1\} \qquad \forall(v,w)\in\hat{E}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)} \tag{35}$$

$$\varphi_{v,w}\in\mathbb{Q} \qquad \forall(v,w)\in\hat{E}^{(p^0,\dots,p^T)}_{(q^0,\dots,q^T)} \tag{36}$$

$$\ell^t_v\in\{0,1\} \qquad \forall t,\forall v\in V \tag{37}$$

Again, "$\forall t$," quantifies over all $t\in\{0,\dots,T\}$. This alternative formulation is illustrated in Fig. 2. We remark that formulation (18)–(37) can be strengthened

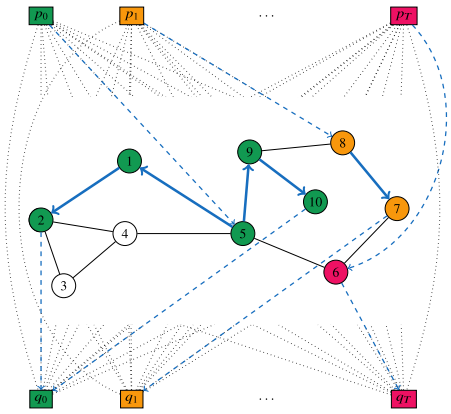

Fig. 2 Relying on a single commodity, the MILP (18)–(37) enforces propagation of vertex labels (indicated by colors) to successors. Edges pointing to q^t, $t=0,\dots,T$, have an edge-capacity of $T-t+2$ imposed for their through-passing flow. $T=2$ and thresholds $\theta_v:=\lfloor\deg(v)/2\rfloor$ for $v\in\{1,\dots,10\}$ are assumed.

similarly as done in in [10] by the fact that vertices on any arborescence must maintain a certain maximum distance to their respective root. More precisely, we can impose

$$y_{p^t,v}+\ell^t_w\le 1 \quad \forall t,\ \forall\{v,w\}\in\{\{v',w'\}\subseteq V:\operatorname{dist}(v',w')>T-t\}. \tag{38}$$

Similarly, to formulation (1)–(17) we may add the constraints

$$y^t_{p,v} + y^t_{w,z} \leq 1 \quad \forall t,\ \forall (v,w,z) \in \mathcal{D}_t \tag{39}$$

$$\text{with } \mathcal{D}_t := \{(v',w',z') \in V^3 : \text{dist}(v',w') \geq T - t, z' \in N(w')\}.$$

To implement the last two constraints, a precalculation of the distances between all pairs of vertices is required, which can be done by Johnson's algorithm.

Remark 1 Both of our models rely on a fixed maximum time-index T. If we are interested in calculating b_θ, or more precisely in finding a shortest sequence of arborescences yielding full penetration, in principle, we can start with $T = 0$ and determine—by a MILP solver—if the property already holds. If this is not the case we iteratively repeat the procedure with incremented T-value, until a desired sequence of arborescences is found. To reduce the number of solver calls, it is conceivable to perform a binary search on the minimum value for T, see also [10, Algorithm 1].

4 A heuristic approach

As we will see in the computational results, the proposed MILP formulations do not scale well to large θ-GBP or even MCADH instances with hundreds of vertices due to in particular the high number of activation and flow variables. Thus, we propose a faster heuristic approach, which is, as we will see, able to produce solutions of reasonable quality also for large-sized instances. A typical greedy approach in the setting of the θ-GBP would be to iteratively find arborescences of decreasing heights and to place them onto the input graph. After the placement of an arborescence, all used vertices are eliminated from the working copy of the input graph before the next arborescence is derived with the bound on the height decreased by one. The procedure terminates after having reached the bound zero for the height. Note that it may happen that all nodes are already covered earlier, in which case we assume here that all remaining arborescences are empty. The core task in this heuristic framework concerns the solution of the intermediate one-arborescence problem MCADH(G, θ, h), as introduced in Section 2. We propose to achieve this by iteratively extending an initial one-vertex arborescence A by finding the most promising current leaf w of A and strategically appending as many further vertices as the threshold of w allows us. This growing process is guided by a greedy scoring function quantifying how useful it is to append to a leaf w of A a neighbor $z \in N(w)$ (according to the adjacency of the original graph G) which is not yet in the arborescence. The scoring function we use is

$$\mathsf{LN}(A, w; z) := d(w) \cdot \frac{\theta'(z)}{\max_{y \in V} \theta'(y)} + (T - 1 - d(w)) \cdot \text{ec}_{G-(A \cup \{w,z\})}(z),$$

where $d(w)$ denotes the depth of w inside A. Value $\theta'(z)$ refers to the remaining effective threshold of node z taking into account that some neighbors of z may already be incorporated into the arborescence; thus this threshold for the count of neighbors that might be started to burn via diffusion can already be smaller. Lastly,

$\mathrm{ec}_{G-(A\cup\{w,z\})}(z)$ denotes the eigencentrality of z after excluding the edge $\{w, z\}$ and the vertices of the so-far built arborescence A from G.

The (renormalized) value of $\theta'(z)$ and the eigencentrality of z—both attaining values in the interval $[0, 1]$—are linearly combined by weights assuring that leaves whose current height is far from the maximum height T receive reward from a good eigencentrality, and gradually, as the heights of the leaves increase, their (renormalized) θ'-value receives more importance. On purpose, in the most extremal case the eigencentrality does not even contribute to LN.

Eventually we want to find a leaf w of A for which

$$\mathsf{LeafScore}(A, w) := \max_{\substack{P\subseteq N(w)\setminus A \\ |P|\le\theta'(w)}} \sum_{z\in P} \mathsf{LN}(A, w; z) \tag{40}$$

is a maximum. When this maximizer w is found, the arborescence A is extended towards all vertices in the set P that was itself responsible for maximizing (40).

We start the whole construction by initializing A with a vertex having highest eigencentrality, apply aforementioned process as long as an extension is possible, and return this maximal arborescence as result for the MCADH instance. We refer later to the approach carried out iteratively for $t = T, T-1, \ldots, 1, 0$ as *Greedy One Arborescence at a Time* (GOAT). To verify the potential of this heuristic, we concurrently also realize the placements of single arborescences by a respective MILP (18)–(37) with a generous (see Section 5) time limit for solving.

5 Computational results

The MILP and GOAT approaches proposed in Section 3 have been implemented in `Julia 1.11.1`. The `Gurobi` solver[2] in version 10.0.3 is used to solve the MILPs. A cluster with an Intel(R) Xeon(R) E5-2640 v4 CPU with 2.40GHz and 160GB RAM running Ubuntu 18.04.6 LTS was used to run all experiments on a single thread.

We rely on the benchmark instances from [10, Table 3]. Here, we show results only for every fourth instance of these benchmark set due to space limitations. The results for all instances can be found online[3]. As thresholds we use $\theta_v := \lfloor v/2 \rfloor$, $v \in V$, comparable to the so-called *majority thresholds* for the Target Set Selection Problem [14]. The following approaches are considered in the comparisons: (i) the MILP (1)–(17) (labeled as MCF); (ii) the MILP (1)–(17) strengthened by (39) (labeled as MCF-s); (iii) the MILP (18)–(37) (labeled as SCF); (iv) the MILP (18)–(37) strengthened by (38) (labeled as SCF-s); (v) M600, the GOAT version realizing placements of single arborescences via the MILP solver with a time limit of 600 seconds; (vi) the original GOAT approach. A time limit of 1000 seconds is given

[2] `https://www.gurobi.com`

[3] `https://www.ac.tuwien.ac.at/research/problem-instances/#Graph_Burning_Problem`

to `Gurobi` to solve MCF, MCF-s, SCF, SCF-s models. No time limit is set for the original GOAT approach.

Table 1 is structured as follows: each column corresponds to a selected instance, labeled with its respective name. The first three rows provide key characteristics of each instance, including the number of vertices, the number of edges, and the standard deviation of vertex degrees. The subsequent rows present the results for MCF, MCF-s, SCF, SCF-s, M600, and GOAT, evaluated for $T \in \{2, 3\}$. For the GOAT approach, both the objective values (GOAT-obj) and execution times (GOAT-time [s]) are provided in separate rows. The best results for each instance are highlighted in bold. Concerning the dual bounds produced by the MILPs, up to single exceptions, they coincide with the trivial values given by the graph order, and therefore are not reported here.

Table 1 Comparison of approaches for $T \in \{2, 3\}$. The entries "/" indicate encountered memory overflow errors.

	karate-club	polbooks	ia-enron-only	sphere3	c-fat500-1	web-polblogs	DD199	lattice3D	lattice2D	ia-fb-messages	TVshow
$\|V\|$	34	105	143	258	500	643	841	1000	1089	1266	3892
$\|E\|$	78	441	623	768	4459	2280	1902	2700	2112	6451	17262
std(deg)	3.88	5.47	6.08	0.30	1.31	11.46	1.47	0.69	0.34	13.24	12.6
$T = 2$											
MCF	**34**	60	85	**18**	40	39	15	**18**	**11**	98	9
MCF-s	**34**	49	107	**18**	35	7	10	8	7	14	/
SCF	33	**62**	57	**18**	43	133	14	17	**11**	11	15
SCF-s	**34**	41	106	17	9	55	17	3	3	3	/
M600	32	48	51	**18**	46	32	15	**18**	**11**	200	27
GOAT-obj	25	**62**	**109**	**18**	**47**	**283**	**26**	**18**	**11**	**697**	**204**
GOAT-time[s]	0.01	0.01	0.02	0.01	0.03	0.24	0.02	0.01	0.02	21.73	27.52
$T = 3$											
MCF	**34**	98	71	49	64	227	33	28	**26**	90	8
MCF-s	**34**	97	127	48	30	130	10	13	9	4	/
SCF	**34**	102	128	**50**	44	146	20	50	**26**	41	11
SCF-s	**34**	87	136	30	4	35	4	4	4	4	/
M600	**34**	**103**	131	45	81	230	33	**57**	**26**	756	62
GOAT-obj	21	99	**138**	47	**89**	**402**	**55**	56	**26**	**1062**	**626**
GOAT-time[s]	0.01	0.03	0.06	0.02	0.08	0.67	0.08	0.07	0.07	42.88	30.73

The following observations can be made from the reported results.

- For $T = 2$, the best approach in terms of the delivered (heuristic) solutions is the GOAT approach as it almost consistently outperforms the other approaches and requires significantly less runtime. For nine out of eleven instances, it finds the

best solutions. The second best approach is MCF which finds best solution in four cases. Formulation MCF seems to be more effective than SCF. Concerning the strengthened MILP versions, the derived cuts helped for small instances where overly better primal bounds are obtained than by the original MILPs. However, the consideration of these cuts has a notable negative effect on the final results in the case of the largest instances. The effectivity of the MILPs hardly depends on the sparsity of the graphs. Finally, it can be seen that the M600 approach scales better than the MILPs.

- For $T = 3$, the best approach in terms of obtained solutions is again GOAT; it is also fastest as all its runs require less than one minute. GOAT delivers the best solutions on seven (out of eleven) instances while consistently outperforming other approaches on the largest instances. The second best approach is M600 which delivers best solution on four instances. Again, for the densest graphs, the produced primal bounds of the MILPs are rather weak and not useful in practice.

6 Conclusion

We introduced a generalized version of the well-studied graph burning problem under a more realistic diffusion process of information spreading among neighboring vertices. The diffusion process here is constrained by adding a capacity threshold to each vertex that limits the number of neighboring vertices to be possibly affected by a burned vertex in the subsequent iteration. This problem is linked to the task of finding a shortest sequence of vertex-disjoint arborescences with specific heights and capacity constraints on their out-degrees. Following this observation, two MILP models based on network flows were proposed. As these exact approaches are limited in their practical applicability, a first heuristic also is proposed to solve in particular also larger problem instances. It is a greedy approach that relies on iteratively creating arborescences. Choosing which vertex to add to an arborescence is the most important decision herein. It is realized by linearly combining the renormalized remaining capacity and the eigencentrality of a considered vertex; those vertices with a better score are preferred to be prolonged. While the recent work [6] remarkably indicates that the burning number can be computed optimally for large graphs, we observe that after the inclusion of the threshold constraints, the problem seems to become considerably more difficult. Our experimental evaluation attests the heuristic a high effectiveness as it outperforms all other approaches. The MILP models perform reasonably well only on small-sized and sparse graphs.

Our greedy-based approach still neglects the aspect of "communication" between the different arborescences. In future work, one may address their simultaneous construction. We plan to consider the connectivity, in particular the number and topology of the arising connected components, to enhance the approach. Furthermore, several open problems arise in the context of the introduced generalized (θ-)burning process:

(i) The complexity for classes of thresholds (as general as possible) is still to be analyzed: Here, it would already be interesting to settle the question if for each

scalar $\alpha \in (0,1)$, the thresholds $\theta(\cdot) := \lfloor \alpha \deg(\cdot) \rfloor$ imply NP-completeness for the problem of determining equality of the θ-burning number and a given number $\beta_\theta \in \mathbb{N}$; settling the hardness of finding the maximum penetration for a given maximum timestep T would be interesting under this proportional threshold model, too. Moreover, constant thresholds are also worth to be examined from this point of view.

(ii) From the perspective of fire spread, the θ-burning number expresses a worst-case that we are interested in precalculating. However, it might be also relevant to consider the setting where a number of unburned neighbors picked uniformly at random are chosen to receive the status burned—at the same time, we might want to assume a random choice of the seeds. This yields a different parameter of vulnerability and coming up with bounds on the expected penetration over all processes obeying these assumptions is an interesting challenge.

(iii) An interesting follow-up problem would be: Given a budget $B \in \mathbb{N}$, try to find a vector of threshold "reducers" $(r_v)_{v \in V}$ with $0 \leq r_v \leq \theta_v$ and $\sum_{v \in V} r_v \leq B$ such that for $\theta' := (\theta_v - r_v)_{v \in V}$ the respective value of $b_{\theta'}$ is minimum, i.e., find a budget-maintaining way to contain the worst-case of fire diffusion as much as possible.

Acknowledgements We thank Johannes Varga for vivid discussions. This research was funded in part by (i) the program VGSCO of the Austrian Science Fund (FWF) [10.55776/W1260-N35], (ii) the bilateral project funded by the Ministry of Civil Affairs of Bosnia and Herzegovina [1259074] and Austria's Agency for Education and Internationalization (OeAD) [BA05/2023]. Moreover, M. Djukanović is partially supported by the Ministry of Scientific and technological development and higher education of the Serb Republic with the project "Utilizing deep learning to enhance the efficiency of optimization algorithms".

References

1. Bessy, S., Bonato, A., Janssen, J., Rautenbach, D., Roshanbin, E.: Burning a graph is hard. Discrete Applied Mathematics **232**, 73–87 (2017)
2. Bonato, A.: A survey of graph burning. Contributions to Discrete Mathematics **16**(1), 185–197 (2021)
3. Bonato, A., Janssen, J., Roshanbin, E.: Burning a graph as a model of social contagion. In: Algorithms and Models for the Web Graph: 11th International Workshop. LNCS, vol. 8882, pp. 13–22. Springer (2014)
4. Bonato, A., Kamali, S.: Approximation algorithms for graph burning. In: Theory and Applications of Models of Computation: 15th Annual Conference. LNCS, vol. 11436, pp. 74–92. Springer (2019)
5. de C. Pereira, F., de Rezende, P.J., Yunes, T., Morato, L.F.B.: A row generation algorithm for finding optimal burning sequences of large graphs. In: 32nd Annual European Symposium on Algorithms. LIPIcs, vol. 308, pp. 94:1–94:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2024)
6. de C. Pereira, F., de Rezende, P.J., Yunes, T., Morato, L.F.B.: Solving the graph burning problem for large graphs. arXiv preprint arXiv:2404.17080 (2024)
7. Farokh, Z.R., Tahmasbi, M., Tehrani, Z.H.R.A., Buali, Y.: New heuristics for burning graphs. arXiv preprint arXiv:2003.09314 (2020)

8. García-Díaz, J., Cornejo-Acosta, J.A.: A greedy heuristic for graph burning. arXiv preprint arXiv:2401.07577 (2024)
9. García-Díaz, J., Pérez-Sansalvador, J.C., Rodríguez-Henríquez, L.M.X., Cornejo-Acosta, J.A.: Burning graphs through farthest-first traversal. IEEE Access **10**, 30395–30404 (2022)
10. García-Díaz, J., Rodríguez-Henríquez, L.M.X., Pérez-Sansalvador, J.C., Pomares-Hernández, S.E.: Graph burning: Mathematical formulations and optimal solutions. Mathematics **10**(15), 2777 (2022)
11. Gautam, R.K., Kare, A.S.: Faster heuristics for graph burning. Applied Intelligence **52**(2), 1351–1361 (2022)
12. Günneç, D., Raghavan, S., Zhang, R.: Least-cost influence maximization on social networks. INFORMS Journal on Computing **32**(2), 289–302 (2020)
13. Hiller, M., Koster, A.M., Triesch, E.: On the burning number of p-caterpillars. In: Graphs and Combinatorial Optimization: from Theory to Applications, AIRO, vol. 5, pp. 145–156. Springer (2021)
14. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 137–146. Association for Computing Machinery (2003)
15. Lieskovský, M., Sgall, J., Feldmann, A.E.: Approximation algorithms and lower bounds for graph burning. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. LIPIcs, vol. 275, pp. 9:1–9:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023)
16. Liu, H., Hu, X., Hu, X.: Burning number of caterpillars. Discrete Applied Mathematics **284**, 332–340 (2020)
17. Murakami, Y.: The burning number conjecture is true for trees without degree-2 vertices. Graphs and Combinatorics **40**(4), 82 (2024)
18. Norin, S., Turcotte, J.: The burning number conjecture holds asymptotically. Journal of Combinatorial Theory, Series B **168**, 208–235 (2024)
19. Šimon, M., Huraj, L., Dirgová Luptáková, I., Pospíchal, J.: Heuristics for spreading alarm throughout a network. Applied Sciences **9**(16), 3269 (2019)
20. Wu, F., Huberman, B.A.: Novelty and collective attention. Proceedings of the National Academy of Sciences **104**(45), 17599–17601 (2007)