Object Allocation and Positive Graph Externalities

Dimitris Fotakis[1] and **Laurent Gourvès**[2] and **Stelios Kasouridis**[3] and **Aris Pagourtzis**[4]

Abstract. The worth of an entity does not only come from its intrinsic value. The other entities in the neighborhood also influence this quantity. We introduce and study a model where some heterogeneous objects have to be placed on a network so that the elements with high value may exert a positive externality on neighboring elements whose value is lower. We aim at maximizing this positive influence called *graph externality*. By exploiting a connection with the minimum dominating set problem, we prove that the problem is **NP**-hard when the maximum degree is 3, but polynomial time solvable when the maximum degree is 2. We also present exact and approximation algorithms for special cases. In particular, if only two valuations exist, then a natural greedy strategy, which works well for maximum coverage problems, leads to a constant approximation algorithm. With extensive numerical experiments we finally show that a greedy algorithm performs very well for general valuations.

1 INTRODUCTION

The worth of an entity depends on its intrinsic value and the value of the other entities in the neighborhood: commercials during the Super Bowl, advertising banners on popular websites, hotels and restaurants by the sea or close to historic monuments, etc. All these elements take advantage from their environment. This effect is part of a general phenomenon usually called *externality*.

The previous examples describe how the value of an item is enhanced by the presence of something desirable in its neighborhood. More generally, externality is not necessarily local (e.g. the success of a technology depends on the number of its users [22]), it can be negative (e.g. when some competitors are nearby as for ads in sponsored search auctions [21]), and it can be originated from the combination of multiple factors (e.g. genetic disease). Understanding the strength of this phenomenon is an important challenge for the A.I. community involved in resource allocation and its impact on the society [4]. The present work follows this direction.

We propose to analyze local and positive externality on a set of entities. To do so, we introduce a model where a given set of goods (objects with non-negative values) has to be assigned to a given set of locations. An object derives positive externality from another object if these objects are placed on neighboring locations, and the value of the neighboring object is larger. The externality is just the difference of their values. In the model, the locations are the vertices of a graph. The edges of the graph capture the spatial or temporal relations of the locations (e.g. time slots of a TV channel, regions of a web page, spots, etc.).

[1] National Technical University of Athens, Greece, email: fotakis@cs.ntua.gr
[2] Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016, Paris, France, email: laurent.gourves@dauphine.fr
[3] National Technical University of Athens, Greece, email: stel.kasouridis@gmail.com
[4] National Technical University of Athens, Greece, email: pagour@cs.ntua.gr

We study the problem of placing the goods on the locations so as to maximize the *positive graph externality*, i.e., externality is non-negative and only possible along the edges of the graph. In our problem, called OPT-EXT, we assume that the externality of an object, if any, is reduced to the influence a single neighboring object whose value is locally the largest. This assumption makes sense in situations where having more than one superior neighbors is unhelpful. For example, a facility (e.g., a post office) provides externality to a neighboring area with no such facility, while a second nearby facility does not (unless it is larger than the first one, in which case it is the one that determines the externality exerted to the area with no facility—and the first facility plays no role in this case). Of course, considering a more sophisticated notion of externality that takes into account the presence of multiple neighbors with larger values is relevant and deserves attention (possible extensions are discussed in the final section of this article). However, as we shall see, even the simple case of externality coming from a single neighbor is computationally challenging and constitutes a first step towards the understanding and exploitation of positive externality.

Indeed, our first contribution is a hardness proof of OPT-EXT in the restricted case of two valuations, even if the graph has maximum degree 3 (Proposition 1). The two valuation case is interesting on its own, as it is reminiscent (but distinct) of **NP**-hard covering problems like DOMINATING SET or MAX COVERAGE. Our positive results for this two valuation case is a $(e-1)/(1+e)$-approximation algorithm for general graphs (Theorem 1) and an exact algorithm based on previous results for a generalization of DOMINATING SET (Proposition 3). For general non-negative valuations (i.e. not only two valuations), we prove that OPT-EXT is polynomial time solvable in graphs of degree at most 2 (Theorem 2) and graphs forming a collection of stars (Proposition 4). Thus, the maximum degree, whether it is at most 2 or at least 3, allows us to distinguish between easy and hard cases of OPT-EXT. We also propose a 0.5-approximation algorithm for caterpillar graphs (Proposition 5), a class of graphs which is a proper subset of trees. Finally, we propose a greedy algorithm as well as a nontrivial upper bound for the optimal externality. We conduct experiments on benchmarks and random graphs in order to measure the approximation ratio achieved by our algorithm. Our results show that the proposed algorithm yields an externality which is usually at least 90% of the upper bound, exhibiting an even better behavior as the density of the graph increases.

This paper is organized as follows. The next section gives a (non-exhaustive) list of works related to our model. Section 3 defines the model, the optimization problem OPT-EXT, and its complexity. The restriction to two valuations is treated in Section 4, followed by the general case (Section 5). We conclude with open questions and suggestions for future work.

Due to space constraints, some proofs are omitted or sketched.

2 RELATED WORK

Externality refers to the situation where the value of a set of objects does not solely depend on them, but is affected by something outside, typically the other objects and how they are allocated. Externality has been studied in economics [29, 22, 31] and computer science. For the latter domain, externality mainly appears in matchings [5], ad auctions [15, 24, 12, 17, 25, 21] and the fair allocations of goods that are either divisible [6, 26] or indivisible [28, 30].

Regarding ad auctions, the performance of an item depends on which other items are selected and displayed at the same time [15]. Externality in that case is often negative as the other ads may attract the user who only selects a single item. In *position auctions* [2, 21], the selected ads are placed in given parts of a web page. Apart from the fact that some parts are better than others (the higher the better), an ad derives most of its negative externality from the other ads placed alongside.

In the context of resource allocation, one often supposes that an agent's utility solely depends on her own share. Externality illustrates the fact that an agent's utility can also be influenced by the shares of the other agents. In [30] Seddighin et al. consider the problem of fairly allocating some indivisible objects and their goal is to satisfy an adapted notion of the maximin-share criterion. The connection with our model is the use of a social graph whose vertices are the agents and positive externalities are along the arcs.

OPT-EXT bears a resemblance to influence maximization models in social networks [23]. In influence maximization, we want to maximize the adoption of a new product in a social network. The adoption process takes place in rounds. In each round, a node adopts the product with probability proportional to the fraction of its neighbors that have already adopted it. We want to compute a set of k initial adopters that maximizes the expected final number of adopters. Influence maximization problems have important applications to marketing and pricing in social networks and have received significant attention (see e.g., [19, 1, 13] and the references therein). In contrast to influence maximization models, OPT-EXT assumes deterministic externalities that occur within a single round and deals with different object valuations. A crucial technical difference is that in influence maximization models, the expected final number of adopters is a monotone submodular function of the initial adopters (e.g., [23]), while in OPT-EXT (even with binary valuations, see Section 4), the total graph externality is not necessarily monotone or submodular.

OPT-EXT$(0,1)$, the restriction of OPT-EXT to objects valued either 0 or 1, belongs in the family of covering problems. Vertices receiving an object of value 1 somehow cover their neighbors having an object of value 0, and one wants to maximize the number of covered objects of value 0. DOMINATING SET and MAX COVERAGE are two typical covering problems. DOMINATING SET is a graph problem (see [14] and [8] for a survey) where, by selecting a vertex v, one *dominates* v and its neighbors. DOMINATING SET is to choose a subset of vertices of minimal cardinality such that the whole graph is dominated. More generally, for a given integer $t \geq 0$, the goal of PARTIAL DOMINATING SET (PDS for short) is to dominate at least t vertices with the minimum number of vertices [9, 11]. In the MAX COVERAGE problem, we are given a universe U, a collection $\mathcal{C}$ of subsets of U and an integer $k < |\mathcal{C}|$. The goal is to cover a maximum number of elements of U with at most k members of $\mathcal{C}$. This **NP**-hard problem is approximable within the ratio $(1-(1-1/k)^k) > 1-1/e$ [20]. More generally, if $f(X)$ denotes the number of elements covered by $X \subseteq \mathcal{C}$, then f is a monotone submodular function. The problem of maximizing a monotone submodular function admits a $(1-1/e)$-approximation algorithm [27].

Finally, our model is related to the problems studied in [3, 7] as these articles deal with the allocation of goods on the vertices on a graph. However, their objective and motivation are totally different.

3 MODEL, PROBLEM AND COMPLEXITY

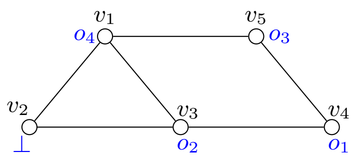
A set of objects (also called goods) $O = \{o_1, \ldots, o_m\}$ and a non-negative valuation $\nu(o_i) \in \mathbb{N}$ for each $o_i \in O$ are given. There is an undirected graph $G = (V, E)$ such that $V = \{v_1, v_2, \ldots, v_n\}$ and $n = |V| \geq |O| = m$.

One has to place the objects on the vertices so that every vertex gets at most one object, and every object is placed on exactly one vertex. An *allocation* is a function $\pi : V \to O \cup \{\perp\}$ where every object has exactly one ancestor and $\pi(v) = \perp$ means that v does not receive any object.

Two vertices i, j are neighbors if $\{i, j\} \in E$. $\mathcal{N}(v)$ and $\mathcal{N}[v] := \mathcal{N}(v) \cup \{v\}$ denote the *neighborhood* and the *closed neighborhood* of v, respectively. In the present work we assume that a vertex *derives externality* from at most one neighbor. Concretely, a vertex w derives externality from v if the following conditions are met: (i) $\pi(w) \neq \perp$ and $\pi(v) \neq \perp$ (i.e. they both receive an object), (ii) $\{w, v\} \in E$, (iii) $\nu(\pi(v)) > \nu(\pi(w))$, and (iv) $\pi(v)$ is the object with largest valuation in $\mathcal{N}(w)$.

For a given allocation π, the *graph externality* of a vertex w, denoted by $ext_\pi(w)$, is 0 when $\pi(w) = \perp$. Otherwise $ext_\pi(w)$ is equal to $\nu^* - \nu(\pi(w))$ where $\nu^* = \max\{\nu(\pi(v)) : v \in \mathcal{N}[w] \text{ and } \pi(v) \neq \perp\}$. Let $\mathsf{Ext}_\pi(G) := \sum_{v \in V} ext_\pi(v)$ be the graph externality of G under π.

Example 1 *Consider the following instance with 4 objects.*



$\nu(o_1) = 1$
$\nu(o_2) = 4$
$\nu(o_3) = 5$
$\nu(o_4) = 9$

Let $\pi = (o_4, \perp, o_2, o_1, o_3)$. *We have* $ext_\pi(v_4) = \nu(o_3) - \nu(o_1) = 4$, $ext_\pi(v_3) = \nu(o_4) - \nu(o_2) = 5$, $ext_\pi(v_5) = \nu(o_4) - \nu(o_3) = 4$, $ext_\pi(v_1) = \nu(o_4) - \nu(o_4) = 0$, $ext_\pi(v_2) = 0$ *and* $\mathsf{Ext}_\pi(G) = 13$.

The location of an object o is the vertex $\pi^{-1}(o)$. We extend the notions of (closed) neighborhood and externality to the objects when π is fixed. The same notations $\mathcal{N}$, ext and Ext are used for the sake of simplicity. We say that there is graph externality along an edge $\{v, v'\} \in E$ when v derives externality from v'. In this case, the externality along $\{v, v'\}$ is equal to $ext_\pi(v)$.

This article is devoted to a problem called OPT-EXT. An instance consists of a graph G, a set of objects O and their valuations ν. The objective is to find π that maximizes $\mathsf{Ext}_\pi(G)$. As it is obvious that all available objects must be placed on the graph, the total sum of intrinsic values is a constant. The objective function solely captures the positive graph externalities.

As a motivation for OPT-EXT, consider a TV channel whose schedule is an ordered list of TV programs. OPT-EXT where G is a path models this situation: each object is a TV program and each vertex of the path is a time slot. An attractive program can increase the audience of the programs that are broadcast right before or right after. Thus, besides the audience due to the intrinsic quality of the programs, the head of the TV channel wants to maximize the "external" audience.

A second motivation for OPT-EXT is how the contents of a web page are arranged[5]. The graph is like a grid whose vertices correspond to different regions of the screen. There is an edge between two vertices if their regions are contiguous. Some given contents need to be placed on the regions. Some contents may attract the reader's attention and increase the interest for other contents displayed in a neighboring region.

Our first result states that OPT-EXT is computationally hard. The proof relies on the DOMINATING SET problem. A *dominating set* in a graph $G = (V, E)$ is a set $D \subseteq V$ such that every $v \in V \setminus D$ has a neighbor in D. Given G and k, the DOMINATING SET problem is to decide if a dominating set of size at most k exists. The problem is **NP**-complete, even if G is planar with maximum vertex degree 3 [14], and also in bipartite graphs and split graphs [8].

Proposition 1 *Given an instance (G, O, ν) and a number t, deciding if an allocation π such that $\mathsf{Ext}_\pi(G) \geq t$ exists is* **NP***-complete, even if the valuations are 0 or 1.*

Proof: Take an instance of DOMINATING SET (graph G and k) and create k objects with valuation 1 and $|V|-k$ objects with valuation 0. We claim that there is an allocation π such that $\mathsf{Ext}_\pi(G) \geq |V| - k$ iff G admits a dominating set of size k.

If D is a dominating set of size k of G then assign the k objects of valuation 1 to D; the remaining vertices receive an object of valuation 0. We have $ext_\pi(v) = 0$ when $v \in D$ and $ext_\pi(v) = 1$ when $v \in V \setminus D$. Hence, $\mathsf{Ext}_\pi(G) \geq |V| - k$.

Conversely, if there exists an allocation π such that $\mathsf{Ext}_\pi(G) \geq |V| - k$, then let D be the vertex set where the objects of valuation 1 are located. Exactly $|V| - k$ vertices hosting a 0-valuation item are adjacent to D. Thus, D is a dominating set. ■

In order to circumvent this hardness result, we will sometimes resort to approximate solutions. Given an instance (G, O, ν) and $\rho \in (0, 1]$, an allocation π is ρ-approximate if $\mathsf{Ext}_\pi(G) \geq \rho\, \mathsf{Ext}_{\pi^*}(G)$ where π^* is an optimal solution. A ρ-approximation algorithm produces a ρ-approximate solution in polynomial time.

4 TWO VALUATIONS

This section is devoted to OPT-EXT when there are only two possible valuations for the objects. We are going to assume w.l.o.g. that the two possible valuations are either 0 or 1. This case, denoted by OPT-EXT(0,1), is **NP**-hard by Proposition 1.

As a motivation, OPT-EXT(0,1) covers the situation where there are two classes of objects. Concerning the example of placing contents on a web page such as a blog, objects can be ads and posts, respectively. Another application comes from agronomy. Some plants like actinidias are either male or female. The fruits (kiwis) grow on female trees provided that a male tree is nearby. Then, OPT-EXT(0,1) describes the situation where there are fixed locations for planting the trees, an edge indicates that two locations are close enough for fertilization, and objects valued 0 and 1 are female and male trees, respectively.

An instance of OPT-EXT(0,1) is a graph G with n vertices, k objects valued 1 and z objects valued 0, with $m = k+z \leq n$. Actually we shall see that we can assume $k + z = n$.

[5] This example does not refer to the pages produced by a search engine where externality, as it is mentioned in the related work section, is mostly negative because the user selects a single link in an ordered list. Instead, we refer to a news web site or a blog.

Proposition 2 *Regarding the approximation of* OPT-EXT(0,1)*, we can always suppose that the number of objects is equal to the number of vertices. If it is not the case then complete the instance with $n - k - z$ new objects valued* 0.

Proof: Take an instance $\mathcal{I}_1$ of OPT-EXT(0,1) on a graph G, with k objects valued 1, z_1 objects valued 0, and such that $k + z_1 < n$. Create another instance $\mathcal{I}_2$ from $\mathcal{I}_1$ where the graph is identical, the number of objects valued 1 is k and the number of objects valued 0 is $z_2 = n - k$. $\mathcal{I}_2$ contains $\delta := z_2 - z_1 > 0$ more objects valued 0 than $\mathcal{I}_1$.

Let π_1^* and π_2^* be optimal allocations for $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. Let $\hat{\pi}$ be a ρ-approximate solution for OPT-EXT(0,1) on $\mathcal{I}_2$ for any $\rho \in (0, 1]$. Thus

$$\mathsf{Ext}_{\hat{\pi}}(G) \geq \rho \mathsf{Ext}_{\pi_2^*}(G). \tag{1}$$

A *useless zero* (resp. *useful zero*) is an object valued 0 whose externality is 0 (resp. 1).

If $\hat{\pi}$ contains at most δ useless zeros then we can remove δ objects valued 0 including the useless zeros in order to get an allocation $\hat{\pi}'$ that is feasible and optimal for $\mathcal{I}_1$ since all its zeros are useful. In other words, $\mathsf{Ext}_{\hat{\pi}'}(G) = \mathsf{Ext}_{\pi_1^*}(G) \geq \rho \mathsf{Ext}_{\pi_1^*}(G)$.

Now suppose the number of useless zeros in $\hat{\pi}$ is strictly larger than δ. Remove δ useless zeros in order to get an allocation $\hat{\pi}'$ that is feasible for $\mathcal{I}_1$ and such that

$$\mathsf{Ext}_{\hat{\pi}'}(G) = \mathsf{Ext}_{\hat{\pi}}(G). \tag{2}$$

Construct a feasible allocation π of $\mathcal{I}_2$ as follows: if $\pi_1^*(v) \neq \perp$ then $\pi(v) := \pi_1^*(v)$, otherwise $\pi(v)$ gets an object valued 0. We get that

$$\mathsf{Ext}_{\pi_2^*}(G) \geq \mathsf{Ext}_{\pi}(G) \geq \mathsf{Ext}_{\pi_1^*}(G). \tag{3}$$

By the combination of Inequalities (2), (1) and (3) multiplied by ρ, we get that $\mathsf{Ext}_{\hat{\pi}'}(G) \geq \rho \mathsf{Ext}_{\pi_1^*}(G)$. To conclude, for every $\rho \in (0, 1]$, we can always derive a ρ-approximate solution for $\mathcal{I}_1$ from a ρ-approximate solution for $\mathcal{I}_2$. ■

Hence, we will suppose in this section that $m = n$.

4.1 A Constant Approximation

Our first positive result is the following theorem where $(e-1)/(1+e) \approx 0.46$.

Theorem 1 *There exists a $(e-1)/(1+e)$-approximation algorithm for* OPT-EXT(0,1).

Let us begin with an auxiliary covering problem on a graph. We are given an undirected graph G with n vertices, and two integers $b \in [n]$ and $r \in [n]$ such that $b + r \leq n$. Each vertex receives exactly one of the following colors: blue, red or white. Moreover, at most b vertices are blue, at most r vertices are red, and a vertex can be red only if it has at least one blue neighbor. A vertex is *covered* if it is blue or red. The objective is to color every vertex of G in blue, red, or white, such that the number of covered vertices is maximum.

Take an optimal blue-red-white coloring of G whose set of covered vertices is denoted by S^*. Create a partition of S^* in b sets as follows. Take the blue vertex v which has the largest number of red neighbors (break ties arbitrarily). The red vertices in $\mathcal{N}(v)$, together with v, form S_1^*. Take the next blue vertex having the largest number of red neighbors (not put in a previous set) and put it with its red neighbors in a set S_2^*, and so on. We eventually get a collection of b disjoint sets, each of them having exactly one blue vertex. Let t^* be the largest index such that $S_1^*, \ldots, S_{t^*}^*$ contain at least one red vertex. Either $t^* = b$ or $S_{t^*+1}^*, \ldots, S_b^*$ do not contain any red vertex.

An instance of the covering problem can be derived from an instance of OPT-EXT(0,1): G is the same, $b = k$, and $r = z$. A solution to OPT-EXT(0,1) can be derived from a solution to the covering problem: blue vertices receive objects with valuation 1, while red and white vertices receive objects with valuation 0. Here we assume w.l.o.g. that any solution to the covering problem has exactly k blue vertices (as many as the objects with valuation 1), because the covering problem is to maximize the number of covered vertices. Indeed, if strictly less than b vertices are blue, then we can color a red or white vertex in blue, and the number of covered vertices increases. Regarding red and white vertices, by Proposition 2, we can assume w.l.o.g. that $k + z = n$. Therefore, it is always possible that each red and white vertex receives an object with valuation 0. As a red vertex must have a blue neighbor, its externality is 1. The externality is 0 for any other vertex. Thus, the total graph externality is equal to the number of red vertices.

We use Algorithm 1 and produce an approximate solution $sol(b)$ to the covering problem on G with $b = k$ and $r = z$. In what follows, π_1 and π^* are the allocations derived from $sol(b)$ and S^*, respectively. A second allocation π_2 is obtained by constructing a maximum matching M of G. Choose $\min(k, z, |M|)$ edges of M arbitrarily. Each selected edge receives an object of value 1 and an object of value 0 on its extremities. The rest of the allocation π_2 is arbitrary.

Proof: *(of Theorem 1)* Our algorithm is to output the best allocation out of $\{\pi_1, \pi_2\}$ in terms of externality. We have

$$\mathsf{Ext}_{\pi^*}(G) = |\bigcup_{i=1}^{t^*} S_i^*| - t^* \tag{4}$$

and

$$\mathsf{Ext}_{\pi_1}(G) \geq |sol(t^*)| - t^*. \tag{5}$$

The proof of the following inequality is obtained with arguments similar to those of [20, Lemma 3.14].

$$|sol(t^*)| \geq (1 - 1/e)|\bigcup_{i=1}^{t^*} S_i^*| \tag{6}$$

Inequalities (5) and (6) give us

$$\mathsf{Ext}_{\pi_1}(G) \geq (1 - 1/e)|\cup_{i=1}^{t^*} S_i^*| - t^*. \tag{7}$$

For any $\alpha \in (0, 1)$ we can change Inequality (7) for

$$\mathsf{Ext}_{\pi_1}(G) \geq (1 - 1/e)|\cup_{i=1}^{t^*} S_i^*| - \alpha t^* - (1 - \alpha)t^*. \tag{8}$$

Algorithm 1

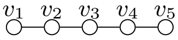
Input: b, r, G of order n
Output: A feasible solution to the covering problem

```
Every vertex of G is white.
sol(0) ← ∅
for t = 1 to b do
    Color in blue a vertex v that is not already blue, and if possible, color in red some white neighbors of v. Do this so as to maximize the number of newly covered vertices, under the constraint that the total number of red vertices is at most r.
    sol(t) ← sol(t − 1) ∪ {the newly covered vertices}
end for
return sol(b)
```

If $t^* \leq |\bigcup_{i=1}^{t^*} S_i^*|/(3 + \delta)$ for $\delta := \frac{3-e}{e-1}$, then Inequality (8) becomes $\mathsf{Ext}_{\pi_1}(G) \geq (1 - 1/e - \alpha/(3+\delta))|\cup_{i=1}^{t^*} S_i^*| - (1 - \alpha)t^*$. Let $\alpha = (3+\delta)/e(2+\delta)$ to get that $\mathsf{Ext}_{\pi_1}(G) \geq (1 - (3+\delta)/e(2+\delta)) \left(|\cup_{i=1}^{t^*} S_i^*| - t^*\right) = (e - 1)/(1 + e)\mathsf{Ext}_{\pi^*}(G)$.

It remains to study the case $|\bigcup_{i=1}^{t^*} S_i^*|/(3 + \delta) < t^*$. The number of red vertices in π^* is at most $|\bigcup_{i=1}^{t^*} S_i^*| - t^* < (2+\delta)t^*$. Therefore $\mathsf{Ext}_{\pi^*}(G) < (2 + \delta)t^*$ by (4). Allocation π_2 is a solution of size at least $t^* > \mathsf{Ext}_{\pi^*}(G)/(2 + \delta)$ since in π^* at least one red vertex is attached to each of the t^* blue vertices. Therefore π_2 is a $1/(2 + \delta)$-approximation in this case, with $1/(2 + \delta) = (e - 1)/(1 + e)$. ■

Example 2 *Consider the following graph with 2 objects of value 1 and 3 objects with value 0.*



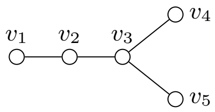
The approximation algorithm can place the objects of value 1 on v_3 and v_5; the resulting externality is 2. The optimal solution, of externality 3, places the objects of value 1 on vertices v_2 and v_4. Thus, the approximation ratio of the algorithm is at most $2/3$.

Discussion: There is a gap between our lower and upper bounds, i.e. $(e - 1)/(1 + e) \approx 0.46$ and $2/3$. This naturally triggers the question of determining the exact approximability of OPT-EXT(0,1).

There are greedy algorithms for approximating *coverage problems* whose objective function is *monotone submodular*; see for example [27] for a $(1 - (1 - 1/k)^k)$-approximation algorithm where $(1 - (1 - 1/k)^k) \geq 1 - 1/e$. A function Φ is submodular over Ω if, for every X, Y such that $X \subseteq Y \subseteq \Omega$, and $u \in \Omega \setminus Y$, $\Phi(X \cup \{u\}) - \Phi(X) \geq \Phi(Y \cup \{u\}) - \Phi(Y)$. Moreover, Φ is monotone if for every X, Y such that $X \subseteq Y \subseteq \Omega$, $\Phi(X) \leq \Phi(Y)$.

As Theorem 1 is reminiscent of the known results for coverage problems (i.e. same kind of algorithm and ratio), we need to clarify why, from our understanding, OPT-EXT(0,1) does not reduce to the maximization of a monotone submodular function.

Given an instance of OPT-EXT(0,1), define a function $f : 2^V \to \mathbb{N}$ as follows. For any $S \subseteq V$ such that $|S| \leq k$, $f(S)$ is the minimum between z and the number of vertices in $V \setminus S$ which have a neighbor in S. In other words, if every vertex of S receives an object of value 1 while the neighbors of S (at most z) receive objects valued 0, then $f(S)$ is the externality of the current solution. One can see with the following instance that f is neither submodular nor monotone.



3 objects with value 1

2 objects with value 0

Let $X = \{v_2\}$, $Y = \{v_2, v_3\}$ and $u = v_1$. Then, $f(\{v_1, v_2\}) - f(\{v_2\}) = 1 - 2 < 2 - 2 = f(\{v_1, v_2, v_3\}) - f(\{v_2, v_3\})$.

In addition, we have just seen that Algorithm 1 outputs a $2/3$-approximate solution for the instance of Example 2 where $k = 2$. Since $(1 - (1 - 1/k)^k) = 3/4$ when $k = 2$, $(1 - (1 - 1/k)^k)$ cannot be the correct ratio of Algorithm 1.

Hence, it is important to stress a difference between OPT-EXT(0,1) and coverage problems. For MAX COVERAGE there is a clear separation between elements to be covered and sets that cover them. Furthermore a vertex of a dominating set dominates itself and its neighbors. In contrast, a vertex in OPT-EXT(0,1) can either derive or exert externality, but not both.

4.2 OPT-EXT$(0,1)$ and Partial Domination

This section deals with exact algorithms for OPT-EXT(0,1) which are built upon exact algorithms for PARTIAL DOMINATING SET (PDS for short). For a given integer $t \geq 0$, the goal of PDS is to dominate at least t vertices with the minimum number of vertices. Let D_t denote an optimal set of vertices that dominates t vertices of G. Following Demaine *et al.* [9], D_t can be computed in time $3^{1.5\mathsf{tw}} n^{\mathcal{O}(1)}$ where tw and n are the treewidth and the number of vertices of G, respectively. Later, Fomin *et al.* [11] proposed a subexponential algorithm for PDS in apex-minor-free graphs (this class comprises planar graphs) which runs in $2^{\mathcal{O}(\sqrt{s})} n^{\mathcal{O}(1)}$ where $s := |D_t|$.

Proposition 3 *Every algorithm that solves* PDS *in $T(n)$ time gives a $nT(n)$ time exact algorithm for* OPT-EXT(0,1).

Proof: Let D_t denote a minimum size set of vertices that dominates at least t vertices of G. Suppose D_t can be computed in $T(n)$ time. Compute D_t for $t = 1$ to n. For each D_t, let H_t be the set of vertices of $V \setminus D_t$ that are dominated by at least one element of D_t.

Let $t^\star$ be the index such that $|H_t|$ is maximized under the constraint $|D_t| \leq k$. Build a solution π of OPT-EXT(0,1) as follows. Every vertex of $D_{t^\star}$ gets an object valued 1. Place objects valued 0 on the vertices of $H_{t^\star}$, until $H_{t^\star}$ is full, or we run out of objects valued 0. If necessary, complete the solution arbitrarily, i.e. the free vertices get the remaining objects.

If all the objects valued 0 are in $H_{t^\star}$, then π is optimal ($\mathsf{Ext}_\pi(G)$ cannot be larger, since every 0 object has externality 1). Otherwise, at least one object valued 0 does not get externality. Suppose π is not optimal and let $\tilde{\pi}$ be an allocation maximizing Ext. Let $D_{\tilde{t}}$ be the set of vertices hosting an object valued 1 in $\tilde{\pi}$. Let $H_{\tilde{t}}$ be the set of vertices hosting an object valued 0 and deriving positive externality in $\tilde{\pi}$. We get that $|D_{\tilde{t}}| \leq k$ and $|H_{\tilde{t}}| > |H_{t^\star}|$, contradiction. ■

5 GENERAL VALUATIONS

This section is devoted to OPT-EXT where the objects' valuations are general (not restricted to 0 and 1). Proposition 2 does not apply so we do not assume that $|O| = |V|$. We start with two useful tools: a graphical representation of the externality and a reformulation of Ext_π.

Given an allocation π and $G = (V, E)$, we associate a digraph $\mathcal{D}$ with vertex set V and arc set A. There is an arc $(v_i, v_j) \in A$ if v_i derives externality from v_j. If two neighbors v_i, v_j host two objects having the same valuation then A contains either (v_i, v_j) or (v_j, v_i) (choose one arbitrarily). As a vertex (resp. object) derives utility from at most one neighbor, each vertex of $\mathcal{D}$ has outdegree at most 1. The digraph associated with the instance of Example 1 has 3 arcs: (v_3, v_1), (v_5, v_1), and (v_4, v_5).

The graph externality can be formulated as a dot product:

$$\mathsf{Ext}_\pi(G) := \sum_{v \in V \text{ s.t. } \pi(v) \neq \perp} h(v) \cdot \nu(\pi(v)) \tag{9}$$

where $h(v)$ is defined as the in-degree of v in $\mathcal{D}$ minus the out-degree of v in $\mathcal{D}$. Said differently, $h(v)$ is the number of vertices deriving externality from v, minus 1 if v derives externality from one of its neighbors.

5.1 When the Graph has Maximum Degree 2

A graph with maximum degree 2 is a collection of paths and cycles. We are going to see that this case can be solved efficiently. In contrast, OPT-EXT is **NP**-hard when the maximum degree is 3 by Proposition 1.

Theorem 2 OPT-EXT *can be solved in polynomial time when G has degree at most 2.*

The allocation is built in two phases. During the first phase, G is partially covered with a collection $P_1, \ldots, P_z$ of disjoint paths. Each path P_ℓ has length at most 2 (the length of a path is the number of its edges, or equivalently, its number of vertices minus 1). In total the collection of paths covers exactly $|O|$ vertices. The first phase is done with Algorithm 2. The final allocation π is constructed during the second phase with the help of Algorithm 3.

Proof: (of Theorem 2) Consider the digraph $\mathcal{D}^*$ associated with G and π^* where π^* is an optimal solution to OPT-EXT. We are going to describe an operation called *Reversal*. Suppose $\mathcal{D}^*$ contains a directed path of length at least 2: $((v_i, v_{i+1}), \ldots, (v_{i+k-1}, v_{i+k})$. Modify π^* by reversing the allocation between v_i and v_{i+k-1}. That is, v_{i+k-1} gets the object of v_i, v_{i+k-2} gets the object of v_{i+1}, and so on. It is not difficult to see that the total externality does not decrease because $(\nu(i+k) - \nu(i+k-1)) + (\nu(i+k-1) - \nu(i+k-2)) + \ldots + (\nu(i+1) - \nu(i)) = \nu(i+k) - \nu(i)$ where the left and right parts are the contributions to the total externality before and after the reversal, respectively.

Do *Reversal* on π^* until it is not possible. The process is finite because each operation strictly decreases the number of pairs of consecutive arcs. The value $\mathsf{Ext}_{\pi^*}(G)$ has not decreased, π^* remains an optimal solution. The digraph $\mathcal{D}^*$ associated with π^* consists of connected components of at most 3 vertices. Therefore $h(v) \in \{2, 1, 0, -1\}$ for any vertex v because there are 4 possible situations: in-degree 2 and out-degree 0, in-degree 1 and out-degree 0, in-degree 0 and out-degree 0, and in-degree 0 and out-degree 1.

Observe that if there are two vertices v and v' such that $\nu(\pi(v)) > \nu(\pi(v'))$ and $h(v) < h(v')$ then we can swap their objects and strictly increase $\mathsf{Ext}(\pi^*)$. Since π^* is optimal, it holds that $\nu(\pi(v)) \leq \nu(\pi(v'))$ when $h(v) < h(v')$. The objects taken by non increasing valuation are placed on the vertices with non increasing h-value in π^*. Thus, $\mathsf{Ext}_{\pi^*}(G)$ is a dot product $\vec{\nu} \cdot \vec{x}$ where $\vec{\nu}$ (resp. $\vec{y}$) consists of $\{\nu(o) : o \in O\}$ (resp. $\{h(v) : v \in V\}$) sorted in non increasing order.

Take the output π of Algorithms 3 and let $\mathcal{D}$ be its associated digraph $\mathcal{D}$. As for the optimal solution, it consists of connected components of at most 3 vertices, and the objects taken by non increasing valuation are placed on the vertices with non increasing h-value. Thus, $\mathsf{Ext}_\pi(G)$ is also a dot product $\vec{\nu} \cdot \vec{y}$ where $\vec{\nu}$ (resp. $\vec{x}$) consists of $\{\nu(o) : o \in O\}$ (resp. $\{h(v) : v \in V\}$) sorted in non increasing order.

The possible difference between $\vec{x}$ and $\vec{y}$ comes from the h-values. By definition, the sum of the coordinates of both $\vec{x}$ and $\vec{y}$ is 0. Since $\vec{y}$ is, by construction, lexicographically larger than $\vec{x}$, it follows that $\mathsf{Ext}_\pi(G) = \vec{\nu} \cdot \vec{y} \geq \vec{\nu} \cdot \vec{x} = \mathsf{Ext}_{\pi^*}(G)$. Algorithms 2 and 3 solve OPT-EXT optimally. ■

5.2 When the Graph is a Special Tree

Many **NP**-hard problems on graphs can be solved in polynomial time on trees (e.g. DOMINATING SET). Trees for OPT-EXT are meaningful as they represent a hierarchy. Though we leave the computational complexity of OPT-EXT open for trees, this section contains positive results for special cases.

Algorithm 2

Input: $|O|$ and G which has maximum degree 2
Output: A set of disjoint sub-paths of G, each of length at most 2, which spans exactly $|O|$ vertices of G

Remove an arbitrary edge of each cycle of G so that G becomes a collection of paths
$spn \leftarrow 0$ $\{spn$ is the number of vertices spanned so far$\}$
$z \leftarrow 0$ $\{z$ is the number of sub-paths built so far$\}$
while $spn < |O|$ **do**
  $z \leftarrow z + 1$
  Let s be the minimum between 3, 1+the length of the longest path of G, and $|O| - spn$
  Choose a sub-path of G on s vertices and call it P_z (P_z must contain a vertex whose degree in G is 1)
  $G \leftarrow G \setminus P_z$
  $spn \leftarrow spn - s$
end while
return $P_1, \ldots, P_z$

Algorithm 3

Input: G, O and a collection $P_1, \ldots, P_z$ of paths of length at most 2
Output: An allocation π

In the collection, each path P_ℓ of length 2 consists of 3 contiguous vertices whose center is denoted by c_ℓ. Each path P_ℓ of length 1 consists of 2 contiguous vertices; choose one of them arbitrarily to be the center c_ℓ. Each path of length 0 consists of a single vertex which is the center
π is initially empty
for $\ell = 1$ to z **do**
  Let o^* be the object with largest valuation in O
  $\pi(c_\ell) \leftarrow o^*$
  $O \leftarrow O \setminus \{o^*\}$
end for
Complete π by placing arbitrarily the remaining objects on the free vertices (i.e. the non-centers) of $P_1, \ldots, P_\ell$
return π

Proposition 4 OPT-EXT *can be solved in polynomial time when G is a collection of stars.*

Sketch of Proof: If $|O| = |V|$ then run an algorithm which takes the objects by non increasing order of valuation and assign them to the vertices sorted by non increasing order of their degree.

If $|O| < |V|$, order the vertices of G such that the center of every star comes before its leaves, and for every pair of stars S and S' such that $|S| > |S'|$, the elements of S get smaller indices than the elements of S'. Apply the previous algorithm on the sub-graph of G consisting of the $|O|$ first vertices. ■

A *caterpillar* consists of a path of β vertices $v_1, \ldots v_\beta$, also called the *backbone*, and $n - \beta$ pendant edges [18]. Each vertex $v_{\beta+1}, \ldots, v_n$ has exactly one neighbor in the backbone. We are going to propose a 0.5-approximation algorithm for the case of a caterpillar which relies on solving OPT-EXT on a sub-graph of G which is a path, and when the number of available objects exceeds the number of vertices (Theorem 2 does not apply here). The result relies on the following Lemma (omitted proof based on the proof of Theorem 2).

Lemma 1 OPT-EXT *can be solved in polynomial time on a path having strictly less than $|O|$ vertices.*

Proposition 5 *A 0.5-approximation algorithm exists for* OPT-EXT *when G is a caterpillar.*

Proof: For an optimal allocation π^*, the externality along the backbone edges and the pendant edges, which are disjoint, are denoted by $\mathcal{E}_b^*$ and $\mathcal{E}_p^*$, respectively. An optimal allocation π_1 for the backbone edges (resp. π_2 for the pendant edges) can be found in polynomial time, see Lemma 1 (resp. Proposition 4). Since $\mathsf{Ext}_{\pi_1}(G[\{v_1 \ldots, v_\beta\}]) \geq \mathcal{E}_b^*$ and $\mathsf{Ext}_{\pi_2}(G[\{v_{\beta+1} \ldots, v_n\}]) \geq \mathcal{E}_p^*$, the best solution out of $\{\pi_1, \pi_2\}$ is 0.5-approximate. ■

5.3 Experimental Results

In this section, we propose a greedy algorithm for the general valuation case and evaluate it experimentally by comparing its solutions against an efficiently computable nontrivial upper bound on the optimal externality. We will next describe how to obtain this upper bound.

Algorithm 4

Input: G, O
Output: An allocation π

Color every vertex of G white
Mark all objects in O as available
while there exist available objects in O **do**
  Let o be the object of largest valuation among available objects in O, and v be the vertex of G that has the largest number of white vertices in its closed neighborhood and is either red with a valuation $\nu(\pi(v)) < \nu(o)$, or white
  if v is red **then**
    Mark object $\pi(v)$ as available
  end if
  $\pi(v) \leftarrow o$; Color v blue
  Mark object o as unavailable (allocated)
  while there exists a white neighbor w of v **do**
    Let o' be the object of smallest valuation among available objects in O
    $\pi(w) \leftarrow o'$; Color w red and mark o' as unavailable
  end while
end while
return π

Let us first define a trivial upper bound $T(G)$ on the optimal externality. Given $G(V, E)$ and O, consider a star S_T of degree $|O| - 1$, with the object of maximum valuation on its center and all other objects on its leaves. Let objects be sorted in non-increasing order of valuation, i.e., $\nu(o_1) \geq \nu(o_2) \geq \ldots \geq \nu(o_m)$. A leaf of S_T that is assigned an object o_i derives externality from the center equal to $\nu(o_1) - \nu(o_i)$. Let $T(G)$ be the total externality of S_T, i.e., $T(G) = \nu(o_1) \cdot |O| - \sum_{i=1}^{m} \nu(o_i)$.

In order to define the nontrivial upper bound $U(G)$, let $\{d_1, d_2, ..., d_n\}$ be the degrees of V in non-increasing order, and let $\{v_1, v_2, ..., v_n\}$ be the corresponding vertices. Consider k to be the smallest integer satisfying

$$k + \sum_{i=1}^{k} d_i \geq |O|. \tag{10}$$

Consider also a collection of stars S_U with centers $\{v_1, v_2, ..., v_k\}$, each center having the same neighbors as in G (note that some vertices of G may appear multiple times in S_U). Assign the objects with the largest k valuations to the centers of S_U, i.e. the object of valuation $\nu(o_i)$ is assigned to v_i, $1 \leq i \leq k$. Objects of smallest valuation are assigned to neighbors of $v_1, v_2, \ldots$, in non-decreasing order of their valuation. Note that some neighbors of v_k may not get any object. Let $U(G)$ be the total externality that leaves of S_U derive under this allocation. Clearly, the above allocation may not be feasible for the original instance as it may assign multiple objects to the same vertex. However it provides an upper bound on the optimal externality which is usually better than the trivial one.

Proposition 6 *For any instance (G, O) of* OPT-EXT *and any allocation π,* $\mathsf{Ext}_\pi(G) \leq U(G) \leq T(G)$.

Sketch of Proof: Let us first note that $T(G)$ is clearly an upper bound on the externality of any allocation of objects from O since all objects but the 'heaviest' one derive the maximum possible externality; the heaviest object cannot derive externality under any allocation.

Given the partition into stars induced by an allocation π, we can convert it to S_U (the collection of stars giving the bound $U(G)$) as follows. Let us first 'isolate' the k (as given by Eq. 10) stars centered on vertices with the largest valuation objects $o_1, \ldots, o_k$ in π, then move any leftover objects to the leaves of these k stars (that include all the neighbors of the centers in G). Then, we obtain $U(G)$ by re-allocating the objects of centers (in case the k centers are not exactly the same as in $U(G)$) and then the objects of leaves in a greedy manner; such reallocations can only increase the total externality since we may move objects that are on centers to other centers of at least the same degree and objects that are on leaves to stars with center of at least the same valuation. ■

We run Algorithm 4 on 10 DIMACS datasets from the 10th DIMACS Implementation Challenge [10]. We chose both sparse and dense datasets. The chosen valuations follow the uniform distribution, with values ranging from 0 to $4|V|$. The benchmarks we used are shown in Table 1, sorted in non-decreasing order of density $|E|/|V|$.

Benchmark Characteristics			
ID	**Name**	**# Vertices**	**# Edges**
b1	karate	34	78
b2	dolphins	62	159
b3	lesmis	77	254
b4	adjnoun	112	425
b5	polbooks2	105	441
b6	chesapeake	39	170
b7	celegans_metabolic	453	2025
b8	football	115	613
b9	celegansneural	297	2148
b10	jazz	198	2742

Table 1. List of used benchmarks

Figure 1 shows how Algorithm 4 performs compared to the upper bound $U(G)$. Although $U(G)$ is clearly an overestimation, We observe that the difference does not exceed 5% in any of the tested benchmarks. We also observe a correlation between the density of the graph and the achieved ratio of the externality obtained to $U(G)$. This can be explained by the fact that the denser the graphs, the lower the number of stars in S_U (the collection of stars yielding $U(G)$) will be, and Algorithm 4 is more likely to produce a similar star partition. Certain singularities, most strikingly the Karate dataset, can also be explained: the Karate dataset is famous as an archetypical example in community detection papers [16]; it contains 2 'leaders', with whom almost all other vertices are connected; this is a case for which the greedy approach works quite well.

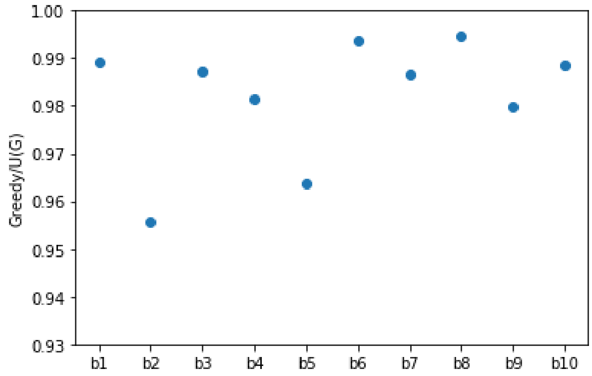


Figure 1. Percentage of $U(G)$ achieved by Algorithm 4

We also run Algorithm 4 on several randomly generated graphs. Instances of 50, 100, 150, 200 and 300 vertices were tested. For each number of vertices $|V|$, connected graphs of $|V| - 1$ up to $20|V|$ edges were created, by generating a tree from a random Prüfer sequence and then filling the remaining number of edges following the $G(n, p)$ random graph model. Figure 2 shows how the ratio of the solution of Algorithm 4 to $U(G)$ increases almost logarithmically as the density of the graph increases. We also note that for the same density, our algorithm achieves better results on graphs of smaller size; this is due to the fact that the smaller the size, the closer we get to a clique graph, for which the greedy approach would give the optimal solution (for a clique $\mathsf{Ext}_{OPT}(G) = U(G) = T(G)$). Therefore, it seems that the higher the ratio density/size, the better the quality of the solution becomes.

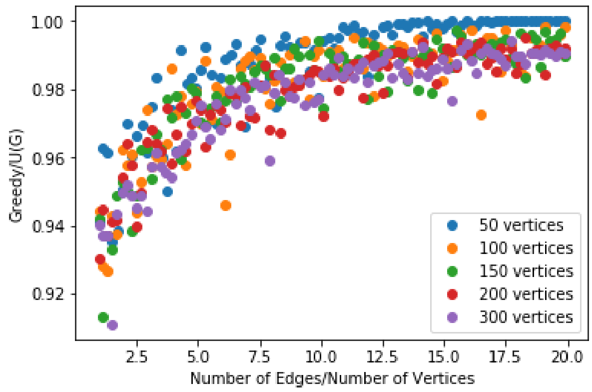


Figure 2. Density of random graphs and Greedy/$U(G)$

6 CONCLUSION

This paper investigated the impact of externality with a new model which focuses on two aspects: locality and positiveness. We took the viewpoint of a single agent who wants to place a given set of goods on some locations so as to maximize the total externality.

OPT-EXT is connected to coverage problems. However the fact that a vertex cannot host a good which exerts and derives externality at the same time makes the problem special. Our findings trigger stimulating open questions. Chief among them is to design an algorithm with a performance guarantee for OPT-EXT with general valuations. Our experiments indicate that the greedy strategy is a good candidate. For special graphs such as trees, where coverage problems are often solvable in polynomial time, it would be interesting to determine the complexity of OPT-EXT with general valuations.

Regarding the two valuation case, the exact approximation ratio of the greedy algorithm remains open since our upper bound of $2/3$ does not match with our lower bound of $(e-1)/(1+e) \approx 0.46$. We believe that both bounds can be improved.

Finally, there are avenues of possible extensions of the model. Here we assumed that externality is due to a single neighboring object whose value is the largest. However, an object may derive externality from multiple sources instead of a single one. In this case several meaningful ways to aggregate the externalities can be proposed (a weighted sum for example). Externality can be either positive or negative depending on whether the objects are (possibly competing) goods or bads. Finally, the model can incorporate non deterministic externalities in the future.

REFERENCES

[1] N. AhmadiPourAnari, S. Ehsani, M. Ghodsi, N. Haghpanah, N. Immorlica, H. Mahini, and V. S. Mirrokni, 'Equilibrium pricing with positive externalities', *Theor. Comput. Sci.*, **476**, 1–15, (2013).

[2] S. Athey and G. Ellison, 'Position auctions with consumer search', *The Quarterly Journal of Economics*, **126**(3), 1213–1270, (2011).

[3] A. Beynier, Y. Chevaleyre, L. Gourvès, J. Lesca, N. Maudet, and A. Wilczynski, 'Local envy-freeness in house allocation problems', in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pp. 292–300, (2018).

[4] *Handbook of Computational Social Choice*, eds., F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Cambridge University Press, 2016.

[5] S. Brânzei, T. P. Michalak, T. Rahwan, K. Larson, and N. R. Jennings, 'Matchings with externalities and attitudes', in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pp. 295–302, (2013).

[6] S. Brânzei, A. D. Procaccia, and J. Zhang, 'Externalities in cake cutting', in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 55–61, (2013).

[7] R. Bredereck, A. Kaczmarczyk, and R. Niedermeier, 'Envy-free allocations respecting social networks', in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS-18)*, pp. 283–291, Stockholm, Sweden, (July 2018). International Foundation for Autonomous Agents and Multiagent Systems.

[8] G. J. Chang, 'Algorithmic aspects of domination in graphs', Technical report, Department of Mathematics, National Taiwan University, (2011). Available at http://www.math.ntu.edu.tw/∼mathlib/preprint/2011-01.pdf.

[9] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, 'Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs', *J. ACM*, **52**(6), 866–893, (2005).

[10] Dimacs. www.cc.gatech.edu/dimacs10/archive/clustering.shtml.

[11] F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh, 'Subexponential algorithms for partial cover problems', *Inf. Process. Lett.*, **111**(16), 814–818, (2011).

[12] D. Fotakis, P. Krysta, and O. Telelis, 'Externalities among advertisers in sponsored search', in *Algorithmic Game Theory, 4th International Symposium, SAGT 2011, Amalfi, Italy, October 17-19, 2011. Proceedings*, pp. 105–116, (2011).

[13] D. Fotakis and P. Siminelakis, 'On the efficiency of influence-and-exploit strategies for revenue maximization under positive externalities', *Theor. Comput. Sci.*, **539**, 68–86, (2014).

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

[15] A. Ghosh and M. Mahdian, 'Externalities in online advertising', in *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pp. 161–168, (2008).

[16] M. Girvan and M. E. J. Newman, 'Community structure in social and biological networks', *Proceedings of the National Academy of Sciences*, **99**(12), 7821–7826, (2002).

[17] N. Haghpanah, N. Immorlica, V. S. Mirrokni, and K. Munagala, 'Optimal auctions with positive network externalities', in *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011)*, pp. 11–20, (2011).

[18] F. Harary and A. J. Schwenk, 'The number of caterpillars', *Discrete Mathematics*, **6**(4), 359 – 365, (1973).

[19] J. Hartline, V. S. Mirrokni, and M. Sundararajan, 'Optimal marketing strategies over social networks', in *Proceedings 17th Conference on World Wide Web (WWW-2008)*, pp. 189–198, (2008).

[20] *Approximation Algorithms for NP-Hard Problems*, ed., D. S. Hochbaum, PWS Publishing Company, Boston, MA, USA, 1997.

[21] P. Hummel and R. P. McAfee, 'Position auctions with externalities', in *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, pp. 417–422, (2014).

[22] M. L. Katz and C. Shapiro, 'Network externalities, competition, and compatibility', *The American Economic Review*, **75**(3), 424–440, (1985).

[23] D. Kempe, J. M. Kleinberg, and É. Tardos, 'Maximizing the spread of influence through a social network', in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pp. 137–146, (2003).

[24] P. Krysta, T. P. Michalak, T. Sandholm, and M. Wooldridge, 'Combinatorial auctions with externalities', in *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, pp. 1471–1472, (2010).

[25] R. Paes Leme, V. Syrgkanis, and É. Tardos, 'Sequential auctions and externalities', in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pp. 869–886, (2012).

[26] M. Li, J. Zhang, and Q. Zhang, 'Truthful cake cutting mechanisms with externalities: Do not make them care for others too much!', in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 589–595. AAAI Press, (2015).

[27] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, 'An analysis of approximations for maximizing submodular set functions - I', *Math. Program.*, **14**(1), 265–294, (1978).

[28] E. Pataki, A. Sagi, and K. Jozef, 'Externalities and the optimal allocation of economic resources', in *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 185–188, (Sep. 2015).

[29] J. Rohlfs, 'A theory of interdependent demand for a communications service', *The Bell Journal of Economics and Management Science*, **5**(1), 16–37, (1974).

[30] M. Seddighin, H. Saleh, and M. Ghodsi, 'Externalities and fairness', in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pp. 538–548, (2019).

[31] R. A. Velez, 'Fairness and externalities', *Theoretical Economics*, **11**, 381–410, (2016).