Efficient presolving methods for the influence maximization problem in social networks

Sheng-Jie Chen[1], Wei-Kun Chen[2], Yu-Hong Dai[1], Jian-Hua Yuan[3], Hou-Shan Zhang[3]

May 23, 2022

Abstract

We consider the influence maximization problem (IMP) which asks for identifying a limited number of key individuals to spread influence in a social network such that the expected number of influenced individuals is maximized. The stochastic maximal covering location problem (SMCLP) formulation is a mixed integer programming formulation that effectively approximates the IMP by the Monte-Carlo sampling. For IMPs with a large-scale network or a large number of samplings, however, the SMCLP formulation cannot be efficiently solved by existing exact algorithms due to its large problem size. In this paper, we concentrate on deriving presolving methods to reduce the problem size and hence enhance the capability of employing exact algorithms in solving large-scale IMPs. In particular, we propose two effective presolving methods, called strongly connected nodes aggregation (SCNA) and isomorphic nodes aggregation (INA), respectively. The SCNA enables to build a new SMCLP formulation that is potentially much more compact than the existing one, and the INA further eliminates variables and constraints in the SMCLP formulation. A theoretical analysis on two special cases of the IMP is provided to demonstrate the strength of the SCNA and INA in reducing the problem size of the SMCLP formulation. Finally, we integrate the proposed presolving methods, SCNA and INA, into the Benders decomposition algorithm, which is recognized as one of the state-of-the-art exact algorithms for solving the IMP. Numerical results demonstrate that with the SCNA and INA, the Benders decomposition algorithm is much more effective in solving the IMP in terms of solution time.

Keywords Influence maximization · Integer programming · Presolving methods · Benders decomposition · Social network · Stochastic programming

Mathematics Subject Classification 90C10 · 90C15

1 Introduction

Nowadays, with the popularity of online social network sites such as Facebook, Instagram, and Twitter, propagation of influence in social networks has received more and more attention. Promotion of products, ideas, and specific behavior patterns can all be viewed as propagation of influence. In practice, influence spreads among individuals through the

[1] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China,* {shengjie_chen,dyh}@lsec.cc.ac.cn

[2] *School of Mathematics and Statistics/Beijing Key Laboratory on MCAACI, Beijing Institute of Technology, Beijing 100081, China,* chenweikun@bit.edu.cn

[3] *School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China,* {jianhuayuan,houshan_zhang}@bupt.edu.cn

so-called "word-of-mouth" exchanges. Individuals with more social connections can be seen as more influential, which means they are more likely to exert influence on others. In this setting, one related optimization problem, called the *influence maximization problem* (IMP), is to select a limited number of key individuals as a seed set, denoted as $\mathcal{S}$, to trigger a spread process in the social network such that the expected number of influenced individuals is maximized after the spread. Mathematically, the IMP can be written as

$$\max_{\mathcal{S}\subseteq\mathcal{V},\ |\mathcal{S}|\leq K} \sigma(\mathcal{S}), \tag{1}$$

where $\mathcal{V}$ is the set of individuals in the social network, $\mathcal{S} \subseteq \mathcal{V}$ ($|\mathcal{S}| \leq K \in \mathbb{Z}_{++}$) is the seed set of key individuals that need to be identified, and $\sigma(\mathcal{S})$ is the influence function measuring the expected number of individuals in the social network that can be influenced by the individuals in the seed set $\mathcal{S}$.

The IMP plays a crucial role in various social network applications, such as viral marketing [11, 18], rumor control [7, 27], and network monitoring [35]. Online social network tools enable to collect a huge number of individuals and a huge amount of information about the network structures, and hence provide good opportunities to address these problems. However, they also lead to large-scale social networks (with millions/billions of nodes and arcs), presenting new challenges to solve large-scale IMPs. Therefore, developing efficient algorithms to obtain a high-quality solution for large-scale IMPs is greatly needed.

1.1 Literature review

Kempe et al. [31] first proposed the discrete optimization problem formulation (1) for the IMP. Depending on different influence diffusion processes, they introduced two fundamental influence propagation models for the IMP: the *independent cascade model* (ICM) and the *linear threshold model* (LTM). They showed that under both the ICM and LTM, the IMP is NP-hard, indicating that achieving an optimal solution of the IMP is challenging for the large-scale cases. By proving that the influence function $\sigma(\mathcal{S})$ is monotone and submodular, they were able to design a greedy algorithm, which starts with $\mathcal{S} = \varnothing$ and iteratively adds the individual with maximal marginal gain, with an approximation ratio of $(1 - 1/e)$ (here e denotes the base of the natural logarithm). Unfortunately, as shown in [11, 12], given a fixed seed set $\mathcal{S}$, it is #P-hard to compute $\sigma(\mathcal{S})$ exactly. Therefore, Kempe et al. [31] proposed Monte-Carlo sampling, which provides a subset of equiprobable scenarios, of a reasonable size, to estimate $\sigma(\mathcal{S})$ (each scenario is represented by a *live-arc graph*). Following [31], many researchers focused on improving the greedy algorithm and developing other heuristic algorithms for solving the IMP. Specifically, Leskovec et al. [35] utilized the submodularity of $\sigma(\mathcal{S})$ and presented an improved greedy algorithm called *cost-effective lazy forward*. According to their numerical results, their method is almost 700 times faster than the basic greedy algorithm of Kempe et al. [31]. Chen et al. [10] proposed another algorithmic enhancement for the greedy algorithm which reduces the graph searching time on computing the marginal increment of a given individual. Furthermore, they developed a much more efficient algorithm, called *degree discount*, for the IMP under the ICM that nearly matches the performance of the greedy algorithm. The *two-phase influence maximization* [50] and the *influence maximization via martingales* [51] heuristic algorithms also deserve special attention. They can not only guarantee an approximation ratio of $(1 - 1/e)$, but also enable to solve large-scale IMPs in nearly linear time. We refer to [11, 14, 21, 32] for more greedy or heuristic algorithms for solving the IMP and [40] for a detailed comparison among different heuristic algorithms.

Most heuristic algorithms find a suboptimal solution for the IMP with some worst case guarantees. However, in some applications, it is crucial to identify an optimal

solution instead of just a suboptimal one; see [23]. As a result, using exact algorithms to solve the IMP has attracted more and more attention recently. In most exact algorithms, a *mixed integer programming* (MIP) formulation is established. In particular, Wu and Küçükyavuz [53] transferred the IMP into the so-called *two-stage stochastic submodular MIP model* and proposed a *delayed constraint generation* algorithm to solve the problem to optimality. The computational results indicate that their algorithm is more efficient than the basic greedy algorithm in [31], especially when K is large. Given a collection of scenarios Ω, Güney [23], Güney et al. [24], and Li et al. [39] formulated the IMP as a *stochastic maximal covering location problem* (SMCLP) with $\mathcal{O}(|\mathcal{V}||\Omega|)$ variables and linear constraints. Following this line, Güney et al. [24] developed a reformulation of the SMCLP and proposed a *Benders decomposition* (BD) algorithm. Their experiment results show that the BD algorithm outperforms the one in [53] by several orders of magnitude in terms of solution time. We refer to [20, 25, 29, 30, 45] for employing exact algorithms in solving several variants of the IMP.

However, due to the NP-hardness of the IMP, the above exact algorithms are still inefficient, especially when the size of the social network or the number of scenarios is large. *Presolving* [2] is an appealing strategy to address this issue. It removes redundant information and strengthens the model formulation with the aim of improving the performance of the subsequent solution procedure (e.g., the branch-and-cut or the BD approach). Indeed, presolving has been recognized as a standard routine of the state-of-the-art MIP solvers. For the problems with specific structures, developing customized presolving methods is often much more effective; see [6, 15, 28, 41] for using customized presolving methods to solve various problems. In terms of the IMP, few articles are devoted to designing customized presolving methods. To the best of our knowledge, only two simple presolving methods have been developed in the literature [24, 29] and they have been proved to be beneficial to solving the IMP in certain cases. Consequently, it is crucial to develop more customized presolving methods to further enhance the capability of using exact algorithms to solve large-scale IMPs.

1.2 Contributions and outline

In this paper, we attempt to develop more presolving methods based on the SMCLP formulation to improve the solution efficiency for solving the IMP. The main contributions of this paper are summarized as follows.

- By exploiting the problem structure of the IMP, we propose two new presolving methods including (i) the *strongly connected nodes aggregation* (SCNA) which aggregates nodes in each *strongly connected component* (SCC), in a given live-arc graph, into a single virtual node; and (ii) the *isomorphic nodes aggregation* (INA), which extends the above idea to using *isomorphic* nodes among different live-arc graphs for aggregation (two nodes in different live-arc graphs are called isomorphic if the nodes that can influence them are identical in the corresponding live-arc graphs). We show that the proposed presolving methods can effectively reduce the problem size of the SMCLP formulation. In particular, after applying the SCNA, the IMP can be built on new live-arc graphs obtained by aggregating all SCCs in the original live-arc graphs, leading to a potentially much smaller SMCLP formulation.
- To demonstrate the strength of the proposed SCNA and INA in reducing the problem size of the SMCLP formulation, we provide a theoretical analysis on two special cases of the IMP: one is built on the one-way bipartite social network under the LTM and the other one is built on the complete social network under the ICM. For the first one, we give upper bounds, which are linear with the size of the social network but independent of the number of samplings, for the numbers of variables and constraints in the reduced SMCLP formulation (obtained by applying the proposed presolving

methods). For the second one, we provide a lower bound for the probability that after applying the SCNA and INA, there are only $|\mathcal{V}| + 1$ variables and two linear constraints in the reduced SMCLP formulation. We show that such a probability can tend to one under certain conditions.

- We integrate the SCNA and INA into the BD algorithm [24], which is recognized as one of the state-of-the-art exact algorithms to solve the IMP. We show that the proposed SCNA and INA provide the possibility to develop a much faster separation algorithm for the Benders cuts, as compared with the one in [24].
- Extensive numerical results on real-world social networks demonstrate that (i) the proposed SCNA and INA are quite effective in reducing the problem size of the SMCLP formulation; (ii) when integrating them into the BD algorithm, they can effectively speed up the solution procedure of the IMP.

The remainder of the paper is organized as follows. Section 2 briefly reviews two fundamental influence propagation models (ICM and LTM) and the SMCLP formulation for the IMP. Section 3 presents the SCNA and INA and section 4 further shows their theoretical strength in reducing the problem size of the SMCLP formulation. Section 5 describes the implementation of the proposed presolving methods and the integration of them with the BD algorithm. Section 6 provides the computational results. Section 7 studies a generalization of the IMP and shows that the proposed SCNA and INA can also be applied under some realistic conditions. Finally, section 8 gives some concluding remarks.

2 Propagation models and problem formulation

In this section, we briefly review the propagation models and the SMCLP formulation for the IMP [23, 24, 39]. We use a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ to refer to a social network, in which a node $i \in \mathcal{V}$ represents the individual involved in the influence spread, and an arc $(i, j) \in \mathcal{A}$ represents that individual i has the potential ability to influence (or activate) individual j. The spread of influence in a given social network $\mathcal{G}$ needs to obey certain propagation rules. In [31], Kempe et al. provided the following two fundamental influence propagation models called ICM and LTM.

- In the ICM, each arc $(i, j) \in \mathcal{A}$ is assigned an activation probability π_{ij}. The propagation process starts with a given seed set $\mathcal{S}$. If node i has been activated at the beginning of step t, then during step t, it has a single chance to activate its (inactive) neighbor node j with probability π_{ij} independently. If the activation is unsuccessful, node i has no chance to influence node j any more. Besides, for those nodes that are successfully activated during step t, they will remain active and attempt to activate their inactive neighbor nodes during step $t + 1$. When no more inactive nodes are activated, the diffusion process is terminated.
- In the LTM, each arc $(i, j) \in \mathcal{A}$ is associated with a predefined weight $b_{ij} \geq 0$ satisfying $\sum_{i:(i,j)\in\mathcal{A}} b_{ij} \leq 1$ for all $j \in \mathcal{V}$ and each node $i \in \mathcal{V}$ selects a threshold value θ_i randomly chosen from $[0, 1]$ before the propagation process. Let $\mathcal{S}_t$ denote the activated nodes set at the beginning of step t ($\mathcal{S}_0 := \mathcal{S}$ is the seed set). Then an inactive node $j \in \mathcal{V}$ can be activated during step t if and only if $\sum_{i\in\mathcal{S}_t} b_{ij} \geq \theta_j$, i.e. the total contribution of its active neighbors' influence weights exceeds its threshold value θ_j. In analogy to the ICM, if node j is successfully activated during some step, it will remain active during the following steps, and the entire propagation process stops until no more nodes can be activated.

Given a seed set $\mathcal{S}$, the results (i.e., the distributions of influenced nodes) returned by the ICM and LTM can be different [13]. For a comparison on the performance of the two models in different applications, we refer to [3, 38].

Although the influence spread under the ICM or LTM is a stochastic process, Kempe et al. [31] showed that it can be equivalently converted into a (discrete) deterministic process. More specifically, let Ω be the set of all possible scenarios of influence spread. Each scenario $\omega \in \Omega$ corresponds to a subgraph $\mathcal{G}^\omega = (\mathcal{V}, \mathcal{A}^\omega)$ of $\mathcal{G}$ (called a live-arc graph) with a probability p^ω (satisfying $\sum_{\omega\in\Omega} p^\omega = 1$). Here arc $(i, j) \in \mathcal{A}^\omega$ indicates that in scenario ω, if node i is activated during the influence spread, then node j must be activated by it. Let $\sigma^\omega(\mathcal{S})$ represent the number of activated nodes in $\mathcal{G}^\omega$. Then, $\sigma^\omega(\mathcal{S}) = |\{i \in \mathcal{V} : \text{there exists a directed path from a node in } \mathcal{S} \text{ to node } i \text{ in graph } \mathcal{G}^\omega\}|$. The influence function $\sigma(\mathcal{S})$ can equivalently be calculated by

$$\sigma(\mathcal{S}) = \sum_{\omega\in\Omega} p^\omega \sigma^\omega(\mathcal{S}). \tag{2}$$

We next discuss the computation of the probability p^ω of each scenario ω and the number of scenarios under the ICM or LTM. Under the ICM, to construct a live-arc graph $\mathcal{G}^\omega$, each arc $(i, j) \in \mathcal{A}$ is independently determined to be live with probability π_{ij}. Hence, the probability of $\mathcal{G}^\omega$ is $p^\omega = \prod_{(i,j)\in\mathcal{A}^\omega} \pi_{ij} \prod_{(i,j)\in\mathcal{A}\setminus\mathcal{A}^\omega}(1 - \pi_{ij})$ and the number of all possible live-arc graphs is $2^{|\mathcal{A}|}$. Under the LTM, for each node $j \in \mathcal{V}$, we select at most one incoming arc in $\mathcal{G}$, namely, (i, j), to be live with probability b_{ij}, and do not select any arc with the probability $1 - \sum_{i:(i,j)\in\mathcal{A}} b_{ij}$. As a result, (i) each live-arc graph $\mathcal{G}^\omega$ has a probability $p^\omega = \prod_{j\in\mathcal{V}} I_j$, where $I_j := b_{ij}$ if $(i, j) \in \mathcal{A}^\omega$; and $I_j := 1 - \sum_{i:(i,j)\in\mathcal{A}} b_{ij}$, otherwise; and (ii) the number of all possible live-arc graphs is $\prod_{i\in\mathcal{V}}(n_i + 1)$, where n_i denotes the number of incoming arcs of node i in graph $\mathcal{G}$. It is worthwhile remarking that the stochastic IMP under the ICM or LTM is equivalent to the IMP constructed via a finite number of live-arc graphs (in the sense that the distributions of nodes influenced by a given seed set are equivalent); see Kempe et al. [31].

Given a social network $\mathcal{G}$ and a set of scenarios Ω, we next review the SMCLP formulation for the IMP [23, 24, 39]. First, for each scenario $\omega \in \Omega$ and node $i \in \mathcal{V}$, we denote $\mathcal{R}(\mathcal{G}^\omega, i)$ as the reachability set of nodes that can activate node i in live-arc graph $\mathcal{G}^\omega$ (i.e., $\mathcal{R}(\mathcal{G}^\omega, i) = \{j \in \mathcal{V} : \text{there exists a directed path from node } j \text{ to node } i \text{ in graph } \mathcal{G}^\omega\}$). Then, for each $\omega \in \Omega$ and $i \in \mathcal{V}$, we introduce binary variables y_i and z_i^ω to denote whether node i is selected as a seed node and whether node i can be activated in scenario ω, respectively, i.e.,

$$y_i = \begin{cases} 1, & \text{if node } i \in \mathcal{V} \text{ is chosen as a seed node;} \\ 0, & \text{otherwise;} \end{cases}$$

$$z_i^\omega = \begin{cases} 1, & \text{if node } i \in \mathcal{V} \text{ can be activated in scenario } \omega \in \Omega; \\ 0, & \text{otherwise.} \end{cases}$$

Using the above notations, the authors in [23, 24, 39] formulated the IMP as the following SMCLP:

$$\max_{\boldsymbol{y},\boldsymbol{z}} \sum_{\omega\in\Omega} p^\omega \sum_{i\in\mathcal{V}} z_i^\omega \tag{3a}$$

$$\text{s.t.} \sum_{j\in\mathcal{R}(\mathcal{G}^\omega,i)} y_j \geq z_i^\omega, \qquad \forall\, \omega \in \Omega,\ \forall\, i \in \mathcal{V}, \tag{3b}$$

$$\sum_{j\in\mathcal{V}} y_j \leq K, \tag{3c}$$

$$y_j \in \{0, 1\}, \qquad \forall\, j \in \mathcal{V}, \tag{3d}$$

$$z_i^\omega \in \{0, 1\}, \qquad \forall\, \omega \in \Omega,\ \forall\, i \in \mathcal{V}. \tag{3e}$$

In this formulation, the objective function (3a) maximizes the expected number of influenced nodes in the social network $\mathcal{G}$. Reachability constraints (3b) indicate that if

node i in scenario ω can be activated, then at least one of the nodes in its reachability set $\mathcal{R}(\mathcal{G}^\omega, i)$ is chosen as a seed node. Constraint (3c) limits the cardinality of the set of seed nodes up to K. Finally, constraints (3d) and (3e) restrict variables $\boldsymbol{y}$ and $\boldsymbol{z}$ to be binary.

Unfortunately, formulation (3) is computationally intractable for the network with realistic dimensions due to the huge number of scenarios (for the IMP under the ICM and LTM, the numbers of all possible scenarios are both exponential). Hence, the Monte-Carlo sampling approach is often used to approximate the influence diffusion process in which a reasonable size of equiprobable scenarios set $\Omega' \subseteq \Omega$ is generated, and the objective function in (3a) is replaced by $\sum_{\omega\in\Omega'} p'_\omega \sum_{i\in\mathcal{V}} z_i^\omega$ where $p'_\omega = 1/|\Omega'|$ and hence $\sum_{\omega\in\Omega'} p'_\omega = 1$ [23, 53]. The rationale behind this is that from the approximation result in [33], the probability of obtaining an optimal solution of the IMP (3) by solving the sampling version of the IMP converges to one exponentially fast as $|\Omega'| \to \infty$. In practice, however, the selection of the scenarios size $|\Omega'|$ is crucial for the approximation quality of the sampling version of the IMP. In general, the larger the $|\Omega'|$, the smaller the approximation error is. We refer to [23, subsection 4.6] for an empirical study of the effect of scenarios size on the approximation quality of the sampling version of the IMP. Here we also want to highlight that the problem size of the sampling version of the IMP also grows linearly with the number of scenarios $|\Omega'|$ (as both numbers of variables and constraints are $\mathcal{O}(|\mathcal{V}||\Omega'|)$). This further makes it difficult to solve the problem by standard MIP solvers or the BD approach in [24], especially when the size of graph $\mathcal{G}$ is also large. In the next section, we shall resolve this difficulty by proposing two new presolving methods to reduce the problem size of the SMCLP formulation (3).

In the remaining of this paper, we will consider the sampling version of the IMP. For simplicity of notations, we continue to use Ω and p^ω to represent the set of sampling scenarios and the probability of occurrence of sampling scenario ω, respectively.

3 Two presolving methods

In this section, by exploiting the problem structure of formulation (3), we propose two presolving methods to reduce the problem size of formulation (3). Specifically, subsection 3.1 studies the SCNA which aggregates the nodes in each SCC, in a given live-arc graph, into a single node, and subsection 3.2 investigates the INA which extends the idea of the SCNA to applying isomorphic nodes aggregations among different live-arc graphs.

3.1 Strongly connected nodes aggregation

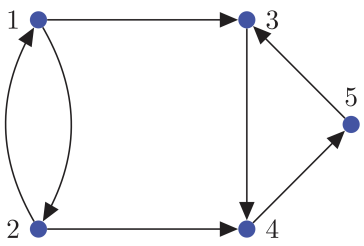

Figure 1: The example live-arc graph which includes two SCCs {1,2} and {3,4,5}.

In this subsection, we present a presolving method by considering the SCCs in a given live-arc graph. To begin with, we consider the example live-arc graph (corresponds to some scenario $\omega \in \Omega$) in Figure 1. In this graph, there exists a directed path (an arc)

from node 1 to node 2, and as a result, if node 1 is activated by some seed node, node 2 can also be activated. Conversely, the fact that there exists a directed path (an arc) from node 2 to node 1 implies that if node 2 is activated by some seed node, node 1 can also be activated. This means that either (i) nodes 1 and 2 are simultaneously activated by some seed node; or (ii) neither of them can be activated. Consequently, $z_1^\omega = z_2^\omega$ must hold in formulation (3). This reveals some redundancy in formulation (3) as it uses two variables z_1^ω and z_2^ω and two constraints in (3b) without considering $z_1^\omega = z_2^\omega$. Indeed, to simplify the problem formulation, we can remove variable z_1^ω and its corresponding reachability constraint in (3b) and add the objective coefficient of variable z_1^ω into that of variable z_2^ω.

In general, for any given two strongly connected nodes $i_1,\ i_2 \in \mathcal{V}$ in a live-arc graph $\mathcal{G}^\omega$ (i.e., there exists a directed path from node i_1 to node i_2 in $\mathcal{G}^\omega$ and vice versa), we can remove one of the two variables and the corresponding reachability constraint in (3b) from formulation (3). Notice that for a given SCC in a live-arc graph, as all of its nodes are strongly connected, we can recursively apply the above argument until there remains only a single variable and a single constraint in (3b) associated with this SCC. This provides us with the following presolving method.

SCNA. For each live-arc graph $\mathcal{G}^\omega$, let $\{\mathcal{SC}_u^\omega\}_{u=1}^{n_\omega}$, $n_\omega \in \mathbb{Z}_{++}$, be all its SCCs. For each SCC $\mathcal{SC}_u^\omega$, variables z_j^ω, $j \in \mathcal{SC}_u^\omega$, are first substituted by a new variable z_u^ω with its objective coefficient being $p^\omega|\mathcal{SC}_u^\omega|$. Then, all but one of constraints in (3b) associated with variable z_u^ω are removed from the SMCLP formulation (3).

To implement the SCNA, we only need to identify all SCCs in all live-arc graphs $\mathcal{G}^\omega$, $\omega \in \Omega$. For each graph $\mathcal{G}^\omega$, this can be done in linear time $\mathcal{O}(|\mathcal{V}| + |\mathcal{A}^\omega|)$ using, e.g., the Kosaraju-Sharir's algorithm [48]. Consequently, the overall complexity to implement the SCNA for formulation (3) is $\mathcal{O}(\sum_{\omega\in\Omega}(|\mathcal{V}| + |\mathcal{A}^\omega|))$.

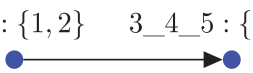

Figure 2: The compact live-arc graph obtained by aggregating each SCC into a single node of the graph in Figure 1. Notice that there are only two nodes 1_2 (corresponds to SCC $\{1, 2\}$) and 3_4_5 (corresponds to SCC $\{3, 4, 5\}$) and one arc between these two nodes.

After applying the SCNA, the IMP can be equivalently constructed based on a new set of live-arc graphs, which are potentially much more compact than the original live-arc graphs. To be more specific, by aggregating each SCC of $\mathcal{G}^\omega$ into a single node with the weight being the size of the SCC, we can get a directed acyclic graph, denoted as $\bar{\mathcal{G}}^\omega = (\bar{\mathcal{V}}^\omega, \bar{\mathcal{A}}^\omega)$. Each node u in $\bar{\mathcal{V}}^\omega$ represents a distinct SCC in the original live-arc graph $\mathcal{G}^\omega$ and each arc $(u, v) \in \bar{\mathcal{A}}^\omega$ denotes that there exists an arc $(i, j) \in \mathcal{A}^\omega$ with $i \in \mathcal{SC}_u^\omega$ and $j \in \mathcal{SC}_v^\omega$ in the original live-arc graph $\mathcal{G}^\omega$ (see Figure 2 for an example of this transformation of the graph in Figure 1). As the reachability sets of the nodes inside a given SCC $\mathcal{SC}_u^\omega$ are identical, we use the notation $\mathcal{R}(\mathcal{G}^\omega, \mathcal{SC}_u^\omega)$ to represent the reachability set of this SCC, which is equal to each $\mathcal{R}(\mathcal{G}^\omega, j)$, $j \in \mathcal{SC}_u^\omega$. It follows immediately that

$$\mathcal{R}(\mathcal{G}^\omega, \mathcal{SC}_u^\omega) = \bigcup_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} \mathcal{SC}_v^\omega, \tag{4}$$

where $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$ is the reachability set of node u in live-arc graph $\bar{\mathcal{G}}^\omega$. Then constraints (3b) reduce to

$$\sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} y(\mathcal{SC}_v^\omega) \geq z_u^\omega,\ \forall\ \omega \in \Omega,\ \forall\ u \in \bar{\mathcal{V}}^\omega, \tag{5}$$

where $y(\mathcal{SC}_v^\omega) := \sum_{j \in \mathcal{SC}_v^\omega} y_j$. Based on the above notations, the reduced SMCLP formulation after applying the SCNA can be written as

$$\begin{aligned} \max_{\boldsymbol{y},\, \boldsymbol{z}} \ & \sum_{\omega \in \Omega} p^\omega \sum_{u \in \bar{\mathcal{V}}^\omega} |\mathcal{SC}_u^\omega| z_u^\omega \\ \text{s.t. } & (3c),\ (3d),\ (5), \\ & z_u^\omega \in \{0,1\}, \qquad\qquad \forall\ \omega \in \Omega,\ \forall\ u \in \bar{\mathcal{V}}^\omega. \end{aligned} \tag{6}$$

It is worthwhile remarking that the numbers of variables z_u^ω and the corresponding reachability constraints in the reduced formulation are equal to $\sum_{\omega \in \Omega} |\bar{\mathcal{V}}^\omega|$ (which is the number of the SCCs in the original live-arc graphs). This can be potentially much smaller than those in formulation (3), especially for the case where the numbers of SCCs are much smaller than the numbers of nodes in the live-arc graphs. As a result, it can be expected that solving formulation (6) is much more efficient than solving formulation (3). In addition, the fact that formulation (6) is built on the (potentially) compact and directed acyclic live-arc graphs provides the possibility to reduce the runtime of graph searching, which plays an important role in improving the performance of the BD algorithm (see section 5 further ahead).

3.2 Isomorphic nodes aggregation

The SCNA performs reductions on two nodes i_1 and i_2 in a given live-arc graph $\mathcal{G}^\omega$ where nodes i_1 and i_2 are strongly connected, or equivalently, the reachability sets of nodes i_1 and i_2 are identical, i.e., $\mathcal{R}(\mathcal{G}^\omega, i_1) = \mathcal{R}(\mathcal{G}^\omega, i_2)$. In this subsection, we concentrate on extending the result to the isomorphic nodes among different live-arc graphs. As it has been previously mentioned, two nodes i_1 and i_2 in two different live-arc graphs $\mathcal{G}^\omega$ and $\mathcal{G}^\eta$ are called isomorphic if their reachability sets are identical, i.e.,

$$\mathcal{R}(\mathcal{G}^\omega, i_1) = \mathcal{R}(\mathcal{G}^\eta, i_2). \tag{7}$$

We begin with the following observation stating that the values of variables $\boldsymbol{z}$ are determined by the values of variables $\boldsymbol{y}$ in formulation (3).

Observation 3.1. *There must exist an optimal solution* $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ *of formulation* (3) *such that*

$$\bar{z}_i^\omega = \min\left\{1, \sum_{j \in \mathcal{R}(\mathcal{G}^\omega, i)} \bar{y}_j\right\},\ \forall\ \omega \in \Omega,\ \forall\ i \in \mathcal{V}. \tag{8}$$

In particular, if node $i \in \mathcal{V}$ does not have any incoming arc in $\mathcal{G}^\omega$ for some $\omega \in \Omega$, i.e., $\mathcal{R}(\mathcal{G}^\omega, i) = \{i\}$ is a singleton, the associated reachability constraint in (3b) reduces to $y_i \geq z_i^\omega$. By Observation 3.1, we can get $z_i^\omega = \min\{1, y_i\} = y_i$. As a result, we can perform a reduction on formulation (3) by aggregating $z_i^\omega := y_i$ and removing the associated constraint in (3b). We call this reduction the *singleton node aggregation* (SNA). Indeed, this is exactly the "P2" presolving method proposed in Güney et al. [24].

We next use Observation 3.1 to derive the INA. Let i_1 and i_2 be two isomorphic nodes in two different live-arc graphs $\mathcal{G}^\omega$ and $\mathcal{G}^\eta$. By Observation 3.1, there must exist an optimal solution $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ such that

$$\bar{z}_{i_1}^\omega = \min\left\{1, \sum_{j \in \mathcal{R}(\mathcal{G}^\omega, i_1)} \bar{y}_j\right\} \quad \text{and} \quad \bar{z}_{i_2}^\eta = \min\left\{1, \sum_{j \in \mathcal{R}(\mathcal{G}^\eta, i_2)} \bar{y}_j\right\}.$$

By (7), we have $\bar{z}_{i_1}^\omega = \bar{z}_{i_2}^\eta$. This implies that setting $z_{i_1}^\omega := z_{i_2}^\eta$ in formulation (3) does not change its optimal value. Consequently, we have the following presolving method.

INA. If nodes i_1 and i_2 in two different live-arc graphs $\mathcal{G}^\omega$ and $\mathcal{G}^\eta$ are isomorphic, variable $z^\omega_{i_1}$ can be replaced by variable $z^\eta_{i_2}$ and constraint $\sum_{j\in\mathcal{R}(\mathcal{G}^\omega,i_1)} y_j \geq z^\omega_{i_1}$ can be removed from formulation (3).

The SCNA can be regarded as a special case of the INA, which is restricted to aggregating the isomorphic (strongly connected) nodes inside each live-arc graph. However, as it has been discussed in subsection 3.1, implementing the SCNA can be done in $\mathcal{O}(\sum_{\omega\in\Omega}(|\mathcal{V}|+|\mathcal{A}^\omega|))$, which is much faster than that of implementing the INA. The latter requires to check whether or not condition (7) holds for all 4-tuples (ω,η,i_1,i_2) with an overall complexity of $\mathcal{O}(|\Omega|^2|\mathcal{V}|^3)$. This shows that, to implement the INA, it is better to first implement the SCNA, and then detect isomorphic nodes among different scenarios based on the compact formulation (6) (in subsection 5.1, we shall provide a fast heuristic algorithm for implementing the INA). After applying the INA on formulation (6), the formulation of the IMP can be presented as follows:

$$\begin{aligned}
\max_{\boldsymbol{y},\boldsymbol{z}} \ & \sum_{\omega\in\Omega}\sum_{u\in\tilde{\mathcal{V}}^\omega} f^\omega_u z^\omega_u \\
\text{s.t. } & (3c),\ (3d), \\
& \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} y(\mathcal{SC}^\omega_v) \geq z^\omega_u, \quad \forall\ \omega\in\Omega,\ \forall\ u\in\tilde{\mathcal{V}}^\omega, \\
& z^\omega_u \in\{0,1\}, \qquad\qquad \forall\ \omega\in\Omega,\ \forall\ u\in\tilde{\mathcal{V}}^\omega,
\end{aligned} \tag{9}$$

where $\tilde{\mathcal{V}}^\omega$ denotes the set of nodes in $\bar{\mathcal{G}}^\omega$ that are not aggregated by other isomorphic nodes and f^ω_u denotes the objective coefficient of variable z^ω_u after applying the INA. For simplicity, for two isomorphic nodes i_1 and i_2 in two different live-arc graphs $\mathcal{G}^\omega$ and $\mathcal{G}^\eta$, we aggregated variable $z^\omega_{i_1}$ by variable $z^\eta_{i_2}$ if $\omega>\eta$. It is worthwhile to highlight that, with the increasing number of scenarios $|\Omega|$, a node in a live-arc graph $\bar{\mathcal{G}}^\omega$ is more likely to be aggregated by other isomorphic nodes (in other live-arc graphs), and consequently, $|\tilde{\mathcal{V}}^\omega|$ in formulation (9) tends to be smaller.

Remark 3.2. *The SNA can be used to further simplify formulation* (9). *In particular, if* $\bigcup_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)}\mathcal{SC}^\omega_v=\{j\}$, *we can aggregate* $z^\omega_u:=y_j$ *and remove the corresponding reachability constraint in* (9).

4 Theoretical analysis

In this section, we demonstrate the strength of the proposed SCNA and INA in reducing the problem size of the SMCLP formulation (3) by analyzing two special cases of the IMP. In particular, for the first case where the IMP is built upon a one-way bipartite social network under the LTM, we are able to give upper bounds, which are linear with the size of the social network but independent of the number of samplings, for the numbers of variables and constraints in the reduced SMCLP formulation (obtained by applying the proposed presolving methods). For the second case where the IMP is built upon a complete social network under the ICM, we provide a lower bound for the probability that after applying the SCNA and INA, there are only $|\mathcal{V}|+1$ variables and two linear constraints in the reduced SMCLP formulation (9). Such a probability can tend to one under certain conditions.

4.1 One-way bipartite social network under the LTM

A one-way bipartite social network is a bipartite graph in which all arcs are from one side (the source nodes) to another side (the target nodes). One-way bipartite social

networks also arise from several applications [4, 5, 19, 26, 49]. For example, the authors in [4, 26, 49] considered one of the major decisions in a marketing plan that deals with the allocation of a given budget among media channels (i.e., source nodes) in order to maximize the influence on a set of potential customers (i.e., target nodes), which can be characterized by a one-way bipartite social network. Other applications on the one-way bipartite social networks include, e.g., the human sexual contact network in [19] (where the nodes denote the groups of two different genders and the arcs denote the sexual connections between males and females) and the collaboration network in [5] (the source and target nodes denote the organizations and the projects, respectively, and an arc between organization i and project j denotes that organization i participates in project j). Due to the simple structure, some theoretical properties on the influence propagation in one-way bipartite social networks have also been established. In particular, it is possible to compute the exact influence coverage $\sigma(S)$ by a dynamic programming procedure; see [54, 55]. We remark that most existing works assume that the influence spreads from the source nodes to the target nodes. In this subsection, we consider the generalized case where the source nodes and target nodes can also exert influence to themselves. In the following, we consider the IMP built upon such a one-way bipartite social network under the LTM.

Let $\mathcal{G}_{\mathrm{B}} = (\mathcal{M} \cup \mathcal{N}, \mathcal{A})$ be a given one-way bipartite graph. All arcs in $\mathcal{A}$ are from the source nodes set $\mathcal{M}$ to the target nodes set $\mathcal{N}$. For each scenario $\omega \in \Omega$, we denote its live-arc graph as $\mathcal{G}_{\mathrm{B}}^{\omega} = (\mathcal{M} \cup \mathcal{N}, \mathcal{A}^{\omega})$. Notice that under the LTM, in each live-arc graph $\mathcal{G}_{\mathrm{B}}^{\omega}$, each node i, $i \in \mathcal{M}$, does not have any incoming arc and each node j, $j \in \mathcal{N}$, has at most one incoming arc. Therefore, under the LTM, the reachability set of a node $i \in \mathcal{M}$ is $\mathcal{R}(\mathcal{G}_{\mathrm{B}}^{\omega}, i) = \{i\}$, and the reachability set of a node $j \in \mathcal{N}$ is

$$\mathcal{R}(\mathcal{G}_{\mathrm{B}}^{\omega}, j) = \begin{cases} \{j\}, & \text{if node } j \text{ does not have any incoming arc in } \mathcal{G}_{\mathrm{B}}^{\omega}; \\ \{i, j\}, & \text{if there exists some node } i \in \mathcal{M} \text{ such that } (i, j) \in \mathcal{A}^{\omega}. \end{cases} \tag{10}$$

Next, for each node $i \in \mathcal{M} \cup \mathcal{N}$, we define a set of scenarios

$$\Omega(i) := \{\omega \in \Omega \,:\, \mathcal{R}(\mathcal{G}_{\mathrm{B}}^{\omega}, i) = \{i\}\}, \tag{11}$$

and for each arc $(i, j) \in \mathcal{A}$, we define another set of scenarios

$$\Omega(i, j) := \{\omega \in \Omega \,:\, \mathcal{R}(\mathcal{G}_{\mathrm{B}}^{\omega}, j) = \{i, j\}\}. \tag{12}$$

By definition, it follows that

$$\begin{cases} \Omega(i) = \Omega, & \text{for } i \in \mathcal{M}; \\ \left(\bigcup\limits_{i \in \mathcal{M}} \Omega(i, j) \right) \cup \Omega(j) = \Omega, & \text{for } j \in \mathcal{N}. \end{cases} \tag{13}$$

Then, we have the followings:

(i) by applying the SNA for each $i \in \mathcal{M} \cup \mathcal{N}$, we can aggregate $z_i^{\omega} := y_i$ for all $\omega \in \Omega(i)$ and remove the corresponding constraints in (3b);

(ii) by applying the INA for each $(i, j) \in \mathcal{A}' := \{(i, j) \in \mathcal{A} \,:\, \Omega(i, j) \neq \varnothing\}$, we can aggregate all variables z_j^{ω}, $\omega \in \Omega(i, j)$, into a single variable, denoted as z_{ij}, and remove the redundant constraints in (3b).

As a result, the reduced formulation is given by

$$\max_{\boldsymbol{y},\,\boldsymbol{z}} \sum_{i\in\mathcal{M}\cup\mathcal{N}} s_i y_i + \sum_{(i,j)\in\mathcal{A}'} c_{ij} z_{ij} \tag{14a}$$

$$\text{s.t. } y_i + y_j \geq z_{ij}, \qquad \forall\ (i,j)\in\mathcal{A}', \tag{14b}$$

$$\sum_{i\in\mathcal{M}\cup\mathcal{N}} y_i \leq K, \tag{14c}$$

$$y_i \in \{0,1\}, \qquad \forall\ i\in\mathcal{M}\cup\mathcal{N}, \tag{14d}$$

$$z_{ij} \in \{0,1\}, \qquad \forall\ (i,j)\in\mathcal{A}', \tag{14e}$$

where $s_i := \sum_{\omega\in\Omega(i)} p^\omega$ for $i\in\mathcal{M}\cup\mathcal{N}$ and $c_{ij} := \sum_{\omega\in\Omega(i,j)} p^\omega$ for $(i,j)\in\mathcal{A}'$, respectively. Since $|\mathcal{A}'|\leq|\mathcal{A}|$, we have the following theorem providing upper bounds for the numbers of variables and constraints in the reduced SMCLP formulation (14).

Theorem 4.1. *Consider the IMP on the one-way bipartite social network $\mathcal{G}_\mathrm{B}$ with a finite set of scenarios Ω under the LTM. Applying the SNA and INA on formulation (3), the numbers of variables and constraints in the reduced SMCLP formulation (14) are at most $|\mathcal{M}|+|\mathcal{N}|+|\mathcal{A}|$ and $|\mathcal{A}|+1$, respectively.*

Finally, we provide more analysis results for formulation (14). To proceed, we note that using (13) and $\sum_{\omega\in\Omega} p^\omega = 1$, we have the following properties on the objective coefficients of formulation (14).

Remark 4.2. *(i) $s_i = 1$ for all $i\in\mathcal{M}$; and (ii) $s_j + \sum_{i\,:\,(i,j)\in\mathcal{A}'} c_{ij} = 1$ for all $j\in\mathcal{N}$.*

Proposition 4.3. *The linear programming (LP) relaxation of formulation (14) is tight. Moreover, formulation (14) can be solved in strongly polynomial-time.*

Proof. The proof is relegated to the appendix. □

4.2 Complete social network under the ICM

In this subsection, we study another special case of the IMP where the considered social network is a complete graph (denoted as $\mathcal{G}_\mathrm{C} = (\mathcal{V}_\mathrm{C}, \mathcal{A}_\mathrm{C})$) and the influence propagation model is the ICM. Let $\mathcal{G}_\mathrm{C}^\omega = (\mathcal{V}_\mathrm{C}, \mathcal{A}_\mathrm{C}^\omega)$ be a live-arc graph and denote $n = |\mathcal{V}_\mathrm{C}|$. Recall that for the live-arc graph constructed under the ICM, each arc is determined to be live independently with probability π_{ij}. Consequently, $\mathcal{G}_\mathrm{C}^\omega$ can be seen as a directed *Erdős-Rényi* (ER) random graph [8, 17]. If the arc probabilities are homogeneous, i.e., $\pi_{ij} = p$ for all $(i,j)\in\mathcal{A}_\mathrm{C}$ and some $p\in(0,1]$, $\mathcal{G}_\mathrm{C}^\omega$ is a homogeneous directed ER random graph; otherwise it is an inhomogeneous directed ER random graph. Homogeneous directed ER random graph is shown to be strongly connected with a probability tending to one (as $n\to\infty$) under certain conditions; see, e.g., [22]. The following lemma further provides a lower bound for the probability of the strong connectivity of $\mathcal{G}_\mathrm{C}^\omega$ with respect to the number of nodes n and the arc probability p.

Lemma 4.4. *Suppose that $\pi_{ij} = p\in(0,1]$ for all $(i,j)\in\mathcal{A}_\mathrm{C}$ and*

$$\max\left\{(n-1)(1-p^2)^{\frac{n}{2}+1}, 2(1-p^2)^{\frac{3n}{16}-1}\right\} \leq 1. \tag{15}$$

Then

$$\mathbb{P}(\textit{Graph } \mathcal{G}_\mathrm{C}^\omega \textit{ is strongly connected}) \geq 1 - n(n-1)(1-p^2)^{n-1}. \tag{16}$$

Proof. The proof can be found in section 1 of [9]. □

Theorem 4.5. *Consider the IMP on the complete social network $\mathcal{G}_{\mathrm{C}}$ with a finite set of scenarios Ω under the ICM. Suppose that $\pi_{ij} \geq p$ for all $(i,j) \in \mathcal{A}_{\mathrm{C}}$ and (15) holds. Then, by applying the SCNA and INA, there are only $n+1$ variables and two linear constraints in the reduced SMCLP formulation (9) with a probability at least p^* where*

$$p^* = (1 - n(n-1)(1-p^2)^{n-1})^{|\Omega|}. \tag{17}$$

Proof. Notice that the probability that $\mathcal{G}_{\mathrm{C}}^{\omega}$ is strongly connected with $\pi_{ij} \geq p$ for all $(i,j) \in \mathcal{A}_{\mathrm{C}}$ is larger than or equal to that with $\pi_{ij} = p$ for all $(i,j) \in \mathcal{A}_{\mathrm{C}}$. This, together with Lemma 4.4 and the fact that each live-arc graph $\mathcal{G}_{\mathrm{C}}^{\omega}$ is constructed independently, shows that the probability that all live-arc graphs $\mathcal{G}_{\mathrm{C}}^{\omega}$, $\omega \in \Omega$, are strongly connected is at least p^* (defined in (17)). The strong connectivity of graph $\mathcal{G}_{\mathrm{C}}^{\omega}$ implies that the number of SCCs in $\mathcal{G}_{\mathrm{C}}^{\omega}$ is one and $\mathcal{R}(\mathcal{G}_{\mathrm{C}}^{\omega}, i) = \mathcal{V}_{\mathrm{C}}$ for all $i \in \mathcal{V}_{\mathrm{C}}$. As a result, with a probability at least p^*, (i) the number of variables $\boldsymbol{z}$ in the reduced formulation (6) (after applying the SCNA on (3)) is equal to $|\Omega|$; and (ii) the number of variables $\boldsymbol{z}$ in the reduced formulation (9) (after applying the INA on (6)) is equal to one. The proof is completed. □

The value p^* in Theorem 4.5 can tend to one. For instance, if $p \in (0,1]$ is a constant, then $p^* \to 1$ as $n \to \infty$ (notice that in this case, condition (15) also holds). This shows that after applying the SCNA and INA, there are only $n+1$ variables and two linear constraints in the reduced SMCLP formulation (9) with a probability tending to one.

It is worthwhile remarking that Theorem 4.5 also sheds a useful insight that for the IMP with a general large and dense network $\mathcal{G}$ (not necessary to be complete) with high arc probabilities π_{ij}, the SCNA and INA can be expected to effectively reduce the sizes of the live-arc graphs and SMCLP formulation (3). Indeed, a dense network is likely to contain big complete subgraphs. By Theorem 4.5, with high arc probabilities, the nodes in these complete subgraphs are likely to be strongly connected in all live-arc graphs $\mathcal{G}^{\omega}$, $\omega \in \Omega$, and as a result, more reductions are likely to be detected. This is consistent with the computational results in subsection 6.2 where more reductions can be detected by the proposed SCNA and INA for large and dense networks with large arc probabilities (see Table 2 further ahead).

5 The algorithms

In this section, we first discuss the implementation of the INA in subsection 5.1. Then we present the BD algorithm with the proposed SCNA and INA for solving the IMP in subsection 5.2.

5.1 An algorithm for identifying presolving reductions by the INA

A straightforward implementation of the INA requires to first precompute and store the reachability sets of all nodes in all live-arc graphs $\mathcal{G}^{\omega}$, $\omega \in \Omega$, and then detect all 4-tuples (ω, η, i_1, i_2) that satisfy condition (7). However, this leads to a high runtime complexity and a large memory consumption, which are $\mathcal{O}(|\Omega|^2|\mathcal{V}|^3)$ and $\mathcal{O}(|\Omega||\mathcal{V}|^2)$, respectively. In this subsection, we shall overcome this weakness by presenting a hashing-based heuristic algorithm.

We first discuss the computation of the reachability sets. Güney et al. [24] computed the reachability set $\mathcal{R}(\mathcal{G}^{\omega}, i)$ of each node i in each live-arc graph $\mathcal{G}^{\omega}$, $\omega \in \Omega$, by applying a reverse BFS starting from node i. The computational complexity is $\mathcal{O}(|\mathcal{V}| + |\mathcal{A}^{\omega}|)$. Here we notice that it can be (possibly) much faster to compute $\mathcal{R}(\mathcal{G}^{\omega}, i)$ based on the compact graph $\bar{\mathcal{G}}^{\omega}$. Indeed, as it has been mentioned in subsection 3.1, the reachability sets of nodes inside a given SCC $\mathcal{SC}_u^{\omega}$ of $\mathcal{G}^{\omega}$ are identical, where u is the corresponding

node in graph $\bar{\mathcal{G}}^\omega$. Hence, to compute the reachability sets of the nodes in SCC $\mathcal{SC}_u^\omega$, we only need to compute the reachability set $\mathcal{R}(\mathcal{G}^\omega, \mathcal{SC}_u^\omega)$. To compute the latter one, we can apply a reverse BFS in the compact graph $\bar{\mathcal{G}}^\omega$ and use relation (4). The related complexity is $\mathcal{O}(|\mathcal{V}| + |\bar{\mathcal{A}}^\omega|)$, which is potentially much smaller than $\mathcal{O}(|\mathcal{V}| + |\mathcal{A}^\omega|)$, especially when $|\bar{\mathcal{A}}^\omega|$ is much smaller than $|\mathcal{A}^\omega|$. Furthermore, as $\bar{\mathcal{G}}^\omega$ is a directed acyclic graph (as each node forms an SCC), we can perform a *topological ordering* to further speed up the procedure of computing the reachability sets of all nodes in $\bar{\mathcal{G}}^\omega$. To be more specific, topological ordering for the directed acyclic graph $\bar{\mathcal{G}}^\omega$ is a linear ordering of nodes such that for each arc $(u_1, u_2) \in \bar{\mathcal{A}}^\omega$, node u_1 comes before node u_2 in the ordering. In our implementation, we traverse all nodes in $\bar{\mathcal{G}}^\omega$ to compute their reachability sets according to the topological ordering. In other words, when computing the reachability set of node $u \in \bar{\mathcal{V}}^\omega$, the reachability sets of nodes u', $u' \in \mathcal{N}_\omega^-(u)$, have been computed, where $\mathcal{N}_\omega^-(u) := \{u' : (u', u) \in \bar{\mathcal{A}}^\omega\}$ is the set of node u's neighbor nodes. Therefore, to compute node u's reachability set, we only need to traverse its neighbor nodes and use relation $\mathcal{R}(\bar{\mathcal{G}}^\omega, u) = \bigcup_{u' \in \mathcal{N}_\omega^-(u)} \mathcal{R}(\bar{\mathcal{G}}^\omega, u') \cup \{u\}$. This avoids performing a whole reverse BFS and generally accelerates the computation of $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$.

Next, we discuss the storage of the reachability sets. First, we can implement SCNA to alleviate the memory consumption of storing the reachability sets (as for each SCC, only a single node's reachability set needs to be stored). Second, to further avoid a large memory consumption, we restrict to storing those nodes' reachability sets whose sizes are smaller than or equal to a predefined parameter **MaxReacSize**. This also means that only nodes satisfying this criterion will be used for detecting the reductions by the INA. The rationale behind this strategy is that for the nodes with smaller reachability sets, it is more likely to detect nodes that are isomorphic to them, as illustrated in our computational results (see subsection 6.4 further ahead).

Finally, we apply the INA by detecting the pairs whose reachability sets are identical, i.e.,

$$\bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega = \bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^{\omega_0}, u_0)} \mathcal{SC}_v^{\omega_0} \tag{18}$$

for some $u \in \bar{\mathcal{V}}^\omega$ and $u_0 \in \bar{\mathcal{V}}^{\omega_0}$. To do this, we follow [2] to use a hashing-based method. The basic idea of the hashing-based method is to simultaneously build a hashing table that remembers the information of the reachability sets and test whether there exists a reachability set in the hashing table that is identical to the one we are currently looking at. Specifically, let $\mathcal{H}$ be the hashing table and for each $\omega \in \Omega$ and $u \in \bar{\mathcal{V}}^\omega$, let $\bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega$ be the key with 3-tuple $(\omega, u, \mathcal{R}(\bar{\mathcal{G}}^\omega, u))$ being the stored value in table $\mathcal{H}$. At first, table $\mathcal{H}$ is initialized to be $\varnothing$. Then, in each iteration, for scenario $\omega \in \Omega$ and node $u \in \bar{\mathcal{V}}^\omega$ (with $|\bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega| \leq$ **MaxReacSize**), table $\mathcal{H}$ is queried for $\bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega$. If condition (18) holds for some corresponding entry $(\omega_0, u_0, \mathcal{R}(\bar{\mathcal{G}}^{\omega_0}, u_0))$ in table $\mathcal{H}$, we apply the INA by removing variable z_u^ω, deleting the associated constraint in (5), and adding the objective coefficient of variable z_u^ω into that of variable $z_{u_0}^{\omega_0}$; otherwise, tuple $(\omega, u, \mathcal{R}(\bar{\mathcal{G}}^\omega, u))$ will be added into table $\mathcal{H}$. The procedure is repeated until all considered reachability sets are tested.

In summary, we present the implementation of the INA in Algorithm 1 to obtain the reduced SMCLP formulation (9). Here, for simplicity of presentation, the improvement of topological ordering is omitted in Algorithm 1. In step 4, we perform a reverse BFS on node u in graph $\bar{\mathcal{G}}^\omega$ to compute $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$. In steps 5-9, we use the hashing-based method to detect whether there exists some entry $(\omega_0, u_0, \mathcal{R}(\bar{\mathcal{G}}^{\omega_0}, u_0))$ in table $\mathcal{H}$ such that (18) holds. If yes, we apply the INA reductions; otherwise, we add the new entry $(\omega, u, \mathcal{R}(\bar{\mathcal{G}}^\omega, u))$ into table $\mathcal{H}$.

Algorithm 1: Implementation of the INA

Input: the compact live-arc graphs $\bar{\mathcal{G}}^\omega = (\bar{\mathcal{V}}^\omega, \bar{\mathcal{A}}^\omega)$ and the SCCs $\mathcal{SC}_u^\omega$, $u \in \bar{\mathcal{V}}^\omega$, of the (original) live-arc graphs $\mathcal{G}^\omega = (\mathcal{V}, \mathcal{A}^\omega)$, $\omega \in \Omega$.

Output: sets $\tilde{\mathcal{V}}^\omega$ and the objective coefficients f_u^ω of variables z_u^ω, $\omega \in \Omega$, $u \in \tilde{\mathcal{V}}^\omega$, in the reduced formulation (9).

Initialize $\tilde{\mathcal{V}}^\omega := \bar{\mathcal{V}}^\omega$, $f_u^\omega := p^\omega |\mathcal{SC}_u^\omega|$ for all $\omega \in \Omega$ and $u \in \bar{\mathcal{V}}^\omega$, and $\mathcal{H} := \varnothing$;
for $\omega \in \Omega$ **do**
  for $u \in \bar{\mathcal{V}}^\omega$ **do**
    Perform a reverse BFS on node u in graph $\bar{\mathcal{G}}^\omega$ to compute $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$;
    if $|\bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega| \leq$ **MaxReacSize** *and* (18) *holds for some* $(\omega_0, u_0, \mathcal{R}(\bar{\mathcal{G}}^{\omega_0}, u_0)) \in \mathcal{H}$ **then**
      Set $f_{u_0}^{\omega_0} := f_{u_0}^{\omega_0} + f_u^\omega$ and $\tilde{\mathcal{V}}^\omega := \tilde{\mathcal{V}}^\omega \backslash \{u\}$;
    else
      $\mathcal{H} := \mathcal{H} \cup \{(\omega, u, \mathcal{R}(\bar{\mathcal{G}}^\omega, u))\}$;
    end
  end
end

5.2 Benders decomposition algorithm for solving the IMP

Güney et al. [24] has proposed the BD algorithm to solve the IMP based on formulation (3). The authors showed the effectiveness of integrating the presolving method SNA into the BD algorithm. In this subsection, to further enhance the capability of using the BD algorithm to solve the IMP, we attempt to integrate the proposed SCNA and INA into the BD algorithm, or equivalently, to design a BD algorithm that is based on the reduced formulation (9) and the compact graphs $\bar{\mathcal{G}}^\omega$, $\omega \in \Omega$.

5.2.1 Reformulation of (9)

We first briefly introduce the BD reformulation of (9) (more details can be found in [24]). To begin with, we note that replacing each binary variable z_u^ω by a continuous variable taking value in $[0, 1]$ does not change the optimal value of formulation (9). For each $\omega \in \Omega$, let φ_ω represent the additional variable that captures the contribution of scenario ω to the objective function. Then, we can project out variables $\boldsymbol{z}$ and equivalently reformulate (9) as

$$\max_{\boldsymbol{y}, \boldsymbol{\varphi}} \left\{ \sum_{\omega \in \Omega} \varphi^\omega \ : \ (3c),\ (3d),\ \varphi^\omega \leq \Phi^\omega(\boldsymbol{y}), \forall\, \omega \in \Omega \right\} \tag{19}$$

where function $\Phi^\omega(\boldsymbol{y})$ is defined as follows:

$$\Phi^\omega(\boldsymbol{y}) := \max_{\boldsymbol{z}} \left\{ \sum_{u \in \tilde{\mathcal{V}}^\omega} f_u^\omega z_u^\omega \ : \ z_u^\omega \leq \sum_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} y(\mathcal{SC}_v^\omega),\ 0 \leq z_u^\omega \leq 1,\ \forall\, u \in \tilde{\mathcal{V}}^\omega \right\}. \tag{20}$$

For a fixed $\bar{\boldsymbol{y}} \in [0, 1]^{|\mathcal{V}|}$, to model the inequalities $\varphi^\omega \leq \Phi^\omega(\boldsymbol{y})$ for all $\omega \in \Omega$, we use the Benders optimality cuts which are derived as follows. First, the dual of formulation (20) when $\boldsymbol{y} = \bar{\boldsymbol{y}}$ is

$$\min_{\boldsymbol{\alpha}^\omega, \boldsymbol{\beta}^\omega} \left\{ \sum_{u \in \tilde{\mathcal{V}}^\omega} \left(\alpha_u^\omega \sum_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \bar{y}(\mathcal{SC}_v^\omega) + \beta_u^\omega \right) \ : \ \alpha_u^\omega + \beta_u^\omega \geq f_u^\omega,\ \alpha_u^\omega,\ \beta_u^\omega \geq 0,\ \forall\, u \in \tilde{\mathcal{V}}^\omega \right\}, \tag{21}$$

where α_u^ω and β_u^ω are the dual variables of constraints $z_u^\omega \leq \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} \bar{y}(\mathcal{SC}_v^\omega)$ and $z_u^\omega \leq 1$, respectively. Clearly, formulation (21) has a closed form solution $(\bar{\boldsymbol{\alpha}}^\omega(\bar{\boldsymbol{y}}), \bar{\boldsymbol{\beta}}^\omega(\bar{\boldsymbol{y}}))$:

$$(\bar{\alpha}_u^\omega(\bar{\boldsymbol{y}}), \bar{\beta}_u^\omega(\bar{\boldsymbol{y}})) = \begin{cases} (0, f_u^\omega), & \text{if } \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} \bar{y}(\mathcal{SC}_v^\omega) \geq 1; \\ (f_u^\omega, 0), & \text{otherwise;} \end{cases} \quad \forall\ u \in \tilde{\mathcal{V}}^\omega. \tag{22}$$

Then the Benders optimality cuts for formulation (19) are given by

$$\varphi^\omega \leq \sum_{u\in\tilde{\mathcal{V}}^\omega} \left(\bar{\alpha}_u^\omega(\bar{\boldsymbol{y}}) \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_u^\omega(\bar{\boldsymbol{y}}) \right),\ \forall\ \omega \in \Omega. \tag{23}$$

For simplicity, we shall abbreviate $\bar{\alpha}_u^\omega(\bar{\boldsymbol{y}})$ and $\bar{\beta}_u^\omega(\bar{\boldsymbol{y}})$ as $\bar{\alpha}_u^\omega$ and $\bar{\beta}_u^\omega$, respectively. Let $C^\omega := \sum_{u\in\tilde{\mathcal{V}}^\omega} \bar{\beta}_u^\omega$ and $c_j^\omega := \sum_{u\in\mathcal{J}_j^\omega} \bar{\alpha}_u^\omega$, $j \in \mathcal{V}$, where $\mathcal{J}_j^\omega := \left\{ u \in \tilde{\mathcal{V}}^\omega\ :\ j \in \bigcup_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} \mathcal{SC}_v^\omega \right\}$. Then Benders optimality cuts (23) can be alternatively written as

$$\varphi^\omega \leq \sum_{j\in\mathcal{V}} c_j^\omega y_j + C^\omega,\ \forall\ \omega \in \Omega. \tag{24}$$

For a given point $\bar{\boldsymbol{y}} \in [0,1]^{|\mathcal{V}|}$, it is interesting to ask whether or not applying the SCNA or INA changes the Benders optimality cuts (23), or equivalently, whether or not the Benders optimality cuts (23) based on formulations (3), (6), and (9) are equivalent. This question is addressed by the following proposition.

Proposition 5.1. *Given a point $\bar{\boldsymbol{y}} \in [0,1]^{|\mathcal{V}|}$ and a scenario $\omega \in \Omega$, (i) only applying the SCNA will not change the Benders optimality cut; (ii) applying the INA may change the Benders optimality cut.*

Proof. The proof is relegated to the appendix. □

5.2.2 Solution approach

We now describe the BD algorithm to solve the Benders formulation (19) of the IMP. To implement the BD algorithm, we use a branch-and-Benders-cut approach in which a branch-and-bound tree is created and the Benders optimality cuts (23) are separated at each branch node. Following [24, 53], we start with a relaxed master problem of (19) in which only the following inequalities (called *submodular inequalities*)

$$\varphi^\omega \leq \sum_{j\in\mathcal{V}} \Phi^\omega(\boldsymbol{e}_j) y_j,\ \forall\ \omega \in \Omega, \tag{25}$$

are added, where $\boldsymbol{e}_j,\ j \in \mathcal{V}$ is the unit vector with appropriate dimension. Having a solution $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\varphi}})$ of the LP relaxation of this relaxed master problem, we next attempt to separate Benders optimality cuts (23). To do this, we need to compute all reachability sets of all nodes of all scenarios. However, as it has been mentioned in subsection 5.1, it is unrealistic to compute and store all reachability sets of all nodes of all scenarios a priori due to the large memory consumption. Hence, similar to [24], we introduce parameter **MemLimPerScen** to denote the maximally allowed memory consumption per scenario. In particular, for each scenario $\omega \in \Omega$, we store the reachability sets of nodes according to their topological ordering in the compact live-arc graph $\bar{\mathcal{G}}^\omega$ until the memory consumption reaches **MemLimPerScen**. As a result, when computing Benders optimality cut (23), if $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$ has been stored, we access it directly; otherwise, we perform a reverse BFS to compute $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$ on the fly. Moreover, to further improve the efficiency of computing Benders optimality cut (23), we can omit to compute $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$ if $(\bar{\alpha}_u^\omega, \bar{\beta}_u^\omega) = (0, f_u^\omega)$ is

known a priori. More specifically, let $\mathcal{S}^\omega := \{u \in \bar{\mathcal{V}}^\omega : \bar{y}(\mathcal{SC}_u^\omega) \geq 1\}$ and $\mathcal{R}(\mathcal{S}^\omega) := \left\{u \in \tilde{\mathcal{V}}^\omega : \text{there exists a directed path from a node in } \mathcal{S}^\omega \text{ to node } u \text{ in } \bar{\mathcal{G}}^\omega\right\}$. Then, for each $u \in \mathcal{R}(\mathcal{S}^\omega)$, we must have $\sum_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \bar{y}(\mathcal{SC}_v^\omega) \geq 1$, and hence $(\bar{\alpha}_u^\omega, \bar{\beta}_u^\omega) = (0, f_u^\omega)$. In summary, we present the separation of Benders optimality cuts in Algorithm 2.

Algorithm 2: Separation of Benders optimality cuts (23)

Input: the compact live-arc graphs $\bar{\mathcal{G}}^\omega = (\bar{\mathcal{V}}^\omega, \bar{\mathcal{A}}^\omega)$, $\omega \in \Omega$, formulation (9), and point $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\varphi}})$.

Output: the set $\mathcal{C}$ of Benders optimality cuts which are violated by point $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\varphi}})$.

1 Initialize $\mathcal{C} := \varnothing$;
2 **for** $\omega \in \Omega$ **do**
3 Compute $\mathcal{S}^\omega := \left\{u \in \bar{\mathcal{V}}^\omega : \bar{y}(\mathcal{SC}_u^\omega) \geq 1\right\}$;
4 Perform a BFS in $\bar{\mathcal{G}}^\omega$ to compute $\mathcal{R}(\mathcal{S}^\omega)$;
5 Initialize $C^\omega := \sum_{u \in \mathcal{R}(\mathcal{S}^\omega)} f_u^\omega$ and $c_j^\omega := 0$, $j \in \mathcal{V}$;
6 **for** $u \in \tilde{\mathcal{V}}^\omega \setminus \mathcal{R}(\mathcal{S}^\omega)$ **do**
7 **if** $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$ *is not stored in the memory* **then**
8 Perform a reverse BFS in $\bar{\mathcal{G}}^\omega$ to compute $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$;
9 **end**
10 **if** $\sum_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \bar{y}(\mathcal{SC}_v^\omega) \geq 1$ **then**
11 Set $C^\omega := C^\omega + f_u^\omega$;
12 **else**
13 Set $c_j^\omega := c_j^\omega + f_u^\omega$ for all $j \in \bigcup_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \mathcal{SC}_v^\omega$;
14 **end**
15 **end**
16 **if** $\bar{\varphi}^\omega > \sum_{j \in \mathcal{V}} c_j^\omega \bar{y}_j + C^\omega$ **then**
17 Set $\mathcal{C} := \mathcal{C} \cup \left\{\varphi^\omega \leq \sum_{j \in \mathcal{V}} c_j^\omega y_j + C^\omega\right\}$;
18 **end**
19 **end**

In Algorithm 2, C^ω and c_j^ω for each $j \in \mathcal{V}$ are used to keep track of the constant term and the coefficient of variable y_j in Benders optimality cut (23), respectively. For each $\omega \in \Omega$, we initialize C^ω and c_j^ω in step 5 and then sequentially update them depending on whether or not $\sum_{v \in \mathcal{R}(\bar{\mathcal{G}}^\omega, u)} \bar{y}(\mathcal{SC}_v^\omega) \geq 1$ in steps 10-14. Finally, in steps 16-18, if Benders optimality cut $\varphi^\omega \leq \sum_{j \in \mathcal{V}} c_j^\omega y_j + C^\omega$ is violated by point $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\varphi}})$, we add it into the set of violated Benders optimality cuts $\mathcal{C}$.

It is worth emphasizing the computational efficiency of our separation algorithm for the Benders optimality cuts over the one in [24]. First, in Algorithm 2, to compute sets $\mathcal{R}(\mathcal{S}^\omega)$ and $\mathcal{R}(\bar{\mathcal{G}}^\omega, u)$, we perform (reverse) BFSes in the compact live-arc graph $\bar{\mathcal{G}}^\omega$, which is potentially much faster than that in [24] where the (reverse) BFSes in the (original) live-arc graph $\mathcal{G}^\omega$ are performed. Second, for each $\omega \in \Omega$, the number of variables z_u^ω is $|\tilde{\mathcal{V}}^\omega|$ which can be potentially much smaller than $|\mathcal{V}|$ (in [24]), and as a result, it can be expected that fewer reverse BFSes will be performed in Algorithm 2.

6 Computational results

In this section, we present the computational results to show the effectiveness of the proposed SCNA and INA. We use the BD algorithm in subsection 5.2, which is implemented in C++ linked with IBM ILOG CPLEX optimizer 20.1.0 [16]. The Benders cuts are added using CALLABLE LIBRARIES under the default settings of the branch-and-cut framework of CPLEX. The time limit is set to 14400 seconds, and all the experiments

were performed on a cluster of Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz computers. Only a single core was used in our experiments. We note here that throughout this section, all averages are taken to be geometric means. Since the statistics can be zero, we use the shifted geometric mean with a shift of 1 (the shifted geometric mean of values $x_1, x_2, \ldots, x_n$ with shift s is defined as $\prod_{k=1}^{n}(x_k + s)^{1/n} - s$; see [1]).

6.1 Networks and settings

Our benchmark data set consists of eight real-world social networks. Four of them have been used in [24, 53] (MSG, GNU, HEP, and ENRON) and the other four networks are from the SNAP database [1] (FACEBOOK, DEEZER, TWITTER, and EPINIONS). The latter ones are large-scale networks which are used to test the performance of the SCNA and INA in large-scale cases. For the undirected networks (GNU, HEP, FACEBOOK, and DEEZER), we convert them into directed networks by adding two directed arcs (i, j) and (j, i) for each edge (i, j). Table 1 summarizes the basic information of these networks where $\rho := |\mathcal{A}|/|\mathcal{V}|$ is used to reflect the density of the networks. In general, the larger ρ, the bigger SCCs in the network.

Table 1: Eight real-world social networks.

Network	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$	ρ	Description
MSG	1899	59835	31.5	Messaging network of UC-Irvine [44]
GNU	10879	79988	7.4	Gnutella peer-to-peer file sharing network [46]
HEP	15233	117782	7.7	High energy physics paper citation network [10]
ENRON	36692	367662	10.0	Email communication network from Enron [36]
FACEBOOK	50515	1638612	32.4	Facebook page network in the category of artist [47]
DEEZER	54573	996404	18.3	Deezer friendship network of users in Croatia [47]
TWITTER	81306	1768149	21.7	Social network from Twitter [34]
EPINIONS	131828	841372	6.4	Who-trust-whom social network of Epinions [37]

The selections of parameters of the IMP are also similar to the one in [24, 53]. More specifically, for the IMP under both the ICM and LTM, the cardinality restriction K in formulation (3) is selected in $\{5, 10, 15, 25\}$. To reflect different approximation levels of the sampling version of the IMP, the number of scenarios $|\Omega|$ is selected in $\{250, 500, 1000\}$. We next discuss the selections of the activation probability π_{ij} (under the ICM) and the weight b_{ij} (under the LTM) for each arc (i, j). Let n_{ij} denote the number of parallel arcs from node i to node j. For the IMP under the ICM, each single arc is assigned the same activation probability p chosen in $\{0.01, 0.05, 0.10\}$. Thus, π_{ij} is set to $1 - (1 - p)^{n_{ij}}$ to represent the probability that at least one of these parallel arcs appears in a live-arc graph. For the IMP under the LTM, we set influence weight on arc (i, j) as $b_{ij} = n_{ij}/n_j$, where $n_j := \sum_{i:(i,j)\in\mathcal{A}} n_{ij}$ (the number of incoming arcs of node j in $\mathcal{G}$) is a normalization factor to ensure that the sum of weights of the incoming arcs to j is 1. We remark that, with more parallel arcs from node i to node j, the activation probability π_{ij} (under the ICM) or weight b_{ij} (under the LTM) is also larger. For the IMP with each combination of the above parameters, 5 instances are randomly generated. Therefore, for each social network in Table 1, we have 180 and 60 instances for the IMP under the ICM and LTM, respectively.

In our experiments, we compare the performance of the following three settings:

[1] https://snap.stanford.edu/data/.

- **Default**: solving the IMP based on the Benders reformulation of (3) with the SNA applied [2];
- **SCNA**: **Default** with the SCNA applied;
- **INA**: **Default** with the SCNA and INA applied.

Following [24], the memory control parameter **MemLimPerScen** is set to $8/|\Omega|$ GB. Unless otherwise stated, in the implementation of the INA (i.e., Algorithm 1), parameter **MaxReacSize** is set to 8 and 4 for the IMP under the ICM and LTM, respectively. Finally, to avoid generating too many Benders optimality cuts at fractional points, we follow [24] to stop the separation procedure if the dual bound improves by less than 0.001.

6.2 Results for the IMP under the ICM

In this subsection, we test the effectiveness of the proposed presolving methods SCNA and INA for the IMP under the ICM.

Table 2 reports the reductions by applying the SNA, SCNA, and INA. For convenience, we only report the results for the case $|\Omega| = 1000$ since the results for the other two cases ($|\Omega| = 250, 500$) are similar. We use ΔZ and ΔNNZ to represent the average percentages of the eliminated variables $\boldsymbol{z}$ and nonzeros in formulation (3) through applying the SNA (which has been embedded in setting **Default**). In addition, we use $+\Delta$Z and $+\Delta$NNZ to denote the additional percentages of the eliminated variables and nonzeros (beyond the SNA) through applying the SCNA/INA. For the SCNA, we additionally list the average percentages of nodes and arcs reductions ($\Delta \mathrm{V} = \sum_{\omega\in\Omega}(|\mathcal{V}| - |\bar{\mathcal{V}}^\omega|)/(|\Omega||\mathcal{V}|)$ and $\Delta \mathrm{A} = \sum_{\omega\in\Omega}(|\mathcal{A}^\omega| - |\bar{\mathcal{A}}^\omega|)/(|\Omega||\mathcal{A}^\omega|)$) to compare the sizes of the compact live-arc graphs $\bar{\mathcal{G}}^\omega = (\bar{\mathcal{V}}^\omega, \bar{\mathcal{A}}^\omega)$ and the original live-arc graphs $\mathcal{G}^\omega = (\mathcal{V}, \mathcal{A}^\omega)$. Notice that for the SCNA, the reductions on the numbers of nodes and variables $\boldsymbol{z}$ are equal, and hence we have $+\Delta \mathrm{Z} = \Delta \mathrm{V}$ under setting **SCNA**.

As it can be seen in Table 2, when the network has a small density ρ or activation probability p, the singleton nodes (nodes without any incoming arc) are more likely to appear in the live-arc graphs. As a result, the SNA can eliminate a considerably large numbers of variables and nonzeros. In contrast, the SCNA is more effective in eliminating the variables and nonzeros when the network has a relatively large value of ρ or p. This is reasonable since as ρ or p increases, the original live-arc graphs contain more arcs, and as a result, bigger SCCs are likely to appear. As for the numbers of nodes and arcs in the compact live-arc graphs, we can observe that they are much smaller than those in the original live-arc graphs and the larger ρ or p is, the more reductions are detected by the SCNA. For the INA, since the isomorphic nodes inside each scenario have been identified by the SCNA, the additional reductions come from the detection of the isomorphic nodes among different scenarios. Nevertheless, we can observe a clear reduction on the number of variables (beyond the SCNA). However, the reduction on the number of nonzeros is relatively small, which is due to the fact that most eliminated variables are associated with small reachability sets as we set **MaxReacSize** $= 8$ in Algorithm 1 (in subsection 6.4, we will perform numerical experiments to confirm that setting **MaxReacSize** $= 8$ in Algorithm 1 is enough to identify almost all pairs of isomorphic nodes).

As discussed in the end of subsection 3.2, with the increasing number of samplings in the IMP, a node in a live-arc graph $\bar{\mathcal{G}}^\omega$ is more likely to be aggregated by other isomorphic nodes (in other live-arc graphs). Table 3 further reports the reductions on the problem size of the SMCLP formulation (3) with different numbers of samplings

[2] This setting can be seen as the implementation in [24]. Unfortunately, we could not access the code of [24] online. Therefore, the results reported in this section are based on our implementation. Notice that, however, due to the differences in hardware and randomness in sampling, it cannot be expected that the results of setting **Default** are the same as those in [24].

Table 2: Average reductions on the sizes of live-arc graphs, numbers of variables $\boldsymbol{z}$, and nonzeros in (3) through applying the SNA, SCNA, and INA (for the IMP under the ICM).

		Default		**SCNA**			**INA**	
Network	p	ΔZ	ΔNNZ	+ΔZ/ΔV	+ΔNNZ	ΔA	+ΔZ	+ΔNNZ
MSG	0.01	80.9%	16.1%	+2.1%	+19.4%	12.9%	+6.9%	+20.9%
($\rho = 31.5$)	0.05	56.4%	0.4%	+25.6%	+62.4%	78.8%	+27.9%	+62.4%
	0.10	44.1%	0.2%	+37.2%	+68.2%	88.6%	+38.6%	+68.3%
GNU	0.01	93.1%	89.3%	<+0.1%	+0.1%	1.0%	+5.4%	+7.9%
($\rho = 7.4$)	0.05	72.3%	46.2%	+0.9%	+2.3%	5.1%	+14.5%	+16.2%
	0.10	55.8%	0.3%	+8.3%	+36.1%	19.9%	+19.0%	+36.2%
HEP	0.01	93.3%	88.4%	+0.2%	+0.4%	4.6%	+5.3%	+8.0%
($\rho = 7.7$)	0.05	75.8%	3.1%	+4.9%	+49.3%	29.7%	+17.8%	+50.2%
	0.10	62.1%	0.4%	+13.4%	+65.4%	52.1%	+29.7%	+65.6%
ENRON	0.01	92.8%	34.4%	+0.1%	+6.5%	2.2%	+3.1%	+8.1%
($\rho = 10.0$)	0.05	76.8%	0.2%	+7.6%	+48.8%	50.0%	+13.6%	+48.8%
	0.10	64.1%	0.1%	+15.5%	+56.6%	66.1%	+22.8%	+56.6%
FACEBOOK	0.01	79.1%	0.6%	+2.4%	+31.0%	13.9%	+8.5%	+31.0%
($\rho = 32.4$)	0.05	47.8%	<0.1%	+31.7%	+69.6%	77.0%	+36.3%	+69.6%
	0.10	32.5%	<0.1%	+51.4%	+80.4%	90.7%	+54.3%	+80.4%
DEEZER	0.01	84.4%	73.7%	+0.1%	+0.2%	1.0%	+9.2%	+12.2%
($\rho = 18.3$)	0.05	50.6%	<0.1%	+16.6%	+54.0%	39.1%	+26.5%	+54.0%
	0.10	32.4%	<0.1%	+44.8%	+75.8%	77.1%	+50.6%	+75.8%
TWITTER	0.01	83.8%	0.9%	+2.1%	+44.9%	24.4%	+8.0%	+45.0%
($\rho = 21.7$)	0.05	57.9%	<0.1%	+18.7%	+64.7%	67.5%	+26.2%	+64.7%
	0.10	43.1%	<0.1%	+34.2%	+72.4%	83.3%	+40.6%	+72.4%
EPINIONS	0.01	95.9%	19.4%	+0.2%	+16.1%	4.1%	+1.7%	+16.5%
($\rho = 6.4$)	0.05	87.7%	0.2%	+3.6%	+44.5%	52.5%	+6.8%	+44.5%
	0.10	81.2%	0.1%	+6.4%	+46.5%	65.9%	+10.8%	+46.5%

under setting **INA**. For convenience, we only report results for the case $p = 0.1$ as the results for the other two cases are similar. As expected, with the increasing number of samplings in the IMP, (generally) more reductions will be derived by the INA, especially for the reductions on the number of variables $\boldsymbol{z}$.

We now evaluate the performance improvement of the integration of the SCNA and INA with the BD algorithm. In Table 4, for each of the eight networks, we report the total number of instances that can be solved within the time limit (#S), the average CPU time in seconds (T), the average presolving time in seconds (PT), the average number of branch nodes (#N), and the average number of added Benders optimality cuts (#C). Notice that the CPU time T includes the presolving time PT spent on applying the SCNA/INA. Detailed statistics of these results can be found in Tables 1a-8a of [9]. As it can be seen in Table 4, the performance of setting **SCNA** is much better than that of setting **Default**, especially for instances with large and dense networks. In total, setting **SCNA** can solve 1320 instances with a CPU time of 297.8 seconds, while setting **Default** can only solve 1122 instances with a CPU time of 670.9 seconds. The main improvement comes from the reductions on graph searching time in separating the Benders optimality cuts since by Proposition 5.1, we know that before and after applying

Table 3: Comparison of effectiveness of the INA with different numbers of scenarios (for the IMP under the ICM).

Network	$\|\Omega\|$	$+\Delta$Z	$+\Delta$NNZ	Network	$\|\Omega\|$	$+\Delta$Z	$+\Delta$NNZ
MSG	250	+38.4%	+68.2%	FACEBOOK	250	+53.8%	+80.3%
($\rho = 31.5$)	500	+38.5%	+68.2%	($\rho = 32.4$)	500	+54.1%	+80.4%
	1000	+38.6%	+68.3%		1000	+54.3%	+80.4%
GNU	250	+16.7%	+36.1%	DEEZER	250	+49.4%	+75.6%
($\rho = 7.4$)	500	+17.9%	+36.1%	($\rho = 18.3$)	500	+50.1%	+75.7%
	1000	+19.0%	+36.2%		1000	+50.6%	+75.8%
HEP	250	+28.1%	+65.4%	TWITTER	250	+39.4%	+72.4%
($\rho = 7.7$)	500	+29.0%	+65.5%	($\rho = 21.7$)	500	+40.0%	+72.4%
	1000	+29.7%	+65.6%		1000	+40.6%	+72.4%
ENRON	250	+22.0%	+56.5%	EPINIONS	250	+10.3%	+46.4%
($\rho = 10.0$)	500	+22.5%	+56.6%	($\rho = 6.4$)	500	+10.6%	+46.5%
	1000	+22.8%	+56.6%		1000	+10.8%	+46.5%

the SCNA method, the Benders optimality cuts are identical, and hence the solution procedure must be identical. The latter is further confirmed by the results of networks MSG and EPINIONS where the numbers of added Benders cuts and branch nodes are identical under settings **Default** and **SCNA**. Notice that it is reasonable to observe that the numbers of added Benders cuts and branch nodes in other networks are different since some instances cannot be solved within the time limit.

As for setting **INA**, we observe that it slightly improves the performance, compared to setting **SCNA**. In total, setting **INA** can solve 11 more instances than setting **SCNA**, with the CPU time decreasing from 297.8 seconds to 255.0 seconds. This is consistent with the former results in Table 2 in which only isomorphic nodes with small reachability sets can be detected by the INA and hence its contribution to speed up the solution procedure is not very large.

From the above simulation results, we can conclude that for the IMP under the ICM, (i) the SCNA can effectively reduce the sizes of networks and hence is beneficial to solving the IMP especially when the network is large and dense; and (ii) the INA can further remove a fairly large fraction of variables $\boldsymbol{z}$ (and the corresponding reachability constraints) from the SMCLP formulation and slightly speed up the solution procedure.

Table 4: Performance improvement through applying the SCNA and INA (for the IMP under the ICM).

	Default				**SCNA**					**INA**				
Network	#S	T	#N	#C	#S	T	PT	#N	#C	#S	T	PT	#N	#C
MSG	**180**	7.9	2	1699	**180**	3.4	0.3	2	1699	**180**	**2.9**	0.4	0	994
GNU	171	98.9	18	1625	172	72.6	1.8	18	1644	**173**	**64.5**	3.7	16	1598
HEP	171	229.8	31	4702	176	142.7	2.4	32	4713	**177**	**96.8**	4.1	33	4301
ENRON	165	1117.2	14	4897	**166**	496.1	5.6	14	4907	**166**	**376.9**	8.0	15	5324
FACEBOOK	56	5458.1	2	682	137	1963.4	10.1	54	2826	**142**	**1753.9**	14.5	57	2517
DEEZER	92	2888.0	1	552	144	1070.5	11.7	13	1620	**148**	**1009.2**	21.3	12	1643
TWITTER	107	5671.2	2	550	**165**	1461.4	17.8	2	3866	**165**	**1417.7**	29.6	2	3641
EPINIONS	**180**	1998.5	1	2200	**180**	867.0	21.6	1	2200	**180**	**750.8**	27.2	2	3060
TOTAL	1122	670.9	5	1524	1320	297.8	6.9	10	2662	**1331**	**255.0**	9.9	9	2530

6.3 Results for the IMP under the LTM

In this subsection, we present similar computational results for the IMP under the LTM in Tables 5-7.

To begin with, we note that for node j in graph $\mathcal{G}$, if it is a singleton node (i.e., it does not have any incoming arc), then in each live-arc graph $\mathcal{G}^\omega$, it is also a singleton node; otherwise, it has at most one incoming arc in each live-arc graph $\mathcal{G}^\omega$ (indeed, in the tested instances, it has exactly one incoming arc as $\sum_{i:(i,j)\in\mathcal{A}} b_{ij} = 1$ holds; see subsection 6.1). As a result, for any pair of nodes i_1 and i_2 in live-arc graph $\mathcal{G}^\omega$, there exists at most one directed path from node i_1 to node i_2. This implies that in graph $\mathcal{G}^\omega$, the subgraph induced by the nodes in SCC $\mathcal{SC}_u^\omega$ with $|\mathcal{SC}_u^\omega| \geq 2$ must be a single circle (which is in sharp contrast to the IMP under the ICM in which the subgraph induced by the nodes in SCC $\mathcal{SC}_u^\omega$ with $|\mathcal{SC}_u^\omega| \geq 2$ can be a union of multiple circles). This property for the IMP under the LTM, however, implies that in the live-arc graphs, the SCCs are likely to be small, and (hence) the reachability sets of the nodes are likely to be small. Consequently, through applying the SCNA, we can only observe a mild reduction on the numbers of nodes and arcs and the numbers of variables and nonzeros in Table 5. However, due to the small sizes of the reachability sets, more nodes are likely to be isomorphic among different scenarios and hence the INA can detect more reductions, as compared to the IMP under the ICM. This is shown in Table 5 in which we observe a fairly large reduction on the number of variables by applying the INA. Similar to the case under the ICM, with the increasing number of scenarios, more reductions can be detected by the INA; see Table 6. The reduction on the number of nonzeros is relatively small which can be explained by the fact that most eliminated variables by the INA are associated with small reachability sets as **MaxReacSize** is set to 4 in our implementation (see subsection 6.4 further ahead for the reason of setting **MaxReacSize** = 4). Notice that in Table 5, the reduction detected by the SNA (**Default**) is marginal since, as it has been mentioned, a node is a singleton node in live-arc graph $\mathcal{G}^\omega$ if and only if it is a singleton node in the original graph $\mathcal{G}$. Therefore, the reductions by applying the SNA for the IMP under the LTM totally depend on the number of singleton nodes in network $\mathcal{G}$, which is very small in most cases. Indeed, only network EPINIONS contains a relatively large percentage of singleton nodes (35.9%); see column ΔZ under setting **Default** in Table 5.

Table 5: Average reductions on the sizes of live-arc graphs, numbers of variables $\boldsymbol{z}$, and nonzeros in (3) through applying the SNA, SCNA, and INA (for the IMP under the LTM).

	Default		**SCNA**			**INA**	
Network	ΔZ	ΔNNZ	+ΔZ/ΔV	+ΔNNZ	ΔA	+ΔZ	+ΔNNZ
MSG	1.9%	0.3%	+2.4%	+0.8%	4.6%	+12.6%	+4.5%
GNU	<0.1%	<0.1%	+5.9%	+1.9%	11.8%	+26.2%	+10.5%
HEP	<0.1%	<0.1%	+22.7%	+15.3%	42.6%	+71.7%	+54.2%
ENRON	<0.1%	<0.1%	+8.7%	+3.0%	16.0%	+31.3%	+12.1%
FACEBOOK	<0.1%	<0.1%	+2.9%	+0.4%	5.5%	+7.2%	+0.9%
DEEZER	<0.1%	<0.1%	+4.4%	+0.9%	8.5%	+11.6%	+2.4%
TWITTER	<0.1%	<0.1%	+3.3%	+1.0%	5.4%	+8.4%	+2.2%
EPINIONS	35.9%	10.3%	+1.8%	+0.8%	5.5%	+21.8%	+10.8%

Table 6: Comparison of effectiveness of the INA with different numbers of scenarios (for the IMP under the LTM).

Network	$\lvert\Omega\rvert$	+ΔZ	+ΔNNZ	Network	$\lvert\Omega\rvert$	+ΔZ	+ΔNNZ
MSG	250	+9.5%	+3.3%	FACEBOOK	250	+6.0%	+0.8%
($\rho = 31.5$)	500	+11.0%	+3.9%	($\rho = 32.4$)	500	+6.6%	+0.9%
	1000	+12.6%	+4.5%		1000	+7.2%	+0.9%
GNU	250	+20.7%	+8.0%	DEEZER	250	+9.2%	+1.9%
($\rho = 7.4$)	500	+23.6%	+9.3%	($\rho = 18.3$)	500	+10.4%	+2.1%
	1000	+26.2%	+10.5%		1000	+11.6%	+2.4%
HEP	250	+67.8%	+50.7%	TWITTER	250	+6.8%	+1.8%
($\rho = 7.7$)	500	+70.0%	+52.7%	($\rho = 21.7$)	500	+7.6%	+2.0%
	1000	+71.7%	+54.2%		1000	+8.4%	+2.2%
ENRON	250	+26.0%	+9.8%	EPINIONS	250	+19.9%	+9.7%
($\rho = 10.0$)	500	+28.7%	+11.0%	($\rho = 6.4$)	500	+20.9%	+10.3%
	1000	+31.3%	+12.1%		1000	+21.8%	+10.8%

We now present the overall performance improvement of integrating the SCNA and INA into the BD algorithm in Table 7. Detailed statistics of these results can be found in Tables 1b-8b of [9]. From Table 7, we can see that the performance of settings **SCNA** and **INA** is slightly better than setting **Default**. Indeed, we only observe a minor improvement on the average CPU time (T) and the number of solved instances (#S) through applying the SCNA and INA. This can be explained by the reasons that (i) the reduction on the sizes of networks through applying the SCNA is small (as shown in Table 5); (ii) the time spent in implementing the SCNA and INA is relative large (as shown in column PT in Table 7); and (iii) only isomorphic nodes with small reachability sets can be detected by the INA (as shown in column +ΔNNZ of setting **INA** in Table 5). In addition, we note from Table 7 that the IMPs under the LTM are generally much easier than those under the ICM. In total, under the LTM, only 16 among 480 instances (3.3%) cannot be solved by setting **Default** within the given time limit while 318 among 1440 instances (22.1%) cannot be solved by the same setting under the ICM.

Table 7: Performance improvement through applying the SCNA and INA (for the IMP under the LTM).

	Default				**SCNA**					**INA**				
Network	#S	T	#N	#C	#S	T	PT	#N	#C	#S	T	PT	#N	#C
MSG	**60**	18.9	10	4240	**60**	18.1	0.4	10	4240	**60**	**17.9**	0.8	9	4167
GNU	**60**	54.1	1	1818	**60**	50.8	2.7	1	1818	**60**	**50.7**	6.6	1	1853
HEP	**60**	87.1	0	2664	**60**	78.0	3.0	0	2664	**60**	**52.4**	10.1	0	2557
ENRON	**60**	186.8	0	2048	**60**	**167.0**	8.0	0	2048	**60**	170.3	32.4	0	1985
FACEBOOK	**49**	1500.1	26	6587	**49**	1313.0	15.9	27	6698	**49**	**1303.2**	23.1	27	6749
DEEZER	**60**	292.2	0	1020	**60**	**261.9**	18.6	0	1020	**60**	272.6	31.1	0	1020
TWITTER	55	1294.3	8	4995	56	**1179.1**	23.2	8	5045	**58**	1181.3	33.9	8	5126
EPINIONS	**60**	822.1	0	2441	**60**	748.8	33.5	0	2441	**60**	**635.4**	66.8	0	2411
TOTAL	464	232.4	2	2769	465	211.6	9.2	2	2778	**467**	**198.4**	17.6	2	2757

6.4 Selection of parameter MaxReacSize

As it has been mentioned in subsection 5.1, **MaxReacSize** is a parameter to achieve a trade-off between the effectiveness and efficiency of implementing the INA: the larger the parameter **MaxReacSize**, the more isomorphic nodes that might be identified and the higher the computational complexity. Therefore, in this subsection, we compare the performance of different selections of parameter **MaxReacSize**. Tables 8 and 9 report the computational results for the IMP under the ICM and LTM, respectively. For simplicity, we only report the results for the case $|\Omega| = 1000$ (for the ICM, we only report the results for the case $p = 0.1$). In the two tables, we use M_0 to represent the average memory consumption (in GB) of storing all reachability sets after removing those detected by the SCNA, and M to denote the average memory consumption (in GB) of only storing the reachability sets with the size restriction. Instead of storing the reachability sets to obtain the required memories M_0 and M (which can be potentially very large on large-scale networks), we calculate the total number of elements of the stored reachability sets and convert it to the needed memory size. In our experiments, we set the size restriction **MaxReacSize** $= 2, 4, 8, 1000, |\mathcal{V}|$, respectively. Notice that when **MaxReacSize** $= |\mathcal{V}|$, we implement the INA without any size restriction on the reachability sets. In addition, we use T_{INA} to denote the average runtime in seconds of implementing the INA, and ΔZ_{INA} to denote the average percentage of variables $\boldsymbol{z}$ that can be eliminated by the INA (it does not include those eliminated by the SCNA). Here, "–" in Table 8 (under column **MaxReacSize** $= |\mathcal{V}|$) indicates that due to the limited memory, we were not able to construct the whole hashing table to implement the INA.

For the ICM, Table 8 shows that it requires a prohibitively large memory M_0 to store all the reachability sets. However, when restricting the size of the considered reachability sets to a small value of **MaxReacSize**, the memory overhead significantly reduces; see column M in Table 8. In addition, with the increasing value of **MaxReacSize**, the improvement on the percentage of the eliminated variables ΔZ_{INA} becomes smaller and smaller. Indeed, for networks MSG, GNU, HEP, and ENRON, the proposed algorithm with **MaxReacSize** $= 8$ can identify almost all isomorphic nodes as those with **MaxReacSize** $= |\mathcal{V}|$. For the other four networks, setting **MaxReacSize** $= 8$ enables to identify almost the same amount of isomorphic nodes as those obtained by setting **MaxReacSize** $= 1000$.

We now discuss the results for the IMP under the LTM in Table 9. On one hand, for the IMP under the LTM, since there exists at most one incoming arc for each node in each live-arc graph, the sizes of the reachability sets are likely to be smaller than those

Table 8: Comparison of effectiveness of the INA with different parameters **MaxReacSize** (for the IMP under the ICM).

MaxReacSize		2			4			8			1000			$\lvert\mathcal{V}\rvert$		
Network	M_0	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}
MSG	1.0	<0.1	<0.1	7.1%	<0.1	<0.1	7.5%	<0.1	<0.1	7.5%	1.0	1.7	7.5%	1.0	1.7	7.5%
GNU	8.9	<0.1	1.6	22.5%	<0.1	4.2	29.8%	<0.1	6.3	29.8%	0.1	7.3	29.8%	8.9	20.1	29.8%
HEP	5.5	<0.1	2.3	44.3%	<0.1	4.0	63.1%	<0.1	5.1	66.4%	0.1	5.4	66.6%	5.5	12.7	66.6%
ENRON	147.9	<0.1	3.3	27.1%	<0.1	5.3	35.2%	<0.1	5.8	35.7%	0.1	6.1	35.7%	147.9	170.6	35.7%
FACEBOOK	753.1	<0.1	2.9	15.6%	<0.1	4.0	17.9%	<0.1	4.1	17.9%	<0.1	4.2	17.9%	753.1	–	–
DEEZER	911.7	<0.1	6.7	21.4%	0.1	11.5	25.4%	0.1	12.1	25.4%	0.1	12.9	25.4%	911.7	–	–
TWITTER	1520.0	0.1	11.6	23.0%	0.1	20.8	27.9%	0.1	24.6	28.1%	0.3	27.8	28.1%	1520.0	–	–
EPINIONS	662.5	0.1	11.2	30.8%	0.1	15.5	34.9%	0.1	16.4	35.0%	0.1	17.2	35.0%	662.5	–	–

Table 9: Comparison of effectiveness of the INA with different parameters **MaxReacSize** (for the IMP under the LTM).

MaxReacSize		2			4			8			1000			$\lvert\mathcal{V}\rvert$		
Network	M_0	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}	M	T_{INA}	ΔZ_{INA}
MSG	0.1	<0.1	0.1	2.6%	<0.1	1.0	10.7%	<0.1	5.5	11.2%	0.1	12.6	11.2%	0.1	12.6	11.2%
GNU	0.4	<0.1	1.1	5.8%	0.1	10.3	21.6%	0.1	40.3	22.5%	0.4	71.8	22.5%	0.4	72.0	22.5%
HEP	0.2	<0.1	3.4	22.2%	0.1	15.8	63.3%	0.2	37.6	71.7%	0.2	45.9	71.7%	0.2	45.9	71.7%
ENRON	1.3	<0.1	3.5	6.6%	0.2	64.5	24.8%	0.4	162.8	26.7%	1.3	359.0	26.7%	1.3	359.2	26.7%
FACEBOOK	5.3	<0.1	2.4	2.1%	0.1	17.3	4.5%	0.3	62.3	4.6%	5.3	348.4	4.6%	5.3	348.5	4.6%
DEEZER	3.5	<0.1	4.1	3.4%	0.1	31.6	7.5%	0.4	118.2	7.7%	3.5	359.7	7.7%	3.5	361.5	7.7%
TWITTER	5.0	<0.1	2.5	1.5%	0.1	24.3	5.4%	0.6	136.7	6.1%	5.0	602.1	6.1%	5.0	619.7	6.1%
EPINIONS	3.0	0.2	23.5	19.5%	0.5	82.0	32.1%	1.0	312.2	32.9%	3.0	708.5	32.9%	3.0	713.8	32.9%

for the IMP under the ICM. This leads to a smaller total memory consumption M_0 and a larger memory consumption M when restricting **MaxReacSize** to a small value, as compared to those for the IMP under the ICM. As a result, the computational overhead of implementing the INA is very high, even for a relatively small **MaxReacSize** (e.g., **MaxReacSize** $= 8$). In addition, in analogy to the IMP under the ICM, with a small value of parameter **MaxReacSize**, the proposed algorithm can identify almost the same amount of isomorphic nodes as the case **MaxReacSize** $= |\mathcal{V}|$. Therefore, for the IMP under the LTM, we choose **MaxReacSize** $= 4$ in the implementation of the INA to achieve a trade-off between the performance and the time complexity.

7 Extensions

In this section, we investigate a generalization of the IMP (3), which arises from many existing applications including the IMP and its variants [42, 43, 54, 55], and discuss how the proposed SCNA and INA can be extended to this generalization.

The considered generalization is of the form

$$\begin{aligned} \max_{\boldsymbol{y},\,\boldsymbol{z}}\ & f(\boldsymbol{y},\boldsymbol{z}) \\ \text{s.t. }\ & (3b), (3d), (3e), \\ & \boldsymbol{y}\in\mathcal{Y},\ \boldsymbol{z}\in\mathcal{Z}, \end{aligned} \tag{26}$$

where $f\colon \{0,1\}^{|\mathcal{V}|}\times\{0,1\}^{|\mathcal{V}||\Omega|}\to\mathbb{R}$, $\mathcal{Y}\subseteq\mathbb{R}^{|\mathcal{V}|}$, and $\mathcal{Z}\subseteq\mathbb{R}^{|\mathcal{V}||\Omega|}$. Similar to the IMP (3), problem (26) is built upon a finite number of live-arc graphs $\mathcal{G}^\omega=(\mathcal{V},\mathcal{A}^\omega)$, $\omega\in\Omega$. However, in contrast to the IMP (3), problem (26) can flexibly allow any objective function and any constraint in sets $\mathcal{Y}$ and $\mathcal{Z}$. Indeed, the IMP (3) can be seen as a special case of problem (26) where $f(\boldsymbol{y},\boldsymbol{z})=\sum_{\omega\in\Omega}p^\omega\sum_{i\in\mathcal{V}}z_i^\omega$, $\mathcal{Y}=\left\{\boldsymbol{y}\in\mathbb{R}^{|\mathcal{V}|}:\sum_{j\in\mathcal{V}}y_j\le K\right\}$, and $\mathcal{Z}=\mathbb{R}^{|\mathcal{V}||\Omega|}$. Due to the flexibility, various variants of the IMP can also be seen as special cases of (26). For instance, by choosing $\mathcal{Y}=\left\{\boldsymbol{y}\in\mathbb{R}^{|\mathcal{V}|}:\sum_{j\in\mathcal{V}}c_jy_j\le B\right\}$ and the same $\mathcal{Z}$ and $f(y,z)$ as that of the IMP, problem (26) reduces to the *budgeted influence maximization problem* (BIMP) studied in [43]. Here c_j, $j\in\mathcal{V}$, is the cost of choosing node j as a seed node and B is the total budget. We next present another two special cases of problem (26).

- The *seed minimization problem* (SMP) [42]. In this problem, $f(\boldsymbol{y},\boldsymbol{z})=-\sum_{j\in\mathcal{V}}y_j$, $\mathcal{Y}=\mathbb{R}^{|\mathcal{V}|}$, and $\mathcal{Z}=\left\{\boldsymbol{z}\in\mathbb{R}^{|\mathcal{V}||\Omega|}:\sum_{\omega\in\Omega}p^\omega\sum_{i\in\mathcal{V}}z_i^\omega\ge D\right\}$ ($D\in\mathbb{R}_{++}$). This problem can be seen as a dual form of the IMP (3), which minimizes the number of seed nodes with an expected influence coverage D in a social network.
- The *seed minimization problem with probabilistic influence coverage guarantee* (SMP-PICG) [54, 55]. In this problem, $f(\boldsymbol{y},\boldsymbol{z})=-\sum_{j\in\mathcal{V}}y_j$, $\mathcal{Y}=\mathbb{R}^{|\mathcal{V}|}$, $\mathcal{Z}=\mathrm{Proj}_{\boldsymbol{z}}(\mathcal{W})$ where $\mathcal{W}=\left\{(\boldsymbol{z},\boldsymbol{\xi})\in\mathbb{R}^{|\mathcal{V}||\Omega|}\times\{0,1\}^{|\Omega|}:\sum_{i\in\mathcal{V}}z_i^\omega\ge D\xi^\omega,\ \forall\,\omega\in\Omega,\ \sum_{\omega\in\Omega}p^\omega\xi^\omega\ge 1-\varepsilon\right\}$ ($\varepsilon\in(0,1)$ is the confidence level). Instead of ensuring an expected influence coverage threshold D, the problem requires to influence at least D nodes with a probability at least $1-\varepsilon$.

We next discuss how the proposed SCNA and INA are applied to problem (26). To proceed, we need the following two realistic assumptions.

(i) $f(\boldsymbol{y},\boldsymbol{z})$ is nondecreasing with respect to variables $\boldsymbol{z}$, i.e., if $\boldsymbol{z}^1,\boldsymbol{z}^2\in\mathcal{Z}$ and $\boldsymbol{z}^1\le\boldsymbol{z}^2$, then $f(\boldsymbol{y},\boldsymbol{z}^1)\le f(\boldsymbol{y},\boldsymbol{z}^2)$.

(ii) Set $\mathcal{Z}$ is up-monotone, i.e., if $\boldsymbol{z}^1\in\mathcal{Z}$ and $\boldsymbol{z}^1\le\boldsymbol{z}^2$, then $\boldsymbol{z}^2\in\mathcal{Z}$ as well (such a set is also called a reverse normal set [52]).

The two assumptions imply that when node i is reachable in scenario ω from some seed nodes (i.e., $\sum_{j\in\mathcal{R}(\mathcal{G}^\omega,i)} y_j \geq 1$), activating node i in scenario ω (i.e., setting $z_i^\omega := 1$) provides a better solution for the decision maker while does not violate his/her requirement. It can be easily verified that for the IMP, BIMP, SMP, and SMPPICG, the two assumptions are satisfied.

Proposition 7.1. *Suppose that problem (26), with assumptions (i) and (ii), has an optimal solution. Then there must exist an optimal solution $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ such that (8) holds.*

Proof. Let $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ be an optimal solution of problem (26). Suppose that (8) does not hold for some $\omega_0 \in \Omega$ and $i_0 \in \mathcal{V}$. Then we must have $\bar{z}_{i_0}^{\omega_0} = 0$ and $\sum_{j\in\mathcal{R}(\mathcal{G}^{\omega_0},i_0)} \bar{y}_j \geq 1$. Setting $\bar{z}_{i_0}^{\omega_0} := 1$, we obtain a new point $(\bar{\boldsymbol{y}}, \boldsymbol{z}')$. By assumption (ii), $\boldsymbol{z}' \in \mathcal{Z}$, and by $\sum_{j\in\mathcal{R}(\mathcal{G}^{\omega_0},i_0)} \bar{y}_j \geq 1$, constraints (3b) hold at point $(\bar{\boldsymbol{y}}, \boldsymbol{z}')$. This implies that $(\bar{\boldsymbol{y}}, \boldsymbol{z}')$ is a feasible solution of problem (26). Moreover, by assumption (i), $f(\bar{\boldsymbol{y}}, \boldsymbol{z}') \geq f(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$, indicating that $(\bar{\boldsymbol{y}}, \boldsymbol{z}')$ must also be an optimal solution of problem (26). Recursively using the above argument, the statement follows. □

By Proposition 7.1, if $\mathcal{R}(\mathcal{G}^\omega, i_1) = \mathcal{R}(\mathcal{G}^\eta, i_2)$, we can set $z_{i_1}^\omega := z_{i_2}^\eta$ in problem (26) (with the two realistic assumptions (i) and (ii)). As a result, the proposed SCNA and INA can also be applied to problem (26) to reduce the problem size and improve the solution efficiency.

8 Concluding remarks

In this paper, we proposed two new presolving methods, called the SCNA and INA, and integrated them into the BD algorithm to solve the IMP. The SCNA enables to build an SMCLP formulation for the considered problem based on the (potentially) much more compact live-arc graphs, which are obtained by aggregating strongly connected nodes in the original live-arc graphs. The INA further reduces the problem size of the SMCLP formulation by aggregating isomorphic nodes among different live-arc graphs. We provided a theoretical analysis on two special cases of the IMP to show the strength of the proposed SCNA and INA in reducing the problem size of the SMCLP formulation. Furthermore, with the SCNA and INA, a (potentially) much faster separation procedure for the Benders optimality cuts is developed, which plays a crucial role in speeding up the BD algorithm. We have performed extensive experiments to analyze the performance impact of the proposed SCNA and INA on solving the IMP with real-world social networks. Computational results show that the proposed SCNA and INA can effectively reduce the problem size and hence improve the performance of using the BD algorithm to solve the IMP, especially for problems with large and dense networks and large numbers of scenarios. We also studied a generalization of the IMP and demonstrated that the proposed presolving methods are applicable to this generalization under some realistic assumptions.

There still exist some instances where the proposed SCNA and INA cannot effectively reduce the problem size of the SMCLP formulation. Indeed, in section 2 of [9], we have provided a worst-case example showing that the percentage of the eliminated variables in the SMCLP formulation of the IMP tends to zero with a probability tending to one. Consequently, it is interesting to develop more powerful presolving methods for solving the IMP. In addition, it also deserves to investigate whether the proposed presolving methods are computationally effective in solving other variants of the IMP [42, 43, 54, 55].

Acknowledgments

The works of S.-J. Chen and Y.-H. Dai were supported in part by the Chinese NSF grants (Nos. 12021001, 11991021, 11991020, and 11971372), the National Key R&D Program of China (Nos. 2021YFA1000300 and 2021YFA1000301), and the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDA27000000). The work of W.-K. Chen was supported in part by the Chinese NSF grants (No. 12101048) and Beijing Institute of Technology Research Fund Program for Young Scholars. The work of J.-H. Yuan and H.-S. Zhang were supported in part by the Chinese NSF grants (No. 12171052).

References

[1] T. Achterberg. *Constraint Integer Programming*. Ph.D. thesis, Technische Universität Berlin, 2007.

[2] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger. Presolve reductions in mixed integer programming. *INFORMS Journal on Computing*, 32(2):473–506, 2020.

[3] S. Akrouf, L. Meriem, B. Yahia, and M. N. Eddine. Social network analysis and information propagation: A case study using Flickr and YouTube networks. *International Journal of Future Computer and Communication*, 2(3):246–252, 2013.

[4] N. Alon, I. Gamzu, and M. Tennenholtz. Optimizing budget allocation among channels and influencers. In *Proceedings of the 21st International Conference on World Wide Web*, pages 381–388, 2012.

[5] R. Berardo. Bridging and bonding capital in two-mode collaboration networks. *Policy Studies Journal*, 42(2):197–225, 2014.

[6] R. Borndörfer. *Aspects of Set Packing, Partitioning and Covering*. Ph.D. thesis, Technische Universität Berlin, 1998.

[7] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, pages 665–674, 2011.

[8] J. Cao and M. Olvera-Cravioto. Connectivity of a general class of inhomogeneous random digraphs. *Random Structures & Algorithms*, 56(3):722–774, 2020.

[9] S.-J. Chen, W.-K. Chen, Y.-H. Dai, J.-H. Yuan, and H.-S. Zhang. A companion technical report of "Efficient presolving methods for the influence maximization problem in social networks". Technical report, 2022. URL https://drive.google.com/file/d/1vmgRBBgwp-zs2rCysBQw-JPcDg3YIXa9/view?usp=sharing.

[10] 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*, pages 199–208, 2009.

[11] W. Chen, C. Wang, and Y. 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*, pages 1029–1038, 2010.

[12] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 88–97, 2010.

[13] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[14] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng. StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 509–518, 2013.

[15] R. L. Church. COBRA: A new formulation of the classic p-median location problem. *Annals of Operations Research*, 122(1):103–120, 2003.

[16] CPLEX. https://www.ibm.com/analytics/cplex-optimizer. 2022.

[17] N. Detering, T. Meyer-Brandis, and K. Panagiotou. Bootstrap percolation in directed and inhomogeneous random graphs. *The Electronic Journal of Combinatorics*, 26(3), 2019.

[18] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[19] G. Ergün. Human sexual contact network as a bipartite graph. *Physica A: Statistical Mechanics and its Applications*, 308(1):483–488, 2002.

[20] M. Fischetti, M. Kahr, M. Leitner, M. Monaci, and M. Ruthmair. Least cost influence propagation in (social) networks. *Mathematical Programming*, 170(1):293–325, 2018.

[21] S. Galhotra, A. Arora, and S. Roy. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *Proceedings of the 2016 International Conference on Management of Data*, pages 743–758, 2016.

[22] A. J. Graham and D. A. Pike. A note on thresholds and connectivity in random directed graphs. *Atlantic Electronic Journal of Mathematics*, 3(1):1–5, 2008.

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

[24] E. Güney, M. Leitner, M. Ruthmair, and M. Sinnl. Large-scale influence maximization via maximal covering location. *European Journal of Operational Research*, 289(1):144–164, 2021.

[25] D. Günneç, S. Raghavan, and R. Zhang. Least-cost influence maximization on social networks. *INFORMS Journal on Computing*, 32(2):289–302, 2020.

[26] D. Hatano, T. Fukunaga, and K.-i. Kawarabayashi. Adaptive budget allocation for maximizing influence of advertisements. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 3600–3608, 2016.

[27] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474, 2012.

[28] S. Heinz, J. Schulz, and J. C. Beck. Using dual presolving reductions to reformulate cumulative constraints. *Constraints*, 18(2):166–201, 2013.

[29] M. Kahr, M. Leitner, M. Ruthmair, and M. Sinnl. Benders decomposition for competitive influence maximization in (social) networks. *Omega*, 100:102264, 2021.

[30] M. Kahr, M. Leitner, and I. Ljubić. The impact of passive social media users in (competitive) influence maximization. 2022. URL http://www.optimization-online.org/DB_FILE/2022/01/8777.pdf.

[31] D. Kempe, J. Kleinberg, and É. 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, 2003.

[32] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *Proceedings of the 10th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 259–271, 2006.

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

[34] J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. In *Proceedings of the 26th the Annual Conference on Neural Information Processing*, pages 539–547, 2012.

[35] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429, 2007.

[36] 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(1):29–123, 2009.

[37] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the 28th SIGCHI Conference on Human Factors in Computing Systems*, pages 1361–1370, 2010.

[38] M. Li, X. Wang, K. Gao, and S. Zhang. A survey on information diffusion in online social networks: Models and methods. *Information*, 8(4):118, 2017.

[39] X. Li, J. D. Smith, T. N. Dinh, and M. T. Thai. TipTop: (Almost) exact solutions for influence maximization in billion-scale networks. *IEEE/ACM Transactions on Networking*, 27(2):649–661, 2019.

[40] 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(10):1852–1872, 2018.

[41] I. Ljubić, P. Putz, and J.-J. Salazar-González. Exact approaches to the single-source network loading problem. *Networks*, 59(1):89–106, 2012.

[42] C. Long and R. C.-W. Wong. Minimizing seed set for viral marketing. In *Proceedings of the 11th IEEE International Conference on Data Mining*, pages 427–436, 2011.

[43] H. Nguyen and R. Zheng. On budgeted influence maximization in social networks. *IEEE Journal on Selected Areas in Communications*, 31(6):1084–1094, 2013.

[44] 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(5):911–932, 2009.

[45] S. Raghavan and R. Zhang. A branch-and-cut approach for the weighted target set selection problem on social networks. *INFORMS Journal on Optimization*, 1(4):304–322, 2019.

[46] M. Ripeanu and I. Foster. Mapping the Gnutella network: Macroscopic properties of large-scale peer-to-peer systems. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, pages 85–93, 2002.

[47] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton. GEMSEC: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 65–72, 2019.

[48] M. Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981.

[49] T. Soma, N. Kakimura, K. Inaba, and K.-i. Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *Proceedings of the 31st International Conference on Machine Learning*, pages 351–359, 2014.

[50] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pages 75–86, 2014.

[51] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554, 2015.

[52] H. Tuy. Monotonic optimization: Problems and solution approaches. *SIAM Journal on Optimization*, 11(2):464–494, 2000.

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

[54] H.-H. Wu and S. Küçükyavuz. Probabilistic partial set covering with an oracle for chance constraints. *SIAM Journal on Optimization*, 29(1):690–718, 2019.

[55] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1306–1315, 2014.

A Appendix

Proof of Proposition 4.3

Proof. Clearly, if $K \geq |\mathcal{M}| + |\mathcal{N}|$, point $(\boldsymbol{y}, \boldsymbol{z}) = (\boldsymbol{e}, \boldsymbol{e})$ is optimal for formulation (14) and its LP relaxation, where $\boldsymbol{e}$ is an all-ones vector with appropriate dimension. As a result, the statement follows. Therefore, in the following, we consider the case $K < |\mathcal{M}| + |\mathcal{N}|$.

Let $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ be an optimal solution of the LP relaxation of formulation (14). If there exists some $i_0 \in \mathcal{M}$ and $j_0 \in \mathcal{N}$ such that $\bar{y}_{i_0} < 1$ and $\bar{y}_{j_0} > 0$, then we can construct a new point $(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ as follows:

$\hat{y}_{i_0} := \bar{y}_{i_0} + \varepsilon,\ \hat{y}_{j_0} := \bar{y}_{j_0} - \varepsilon,$ and $\hat{y}_i := \bar{y}_i$ for $i \in \mathcal{M} \cup \mathcal{N} \backslash \{i_0, j_0\}$;

$\hat{z}_{ij_0} := \max\{\bar{z}_{ij_0} - \varepsilon, 0\}$ for $(i, j_0) \in \mathcal{A}'$ and $\hat{z}_{ij} := \bar{z}_{ij}$ for $(i, j) \in \mathcal{A}'$ with $j \neq j_0$,

where $\varepsilon > 0$ is a sufficiently small value. It is easy to see that point $(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ is also feasible for the LP relaxation of formulation (14). Moreover, point $(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ must be optimal since

$$
\begin{aligned}
&\sum_{i \in \mathcal{M} \cup \mathcal{N}} s_i \hat{y}_i + \sum_{(i,j) \in \mathcal{A}'} c_{ij} \hat{z}_{ij} - \left(\sum_{i \in \mathcal{M} \cup \mathcal{N}} s_i \bar{y}_i + \sum_{(i,j) \in \mathcal{A}'} c_{ij} \bar{z}_{ij} \right) \\
=& s_{i_0}(\bar{y}_{i_0} + \varepsilon) + s_{j_0}(\bar{y}_{j_0} - \varepsilon) + \sum_{i\,:\,(i,j_0) \in \mathcal{A}'} c_{ij_0} \max\{\bar{z}_{ij_0} - \varepsilon, 0\} \\
&\qquad - s_{i_0} \bar{y}_{i_0} - s_{j_0} \bar{y}_{j_0} - \sum_{i\,:\,(i,j_0) \in \mathcal{A}'} c_{ij_0} \bar{z}_{ij_0} \\
=& s_{i_0} \varepsilon - s_{j_0} \varepsilon + \sum_{i\,:\,(i,j_0) \in \mathcal{A}'} c_{ij_0} \max\{-\varepsilon, -\bar{z}_{ij_0}\} \\
\geq& s_{i_0} \varepsilon - s_{j_0} \varepsilon - \sum_{i\,:\,(i,j_0) \in \mathcal{A}'} c_{ij_0} \varepsilon = \varepsilon \left(s_{i_0} - s_{j_0} - \sum_{i\,:\,(i,j_0) \in \mathcal{A}'} c_{ij_0} \right) = 0,
\end{aligned}
$$

where the last equality follows from Remark 4.2. Recursively applying the above argument, we will obtain an optimal solution $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})$ of the LP relaxation of formulation (14) fulfilling:

($\star$) if $\tilde{y}_{j_0} > 0$ for some $j_0 \in \mathcal{N}$, then $\tilde{y}_i = 1$ for all $i \in \mathcal{M}$ must hold.

Furthermore, we can, without loss of generality, assume the followings on point $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})$.

1) $\sum_{i\in\mathcal{M}\cup\mathcal{N}}\tilde{y}_i = K$. Otherwise, we can increase $\tilde{y}_i$, with $\tilde{y}_i < 1$, without decreasing the objective value (as $\sum_{i\in\mathcal{M}\cup\mathcal{N}}\tilde{y}_i < K < |\mathcal{M}| + |\mathcal{N}|$);

2) $\tilde{z}_{ij} = \min\{\tilde{y}_i + \tilde{y}_j, 1\}$ for all $(i,j) \in \mathcal{A}'$. Otherwise, we can increase $\tilde{z}_{ij}$, with $\tilde{z}_{ij} < \min\{\tilde{y}_i + \tilde{y}_j, 1\}$, without decreasing the objective value.

Together with $(\star)$ and 1), point $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})$ must satisfy the followings

(i) if $K \geq |\mathcal{M}|$, then $\tilde{y}_i = 1$ for all $i \in \mathcal{M}$; and

(ii) if $K < |\mathcal{M}|$, then $\tilde{y}_j = 0$ for all $j \in \mathcal{N}$.

We next prove the statement in the proposition by treating cases (i) and (ii) separately.

(i) In this case, $\tilde{y}_i = 1$ for all $i \in \mathcal{M}$ and by 2), $\tilde{z}_{ij} = 1$ for all $(i,j) \in \mathcal{A}'$. Setting $y_i := 1$ for all $i \in \mathcal{M}$ and $z_{ij} := 1$ for all $(i,j) \in \mathcal{A}'$, the LP relaxation of formulation (14) reduces to

$$\max_{\boldsymbol{y}} \left\{ \sum_{j\in\mathcal{N}} s_j y_j + |\mathcal{M}| + \sum_{(i,j)\in\mathcal{A}'} c_{ij} \ : \ \sum_{j\in\mathcal{N}} y_j = K - |\mathcal{M}|, \ y_j \in [0,1], \ \forall\ j \in \mathcal{N} \right\}. \tag{27}$$

Then point $\{\tilde{y}_j\}_{j\in\mathcal{N}}$ must be optimal to formulation (27). On the other hand, suppose that $s_{j_1} \geq \cdots \geq s_{j_{|\mathcal{N}|}}$ where $\{j_1, \ldots, j_{|\mathcal{N}|}\} = \mathcal{N}$. It is easy to show that point $\{y'_j\}_{j\in\mathcal{N}}$ is also optimal to formulation (27) where $y'_{j_\tau} = 1$ for $\tau = 1, \ldots, K - |\mathcal{M}|$ and $y'_{j_\tau} = 0$ otherwise. We next extend point $\{y'_j\}_{j\in\mathcal{N}}$ to a higher dimensional point $(\boldsymbol{y}', \boldsymbol{z}') \in \{0,1\}^{|\mathcal{M}|+|\mathcal{N}|} \times \{0,1\}^{|\mathcal{A}'|}$ by additionally setting $y'_i := 1$ for all $i \in \mathcal{M}$ and $z'_{ij} := 1$ for all $(i,j) \in \mathcal{A}'$. The 0-1 point $(\boldsymbol{y}', \boldsymbol{z}')$ must be optimal to formulation (14) and its LP relaxation. This implies that the LP relaxation of formulation (14) is tight and formulation (14) is strongly polynomial-time solvable for this case.

(ii) In this case, $\tilde{y}_j = 0$ for all $j \in \mathcal{N}$ and by 2), $\tilde{z}_{ij} = \tilde{y}_i$ for all $(i,j) \in \mathcal{A}'$. Setting $y_i := 0$ for all $i \in \mathcal{N}$ and $z_{ij} := y_i$ for all $(i,j) \in \mathcal{A}'$, the LP relaxation of formulation (14) reduces to

$$\max_{\boldsymbol{y}} \left\{ \sum_{i\in\mathcal{M}} \left(1 + \sum_{j\,:\,(i,j)\in\mathcal{A}'} c_{ij} \right) y_i \ : \ \sum_{i\in\mathcal{M}} y_i = K, \ y_i \in [0,1], \ \forall\ i \in \mathcal{M} \right\}. \tag{28}$$

Then point $\{\tilde{y}_i\}_{i\in\mathcal{M}}$ must be optimal to formulation (28). Using a similar argument in case (i), we can also prove the statement in this case. □

Proof of Proposition 5.1

Proof. Note that if only the SCNA is applied, formulation (9) reduces to formulation (6); and if neither the SCNA nor the INA is applied, formulation (9) reduces to formulation (3). Hence, we shall prove case (i) by showing that the Benders optimality cut based on formulation (3) is equivalent to the one based on formulation (6). First, for formulation (3), the Benders optimality cut is given by

$$\varphi^\omega \leq \sum_{i\in\mathcal{V}} \left(\bar{\lambda}_i^\omega \sum_{j\in\mathcal{R}(\mathcal{G}^\omega, i)} y_j + \bar{\eta}_i^\omega \right) \tag{29}$$

where for each $i \in \mathcal{V}$, $(\bar{\lambda}_i^\omega, \bar{\eta}_i^\omega) = (0, p^\omega)$ if $\sum_{j\in\mathcal{R}(\mathcal{G}^\omega,i)} \bar{y}_j \geq 1$; and $(\bar{\lambda}_i^\omega, \bar{\eta}_i^\omega) = (p^\omega, 0)$ otherwise. Note that for all nodes in a given SCC $\mathcal{SC}_u^\omega$ of $\mathcal{G}^\omega$, their reachability sets are identical and equal to $\mathcal{R}(\mathcal{G}^\omega, \mathcal{SC}_u^\omega)$. As a result, $(\bar{\lambda}_i^\omega, \bar{\eta}_i^\omega)$ for all $i \in \mathcal{SC}_u^\omega$ must also be identical, denoted by $(\bar{\lambda}_u^\omega, \bar{\eta}_u^\omega)$. Hence, inequality (29) can be rewritten as

$$\varphi^\omega \leq \sum_{u\in\bar{\mathcal{V}}^\omega} \left(|\mathcal{SC}_u^\omega| \bar{\lambda}_u^\omega \sum_{j\in\mathcal{R}(\mathcal{G}^\omega, \mathcal{SC}_u^\omega)} y_j + |\mathcal{SC}_u^\omega| \bar{\eta}_u^\omega \right). \tag{30}$$

As for formulation (6), the Benders optimality cut reads

$$\varphi^\omega \leq \sum_{u\in\bar{\mathcal{V}}^\omega} \left(\bar{\alpha}_u^\omega \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega, u)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_u^\omega \right), \tag{31}$$

where for each $u \in \bar{\mathcal{V}}^\omega$, $(\bar{\alpha}_u^\omega, \bar{\beta}_u^\omega) = (0, p^\omega|\mathcal{SC}_u^\omega|)$ if $\sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} \bar{y}(\mathcal{SC}_v^\omega) \geq 1$; and $(\bar{\alpha}_u^\omega, \bar{\beta}_u^\omega) = (p^\omega|\mathcal{SC}_u^\omega|, 0)$ otherwise. Together with equation (4) and $\bar{y}(\mathcal{SC}_v^\omega) = \sum_{j\in\mathcal{SC}_v^\omega} \bar{y}_j$, we have $(\bar{\alpha}_u^\omega, \bar{\beta}_u^\omega) = (|\mathcal{SC}_u^\omega|\bar{\lambda}_u^\omega, |\mathcal{SC}_u^\omega|\bar{\eta}_u^\omega)$, and hence the two Benders optimality cuts (30) and (31) are equivalent. This proves the case (i).

We next prove case (ii). Indeed, for two different scenarios ω and η, the corresponding Benders optimality cuts based on formulation (6) are given by

$$\varphi^\omega \leq \sum_{u\in\bar{\mathcal{V}}^\omega} \left(\bar{\alpha}_u^\omega \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_u^\omega \right) \text{ and} \tag{32}$$

$$\varphi^\eta \leq \sum_{u\in\bar{\mathcal{V}}^\eta} \left(\bar{\alpha}_u^\eta \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\eta,u)} y(\mathcal{SC}_v^\eta) + \bar{\beta}_u^\eta \right). \tag{33}$$

Suppose that node u_1 in graph $\bar{\mathcal{G}}^\omega$ is isomorphic to node u_2 in graph $\bar{\mathcal{G}}^\eta$. By applying the INA, we can remove, for example, variable $z_{u_1}^\omega$ and the corresponding reachability constraint in (5) from formulation (6). The new objective coefficient $f_{u_2}^\eta$ is set to the sum of the old objective coefficients of variables $z_{u_1}^\omega$ and $z_{u_2}^\eta$. As a result, the term $\bar{\alpha}_{u_1}^\omega \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u_1)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_{u_1}^\omega$ in Benders optimality cut (32) will be incorporated into Benders optimality cut (33), leading to two new Benders optimality cuts

$$\varphi^\omega \leq \sum_{u\in\bar{\mathcal{V}}^\omega\setminus\{u_1\}} \left(\bar{\alpha}_u^\omega \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_u^\omega \right) \text{ and} \tag{34}$$

$$\varphi^\eta \leq \sum_{u\in\bar{\mathcal{V}}^\eta} \left(\bar{\alpha}_u^\eta \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\eta,u)} y(\mathcal{SC}_v^\eta) + \bar{\beta}_u^\eta \right) + \left(\bar{\alpha}_{u_1}^\omega \sum_{v\in\mathcal{R}(\bar{\mathcal{G}}^\omega,u_1)} y(\mathcal{SC}_v^\omega) + \bar{\beta}_{u_1}^\omega \right). \tag{35}$$

Obviously, the new Benders optimality cuts (34) and (35) may be different from the old ones (32) and (33), respectively. □