Large-scale Influence Maximization via Maximal Covering Location*

Evren Güney[1], Markus Leitner[2], Mario Ruthmair[2], and Markus Sinnl[2]

[1]MEF University, Department of Industrial Engineering, Turkey, guneye@mef.edu.tr
[2]University of Vienna, Department of Statistics and Operations Research, Austria, {markus.leitner|mario.ruthmair|markus.sinnl}@univie.ac.at

December 11, 2018

Abstract

Influence maximization aims at identifying a limited set of key individuals in a (social) network which spreads information based on some propagation model and maximizes the number of individuals reached. We show that influence maximization based on the probabilistic independent cascade model can be modeled as a stochastic maximal covering location problem. A reformulation based on Benders decomposition is proposed and a relation between obtained Benders optimality cuts and submodular cuts for correspondingly defined subsets is established. In a computational study our branch-and-cut approaches outperform the state-of-the-art approaches for influence maximization by orders of magnitude.

1 Introduction

Recently, one can observe an increased interest in optimization problems related to the propagation of information in (social) networks, see, e.g., the seminal work in [11] (and its journal version [12]) and an extensive introduction and overview on the topic in [2]. Variants of the so-called influence maximization problem typically aim at finding a small set of key individuals in the network which is denoted as "seed set". External means (e.g., monetary incentives) are used to convince all members of this seed set to spread some information to all directly connected individuals (e.g., followers or friends). This initial spread of information triggers a propagation process in which every *active* individual forwards (the previously received) information to its neighbors. Thereby, each individual exerts a specific influence to each of its neighbors and a higher influence increases the probability that the receiving individual is activated (e.g., convinced of an information or product). The goal of influence maximization problems is choose a seed set of limited size such that the (expected) number of finally active individuals is maximized. Closely related problem variants want to minimize the costs for a seed set given that a certain fraction of all individuals is reached, see, e.g., [7]. Besides (viral) marketing in (social) networks, similar propagation models have applications in, e.g., epidemiology [6, 13].

Two main classes of information propagation models have been considered in the literature: (i) threshold models and (ii) cascade models. In the linear threshold model proposed by [8], each individual is assigned a hurdle or threshold value and each directed connection between two individuals has a predefined influence value. An individual not contained in the seed set is activated the (weighted) sum of influence values from active neighbors is at least as high as its hurdle. In the general threshold model arbitrary functions of the influence values from active neighbors and incentive values can be used, see, e.g., [7, 16]. Cascade models [11] assign probabilities to individual connections that represent the chance that the source individual activates the target individual. Activation attempts are made immediately after the source gets active and activation probabilities may change with time and depend

*This work was funded by the Vienna Science and Technology Fund through project ICT15-014 and by the Austrian Research Fund (FWF, Project P 26755-N19).

on the set of individuals who already tried to influence a particular target individual. The special case of the *independent cascade model* in which each activation probability is constant and independent from previous attempts to influence the target individual has received a lot of attention in the literature. To this end we refer, e.g., to [1] for a competitive two-player variant and [15] for a survey on fast and scalable heuristics. In [11] the authors have shown that the most general variants of the two classes—the general threshold and the general cascade model—are in fact equivalent, whereas this is not the case for the linear threshold and the independent cascade model.

The *influence maximization problem (IMP)* considered in this article is defined on a directed graph $G = (V, A)$ in which nodes V represent individuals and arcs $(i, j) \in A$ correspond to individual i influencing (or activating) j with probability p_{ij}, $0 \leq p_{ij} \leq 1$. Information propagation is based on the independent cascade model and we assume that active individuals cannot be deactivated again. A feasible solution to the IMP is defined by a seed set $S \subseteq V$ whose size is limited by parameter $K \leq |V|$. We aim to find a seed set S^* which maximizes the expected number of active nodes after the propagation process. Exactly evaluating the objective function for a given seed set would require the consideration of all $2^{|A|}$ combinations of arc influence realizations with according occurrence probabilities. Since this is not tractable for reasonably-sized instances, in [11] the probabilistic propagation process is approximated by a restricted set Ω of scenarios obtained by Monte-Carlo sampling. Each scenario $\omega \in \Omega$ is represented by *live-arc graph* $G^\omega = (V, A^\omega)$, $A^\omega \subseteq A$, where arc $(i, j) \in A^\omega$ indicates that a previously inactive node j is activated by i in scenario ω if i is active. Arcs $(i, j) \in A$ are included with probability p_{ij} (decided by biased coin flipping) in each scenario, i.e., in arc set A^ω. Let $\Phi^\omega(S)$ denote the number of activated nodes in scenario $\omega \in \Omega$ for a given seed set S, and p_ω be the probability of scenario $\omega \in \Omega$. Then, the variant of the IMP considered in this article is formally defined as

$$S^* := \text{argmax}_{S \subseteq V, |S| \leq K} \sum_{\omega \in \Omega} p_\omega \Phi^\omega(S).$$

We note that Kempe et al. [11] show that the linear threshold model can also be represented via live-arc graphs. Thus, while we focus on the independent cascade model our approach can also be used for the linear threshold model after the transformation given in [11].

1.1 Previous work

In their seminal paper, Kempe et al. [11] show that for one particular scenario $\omega \in \Omega$ the real-valued set function $\Phi^\omega(S)$ is non-decreasing and submodular for all $S \subseteq V$. Let $\rho_i^\omega(S) := \Phi^\omega(S \cup \{i\}) - \Phi^\omega(S), \forall S \subset V, i \in V$, be the marginal gain achieved by adding element i to set S. According to the general result in [18], a function $\Phi^\omega(S)$ is non-decreasing and submodular iff $\Phi^\omega(T) \leq \Phi^\omega(S) + \sum_{i \in T \setminus S} \rho_i^\omega(S), \forall S, T \subseteq V$. Furthermore, if there is a cardinality constraint on S, e.g., $|S| \leq K$ (as in our case), a simple greedy heuristic which starts with $S = \emptyset$ and iteratively adds node $i \notin S$ with maximal marginal gain gives an $1 - 1/e$ approximation algorithm [18]. As the objective function of the IMP is a convex combination of submodular functions (which remains submodular), applying the generic greedy algorithm in [18] to the IMP gives the same approximation ratio. The computation of the expected marginal gain for all nodes can, however, be computationally time-consuming, especially for large graphs and a high number of scenarios. Therefore, several acceleration techniques have been proposed to improve the running time of the greedy algorithm, see, e.g., [15] for a survey on this topic.

The exact solution of the IMP received much less attention: Güney [9] uses a time-indexed mixed integer linear programming (MILP) formulation for the IMP to model the propagation process in the scenario-specific live-arc graphs. The same author proposes an approximation algorithm based on the linear programming (LP) relaxation of a covering model for the IMP in [10]. In [22], a MILP formulation is proposed which is based on the fact that maximizing a non-decreasing and submodular function can be reformulated by using an exponentially-sized family of cutting planes, see [18]. To the best of our knowledge, the approach in [22] is state-of-the-art for solving the IMP and thus will be compared with our methods theoretically and in experiments.

1.2 Contribution and outline

In this work, we show how the IMP can be modeled as a *stochastic maximal covering location problem (SMCLP)*. The *maximal covering location problem (MCLP)* [4] is a classical problem from the area

of location science. We present a compact model for the IMP and a reformulation based on Benders decomposition. We also clarify the theoretical relation between the obtained Benders optimality cuts and submodular cuts for sets of arbitrary size. Computational results show that we outperform the state-of-the-art in many cases by orders of magnitude.

The remainder of the paper is organized as follows: Section 2 discusses the relation between influence maximization and stochastic maximal covering location and proposes a cover formulation together with a computationally appealing reformulation. Section 3 describes the algorithm to solve the reformulation and Section 4 shows computational experiments. Finally, Section 5 concludes the paper.

2 Stochastic maximal covering location formulation

In this section, we will establish a close relation between the IMP and a stochastic variant of the maximal covering location problem [4] which has also been observed in [10]. In contrast to the latter work, we not only use this relation to derive an MILP formulation for the IMP but additionally exploit it to obtain a reformulation leading to an effective Benders decomposition algorithm. For this, first observe that the number of activated nodes $\Phi^\omega(S)$ for scenario $\omega \in \Omega$ and seed set $S \subseteq V$ corresponds exactly to the number of nodes which are reachable in live-arc graph G^ω from at least one of the nodes in S. Thus, node i is activated in scenario ω iff set S includes a node from the *reachability* set $R(\omega, i)$ of all nodes with a directed path to i in G^ω. From a different perspective, node j covers node i in scenario $\omega \in \Omega$ if $j \in R(\omega, i)$. The latter point of view reveals the relation to a stochastic maximum covering location problem that is formally shown in Theorem 1.

Theorem 1. *The IMP is a special case of a maximal covering location problem with uncertain facility-customer relations.*

Proof. Given sets of facility sites J, demand nodes I with demands $a_i > 0$, $\forall i \in I$, and facility sites $N_i \subset J$ that cover demand node $i \in I$, the MCLP [4] aims to maximize the demand covered by $p \in \mathbb{N}$ facilities. Based on the discussion above, it is easy to observe, that the IMP corresponds to a two-stage stochastic variant of the maximal covering location problem, in which the sets of facilities and demand nodes are identical and correspond to V (i.e., $I = J = V$) and all demand are equal to one (i.e., $a_i = 1$, $\forall i \in V$). Furthermore, a set of scenarios Ω, each with probability p_ω, $\forall \omega \in \Omega$, is given and the facility sites $N_i^\omega \subseteq V$ covering demand nodes $i \in V$ in scenario $\omega \in \Omega$ are defined by sets $R(\omega, i)$, i.e., some facility j covers customer i iff $j \in R(\omega, i)$. The objective of this problem variant is to maximize the expected demand covered while selecting exactly $p = K$ facility locations. □

For each scenario $\omega \in \Omega$ and node $i \in V$, sets $R(\omega, i)$ can be easily determined by, e.g., a reverse breadth-first search (BFS) starting from i in live-arc graph G^ω. Thus, the result of Theorem 1 can be used to see that formulation (1) is a valid model for the IMP as it is a comparably straightforward scenario-based extension (for the special case described in the proof of Theorem 1) of a classical MILP formulation for the MCLP [4]. Thereby, binary variable z_i is equal to one iff node $i \in V$ is part of the seed set, and binary variable x_i^ω is equal to one iff node $i \in V$ is activated in scenario $\omega \in \Omega$.

$$\max \sum_{\omega \in \Omega} p_\omega \sum_{i \in V} x_i^\omega \tag{1a}$$

$$\sum_{i \in V} z_i \leq K \tag{1b}$$

$$x_i^\omega \leq \sum_{j \in R(\omega, i)} z_j \qquad \forall i \in V, \forall \omega \in \Omega \tag{1c}$$

$$x_i^\omega \in \{0, 1\} \qquad \forall i \in V, \forall \omega \in \Omega \tag{1d}$$

$$z_i \in \{0, 1\} \qquad \forall i \in V \tag{1e}$$

Objective function (1a) computes the expected number of activated nodes. The reachability constraint (1c) for node i and scenario ω guarantees that node i can only be active in scenario ω if at least one of the nodes in $R(\omega, i)$ is chosen as seed node.

2.1 Reformulation

Note that depending on the number of scenarios the set of variables $x_i^\omega, \forall i \in V, \forall \omega \in \Omega$, and constraints (1c) might be quite large. To effectively solve also large instances, we will now introduce a reformulation of (1) based on Benders decomposition, see, e.g., [5] for a similar approach in the case of the MCLP. First observe, that for fixed integer $\bar{z}_i \in \{0,1\}, \forall i \in V$, the integrality of variables x_i^ω can be relaxed. Moreover, the remaining problem decomposes into one subproblem for each scenario $\omega \in \Omega$. These characteristics make the problem attractive for Benders decomposition. Let μ_ω denote the contribution of scenario $\omega \in \Omega$ to the objective function. Then, we reformulate (1) in the following way:

$$\max \sum_{\omega \in \Omega} p_\omega \mu_\omega \tag{2a}$$

$$\sum_{i \in V} z_i \leq K \tag{2b}$$

$$\mu_\omega \leq \Theta^\omega(\boldsymbol{z}) \qquad \forall \omega \in \Omega \tag{2c}$$

$$z_i \in \{0,1\} \qquad \forall i \in V. \tag{2d}$$

Note that each seed set defined by variables $z_i, \forall i \in V$, which satisfies cardinality constraints (2b) leads to a feasible solution. Thus, no Benders feasibility cuts are needed for the validity of reformulation (2). Inequalities (2c) bound the objective by function $\Theta^\omega(\boldsymbol{z})$ which gives the maximal number of active nodes in scenario $\omega \in \Omega$ for any solution $\boldsymbol{z} \in [0,1]^{|V|}$. Benders optimality cuts to model this function are derived by considering the dual problem of the LP relaxation of (1) with optimal objective value $\Theta^\omega(\bar{\boldsymbol{z}})$ for a fixed vector $\bar{\boldsymbol{z}}$: Let $\alpha_i^\omega, \forall i \in V, \forall \omega \in \Omega$, be the dual variables associated with upper bound constraints $x_i^\omega \leq 1$, and $\beta_i^\omega, \forall i \in V, \forall \omega \in \Omega$, be the dual variables associated with covering constraints (1c). Then, for one particular scenario $\omega \in \Omega$ the dual is

$$\Theta^\omega(\bar{\boldsymbol{z}}) := \min \left\{ \sum_{i \in V} \left(\alpha_i^\omega + \beta_i^\omega \sum_{j \in R(\omega,i)} \bar{z}_j \right) : (\boldsymbol{\alpha}^\omega, \boldsymbol{\beta}^\omega) \in P^\omega \right\} \tag{3}$$

with

$$P^\omega := \{(\boldsymbol{\alpha}^\omega, \boldsymbol{\beta}^\omega) : \boldsymbol{\alpha}^\omega + \boldsymbol{\beta}^\omega \geq \mathbf{1}, \boldsymbol{\alpha}^\omega, \boldsymbol{\beta}^\omega \geq \mathbf{0}\}. \tag{4}$$

It is easy to see that there exists an optimal solution to (3) in which $(\alpha_i^\omega, \beta_i^\omega) \in \{(1,0),(0,1)\}$ for each $i \in V$, depending on the corresponding coefficients in the objective function. If $\sum_{j \in R(\omega,i)} \bar{z}_j < 1$, then $\beta_i^\omega = 1$ (and $\alpha_i^\omega = 0$), otherwise, $\alpha_i^\omega = 1$ (and $\beta_i^\omega = 0$). Thus, the dual can be solved by inspection for each scenario $\omega \in \Omega$. Note that we always obtain an integer dual optimal solution, also for fractional values of $\bar{\boldsymbol{z}}$.

We denote by $C^\omega(\bar{\boldsymbol{z}}) := \sum_{i \in V} \alpha_i^\omega$ the constant and by $c_j^\omega(\bar{\boldsymbol{z}}) := \sum_{i : j \in R(\omega,i)} \beta_i^\omega$ the coefficient of variable $z_j, \forall j \in V$, in Benders optimality cuts

$$\mu_\omega \leq C^\omega(\bar{\boldsymbol{z}}) + \sum_{j \in V} c_j^\omega(\bar{\boldsymbol{z}}) z_j. \tag{5}$$

2.2 Relation to submodular cuts

The decomposition approach in [22] uses optimality cuts based on the submodularity and monotonicity of function Φ^ω, cf. section 1.1. Using the variables and notation given above, they are defined by

$$\mu_\omega \leq \Phi^\omega(S) + \sum_{j \in V \setminus S} \rho_j^\omega(S)\ z_j, \qquad \forall S \subseteq V : |S| \leq K, \forall \omega \in \Omega. \tag{6}$$

While only inequalities (6) for sets S of size exactly K are needed for correctness of the formulation, those associated with sets S of smaller size can be computationally beneficial as observed in [17]. In fact, the latter inequalities potentially strengthen the LP relaxation of the formulation including only submodular cuts (6) for sets S of size exactly K, as shown in the following example: Consider graph

G with nodes $V = \{1, 2, 3, 4\}$, arcs $A = \{(1, 4), (2, 4), (3, 4)\}$, a maximal seed set size $K = 2$, and influence probability $p_{ij} = 1, \forall (i, j) \in A$, and a single scenario ω with $G^\omega = G$. An optimal integer solution has objective value 3 and can be obtained by selecting any two of the three nodes $1, 2, 3$. A formulation including all submodular cuts for sets of size exactly $K = 2$ leads to an optimal LP solution $\bar{z}_1 = \bar{z}_2 = \bar{z}_3 = \frac{2}{3}, \bar{z}_4 = 0$, with objective $\mu = \frac{10}{3}$ (bounded above by the six submodular cuts for all pairs of nodes). Submodular cut (6) for the single node set $S = \{4\}$ is defined as $\mu \leq 1 + z_1 + z_2 + z_3$ and thus is violated by the LP solution above. Adding the latter cut to the model results in an optimal integer LP solution with value $\mu = 3$.

Theorem 2. *The Benders optimality cut (5) corresponding to a given (fractional) solution $\bar{\boldsymbol{z}}$ is equivalent to the submodular cut (6) for set $S := \{i \in V : \sum_{j \in R(\omega,i)} \bar{z}_j \geq 1 \wedge \sum_{j \in R(\omega,k)} \bar{z}_j < 1, \forall k \in R(\omega, i) \setminus \{i\}\}$.*

Proof. Note that different sets can lead to the same submodular cut, e.g., $S' := \{i \in V : \sum_{j \in R(\omega,i)} \bar{z}_j \geq 1\} \supseteq S$ results in the same inequality as set S which, however, is a smallest one since it does not contain nodes which are reachable from other nodes in S (due to the second term in the definition of S).

We show that $C^\omega(\bar{\boldsymbol{z}}) = \Phi^\omega(S)$ and $c_j^\omega(\bar{\boldsymbol{z}}) = \rho_j^\omega(S), \forall j \in V$. By definition of S, $\sum_{j \in R(\omega,i)} \bar{z}_j \geq 1$ holds for all nodes $i \in V$ reachable from some node in seed set S. In this case, we set $\alpha_i^\omega = 1$ and $\beta_i^\omega = 0$. For all nodes i which cannot be activated by S, $\sum_{j \in R(\omega,i)} \bar{z}_j < 1$ holds and thus, $\alpha_i^\omega = 0$ and $\beta_i^\omega = 1$. Therefore, $C^\omega(\bar{\boldsymbol{z}}) = \sum_{i \in V} \alpha_i^\omega$ corresponds to the number of nodes activated by seed set S, i.e., $C^\omega(\bar{\boldsymbol{z}}) = \Phi^\omega(S)$. For one particular node $j \in V$, coefficient $c_j^\omega(\bar{\boldsymbol{z}}) = \sum_{i : j \in R(\omega,i)} \beta_i^\omega$ denotes the number of not yet active nodes which would be activated by node j if j is added to the seed set which corresponds to the marginal gain of node j, i.e., $c_j^\omega(\bar{\boldsymbol{z}}) = \rho_j^\omega(S)$. Note that the marginal gain of nodes from S and all other active nodes is zero. □

Corollary 1. *Benders optimality cuts associated to integer solutions $\bar{\boldsymbol{z}}$ and submodular cuts (6) for correspondingly defined seed sets $S := \{i \in V : \bar{z}_i = 1\}$ are equivalent.*

Even though the two sets of inequalities are equivalent for integer solutions, compared to [22] our modeling approach allows an efficient exact separation routine for fractional solutions which does not rely on solving the LP relaxation. These Benders cuts based on fractional solutions turn out to improve the computational results significantly, see Section 4.

Interestingly, set S obtained in Theorem 2 can potentially have any size, also larger than K, even for solutions with $\sum_{i \in V} \bar{z}_i = K$. For the LP solution $\bar{z}_1 = \bar{z}_2 = \bar{z}_3 = \frac{2}{3}, \bar{z}_4 = 0$, in the example above, by applying Theorem 2 we obtain $S := \{4\}$ leading exactly to the submodular inequality mentioned above which cuts off this LP solution. To summarize, the Benders cuts for fractional solutions potentially lead to submodular cuts for sets of size different to K, which might not be generated otherwise since integral solutions handed over by the MILP solver very likely will have size K.

3 Solution algorithm

To solve the IMP we use a branch-and-cut algorithm based on the Benders reformulation (2). In general it is possible to start with a restricted master problem which does not include any of the Benders cuts (2c). However, it turned out to be computationally advantageous to initially include some cuts to bound variables μ_ω from above, otherwise the first solution of the master problem would be unbounded. Thus, we add the following inequalities which correspond to submodular cuts for $S = \emptyset$, i.e.,

$$\mu_\omega \leq \sum_{j \in V} \Phi^\omega(\{j\}) z_j, \qquad \forall \omega \in \Omega. \tag{7}$$

In the remainder of this section we discuss issues related to the reachability sets for each node in each scenario and the separation of Benders cuts (5) for integral and fractional solutions, respectively.

3.1 Determining reachability sets

Reachability sets $R(\omega, i)$ are essential for solving the Benders subproblem and obtaining violated Benders cuts (5). In principle, the reachability sets $R(\omega, i)$ may be pre-computed and stored for all scenarios $\omega \in \Omega$ and nodes $i \in V$. The required memory of $\mathcal{O}(|V|^2|\Omega|)$ may, however, become too large for real-world social

networks with many nodes and when considering a high number of scenarios. Instead of pre-computing and storing sets $R(\omega, i)$ one could alternatively determine them on-the-fly during the separation of Benders cuts (5). The resulting runtime of $\mathcal{O}((|V| + |E|)|V||\Omega|)$ per call of the separation routine might be too time consuming. As a consequence we use an intermediate approach based on external parameter `MemLimPerScen` which denotes the maximally allowed memory consumption per scenario. More precisely, we pre-compute and store reachability sets for a subset of nodes $V^\omega \subseteq V$ in each scenario $\omega \in \Omega$ such that the required memory does not exceed `MemLimPerScen`. Sets $R(\omega, i)$ for remaining nodes $i \in V \setminus V^\omega$ are determined on-the-fly during the cut separation.

For each scenario $\omega \in \Omega$ and node $i \in V$, set $R(\omega, i)$ is determined using reverse BFS. If there is enough memory available for the current scenario, $R(\omega, i)$ is stored for later use and the next node is considered. Otherwise, the current set $R(\omega, i)$ is discarded and the algorithm continues with the next scenario. Two further enhancements are considered in our implementation that help reducing the runtime. The first one is based on the observation that a reverse BFS on G^ω for some node $i \in V$ may encounter a node $j \in V^\omega$ for which $R(\omega, j)$ has already been computed and stored. Since $R(\omega, j) \subseteq R(\omega, i)$ in this case, we add set $R(\omega, j)$ to the (partial) set $R(\omega, i)$ and do not further proceed the reverse BFS from any node in $R(\omega, j)$. The second one is to immediately stop a reverse BFS iteration and discard the (partial) set $R(\omega, i)$ if the memory limit is reached instead of first finishing the computation of this set.

To keep the overall approach as simple and fast as possible set V^ω is determined by simply enumerating through the nodes $i \in V$ in increasing order with respect to their index and computing and storing the associated set $R(\omega, i)$ as long as the memory limit is not reached for the current scenario $\omega \in \Omega$. In Section 4.5 we also test an alternative way of determining V^ω which, however, leads to no improvement.

3.2 Preprocessing

We propose and apply two preprocessing steps that aim to reduce the size of the model and the density of Benders cuts (5), and as a consequence reduce solution times and memory consumption.

P1 The first step eliminates seed set variables $\boldsymbol{z}$ based on the observation that node $i \in V$ is dominated by node $j \in V \setminus \{i\}$, if $j \in R(\omega, i)$ in all scenarios $\omega \in \Omega$. In this case, it is better (or equivalent) to include j in the seed set instead of i since all nodes reached by i can also be reached by j (in all scenarios). Thus, variable z_i can be eliminated from the model. If i and j dominate each other, we can eliminate any of the two associated variables which also reduces symmetries in the solution space.

P2 Observe that the covering constraint (1c) is binding in any optimal solution if $i \in V$ is a node with no incident arcs in A^ω in some scenario $\omega \in \Omega$, i.e., $x_i^\omega = z_i$. Thus, we remove the covering constraint and replace variable x_i^ω by z_i in the objective function for such nodes. As a consequence, variables α_i^ω and β_i^ω do not exist in dual (3). Since node i does not appear in any set $R(\omega, j)$ for $j \neq i$, the coefficient of variable z_i in any Benders cut (5) for scenario ω is zero.

3.3 Separation of Benders cuts

algorithm 1 describes our separation routine of Benders cuts (5) when given a current solution $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{\mu}})$ to the LP relaxation of (2). For each scenario $\omega \in \Omega$, we first determine the set $R^\omega(S)$ of all nodes that can be reached from set $S = \{i \in V : \bar{z}_i = 1\}$. We set $(\alpha_i^\omega, \beta_i^\omega) = (1, 0)$, $\forall i \in R^\omega(S)$, since there always exists an optimal solution to (3) with these values, cf. section 2.1. Next, we evaluate $\sum_{j \in R(\omega,i)} \bar{z}_j$ for all remaining nodes $i \in V \setminus R^\omega(S)$. If this sum is at least one, we set $(\alpha_i^\omega, \beta_i^\omega) = (1, 0)$, and $(\alpha_i^\omega, \beta_i^\omega) = (0, 1)$ otherwise. Note that the second step is simplified if the current LP solution is integral, i.e., if $\bar{z}_i \in \{0, 1\}$ holds for all $i \in V$. In this case $\sum_{j \in R(\omega,i)} \bar{z}_j$ is equal to zero and thus $(\alpha_i^\omega, \beta_i^\omega) = (0, 1)$ for all nodes $i \in V \setminus R^\omega(S)$. Instead of storing the full solution to the dual of (2), algorithm 1 uses C^ω and c_j^ω, $\forall j \in V$, to keep track of $\sum_{i \in V} \alpha_i^\omega$ and $\sum_{i:j \in R(\omega,i)} \beta_i^\omega$, respectively. Finally, we stop the reverse BFS for node $j \in V \setminus R^\omega(S)$ as soon as the sum of relevant z-values reaches one (further increasing this sum does not change the overall result). For the sake of readability, these improvements are omitted in algorithm 1.

Algorithm 1 `separate`

Require: live-arc graphs $G^\omega = (V, A^\omega), V^\omega, \forall \omega \in \Omega$, solution $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{\mu}})$
Ensure: set $\mathcal{C}$ of Benders cuts (5) violated by $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{\mu}})$
  $S = \{i \in V : \bar{z}_i = 1\}$
  $\mathcal{C} = \emptyset$
  for $\omega \in \Omega$ **do**
    obtain active node set $R^\omega(S)$ in G^ω
    $C^\omega = |R^\omega(S)|$
    $c^\omega_j = 0, \forall j \in V$
    for $i \notin R^\omega(S)$ **do**
      if $\sum_{j \in R(\omega,i)} \bar{z}_j \geq 1$ **then**
        $C^\omega = C^\omega + 1$
      else
        $c^\omega_j = c^\omega_j + 1, \forall j \in R(\omega, i)$
      end if
    end for
    if $\bar{\mu}_\omega > C^\omega + \sum_{j \in V} c^\omega_j \bar{z}_j$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{\mu_\omega \leq C^\omega + \sum_{j \in V} c^\omega_j z_j\}$
    end if
  end for
  return $\mathcal{C}$

Table 1: Real-world social networks

Instance	$\lvert V\rvert$	$\lvert A\rvert$	Description
`MSG`	1 899	59 835	messaging network of users at UC-Irvine [19]
`GNU`	10 879	79 988	snapshot 04 of Gnutella filesharing network [20]
`HEP`	37 154	463 168	high energy physics paper citation network [3]
`ENRON`	36 692	367 662	e-mail communication network from Enron [14]

4 Computational experiments

Each experiment has been performed on a single core of an Intel Xeon E5-2670v2 machine with 2.5 GHz. The algorithms are implemented in C++ and IBM ILOG CPLEX 12.7 with default settings is used as branch-and-cut framework and LP solver. A time limit of 3 600 seconds and, unless otherwise stated, a memory limit of 30 GB has been set for each test run with parameter `MemLimPerScen` set to $8/|\Omega|$ GB.

4.1 Instances

We test our solution algorithms on the four real-world social networks used in [22], see table 1, and process them in the same way to allow a fair comparison: In case the instance graph is undirected, we add two directed arcs (i, j) and (j, i) for each undirected edge $\{i, j\}$. Each arc $(i, j) \in A$ is assigned the same influence probability $p_{ij} = 0.1$. Some instances, e.g., `MSG`, include multiple parallel arcs for some $(i, j) \in A$ which are handled in the following way: When generating a simple live-arc graph G^ω for some scenario $\omega \in \Omega$, for each appearance of $(i, j) \in A$ we decide with probability p_{ij} whether that arc is part of A^ω or not. Alternatively, we may compute a combined probability meaning that at least one of the arcs $(i, j) \in A$ appears in G^ω, i.e., $p'_{ij} = 1 - (1 - p_{ij})^{n_{ij}}$, where n_{ij} corresponds to the number of parallel arcs $(i, j) \in A$. Instances `MSG`, `GNU`, and `ENRON` are available for download from the SNAP database[1], while instance `HEP` can be obtained from the website of one of the authors of [3][2].

Moreover, we generated a set `SW` of five directed small-world graphs [21] with 1000 nodes and rewiring probability 0.3 for each average node out-degree $\delta^+ \in \{8, 16\}$. Influence probabilities p_{ij} for arcs $(i, j) \in$

[1] `https://snap.stanford.edu/data/`, accessed on 2018-10-18
[2] `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/weic-graphdata.zip`, accessed on 2018-10-18

A are either set to a common value in $\{0.01, 0.05, 0.1\}$ (similar as above) or are β-distributed with parameters $(2, 18)$ (mean value 0.1).

4.2 Algorithmic variants and settings

Based on the branch-and-cut algorithm described in Section 3 we compare three variants differing in the Benders cut separation:

- `BEN`: In this setting, we separate cuts only for integer solutions.
- `BENFR`: `BEN` with additional separation of fractional solutions but only in the root node of the branch-and-bound tree.
- `BENF`: `BENFR` with separation of fractional solutions in all branch-and-bound nodes.

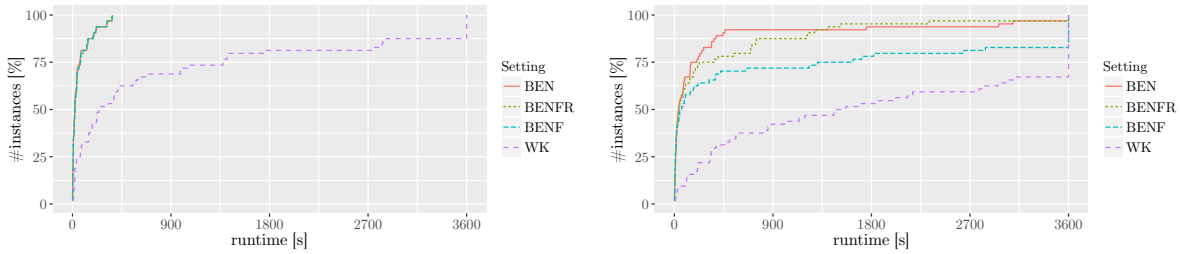
In all settings we initially add submodular cuts (7) for empty sets. In both `BENFR` and `BENF`, we stop separation of fractional cuts in some branch-and-bound node as soon as the absolute decrease of the dual bound compared to the previous iteration falls below 0.001.

4.3 Comparison with state-of-the-art

In this section, we compare our solution algorithm with the delayed row-generation approach based on submodular cuts (6) in [22] which to the best of our knowledge is state-of-the-art for solving the IMP. The authors in [22] iteratively solve their formulation with a subset of submodular cuts to integer optimality, identify and add violated submodular cuts, and update bounds until the global optimality gap is closed. Note, however, that a 1% optimality gap threshold is set for solving the (incomplete) formulation in each iteration. This potentially leads to suboptimal solutions in the end also in cases where the global optimality gap (outer loop) is closed completely. For comparison reasons, we re-implemented their approach to perform tests in which we solve in each iteration the formulation to optimality. Furthermore, the authors in [22] argue that adding submodular cuts (7) for empty sets a priori to the model does not help to (significantly) improve the running time of their approach or reduce the number of generated cuts. This might be explained by the fact that a formulation without any submodular cuts will most probably lead to an optimal solution $\bar{\boldsymbol{z}} = \mathbf{0}$. Thus, the separation in the first iteration will identify and add exactly the submodular cuts (7) for empty sets. Preliminary test showed that both variants proceed quite similarly (within the variability of modern MILP solvers). To avoid this initial iteration we instead immediately add cuts (7). We denote the approach from [22] by `WK` in the following comparison. Tests are performed with the same settings as in [22], i.e., $K \in \{2, 3, 4, 5\}$ and $|\Omega| \in \{100, 200, 300, 500\}$. Figure 1a and fig. 1b give performance profiles comparing the running time of our algorithms with `WK` for 1% and 0% optimality gap threshold, respectively. Note that for our algorithms optimality gap thresholds > 0 apply to the single branch-and-cut execution. As can be clearly seen in the figures, it seems to be beneficial to integrate the generation of Benders cuts within a cutting plane approach in a single branch-and-cut iteration. Our variant `BEN` only separating integer solutions performs best here, immediately followed by `BENFR` adding fractional Benders cuts only in the root node, both outperforming algorithm `WK` by at least one order of magnitude. Detailed results can be found in the Appendix in table 4 and table 5. These instances where the target set size K is relatively small and the influence probability is set to a high value of $p = 0.1$ for all arcs turned out to be quite easy to solve. Because of the high influence probability the live-arc graphs usually contain a giant (weakly) connected component which is also very similar among the scenarios and several quite small or even singleton components. Thus, with a small target set size $K \leq 5$ finding an optimal solution is not too difficult: It seems sufficient to select a promising target node in the giant component to activate most of the other nodes of the giant component and distribute the few residual target nodes to the smaller components improving the objective only slightly.

4.4 Extensive performance comparison

Due to the observations in the last section, we increase the difficulty of the real-world instances by considering also smaller influence probabilities $p \in \{0.01, 0.05, 0.1\}$ (in one particular instance all arcs still have the same probability) and higher target set sizes $K \in \{2, 5, 10, 15, 25\}$. The number of scenarios



(a) 1% optimality gap

(b) 0% optimality gap

Figure 1: Performance profiles of solution runtimes with stopping criterion of 1% and 0% optimality gap, respectively, for the real-world instances tested in [22].

is set to $|\Omega| \in \{250, 500, 750\}$. Smallworld graphs `SW` are tested only for $K \in \{10, 15, 25\}$ but for the same set of influence probabilities with the additional case of β-distributed values to also deal with the (more realistic) situation of varying arc probabilities. Figure 2 and fig. 3 show performance profiles of solution runtimes and optimality gaps grouped by target size K while fig. 4 and fig. 5 show profiles grouped by influence probability p. Detailed results can be found in the appendix in table 6 to table 11.

We immediately observe that enlarging the target set size increases the solution runtime and also the remaining optimality gap in case the time limit is reached. This result is in line with the discussion in the last section and is also quite intuitive since a larger target set size implies a larger set of feasible solutions. With respect to our algorithmic variants, it turns out that when K increases separating fractional solutions gets more and more important, i.e., the performance of `BENF`/`BENFR` significantly improves compared to `BEN`. This behavior can be explained by the fact that the dual bound obtained from the LP relaxation of (2) deteriorates when K increases. This can be clearly observed in the numbers of needed branch-and-bound nodes.

When relating different influence probabilities, interestingly the intermediate case of $p = 0.05$ is more difficult than the other two cases. Mostly, value $p = 0.1$ results in one giant and several very small or singleton components in each scenario (as discussed above) while $p = 0.01$ leads to extremely sparse live-arc graphs containing a large number of singleton components. Both extreme cases are therefore not difficult to solve whereas intermediate values generate many components of different sizes that also differ a lot more between scenarios. This can especially be observed in the extremely large remaining optimality gaps of up to 20% for the `SW` instances with average degree 16, see table 11. The `SW` instances with β-distributed probabilities show quite similar results to the $p = 0.1$ case which can be explained by the fact that the mean value of the chosen distribution corresponds exactly to 0.1.

Increasing the number of scenarios clearly leads to a better approximation of the IMP but also to a much higher time and memory consumption. To summarize, among our algorithmic variants BENF seems to be the most promising one, especially for the `SW` instances, while BENFR with its more conservative and parsimonious separation strategy seems to be better suited for the real-world instances. Common to both winners, our efficient separation of fractional solutions seems to pay off significantly.

4.5 Influence of (partially) storing reachability sets

As discussed in Section 3.1, reachability sets for each node and each scenario (if stored completely) potentially consume a large amount of memory. In this section we evaluate our strategy of partially storing the reachability sets for (tight) memory limits. With the default setting of reserving 8 GB for all reachability sets for an instance we observe that for most considered real-world and small-world instances and numbers of scenarios the limit is not reached and thus all sets could be stored. Table 2 and table 3 show next to preprocessing results (see Section 4.6) also a column named "stored" with the relative average amounts of stored reachability sets (over all scenarios). Only for the largest instance `ENRON` and higher influence probabilities (leading to denser live-arc graphs and potentially larger reachability sets) not all sets can be stored, in some cases even below ten percent.

Therefore, we perform additional tests on instance `ENRON` to compare the solution runtimes of algorithm `BENF` for different amounts of memory reserved for reachability sets controlled via parameter `MemLimPerScen`: (i) `BENF0` with `MemLimPerScen` = 0 (no sets are stored and thus repeatedly deter-

mined on-the-fly), (ii) BENF with MemLimPerScen $= 8/|\Omega|$ (default setting as described above), and (iii) BENF30 with MemLimPerScen $= 30/|\Omega|$ (30 GB are sufficient to store all sets for this instance). The natural question arises which of the reachability sets should be stored and which should be determined on-the-fly in each scenario in case of tight memory limits. In setting BENF for each scenario we iterate through the set of nodes ordered by increasing node index and store the associated reachability sets until MemLimPerScen is reached. To see whether the ordering has an influence on the solution runtime, we alternatively consider setting BENFS sorting the nodes by non-increasing out-degree, thus favoring the storage of potentially large reachability sets. Figure 6 shows performance profiles comparing all four variants. It is easy to see that storing no sets (BENF0) leads to significantly higher solution runtimes even though determining the sets on-the-fly can be done quite efficiently in linear time. Furthermore, reserving huge amounts of memory to store all sets (BENF30) and selecting the sets to store based on some more sophisticated criterion (BENFS), respectively, show no significant benefits compared with our default strategy of partially storing the sets based on index ordering (BENF).

4.6 Preprocessing results

In this section we briefly discuss the influence of preprocessing on our sets of test instances. Table 2 and table 3 show the average numbers of eliminated node variables over all scenarios for real-world and SW instances, respectively.

The requirements for an elimination of a seed set variable ($P1$) are by definition quite hard to fulfill since the according dominance rule has to hold in all scenarios, i.e., the more scenarios the smaller the chance for an elimination. Also, higher graph density is favorable for this preprocessing step which is clearly visible for instance MSG in table 2 in which already the base graph is much denser than in all other considered instances. Increasing influence probabilities lead to a higher elimination rate but we do not observe values above 16%. For the other real-world and especially the SW instances the influence of $P1$ is negligible.

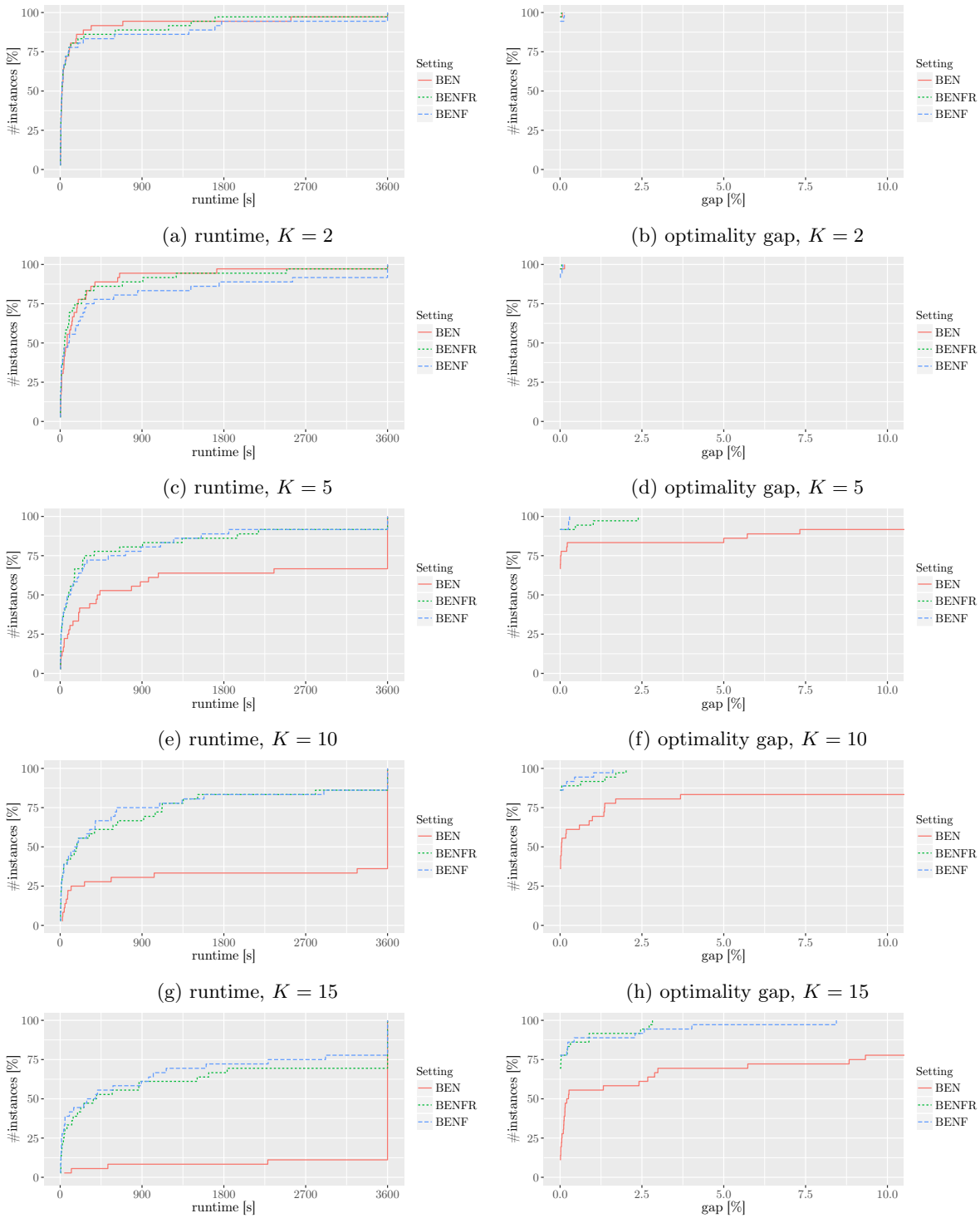
On the contrary, the conditions for preprocessing $P2$ are easier to satisfy, i.e., if a node is not connected in one particular scenario the corresponding activation variable can be eliminated within this scenario. For $P2$, sparse base graphs and small influence probabilities are beneficial increasing the probability of singleton components up to 80% in our instances. Additionally, the nature of $P2$ makes the elimination rate independent of the number of considered scenarios.

5 Conclusions

In this work we solve the influence maximization problem based on the independent cascade model by transforming it to a maximal covering location problem with uncertain facility-customer relations. A Benders reformulation of a compact covering formulation leads to a branch-and-cut approach which outperforms the state-of-the-art algorithm (exploiting submodularity) by orders of magnitude in terms of solution runtime and allows to solve much more difficult instances. In particular, two preprocessing steps together with an efficient separation of fractional Benders cuts significantly decrease the solution runtime. To deal with time and memory issues related to large graphs and a high number of scenarios we apply several algorithmic techniques based on incremental calculation and partial storage of reachability sets.

Table 2: Preprocessing results for real-world instances with relative amounts of stored reachability sets ("stored"), eliminated seed set variables ($P1$), and eliminated activation variables ($P2$), averaged over all scenarios.

name	$\|V\|$	$\|A\|$	$\|\Omega\|$	p	stored	$P1$	$P2$
MSG	1899	59835	250	0.01	1.00	0.01	0.63
				0.05	1.00	0.08	0.45
				0.10	1.00	0.16	0.36
			500	0.01	1.00	0.00	0.63
				0.05	1.00	0.08	0.45
				0.10	1.00	0.15	0.35
			750	0.01	1.00	0.00	0.63
				0.05	1.00	0.07	0.45
				0.10	1.00	0.14	0.35
GNU	10879	79988	250	0.01	1.00	0.00	0.77
				0.05	1.00	0.00	0.52
				0.10	1.00	0.00	0.37
			500	0.01	1.00	0.00	0.77
				0.05	1.00	0.00	0.52
				0.10	1.00	0.00	0.37
			750	0.01	1.00	0.00	0.77
				0.05	1.00	0.00	0.52
				0.10	0.70	0.00	0.37
HEP	15233	117782	250	0.01	1.00	0.00	0.79
				0.05	1.00	0.00	0.59
				0.10	1.00	0.01	0.43
			500	0.01	1.00	0.00	0.79
				0.05	1.00	0.00	0.59
				0.10	1.00	0.01	0.43
			750	0.01	1.00	0.00	0.79
				0.05	1.00	0.00	0.59
				0.10	0.97	0.01	0.43
ENRON	36692	367662	250	0.01	1.00	0.00	0.80
				0.05	0.14	0.01	0.61
				0.10	0.07	0.03	0.46
			500	0.01	1.00	0.00	0.80
				0.05	0.06	0.01	0.61
				0.10	0.04	0.03	0.46
			750	0.01	1.00	0.00	0.80
				0.05	0.04	0.01	0.61
				0.10	0.03	0.02	0.46



(a) runtime, $K = 2$

(b) optimality gap, $K = 2$

(c) runtime, $K = 5$

(d) optimality gap, $K = 5$

(e) runtime, $K = 10$

(f) optimality gap, $K = 10$

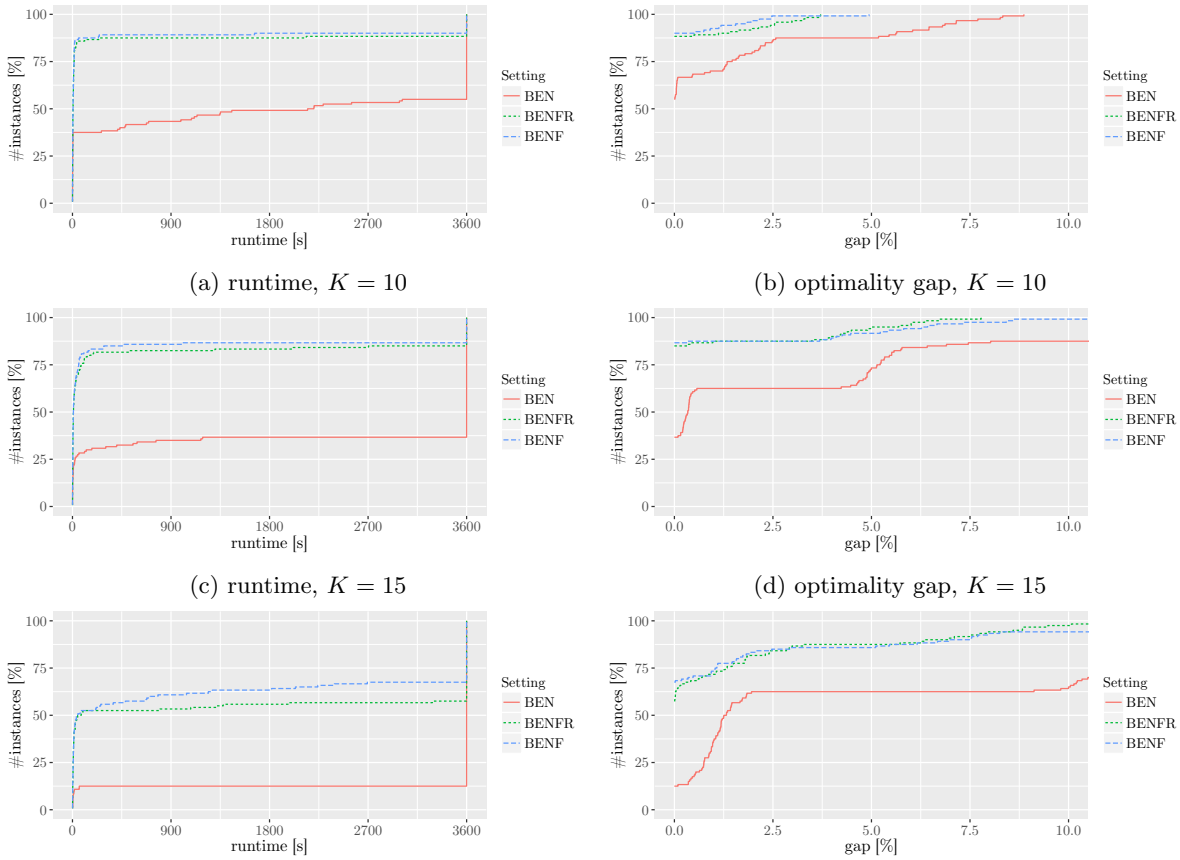
(g) runtime, $K = 15$

(h) optimality gap, $K = 15$

(i) runtime, $K = 25$

(j) optimality gap, $K = 25$

Figure 2: Performance profiles of solution runtimes and optimality gaps on the real-world instances, grouped by target set size K



(a) runtime, $K = 10$

(b) optimality gap, $K = 10$

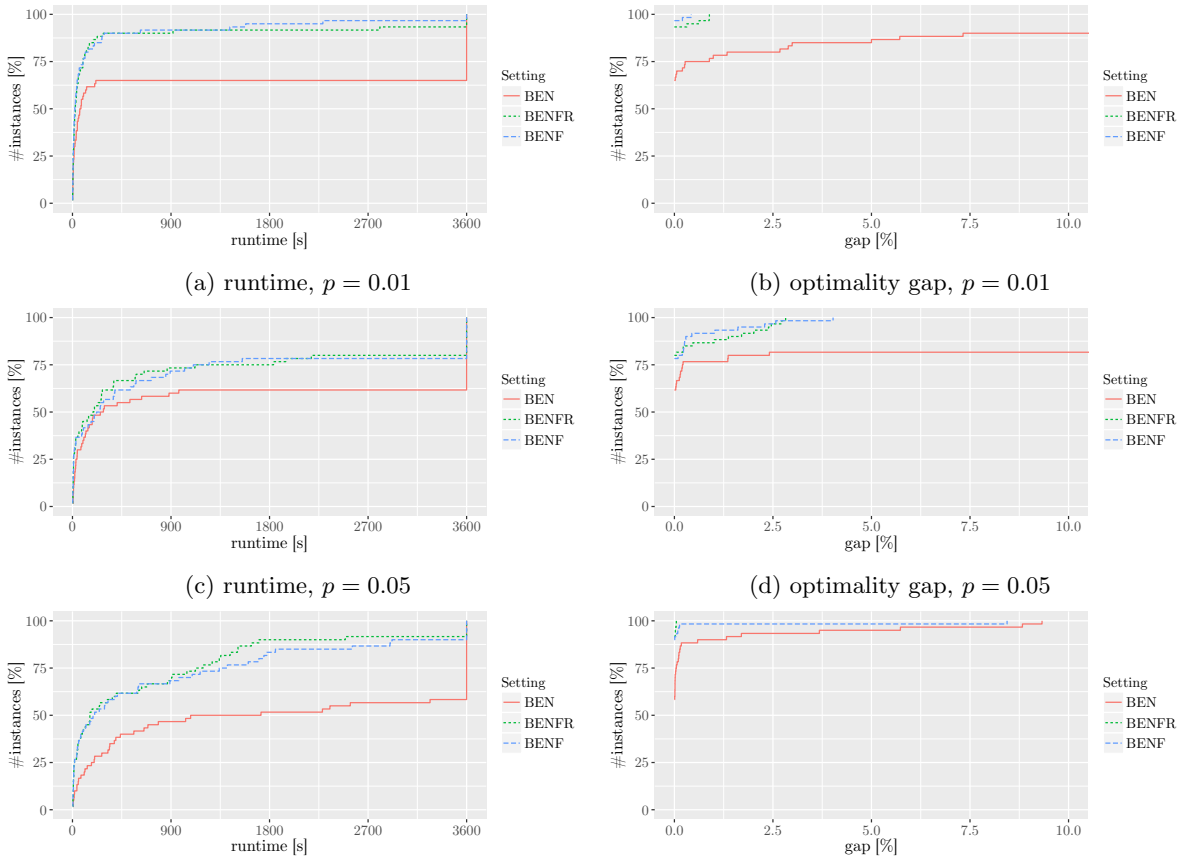
(c) runtime, $K = 15$

(d) optimality gap, $K = 15$

(e) runtime, $K = 25$

(f) optimality gap, $K = 25$

Figure 3: Performance profiles of solution runtimes and optimality gaps on the `SW` instances, grouped by target set size K



(a) runtime, $p = 0.01$

(b) optimality gap, $p = 0.01$

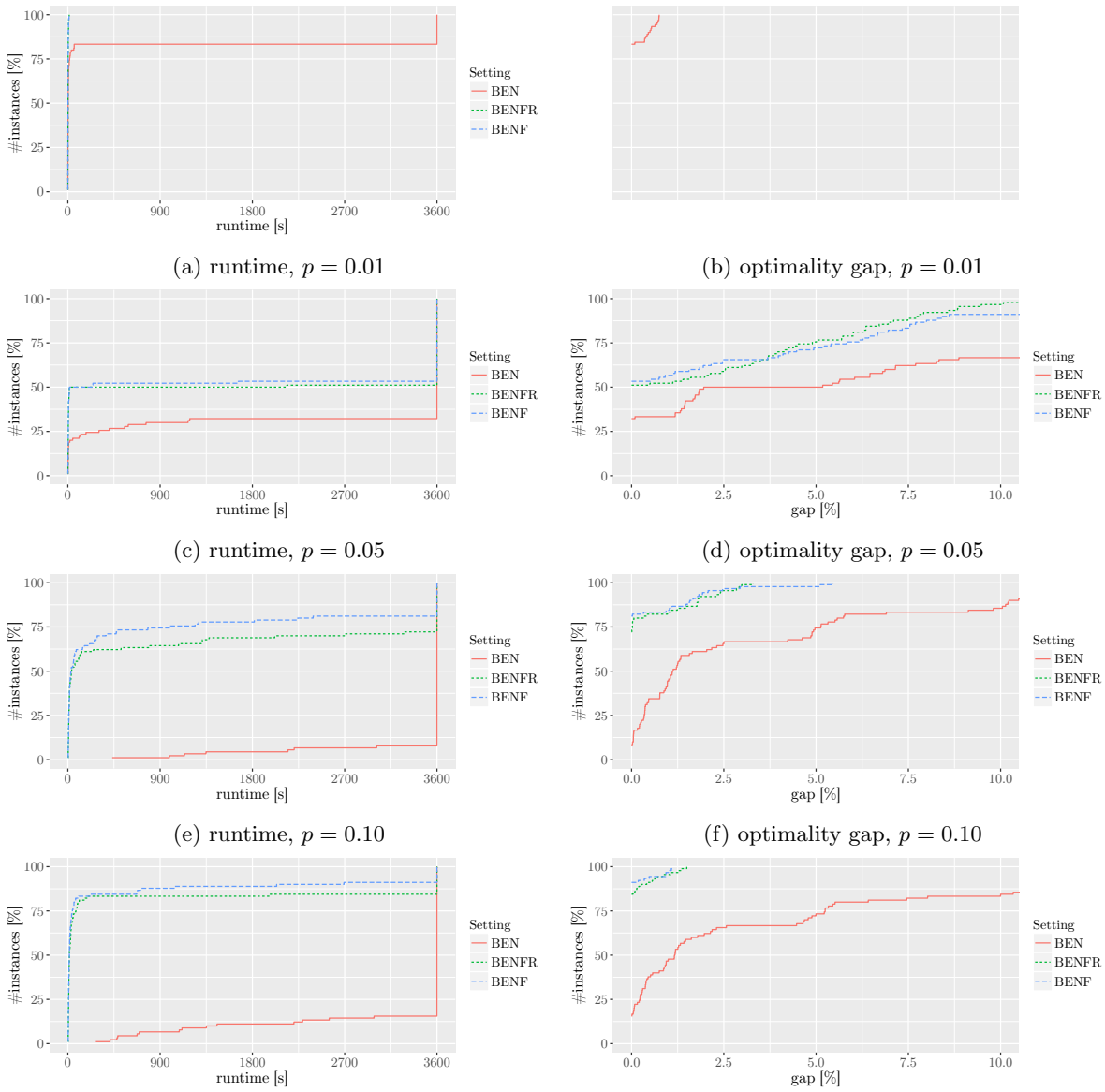
(c) runtime, $p = 0.05$

(d) optimality gap, $p = 0.05$

(e) runtime, $p = 0.10$

(f) optimality gap, $p = 0.10$

Figure 4: Performance profiles of solution runtimes and optimality gaps on the real-world instances, grouped by p



(a) runtime, $p = 0.01$

(b) optimality gap, $p = 0.01$

(c) runtime, $p = 0.05$

(d) optimality gap, $p = 0.05$

(e) runtime, $p = 0.10$

(f) optimality gap, $p = 0.10$

(g) runtime, p β-distributed

(h) optimality gap, p β-distributed

Figure 5: Performance profiles of solution runtimes and optimality gaps on the `SW` instances, grouped by influence probability p

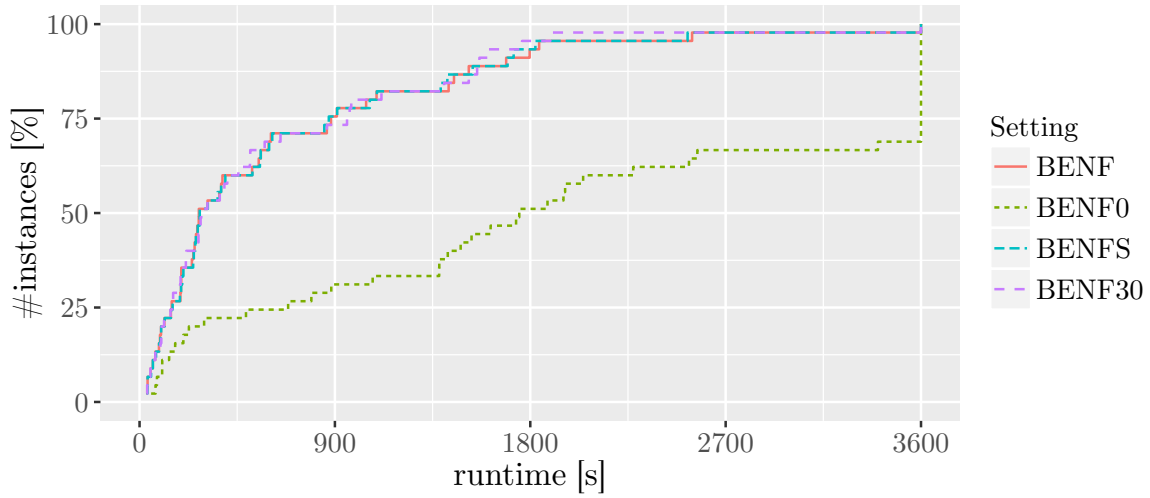


Figure 6: Performance profiles of solution runtimes for different settings of `MemLimPerScen` and sorting strategies.

Table 3: Preprocessing results for `SW` instances with relative amounts of stored reachability sets ("stored"), eliminated seed set variables ($P1$), and eliminated activation variables ($P2$), averaged over all scenarios.

avg. δ^+	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert\Omega\rvert$	p	stored	$P1$	$P2$
8	1000	8000	250	0.01	1.00	0.00	0.72
				0.05	1.00	0.00	0.37
				0.10	1.00	0.00	0.16
				β	1.00	0.00	0.16
			500	0.01	1.00	0.00	0.72
				0.05	1.00	0.00	0.37
				0.10	1.00	0.00	0.16
				β	1.00	0.00	0.16
			750	0.01	1.00	0.00	0.72
				0.05	1.00	0.00	0.37
				0.10	1.00	0.00	0.16
				β	1.00	0.00	0.16
16	1000	16000	250	0.01	1.00	0.00	0.52
				0.05	1.00	0.00	0.14
				0.10	1.00	0.00	0.02
				β	1.00	0.00	0.02
			500	0.01	1.00	0.00	0.52
				0.05	1.00	0.00	0.14
				0.10	1.00	0.00	0.02
				β	1.00	0.00	0.02
			750	0.01	1.00	0.00	0.52
				0.05	1.00	0.00	0.14
				0.10	1.00	0.00	0.02
				β	1.00	0.00	0.02

References

[1] S. Bharathi, D. Kempe, and M. Salek, *Competitive influence maximization in social networks*, in International Workshop on Web and Internet Economics, Springer, 2007, pp. 306–311.

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

[3] W. Chen, Y. Wang, and S. Yang, *Efficient influence maximization in social networks*, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 199–208.

[4] R. Church and C. ReVelle, *The maximal covering location problem*, in Papers of the Regional Science Association, vol. 32, Springer, 1974, pp. 101–118.

[5] J.-F. Cordeau, F. Furini, and I. Ljubic, *Benders decomposition for very large scale partial set covering and maximal covering problems*, tech. report, HEC Montreal, 2018.

[6] P. A. Dreyer, Jr. and F. S. Roberts, *Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion*, Discrete Applied Mathematics, 157 (2009), pp. 1615–1627.

[7] M. Fischetti, M. Kahr, M. Leitner, M. Monaci, and M. Ruthmair, *Least cost influence propagation in (social) networks*, Mathematical Programming, 170 (2018), pp. 293–325.

[8] M. Granovetter, *Threshold models of collective behavior*, The American Journal of Sociology, 83 (1978), pp. 1420–1443.

[9] E. Güney, *On the optimal solution of budgeted influence maximization problem in social networks*, Operational Research, (2017), pp. 1–15.

[10] E. Güney, *An efficient linear programming based method for the influence maximization problem in social networks*, Tech. Report TRIE1801, MEF University, Department of Industrial Engineering, Turkey, 2018.

[11] D. Kempe, J. Kleinberg, and É. Tardos, *Maximizing the spread of influence through a social network*, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 137–146.

[12] D. Kempe, J. Kleinberg, and É. Tardos, *Maximizing the spread of influence through a social network*, Theory of Computing, 11 (2015), pp. 105–147.

[13] J. Lee, J. J. Hasenbein, and D. P. Morton, *Optimization of stochastic virus detection in contact networks*, Operations Research Letters, 43 (2015), pp. 59–64.

[14] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, *Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters*, Internet Mathematics, 6 (2009), pp. 29–123.

[15] Y. Li, J. Fan, Y. Wang, and K.-L. Tan, *Influence maximization on social graphs: A survey*, IEEE Transactions on Knowledge and Data Engineering, 30 (2018), pp. 1852–1872.

[16] E. Mossel and S. Roch, *Submodularity of influence in social networks: From local to global*, SIAM Journal on Computing, 39 (2010), pp. 2176–2188.

[17] G. Nemhauser and L. Wolsey, *Maximizing submodular set functions: formulations and analysis of algorithms*, in North-Holland Mathematics Studies, vol. 59, Elsevier, 1981, pp. 279–301.

[18] G. Nemhauser, L. Wolsey, and M. Fisher, *An analysis of approximations for maximizing submodular set functions—I*, Mathematical Programming, 14 (1978), pp. 265–294.

[19] P. Panzarasa, T. Opsahl, and K. M. Carley, *Patterns and dynamics of users' behavior and interaction: Network analysis of an online community*, Journal of the American Society for Information Science and Technology, 60 (2009), pp. 911–932.

[20] M. Ripeanu and I. Foster, *Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems*, in International Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 85–93.

[21] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world'-networks*, Nature, 393 (1998), pp. 440–442.

[22] H.-H. Wu and S. Küçükyavuz, *A two-stage stochastic programming approach for influence maximization in social networks*, Computational Optimization and Applications, 69 (2018), pp. 563–595.

A Detailed experimental results

Table 4: Comparison of solution times in seconds and numbers of added Benders and submodular cuts, respectively, of our approaches and state-of-the-art algorithm WK with a stopping criterion of 1% optimality gap. TL denotes "time limit reached".

			BEN		BENF		BENFR		WK	
name	K	$\|\Omega\|$	$t[s]$	$\#C$	$t[s]$	$\#C$	$t[s]$	$\#C$	$t[s]$	$\#C$
MSG	2	100	**2**	199	**2**	199	**2**	199	8	200
		200	**3**	400	**3**	400	**3**	400	21	400
		300	**4**	600	**4**	600	**4**	600	21	600
		500	**6**	1000	**6**	1000	**6**	1000	37	1000
	3	100	**2**	201	**2**	201	**2**	201	15	208
		200	**3**	400	**3**	400	**3**	400	20	400
		300	**4**	600	**4**	600	**4**	600	22	600
		500	**6**	1000	8	1000	**6**	1000	39	1000
	4	100	**2**	202	**2**	202	**2**	202	11	200
		200	**3**	400	**3**	400	**3**	400	14	400
		300	**4**	608	**4**	608	**4**	608	22	600
		500	**7**	1022	**7**	1022	**7**	1022	36	1000
	5	100	**2**	236	**2**	236	**2**	236	7	200
		200	**3**	400	**3**	400	**3**	400	14	400
		300	**5**	1548	**5**	620	**5**	620	22	600
		500	**6**	1039	**6**	1039	**6**	1039	36	1000
ENRON	2	100	**72**	200	74	200	73	200	1387	200
		200	**130**	400	131	400	132	400	2757	400
		300	196	600	196	600	**195**	600	TL	600
		500	**314**	1000	**314**	1000	**314**	1000	TL	1000
	3	100	**70**	200	72	200	71	200	1374	200
		200	132	400	**131**	400	**131**	400	2833	400
		300	**188**	600	**188**	600	189	600	TL	600
		500	319	1000	316	1000	**311**	1000	TL	1000
	4	100	76	200	77	200	**75**	200	1412	200
		200	**139**	400	142	400	**139**	400	2812	400
		300	213	600	212	600	**211**	600	TL	600
		500	365	1000	362	1000	**355**	1000	TL	1000
	5	100	77	226	**76**	226	78	226	1422	200
		200	144	411	**142**	411	145	411	2870	400
		300	219	600	**217**	600	220	600	TL	600
		500	**362**	1000	**362**	1000	**362**	1000	TL	1000
GNU	2	100	**8**	200	**8**	200	**8**	200	77	200
		200	**15**	400	**15**	400	**15**	400	149	400
		300	**24**	600	**24**	600	**24**	600	223	600
		500	**39**	1000	**39**	1000	40	1000	375	1000
	3	100	**8**	200	**8**	200	**8**	200	73	200
		200	**15**	400	16	400	16	400	148	400
		300	**23**	600	**23**	600	**23**	600	225	600
		500	41	1000	**40**	1000	41	1000	380	1000
	4	100	**8**	200	**8**	200	**8**	200	74	200
		200	**15**	400	16	400	**15**	400	149	400
		300	**23**	600	**23**	600	**23**	600	223	600
		500	**38**	1000	**38**	1000	39	1000	384	1000
	5	100	**9**	206	**9**	206	**9**	206	84	200
		200	20	401	**19**	415	**19**	415	161	400
		300	31	600	31	636	**30**	636	247	600
		500	**52**	1000	54	1052	54	1052	416	1000
HEP	2	100	**7**	200	8	200	8	200	112	200
		200	**14**	400	17	400	15	400	218	400
		300	**23**	600	**23**	600	24	600	326	600
		500	39	1000	**38**	1000	39	1000	553	1000
	3	100	**9**	279	10	282	10	282	179	279
		200	**16**	595	**16**	595	**16**	595	379	595
		300	**23**	893	**23**	893	**23**	893	580	893
		500	43	1491	43	1491	**42**	1491	985	1491
	4	100	9	279	**8**	279	**8**	279	181	279
		200	**20**	595	**20**	595	**20**	595	414	595
		300	**31**	893	33	893	33	893	600	893
		500	**44**	1491	**44**	1491	**44**	1491	1008	1491
	5	100	**13**	320	19	570	20	570	241	322
		200	**26**	803	40	1135	42	1135	676	793
		300	**40**	1198	64	1711	62	1711	1067	1187
		500	**65**	1972	98	2835	101	2835	1753	1996

Table 5: Comparison of solution times in seconds and numbers of added Benders and submodular cuts, respectively, of our approaches and state-of-the-art algorithm WK with a stopping criterion of 0% optimality gap. TL denotes "time limit reached".

			BEN		BENF		BENFR		WK	
name	K	$\|\Omega\|$	$t[s]$	$\#C$	$t[s]$	$\#C$	$t[s]$	$\#C$	$t[s]$	$\#C$
MSG	2	100	**2**	211	**2**	213	**2**	213	8	200
		200	**4**	882	**4**	755	**4**	755	14	400
		300	**5**	1398	**5**	960	**5**	960	22	600
		500	9	1823	**8**	1334	**8**	1334	36	1000
	3	100	**2**	482	**2**	391	**2**	391	11	208
		200	6	2111	**3**	878	4	878	334	3044
		300	8	3053	**5**	1034	**5**	1034	28	632
		500	12	3602	**8**	1705	**8**	1705	2006	7975
	4	100	**2**	352	**2**	283	**2**	283	TL	6830
		200	8	3425	**3**	957	**3**	957	TL	6249
		300	**4**	1333	**4**	889	**4**	889	TL	9745
		500	19	6741	**9**	2523	**9**	2523	TL	9969
	5	100	**2**	666	**2**	398	**2**	398	TL	7709
		200	10	4340	**4**	1192	**4**	1192	TL	5855
		300	18	6087	6	663	**5**	663	TL	9852
		500	16	5705	**8**	2454	**8**	2454	TL	9940
ENRON	2	100	**80**	207	172	208	174	208	1459	207
		200	**146**	407	375	407	366	407	2826	400
		300	**221**	605	TL	605	743	605	TL	600
		500	**340**	1004	1820	1004	1204	1004	TL	1000
	3	100	**77**	223	216	214	212	214	1691	223
		200	**143**	416	1633	426	544	421	3102	416
		300	**207**	617	TL	683	709	614	TL	600
		500	**360**	1011	2645	1026	1215	1011	TL	1000
	4	100	**84**	227	177	225	139	223	1837	233
		200	**146**	416	379	449	401	449	3056	408
		300	**267**	700	TL	732	749	678	TL	600
		500	**462**	1171	TL	1114	1520	1102	TL	1000
	5	100	**82**	239	91	252	91	252	2177	271
		200	150	411	144	411	**143**	411	2958	416
		300	**234**	695	1231	738	693	694	TL	600
		500	**380**	1166	1711	1170	1281	1170	TL	1000
GNU	2	100	29	203	318	204	**25**	204	113	203
		200	**94**	405	1310	403	95	403	505	404
		300	250	616	2841	614	**228**	614	2784	614
		500	2963	1013	TL	1023	**1394**	1023	TL	1000
	3	100	24	223	91	258	**16**	217	111	216
		200	338	431	TL	974	**193**	417	855	423
		300	3092	641	TL	885	**2329**	620	TL	600
		500	TL	1079	TL	1442	TL	1034	TL	1000
	4	100	**12**	216	16	228	14	226	100	209
		200	**37**	448	664	803	50	442	220	426
		300	444	710	TL	1636	**175**	659	1193	663
		500	TL	1175	TL	3326	TL	1107	TL	1000
	5	100	**11**	227	14	228	13	228	211	267
		200	33	517	49	480	**31**	440	367	546
		300	142	782	423	1211	**59**	686	484	706
		500	1753	1362	TL	2927	**698**	1175	TL	1260
HEP	2	100	**8**	200	**8**	200	**8**	200	116	200
		200	16	400	**14**	400	15	400	217	400
		300	**23**	600	**23**	600	**23**	600	334	600
		500	40	1000	**39**	1000	40	1000	561	1000
	3	100	12	279	**10**	282	**10**	282	184	279
		200	18	595	**16**	595	**16**	595	383	595
		300	**23**	893	26	893	**23**	893	587	893
		500	44	1491	44	1491	**41**	1491	1005	1491
	4	100	16	375	**15**	495	**15**	495	337	371
		200	**29**	713	30	919	**29**	919	887	711
		300	**47**	1048	49	1395	49	1395	845	1033
		500	90	1746	82	2287	**79**	2287	1567	1736
	5	100	21	347	21	570	**19**	570	324	347
		200	**32**	806	41	1135	41	1135	1139	814
		300	**55**	1220	65	1711	64	1711	1472	1215
		500	**77**	1992	101	2835	103	2835	2132	1992

Table 6: Detailed results for instance MSG listing solution times in seconds ($t[s]$), remaining optimality gaps in % ($g[\%]$), primal gaps to the best known solution ($p[\%]$), numbers of added Benders cuts ($\#C$), and numbers of branch-and-bound nodes ($\#B$). TL denotes "time limit reached".

			BEN					BENF					BENFR				
K	$\lvert\Omega\rvert$	p	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$
2	250	0.01	**2**	0.00	0.00	577	7	**2**	0.00	0.00	569	0	**2**	0.00	0.00	569	0
		0.05	**3**	0.00	0.00	923	68	**3**	0.00	0.00	883	0	**3**	0.00	0.00	883	0
		0.10	**4**	0.00	0.00	1271	44	**4**	0.00	0.00	938	0	**4**	0.00	0.00	938	0
	500	0.01	**3**	0.00	0.00	1237	6	**3**	0.00	0.00	1145	0	**3**	0.00	0.00	1145	0
		0.05	7	0.00	0.00	1787	88	**6**	0.00	0.00	1648	0	**6**	0.00	0.00	1648	0
		0.10	9	0.00	0.00	1823	13	**8**	0.00	0.00	1334	0	**8**	0.00	0.00	1334	0
	750	0.01	7	0.00	0.00	1831	12	**5**	0.00	0.00	1897	3	6	0.00	0.00	1840	6
		0.05	12	0.00	0.00	2495	19	**9**	0.00	0.00	2579	0	**9**	0.00	0.00	2579	0
		0.10	13	0.00	0.00	1892	20	**12**	0.00	0.00	1977	0	**12**	0.00	0.00	1977	0
5	250	0.01	**2**	0.00	0.00	733	11	**2**	0.00	0.00	832	3	**2**	0.00	0.00	832	3
		0.05	7	0.00	0.00	2042	305	**3**	0.00	0.00	974	0	**3**	0.00	0.00	974	0
		0.10	8	0.00	0.00	2711	266	5	0.00	0.00	537	3	**4**	0.00	0.00	537	3
	500	0.01	**3**	0.00	0.00	1532	43	**3**	0.00	0.00	1582	3	**3**	0.00	0.00	1552	7
		0.05	15	0.00	0.00	4285	224	**7**	0.00	0.00	2269	0	**7**	0.00	0.00	2269	0
		0.10	16	0.00	0.00	5705	344	**8**	0.00	0.00	2454	0	**8**	0.00	0.00	2454	0
	750	0.01	**5**	0.00	0.00	2376	23	**5**	0.00	0.00	2283	3	**5**	0.00	0.00	2283	3
		0.05	42	0.00	0.00	8265	439	**10**	0.00	0.00	3321	0	**10**	0.00	0.00	3321	0
		0.10	53	0.00	0.00	12684	533	**12**	0.00	0.00	3785	0	**12**	0.00	0.00	3785	0
10	250	0.01	4	0.00	0.00	1471	238	**2**	0.00	0.00	1194	3	**2**	0.00	0.00	1194	3
		0.05	10	0.00	0.00	2196	321	**4**	0.00	0.00	1279	0	**4**	0.00	0.00	1279	0
		0.10	2352	0.00	0.00	12917	39112	**4**	0.00	0.00	1476	0	**4**	0.00	0.00	1476	0
	500	0.01	6	0.00	0.00	2633	157	**5**	0.00	0.00	2633	3	**5**	0.00	0.00	2633	3
		0.05	21	0.00	0.00	5000	322	**7**	0.00	0.00	2912	0	9	0.00	0.00	2912	0
		0.10	1080	0.00	0.00	23515	7861	**8**	0.00	0.00	3107	0	**8**	0.00	0.00	3107	0
	750	0.01	9	0.00	0.00	4164	132	**6**	0.00	0.00	3345	3	**6**	0.00	0.00	3345	3
		0.05	44	0.00	0.00	8747	365	**10**	0.00	0.00	3447	0	**10**	0.00	0.00	3447	0
		0.10	TL	0.01	0.00	37495	12665	**14**	0.00	0.00	5856	0	**14**	0.00	0.00	5856	0
15	250	0.01	23	0.00	0.00	2682	962	**3**	0.00	0.00	1806	3	**3**	0.00	0.00	1804	14
		0.05	26	0.00	0.00	4036	631	**4**	0.00	0.00	2165	3	**4**	0.00	0.00	2165	3
		0.10	3266	0.00	0.00	12510	50517	5	0.00	0.00	1712	0	**4**	0.00	0.00	1712	0
	500	0.01	52	0.00	0.00	6542	700	**6**	0.00	0.00	3691	3	7	0.00	0.00	3744	17
		0.05	29	0.00	0.00	4761	623	**6**	0.00	0.00	2571	0	**6**	0.00	0.00	2571	0
		0.10	TL	0.02	0.00	31250	14825	**10**	0.00	0.00	4390	0	**10**	0.00	0.00	4390	0
	750	0.01	67	0.00	0.00	8778	657	9	0.00	0.00	4890	3	**8**	0.00	0.00	4890	3
		0.05	85	0.00	0.00	10051	769	11	0.00	0.00	4273	0	**10**	0.00	0.00	4273	0
		0.10	TL	0.02	0.00	40805	9806	**15**	0.00	0.00	6842	0	**15**	0.00	0.00	6842	0
25	250	0.01	TL	0.20	0.03	37334	12038	**5**	0.00	0.00	3038	5	10	0.00	0.00	2965	254
		0.05	45	0.00	0.00	5143	1239	**4**	0.00	0.00	1885	3	**4**	0.00	0.00	1885	3
		0.10	2284	0.00	0.00	9972	56266	**5**	0.00	0.00	2920	3	8	0.00	0.00	2926	119
	500	0.01	TL	0.27	0.03	49217	4147	**10**	0.00	0.00	5946	3	143	0.00	0.00	7221	3515
		0.05	123	0.00	0.00	7061	2289	**8**	0.00	0.00	3171	3	**8**	0.00	0.00	3172	9
		0.10	TL	0.01	0.00	22425	20626	**9**	0.00	0.00	5009	0	**9**	0.00	0.00	5009	0
	750	0.01	TL	0.26	0.03	73208	4253	**14**	0.00	0.00	8775	3	192	0.00	0.00	10001	2179
		0.05	525	0.00	0.00	23548	2429	**13**	0.00	0.00	6568	3	24	0.00	0.00	6111	140
		0.10	TL	0.02	0.00	34562	11786	**14**	0.00	0.00	7163	0	**14**	0.00	0.00	7163	0

Table 7: Detailed results for instance HEP listing solution times in seconds ($t[s]$), remaining optimality gaps in % ($g[\%]$), primal gaps to the best known solution ($p[\%]$), numbers of added Benders cuts ($\#C$), and numbers of branch-and-bound nodes ($\#B$). TL denotes "time limit reached".

			BEN					BENF					BENFR				
K	$\|\Omega\|$	p	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$
2	250	0.01	**8**	0.00	0.00	501	8	**8**	0.00	0.00	671	0	**8**	0.00	0.00	671	0
		0.05	**10**	0.00	0.00	500	0	**10**	0.00	0.00	500	0	**10**	0.00	0.00	500	0
		0.10	**19**	0.00	0.00	500	0	20	0.00	0.00	500	0	**19**	0.00	0.00	500	0
	500	0.01	**15**	0.00	0.00	1004	0	**15**	0.00	0.00	1323	0	**15**	0.00	0.00	1323	0
		0.05	22	0.00	0.00	1000	0	**20**	0.00	0.00	1000	0	**20**	0.00	0.00	1000	0
		0.10	41	0.00	0.00	1000	0	**40**	0.00	0.00	1000	0	41	0.00	0.00	1000	0
	750	0.01	25	0.00	0.00	1973	0	**22**	0.00	0.00	1973	0	**22**	0.00	0.00	1973	0
		0.05	32	0.00	0.00	1500	0	30	0.00	0.00	1500	0	**29**	0.00	0.00	1500	0
		0.10	**56**	0.00	0.00	1500	0	57	0.00	0.00	1500	0	**56**	0.00	0.00	1500	0
5	250	0.01	16	0.00	0.00	1117	171	**14**	0.00	0.00	1180	0	**14**	0.00	0.00	1180	0
		0.05	78	0.00	0.00	999	1301	170	0.00	0.00	1667	261	**50**	0.00	0.00	1316	384
		0.10	**41**	0.00	0.00	1014	169	46	0.00	0.00	1430	0	43	0.00	0.00	1430	0
	500	0.01	34	0.00	0.00	2085	344	28	0.00	0.00	2392	3	**27**	0.00	0.00	2392	3
		0.05	154	0.00	0.00	1956	745	220	0.00	0.00	3016	115	**94**	0.00	0.00	2523	145
		0.10	**78**	0.00	0.00	1992	53	103	0.00	0.00	2835	0	100	0.00	0.00	2835	0
	750	0.01	58	0.00	0.00	3975	388	43	0.00	0.00	3871	3	**42**	0.00	0.00	3746	5
		0.05	**183**	0.00	0.00	2934	366	853	0.00	0.00	6169	323	235	0.00	0.00	3911	175
		0.10	338	0.00	0.00	3566	93	168	0.00	0.00	4760	6	**162**	0.00	0.00	4753	8
10	250	0.01	TL	5.00	1.15	14657	6600	**27**	0.00	0.00	2415	3	**27**	0.00	0.00	2415	3
		0.05	882	0.00	0.00	1951	17213	1551	0.00	0.00	4753	2313	**266**	0.00	0.00	2447	2356
		0.10	397	0.00	0.00	1862	2234	77	0.00	0.00	1997	0	**76**	0.00	0.00	1997	0
	500	0.01	TL	5.72	1.10	24149	3583	**62**	0.00	0.00	4857	3	66	0.00	0.00	4682	7
		0.05	TL	0.22	0.04	3765	23104	TL	0.23	0.08	9819	1102	**2182**	0.00	0.00	5065	10790
		0.10	TL	0.02	0.01	4600	12792	203	0.00	0.00	3546	23	**159**	0.00	0.00	3541	24
	750	0.01	TL	7.32	1.98	28185	2053	115	0.00	0.00	7734	7	**113**	0.00	0.00	7179	21
		0.05	TL	0.20	0.02	4727	11584	TL	0.26	0.06	11821	1306	**1949**	0.00	0.00	7532	7366
		0.10	TL	0.03	0.00	7027	2400	296	0.00	0.00	5959	21	**245**	0.00	0.00	5959	21
15	250	0.01	TL	11.69	2.01	16664	2800	**290**	0.00	0.00	10626	170	919	0.00	0.00	4976	11782
		0.05	TL	0.19	0.01	3233	15700	384	0.00	0.00	4434	415	**201**	0.00	0.00	3276	1012
		0.10	TL	0.18	0.04	3558	9472	158	0.00	0.00	3760	2	**154**	0.00	0.00	3760	2
	500	0.01	TL	13.03	2.77	20195	2300	**621**	0.00	0.00	17789	178	2808	0.00	0.00	8860	14506
		0.05	TL	1.36	0.11	8798	3602	TL	0.09	0.00	12014	1917	**1119**	0.00	0.00	6199	4536
		0.10	TL	3.68	0.14	7204	1635	1341	0.00	0.00	6400	146	**1123**	0.00	0.00	6041	345
	750	0.01	TL	13.06	2.25	27208	1493	**1582**	0.00	0.00	31945	209	TL	0.63	0.10	22459	8240
		0.05	TL	1.36	0.00	9633	2500	TL	0.20	0.04	14743	551	TL	0.05	0.00	9354	3571
		0.10	TL	1.70	0.09	7393	1242	2901	0.00	0.00	9210	334	**1512**	0.00	0.00	8964	377
25	250	0.01	TL	17.59	3.98	14458	2565	**2288**	0.00	0.00	26086	380	TL	0.32	0.04	7271	19741
		0.05	TL	2.41	0.40	13726	1700	**1024**	0.00	0.00	7334	796	1837	0.00	0.00	5173	10245
		0.10	TL	5.73	0.56	9910	1713	411	0.00	0.00	4769	51	**401**	0.00	0.00	4607	261
	500	0.01	TL	18.02	4.03	24625	1325	TL	0.43	0.00	41751	199	TL	0.88	0.25	17098	5577
		0.05	TL	25.12	0.53	16051	1060	TL	0.23	0.00	18464	344	TL	0.22	0.00	10676	2666
		0.10	TL	8.83	0.26	9813	815	**2919**	0.00	0.00	9696	452	TL	0.02	0.01	8867	1800
	750	0.01	TL	17.66	3.31	26941	1115	TL	0.20	0.00	43852	188	TL	0.89	0.26	17486	1990
		0.05	TL	25.04	0.72	17386	766	TL	0.24	0.00	25379	107	TL	0.23	0.00	13336	737
		0.10	TL	9.33	0.83	12972	704	TL	8.44	0.01	17847	110	TL	0.03	0.00	15583	558

Table 8: Detailed results for instance GNU listing solution times in seconds ($t[s]$), remaining optimality gaps in % ($g[\%]$), primal gaps to the best known solution ($p[\%]$), numbers of added Benders cuts ($\#C$), and numbers of branch-and-bound nodes ($\#B$). TL denotes "time limit reached".

			BEN					BENF					BENFR				
K	$\|\Omega\|$	p	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$
2	250	0.01	**4**	0.00	0.00	252	0	**4**	0.00	0.00	252	0	**4**	0.00	0.00	252	0
		0.05	**7**	0.00	0.00	416	0	**7**	0.00	0.00	451	0	**7**	0.00	0.00	451	0
		0.10	**114**	0.00	0.00	507	1089	1775	0.00	0.00	505	1061	118	0.00	0.00	505	1061
	500	0.01	**8**	0.00	0.00	502	0	**8**	0.00	0.00	502	0	**8**	0.00	0.00	502	0
		0.05	19	0.00	0.00	950	0	**17**	0.00	0.00	965	0	**17**	0.00	0.00	965	0
		0.10	2538	0.00	0.00	1013	4422	TL	0.11	0.04	1023	553	**1705**	0.00	0.00	1023	3101
	750	0.01	14	0.00	0.00	753	0	**12**	0.00	0.00	753	0	**12**	0.00	0.00	753	0
		0.05	31	0.00	0.00	1357	6	32	0.00	0.00	1605	2	**30**	0.00	0.00	1605	2
		0.10	TL	0.07	0.00	1507	3831	TL	0.13	0.04	1521	246	TL	0.05	0.00	1521	2222
5	250	0.01	**5**	0.00	0.00	294	0	**5**	0.00	0.00	305	0	**5**	0.00	0.00	305	0
		0.05	118	0.00	0.00	4660	1181	32	0.00	0.00	2325	19	**28**	0.00	0.00	1955	92
		0.10	135	0.00	0.00	647	1963	589	0.00	0.00	1066	835	**49**	0.00	0.00	596	144
	500	0.01	11	0.00	0.00	650	0	11	0.00	0.00	671	0	**10**	0.00	0.00	671	0
		0.05	292	0.00	0.00	8468	1309	85	0.00	0.00	5624	26	**60**	0.00	0.00	2922	148
		0.10	1722	0.00	0.00	1362	17240	TL	0.06	0.04	2921	2747	**683**	0.00	0.00	1175	13648
	750	0.01	**16**	0.00	0.00	948	0	**16**	0.00	0.00	971	0	**16**	0.00	0.00	971	0
		0.05	632	0.00	0.00	15624	1159	101	0.00	0.00	6581	19	**84**	0.00	0.00	4169	135
		0.10	TL	0.14	0.03	1841	7248	TL	0.09	0.03	3135	832	TL	0.05	0.00	1761	19343
10	250	0.01	45	0.00	0.00	4249	1617	**9**	0.00	0.00	1024	0	**9**	0.00	0.00	1024	0
		0.05	TL	13.95	2.34	14832	2130	**719**	0.00	0.00	14980	227	TL	0.47	0.00	6351	28910
		0.10	108	0.00	0.00	1451	2886	**46**	0.00	0.00	909	4	50	0.00	0.00	928	24
	500	0.01	214	0.00	0.00	12793	3383	**18**	0.00	0.00	2087	0	**18**	0.00	0.00	2087	0
		0.05	TL	17.29	3.45	24249	1228	TL	0.26	0.00	31892	326	TL	2.39	0.54	17196	2785
		0.10	324	0.00	0.00	1775	2079	**74**	0.00	0.00	1479	4	90	0.00	0.00	1513	219
	750	0.01	82	0.00	0.00	9096	962	**23**	0.00	0.00	2173	0	**23**	0.00	0.00	2173	0
		0.05	TL	15.82	2.61	26384	1056	**1249**	0.00	0.00	30295	161	TL	1.01	0.00	13916	9551
		0.10	TL	0.01	0.00	8228	6000	**123**	0.00	0.00	2710	4	159	0.00	0.00	2654	100
15	250	0.01	TL	0.99	0.08	61044	10000	14	0.00	0.00	2232	0	**13**	0.00	0.00	2232	0
		0.05	TL	20.12	3.74	12605	2033	TL	1.03	0.00	24089	249	TL	1.70	0.18	7077	7150
		0.10	TL	0.04	0.00	8616	7987	40	0.00	0.00	1335	7	**37**	0.00	0.00	1288	14
	500	0.01	TL	1.34	0.06	88278	5746	**19**	0.00	0.00	2682	0	20	0.00	0.00	2682	0
		0.05	TL	18.92	3.53	23165	1300	TL	0.44	0.00	33299	210	TL	1.37	0.00	8873	6370
		0.10	TL	0.02	0.00	7255	6500	**91**	0.00	0.00	2344	7	1355	0.00	0.00	3089	4319
	750	0.01	TL	0.89	0.00	104068	6695	29	0.00	0.00	3944	0	**27**	0.00	0.00	3944	0
		0.05	TL	22.40	4.28	22060	947	TL	1.61	0.05	36161	98	TL	2.02	0.00	18632	2240
		0.10	TL	0.59	0.03	26796	1252	188	0.00	0.00	3316	1	**180**	0.00	0.00	3316	1
25	250	0.01	TL	2.99	0.18	27703	1950	21	0.00	0.00	4063	0	**20**	0.00	0.00	4063	0
		0.05	TL	23.14	2.86	13295	1700	TL	2.57	0.00	25571	120	TL	2.82	0.03	11162	2527
		0.10	TL	0.15	0.03	11431	3804	**53**	0.00	0.00	2069	10	75	0.00	0.00	2112	379
	500	0.01	TL	2.68	0.08	42336	1368	39	0.00	0.00	7130	0	**36**	0.00	0.00	7130	0
		0.05	TL	25.03	3.82	20748	1000	TL	4.03	1.29	27665	66	TL	2.72	0.00	15870	972
		0.10	TL	1.32	0.14	21083	1465	**293**	0.00	0.00	6050	39	1634	0.00	0.00	5981	4818
	750	0.01	TL	2.89	0.34	49376	980	46	0.00	0.00	9722	0	**43**	0.00	0.00	9722	0
		0.05	TL	29.39	5.00	23495	710	TL	2.29	0.00	33577	47	TL	2.46	0.00	15587	430
		0.10	TL	0.11	0.01	22593	1700	**1167**	0.00	0.00	7449	158	TL	0.03	0.01	16409	4734

Table 9: Detailed results for instance ENRON listing solution times in seconds ($t[s]$), remaining optimality gaps in % ($g[\%]$), primal gaps to the best known solution ($p[\%]$), numbers of added Benders cuts ($\#C$), and numbers of branch-and-bound nodes ($\#B$). TL denotes "time limit reached".

			BEN					BENF					BENFR				
K	$\|\Omega\|$	p	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$p[\%]$	$\#C$	$\#B$
2	250	0.01	35	0.00	0.00	500	0	**32**	0.00	0.00	500	0	**32**	0.00	0.00	500	0
		0.05	96	0.00	0.00	504	0	**90**	0.00	0.00	504	0	**90**	0.00	0.00	504	0
		0.10	**174**	0.00	0.00	504	31	599	0.00	0.00	504	4	606	0.00	0.00	504	4
	500	0.01	66	0.00	0.00	1001	0	**61**	0.00	0.00	1001	0	**61**	0.00	0.00	1001	0
		0.05	**180**	0.00	0.00	1003	0	196	0.00	0.00	1003	0	196	0.00	0.00	1003	0
		0.10	**341**	0.00	0.00	1004	18	1703	0.00	0.00	1004	9	1193	0.00	0.00	1004	9
	750	0.01	102	0.00	0.00	1501	0	96	0.00	0.00	1501	0	**95**	0.00	0.00	1501	0
		0.05	255	0.00	0.00	1508	0	**254**	0.00	0.00	1508	0	261	0.00	0.00	1508	0
		0.10	**689**	0.00	0.00	1503	59	1416	0.00	0.00	1503	0	1441	0.00	0.00	1503	0
5	250	0.01	**45**	0.00	0.00	524	244	201	0.00	0.00	668	214	54	0.00	0.00	550	186
		0.05	**105**	0.00	0.00	648	27	252	0.00	0.00	662	16	134	0.00	0.00	653	16
		0.10	**199**	0.00	0.00	606	82	2557	0.00	0.00	649	38	909	0.00	0.00	591	36
	500	0.01	**78**	0.00	0.00	1030	172	270	0.00	0.00	1196	114	99	0.00	0.00	1128	96
		0.05	**188**	0.00	0.00	1286	4	285	0.00	0.00	1454	3	274	0.00	0.00	1454	3
		0.10	**383**	0.00	0.00	1166	12	1749	0.00	0.00	1170	10	1276	0.00	0.00	1170	10
	750	0.01	**129**	0.00	0.00	1529	198	1437	0.00	0.00	2263	524	282	0.00	0.00	1673	192
		0.05	**283**	0.00	0.00	1925	4	374	0.00	0.00	1926	3	372	0.00	0.00	1926	3
		0.10	**654**	0.00	0.00	1674	47	TL	0.01	0.00	2105	5	2492	0.00	0.00	2096	32
10	250	0.01	**35**	0.00	0.00	604	25	36	0.00	0.00	628	0	37	0.00	0.00	628	0
		0.05	**141**	0.00	0.00	1295	153	143	0.00	0.00	1252	0	148	0.00	0.00	1252	0
		0.10	**202**	0.00	0.00	724	27	245	0.00	0.00	847	0	249	0.00	0.00	847	0
	500	0.01	**95**	0.00	0.00	1309	173	190	0.00	0.00	1810	95	96	0.00	0.00	1306	96
		0.05	409	0.00	0.00	3446	345	529	0.00	0.00	2561	26	**375**	0.00	0.00	2436	25
		0.10	**438**	0.00	0.00	1617	90	890	0.00	0.00	2069	4	906	0.00	0.00	2069	4
	750	0.01	203	0.00	0.00	2308	394	265	0.00	0.00	2596	87	**155**	0.00	0.00	2098	231
		0.05	972	0.00	0.00	5112	718	1104	0.00	0.00	4184	25	**655**	0.00	0.00	3655	97
		0.10	**784**	0.00	0.00	2774	124	1854	0.00	0.00	3442	26	1342	0.00	0.00	3321	24
15	250	0.01	45	0.00	0.00	1325	286	41	0.00	0.00	944	0	**39**	0.00	0.00	944	0
		0.05	TL	0.05	0.00	10453	2135	200	0.00	0.00	2294	10	**185**	0.00	0.00	2294	10
		0.10	**268**	0.00	0.00	1674	1007	325	0.00	0.00	1723	4	319	0.00	0.00	1723	4
	500	0.01	82	0.00	0.00	1938	96	80	0.00	0.00	1649	0	**73**	0.00	0.00	1649	0
		0.05	TL	0.06	0.02	16511	849	388	0.00	0.00	3724	4	**378**	0.00	0.00	3724	4
		0.10	**561**	0.00	0.00	3064	1302	604	0.00	0.00	3361	4	633	0.00	0.00	3361	4
	750	0.01	120	0.00	0.00	2462	94	118	0.00	0.00	2652	0	**114**	0.00	0.00	2652	0
		0.05	TL	0.04	0.01	17057	750	**561**	0.00	0.00	5273	3	582	0.00	0.00	5273	3
		0.10	**1034**	0.00	0.00	4654	1169	1094	0.00	0.00	5218	0	1046	0.00	0.00	5218	0
25	250	0.01	TL	0.02	0.00	12897	7861	53	0.00	0.00	1943	3	**52**	0.00	0.00	1890	14
		0.05	TL	0.16	0.04	15228	1112	**259**	0.00	0.00	3906	3	263	0.00	0.00	3902	5
		0.10	TL	0.05	0.02	9250	1538	389	0.00	0.00	3343	6	**364**	0.00	0.00	3255	13
	500	0.01	TL	0.04	0.01	26741	2571	**102**	0.00	0.00	3911	3	128	0.00	0.00	4297	167
		0.05	TL	0.15	0.04	17602	690	581	0.00	0.00	8014	5	**573**	0.00	0.00	8014	5
		0.10	TL	0.12	0.03	15781	458	973	0.00	0.00	5689	9	**868**	0.00	0.00	5464	13
	750	0.01	TL	0.06	0.01	25731	1201	**149**	0.00	0.00	6021	3	226	0.00	0.00	5817	210
		0.05	TL	0.12	0.02	18687	386	892	0.00	0.00	11451	3	**865**	0.00	0.00	11451	3
		0.10	TL	0.10	0.01	20942	245	1606	0.00	0.00	10033	3	**1507**	0.00	0.00	10030	6

Table 10: Detailed results for SW instances with average node degree 8, listing avg. solution times in seconds ($\bar{t}[s]$), avg. remaining optimality gaps in % ($\bar{g}[\%]$), avg. numbers of added Benders cuts ($\#C$), and avg. numbers of branch-and-bound nodes ($\#B$). Each row corresponds to a set of five instances. TL denotes "time limit reached".

			BEN				BENF				BENFR			
K	$\|\Omega\|$	p	$t[s]$	$g[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$\#C$	$\#B$
10	250	β	2088	0.75	17362	12980	**4**	0.00	1870	3	5	0.00	1834	5
		0.01	**1**	0.00	260	0	**1**	0.00	260	0	**1**	0.00	260	0
		0.05	2	0.00	514	79	**1**	0.00	509	0	**1**	0.00	509	0
		0.10	TL	1.58	22919	14910	**6**	0.00	1953	12	**6**	0.00	1768	37
	500	β	3338	1.22	37528	11254	**7**	0.00	3553	3	8	0.00	3504	15
		0.01	**1**	0.00	553	0	**1**	0.00	553	0	**1**	0.00	553	0
		0.05	**3**	0.00	895	152	**3**	0.00	956	0	**3**	0.00	956	0
		0.10	TL	1.35	34495	11105	**11**	0.00	3516	9	13	0.00	3387	83
	750	β	3390	1.58	54354	6628	14	0.00	5160	8	**13**	0.00	4880	16
		0.01	**2**	0.00	802	0	**2**	0.00	802	0	**2**	0.00	802	0
		0.05	5	0.00	1299	102	**4**	0.00	1392	0	**4**	0.00	1392	0
		0.10	TL	1.73	47851	7401	**25**	0.00	5293	31	64	0.00	5150	881
15	250	β	TL	5.48	35686	6399	**19**	0.00	3576	19	46	0.00	2879	793
		0.01	**1**	0.00	302	4	**1**	0.00	311	1	**1**	0.00	311	1
		0.05	109	0.00	2957	8148	**2**	0.00	1138	0	**2**	0.00	1138	0
		0.10	TL	5.51	30340	6776	**108**	0.00	6403	128	692	0.00	4020	8926
	500	β	TL	5.20	48603	3657	**35**	0.00	6386	21	52	0.00	4861	238
		0.01	**2**	0.00	596	1	**2**	0.00	632	0	**2**	0.00	632	0
		0.05	1134	0.02	7859	24863	**4**	0.00	1920	0	**4**	0.00	1920	0
		0.10	TL	5.11	39002	5118	**773**	0.07	13543	251	808	0.20	8723	2426
	750	β	TL	5.80	59317	2968	**52**	0.00	8622	19	70	0.00	7063	134
		0.01	**2**	0.00	917	5	**2**	0.00	950	0	**2**	0.00	950	0
		0.05	572	0.00	9332	15772	**6**	0.00	2816	0	7	0.00	2816	0
		0.10	TL	5.08	52384	3834	**393**	0.00	15561	151	2106	0.15	12343	9168
25	250	β	TL	11.39	34219	5361	**1580**	0.22	22268	211	3275	0.35	6756	20077
		0.01	4	0.00	698	875	**1**	0.00	616	0	**1**	0.00	616	0
		0.05	TL	1.60	38307	6443	**5**	0.00	2844	3	**5**	0.00	2831	8
		0.10	TL	10.86	28183	6193	TL	1.28	35681	279	TL	1.58	11031	11678
	500	β	TL	11.33	49545	3480	**2702**	0.46	36342	187	TL	0.79	14666	7572
		0.01	20	0.00	1323	4288	**2**	0.00	1108	0	**2**	0.00	1108	0
		0.05	TL	1.57	49813	4110	**9**	0.00	5257	1	10	0.00	5257	1
		0.10	TL	10.03	36043	3930	TL	2.42	38417	167	TL	2.22	19619	5291
	750	β	TL	10.87	56315	2829	**3089**	0.53	42854	141	TL	0.67	16720	3708
		0.01	19	0.00	1962	1922	**3**	0.00	1549	1	**3**	0.00	1549	1
		0.05	TL	1.50	57155	3158	**13**	0.00	6885	1	**13**	0.00	6885	1
		0.10	TL	10.56	45887	2797	TL	2.86	41729	116	TL	2.59	27502	2869

Table 11: Detailed results for SW instances with average node degree 16, listing avg. solution times in seconds ($\bar{t}[s]$), avg. remaining optimality gaps in % ($\bar{g}[\%]$), avg. numbers of added Benders cuts ($\#C$), and avg. numbers of branch-and-bound nodes ($\#B$). Each row corresponds to a set of five instances. TL denotes "time limit reached".

			BEN				BENF				BENFR			
K	$\lvert\Omega\rvert$	p	$t[s]$	$g[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$\#C$	$\#B$	$t[s]$	$g[\%]$	$\#C$	$\#B$
10	250	β	926	0.00	6804	11744	**5**	0.00	2057	6	**5**	0.00	2058	41
		0.01	**1**	0.00	370	139	**1**	0.00	357	0	**1**	0.00	357	0
		0.05	TL	6.04	21073	10969	**1869**	0.41	22290	640	3308	1.23	9330	30518
		0.10	1556	0.00	5401	29261	**4**	0.00	1729	1	**4**	0.00	1729	1
	500	β	2068	0.02	12012	12878	**9**	0.00	4190	4	14	0.00	4147	121
		0.01	**2**	0.00	620	22	**2**	0.00	647	0	**2**	0.00	647	0
		0.05	TL	6.68	28225	7110	TL	1.31	37959	711	TL	2.26	15841	14208
		0.10	2859	0.03	12648	18161	**10**	0.00	3719	6	**10**	0.00	3685	16
	750	β	3322	0.05	24674	9721	**16**	0.00	6643	5	45	0.00	6980	325
		0.01	**3**	0.00	935	34	**3**	0.00	956	0	**3**	0.00	956	0
		0.05	TL	7.68	34777	4488	TL	2.66	44292	358	TL	3.16	25780	6934
		0.10	TL	0.05	19238	14090	**17**	0.00	5029	10	18	0.00	4934	59
15	250	β	TL	0.22	17116	18145	**5**	0.00	2162	5	**5**	0.00	2139	13
		0.01	17	0.00	1026	2915	**2**	0.00	609	0	**2**	0.00	609	0
		0.05	TL	12.72	18115	7661	TL	4.10	35572	275	TL	3.94	11155	11757
		0.10	TL	0.22	14295	20434	**6**	0.00	2052	8	**6**	0.00	2011	15
	500	β	TL	0.33	29932	8375	**9**	0.00	3933	4	10	0.00	3912	9
		0.01	8	0.00	1364	1249	**3**	0.00	1083	0	**3**	0.00	1083	0
		0.05	TL	12.39	25268	4840	TL	6.48	32090	97	TL	5.23	17107	4791
		0.10	TL	0.36	26515	9682	**21**	0.00	4291	44	22	0.00	3804	278
	750	β	TL	0.44	38202	5862	**17**	0.00	5976	13	18	0.00	5772	57
		0.01	20	0.00	2039	3422	**4**	0.00	1532	0	**4**	0.00	1532	0
		0.05	TL	12.68	32180	3127	TL	8.06	33507	71	TL	6.24	21023	2854
		0.10	TL	0.38	32407	6966	**36**	0.00	6411	53	41	0.00	5602	325
25	250	β	TL	0.89	23954	11487	**11**	0.00	3095	23	14	0.00	2900	115
		0.01	TL	0.50	23431	42032	**3**	0.00	1840	1	**3**	0.00	1838	1
		0.05	TL	19.82	17887	5471	TL	7.12	26599	52	TL	6.64	11878	4151
		0.10	TL	0.88	17708	11992	**312**	0.00	6383	620	1014	0.01	3188	21827
	500	β	TL	1.13	30181	5443	**29**	0.00	5647	26	40	0.00	5222	165
		0.01	TL	0.60	37009	8901	**7**	0.00	3210	7	**7**	0.00	3111	19
		0.05	TL	20.52	25550	3028	TL	15.28	24881	27	TL	8.73	19012	1752
		0.10	TL	1.02	26661	4891	**835**	0.00	12438	1074	2520	0.01	5355	26615
	750	β	TL	1.21	37604	3450	**48**	0.00	8044	27	57	0.00	7258	125
		0.01	TL	0.48	40315	8274	9	0.00	4217	4	**8**	0.00	4140	7
		0.05	TL	20.55	31485	2226	TL	20.24	23668	16	TL	9.38	19664	512
		0.10	TL	1.13	33723	2671	**1853**	0.01	18360	1132	3097	0.03	8083	16205