Benders decomposition for competitive influence maximization in (social) networks

Michael Kahr[1], Markus Leitner[2], Mario Ruthmair[1] and Markus Sinnl[3]

[1]ISOR, University of Vienna, Austria. {m.kahr|mario.ruthmair}@univie.ac.at
[2]Vrije Universiteit Amsterdam, Departnment of Supply Chain Analytics, Amsterdam, Netherlands. m.leitner@vu.nl
[3]Johannes Kepler University Linz, Institute of Production and Logistics Management, Linz, Austria, markus.sinnl@jku.at

January 15, 2020

Abstract

Online social networks have become crucial to propagate information. Prominent use cases include marketing campaigns for products or political candidates in which maximizing the expected number of reached individuals is a common objective. The latter can be achieved by incentivizing an appropriately selected seed set of influencers that trigger an influence cascade with expected maximum impact. In real-world settings competing influence spreads need to be considered frequently. These may, e.g., stem from marketing activities for a substitute product of a different company or bad actors that spread (mis-)information about a political candidate. This article focuses on competitive settings in which the seed individuals of one entity is already known. Another entity wants to choose its seed set of individuals that triggers an influence cascade of maximum impact. The propagation process is modeled by a variant of the probabilistic independent cascade model. An algorithmic framework based on a Benders decomposition is developed that also employs preprocessing and initial heuristics. This framework is used within a sample average approximation scheme that allows to approximate the exact objective function value. The algorithms are tested on real-world instances from the literature and newly-obtained ones from Twitter. A computational study reports on the algorithms' performance next to providing further insights. The latter are based on analyzing expected losses that are caused by competition, the gain from solving subproblems to optimality using the Benders decomposition based algorithm, and the influence of different seed set choices of the first entity.

Keywords: competitive influence maximization, social networks, integer linear programming, Benders decomposition

1 Introduction

Online social networks have evolved to important communication channels over the last decades and are used by millions of people. The digital records of such networks allow to reveal relationship structures between individuals and to observe or predict user attributes and interests from their individual behavior [23]. This knowledge is of notable interest for firms and other entities that use social networks for marketing campaigns. Such campaigns can benefit from the user's network values [12] and word-of-mouth effects and therefore be considerably more effective than traditional approaches [37]. Also *social influence*, which may cause individuals to adjust their opinions based on the opinions of their peers, can be supportive to stimulate certain consumer decisions [24] or even to sway political election outcomes [4]. Social networks may also be used by bad actors, who spread misinformation and so-called *fake news*. The spread of such misinformation has been identified as among the world's top global risks by the World Economic Forum [14], with economic cost caused by it estimated to be at least $78 billion per year [30].

The decentralized viral spread of information in social networks such as news, opinions or advertisements is also referred to as *influence propagation*. Influence cascades are triggered by the injection of new information which is performed, for instance, by network nodes such as (paid) *influencers* or, more generally, individuals that are incentivized to do so. The influencer marketing industry is estimated to be worth up to $16 billion in 2020 [36].

A common goal is to maximize the number of network nodes that are covered by an influence cascade, e.g., to increase awareness or expected sales. Variants of the underlying *influence maximization problem* (IMP) introduced by Kempe et al. [19] have recently received significant attention from the computer science and operations research communities. IMPs aim to identify a *seed set* of initially influenced nodes that trigger an influence cascade of maximum impact. Seed set members are assumed to be influenced by external means such as discounts or other monetary incentives. The size of a seed set is therefore typically constrained by cardinality. Different propagation models formalizing the influence propagation process have been considered, see, e.g., [17] for a comprehensive survey. The majority of articles consider, however, either the probabilistic *independent cascade model* or the deterministic *linear threshold model*, see, e.g., [19, 20]. Kempe et al. [19] showed that the IMP is NP-hard under each of these two propagation models and that its objective function is monotone and submodular. The latter properties triggered the development of approximation algorithms for the IMP (see, e.g., Chen [7] and the references therein) which are based on the seminal work of Nemhauser et al. [29]. Topological metrics such as betweenness centrality of the underlying social network have been used to develop heuristic methods without performance guarantee [28, 38]. Only recently, variants of the IMP have also been tackled by exact solution methods based on *integer linear programming* (ILP), see, e.g., [13, 16, 34, 39].

All so-far mentioned variants of the IMP neglect the existence of competition by assuming a single, information propagating entity. This limits their applicability since competing entities often exist in practice. These may refer, for instance, to companies that promote substitute products or opposing political parties that try to convince indecisive voters. The influence propagation of competing entities is sometimes modeled implicitly by ubiquitous discount factors that affect the opinions of influenced individuals during the propagation process [35]. Frequent assumptions in the sparse literature related to IMPs with competition include the existence of two competing entities (to which we refer to as *leader* and *follower*) and that the seed set of the leader is known in advance. Thus, the follower aims to make an optimal decision as a reaction to the leaders choice. Objectives that have been considered include maximizing the influence of the follower [3, 27] or minimizing the one of the leader [6, 39]. The latter variant is sometimes referred to as rumor blocking. Also the perspective of social network operators that may try to maximize the overall welfare of their sponsoring customers has been considered [5]. Other problem variants focus on time-critical issues (e.g., protection against leakage) during the propagation process [31].

No consistent naming conventions and standard benchmark problems have been established for IMPs with competition yet. One common requirement is, however, to correspondingly extend a particular influence propagation model and in particular include certain tie-breaking rules that govern the outcome of a situation in which a network individual is simultaneously influenced by different entities; see Chen et al. [9] for an overview of so-far considered tie-breaking rules. Most of the articles treating IMPs with competition follow the approach of Kempe et al. [19] and develop greedy $1-1/e$ approximation algorithms to solve the respective problem variants, see, e.g., [6, 27, 39]. To our knowledge, Keskin and Güler [21] who develop a time-indexed formulation based on the linear threshold model are the only ones who consider ILP methods for IMPs with competition explicitly.

1.1 Scientific contribution

As discussed above, deriving ILP formulations and developing exact solution algorithms based on them has been almost neglected in the literature related to IMPs that consider competition. In this paper, we study the competitive influence maximization problem (CIMP) based on the competitive independent cascade model (CIC-M) and make the following contributions:

- We show that the related rumor blocking problem can be solved as a slightly modified CIMP (Section 2).
- We propose a new time-indexed ILP formulation for the CIMP (Section 3.1).
- We show that the CIMP can be seen as a stochastic variant of the maximum covering location problem and propose a Benders decomposition based solution algorithm following a recently proposed, highly successful approach for the latter problem (Sections 3.2 and 3.3).
- We develop and implement an algorithmic framework based on this Benders decomposition algorithm that also employs preprocessing and initial heuristics. To approximate the exact objective function value, our algorithms are also embedded into a sample average approximation scheme (Section 4).
- We test our algorithmic framework on instances known from the literature and also propose a new set of benchmark instances based on data of the social network Twitter. Besides evaluating the performance of our algorithms, we particularly focus on managerial insights such as the expected losses caused by competition (Section 5).

2 Problem definition

The *competitive influence maximization problem* (CIMP) considered in this article is defined on a simple directed graph $G = (V, A)$ modeling a social network. While the network participants correspond to node set V, their relations are represented by arc set $A \subseteq V \times V$. A value $p_{ij} \in [0, 1]$ is associated with each arc $(i, j) \in A$ that represents the probability that an *active* node i successfully *activates* node j. The term *active* indicates that an individual i adopts a product or information and starts exerting influence by sharing that information with its neighbors j along arcs $(i, j) \in A$. The CIMP considers two competing entities denoted by *leader* and *follower* and adopts the frequent assumption that the seed set $L \subset V$ of the leader is known (e.g., as in [3, 6, 27]). Having full information about the leader's decision, the follower aims to identify a seed set $F^* \subseteq V \setminus L$ of cardinality at most $k \in \mathbb{N}$ that maximizes the expected number of activated nodes triggered by this seed set, i.e.,

$$F^* = \text{argmax}_{F \subseteq V \setminus L, |F| \leq k}\ \sigma(L, F).$$

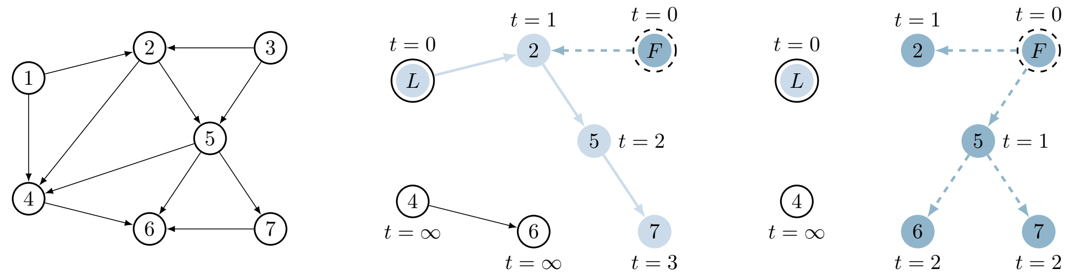
Here, $\sigma(L, F) \in \mathbb{R}^+$ denotes the expected number of nodes activated by follower seed set F when assuming a leader seed set L under the following *competitive independent cascade model*.

Competitive independent cascade model (CIC-M) Extending the independent cascade model, the propagation process of the CIC-M occurs in a discrete time setting in which only nodes in seed sets $L \subset V$ and $F \subseteq V \setminus L$ are active at time zero. A node i activated at a certain time step t immediately tries to activate all its neighbors j along arcs $(i, j) \in A$. If an attempt of activating node j succeeds (which happens with probability p_{ij}), node j gets active at time $t + 1$ and consequently tries to influence its neighbors. Each node can be activated either by an influence cascade that originates from L or by an influence cascade triggered by F. Activated nodes remain active and cannot be additionally activated by the other entity at a later point in time. We break ties in favor of the leader in case of simultaneous activations of a node: If an activation attempt is successful both from the leader and the follower at the same time t, the node is assumed to be activated by the leader. Notice that an active node can try to activate each neighbor only once during the propagation process, and that each activation attempt is assumed to happen independently from all other activation attempts. The propagation process stops if no node is activated in some time step.

In the following, we will use a discrete set of scenarios Ω instead of explicitly considering activation probabilities p_{ij}, $(i, j) \in A$. Each scenario $\omega \in \Omega$ is represented by a so-called *live-arc graph* (V, A^ω), $A^\omega \subseteq A$. Arc set A^ω contains all arcs (i, j) for which an attempt of i to activate j succeeds in scenario ω. Thus, we have $|\Omega| = 2^{|A|}$ to include all possible outcomes and a specific scenario $\omega \in \Omega$ occurs with probability $p^\omega = \Pi_{(i,j) \in A^\omega} p_{ij} \cdot \Pi_{(i,j) \in A \setminus A^\omega} (1 - p_{ij})$. We denote the set of nodes activated by the follower seed set F depending on the leader seed set L in scenario $\omega \in \Omega$ by $\rho^\omega(L, F)$. An illustrative instance graph of the CIMP is shown in Figure 1a. For simplicity we omit introducing precise influence probabilities and instead discuss two exemplary live-arc graphs $G^1 = (V, A^1)$ and $G^2 = (V, A^2)$ together with corresponding propagation processes based on leader seed set $L = \{1\}$ and follower seed set $F = \{3\}$ in Figures 1b and 1c. Influence spread from the leader and the follower is shown using solid and dashed arcs, respectively, and activation times are given next to the nodes. Non-activated nodes are marked with time $t = \infty$. Note that $|\rho^1(L, F)| = 1$ and $|\rho^2(L, F)| = 5$ in our example. Further notice that the tie-breaking rule of the CIC-M is indicated in Figure 1b, in which a tie at node 2 is broken in favor of the leader at $t = 1$.

A benefit of considering discrete scenarios is that the propagation process is deterministic for each of them. Moreover, for a fixed scenario $\omega \in \Omega$ and given seed sets L and F, breadth-first-search (BFS) can be used to efficiently calculate nodes activated by the leader and follower, respectively, together with the corresponding activation times. Observation 1 which is based on Budak et al. [6] draws conclusions from the particular case when $F = \emptyset$.

Observation 1. Let $d^\omega(L, i)$ denote the time when node i is activated by the leader in scenario $\omega \in \Omega$ if $F = \emptyset$, i.e., the length of a shortest path (measured in the number of arcs) in G^ω from node set L to node i. Assuming that $d^\omega(L, i) = \infty$ if node $i \in V$ is not reachable by the leader,



(a) Instance graph G (b) Live-arc graph G^1 (c) Live-arc graph G^2

Figure 1: Instance graph G and live-arc graphs G^1 and G^2, which also show the corresponding influence spreads of the leader (solid, light blue arcs) and the follower (dashed, blue arcs) for seed sets $L = \{1\}$ and $F = \{3\}$, respectively.

the definition of CIC-M implies that the follower can only activate node i in scenario ω if there is at least one seed node of the follower for which the length of a shortest path to i is smaller than $d^\omega(L, i)$.

At first place, this observation enables us to neglect all decision variables that model the leader's propagation in the ILP formulations presented in Section 3. Additionally, it is the basis for observing that the CIMP can be modeled as a classical IMP with marginal modifications. Finally, we observe that the ability of efficiently pre-computing all nodes that are reachable by the leader if $F = \emptyset$ enables a transformation of the CIMP to the influence blocking maximization problem [18] (or rumor blocking problem [6]). Here, we simply adapt the objective function such that only those nodes appear therein. Clearly, the objective value of the CIMP then corresponds to the expected number of nodes the follower prevents from getting activated by the leader.

Observe that the leader might represent an arbitrary number $n \geq 1$ of competitors whose seed sets $L_i, i = 1, \dots, n$, are known a priori and which can be accumulated in a single seed set $L = L_1 \cup \dots \cup L_n$. We also note that only minor modifications would be necessary for applying the algorithmic framework developed in the following to more general and likely more realistic versions of the CIMP. In particular, we refer to variants in which the seed set is constrained by budget rather than cardinality and with node-dependent costs for including an entity in a seed set. Likewise, different values for the objective function coefficients of nodes (e.g., depending on the "estimated relevance" of an entity) could be considered. To stay in line with the related literature we do refrain from explicitly considering these aspects in the following. The flexibility to integrate these aspects can, however, be seen as an additional advantage of the methods proposed in this article compared to existing approximation algorithms whose tight approximation ratios would not persist in more complex and realistic problem settings.

We conclude this section, by observing that it is also easy to see that the IMP is a special case of the CIMP when $L = \emptyset$. Thus, the CIMP is NP-hard [19] and the evaluation of the function $\sigma(L, F)$ under the CIC-M is #P-hard [8]. Furthermore, function $\sigma(L, F)$ is monotone and submodular under the CIC-M so that $\sigma(L, F \cup \{j\}) - \sigma(L, F) \geq \sigma(L, F' \cup \{j\}) - \sigma(L, F')$ holds for any $F \subseteq F' \subseteq V \setminus L$ and $j \in V \setminus (L \cup F')$, cf., [3].

3 Stochastic integer linear programming formulations

In this section, we propose three stochastic ILP formulations for the CIMP. Starting from a stochastic time-indexed formulation we establish a link to a stochastic variant of the maximal covering location problem for which an exact reformulation based on Benders decomposition is presented.

3.1 Stochastic time-indexed model

Formulation (1) proposed in this subsection is based on time-indexed variables that indicate if and at what time a node is activated by the follower. Similar time-indexed variables for the leader's propagation are avoided due to Observation 1 which allows to pre-compute all time points $T_i^\omega = \{t \in \mathbb{N}_0 : t < d^\omega(L,i)\}$ at which the follower can activate node $i \in V$ in scenario $\omega \in \Omega$. Time-indexed binary variables y_{it}^ω, $\forall t \in T_i^\omega, i \in V \setminus L, \omega \in \Omega$, are used to indicate whether or not node i gets activated at time t in scenario ω. Besides, formulation (1) uses binary variables z_i, $\forall i \in V \setminus L$, that indicate whether or not node i is part of the follower's seed set F.

$$
\begin{aligned}
\text{(TI)} \quad \max \quad & \sum_{\omega \in \Omega} p^\omega \sum_{i \in V \setminus L} \sum_{t \in T_i^\omega} y_{it}^\omega && && \text{(1a)} \\
\text{s.t.} \quad & \sum_{i \in V \setminus L} z_i \le k && && \text{(1b)} \\
& z_i = y_{i0}^\omega && \forall i \in V \setminus L, \forall \omega \in \Omega, && \text{(1c)} \\
& \sum_{\substack{(i,j) \in A^\omega, \\ t-1 \in T_i^\omega}} y_{i(t-1)}^\omega \ge y_{jt}^\omega && \forall j \in V \setminus L, \forall t \in T_j^\omega \setminus \{0\}, \forall \omega \in \Omega, && \text{(1d)} \\
& \sum_{t \in T_i^\omega} y_{it}^\omega \le 1 && \forall i \in V \setminus L, \forall \omega \in \Omega, && \text{(1e)} \\
& y_{it}^\omega \in \{0,1\} && \forall i \in V \setminus L, \forall t \in T_i^\omega, \forall \omega \in \Omega, && \\
& z_i \in \{0,1\} && \forall i \in V \setminus L. &&
\end{aligned}
$$

The objective function (1a) maximizes the expected number of nodes activated by the follower's seed set whose cardinality is constrained by a given positive integer k in (1b). Constraints (1c) link the seed set variables to the follower's activation variables at time zero for each scenario. The follower's propagation for each scenario along a path in G^ω is modeled in constraints (1d), i.e., node j can only be activated in time step t if at least one incoming neighbor has been activated at time $t-1$. Equations (1e) ensure that each node can be activated at most once. Finally, notice that the binary requirements on the activation variables can be relaxed to $y_{it}^\omega \in [0,1]$, $\forall i \in V \setminus L, t \in T_i^\omega, \omega \in \Omega$, since constraints (1c)-(1e) together with the objective function ensure integral values for these variables.

3.2 Stochastic maximal covering location model

The latest activation times per node resulting from Observation 1 can be used to propose a formulation that avoids the use of time-indexed variables. Following Güney et al. [16], we define the *reachability set* R_i^ω of node $i \in V$ in scenario $\omega \in \Omega$ as the set of nodes for which a path to i consisting of less than $d^\omega(L,i)$ arcs exists in G^ω. Thus, if a node $j \in R_i^\omega$ is contained in the followers seed set, node i is activated by the follower in scenario ω. This point of view enables the formulation of the CIMP as an instance of the maximum covering location problem (MCLP) [10]. This relation has also been used to model the IMP, e.g., in [16, 26]. The main difference when considering competing entities is that the reachability sets need to be adapted according to the (given) leader's propagation. To keep this paper self-contained we will repeat the formulation of the MCLP, and briefly discuss an existing reformulation based on Benders decomposition [2, 11, 16].

Let variables y_i^ω and z_i indicate whether or not node i is activated by the follower in scenario ω, and whether or not $i \in F$, respectively. Then, the CIMP can be formulated by

$$
\begin{aligned}
\text{(COV)} \quad \max \quad & \sum_{\omega \in \Omega} p^\omega \sum_{i \in V \setminus L} y_i^\omega && && \text{(2a)}\\
\text{s.t.} \quad & \sum_{i \in V \setminus L} z_i \le k && && \text{(2b)}\\
& y_i^\omega \le 1 && (\alpha_i^\omega) && \forall i \in V \setminus L, \forall \omega \in \Omega, \quad \text{(2c)}\\
& \sum_{j \in R_i^\omega} z_j \ge y_i^\omega && (\beta_i^\omega) && \forall i \in V \setminus L, \forall \omega \in \Omega, \quad \text{(2d)}\\
& y_i^\omega \ge 0 && && \forall i \in V \setminus L, \forall \omega \in \Omega,\\
& z_i \in \{0,1\} && && \forall i \in V \setminus L.
\end{aligned}
$$

The objective function (2a) maximizes the expected number of the nodes triggered by the follower's seed set whose cardinality is constrained in (2b). Constraints (2d) ensure that the follower can only activate a node by including at least one node from the corresponding reachability set in the seed set. Variables y_i^ω, $\forall i \in V$, $\omega \in \Omega$, again attain integral values in optimal solutions, which is enforced by the objective function together with constraints (2d).

3.3 Reformulation based on Benders decomposition

Formulation COV is appealing for Benders decomposition [2] since it reduces to a linear program (LP) for fixed values $\bar{\mathbf{z}}$ of variables $\mathbf{z}$. We project out variables y_i^ω, $\forall i \in V$, $\forall \omega \in \Omega$, and instead enforce Benders optimality cuts obtained by solving the dual of the latter LP, see Güney et al. [16] for details. Using μ^ω to denote the contribution of scenario $\omega \in \Omega$ to the objective function, we obtain the exact reformulation

$$
\begin{aligned}
\text{(BEN)} \quad \max \quad & \sum_{\omega \in \Omega} p^\omega \mu^\omega && \text{(3a)}\\
\text{s.t.} \quad & \sum_{i \in V \setminus L} z_i \le k && \text{(3b)}\\
& \mu^\omega \le C^\omega(\bar{\mathbf{z}}) + \sum_{j \in V \setminus L} c_j^\omega(\bar{\mathbf{z}}) z_j && \forall \bar{\mathbf{z}} \in P(\mathbf{z}), \forall \omega \in \Omega, \quad \text{(3c)}\\
& \mu^\omega \ge 0 && \forall \omega \in \Omega,\\
& z_i \in \{0,1\} && \forall i \in V \setminus L.
\end{aligned}
$$

The objective function (3a) maximizes the expected contribution of all scenarios, while constraint (3b) restricts the follower's seed set cardinality by a given value k. Inequalities (3c) correspond to Benders optimality cuts which are defined for each scenario $\omega \in \Omega$ and each extreme point $\bar{\mathbf{z}}$ of the polyhedron $P(\mathbf{z}) := \{\mathbf{z} \in [0,1]^{V\setminus L} : \sum_{i \in V\setminus L} z_i \leq k\}$. For one particular scenario $\omega \in \Omega$ and extreme point $\bar{\mathbf{z}} \in P(\mathbf{z})$, we obtain constant $C^\omega(\bar{\mathbf{z}}) := \sum_{i\in V} \bar{\alpha}_i^\omega$ and coefficients $c_j^\omega(\bar{\mathbf{z}}) := \sum_{i: j \in R_i^\omega} \bar{\beta}_i^\omega, \forall j \in V \setminus L$, where $\bar{\alpha}_i^\omega$ and $\bar{\beta}_i^\omega$ are optimal values for the dual variables associated to constraints (2c) and (2d), respectively. Similar to Güney et al. [16], for integer vectors $\bar{\mathbf{z}}$, constant $C^\omega(\bar{\mathbf{z}})$ corresponds to the number of activated nodes $|\rho^\omega(L,F)|$ triggered by follower seed set $F = \{i \in V \setminus L : \bar{z}_i = 1\}$ in scenario $\omega \in \Omega$. Coefficient $c_j^\omega(\bar{\mathbf{z}})$ can be interpreted as the marginal gain of adding node $j \in V \setminus L$ to seed set F in scenario ω, i.e., $|\rho^\omega(L, F \cup \{j\})| - |\rho^\omega(L,F)|$.

4 Algorithmic framework

To solve the CIMP, we focus on the Benders reformulation BEN used within a branch-and-cut framework. We do not report results obtained from the other formulations TI and COV because in preliminary tests it turned out that they are not applicable to our large-sized instance set. Since handling the exponential number of scenarios $|\Omega| = 2^{|A|}$ is computationally not tractable for reasonable graph sizes, we approximate the objective function by *sample average approximation* (SAA) [22] in which only a randomly drawn subset $\Omega' \subset \Omega$ (based on Monte-Carlo sampling) is considered in each SAA iteration. Consequently, the objective function (3a) is replaced by $\frac{1}{|\Omega'|}\sum_{\omega\in\Omega'} \mu^\omega =: \hat{\sigma}_{\Omega'}(L,F)$, where $\hat{}$ indicates an estimator. The solutions, i.e., the seed sets, obtained in each SAA iteration are evaluated a posteriori on a much larger set of scenarios Ω'' with $|\Omega''| \gg |\Omega'|$, and the solution which performs best on set Ω'' is selected as estimated optimal seed set $\hat{F}^*$. The corresponding approximated expected objective value is denoted by $\hat{\sigma}_{\Omega''}(L, \hat{F}^*)$. Sections 4.1-4.3 apply for each SAA iteration.

4.1 Preprocessing

We create the live-arc graphs G^ω by a biased coin flipping procedure based on probabilities p_{ij}, $\forall (i,j) \in A$, for all considered scenarios $\omega \in \Omega'$ in $\mathcal{O}(|A||\Omega'|)$ runtime. We then compute and store the reachability sets R_i^ω for each node $i \in V$ and scenario $\omega \in \Omega'$, requiring $\mathcal{O}(|V|^2|\Omega'|)$ memory. This is done in two steps: (i) We derive the leader's activation times $d^\omega(L,i)$, $\forall i \in V$, $\forall \omega \in \Omega'$, by a BFS starting from seed set L in $\mathcal{O}((|V|+|A|)|\Omega'|)$ runtime. (ii) Then, we run a reverse BFS from each node i and each scenario ω and add to R_i^ω all nodes that are reachable within distance $d^\omega(L,i) - 1$ (in $\mathcal{O}(|V|(|V|+|A|)|\Omega'|)$ runtime). If $d^\omega(L,i) = \infty$, $R_j^\omega \subseteq R_i^\omega$ holds for all $j \in R_i^\omega$. Thus, if the reverse BFS from a node that is not reachable from the leader seed set L encounters a node j for which R_j^ω is already known, we do not need to proceed beyond j but simply add R_j^ω to R_i^ω. Conversely, if $d^\omega(L,i) < \infty$, this does not hold in general.

We apply the following reduction rule introduced in [16]: If some node i is a *singleton* in some scenario ω (i.e., it has no incident arcs in A^ω) the corresponding constraint in (2d) is binding. Let Z^ω denote the set of singletons in live-arc graph G^ω. We remove the associated constraint and replace variable y_i^ω with z_i in the objective function. Due to the removal of these constraints, the corresponding dual variables α_i^ω and β_i^ω do not exist anymore. Hence, for each node $i \in Z^\omega$ the coefficient of z_i in any associated Benders cut (3c) is zero and thus can be ignored in the separation procedure. Note, however, that variable z_i now appears in the objective function with coefficient $\sum_{\omega\in\Omega': i \in Z^\omega} p^\omega$.

4.2 Heuristics

Marginal Gain Heuristic (MAR) We implemented the greedy approximation algorithm proposed by Bharathi et al. [3] which is a straight-forward extension of the greedy algorithm by Kempe et al. [19]. The algorithm starts with an empty seed set $F = \emptyset$ and iteratively adds the node j with the largest marginal gain, i.e., $j = \text{argmax}_{i \in V \setminus (L \cup F)} \hat{\sigma}_{\Omega'}(L, F \cup \{i\}) - \hat{\sigma}_{\Omega'}(L, F)$, until $|F| = k$.

Best Individuals Heuristic (BIN) We also introduce a simple but fast heuristic based on the individual influence $\hat{\sigma}_{\Omega'}(L, \{i\})$ of each node $i \in V \setminus L$. We sort all nodes $i \in V \setminus L$ in non-increasing order of the latter value and build a follower's seed set with the first k nodes in this order. BIN is used as initial heuristic for our branch-and-cut algorithm since preliminary experiments showed no (significant) benefits in terms of total solution time when using MAR instead of BIN. We also exploit this idea in a primal heuristic within the branch-and-bound phase to find new incumbent solutions. Here, we consider current LP solution values $\bar{\mathbf{z}}$ when determining the k nodes in this order. This is realized by sorting the nodes $i \in V \setminus L$ by non-increasing values of $\hat{\sigma}_{\Omega'}(L, \{i\})\bar{z}_i$. Since BIN runs in a fraction of a second, we invoke it for each obtained LP solution in the root node of the branch-and-bound tree and once after adding all violated Benders cuts for each other branch-and-bound node.

4.3 Separation of Benders cuts

Benders cuts (3c) are added dynamically in a cutting plane fashion at each node of the branch-and-bound tree. We initially include Benders cuts

$$\mu^\omega \leq \sum_{i \in V \setminus L} |\rho^\omega(L, \{i\})| z_i, \qquad \forall \omega \in \Omega', \tag{4}$$

corresponding to $\bar{\mathbf{z}} = \mathbf{0}$ to impose initial upper bounds on μ^ω for each scenario $\omega \in \Omega'$. In Algorithm 1 we describe the separation routine of Benders cuts (3c) for a given solution $(\bar{\mathbf{z}}, \bar{\boldsymbol{\mu}})$ which generalizes the method in [16] to the competitive case.

Note that we do not compute the optimal dual values $\bar{\alpha}^\omega$ and $\bar{\beta}^\omega$ explicitly, instead we derive the Benders cut constant and variable coefficients directly from the current primal solution $(\bar{\mathbf{z}}, \bar{\boldsymbol{\mu}})$, see Güney et al. [16] for details. To speed up the computation of the cut constant $C^\omega(\bar{\mathbf{z}})$, we build a partial seed set F' containing all nodes i for which $\bar{z}_i = 1$ and initialize $C^\omega(\bar{\mathbf{z}})$ for a particular scenario $\omega \in \Omega'$ with the number of activated nodes $\rho^\omega(L, F')$ (via a BFS). Note that we exclude singletons Z^ω in all relevant steps in the separation method, see Section 4.1. For each remaining node i, we decide whether to increase constant $C^\omega(\bar{\mathbf{z}})$ or the coefficients $c_j^\omega(\bar{\mathbf{z}})$ for each node j in reachability set R_i^ω, based on value $\sum_{j \in R_i^\omega} \bar{z}_j$. If the corresponding Benders cut is violated, we add it to set $\mathcal{C}$.

5 Computational results

This section includes the description of our benchmark instances, an analysis of the algorithmic performance of the Benders decomposition algorithm, and a discussion of the influence of certain parameters such as the number of scenarios $|\Omega'|$ per SAA iteration. We also investigate the impact of varying the seed set cardinalities in relation to different possibilities of choosing the leaders seed set. Each experiment is based on ten SAA iterations whose solutions, i.e., seed sets, are subsequently

Require: live-arc graphs $G^\omega = (V, A^\omega), \forall \omega \in \Omega'$, solution $(\bar{\mathbf{z}}, \bar{\boldsymbol{\mu}})$
Ensure: set $\mathcal{C}$ containing a maximally violated Benders cut (3c) for each scenario $\omega \in \Omega'$ (if exists)

$F' = \{i \in V \setminus L : \bar{z}_i = 1\}$
$\mathcal{C} = \emptyset$
for $\omega \in \Omega'$ **do**
  $C^\omega(\bar{\mathbf{z}}) = |\rho^\omega(L, F') \setminus Z^\omega|$
  $c_j^\omega(\bar{\mathbf{z}}) = 0, \forall j \in V \setminus L$
  for $i \in V \setminus (\rho^\omega(L, F') \cup Z^\omega)$ **do**
    if $\sum_{j \in R_i^\omega} \bar{z}_j \geq 1$ **then**
      $C^\omega(\bar{\mathbf{z}}) = C^\omega(\bar{\mathbf{z}}) + 1$
    else
      $c_j^\omega(\bar{\mathbf{z}}) = c_j^\omega(\bar{\mathbf{z}}) + 1, \forall j \in R_i^\omega$
    end if
  end for
  if $\bar{\mu}^\omega > C^\omega(\bar{\mathbf{z}}) + \sum_{i \in V \setminus (L \cup Z^\omega)} c_j^\omega(\bar{\mathbf{z}})$ **then**
    $\mathcal{C} = \mathcal{C} \cup \{\mu^\omega \leq C^\omega(\bar{\mathbf{z}}) + \sum_{j \in V \setminus (L \cup Z^\omega)} c_j^\omega(\bar{\mathbf{z}}) z_j\}$
  end if
end for
return $\mathcal{C}$

Algorithm 1: `Separation of Benders cuts`

evaluated on $|\Omega''| = 100\,000$ independently generated scenarios. The solution with the largest objective value on set Ω'' is finally selected. To ease the comparison of different experiments, we decided to use a fixed number of SAA iterations instead of using a dynamic stopping criterion as suggested, e.g., in Kleywegt et al. [22]. The leader seed set is precomputed based on ten SAA iterations using the BEN method (by solving the CIMP with an empty leader seed set) if not stated otherwise and by considering $|\Omega'| = 750$ scenarios. As discussed in Section 5.2 the latter choice offered a reasonable trade-off between solution quality and computational effort.

All algorithms have been implemented in *julia* 1.1.0 and each experiment has been performed on a single core of an Intel Xeon E5-2670v2 machine with 2.5 GHz and 16 GB RAM (except instance `tw-datascience` for which the memory limit has been set to 32 GB RAM). We used IBM CPLEX 12.8 as ILP solver, and set the time limit for a single SAA iteration to two hours.

5.1 Instance description

In the following we describe how our first set of real-world benchmark instances is created. By using the developer interface of Twitter in its freely available standard version it is possible to query information about arbitrary users, tweets, and their relation to each other in the Twitter network [32]. There are, however, restrictions on the amount and type of information obtained in each time slot of 15 minutes. This makes it impossible to re-construct a subgraph of the Twitter network based on friend and follower relations in reasonable time. Thus, we build instance graphs in a different way: We choose some hashtag (e.g., #giftideas) and search for tweets including this hashtag (limited to the last seven days). The authors of these tweets build the initial set of nodes

Table 1: Real-world social networks: number of nodes $|V|$, number of directed arcs $|A|$, average node degree $\overline{\delta}(i)$, expected node degree $\mathbb{E}[\delta(i)] = \sum_{j \in V:(i,j) \in A} p_{ji} + \sum_{j \in V:(j,i) \in A} p_{ij}$.

instance	$\|V\|$	$\|A\|$	$\overline{\delta}(i)$	$\mathbb{E}[\delta(i)]$	description (retrieval date)
`tw-datascience`	25 463	405 932	31.8	1.4	#datascience (2019-04-14)
`tw-giftideas`	6 263	358 705	114.5	6.7	#giftideas (2019-10-10)
`tw-nrw2019`	4 684	224 459	95.8	1.4	#nrw2019 (2019-10-09)
`tw-orms`	758	4 376	11.5	0.9	#orms (2019-10-09)
`tw-valentinesday`	1 782	23 467	26.3	2.3	#valentinesday (2019-10-10)
`tw-vienna`	4 585	20 179	8.8	0.8	#vienna (2019-04-12)
`msg-college`	1 899	20 296	21.4	0.9	messaging network of UC-Irvine [25]
`msg-email-eu`	1 005	24 929	49.6	2.2	email network from eu workers [25]
`soc-advogato`	6 551	47 322	14.4	0.7	free software development community [33]
`soc-anybeat`	12 645	67 053	10.6	0.5	'anti-Facebook' [33]

in our instance. Then, for each user we query all tweets in the year 2019 (up to 3200) and analyze them in detail: We consider each tweet which includes the defined hashtag and check whether it retweets, quotes, replies, or mentions other users. These related users are added to the instance and analyzed in the same way. The procedure stops when no more new users can be added. In particular we obtained instances using the hashtags #datascience, #giftideas, #nrw2019 (national elections 2019 in Austria), #orms (operations research and management science), #valentinesday, and #vienna. These hashtags could be used, for instance, to promote products, events, political messages, or cultural activities.

The influence probability p_{ij} of each arc $(i, j) \in A$ is set to the number of tweets user j retweets original tweets written by user i relative to the total number of tweets written by user i. These approximately 10^6 observations (over all instances) result in an empirical distribution of the influence probability with characteristics that are as follows: minimum = 0.02%, Q_1 = 0.2%, Q_2 = 0.6%, Q_3 = 2.2%, maximum = 100%, where Q_x denotes the x*th* quartile of the distribution.

The latter distribution is used to extend benchmark instances from the literature. We estimate the missing influence probability values by drawing random samples from the aforementioned empirical distribution. Some of those graphs also contain parallel arcs that reflect messages sent at different points in time. Since we are mainly interested in node relationships, we consider only one of those arcs.

A summary of all considered instances is reported in Table 1. In Figure 2 we report the expected node in- and out-degrees, i.e., $\mathbb{E}[\delta^-(i)] = \sum_{(j,i) \in A} p_{ji}$ and $\mathbb{E}[\delta^+(i)] = \sum_{(i,j) \in A} p_{ij}$, respectively.

5.2 Performance analysis and parameter fixing

Figure 3 contains four performance profiles visualizing relative numbers of SAA iterations solved within a certain runtime. These results are obtained from 10 SAA iterations per instance using parameter values $|L| = |F| = 10$ and $|\Omega'| \in \{250, 500, 750, 1\,000\}$. Results for Twitter instances are shown in Figures 3a and 3b per instance and for each considered number of scenarios, respectively. Similarly, Figures 3c and 3d detail results for instances based on graphs from the literature. Figure 3b is based on 60 data points since we consider six different Twitter instances and ten SAA iterations. The other three figures result from 40 data points as we consider four graphs from the

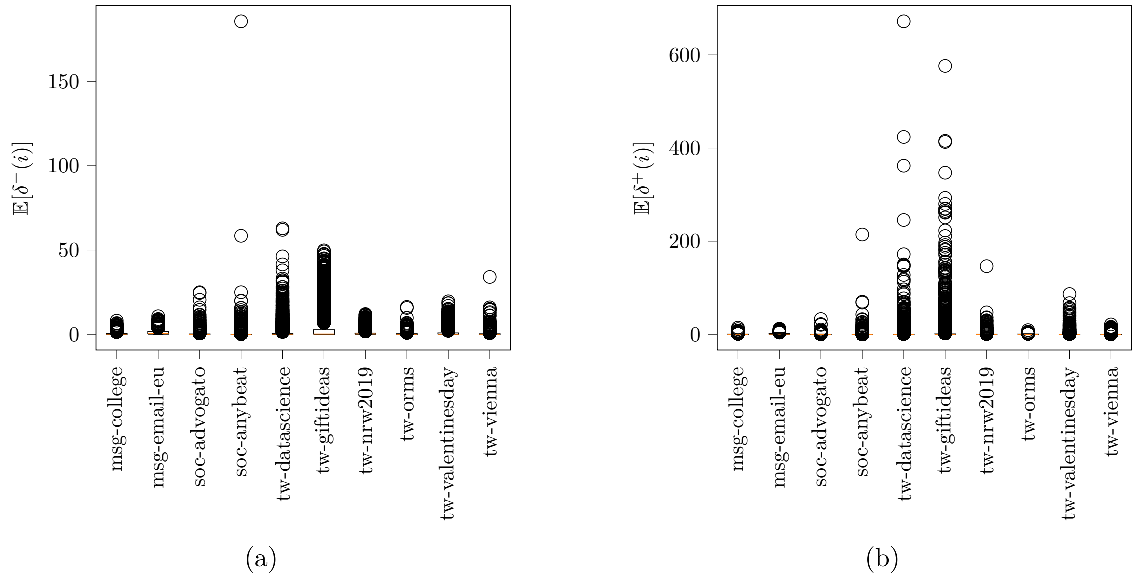


Figure 2: Boxplots of expected node in- and out-degrees for each instance graph.

literature and four different values for the numbers of scenarios.

Comparing Figure 3b and Figure 3d, we observe that the impact of increasing the number of scenarios on the time required to solve a single SAA iteration seems to be smaller for Twitter instances than for those from the literature. We also observe that all SAA iterations of instances different from `tw-datascience` could be solved within the time limit of two hours. On the contrary, almost all SAA iterations of instance `tw-datascience` were not solved to optimality when $|\Omega'| \in \{500, 750, 1\,000\}$. To ease the comparison we will not consider `tw-datascience` in the further presentation of results. We remark, however, that the latter instance graph could be tackled using a relatively small number of scenarios (e.g., 250 or even less) but a larger number of SAA iterations.

Kleywegt et al. [22] suggest to assess the quality of a solution obtained via SAA by the so-called approximation gap which estimates the difference between the objective value obtained from the SAA and the true objective value $\sigma(L, F^*)$. In the following, we will use the *relative approximation gap*

$$\Delta = \frac{\left|\hat{\sigma}_{\Omega''}(L, \hat{F}^*) - \frac{1}{10}\sum_{i=1}^{10} \hat{\sigma}_{\Omega'_i}(L, \hat{F}_i)\right|}{\hat{\sigma}_{\Omega''}(L, \hat{F}^*)}, \tag{5}$$

which relates such an overestimated difference to the best known objective value. Here, $\hat{F}_i$ is the seed set obtained in the ith SAA iteration and $\hat{\sigma}_{\Omega'_i}(L, \hat{F}_i)$ the corresponding objective value.

We analyze the influence of the number of considered scenarios on the relative approximation gaps, the number of SAA iterations needed to identify the best known solution $\hat{F}^*$, and the similarities $|\bigcap_{i=1}^{10} \hat{F}_i|/|F|$ of the solutions obtained in the different SAA iterations. These results are shown in Figure 4 and based on the problem parameters outlined above (excluding `tw-datascience`). Figure 4a reveals a notable reduction of the approximation errors when considering 500 scenarios instead of 250. However, the impact of further increasing the number of considered scenarios to 750 or 1 000 seems to be minor. From Figure 4b we conclude that most of the time, the best solutions are found in the first two SAA iterations, at least when considering 750 or 1 000 scenarios.

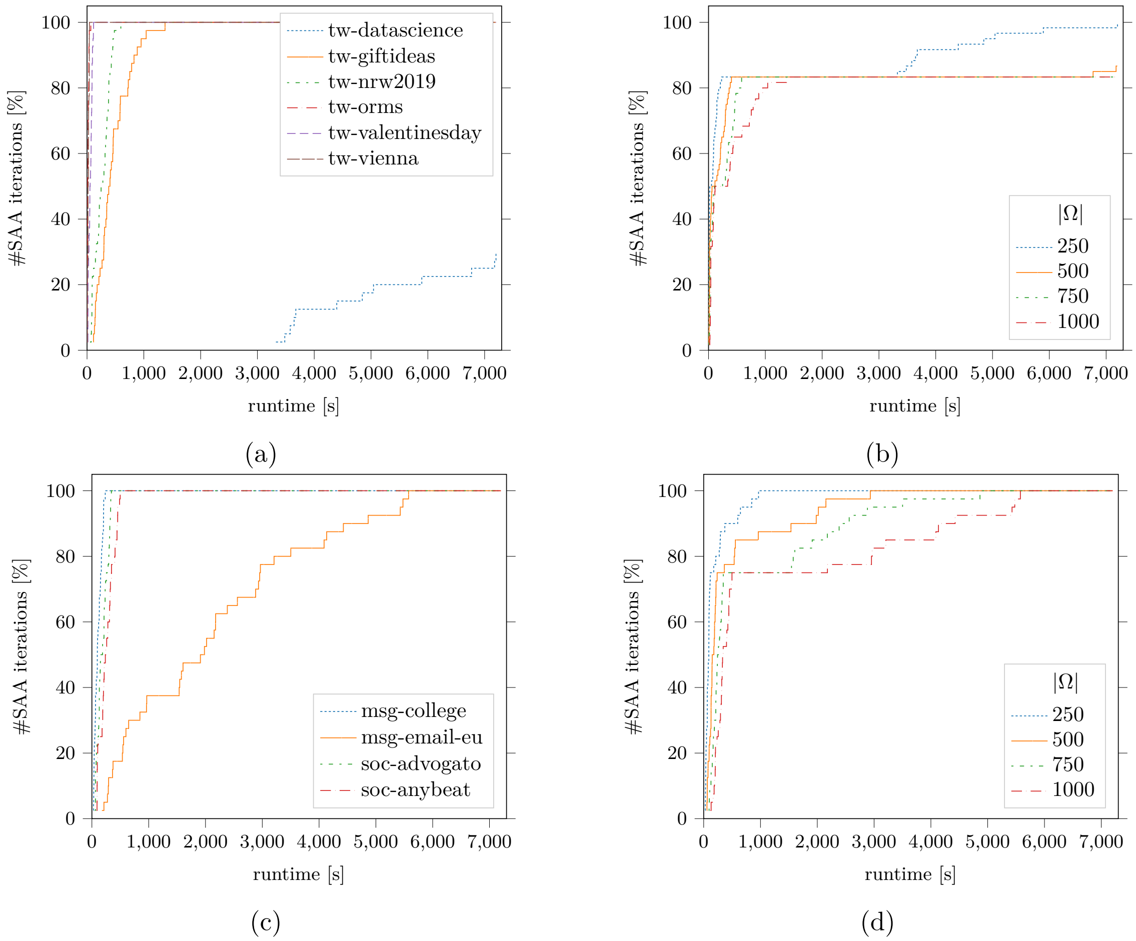
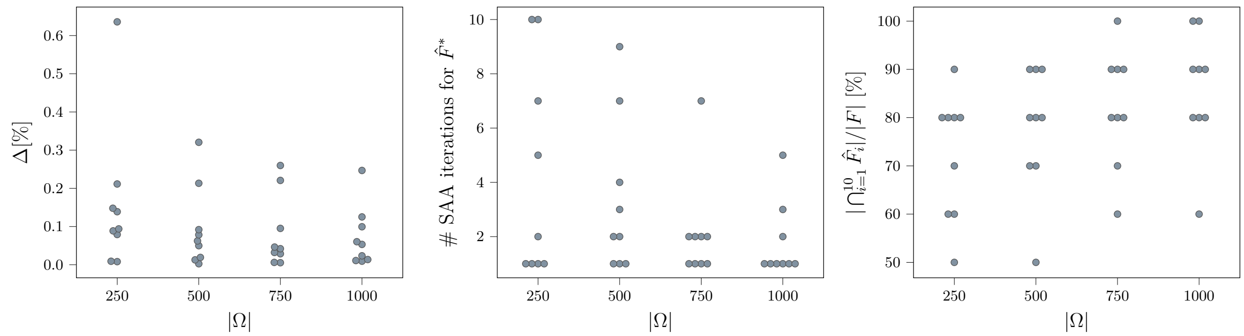


Figure 3: Numbers of SAA iterations solved within a certain runtime in percent (#SAA iterations [%]). Figures (a) and (b) show the performance profiles of the Twitter instances grouped by instance names and number of considered scenarios, respectively. Similarly, figures (c) and (d) show the performance profiles of the instances from the literature grouped by instance names and number of considered scenarios, respectively.

Kleywegt et al. [22] suggested to stop an SAA algorithm (or adjust its parameters) if the solution is not improved for a few consecutive iterations. We conclude that ten SAA iterations seem to be sufficient in our case even if we want to cover the outliers (which in fact arise for instance `tw-valentinesday`). Figure 4c depicts the similarities of the solutions over all SAA iterations. As expected, a larger number of scenarios increases the similarities of the solutions obtained in the different SAA iterations. If $|\Omega'| \in \{750, 1\,000\}$, eight out of ten seed nodes remain the same over all SAA iterations in seven out of nine cases.

We propose that $|\Omega'_i| = 750$ is a reasonable choice for further investigations on the considered instances, since increasing the number of scenarios to $|\Omega'_i| = 1\,000$ results only in marginal improvements in the relative approximation gap, the number of SAA iterations needed to identify $\hat{F}^*$, and the similarity of solutions obtained from each SAA iteration. We further remark that all best known solutions $\hat{F}^*$ when considering 750 scenarios coincide with the corresponding solutions obtained when considering 1000 scenarios, which is not the case if 250 or 500 scenarios are considered.



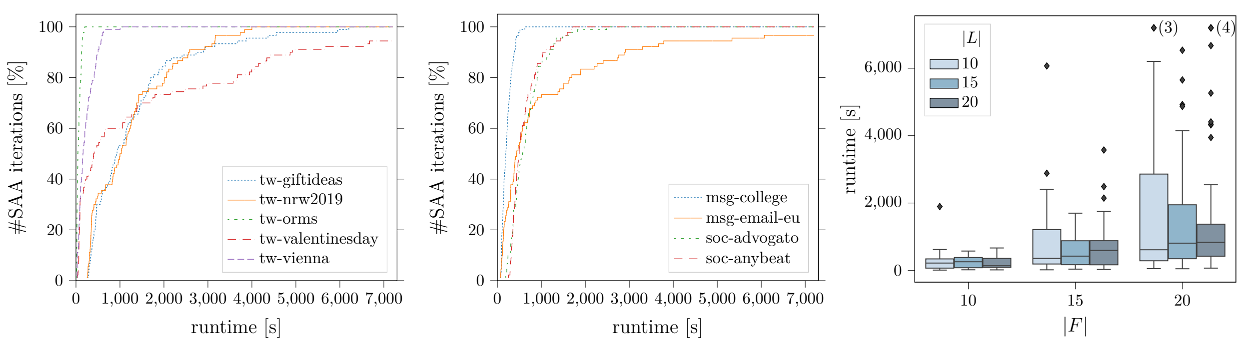
(a) (b) (c)

Figure 4: Impact of different numbers of scenarios on (a) the relative approximation gaps Δ in percent (5), (b) the number of SAA iterations needed to find $\hat{F}^*$, and (c) the relative similarities of the identified seed sets over all SAA iterations. Each data point corresponds to one instance.

After fixing the number of scenarios to $|\Omega'| = 750$, we now focus on the variation of seed set sizes, i.e., $|L| \in \{10, 15, 20\}$ and $|F| \in \{10, 15, 20\}$. The corresponding performance profiles are shown in Figure 5 which indicate that the problem becomes more difficult with increasing size of the follower seed sets F which is consistent with the results in [16]. In contrast, Figure 5c shows that for the considered problem instances the solution time is reduced when enlarging the leader seed sets.

We further remark that in none of the rare cases in which an SAA iteration hits the time limit, the corresponding incumbent seed set $\hat{F}_i$ evaluated on the larger set Ω'' lead to the estimated optimal seed set $\hat{F}^*$. If this would be the case, one would need to be careful with interpreting the approximation gap (see [1] for further details). We further remark that the optimality gap computed by $(UB - \hat{\sigma}_{\Omega_i}(L, \hat{F}_i))/UB$, where UB denotes the best upper bound, is at most 0.8%.

Finally, we remark that detailed results for $|\Omega'| = 750$ are shown in Table 2 in the appendix.



(a) (b) (c)

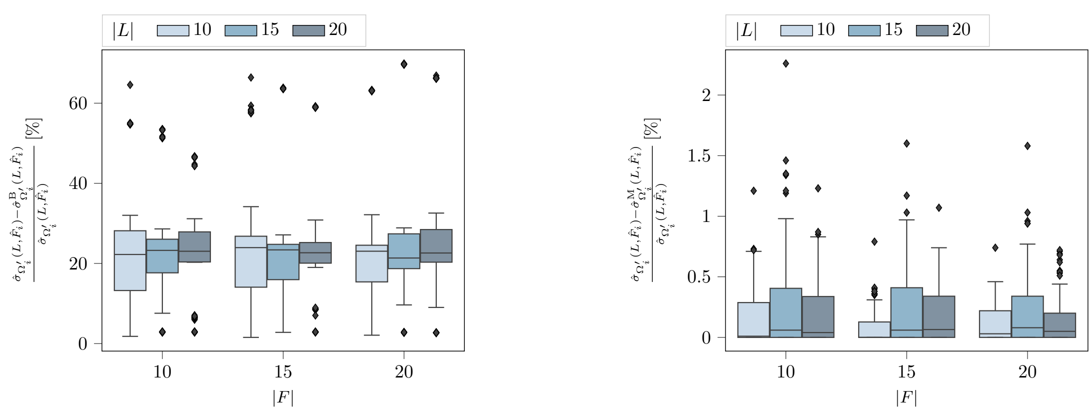
Figure 5: Number of SAA iterations solved within the runtime limit in percent (#SAA iterations [%]) grouped by instances and seed set sizes $|L|$ and $|F|$.

5.3 Exact versus heuristic solution quality

Here, we analyze the improvements one can expect from using the exact approach BEN compared to the considered heuristics MAR and BIN. Figure 6 shows relative improvements of the objective values over those obtained from using MAR (denoted as $\hat{\sigma}^{\mathrm{M}}_{\Omega'_i}(L, \hat{F}_i)$) and BIN (denoted as $\hat{\sigma}^{\mathrm{B}}_{\Omega'_i}(L, \hat{F}_i)$), respectively, for each SAA iteration and instance. We observe that compared to the objective values obtained by BEN the losses are quite large when using heuristic BIN (Figure 6a) but only marginal when considering heuristic MAR (Figure 6b). These observations are consistent to the empirical tests on different instances for the IMP, see, e.g. [15]. These losses are even lower if only the estimated optimal seed sets $\hat{F}^*$ (evaluated on $|\Omega''|$) are compared. The latter gaps are characterized by minimum $= Q_1 = Q_2 = 0\%$, $Q_3 = 0.05\%$, maximum $= 0.7\%$.

However, using solutions obtained from heuristic MAR as initial solution for the branch-and-cut approach does in average not improve the total runtime of BEN. This indicates that good primal bounds are either easily found in the branch-and-bound process or not that important for pruning the search tree.

Overall, the results of this subsection indicate that using MAR instead of an exact approach such as BEN is likely to be sufficient in practical settings, at least for the considered instances. To this end, we stress the fact that BEN can be easily extended to more complex variants of the CIMP, cf. the discussion at the end of Section 2. Given the fact that the tight approximation ratio of MAR would not carry over to such problem variants it seems likely that significantly larger benefits of exact methods over a correspondingly extended variant of MAR can be expected in such settings.



(a) (b)

Figure 6: Relative losses in the objective values when considering heuristics BIN (Figure 6a) and MAR (Figure 6b), respectively, compared to the exact method BEN over all SAA iterations.

5.4 The price of competition

In this section we discuss the relative losses (measured in numbers of activated nodes) the leader and the follower may experience due to their rivalry which can be seen as the *price of competition*. Figures 7a and 7c show the leader's relative losses if the follower propagates from its estimated optimal seed set $\hat{F}^*$ compared to the setting in which there is no follower, i.e.,

$(\hat{\lambda}_{\Omega''}(L, \emptyset) - \hat{\lambda}_{\Omega''}(L, \hat{F}^*))/\hat{\lambda}_{\Omega''}(L, \emptyset)$, where $\hat{\lambda}_{\Omega''}(L, F)$ denotes the expected number of nodes activated by the leader when assuming leader seed set L and follower seed set F. Conversely, Figures 7b and 7d show the follower's relative losses in presence of a leader compared to the case where no leader exists. Denoting the estimated optimal seed set of the follower when no leader exists by $\hat{F}^*_\emptyset$ these relative losses are formally defined as $(\hat{\sigma}_{\Omega''}(\emptyset, \hat{F}^*_\emptyset) - \hat{\sigma}_{\Omega''}(L, \hat{F}^*))/\hat{\sigma}_{\Omega''}(\emptyset, \hat{F}^*_\emptyset)$. The relative losses of the leader seem to be higher than the follower's losses which could be explained by the problem's assumption that the follower knows about the leader's activities in advance but not vice versa. An extreme example is given in Figure 8 which illustrates that the leader can face very large losses due to competition. In fact, the follower can activate most nodes that have been activated by the leader without competition. Figures 7c and 7d in which we group these results by instance show, however, that there are exceptions to these observations. For some Twitter instances the follower's losses are higher which might indicate that the leader's seed set L is a very good choice (see also Section 5.5) which can hardly be compromised by the follower, especially since the leader is the first one to act and also is prioritized in the propagation model in case of ties.

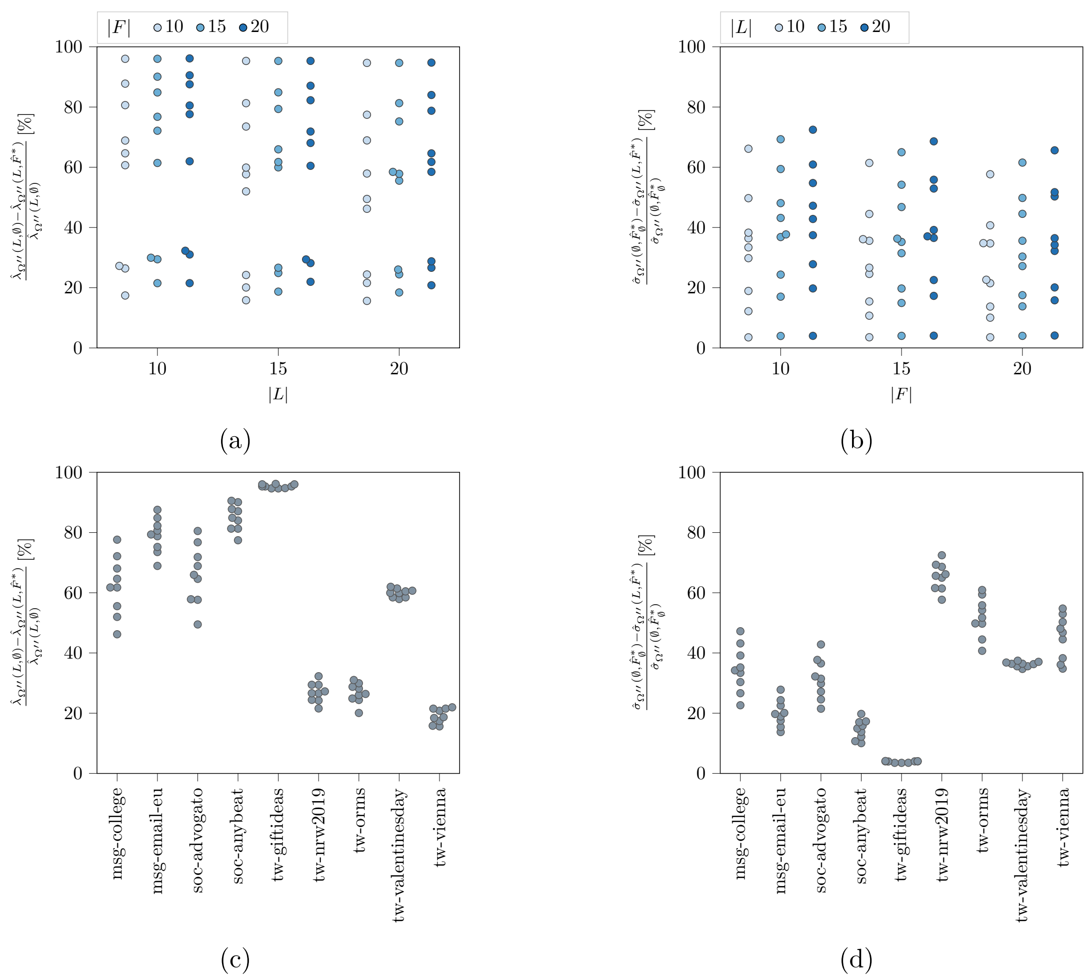


Figure 7: Relative losses of activated nodes for the leader (Figures 7a and 7c) and the follower (Figures 7b and 7d) due to competition, grouped by seed set sizes (Figures 7a and 7b) and instances (Figures 7c and 7d).

5.5 The impact of the leaders choice

Finally, we discuss potential impacts of the leader's strategy to choose a seed set on the expected outcome for the follower. We focus on seed set sizes $|L| = |F| = 20$ and report the relative changes of the follower's (and the leader's) expected number of reached nodes when the leader selects its seed set by heuristic BIN (denoted by L^{B}) instead of the exact method BEN (simply denoted by L). We compute those changes by $\Delta_{\mathrm{F}} = (\hat{\sigma}_{\Omega''}(L, \hat{F}^*) - \hat{\sigma}_{\Omega''}(L^{\mathrm{B}}, \hat{F}^*_{\mathrm{B}}))/\hat{\sigma}_{\Omega''}(L^{\mathrm{B}}, \hat{F}^*_{\mathrm{B}})$ for the follower's outcome, and by $\Delta_{\mathrm{L}} = (\hat{\lambda}_{\Omega''}(L, \hat{F}^*) - \hat{\lambda}_{\Omega''}(L^{\mathrm{B}}, \hat{F}^*_{\mathrm{B}}))/\hat{\lambda}_{\Omega''}(L^{\mathrm{B}}, \hat{F}^*_{\mathrm{B}})$ for the leader's outcome. Here, $\hat{F}_{\mathrm{B}}$ denotes the estimated optimal follower seed set when the leader selects its seed set by heuristic BIN. We do not report results for heuristic MAR, since the correspondingly obtained solutions are quite similar to those obtained with exact method BEN, cf. Section 5.3.

Figure 9 shows the relative changes for each instance where positive values indicate a preference for exact method BEN. We observe, somewhat surprisingly, that the follower improves in most of the cases if the leader chooses a seed set using the exact method BEN. The results for the leader are ambiguous and show no clear trend. A potential explanation for this outcome might be that in estimated optimal leader seed sets there is less overlap between the sets of reached nodes for different seed nodes (which is clearly more efficient in the view of a single entity) than for heuristic seed sets. As a consequence, less overlap makes it potentially easier for the follower to block certain parts of the graph from being activated by the leader.

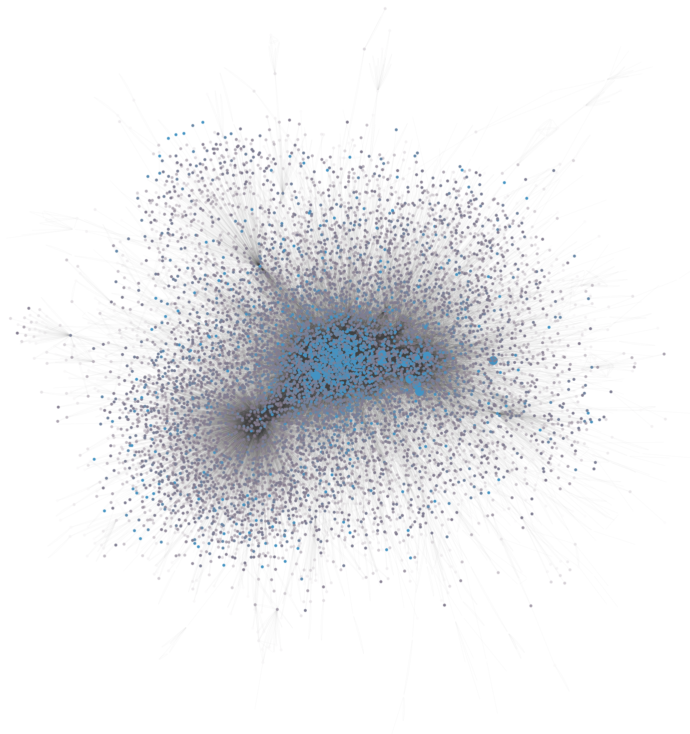
6 Concluding remarks

We studied the competitive influence maximization problem based on a correspondingly adapted independent cascade model. We showed that that this problem can be formulated as an instance of a stochastic maximal covering location problem. Solutions are obtained via two heuristic methods and a Benders decomposition approach based on a set covering formulation, embedded in a sample average approximation framework (SAA). The first part of our extensive computational study focused on determining reasonable framework parameters, particularly, a number of scenarios that lead to an acceptable approximation gap and low numbers of SAA iterations to identify high-quality solutions for the considered instances.

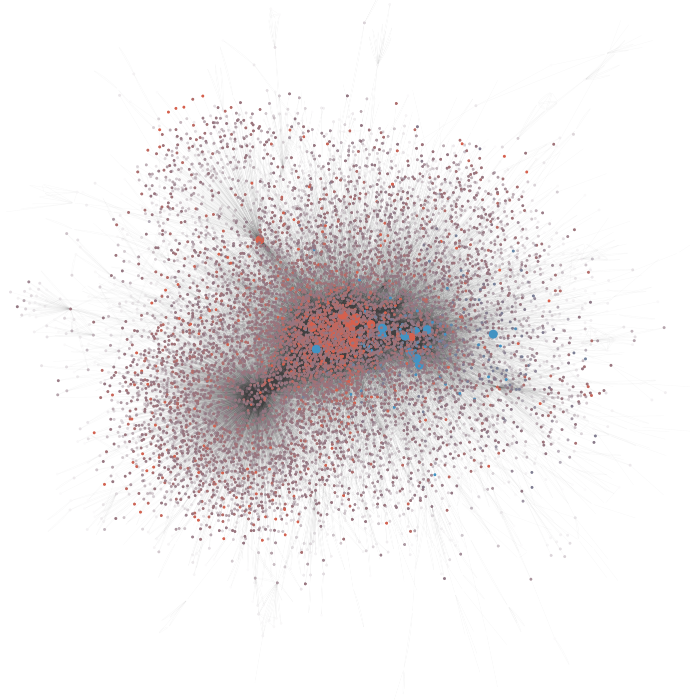
Further insights have been derived in the second part of our empirical study. We showed that the price of competition that multiple entities may experience depends strongly on the structure of the instance graphs but in general has worse effects on the leader. Since the leading entity may suffer from enormous losses compared to the follower, it is advisable to consider different first mover strategies. Such investigations open a potential research avenue in the direction of game theory and bi-level programming.

Acknowledgments

This work has been supported by the Vienna Science and Technology Fund (project ICT15-014), by the Federal Ministry of Education, Science and Research of Austria, and by the the Austrian Agency for International Mobility and Cooperation in Education, Science and Research (reference number ICM-2019-13384).



(a) $|L| = 10, F = \emptyset$



(b) $|L| = 10, |F| = 10$

Figure 8: Expected numbers of activations of instance `soc-anybeat`. Nodes that are activated by the leader in all scenarios are colored blue, while nodes that are activated by the follower in all scenarios are colored red. Nodes that are activated by both the leader and the follower in different scenarios have a correspondingly mixed color. The total number of activations is illustrated by the opacity of a node. Seed nodes have a larger radius than the remaining nodes.

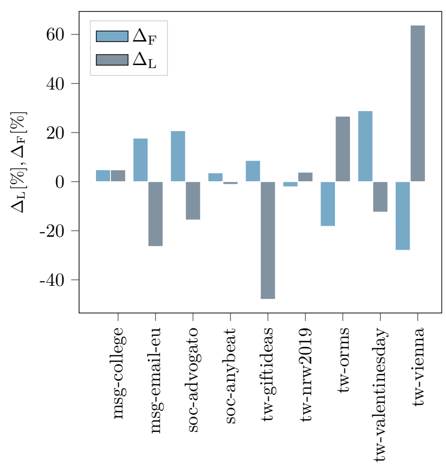


Figure 9: Relative changes Δ_L and Δ_F in the objective values for the leader and the follower, respectively, if the leader seed sets are obtained by heuristic BIN instead of exact method BEN. Positive values indicate a preference for method BEN.

References

[1] Gisela M. Bardossy and S. Raghavan. An inexact sample average approximation approach for the stochastic connected facility location problem. *Networks*, 70(1):19–33, 2017.

[2] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[3] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, pages 306–311. Springer, 2007.

[4] Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam DI Kramer, Cameron Marlow, Jaime E. Settle, and James H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295, 2012.

[5] Allan Borodin, Mark Braverman, Brendan Lucier, and Joel Oren. Strategyproof mechanisms for competitive influence in networks. *Algorithmica*, 78(2):425–452, 2017.

[6] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674. ACM, 2011.

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

[8] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1029–1038, 2010.

[9] Wei Chen, Laks V.S. Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[10] Richard Church and Charles ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32(1):101–118, 1974.

[11] Jean-François Cordeau, Fabio Furini, and Ivana Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882 – 896, 2019.

[12] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66. ACM, 2001.

[13] Matteo Fischetti, Michael Kahr, Markus Leitner, Michele Monaci, and Mario Ruthmair. Least cost influence propagation in (social) networks. *Mathematical Programming*, 170(1):293–325, 2018.

[14] World Economic Forum. *The Global Risks Report 2013.* 2013.

[15] Evren Güney. An efficient linear programming based method for the influence maximization problem in social networks. *Information Sciences*, 503:589–605, 2019.

[16] Evren Güney, Markus Leitner, Mario Ruthmair, and Markus Sinnl. Large-scale influence maximization via maximal covering location. Technical report, University of Vienna, 2019. submitted to a journal.

[17] Meng Han and Yingshu Li. Influence analysis: A survey of the state-of-the-art. *Mathematical Foundations of Computing*, 1(3):201–253, 2018.

[18] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the SIAM International Conference on Data Mining*, pages 463–474, 2012.

[19] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146. ACM, 2003.

[20] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[21] M. Emre Keskin and Mehmet Güray Güler. Influence maximization in social networks: an integer programming approach. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(6):3384–3397, 2018.

[22] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12 (2):479–502, 2002.

[23] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[24] Matthew K.O. Lee, Na Shi, Christy M.K. Cheung, Kai H. Lim, and Choon Ling Sia. Consumer's decision to shop online: The moderating role of positive informational social influence. *Information & Management*, 48(6):185–191, 2011.

[25] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, 2014. http://snap.stanford.edu/data, last accessed on 2019/12/12.

[26] Xiang Li, J. David Smith, Thang N. Dinh, and My T. Thai. Tiptop: Almost exact solutions for influence maximization in billion-scale networks. *IEEE/ACM Transactions on Networking*, 27(2):649–661, 2019.

[27] Yishi Lin and John C.S. Lui. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation*, 91: 187–204, 2015.

[28] Dong Liu, Yun Jing, Jing Zhao, Wenjun Wang, and Guojie Song. A fast and efficient algorithm for mining top-k nodes in complex networks. *Scientific Reports*, 7(1), 2017.

[29] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[30] Roberto Cavazos (University of Boltimore) and Cheq. The economic cost of bad actors on the internet: fake news, 2019. https://www.cheq.ai/fakenews, last accessed on 2019/12/22.

[31] Canh V. Pham, My T. Thai, Dung Ha, Dung Q. Ngo, and Huan X. Hoang. Time-critical viral marketing strategy with the competition on online social networks. In *Computational Social Networks: 5th International Conference Proceedings*, pages 111–122, 2016.

[32] Fabián Riquelme and Pablo González-Cantergiani. Measuring user influence on twitter: A survey. *Information processing & management*, 52(5):949–975, 2016.

[33] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015. http://networkrepository.com, last accessed on 2019/12/12.

[34] Mohammadreza Samadi, Alexander Nikolaev, and Rakesh Nagi. A subjective evidence model for influence maximization in social networks. *Omega*, 59:263–278, 2016.

[35] Mohammadreza Samadi, Rakesh Nagi, Alexander Semenov, and Alexander Nikolaev. Seed activation scheduling for influence maximization in social networks. *Omega*, 77:96 – 114, 2018.

[36] Mike Schmidt. Calculating the true size of the influencer marketing industry. *Forbes*, 2019. https://www.forbes.com/sites/forbestechcouncil/2019/02/13/calculating-the-true-size-of-the-influencer-marketing-industry/, last accessed on 2019/12/22.

[37] Michael Trusov, Randolph E. Bucklin, and Koen Pauwels. Effects of word-of-mouth versus traditional marketing: Findings from an internet social networking site. *Journal of Marketing*, 73(5):90–102, 2009.

[38] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*, volume 8. Cambridge University Press, 1994.

[39] Hao-Hsiang Wu and Simge Küçükyavuz. A two-stage stochastic programming approach for influence maximization in social networks. *Computational Optimization and Applications*, 69 (3):563–595, 2018.

A Detailed results

Table 2: Detailed results for $|\Omega'| = 750$. We report the average runtimes $\bar{t}^{\mathrm{X}}[s]$ over all 10 SAA iterations in seconds, the estimated objective values of the follower $\hat{\sigma}^{\mathrm{X}}_{\Omega''}(L, \hat{F}^*)$ and the leader $\hat{\lambda}^{\mathrm{X}}_{\Omega''}(L, \hat{F}^*)$. Superscripts ^{X} indicate the used method, i.e., an empty superscript indicates the BEN method while B and M the BIN and MAR methods, respectively. We further report the and the estimated approximation gap $\Delta[\%]$ in percent.

Instance	$\|L\|$	$\|F\|$	$\bar{t}[s]$	$\bar{t}^{\mathrm{M}}[s]$	$\bar{t}^{\mathrm{B}}[s]$	$\hat{\sigma}_{\Omega''}(L,\hat{F}^*)$	$\hat{\sigma}^{\mathrm{M}}_{\Omega''}(L,F)$	$\hat{\sigma}^{\mathrm{B}}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}^{\mathrm{M}}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}^{\mathrm{B}}_{\Omega''}(L,\hat{F}^*)$	$\Delta[\%]$
msg-college	0	10	26	20	0	355.8	355.8	308.3	0.0	0.0	0.0	0.02
		15	42	21	0	376.2	376.2	326.7	0.0	0.0	0.0	0.0
		20	55	21	0	393.4	393.4	339.9	0.0	0.0	0.0	0.04
	10	10	129	7	0	237.0	236.2	190.5	125.9	130.9	173.2	0.03
		15	316	8	0	276.0	275.8	229.9	99.1	100.6	141.3	0.03
		20	317	9	0	304.4	303.3	251.0	79.6	83.3	127.2	0.08
	15	10	97	6	0	202.2	202.2	156.0	180.7	180.7	225.0	0.05
		15	167	7	0	243.8	243.8	190.6	143.9	143.9	198.1	0.04
		20	214	7	0	274.1	273.4	222.5	120.2	119.9	170.2	0.03
	20	10	104	6	0	187.8	187.2	143.7	211.6	213.1	253.8	0.09
		15	233	7	0	228.9	227.3	184.1	174.9	180.1	217.0	0.14
		20	363	7	0	258.7	257.0	205.7	150.4	151.9	200.5	0.03
msg-email-eu	0	10	31	32	0	444.8	444.8	436.3	0.0	0.0	0.0	0.01
		15	47	30	0	454.5	454.5	441.5	0.0	0.0	0.0	0.02
		20	60	30	0	463.3	463.3	445.7	0.0	0.0	0.0	0.02
	10	10	603	9	0	360.6	360.5	329.8	86.2	86.4	116.3	0.08
		15	2316	10	0	384.3	383.5	342.5	67.4	66.7	104.8	0.02
		20	4391	11	0	399.7	398.6	359.5	55.4	53.9	89.1	0.06
	15	10	336	8	0	336.5	336.1	310.2	120.3	120.2	145.5	0.13
		15	324	9	0	364.8	364.2	325.3	93.8	94.3	131.5	0.05
		20	1311	10	0	382.1	381.3	344.6	80.7	80.7	114.4	0.01
	20	10	92	6	0	321.1	321.1	300.9	144.0	144.0	163.4	0.12
		15	136	7	0	352.0	351.2	326.6	114.8	115.7	138.0	0.06
		20	525	8	0	370.2	369.9	335.9	98.4	100.1	130.8	0.09
soc-advogato	0	10	166	51	0	556.1	556.1	481.8	0.0	0.0	0.0	0.11
		15	316	53	0	595.7	594.2	498.4	0.0	0.0	0.0	0.12
		20	512	54	0	628.6	626.7	499.5	0.0	0.0	0.0	0.08
	10	10	226	24	0	390.3	390.3	281.2	173.1	173.1	287.5	0.22
		15	372	26	0	449.2	449.2	343.9	129.2	129.2	237.0	0.05
		20	647	28	0	493.4	493.3	385.7	108.4	114.1	207.9	0.09
	15	10	413	22	0	346.5	346.5	258.2	252.2	252.2	345.3	0.0
		15	861	23	0	408.2	407.8	312.4	202.8	207.0	300.8	0.05
		20	726	25	0	457.8	456.9	332.8	167.6	167.5	284.0	0.03
	20	10	441	22	0	318.0	318.0	252.7	317.6	317.6	388.2	0.01
		15	816	22	0	378.1	378.1	293.7	265.0	265.0	350.8	0.01
		20	1440	24	0	426.1	425.7	311.0	222.4	227.4	338.6	0.0
soc-anybeat	0	10	358	243	0	1179.1	1179.1	1134.0	0.0	0.0	0.0	0.03
		15	394	197	0	1203.3	1203.3	1146.6	0.0	0.0	0.0	0.02
		20	691	215	0	1221.5	1221.5	1155.0	0.0	0.0	0.0	0.01
	10	10	302	76	0	1035.3	1030.5	898.4	144.2	148.7	283.5	0.01
		15	340	70	0	1074.4	1071.0	924.9	117.0	115.5	258.8	0.01
		20	494	75	0	1098.7	1096.1	930.5	111.5	111.7	258.8	0.01
	15	10	407	67	0	978.4	976.6	815.6	225.2	226.6	390.5	0.02
		15	603	78	0	1023.6	1022.1	849.2	181.9	183.2	359.6	0.01
		20	979	81	0	1052.9	1052.6	854.7	155.5	153.2	359.6	0.01
	20	10	433	62	0	946.1	940.2	742.9	275.6	281.3	481.5	0.03
		15	681	74	0	995.3	995.3	750.3	228.4	228.3	479.2	0.03
		20	1283	78	0	1028.3	1028.2	755.7	195.5	195.7	478.5	0.01
tw-giftideas	0	10	345	528	0	2307.6	2307.6	2269.0	0.0	0.0	0.0	0.0
		15	802	527	0	2324.5	2324.5	2274.0	0.0	0.0	0.0	0.0
		20	2051	564	0	2339.7	2339.6	2278.3	0.0	0.0	0.0	0.0
	10	10	471	216	0	2226.6	2226.0	2186.5	92.0	92.7	121.1	0.01
		15	1644	220	0	2242.2	2241.4	2207.4	92.0	92.7	100.3	0.0
		20	4035	217	0	2257.3	2256.7	2209.7	88.9	92.4	97.9	0.01
	15	10	369	216	0	2216.0	2215.1	2151.6	109.1	110.0	172.9	0.0
		15	1306	203	0	2231.3	2230.3	2165.8	109.0	110.0	158.5	0.0
		20	1753	210	0	2246.0	2244.9	2182.7	108.9	110.0	141.8	0.0
	20	10	350	216	0	2215.1	2214.3	2150.9	125.0	126.0	188.7	0.0
		15	903	211	0	2229.9	2229.0	2165.2	125.0	125.9	174.5	0.0
		20	783	209	0	2243.3	2242.7	2181.9	122.9	122.9	157.7	0.0

Table 2 continued.

Instance	$\lvert L\rvert$	$\lvert F\rvert$	$\bar{t}[s]$	$\bar{t}^{\mathrm{M}}[s]$	$\bar{t}^{\mathrm{B}}[s]$	$\hat{\sigma}_{\Omega''}(L,\hat{F}^*)$	$\hat{\sigma}^{\mathrm{M}}_{\Omega''}(L,F)$	$\hat{\sigma}^{\mathrm{B}}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}^{\mathrm{M}}_{\Omega''}(L,\hat{F}^*)$	$\hat{\lambda}^{\mathrm{B}}_{\Omega''}(L,\hat{F}^*)$	$\Delta[\%]$
tw-nrw2019	0	10	133	59	0	861.2	861.2	831.6	0.0	0.0	0.0	0.01
		15	238	62	0	915.7	915.7	863.4	0.0	0.0	0.0	0.05
		20	384	65	0	957.9	957.9	867.9	0.0	0.0	0.0	0.06
	10	10	356	15	0	291.5	291.5	231.2	626.6	626.6	653.8	0.04
		15	1267	17	0	353.2	353.2	267.9	607.6	607.6	635.0	0.07
		20	2943	17	0	405.4	405.4	307.8	583.2	586.7	609.8	0.16
	15	10	350	14	0	264.4	264.4	195.9	693.9	693.9	731.1	0.02
		15	1134	15	0	320.6	320.6	240.4	671.8	671.8	701.2	0.06
		20	2298	17	0	368.3	368.2	277.2	646.5	646.7	681.1	0.17
	20	10	313	14	0	237.2	237.2	181.9	751.2	751.2	785.4	0.03
		15	788	14	0	287.5	287.5	229.1	723.9	723.9	748.7	0.22
		20	1536	16	0	329.3	329.3	262.9	703.0	707.4	726.6	0.22
tw-orms	0	10	27	2	0	91.2	91.2	78.0	0.0	0.0	0.0	0.1
		15	43	2	0	108.0	108.0	85.6	0.0	0.0	0.0	0.03
		20	67	3	0	120.9	120.9	89.7	0.0	0.0	0.0	0.09
	10	10	18	2	0	45.8	45.8	35.6	67.1	67.1	72.7	0.26
		15	31	2	0	59.9	59.9	43.3	63.9	63.9	69.8	0.04
		20	102	2	0	71.7	71.7	50.2	62.9	62.9	68.0	0.07
	15	10	21	1	0	37.0	37.0	29.3	86.3	86.3	89.5	0.01
		15	57	2	0	49.5	49.5	41.6	81.1	81.1	81.0	0.05
		20	119	2	0	60.7	60.7	50.6	77.5	76.9	78.0	0.03
	20	10	27	1	0	35.7	35.7	25.4	91.4	91.4	102.1	0.11
		15	71	1	0	47.7	47.7	38.7	89.5	89.5	91.2	0.06
		20	114	2	0	58.4	58.4	45.2	86.1	86.1	87.8	0.12
tw-valentinesday	0	10	24	13	0	452.3	452.4	361.5	0.0	0.0	0.0	0.03
		15	60	15	0	466.5	466.4	366.0	0.0	0.0	0.0	0.02
		20	209	14	0	477.7	477.8	365.9	0.0	0.0	0.0	0.04
	10	10	75	7	0	287.7	287.7	223.8	177.8	177.8	228.7	0.03
		15	256	8	0	300.5	300.5	223.9	174.5	174.5	228.5	0.03
		20	1288	8	0	311.9	311.9	235.6	172.0	172.0	216.9	0.05
	15	10	65	7	0	285.7	285.7	223.8	187.1	187.1	242.8	0.05
		15	401	8	0	297.2	297.2	223.9	186.8	186.8	242.7	0.05
		20	4443	7	0	307.7	307.7	235.6	184.4	184.4	231.1	0.05
	20	10	77	7	0	282.9	282.9	223.3	201.0	201.1	254.6	0.05
		15	1727	7	0	293.6	293.6	223.4	198.5	198.5	254.5	0.06
		20	5776	7	0	303.6	303.6	235.1	198.4	198.5	242.9	0.02
tw-vienna	0	10	68	13	0	190.9	190.9	125.5	0.0	0.0	0.0	0.1
		15	124	14	0	250.5	250.5	150.0	0.0	0.0	0.0	0.12
		20	143	14	0	297.0	297.0	153.6	0.0	0.0	0.0	0.12
	10	10	27	12	0	117.8	117.8	53.1	157.7	157.7	169.2	0.05
		15	65	13	0	160.1	160.1	67.8	149.8	149.9	167.4	0.08
		20	219	13	0	193.7	193.7	71.4	149.8	149.8	166.6	0.06
	15	10	82	13	0	99.1	99.1	48.1	210.9	210.9	214.8	0.09
		15	176	12	0	133.3	133.3	48.5	203.7	203.7	214.4	0.04
		20	420	13	0	164.8	164.8	50.0	195.5	195.5	214.1	0.11
	20	10	134	12	0	86.4	86.4	48.1	250.6	250.6	261.3	0.04
		15	336	12	0	117.9	117.9	48.5	242.3	242.3	260.9	0.19
		20	560	13	0	147.6	147.6	50.0	235.2	235.2	260.6	0.11