A Computational Study of the Perfect Awareness Problem

Felipe de C. Pereira[1], **Pedro J. de Rezende**[1*], **Cid C. de Souza**[1†]

[1]Institute of Computing, University of Campinas, Campinas, Brazil

{felipe.pereira,rezende,cid}@ic.unicamp.br

Abstract. *The Perfect Awareness Problem (*`PAP`*) is a combinatorial optimization problem that models the spreading of information in social networks. The objective is to find a smallest subset of individuals that are able to start a viral propagation whereby a given news reaches everyone on a network, under certain dissemination restrictions. Considering that* `PAP` *is* NP*-hard, we present four novel heuristics based on the metaheuristic* `GRASP` *and show that the best one of our methods outperforms the only previously known heuristic designed for the problem. Our contributions also include: (i) a new publicly available benchmark of 840 instances that simulate social network relations, (ii) procedures for preprocessing instances, and (iii) two integer programming models to generate exact solutions for* `PAP`*. We conducted an exhaustive set of comparative experiments, followed by statistical analyses, showing the efficacy and efficiency of our algorithms.*

1. Introduction

Studies on social networks have been carried out by social scientists since the last century but, for a long time, researchers only had small datasets at their disposal, due to technological limitations. However, the emergence of the Internet led to the advent of online social networks, such as Facebook, Instagram, Twitter, etc. These networks contributed to an increase in availability of large amounts of data, resulting in a growth in interest, both in academia and industry, for problems involving social networks [Chen et al. 2013].

Much of the research carried out on online networks falls into an area of study called *analysis of influence and information propagation in social networks*. In this area, a noteworthy topic that covers a wealth of research within the field of computer science is known as *viral marketing* [Chen et al. 2013].

Suppose that a Facebook user made a post about a new song by his favorite artist. When a friend of this user comments on or shares the post, the information is passed on to members of a second level of relationship and so forth, with the potential of propagating throughout some portion of the network. Thus, if an artist wants his music to be known to many Facebook users, an immediate problem is to determine "good" first users who should make posts about his/her song in order to achieve the goal.

Provided that the process of disseminating information has been modeled, it is possible to formulate computational problems that aim to optimize the reach of data propagation throughout a network. Essentially, the goal of viral marketing is to "activate" an

[*]Advisor
[†]Co-advisor

ideally small number of influencers on a social network so that a large number of other individuals become influenced after the resulting propagation [Chen et al. 2013].

Experimental studies that investigate the way in which dissemination takes place in most social networks indicate that information does not flow freely within a network, but requires active sharing, which, in turn, depends on individual conviction before passing it on [Cordasco et al. 2019]. Moreover, individuals who become aware of an information from their own friends are known to be significantly more inclined to spread it, and to do so sooner, than those who are informed of it by unacquainted sources [Bakshy et al. 2012].

These are some of the primary observations that underlie a propagation model associated with the Perfect Awareness Problem (`PAP`) – a combinatorial optimization problem originally proposed in [Cordasco et al. 2019].

This article constitutes a summary of the Dissertation [Pereira 2021] presented by Felipe de C. Pereira, on March 22nd, 2021, to the Institute of Computing of the University of Campinas, as part of the requirements for obtaining the title of Master in Computer Science. The research described therein aimed to investigate `PAP` with the objective of designing effective and efficient algorithms for its solution.

2. The Perfect Awareness Problem

The input for `PAP` consists of a pair (G, t), where $G = (V, E)$ is an undirected graph and $t : V \to \mathbb{Z}^+$ is a *threshold function*. The vertex set V corresponds to the collection of individuals in a social network and the edge set E represents communications between individuals.[1] That is, $u, v \in V$ and $\{u, v\} \in E$ if and only if u and v can communicate with each other on the network. We denote by $N(v) = \{u \in V \mid \{u, v\} \in E\}$ the neighborhood of v in G and we say that $t(v)$ is the *threshold* of v. More on this, below.

For this problem, the passage of time is discretized in *rounds*, denoted by $\tau \in \mathbb{N}$. In every round τ, each vertex v assumes at least one state, among the three described below, related to the information being propagated:

- Ignorant: if v has not received the information from any of its neighbors and, therefore, is not aware of it;
- Aware: if v received the information from at least one of its neighbors and, consequently, v is aware of it;
- Spreader: if v is one of the initial influencers or received the information from at least $t(v)$ of its neighbors. In either case, v spreads the information.

Notice that a spreader is necessarily aware, also.

The spreading process starts at round $\tau = 0$ from an initial non-empty set of individuals $S \subseteq V$, called a *seed set*. The vertices of the seed set are called *seeds*. In a propagation started from S, the set of vertices that are spreaders at round τ is denoted by $\mathrm{Spr}[S, \tau]$. That is, $\mathrm{Spr}[S, \tau]$ equals:

- S, if $\tau = 0$;
- $\mathrm{Spr}[S, \tau - 1] \cup \{u : |N(u) \cap \mathrm{Spr}[S, \tau - 1]| \geq t(u)\}$, if $\tau \geq 1$.

[1]Note that Facebook is a prime example of a social network that fits this representation well.

In other words, at round $\tau = 0$, the spreaders are the seeds and, for $\tau \geq 1$, the set of spreaders is obtained by merging the vertices that were spreaders at round $\tau - 1$ with every vertex u that had, at round $\tau - 1$, a number of spreader neighbors greater than or equal to $t(u)$. Note that for all $\tau \geq 1$, it holds that $\text{Spr}[S, \tau - 1] \subseteq \text{Spr}[S, \tau]$.

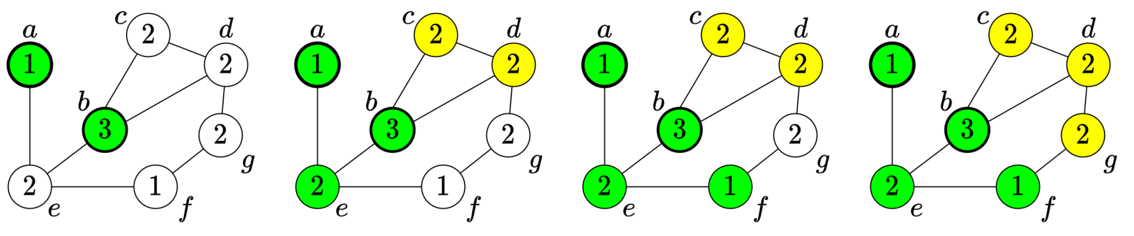
The propagation ends when $\text{Spr}[S, \rho] = \text{Spr}[S, \rho - 1]$ for some $\rho \geq 1$, i.e., when no new spreaders arise in the transition between two consecutive rounds. The final sets of vertices that are spreaders or (merely) aware are denoted and described, respectively, by:

- $\text{Spr}[S] = \text{Spr}[S, \rho]$;
- $\text{Awr}[S] = S \cup \{u : |N(u) \cap \text{Spr}[S]| \geq 1\}$.

In other words, the final set of aware vertices is formed by the seeds along with every vertex u that has at least one spreader neighbor at round ρ. Furthermore, a seed set S is called a *perfect seed set*, if $\text{Awr}[S] = V$, that is, if, at the end of the propagation, all vertices in V are aware.

Problem (PAP)**.** *Given an instance (G, t), the objective of* PAP *is to find a perfect seed set of minimum size.*

Observe that, for PAP, a solution is considered feasible if and only if it is a perfect seed set. Figure 1 exemplifies the propagation process on an instance of PAP, where the thresholds associated with the vertices are indicated inside the circles. At each round, the spreader, aware and ignorant vertices are represented in green, yellow and white, respectively. For this example, the chosen seed set consists of the vertices a and b, and it constitutes an optimal solution.



(a) $\tau = 0$ (b) $\tau = 1$ (c) $\tau = 2$ (d) $\tau = 3$

Figure 1. Example of a propagation in PAP.

It is known that PAP cannot be approximated within a ratio of $\mathcal{O}(2^{\log^{1-\varepsilon} n})$, for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$ [Cordasco et al. 2019] and, therefore, the problem is NP-hard. This result also holds when $t(v) \leq 2$ for all $v \in V$. In [Cordasco et al. 2019], a heuristic for PAP (referred to, here, as CGR) is presented, which runs in $\mathcal{O}(|E| \log |V|)$. Also, a linear time exact algorithm for PAP is described but only for when G is a tree.

Moreover, if $t(v) = d(v)$ for all $v \in V$, where $d(v)$ denotes the degree of v in G, PAP corresponds to the Minimum Dominating Set Problem (MDSP) [Grandoni 2006], a classic NP-hard problem known to be hard to approximate [Hochbaum 1997]. By extending existing results for MDSP, in [Cordasco et al. 2019] it is shown that if $t(v) = d(v)$ for any vertex v, PAP cannot be approximated within a ratio of $(1 + o(1)) \ln \Delta$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$, where Δ denotes the maximum degree in G.

There are also theoretical results for PAP when we consider a class of dense graphs called *Ore graphs*. A graph $G = (V, E)$ is an Ore graph, if $d(u) + d(v) \geq |V|$ for all $u, v \in V$ such that $\{u, v\} \notin E$. In [Cordasco et al. 2019], it is proved that if G is an Ore graph, then there is a perfect seed set for G of size at most:

$$\min\left\{3 + \left\lfloor \frac{\delta^2 - 5\delta + 6}{|V| - \delta} \right\rfloor, \left\lfloor \frac{|V|}{4} \right\rfloor + 1, \delta\right\},$$

where δ denotes the minimum degree in G.

Another problem related to PAP is the Target Set Selection Problem [Chen 2009] in which it is required that all vertices be spreaders at the end of the propagation. Additionally, in [Cordasco et al. 2018], the Perfect Evangelizing Set Problem (PESP) is introduced, where a second threshold function controls the awareness of vertices. Clearly, PAP is a special case of PESP since a single neighboring spreader suffices for a vertex to become aware.

For more details on the origin and bibliography of PAP and related problems, see Chapters 1 and 3 of [Pereira 2021].

3. Contributions

The research work presented in the Dissertation [Pereira 2021] consists of a computational study of PAP with the objective of designing effective and efficient heuristics for the problem. The main contributions can be summarized as follows:

- We designed *four* heuristics based on the metaheuristic *Greedy Randomized Adaptive Procedure* (GRASP) [Resende and Ribeiro 2010];
- We developed three instance preprocessing techniques for reducing the input size;
- In order to evaluate our algorithms, we generated a benchmark of 840 instances of PAP with characteristics of social networks;
- With the purpose of obtaining the optimal values of the instances of that benchmark and comparing them with the values of solutions produced by the heuristics, we designed the first two known integer programming (IP) formulations for general instances of PAP;
- We performed an extensive statistical analysis of the performance of our heuristics and ranked them according to their success in solving all the instances of the *benchmark*;
- We applied our best heuristic to 17 instances from the literature that represent real social networks and verified that, for these instances, our best algorithm outperforms the CGR [Cordasco et al. 2019], which is the only previously known heuristic designed for PAP.

In the following subsections, we address the main contributions in more detail.

3.1. Benchmark instances

The literature on spreading of information in social networks shows that real social networks tend to exhibit two main characteristics [Barabási and Albert 1999]. The first is the growth in the number of network members. The second is that when a new individual is incorporated into the network, it tends to form connections with existing individuals who already have a higher number of connections. One of the main algorithms used to generate graphs that represent social networks with these attributes is the Barábasi-Albert (BA) method [Barabási and Albert 1999], which produces graphs that capture the scale-free distribution of real-world social networks.

We introduced a procedure that is a slight modification of the BA algorithm, and employed it as follows. For each $n \in \{10, 15, 20, \ldots, 95\} \cup \{100, 200, \ldots, 1000\}$, we generated 30 undirected graphs with n vertices. For a fixed n, the graphs with n vertices vary in number of edges m and, consequently, vary also in density, which is calculated by $2m/(n(n-1))$. The densities range from the lowest possible value, whose graph corresponds to a tree, to values close to 0.5.

For each generated graph G, we built an instance (G, t) with $t(v) = \lceil 0.5 \cdot d(v) \rceil$, where t is known as the *majority threshold function* [Cordasco et al. 2019]. An implementation of the modified BA algorithm, as well as the 840 produced instances and the best known solution for each instance are available in [Pereira et al. 2020b]. For details on the benchmark generation process, see Section 5.1 of the Dissertation.

3.2. Preprocessing

We also introduced preprocessing techniques for reducing the input size of PAP instances. The first one amounts simply to separately solve an instance (G, t) of PAP for each connected component of the input graph $G = (V, E)$. It is easy to see that if $S_1, S_2, \ldots, S_c$ are perfect seed sets for the c connected components of G, respectively, then $\bigcup_{i=1}^{c} S_i$ is a perfect seed set for G.

Secondly, let $u, v \in V$ such that $t(u) = t(v) = 1$ and $\{u, v\} \in E$. If u becomes a spreader, then so does v. Hence, we can contract the edge $\{u, v\}$ in order to collapse u and v to a new vertex w with $t(w) = 1$. Notice that w is an endpoint of all edges that had one endpoint in u or v (except $\{u, v\}$), so multiple edges may occur. Moreover, suppose that a vertex u is an endpoint of several (multiple) edges, but has only one neighbor v and suppose that $t(v) = 1$. In this case, we can collapse u and v into a new vertex w with $t(w) = 1$.

In both cases, it is possible to conclude that if w is a seed in a feasible solution S' of an instance (G', t) that results from preprocessing (G, t), there is a corresponding feasible solution S of (G, t) where w can be replaced by either u or v. The validity of the presented techniques are demonstrated in Section 4.1 of the Dissertation.

3.3. Integer programming models

Let $I = (G, t)$ be an instance of PAP with $|V| = n$. Below, we describe our first IP formulation for PAP, denoted by ROUNDS-IP. Let $\{s_{v,\tau} : v \in V, \tau \in \{0, 1, \ldots, n\}\}$ be a set of binary variables of the ROUNDS-IP model associated with I. Given a solution for the model, we obtain a seed set $S = \{v \in V : s_{v,0} = 1\}$ such that, in a propagation started by S, v is spreader at round τ if and only if $s_{v,\tau} = 1$. The formulations reads:

$$\min z = \sum_{v \in V} s_{v,0} \tag{1}$$

$$s_{v,\tau} - s_{v,\tau-1} \geq 0 \qquad \forall v \in V \; \forall \tau \in [1, n] \tag{2}$$

$$\sum_{u \in N(v)} (s_{u,\tau-1}) - t(v)(s_{v,\tau} - s_{v,0}) \geq 0 \qquad \forall v \in V \; \forall \tau \in [1, n] \tag{3}$$

$$\sum_{u \in N(v)} (s_{u,\tau-1}) - (d(v) - t(v) + 1)s_{v,\tau} \leq t(v) - 1 \qquad \forall v \in V \; \forall \tau \in [1, n] \tag{4}$$

$$s_{v,0} + \sum_{u \in N(v)} s_{u,n-1} \geq 1 \qquad \forall v \in V \tag{5}$$

$$s_{v,\tau} \in \{0, 1\} \qquad \forall v \in V \; \forall \tau \in [0, n] \tag{6}$$

The objective function (1) minimizes the number of spreaders at round 0, that is, the size of the seed set. Constraints (2) guarantee that a vertex v that has become a spreader remains in that state until the end of the propagation. Constraints (3) forbid a non-seed vertex v to be a spreader at round τ, if the number of its neighboring spreaders at round $\tau - 1$ is smaller than its threshold. As a complement to (3), constraints (4) forces v to be a spreader at round τ, if the number of its neighboring spreaders at round $\tau - 1$ is at least $t(v)$.

Since $|V| = n$, it holds that a full propagation takes at most $n + 1$ rounds to end. So, constraints (5) enforce that either v is a seed or has at least one neighboring spreader at round $\tau = n - 1$, which means that v is necessarily aware at round n. Lastly, constraints (6) ensure that all variables are binary. It is easy to see that the ROUNDS-IP formulation has a total of $|V|^2 + |V|$ variables and $\mathcal{O}(|V|^2)$ constraints.

Additionally, we introduced a second IP formulation for PAP, denoted by ARCS-IP. The model is based on the following result. Let $I = (G = (V, E), t)$ be an instance of PAP and $D = (V, A)$ be a directed graph such that $A = \{(u, v) : \{u, v\} \in E\}$. If $S \subseteq V$ is a solution for I, then S is feasible if and only if there is subgraph $D' = (V, A')$ of D which, together with S, fulfills the following properties:

(i) If $v \in S$, then there is no arc ending at v.
(ii) If $(u, v) \in A'$, then either $u \in S$ or there are at least $t(u)$ arcs that end at u.
(iii) For all $v \in V$, either $v \in S$ or there is at least one arc that ends at v.
(iv) D' is acyclic.

In the ARCS-IP model, there are two sets of binary variables, namely $\{s_v : v \in V\}$ and $\{a_{u,v} : (u, v) \in A\}$. Given a solution for the formulation, the subgraph $D' = (V, A')$ of D with $A' = \{(u, v) \in A : a_{u,v} = 1\}$ and the seed set $S = \{v \in V : s_v = 1\}$ satisfy the aforementioned properties.

For each $v \in V$, we denote the *in-neighborhood* of v in D by $N_{\text{in}}(v) = \{u \in V : (u, v) \in A\}$ and the *in-degree* of v in D by $d_{\text{in}}(v) = |N_{\text{in}}(v)|$. Furthermore, Ξ corresponds to the set of all oriented cycles of D. The ARCS-IP formulation reads:

$$\min z = \sum_{v \in V} s_v \tag{7}$$

$$\sum_{u \in N_{\text{in}}(v)} (a_{u,v}) - d_{\text{in}}(v)(1 - s_v) \leq 0 \qquad \forall v \in V \tag{8}$$

$$\sum_{u \in N_{\text{in}}(v)} (a_{u,v}) - t(v)(a_{v,w} - s_v) \geq 0 \qquad \forall (v,w) \in A \tag{9}$$

$$s_v + \sum_{u \in N_{\text{in}}(v)} a_{u,v} \geq 1 \qquad \forall v \in V \tag{10}$$

$$\sum_{(u,v) \in \xi} a_{u,v} \leq |\xi| - 1 \qquad \forall \xi \in \Xi \tag{11}$$

$$s_v, a_{u,v} \in \{0,1\} \qquad \forall v \in V \ \forall u \in V \tag{12}$$

The objective function (7) minimizes the number of vertices in the seed set S. Constraints (8), (9), (10) and (11) guarantee that D and S satisfy properties (i), (ii), (iii) and (iv). Lastly, constraints (12) ensure that all variables are binary. The ARCS-IP formulation has a total of $|V| + 2|E|$ variables and, except for (11), there are $\mathcal{O}(|V| + |E|)$ constraints. Clearly, there is an exponential number of constraints of type (11). For this reason, a separation procedure for (11), presented in Section 4.2.2 of the Dissertation, is employed.

Proofs of correctness and further discussions on both formulations can be seen in Section 4.2 of the Dissertation.

3.4. GRASP-based heuristics

GRASP is a metaheuristic best described as an iterative algorithm comprised of two distinct phases, the *construction phase* and the *local search phase*, which are executed in each iteration.

In the construction phase, a feasible solution is incrementally built following choices that combine greediness and randomization. Consider a set S, initially empty, that represents a solution. Firstly, let a *candidate list* (CL) be the set containing all elements not in S that can be added to S. At each step, we evaluate the benefit of inserting each element from the CL into S. Secondly, let a *restricted candidate list* (RCL) be a subset of the CL containing the elements with the highest benefits. An element from the RCL is randomly selected to be added to S. This process is repeated until S becomes feasible.

In the local search phase, a neighborhood of the constructed solution is explored in the search space of feasible solutions until a local optimum is found.

The "halt condition" of GRASP's main loop can either depend on a fixed number of iterations or on an upper bound for the execution time. Also, the solution returned by the algorithm corresponds to the best solution found over all the iterations.

As mentioned earlier, we developed a total of *four* heuristics based on GRASP, in order to solve PAP. From each of them, we obtain a solution S that constitutes a seed set. In the construction phase, S is built by adding, in each step, a new seed according to certain criteria. These criteria vary among the different heuristics and, for a detailed discussion, the reader is referred to Section 4.3 of the Dissertation. Generally speaking, our main greedy criterion is based on the benefit of inserting a vertex $v \notin S$ into S, which is proportional to the number of neighbors of v that do not become aware along a propagation initiated from S alone. Notice that these vertices would immediately become aware with the addition of v to the seed set.

Moreover, observe that in each step of the construction phase, we need to check the feasibility of the solution S that has been built up to that point. In the case of `PAP`, this validation can be realized by simulating the spreading process starting from S and checking whether all vertices are aware at the end of the propagation. Each simulation can be done in $\mathcal{O}(|V| + |E|)$ time.

Our heuristics were designed so that a single simulation is performed along the entire construction phase (instead of performing a simulation whenever a new seed is added to the seed set). Briefly, we store in memory the final state of the propagation associated with the seed set of the current step and, in the next step, we continue the propagation considering the newly added seed. This strategy proved to be fundamental in terms of efficiency in the construction phase of the algorithms.

Our heuristics are called: *Greedy Randomized* (`GR`), *Weighted Greedy Randomized* (`WGR`), *Random plus Greedy* (`RG`) and *Sampled Greedy* (`SG`). Some of these designations refer to construction strategies already established in the literature [Resende and Ribeiro 2010], where `GR` corresponds to the standard approach presented earlier.

Unlike `GR`, in `WGR`, the probability of an element from RCL being chosen does not follow a uniform distribution and is weighted by a greedy criterion. In `RG`, the main characteristic is that the RCL coincides with the CL, so that, for a given integer p, in the first p steps, we choose a random element from the RCL as a seed and, in the remaining steps, we take the element of the RCL with the greatest benefit. On the other hand, in `SG`, at each step of the construction phase, the RCL corresponds to a random subset of the CL and the element of RCL with the largest benefit is chosen as a new seed.

The local search phase is common to all four heuristics and relies on the fact that if S is a feasible solution, then, for every non-empty set $S' \subset S$, for any $v \in S \setminus S'$, if v becomes a spreader along a propagation started from S', then v can be removed from S without compromising the feasibility of the solution.

All the algorithms reported can be seen in detail in Section 4.3 of the Dissertation. In that same section, proofs of the correctness of the heuristics are presented, as well as their time complexity analyses.

4. Results

From the 840 instances of our benchmark, we attained optimal solutions for 281 using the integer programming solver `CPLEX` with the `IP` formulations previously presented. We verified that all the heuristics reached optimal solutions for at least 91.81% of these 281 instances. Furthermore, even when the heuristics did not produce optimal solutions, the values obtained were very close to the optimal value. These are evidence that our algorithms are able of finding very good solutions.

For instances where the optimal value is unknown, the algorithms GR and SG were superior to the other two heuristics, attaining solutions with a value equal to the best known one for 95.35% and 97.49% of the instances, respectively.

To confirm that GR and SG are the most effective heuristics, we conducted a rigorous statistical analysis of their performance on our benchmark and demonstrated that the quality of the solutions produced by GR and SG were significantly superior to the others, notwithstanding that these two were statistically indistinguishable from one another. Additionally, we applied a speed and robustness test restricted to these two heuristics and, even though both presented satisfactory results, the SG heuristic proved to be superior to GR on these points.

Lastly, we employed the SG heuristic to solve 17 instances reported in the literature comprised of graphs that represent *real social networks*. We verified that for 15 of these instances, SG outperformed CGR – the state-of-art heuristic for PAP up until then, and for the two remaining instances, SG obtained solutions with same value as CGR.

Part of the work developed during this research effort gave rise to a paper published in the proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium – LAGOS 2021 [Pereira et al. 2021]. Also, preliminary results of the research were presented in the form of a poster [Pereira et al. 2020a] in the 14th Latin American Theoretical Informatics Symposium – LATIN 2020.

Since 2021, Felipe de C. Pereira has been working towards his PhD in Computer Science at the Institute of Computing at the University of Campinas. His current research consists of an extension of wider scope and depth of the work carried out during his master's degree studies. In particular, versions of PAP and related problems that encompass more realistic characteristics of social networks and of the spreading process are being investigated. Of especial interest are the consideration of directed graphs to represent non-reciprocal relations that occur in networks such as Twitter and Instagram, as well as arc weights to typify different levels of influence in the relationship between individuals.

References

Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web*, pages 519–528.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

Chen, N. (2009). On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415.

Chen, W., Lakshmanan, L., and Castillo, C. (2013). Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5:1–177.

Cordasco, G., Gargano, L., Rescigno, A., and Vaccaro, U. (2018). Evangelism in social networks: Algorithms and complexity. *Networks*, 71(4):346–357.

Cordasco, G., Gargano, L., and Rescigno, A. A. (2019). Active influence spreading in social networks. *Theoretical Computer Science*, 764:15–29.

Grandoni, F. (2006). A note on the complexity of minimum dominating set. *Journal of Discrete Algorithms*, 4(2):209–214.

Hochbaum, D. S. (1997). *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., USA.

Pereira, F. C. (2021). A computational study of the Perfect Awareness Problem. Master's thesis, University of Campinas, Brasil.

Pereira, F. C., de Rezende, P. J., and de Souza, C. C. (2020a). Exact Algorithms and Heuristics for the Perfect Awareness Problem. In *Posters of the LATIN Conference 2020*, São Paulo. Abstract and poster available on `https://latin2020.ime.usp.br/posters/resumo05.pdf` and `https://latin2020.ime.usp.br/posters/poster05.pdf`.

Pereira, F. C., de Rezende, P. J., and de Souza, C. C. (2021). Effective heuristics for the perfect awareness problem. *Procedia Computer Science*, 195:489–498. Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium.

Pereira, F. C., de Souza, C. C., and de Rezende, P. J. (2020b). The Perfect Awareness Problem - Benchmark Instances. Available on `www.ic.unicamp.br/~cid/Problem-instances/Perfect-Awareness-Problem/`.

Resende, M. G. and Ribeiro, C. C. (2010). Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In *Handbook of Metaheuristics*, pages 283–319. Springer, USA.