In and Out: Optimizing Overall Interaction in Probabilistic Graphs under Clustering Constraints

Domenico Mandaglio
DIMES Dept., University of Calabria
Rende (CS), Italy
d.mandaglio@dimes.unical.it

Andrea Tagarelli
DIMES Dept., University of Calabria
Rende (CS), Italy
andrea.tagarelli@unical.it

Francesco Gullo
UniCredit, R&D Dept.
Rome, Italy
gullof@acm.org

ABSTRACT

We study two novel clustering problems in which the pairwise interactions between entities are characterized by probability distributions and conditioned by external factors within the environment where the entities interact. This covers any scenario where a set of actions can alter the entities' interaction behavior. In particular, we consider the case where the interaction conditioning factors can be modeled as cluster memberships of entities in a graph and the goal is to partition a set of entities such as to maximize the overall vertex interactions or, equivalently, minimize the loss of interactions in the graph. We show that both problems are **NP**-hard and they are equivalent in terms of optimality. However, we focus on the minimization formulation as it enables the possibility of devising both practical and efficient approximation algorithms and heuristics. Experimental evaluation of our algorithms, on both synthetic and real network datasets, has shown evidence of their meaningfulness as well as superiority with respect to competing methods, both in terms of effectiveness and efficiency.

CCS CONCEPTS

• **Mathematics of computing → Graph theory**; • **Information systems → Web searching and information discovery**.

KEYWORDS

interaction loss, correlation clustering, uncertain graphs

ACM Reference Format:
Domenico Mandaglio, Andrea Tagarelli, and Francesco Gullo. 2020. In and Out: Optimizing Overall Interaction in Probabilistic Graphs under Clustering Constraints. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403190

1 INTRODUCTION

Modeling and mining behavioral patterns of users of online as well as offline systems is central to enhance the user engagement and experience in the systems. In this regard, *uncertain graph models* are seen as a powerful tool to capture the inherent uncertainty in user behaviors into a representation of user interaction patterns [14]. A common way of modeling uncertainty in a graph, which we refer to in this work, is to associate each pair of (linked) users with a probability value that expresses the likelihood of *observing* and *quantifying* an interaction between the two users. In this regard, one important aspect is that the modeling of user interactions should also account for exogenous conditions or events that occur within the social environment where the users belong to, which indeed can significantly affect the users' interaction behaviors. For example, delivering a post on a user's page (e.g., Facebook wall) that contains a message of friend recommendation will likely favor or not a meeting between two users, and so their interactions. Intuitively, it is of high interest to identify proper settings of a network system and relating conditions that can maximize the overall user interactions within the system. In this work, we extend the uncertain graph modeling framework to capture *the dependency of interactions on conditioning factors*, in a network system. In particular, we focus on the case when *the interaction behaviors depend on a clustering of the set of users* in a graph, so that the probability of interaction between any two users varies depending on whether they belong to the same cluster or not. Modeling such interaction conditioning factors in terms of cluster memberships of users arises in several relevant application scenarios. Let us discuss on a couple of them.

Applications. Consider a social-media platform where users produce, exchange and consume content items. Each user is typically associated with a personal profile page. This acts also as an interface for the platform to deliver recommendations and advertisements to a target user u, including those contents produced by other users which u may interact with. The probability that an interaction between two users u and v will occur, in relation to a content item c possibly produced or endorsed by any of them, can depend on whether the two users have been informed or not about c through their corresponding homepages. Clearly, if a grouping of users into communities was available, the platform administrators would likely drive the attention of users towards contents that are produced by members of the same community, according to a homophily effect. On the other hand, any user may also want to seek for relevant contents and similar users outside the boundary of her community, which would also have the effect of mitigating information-bubble issues that may arise inside each community. In this regard, it would be strategic for the administrators to know which links to users and associated contents are worthy to be recommended within other users' pages, in order to incentivate the overall interactions across the platform.

Another application scenario corresponds to a team formation task for a collaborative system, like Wikipedia, where users should be grouped into teams to contribute in the editing of different parts

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403190

of a Wikipedia page. In this context, the likelihood of collaboration between any pair of users will vary in relation to their assignment to the same team. The goal becomes to partition the set of users into teams in order to maximize the total collaboration. The greater the overall collaboration is, the higher the contamination will be, and the probability of successfully accomplishing the task will increase.

Contributions. To the best of our knowledge, we are the first to address the problem of optimizing the overall interaction among a set of entities in a probabilistic graph, subject to the cluster memberships of the entities. In particular, our main contributions are:

- We define the Max-Interaction-Clustering and Min-Interaction-Loss-Clustering problems for graph entities whose interaction patterns depend on their cluster memberships (Section 3). We show that both problems are special instances of the well-studied correlation-clustering framework [2], and we delve into their theoretical properties and complexities.
- Although the two problems are equivalent in terms of optimality, we focus on the minimization problem, as it enables the use of more practical yet efficient algorithms inspired by correlation-clustering theory. To this purpose, we devise both approximation algorithms and heuristics for the Min-Interaction-loss-Clustering problem (Section 4).
- Experimental evaluation of our algorithms, on both synthetic and real network datasets, has shown evidence of their meaningfulness as well as superiority with respect to competing methods, both in terms of effectiveness and efficiency (Section 5).

2 RELATED WORK

Clustering uncertain graphs. The problem we tackle in this work is close to that of clustering uncertain graphs, which is to cluster vertices of a graph whose edges are assigned a probability of existence, according to some (possible-world) semantics [4, 8, 10, 11, 15–17]. A major difference is that our problem is more general than clustering uncertain graphs since the probabilities of interaction are affected by cluster memberships, which poses additional challenges that we address in this work. Further differences are that (*i*) the classic uncertain-graph model is a special case of interaction graph we consider in this work (where the probability distributions of interaction are binary), and (*ii*) existing methods for clustering uncertain graphs aim to maximize the intra-cluster connectivity and minimize the inter-cluster connectivity, whereas we seek clusterings such that both types of connectivity are maximized.

Community detection in signed graphs. In signed graphs, which have positive and negative signs as a property on the edges (e.g., trust vs. distrust relations), the problem of community detection is to produce a structure whereby many positive (resp. negative) links are observed within (resp. between) communities. For example, [21] considers a directed graph and solves the above problem by optimizing a new notion of modularity that combines linearly the contribution of positive and negative weights, and extending the Potts Model to incorporate negative links. The method in [9] also introduces a generalization of modularity that is able to deal with both positive and negative weights. That definition of modularity is a special case of the one defined in [21], as it can be obtained from the general definition of [21] by properly setting the parameters that control the balance between the importance of present and absent (positive and negative) edges within a community. [7] reformulates the Map Equation to measure the quality of partitions, known as Minimum Description Length (MDL), and extends Constant Potts Model (CPM) to collect a spectrum of partitions from highly simplified to detailed ones, by varying its parameter λ from zero to one. Based on these extensions, the community detection is carried out by minimizing MDL on λ-spectrum.

The aforementioned methods will be considered in our experiments (cf. Section 5), given the similarity in the requirements of within- and across-cluster interactions. Nonetheless, we remark that edges in a signed graph can have either positive or negative weight, while in our setting each edge is assigned a probability distribution of the interaction strength between two vertices.

Correlation clustering. Originally introduced by Bansal *et al.* [2], correlation clustering is, given a complete signed graph G where every pair of vertices is labeled either as positive or negative, partition the vertices of G so as to *either* minimize the number of negative pairs within the same cluster plus the positive pairs across different clusters, *or* maximize the positive pairs within the same cluster plus the negative pairs across different clusters. In Sections 3.1–3.2, we shall discuss the profound relation with our problem formulations.

3 PROBLEM DEFINITION

We are given a set of users who *interact* with each other, where "interaction" is meant here as referring to any, symmetric or reciprocated, relation between two users (e.g., commenting posts of each other, collaborating for a task, etc.). We assume that the strength of interaction between any two users is represented by a nonnegative real value, however the exact interaction strength is not known beforehand; rather, a set of possible strengths are given, each one being assigned a probability of corresponding to the actual strength. This scenario is here modeled by a probabilistic interaction graph, or simply *interaction graph*, we define as a key notion in this work.

Definition 1 (Probabilistic interaction graph). *A probabilistic interaction graph is a triple $\mathcal{G} = (V, E, P)$, with V set of vertices, $E \subseteq V \times V$ set of undirected edges, and $P = \{p_{uv}\}_{(u,v)\in E}$ set of probability distributions, each one defined on a domain $\mathcal{D}(p_{uv}) \subseteq \mathbb{R}_0^+$. For all $(u, v) \in E$ and all $x \in \mathcal{D}(p_{uv})$, $p_{uv}(x)$ is the probability that the strength of the interaction between u and v is equal to x. For any $(u, v) \notin E$, $p_{uv}(0) = 1$ and $p_{uv}(x) = 0$ for any $x \neq 0$.*

Given an interaction graph $\mathcal{G} = (V, E, P)$, a set $\{G = (V, E, w_G)\}_{G \sqsubseteq \mathcal{G}}$ of *deterministic graphs* can be derived as instances of $\mathcal{G}$ — following the literature on uncertain data/graphs [14], these are also called *worlds*. Every instance G that can be derived from $\mathcal{G}$, here denoted as $G \sqsubseteq \mathcal{G}$, is a weighted graph that is defined over the same sets V, E of $\mathcal{G}$, and whose weighting function $w_G : E \rightarrow \mathbb{R}_0^+$ assigns a weight to every edge $(u, v) \in E$ so that $w_G(u, v)$ is sampled from $p_{uv} \in P$, i.e., $w_G(u, v) \in \mathcal{D}(p_{uv})$. Note that, as for any $(u, v) \in E : \mathcal{D}(p_{uv}) \subseteq \mathbb{R}_0^+$, a possible world $G \sqsubseteq \mathcal{G}$ may contain edges (u, v) from $\mathcal{G}$ that do not exist in G, i.e., $w_G(u, v) = 0$.

Assuming independence between probability distributions — as usual in the literature on uncertain graphs [3, 4, 12–15, 17, 19] — the probability of a possible world $G = (V, E, w_G) \sqsubseteq \mathcal{G}$ is:

$$\Pr(G) = \prod_{(u,v)\in E} p_{uv}(w_G(u, v)). \qquad (1)$$

A key aspect in our study is that an interaction graph $\mathcal{G} = (V, E, P)$ can be provided in terms of two probabilistic graphs, say $\mathcal{G}^+$ and $\mathcal{G}^-$, both defined over V and E, such that the one accounts for interactions within the same clusters of vertices, and the other one for interactions between different clusters of vertices. To this end, let $C : V \longrightarrow \mathbb{N}$ denote an injective function that expresses the *cluster-membership* for the vertices in V. Conditionally to the cluster-memberships for the vertices in $\mathcal{G}^+$ and $\mathcal{G}^-$, the two graphs can be "merged" into a single interaction graph we define as follows.

Definition 2 (Clustering-conditional interaction graph). *Let $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$ be two interaction graphs defined over the same vertex- and edge-sets, and $C : V \longrightarrow \mathbb{N}$ be the cluster-membership function for the vertices. A* clustering-conditional interaction graph *is defined as $\mathcal{G}_C = (V, E, P_C)$, such that each edge (u, v) of $\mathcal{G}_C$ is assigned the corresponding probability distribution $p_{uv} \in P^+$ from $\mathcal{G}^+$, if u and v belongs to the same cluster according to C, otherwise the distribution $p_{uv} \in P^-$ from $\mathcal{G}^-$, i.e., $P_C = \{p_{uv} \in P^+ \mid C(u) = C(v)\} \cup \{p_{uv} \in P^- \mid C(u) \neq C(v)\}$.*

Upon the above definition, we focus on two optimization problems with complementary yet conceptually equivalent goals, that is, to cluster V so as to either (*i*) *maximize the expected overall interaction*, or (*ii*) *minimize the expected overall interaction loss.* These goals lead to two different formulations, which we state in detail next.

3.1 Maximizing interaction

Let the overall interaction $f(G)$ of a deterministic graph $G = (V, E, w_G)$ be the sum of the interactions on all edges, i.e.,

$$f(G) = \sum_{(u,v) \in E} w_G(u, v) \tag{2}$$

As a consequence, the expected overall interaction $\bar{f}(\mathcal{G})$ of an interaction graph $\mathcal{G} = (V, E, P)$ is defined as:

$$\bar{f}(\mathcal{G}) = \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}}[f(G)] = \sum_{G \sqsubseteq \mathcal{G}} f(G)\Pr(G), \tag{3}$$

where $\Pr(G)$ is the probability of observing G (Equation (1)).

The first problem we tackle in this work is as follows:

Problem 1 (Max-Interaction-Clustering). *Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$ sharing the same vertex set and edge set, find a clustering $C^* : V \longrightarrow \mathbb{N}$ that maximizes the expected overall interaction of the clustering-conditional interaction graph, i.e.,*

$$C^* = \arg\max_C \bar{f}(\mathcal{G}_C). \tag{4}$$

Connection with Correlation Clustering. Since its introduction, correlation clustering has received a great deal of attention, with a focus on various aspects, such as theoretical results, algorithms, and problem generalizations/variants [18]. To date, the most general formulation of correlation clustering [1] takes as input a set Ω of objects, and two nonnegative weights $\omega^+_{xy}, \omega^-_{xy}$ for every unordered pair $x, y \in \Omega$ of objects. The weights assigned to an object pair (x, y) intuitively express the advantage of putting x and y in the same cluster (ω^+_{xy}) or in separate clusters (ω^-_{xy}). The objective is to partition Ω so as to either *minimize* the sum of the negative weights between objects within the same cluster plus the sum of the positive weights between objects in separate clusters (Min-CC), or *maximize* the sum of the positive weights between objects within the same cluster plus the sum of the negative weights between objects in separate clusters (Max-CC):

Problem 2 (Min-CC [1]). *Given a set Ω of objects, and nonnegative weights $\omega^+_{xy}, \omega^-_{xy} \in \mathbb{R}^+_0$ for all unordered pairs $x, y \in \Omega$ of objects, find a clustering $C : \Omega \longrightarrow \mathbb{N}^+$ that minimizes*

$$\sum_{\substack{x,y \in \Omega,\\ C(x)=C(y)}} \omega^-_{xy} + \sum_{\substack{x,y \in \Omega,\\ C(x)\neq C(y)}} \omega^+_{xy}. \tag{5}$$

Problem 3 (Max-CC [1]). *Given a set Ω of objects, and nonnegative weights $\omega^+_{xy}, \omega^-_{xy} \in \mathbb{R}^+_0$ for all unordered pairs $x, y \in \Omega$ of objects, find a clustering $C : \Omega \longrightarrow \mathbb{N}^+$ that maximizes*

$$\sum_{\substack{x,y \in \Omega,\\ C(x)=C(y)}} \omega^+_{xy} + \sum_{\substack{x,y \in \Omega,\\ C(x)\neq C(y)}} \omega^-_{xy}. \tag{6}$$

Min-CC and Max-CC are equivalent in terms of optimality and complexity class (both **NP**-hard), but have different approximation-guarantee properties, with the latter being easier in this regard.

As a noteworthy result, our Max-Interaction-Clustering problem can be shown to be an instance of Max-CC:

Theorem 1. *Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$, solving Max-Interaction-Clustering on input $\left\langle \mathcal{G}^+, \mathcal{G}^- \right\rangle$ is equivalent to solving Max-CC by setting $\Omega = V$, $\omega^+_{uv} = \mathbb{E}[p^+_{uv}]$, $\omega^-_{uv} = \mathbb{E}[p^-_{uv}]$, for all $(u, v) \in E$, and $\omega^+_{uv} = \omega^-_{uv} = 0$, for all $(u, v) \in \overline{E}$ (where $\overline{E} = V \times V \setminus E$).*

Proof. The objective function of Max-Interaction-Clustering (Equation (4)) can be rearranged as follows:

$$\bar{f}(\mathcal{G}_C) = \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}_C}[f(G)] = \sum_{G \sqsubseteq \mathcal{G}_C} f(G)\Pr(G) = \sum_{G \sqsubseteq \mathcal{G}_C} \left(\sum_{(u,v) \in E} w_G(u, v) \right) \Pr(G) =$$

$$= \sum_{G \sqsubseteq \mathcal{G}_C} \left(\sum_{\substack{(u,v) \in E,\\ C(u)=C(v)}} w_G(u, v) \right) \Pr(G) + \sum_{G \sqsubseteq \mathcal{G}_C} \left(\sum_{\substack{(u,v) \in E,\\ C(u)\neq C(v)}} w_G(u, v) \right) \Pr(G) =$$

$$= \sum_{\substack{(u,v) \in E,\\ C(u)=C(v)}} \underbrace{\left(\sum_{G \sqsubseteq \mathcal{G}_C} w_G(u, v)\Pr(G) \right)}_{= \mathbb{E}[p^+_{uv}]} + \sum_{\substack{(u,v) \in E,\\ C(u)\neq C(v)}} \underbrace{\left(\sum_{G \sqsubseteq \mathcal{G}_C} w_G(u, v)\Pr(G) \right)}_{= \mathbb{E}[p^-_{uv}]} =$$

$$= \sum_{\substack{(u,v) \in E,\\ C(u)=C(v)}} \mathbb{E}[p^+_{uv}] + \sum_{\substack{(u,v) \in E,\\ C(u)\neq C(v)}} \mathbb{E}[p^-_{uv}] + \underbrace{\sum_{\substack{(u,v) \in \overline{E},\\ C(u)=C(v)}} \mathbb{E}[p^+_{uv}] + \sum_{\substack{(u,v) \in \overline{E},\\ C(u)\neq C(v)}} \mathbb{E}[p^-_{uv}]}_{= 0} =$$

$$= \sum_{\substack{x,y \in \Omega,\\ C(x)=C(y)}} \omega^+_{xy} + \sum_{\substack{x,y \in \Omega,\\ C(x)\neq C(y)}} \omega^-_{xy},$$

which corresponds to the objective function of Max-CC. □

The connection with correlation clustering also unveils the **NP**-hardness of Max-Interaction-Clustering:

Theorem 2. *Max-Interaction-Clustering is* **NP***-hard.*

Proof. (Sketch) The fact of being a special case of Max-CC clearly does not necessarily imply that Max-Interaction-Clustering is **NP**-hard too. However, **NP**-hardness can be shown by reducing from the basic Bansal *et al.*'s variant of correlation clustering on general graphs [2], which corresponds to Max-CC when $(\omega^+_{xy}, \omega^-_{xy}) \in$

$\{(1,0),(0,0),(0,1)\}$. Even such a simpler variant is **NP**-hard, and can easily be observed to correspond to Max-Interaction-Clustering when $\forall (u,v) \in E\colon p^{+}_{uv}(\omega^{+}_{uv}) = p^{-}_{uv}(\omega^{-}_{uv}) = 1$. □

3.2 Minimizing interaction loss

Given two interaction graphs $\mathcal{G}^{+} = (V, E, P^{+})$, $\mathcal{G}^{-} = (V, E, P^{-})$, let $M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) \in \mathbb{R}^{+}$ be a constant larger than the maximum interaction strength in $\mathcal{G}^{+}$ and $\mathcal{G}^{-}$, i.e., $M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) > \max\{x \in \mathcal{D}(p_{uv}) \mid p_{uv} \in P^{+} \cup P^{-}, (u,v) \in E\}$. Based on $M(\mathcal{G}^{+}\!,\mathcal{G}^{-})$, let the overall interaction loss $\ell(G)$ of a deterministic graph $G = (V, E, w_G)$ be:

$$\begin{aligned}\ell(G) = & \sum_{(u,v)\in E} (M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - w_G(u,v)) \; + \; |\overline{E}| M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) = \\ = & \;\; M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) \binom{|V|}{2} - \sum_{(u,v)\in E} w_G(u,v),\end{aligned} \tag{7}$$

and the expected overall interaction loss $\bar{\ell}(\mathcal{G})$ of an interaction graph $\mathcal{G} = (V, E, P)$ be:

$$\bar{\ell}(\mathcal{G}) \; = \; \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}}[\ell(G)] \; = \; \sum_{G \sqsubseteq \mathcal{G}} \ell(G) \Pr(G). \tag{8}$$

The minimization version of the problem we tackle in this work is:

Problem 4 (Min-Interaction-loss-Clustering). *Given two interaction graphs $\mathcal{G}^{+} = (V, E, P^{+})$ and $\mathcal{G}^{-} = (V, E, P^{-})$ sharing the same vertex set and edge set, find a clustering $C^{*} : V \longrightarrow \mathbb{N}^{+}$ so that*

$$C^{*} = \arg\min_{C} \bar{\ell}(\mathcal{G}_C). \tag{9}$$

Such a minimization formulation is equivalent to the maximization one in terms of optimality (and complexity class), since:

$$\begin{aligned}\bar{\ell}(\mathcal{G}) = & \sum_{G \sqsubseteq \mathcal{G}} \left(M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) \binom{|V|}{2} - \sum_{(u,v)\in E} w_G(u,v) \right) \Pr(G) = \\ = & \; \underbrace{- \sum_{G \sqsubseteq \mathcal{G}} \left(\sum_{(u,v)\in E} w_G(u,v) \right) \Pr(G)}_{= -\bar{f}(\mathcal{G})} + \underbrace{M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) \binom{|V|}{2}}_{\text{constant } > \, 0}.\end{aligned} \tag{10}$$

The result in Theorem 3 immediately follows:

Theorem 3. *Min-Interaction-loss-Clustering is* **NP**-*hard.*

Connection with Correlation Clustering. Similarly to the maximization version, our Min-Interaction-loss-Clustering can be shown to be an instance of Min-CC:

Theorem 4. *Given two interaction graphs $\mathcal{G}^{+} = (V, E, P^{+})$ and $\mathcal{G}^{-} = (V, E, P^{-})$, solving Min-Interaction-loss-Clustering on input $\langle \mathcal{G}^{+}, \mathcal{G}^{-} \rangle$ is equivalent to solving Min-CC by setting $\Omega = V$, $\omega^{+}_{uv} = M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - \mathbb{E}[p^{-}_{uv}]$, $\omega^{-}_{uv} = M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - \mathbb{E}[p^{+}_{uv}]$, for all $(u,v) \in E$, and $\omega^{+}_{uv} = \omega^{-}_{uv} = M(\mathcal{G}^{+}\!,\mathcal{G}^{-})$, for all $(u,v) \in \overline{E}$ (where $\overline{E} = \binom{V}{2} \setminus E$).*

Proof. Let us define the *discounted interaction loss $\mathcal{G}_C$* which discards the loss contribution due to non-linked pairs, as follows:

$$\begin{aligned}\mathcal{L}(\mathcal{G}_C) = & \sum_{\substack{(u,v)\in E,\\ C(u)=C(v)}} \underbrace{\left(\sum_{G \sqsubseteq \mathcal{G}_C} (M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - w_G(u,v)) \right) \Pr(G)}_{= M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - \mathbb{E}[p^{+}_{uv}]} + \\ & + \sum_{\substack{(u,v)\in E,\\ C(u)\neq C(v)}} \underbrace{\left(\sum_{G \sqsubseteq \mathcal{G}_C} (M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - w_G(u,v)) \right) \Pr(G)}_{= M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) - \mathbb{E}[p^{-}_{uv}]}.\end{aligned} \tag{11}$$

With similar arguments as Theorem 1, it can be shown that:

$$\bar{\ell}(\mathcal{G}_C) = \mathcal{L}(\mathcal{G}_C) + \; |\overline{E}| M(\mathcal{G}^{+}\!,\mathcal{G}^{-}) = \sum_{\substack{x,y\in\Omega,\\ C(x)=C(y)}} \omega^{-}_{xy} \; + \sum_{\substack{x,y\in\Omega,\\ C(x)\neq C(y)}} \omega^{+}_{xy},$$

which corresponds to the objective function of Min-CC. □

4 ALGORITHMS

The connection with correlation clustering forms the basis of algorithm design for our problems. Specifically, our main idea here is to investigate whether the considerable amount of work on correlation-clustering algorithms with proved quality guarantees can be fruitfully exploited in our setting too.

Our first remark in this regard is that, thanks to Theorem 1, it is not hard to demonstrate that (constant-factor) approximation algorithms designed for Max-CC keep their guarantees on Max-Interaction-Clustering too (see *Appendix A*). However, the state-of-the-art approximation algorithms for Max-CC (on general, weighted graphs) correspond to the semidefinite-programming-based ones devised by Swamy [20]. Those algorithms are inefficient and, more importantly, rather impractical, since they are not able to output more than a fixed number of clusters (i.e., six). This is a showstopper in our context, as we are interested in algorithms that are effective and theoretically solid, yet capable of handling large-scale inputs and providing outputs whose quality is recognizable in practice too, not only theoretically.

More interesting results instead hold for the minimization version of our problem. Specifically, we derive a clever rearrangement of Min-Interaction-loss-Clustering's objective function, which unveils that, under mild conditions, the algorithms designed for Min-CC preserve their approximation properties when applied (with minor modifications) to our problem. This is particularly appealing, as Min-CC admits approximation algorithms that do not suffer from the limitations of the maximization counterpart, i.e., they are efficient and capable of finding general clusterings [1]. For this reason, our algorithm-design process focuses on the minimization version of our problem, and the remainder of this section provides the details of this process.

4.1 An approximation algorithm

Ailon *et al.*'s KwikCluster [1] is a well-established algorithm for Min-CC. It iteratively picks an object x (uniformly at random among the unclustered objects), and builds a cluster comprised of x and all unclustered objects y such that $\omega^{+}_{xy} > \omega^{-}_{xy}$. KwikCluster is particularly appealing, due to its (*i*) constant-factor (expected) approximation guarantees (i.e., factor-5 or factor-2, depending on the conditions satisfied by the input weights, cf. later in this section), (*ii*) efficiency (i.e., it takes linear time in the number of edges of the input graph), and (*iii*) easiness of implementation. All these aspects make it generally preferable to other algorithms (such as the one by Charikar *et al.* [5]) that have slightly better approximation guarantees, but are less efficient and more difficult to implement.

Theoretical basis. With the above motivations, we investigate possible exploitation of KwikCluster for our Min-Interaction-loss-Clustering, and the theoretical basis for which its appealing features are still valid. In this regard, a major remark is that the constant-factor approximation guarantees of KwikCluster hold for input graphs whose weights on every edge satisfy the probability constraint, i.e., $\omega^+_{xy} + \omega^-_{xy} = 1$, for all $x, y \in \Omega$. Although this is a requirement that does not generally hold for the input to Min-Interaction-loss-Clustering, in the following we show that the objective function of our problem can be manipulated in such a way that the probability constraint *is satisfied under mild conditions.*

Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$, $\mathcal{G}^- = (V, E, P^-)$, for every unordered pair $u, v \in V$, let σ_{uv}, τ^+_{uv}, τ^-_{uv} be:

$$\sigma_{uv} = M(\mathcal{G}^+\!,\mathcal{G}^-) - \left(\mathbb{E}[p^+_{uv}] + \mathbb{E}[p^-_{uv}]\right), \tag{12}$$

$$\tau^+_{uv} = \frac{1}{M(\mathcal{G}^+\!,\mathcal{G}^-)}\left(\mathbb{E}[p^+_{uv}] + \frac{\sigma_{uv}}{2}\right), \quad \tau^-_{uv} = \frac{1}{M(\mathcal{G}^+\!,\mathcal{G}^-)}\left(\mathbb{E}[p^-_{uv}] + \frac{\sigma_{uv}}{2}\right). \tag{13}$$

It is easy to see that $\sigma_{uv} \in [-M(\mathcal{G}^+\!,\mathcal{G}^-), M(\mathcal{G}^+\!,\mathcal{G}^-)]$, and τ^+_{uv}, τ^-_{uv} satisfy the above probability constraint, as stated in Lemma 1.

Lemma 1. *It holds that $\tau^+_{uv}, \tau^-_{uv} \geq 0$ and $\tau^+_{uv} + \tau^-_{uv} = 1$, $\forall u, v \in V$.*

Function $\bar{\ell}(\cdot)$ of Min-Interaction-loss-Clustering can be rewritten in terms of τ^+_{uv}, τ^-_{uv}, as follows (proof in *Appendix B*).

Lemma 2. *Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$, and a clustering $C : V \longrightarrow \mathbb{N}^+$, let*

$$g(\mathcal{G}_C) = \sum_{\substack{u,v \in V,\\ C(u)=C(v)}} \tau^-_{uv} + \sum_{\substack{u,v \in V,\\ C(u)\neq C(v)}} \tau^+_{uv}, \quad K(\mathcal{G}^+\!,\mathcal{G}^-) = \sum_{u,v \in V} \frac{\sigma_{uv}}{2}. \tag{14}$$

It holds that $\bar{\ell}(\mathcal{G}_C) = M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_C) + K(\mathcal{G}^+\!,\mathcal{G}^-)$.

Moreover, in Lemma 3, we state that constant-factor approximation guarantees for Min-CC carry over to our problem.

Lemma 3. *If $K(\mathcal{G}^+\!,\mathcal{G}^-) \geq 0$ (Equation (14)), then any α-approximation algorithm for Min-CC is an α-approximation algorithm for Min-Interaction-loss-Clustering, for every constant $\alpha > 1$.*

Proof. Let $I_1 = \langle \mathcal{G}^+, \mathcal{G}^- \rangle$ be an instance of Min-Interaction-loss-Clustering, and $I_2 = \langle V, \{\tau^+_{uv}\}_{u,v \in V}, \{\tau^-_{uv}\}_{u,v \in V} \rangle$ be an instance of Min-CC derived from I_1 by employing the weights defined in Equation (13). Let also $C^*_{\bar{\ell}}$ and C^*_g be the optimal clusterings for the I_1 instance according to the $\bar{\ell}(\cdot)$ and $g(\cdot)$ functions, respectively. Finally, let $\widetilde{C}$ denote the clustering yielded by the given α-approximation algorithm for Min-CC on input I_2.

The goal is to demonstrate that, for every I_1, I_2, $\bar{\ell}(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times \bar{\ell}(\mathcal{G}_{C^*_{\bar{\ell}}})$. First, it is straightforward to note that $g(\cdot)$ corresponds to Min-CC's objective function. By definition of approximation algorithm, $g(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times g(\mathcal{G}_{C^*_g})$, therefore it holds that:

$$\begin{aligned}
& g(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times g(\mathcal{G}_{C^*_g}) \\
\Leftrightarrow\; & M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{C^*_g}) \\
\Leftrightarrow\; & M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{\widetilde{C}}) + K(\mathcal{G}^+\!,\mathcal{G}^-) \leq \alpha \times M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{C^*_g}) + K(\mathcal{G}^+\!,\mathcal{G}^-) \\
\Rightarrow\; & M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{\widetilde{C}}) + K(\mathcal{G}^+\!,\mathcal{G}^-) \leq \alpha \times M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{C^*_g}) + \alpha \times K(\mathcal{G}^+\!,\mathcal{G}^-) \\
\Leftrightarrow\; & \underbrace{M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{\widetilde{C}}) + K(\mathcal{G}^+\!,\mathcal{G}^-)}_{= \bar{\ell}(\mathcal{G}_{\widetilde{C}})\ \{\text{Lemma 2}\}} \leq \alpha \times \underbrace{\left(M(\mathcal{G}^+\!,\mathcal{G}^-) \times g(\mathcal{G}_{C^*_g}) + K(\mathcal{G}^+\!,\mathcal{G}^-)\right)}_{= \bar{\ell}(\mathcal{G}_{C^*_g})\ \{\text{Lemma 2}\}} \\
\Leftrightarrow\; & \bar{\ell}(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times \bar{\ell}(\mathcal{G}_{C^*_g}) \;\Leftrightarrow\; \bar{\ell}(\mathcal{G}_{\widetilde{C}}) \leq \alpha \times \bar{\ell}(\mathcal{G}_{C^*_{\bar{\ell}}}),
\end{aligned}$$

where the first equivalence step holds since $M(\mathcal{G}^+\!,\mathcal{G}^-) > 0$, the second one holds because of the assumption $K(\mathcal{G}^+\!,\mathcal{G}^-) \geq 0$, the third step holds as $\alpha > 1$, and the last step holds since, based on Lemma 2, the optimum of $g(\cdot)$ corresponds to the optimum of $\bar{\ell}(\cdot)$. □

Algorithm 1 MIL

```
Input: Interaction graphs $\mathcal{G}^+ = (V, E, P^+)$, $\mathcal{G}^- = (V, E, P^-)$
Output: A clustering $C$ of $V$
1: compute $\tau^+_{uv}$, $\tau^-_{uv}$ for all $(u, v) \in E$        {Equation (13)}
2: $C \leftarrow \emptyset$, $V' \leftarrow V$
3: while $V' \neq \emptyset$ do
4:     pick a pivot vertex $u \in V'$ uniformly at random
5:     $C_u \leftarrow \{u\} \cup \{v \in V' \mid (u, v) \in E, \tau^+_{uv} > \tau^-_{uv}\}$
6:     add cluster $C_u$ to $C$ and remove all vertices in $C_u$ from $V'$
```

The MIL algorithm. Lemmas 1–3 provide the theoretical support and motivation for the first algorithm we propose for our Min-Interaction-loss-Clustering problem, named MIL, whose pseudocode is shown in Algorithm 1. Given two interaction graphs $\mathcal{G}^+$, $\mathcal{G}^-$, MIL simply builds an instance of Min-CC as per Equation (13), and applies the KwikCluster algorithm on it.

Proposition 1. (cf. *Appendix D*) *MIL takes $O(|V|+|E|)$ time.*

Approximation guarantees. Thanks to Lemmas 1–3, the MIL algorithm can be shown to achieve expected factor-5 approximation guarantees for Min-Interaction-loss-Clustering if $K(\mathcal{G}^+\!,\mathcal{G}^-) \geq 0$:

Theorem 5. *If $K(\mathcal{G}^+\!,\mathcal{G}^-) \geq 0$, Algorithm 1 is a randomized expected 5-approximation algorithm for Problem 4.*

Proof. The weights τ^+_{uv}, τ^-_{uv} satisfy the probability constraint (Lemma 1). Thus, running the KwikCluster algorithm (i.e., Lines 3–6 of Algorithm 1) on a Min-CC instance with τ^+_{uv}, τ^-_{uv} weights is proved to achieve expected 5-approximation guarantees for Min-CC [1]. According to Lemma 3, If $K(\mathcal{G}^+\!,\mathcal{G}^-) \geq 0$, such a factor-5 approximation carries over to Problem 4. □

Condition for approximation guarantees. The condition for MIL to be a 5-approximation algorithm for Min-Interaction-loss-Clustering is that the constant $K(\mathcal{G}^+\!,\mathcal{G}^-)$ (Equation (14)) is nonnegative. Here we show that this is a rather mild assumption, which is expected to hold for real-world interaction graphs. In fact:

$$\begin{aligned}
K(\mathcal{G}^+\!,\mathcal{G}^-) \; & = \sum_{u,v \in V} \frac{\sigma_{uv}}{2} \; = \sum_{(u,v) \in E} \frac{\sigma_{uv}}{2} + \sum_{(u,v) \notin E} \frac{\sigma_{uv}}{2} \\
& = \sum_{(u,v) \in E} \left(\frac{M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}] - \mathbb{E}[p^-_{uv}]}{2}\right) + \frac{M(\mathcal{G}^+\!,\mathcal{G}^-)}{2}\left(\binom{|V|}{2} - |E|\right) \geq 0 \\
& \Leftrightarrow \sum_{(u,v) \in E} \left(\frac{\mathbb{E}[p^+_{uv}] + \mathbb{E}[p^-_{uv}]}{2}\right) \leq \frac{M(\mathcal{G}^+\!,\mathcal{G}^-)}{2}|E| + \frac{M(\mathcal{G}^+\!,\mathcal{G}^-)}{2}\left(\binom{|V|}{2} - |E|\right) \\
& \Leftrightarrow \sum_{(u,v) \in E} \left(\mathbb{E}[p^+_{uv}] + \mathbb{E}[p^-_{uv}]\right) \leq M(\mathcal{G}^+\!,\mathcal{G}^-)\binom{|V|}{2}.
\end{aligned} \tag{15}$$

Thus, as $\mathbb{E}[p^+_{uv}], \mathbb{E}[p^-_{uv}] \leq M(\mathcal{G}^+\!,\mathcal{G}^-)$, the worst case to have the condition in the above Equation (15) satisfied is when $\mathbb{E}[p^+_{uv}] = \mathbb{E}[p^-_{uv}] = M(\mathcal{G}^+\!,\mathcal{G}^-)$, for all $(u, v) \in E$. This means that, in the worst case, $K(\mathcal{G}^+\!,\mathcal{G}^-)$ is guaranteed to be nonnegative if $|E| \leq \binom{|V|}{2}/2$, i.e., if the number of edges in the input interaction graphs is no more than half of the number of all unordered pairs of vertices. Note this relates to sparseness, which is typical in real-world graphs.

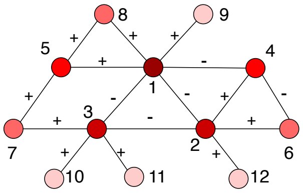


Figure 1: MIL algorithm: Effect of sampling pivots uniformly at random in general graphs.

Stronger approximation guarantees. If the input weights, apart from satisfying the probability constraint, also obey the triangle inequality, then the KwikCluster algorithm for Min-CC is shown to achieve better approximation guarantees, i.e., 2 instead of 5 [1]. In our setting this means that, if the weights defined in Equation (13) are such that $\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv}$, for all $u, v, z \in V$, then the proposed MIL algorithm becomes a 2-approximation algorithm for Min-Interaction-loss-Clustering.

To this purpose, let us denote with Δ^+_{uv} the difference $\mathbb{E}[p^+_{uv}] - \mathbb{E}[p^-_{uv}]$, for any $u, v \in V$; also, let $\Delta^-_{uv} := -\Delta^+_{uv}$. It can first be noted that, for any $u, v, z \in V$, whenever the triangle inequality $\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv}$ holds, then there exists an equivalent inequality in terms of the expectation differences, up to the constant $M(\mathcal{G}^+;\mathcal{G}^-)$.

Lemma 4. *Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$, it holds that $\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv} \Leftrightarrow \Delta^-_{uv} \leq \Delta^-_{uz} + \Delta^-_{zv} + M(\mathcal{G}^+;\mathcal{G}^-)$, with $u, v, z \in V$.*

Proof. By definition (Equation (13)), $\tau^-_{uv} = \frac{1}{M(\mathcal{G}^+;\mathcal{G}^-)}(\mathbb{E}[p^-_{uv}] + \frac{\sigma_{uv}}{2})$, where $\sigma_{uv} = M(\mathcal{G}^+;\mathcal{G}^-) - (\mathbb{E}[p^+_{uv}] + \mathbb{E}[p^-_{uv}])$, thus it holds:

$$\begin{aligned}
\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv} &\Leftrightarrow \mathbb{E}[p^-_{uv}] + \frac{\sigma_{uv}}{2} \leq \mathbb{E}[p^-_{uz}] + \frac{\sigma_{uz}}{2} + \mathbb{E}[p^-_{zv}] + \frac{\sigma_{zv}}{2} \\
&\Leftrightarrow \mathbb{E}[p^-_{uv}] + M(\mathcal{G}^+;\mathcal{G}^-) - \mathbb{E}[p^+_{uv}] \leq \mathbb{E}[p^-_{uz}] + M(\mathcal{G}^+;\mathcal{G}^-) - \mathbb{E}[p^+_{uz}] + \\
&\qquad + \mathbb{E}[p^-_{zv}] + M(\mathcal{G}^+;\mathcal{G}^-) - \mathbb{E}[p^+_{zv}] \\
&\Leftrightarrow \Delta^-_{uv} \leq \Delta^-_{uz} + \Delta^-_{zv} + M(\mathcal{G}^+;\mathcal{G}^-), \text{ with } u, v, z \in V. \qquad \square
\end{aligned}$$

The following Lemma 5 states the condition for stronger approximation guarantees, which requires that the difference $\mathbb{E}[p^+_{uv}] - \mathbb{E}[p^-_{uv}]$ lies in the range $[0, M(\mathcal{G}^+;\mathcal{G}^-)/2]$.

Lemma 5. *Given interaction graphs $\mathcal{G}^+ = (V, E, P^+)$, $\mathcal{G}^- = (V, E, P^-)$, if $\Delta^+_{uv} \in [0, M(\mathcal{G}^+;\mathcal{G}^-)/2]$, then $\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv}$, for any $u, v, z \in V$.*

Proof. By Lemma 4, it follows that:

$$\begin{aligned}
\tau^-_{uv} \leq \tau^-_{uz} + \tau^-_{zv} &\Leftrightarrow \Delta^-_{uv} + \Delta^+_{uz} + \Delta^+_{zv} \leq M(\mathcal{G}^+;\mathcal{G}^-) \\
&\Leftrightarrow \Delta^-_{uv} + \Delta^+_{uz} + \Delta^+_{zv} \leq 0 + \frac{M(\mathcal{G}^+;\mathcal{G}^-)}{2} + \frac{M(\mathcal{G}^+;\mathcal{G}^-)}{2} \\
&\Leftarrow \Delta^-_{uv} \leq 0, \;\; \Delta^+_{uv} \leq \frac{M(\mathcal{G}^+;\mathcal{G}^-)}{2}, \forall u, v \in V,
\end{aligned}$$

which corresponds to $\Delta^+_{uv} \in [0, M(\mathcal{G}^+;\mathcal{G}^-)/2]$, as $\Delta^+_{uv} = -\Delta^-_{uv}$. □

Theorem 6. *If $K(\mathcal{G}^+;\mathcal{G}^-) \geq 0$ and $\Delta^+_{uv} \in [0, M(\mathcal{G}^+;\mathcal{G}^-)/2]$, $\forall u, v \in V$, Algorithm 1 is a randomized expected 2-approximation algorithm for Problem 4.*

Thus, the stronger approximation guarantees of MIL hold if the expected interaction between any two users u and v when they are put in the same cluster is higher than the expected interaction when they are part of different clusters, and the former does not exceed the latter by more than half of the maximum interaction. This is actually not a strict condition, since it can be observed in application scenarios (especially $\Delta^+_{uv} \geq 0$).

Algorithm 2 D-MIL

```
Input: Interaction graphs $\mathcal{G}^+ = (V, E, P^+)$, $\mathcal{G}^- = (V, E, P^-)$
Output: A clustering $\mathcal{C}$ of $V$
compute $\tau^+_{uv}$, $\tau^-_{uv}$ for all $(u, v) \in E$        {Equation (13)}
$\mathcal{C} \leftarrow \emptyset$, $V' \leftarrow V$
while $V' \neq \emptyset$ do
    compute $d_{V'}(u) = |\{v \in V' \mid (u, v) \in E\}|$, for all $u \in V'$
    sample a pivot vertex $u \in V'$ with probability proportional to $d_{V'}(u)$
    $C_u \leftarrow \{u\} \cup \{v \in V' \mid (u, v) \in E, \tau^+_{uv} > \tau^-_{uv}\}$
    add cluster $C_u$ to $\mathcal{C}$ and remove all vertices in $C_u$ from $V'$
```

Relation with correlation-clustering theory. Correlation-clustering (in)approximability result states that Min-CC on general (i.e., not necessarily complete) graphs is **APX**-hard, with best known approximation factor $O(\log |V|)$ [5]. By constrast, in Theorems 5–6, we have shown that our Min-Interaction-loss-Clustering problem has constant-factor approximation guarantees for general instances of our problem, and such results are obtained by adapting correlation-clustering algorithms.

The above would apparently contradict the theory on correlation clustering. However, this is not the case, for the following reasons. First, although the original input interaction graphs we deal with are general (i.e., they may have missing edges), the way how we formulate our Min-Interaction-loss-Clustering problem, through the $M(\mathcal{G}^+;\mathcal{G}^-)$ constant (Equation (7)), guarantees that the actual graphs processed by the Min-Interaction-loss-Clustering algorithms are complete, i.e., they have an edge with a positive weight between every pair of vertices. Second, the actual edge weights handled by the Min-Interaction-loss-Clustering algorithms (Equation (13)) are not arbitrary. Indeed, they are derived from the original weights with an ad-hoc rearrangement that guarantees the appealing properties we show above (i.e., fulfilment of the probability constraint and the fact that constant-factor guarantees for correlation clustering carry over to our problem). Such a rearrangement is a nice peculiarity of our problem, which is not possible in general: that is the main reason why this result is not in contrast with the inapproximability of Min-CC on general graphs.

4.2 Enhanced pivot-sampling strategy

The proposed MIL basically resembles the correlation-clustering KwikCluster algorithm [1] on a graph with ad-hoc-defined edge weights. However, KwikCluster is explicitly designed for complete graphs, whereas the input graphs for our Min-Interaction-loss-Clustering problem are general at first. Clearly, KwikCluster could be modified to handle general graphs, but this may lead to ineffectiveness. More specifically, we recall that the edge reweighting adopted in our MIL algorithm (Equation (13)) makes the input graph complete by assigning an equal positive and negative weight (equal to 1/2) to those vertex pairs that do not share an edge in the original graph. This way, putting those non-linked pairs in the same cluster or in different clusters does not make any difference in terms of objective-function value of the resulting solution. This fact is overlooked by KwikCluster, which, being designed for complete graphs, samples pivots uniformly at random, without taking into account how many neighbors a candidate pivot has in the original graph. This may raise inaccuracies, as shown in the next example.

Example 1. *Figure 1 shows an interaction graph where* (u, v) *edges are labeled according to what distribution prevails on the other in terms of expected value, i.e., "+" if* $\mathbb{E}[p^+_{uv}] > \mathbb{E}[p^-_{uv}]$*, "−" otherwise, while vertices are colored according to their degrees, i.e., the darker the shade, the more the number of edges adjacent to it. The optimal solution of* Min-Interaction-loss-Clustering *on this example consists of clusters* $\{3, 7, 10, 11\}$, $\{2, 4, 6, 12\}$, $\{1, 5, 8, 9\}$. *It is apparent that this optimal clustering may be found by the MIL algorithm if darker-colored vertices (i.e., vertices 1, 2, 3) are selected as pivots. Nevertheless, as MIL samples pivots uniformly at random, it is likely that one of the lighter-colored vertices (that are more than the darker-colored ones) becomes instead a pivot in the first place. This may lead to ineffective clusterings. For instance, assume vertex 10 is selected as a very first pivot. Even assuming that the next pivots are the darker-colored vertices 1 and 2, the ultimate clustering will be* $\{3, 10\}$, $\{1, 5, 8, 9\}$, $\{2, 4, 6, 12\}$, $\{11\}$, $\{7\}$, *which is far from the optimal one.*

The above example shows that, on general graphs, sampling pivots according to their degrees may be more appropriate than uniform sampling. This is the main intuition behind the second algorithm we propose in this work, which is termed D-MIL and whose outline is reported as Algorithm 2.

Proposition 2. (cf. *Appx. D*) *D-MIL takes* $O(|E| \log |V|)$ *time.*

4.3 Hill climbing

The proposed MIL and D-MIL algorithms can be further improved by performing an a-posteriori hill-climbing step on the clusterings yielded by them. In particular, the idea is to consider relocating vertices in other clusters, as long as the resulting clustering has a better objective-function value. Such relocation steps may be efficiently implemented by incrementally computing marginal objective-function losses/gains. Specifically, given a clustering C, let C' be the clustering obtained from C by moving a vertex u from cluster $C_u \in C$ to a cluster $C'_u \neq C_u$. Taking into account the rearrangement of the $\bar{\ell}(\cdot)$ objective function stated in Theorem 4, removing u from C_u leads to a *decrease* in the $\bar{\ell}(\cdot)$ function equal to:

$$\sum_{v \in C_u \setminus \{u\}} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}]\right) \;+ \sum_{v \in V \setminus C_u} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\right).$$

At the same time, adding u to C'_u leads to an *increase* of $\bar{\ell}(\cdot)$ equal to:

$$\sum_{v \in C'_u} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}]\right) \;+ \sum_{v \in V \setminus C'_u} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\right).$$

Combining the expressions above, and denoting $\Delta^+_{uv} = \mathbb{E}[p^+_{uv}] - \mathbb{E}[p^-_{uv}]$, $\Delta^-_{uv} = -\Delta^+_{uv}$, we obtain:

$$\begin{aligned}
\bar{\ell}(\mathcal{G}_{C'}) \;=\;& \bar{\ell}(\mathcal{G}_C) + \sum_{v \in C_u \setminus \{u\}} \underbrace{\Big(\big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\big) - \big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}]\big)\Big)}_{= \mathbb{E}[p^+_{uv}] - \mathbb{E}[p^-_{uv}] = \Delta^+_{uv}} + \\
& + \sum_{v \in C'_u} \underbrace{\Big(\big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}]\big) - \big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\big)\Big)}_{= \mathbb{E}[p^-_{uv}] - \mathbb{E}[p^+_{uv}] = \Delta^-_{uv}} + \\
& + \sum_{v \in V \setminus \{C_u \,\cup\, C'_u\}} \underbrace{\Big(\big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\big) - \big(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\big)\Big)}_{= 0} \\
=\;& \bar{\ell}(\mathcal{G}_C) + \sum_{v \in C_u \setminus \{u\}} \Delta^+_{uv} + \sum_{v \in C'_u} \Delta^-_{uv} \;=\; \bar{\ell}(\mathcal{G}_C) + \sum_{\substack{v \in C_u \setminus \{u\},\\ (u,v) \in E}} \Delta^+_{uv} + \sum_{\substack{v \in C'_u,\\ (u,v) \in E}} \Delta^-_{uv}, \qquad (16)
\end{aligned}$$

where the last equivalence holds as, for vertices $v : (u, v) \notin E$, $\mathbb{E}[p^+_{uv}] = \mathbb{E}[p^-_{uv}]$, and, then, $\Delta^+_{uv} = \Delta^-_{uv} = 0$. The hill-climbing step consists in iteratively picking a vertex u and a cluster $C'_u \neq C_u$ that minimize Eq. (16), and moving u from C_u to C'_u. This local-search process goes on until either no movement leading to a decrease in the $\bar{\ell}(\cdot)$ function exists, or a certain number of iterations I has been hit. The process is outlined as Algorithm 3 (cf. *Appendix C*).

Proposition 3. (cf. *Appendix D*) *Hill-climbing for MIL and D-MIL takes* $O(I(|V| + |E|))$ *time, with I number of iterations.*

Final remark on approximation guarantees. Here we summarize the approximation guarantees of all the proposed algorithms. Algorithm 1 achieves constant-factor approximation guarantees under mild conditions (either factor-5 or factor-2, see Theorems 5–6). Algorithm 2 does not come instead with any guarantees as of now: the study of its approximation properties is indeed an interesting open problem that we defer to future work. However, we remark that one can still have both the guarantees of Algorithm 1 without sacrificing the practical benefits of Algorithm 2: simply run both the algorithms and take the best one (in terms of objective-function value) among the two yielded solutions. This way, the approximation guarantees of Algorithm 1 would be preserved. As far as hill climbing procedure, instead, being a post-processing strategy that can only improve the outputs of Algorithm 1 or Algorithm 2, it does not alter the approximation properties of those algorithms (i.e., it achieves guarantees if applied to Algorithm 1's solutions, while no guarantees hold for the combo Algorithm 2 + hill climbing).

We hereinafter denote with suffix *_R* the combo algorithms obtained by executing Algorithm 3 in cascade of Algorithm 1 (MIL_R) or Algorithm 2 (D-MIL_R).

5 EXPERIMENTAL EVALUATION

Data. We considered real-world networks as well as data produced by selected random network generation models. More specifically, we used 10 real-world, undirected, unweighted and timestamped networks, available from the KONECT project, except PrimarySchool and HighSchool from SocioPatterns, and StackOverflow from SNAP.[1] Please see Table 3 in *Appendix E* for details.

Each of the input temporal networks is treated as a sequence of undirected snapshot graphs $\langle G_1, \ldots, G_T \rangle$, where each $G_t = (V, E_t)$ ($t = 1..T$) models the vertex interactions at time t. We defined the interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$ by "flattening" the temporal network and estimating the p^+_{uv} and p^-_{uv} distributions, respectively, based on the fractions of clusters shared by u, v over all graphs, according to a precomputed clustering solution on each graph G_t. Insights on this can be found in *Appendix E*.

Concerning the synthetic data, we focused on two well-known random-graph models, namely Barabasi-Albert (hereinafter BA)

[1] http://konect.cc/, http://www.sociopatterns.org/datasets/, http://snap.stanford.edu/

Table 1: Average loss values and clustering sizes

	MIL		MIL_R		D-MIL		D-MIL_R		CPM [21]		GJA [9]		CPMap [7]	
	loss	#clusters	loss	#clusters	loss	#clusters	loss	#clusters	loss	#clusters	loss	#clusters	loss	#clusters
Amazon	4.80×10^6	1.51×10^6	3.82×10^6	1.36×10^6	4.49×10^6	1.47×10^6	3.69×10^6	1.34×10^6	4.38×10^6	1.17×10^6	4.33×10^6	1.03×10^6	**3.66×10^6**	1.34×10^6
DBLP	3.94×10^6	986.02×10^3	3.17×10^6	614.86×10^3	3.70×10^6	858.93×10^3	3.01×10^6	557.90×10^3	**2.55×10^6**	354.03×10^3	2.89×10^6	506.72×10^3	2.81×10^6	393.38×10^3
Epinions	12.92×10^6	76.81×10^3	4.71×10^6	47.54×10^3	9.06×10^6	65.59×10^3	**4.70×10^6**	47.51×10^3	9.80×10^6	16.73×10^3	8.82×10^6	16.68×10^3	5.06×10^6	65.31×10^3
HighSchool	4.59×10^3	45.26	3.50×10^3	8.16	4.44×10^3	37.66	3.35×10^3	6.38	4.29×10^3	9.00	3.43×10^3	7.00	**3.29×10^3**	8.00
Last.fm	164.67×10^3	57.04	**150.25×10^3**	37.64	163.35×10^3	42.10	**150.25×10^3**	36.94	161.53×10^3	3.00	160.66×10^3	4.00	151.60×10^3	37.00
PrimarySchool	6.95×10^3	16.44	5.01×10^3	1.20	6.80×10^3	15.12	**4.92×10^3**	1.04	6.48×10^3	5.00	6.27×10^3	5.00	5.46×10^3	2.00
ProsperLoans	1.82×10^6	39.60×10^3	1.30×10^6	3.75×10^3	1.81×10^6	26.06×10^3	**1.28×10^6**	3.70×10^3	**1.28×10^6**	1.54×10^3	1.30×10^6	1.13×10^3	1.39×10^6	7.49×10^3
StackOverflow	12.39×10^6	1.74×10^6	8.83×10^6	308.66×10^3	11.91×10^6	1.27×10^6	**8.65×10^6**	237.36×10^3	9.90×10^6	106.58×10^3	9.26×10^6	13.78×10^3	10.81×10^6	188.44×10^3
Wikipedia	6.74×10^6	276.14×10^3	5.31×10^6	157.77×10^3	6.44×10^6	246.29×10^3	**5.26×10^6**	168.80×10^3	5.84×10^6	113.64×10^3	5.84×10^6	108.82×10^3	5.83×10^6	209.68×10^3
WikiTalk	6.29×10^6	2.77×10^6	3.72×10^6	381.88×10^3	5.41×10^6	1.99×10^6	**3.38×10^6**	485.66×10^3	3.68×10^6	351.73×10^3	5.13×10^6	1.69×10^6	NA	NA
tot. average	4.91×10^6	7.40×10^5	3.10×10^6	2.87×10^5	4.30×10^6	5.93×10^5	3.01×10^6	2.84×10^5	3.76×10^6	2.11×10^5	3.77×10^6	3.37×10^5	3.30×10^6	2.45×10^5

and Watts-Strogatz (hereinafter WS) models. For the BA model, we varied the number of edges to attach with a new vertex, denoted as m, and for the WS model, we varied the distance (i.e., number of steps) within which two vertices will be connected, denoted as *neigh*. More details can be found in *Appendix E*.

Evaluation goals and competitors. We evaluated the *interaction loss*, the *size of the clustering* produced, and the *runtime performance* of the proposed methods, i.e., MIL, D-MIL, MIL_R, and D-MIL_R. All criteria measurements reported correspond to averages over 100 runs. We set the number of iterations I to 8, which experimentally revealed to be a good trade-off for balancing the three criteria. Moreover, we compared our proposed methods against three selected methods for community detection in signed graphs, namely CPM [21], GJA [9], and CPMap [7] (cf. Sect. 2). Since all such methods require only one weight, either positive or negative, for each edge, we set any weight to be the highest expected value between the positive and negative distribution (i.e., $\max\{\mathbb{E}[p^+_{uv}], \mathbb{E}[p^-_{uv}]\}$), changing the sign of the weight in case the maximum corresponds to the negative distribution. Since [21] deals with directed networks, our evaluation networks were modified by replacing each edge with two reciprocal directed edges. Also, the objective function of the methods in [21] and [9] corresponds to that used in the Louvain modularity-optimization-based method. For all methods, we used the default parameter setting.

5.1 Results on real data

Interaction loss. Table 1 reports the values of the discounted interaction loss (cf. Eq. (11)). As expected, the clustering solutions of the enhanced methods (i.e., MIL_R and D-MIL_R) show consistently lower loss than the solutions produced by MIL and D-MIL. Also, we observe that, on all datasets, D-MIL outperforms MIL and, in turn, D-MIL_R outperforms MIL_R, which confirms our initial hypothesis that the degree-based heuristic should be preferred on real-world networks. Notably, considering the total average of loss values over all networks (last row in the table), the percentage loss-decrease values obtained by MIL_R are 37% and 28% against MIL and D-MIL, respectively, while the values obtained by D-MIL_R are 39%, 30% and 3% against MIL, D-MIL, and MIL_R, respectively.

Number of clusters. Table 1 also shows the size of the clusterings produced by the various methods. D-MIL always yields a smaller number of clusters than MIL. This happens since, by pivoting over vertices with higher degree, it is more likely to sample vertices having a larger number of incident edges such that $p^+_{uv} > p^-_{uv}$. Also, note that MIL and D-MIL tend to produce more clusters than MIL_R and D-MIL_R, up to 157% and 160%, respectively, of percentage size-increase. This is not surprising since the reduction of loss is related to a decrease in the clustering size.

Table 2: Execution times (in seconds)

	MIL	MIL_R	opt. time	D-MIL	D-MIL_R	opt. time	CPM [21]	GJA [9]	CPMap [7]
Amazon	8.63	347.77	339.14	97.40	427.28	329.88	2 248.9	1 020 122.23	669.114
DBLP	6.11	189.63	183.52	71.15	251.24	180.09	1 570.41	147 159.68	601.044
Epinions	5.90	327.27	321.38	18.90	348.11	329.21	797.71	34 998.9	592.901
High School	0.00	0.04	0.04	0.01	0.04	0.03	0.2	0.19	2.716
Last.fm	0.03	3.48	3.45	0.14	3.72	3.58	7.73	21.54	10.467
PrimarySchool	0.00	0.06	0.05	0.01	0.05	0.04	0.125	0.1	3.698
ProsperLoans	0.70	48.74	48.04	4.31	52.06	47.75	179.78	30 152.47	116.59
StackOverflow	7.88	319.67	311.79	105.68	397.39	291.72	2 465.76	1 140 054.23	1519.943
Wikipedia	2.41	150.03	147.62	19.05	160.69	141.64	826.93	189 345.74	316.438
WikiTalk	13.92	203.49	189.56	129.10	300.68	171.58	1 165.01	650 282.4	NA

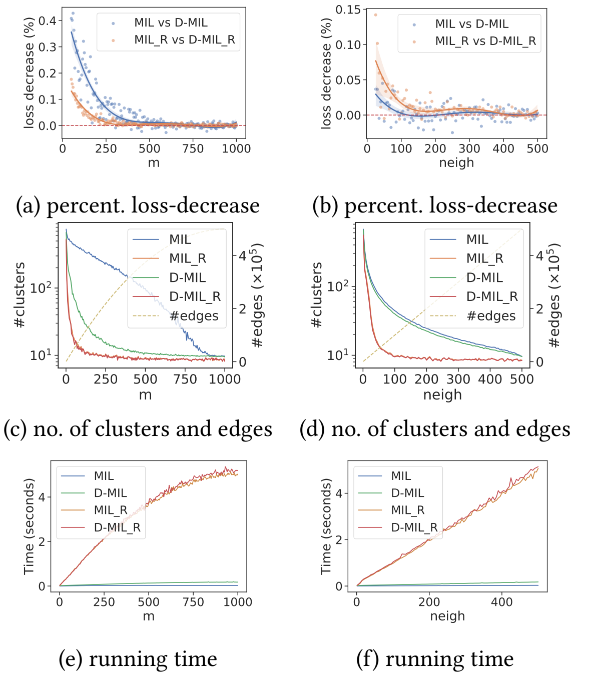
Time performance. Table 2 reports the average time performance of the various methods. For MIL_R and D-MIL_R, we show details about the optimization phase time (i.e., Algorithm 3).[2] Consistently with the computational complexity analysis (Sect. 4), D-MIL tends to perform worse than MIL, and so D-MIL_R against MIL_R. Nevertheless, in PrimarySchool, D-MIL_R runtime is found to be slightly better than MIL_R: this happens since, despite the two methods converge to almost the same local optimum, D-MIL_R starts from a solution which is closer to the final solution as compared to the one produced by MIL_R (cf. Table 1), thus requiring a fewer number of optimization steps (10% decrease), and hence execution time.

5.2 Results on synthetic data

We analyzed loss, clustering size and time performance of the proposed methods, averaged over 100 network-generation runs. Each of the assessment criteria was measured by varying the m parameter for BA networks and the *neigh* parameter for WS networks.

Interaction loss. Figures 2(a)-(b) show the percentage loss-decrease of D-MIL over MIL, and of D-MIL_R over MIL_R. In agreement with the results obtained on real-world networks, the pairwise loss variation is relatively low, for either pair of methods, as long as the network is sparse (i.e., lower m or *neigh* values); more specifically, the percentage loss-decrease of D-MIL w.r.t. MIL is just 0.4% and 0.15% for BA and WS networks, respectively, while corresponding values for D-MIL_R w.r.t. MIL_R are further lower (i.e., below

[2] Experiments were carried out on a Ubuntu 18.04.2 LTS machine with Intel Xeon(R) Gold 5118 CPU @ 2.30GHz × 48 processor and 256GB ram



(a) percent. loss-decrease (b) percent. loss-decrease

(c) no. of clusters and edges (d) no. of clusters and edges

(e) running time (f) running time

Figure 2: Results on BA networks (left side) and on WS networks, with rewiring probability 0.5 (right side).

0.1% for BA and 0.05% for WS). In all cases, the loss variation becomes negligible already for mid regimes of the x-axis. Also, for WS networks having rewiring probability lower than 0.5 (results not shown), the pairwise loss variation would be negligible even for low values of *neigh* (i.e., higher sparsity).

Number of clusters. The clustering size (Figs. 2(c)-(d)) decreases as the number of edges increases with the value of m or *neigh*. MIL always yields a larger number of clusters than the other methods, especially on BA networks still with highest values of m, followed by D-MIL and the enhanced methods, which produce almost the same number of clusters. In general, the difference among the methods is emphasized for sparser networks and decreases as the networks tend to become almost complete.

Time performance. Figures 2(e)-(f) show the running times of the methods. Like for real networks (cf. Table 2), MIL is the fastest method, immediately followed by D-MIL, showing to be very robust as the number of edges (and hence, density) of the network increases, i.e., as m and *neigh* parameter values increase for BA and WS networks, respectively. On the contrary, the enhanced methods achieve higher runtime, with D-MIL_R being slightly slower than MIL_R; nonetheless, they scale linearly on WS networks, and sublinearly on BA networks, with the density of the network.

5.3 Evaluation with competing methods

On real networks, considering the interaction-loss values reported in the last three groups of columns in Table 1, it is worth noticing that our D-MIL_R and MIL_R outperform all competing methods in most cases, with average percentage loss-decreases of 20% for D-MIL_R, resp. 18% for MIL_R, against both CPM and GJA, and 10% for D-MIL_R, resp. 8% for MIL_R, against CPMap. In terms of clustering size, GJA generally produces the lowest number of clusters (6 cases out of 10), though it holds the opposite on average due to the performance on WikiTalk, while CPMap generates solutions with higher size than the others (7 cases out of 10).

Concerning execution times, GJA is the slowest method among the competitors, while CPMap is the fastest. Remarkably, in all cases, the fastest competitor is outperformed by all of our MLI methods, with a minimum gap (w.r.t. D-MIL_R) of 119% time-increase.

Results on synthetic networks obtained by the competitors are shown in *Appendix F*.

6 CONCLUSIONS

We introduced the problem of optimizing the overall interaction in probabilistic graphs under clustering constraints. We theoretically characterized the problem and devised both approximation algorithms and heuristics, whose effectiveness, efficiency and superiority w.r.t. competing methods was assessed in the experiments.

As future work, we plan to extend the problem formulation in order to capture overlapping clusters as well as consider the case when the probability distributions of interaction are not given but only samples coming from that distributions can be observed.

For reproducibility purposes, we make source code and data available at: https://github.com/Ralyhu/optimize_interactions.

REFERENCES

[1] N. Ailon, M. Charikar, and A. Newman. 2008. Aggregating inconsistent information: Ranking and clustering. *JACM* 55, 5 (2008), 23:1–23:27.
[2] N. Bansal, A. Blum, and S. Chawla. 2004. Correlation Clustering. *Machine Learning* 56, 1 (2004), 89–113.
[3] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. 2014. Core decomposition of uncertain graphs. In *Proc. ACM KDD Conf.* 1316–1325.
[4] M. Ceccarello, C. Fantozzi, A. Pietracaprina, G. Pucci, and F. Vandin. 2017. Clustering Uncertain Graphs. *PVLDB* 11, 4 (2017), 472–484.
[5] M. Charikar, V. Guruswami, and A. Wirth. 2005. Clustering with qualitative information. *JCSS* 71, 3 (2005), 360–383.
[6] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. 2006. Correlation clustering in general weighted graphs. *TCS* 361, 2-3 (2006), 172–187.
[7] P. Esmailian and M. Jalili. 2015. Community detection in signed networks: the role of negative ties in different scales. *Scientific reports* 5 (2015), 14339.
[8] Yu G., Chunpeng G., Gao C., and Ge Y. 2014. Effective and Efficient Clustering Methods for Correlated Probabilistic Graphs. *IEEE TKDE* 26, 5 (2014), 1117–1130.
[9] S. Gómez, P. Jensen, and A. Arenas. 2009. Analysis of community structure in networks of correlated data. *Physical Review E* 80, 1 (2009), 016114.
[10] Z. Halim, M. Waqas, and S. F. Hussain. 2015. Clustering large probabilistic graphs using multi-population evolutionary algorithm. *Inf. Sci.* 317 (2015), 78–95.
[11] K. Han, F. Gui, X. Xiao, J. Tang, Y. He, Z. Cao, and H. Huang. 2019. Efficient and Effective Algorithms for Clustering Uncertain Graphs. *PVLDB* 12, 6 (2019), 667–680.
[12] A. Khan, F. Bonchi, A. Gionis, and F. Gullo. 2014. Fast Reliability Search in Uncertain Graphs. In *Proc. EDBT Conf.* 535–546.
[13] A. Khan, F. Bonchi, F. Gullo, and A. Nufer. 2018. Conditional Reliability in Uncertain Graphs. *IEEE TKDE* 30, 11 (2018), 2078–2092.
[14] A. Khan, Y. Ye, and L. Chen. 2018. *On Uncertain Graphs.* Morgan & Claypool.
[15] G. Kollios, M. Potamias, and E. Terzi. 2013. Clustering Large Probabilistic Graphs. *IEEE TKDE* 25, 2 (2013), 325–336.
[16] Y. Li, X. Kong, C. Jia, and J. Li. 2018. Clustering Uncertain Graphs with Node Attributes. In *Proc. ACML Conf.* 232–247.
[17] L. Liu, R. Jin, C. C. Aggarwal, and Y. Shen. 2012. Reliable Clustering on Uncertain Graphs. In *Proc. IEEE ICDM Conf.* 459–468.
[18] D. Pandove, S. Goel, and R. Rani. 2018. Correlation clustering methodologies and their fundamental results. *Expert Systems* 35, 1 (2018).
[19] P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. 2015. Uncertain Graph Processing through Representative Instances. *ACM TODS* 40, 3 (2015), 20:1–20:39.
[20] C. Swamy. 2004. Correlation Clustering: maximizing agreements via semidefinite programming. In *Proc. ACM-SIAM SODA Conf.* 526–527.
[21] V. A. Traag and J. Bruggeman. 2009. Community detection in networks with positive and negative links. *Physical Review E* 80, 3 (2009), 036115.

A APPROXIMATION OF PROBLEM 1

In the following we show that the state-of-the-art (constant-factor) approximation algorithms designed for Max-CC keep their guarantees on Max-Interaction-Clustering too. Theorem 1 states that Max-Interaction-Clustering is an instance of Max-CC. Specifically, as the various p^+_{uv}, p^-_{uv} are general, Max-Interaction-Clustering is an instance of Max-CC with weights $(\omega^+_{uv}, \omega^+_{uv}) \in \mathbb{R}^+_0 \times \mathbb{R}^+_0$, for all $u, v \in V$. Such a variant of Max-CC is not studied in the literature. The closest variant for which theoretical results have been derived is the one where, for every pair (u, v) of vertices, at most one between ω^+_{uv} and ω^-_{uv} is non-zero, i.e., the variant where $(\omega^+_{uv}, \omega^-_{uv}) \in \{(\omega', 0), (0, \omega'')\}_{\omega', \omega'' \in \mathbb{R}^+_0}$, $\forall u, v \in V$ [6, 20]. For this variant, Swamy [20] devises a 0.7666-approximation algorithm based on a semidefinite-programming, and a further, more practical 0.75-approximation algorithm. Next we show that Swamy's approximation result carries over to the Max-CC variant underlying our Max-Interaction-Clustering problem.

Given two interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$, for all $(u, v) \in E$, let $\bar{p}_{uv}$, $\hat{\tau}^+_{uv}$ and $\hat{\tau}^-_{uv}$ be defined as:

$$\bar{p}_{uv} = \min\{\mathbb{E}[p^+_{uv}], \mathbb{E}[p^-_{uv}]\},\ \hat{\tau}^+_{uv} = \mathbb{E}[p^+_{uv}] - \bar{p}_{uv},\ \hat{\tau}^-_{uv} = \mathbb{E}[p^-_{uv}] - \bar{p}_{uv}.$$

Thus, by definition, $(\hat{\tau}^+_{uv}, \hat{\tau}^-_{uv}) \in \{(\omega', 0), (0, \omega'')\}_{\omega', \omega'' \in \mathbb{R}^+_0}$, $\forall u, v \in V$, like in Swamy's setting. Moreover, the objective function $\bar{f}(\cdot)$ of Max-Interaction-Clustering can be rewritten as:

$$\begin{aligned}
\bar{f}(\mathcal{G}_C) &= \sum_{\substack{(u,v)\in E,\\ C(u)=C(v)}} \mathbb{E}[p^+_{uv}] + \sum_{\substack{(u,v)\in E,\\ C(u)\neq C(v)}} \mathbb{E}[p^-_{uv}] \qquad \{\text{Theorem 1}\}\\
&= \sum_{\substack{(u,v)\in E,\\ C(u)=C(v)}} \left(\hat{\tau}^+_{uv} + \bar{p}_{uv}\right) + \sum_{\substack{(u,v)\in E,\\ C(u)\neq C(v)}} \left(\hat{\tau}^-_{uv} + \bar{p}_{uv}\right)\\
&= \sum_{\substack{(u,v)\in E,\\ C(u)=C(v)}} \hat{\tau}^+_{uv} + \sum_{\substack{(u,v)\in E,\\ C(u)\neq C(v)}} \hat{\tau}^-_{uv} + \underbrace{\sum_{(u,v)\in E} \bar{p}_{uv}}_{H(\mathcal{G}^+, \mathcal{G}^-)}\\
&= \underbrace{\sum_{\substack{(u,v)\in E,\\ C(u)=C(v)}} \hat{\tau}^+_{uv} + \sum_{\substack{(u,v)\in E,\\ C(u)\neq C(v)}} \hat{\tau}^-_{uv}}_{:= h(\mathcal{G}_C)} + \underbrace{H(\mathcal{G}^+, \mathcal{G}^-)}_{\text{constant} \geq 0}. \qquad (17)
\end{aligned}$$

As a result, Max-Interaction-Clustering's objective function $\bar{f}(\cdot)$ corresponds to the sum of the objective function of Max-CC (where the weights assigned to every pair (u, v) of vertices are $\hat{\tau}^+_{uv}$ and $\hat{\tau}^-_{uv}$) plus a nonnegative constant. Hence, Max-Interaction-Clustering and Max-CC are equivalent in terms of optimal value. Specifically, let $I_1 = \langle \mathcal{G}^+, \mathcal{G}^- \rangle$ be an instance of Max-Interaction-Clustering, and $I_2 = \langle V, \{\hat{\tau}^+_{uv}\}_{u,v\in V}, \{\hat{\tau}^-_{uv}\}_{u,v\in V} \rangle$ be an instance of Max-CC derived from I_1 by employing the weights defined above. Let also $C^*_{\bar{f}}$ and C^*_h be the optimal clusterings for the I_1 instance according to the $\bar{f}(\cdot)$ and $h(\cdot)$ functions, respectively. Finally, let $\widetilde{C}$ denote the clustering yielded by the given α-approximation algorithm for Max-CC on input I_2 (e.g., aforementioned factor-0.7666 Swamy's algorithm [20]). By definition of approximation algorithm, we know that, for every input: $h(\mathcal{G}_{\widetilde{C}}) \geq \alpha \times h(\mathcal{G}_{C^*_h})$, where $\alpha \leq 1$. Therefore, it holds that:

$$h(\mathcal{G}_{\widetilde{C}}) \geq \alpha \times h(\mathcal{G}_{C^*_h}) \Leftrightarrow h(\mathcal{G}_{\widetilde{C}}) + H(\mathcal{G}^+, \mathcal{G}^-) \geq \alpha \times h(\mathcal{G}_{C^*_h}) + H(\mathcal{G}^+, \mathcal{G}^-)$$

$$\Rightarrow h(\mathcal{G}_{\widetilde{C}}) + H(\mathcal{G}^+, \mathcal{G}^-) \ \geq\ \alpha \times \left(h(\mathcal{G}_{C^*_h}) + H(\mathcal{G}^+, \mathcal{G}^-)\right)$$

$$\Leftrightarrow \bar{f}(\mathcal{G}_{\widetilde{C}}) \ \geq\ \alpha \times \bar{f}(\mathcal{G}_{C^*_g}) \Leftrightarrow \ \bar{f}(\mathcal{G}_{\widetilde{C}}) \ \geq\ \alpha \times \bar{f}(\mathcal{G}_{C^*_{\bar{f}}}).$$

B PROOF OF LEMMA 2

Assuming w.l.o.g. $\mathbb{E}[p^+_{uv}] = \mathbb{E}[p^-_{uv}] = 0$, for all $(u, v) \notin E$, it holds that:

$$\begin{aligned}
\bar{\ell}(\mathcal{G}_C) &= \sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^+_{uv}]\right) + \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - \mathbb{E}[p^-_{uv}]\right) \qquad \{\text{Theorem 4}\}\\
&= \sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - M(\mathcal{G}^+\!,\mathcal{G}^-)\tau^+_{uv} + \frac{\sigma_{uv}}{2}\right) + \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - M(\mathcal{G}^+\!,\mathcal{G}^-)\tau^-_{uv} + \frac{\sigma_{uv}}{2}\right)\\
&= M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \frac{1}{M(\mathcal{G}^+\!,\mathcal{G}^-)} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - M(\mathcal{G}^+\!,\mathcal{G}^-)\tau^+_{uv} + \frac{\sigma_{uv}}{2}\right) +\\
&\quad + M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \frac{1}{M(\mathcal{G}^+\!,\mathcal{G}^-)} \left(M(\mathcal{G}^+\!,\mathcal{G}^-) - M(\mathcal{G}^+\!,\mathcal{G}^-)\tau^-_{uv} + \frac{\sigma_{uv}}{2}\right)\\
&= M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \left(1 - \tau^+_{uv} + \frac{\sigma_{uv}}{2M(\mathcal{G}^+\!,\mathcal{G}^-)}\right) + M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \left(1 - \tau^-_{uv} + \frac{\sigma_{uv}}{2M(\mathcal{G}^+\!,\mathcal{G}^-)}\right)\\
&= M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \underbrace{(1 - \tau^+_{uv})}_{= \tau^-_{uv}} + M(\mathcal{G}^+\!,\mathcal{G}^-) \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \underbrace{(1 - \tau^-_{uv})}_{= \tau^+_{uv}} + \underbrace{\sum_{u,v\in V} \frac{\sigma_{uv}}{2}}_{= K(\mathcal{G}^+\!,\mathcal{G}^-)}\\
&= \underbrace{M(\mathcal{G}^+\!,\mathcal{G}^-) \Bigg(\sum_{\substack{u,v\in V,\\ C(u)=C(v)}} \tau^-_{uv} + \sum_{\substack{u,v\in V,\\ C(u)\neq C(v)}} \tau^+_{uv} \Bigg)}_{= g(\mathcal{G}_C)} + K(\mathcal{G}^+\!,\mathcal{G}^-). \qquad \square
\end{aligned}$$

C HILL CLIMBING DETAILS

Algorithm 3 shows the pseudocode of the hill-climbing refinement for MIL and D-MIL methods (cf. Section 4.3).

Algorithm 3 HillClimbing

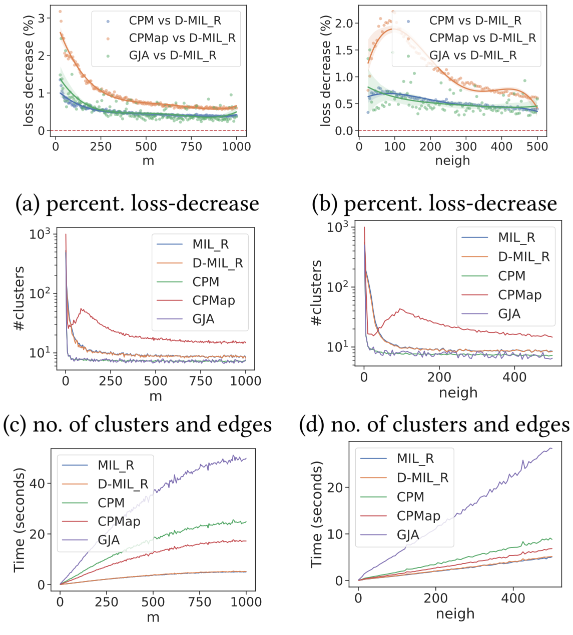
```
Input: Interaction graphs $\mathcal{G}^+ = (V, E, P^+)$, $\mathcal{G}^- = (V, E, P^-)$; A clustering $C$ of $V$; An integer $I > 0$
Output: A clustering $C'$ of $V$
$C' \leftarrow C$
for all $i = 1, \ldots, I$ do
    for every $u \in V$ let $C_u \in C'$ the cluster of $C'$ where $u$ belongs to
    pick $u \in V$ and cluster $C'_u \in C'$ ($C'_u \neq C_u$) that minimize Eq. (16)
    $C'' \leftarrow$ clustering obtained from $C'$ by moving $u$ from $C_u$ to $C'_u$
    if $\bar{\ell}(\mathcal{G}_{C''}) < \bar{\ell}(\mathcal{G}_{C'})$ then
        $C' \leftarrow C''$
```

D TIME COMPLEXITY

Proposition 1: MIL runs in $O(|V| + |E|)$ time. The vertex-sampling step (Line 4 of Algorithm 1) can be implemented so as to take $O(|V|)$ time overall, by preliminarily generating a random permutation of V (e.g., via $O(|V|)$-time *Fisher-Yates shuffle* algorithm), and picking vertices u according to the ordering of that permutation. Also, the computation of τ^+_{uv}, τ^-_{uv} (Line 1 of Algorithm 1) can be restricted to the linked (u, v) pairs; in fact, for $(u, v) \notin E$, it holds that $\tau^+_{uv} = \tau^-_{uv}$, thus, in the next cluster-building step (Line 5 of Algorithm 1) the vertices v such that $(u, v) \notin E$ can be discarded. This makes the weight-computation and cluster-building take $O(|E|)$ time overall.

Table 3: Summary of real networks used in our evaluation: original data (cols. 2-5) and preprocessed data (cols. 6-7)

	$\|V\|$	$\sum_{t=1}^{T} \|E_t\|$	T	edge semantics	$\|E\|$	$\%\{\Delta^+_{uv} > 0\}$
Amazon	2 146 057	22 728 036	115	co-rating	22 507 680	50
DBLP	1 824 701	11 865 584	80	co-authorship	8 344 615	52
Epinions	120 492	33 412 111	25	co-rating	24 994 363	50
HighSchool	327	47 589	1212	face-to-face	5 818	69
Last.fm	992	4 342 951	77	co-listening	369 973	50
PrimarySchool	242	55 043	390	face-to-face	8 317	66
ProsperLoans	89 269	3 343 271	307	economic	3 330 022	50
StackOverflow	2 433 067	16 200 209	51	Q/A	15 786 816	49
Wikipedia	343 860	18 086 734	101	co-editing	10 519 921	50
WikiTalk	2 863 439	10 335 318	192	communication	8 146 544	54



(a) percent. loss-decrease (b) percent. loss-decrease

(c) no. of clusters and edges (d) no. of clusters and edges

(e) running time (f) running time

Figure 3: Results on BA networks (left side) and on WS networks, with rewiring probability 0.5 (right side).

Proposition 2: Sampling a vertex u with probability proportional to its (current) $d_{V'}(u)$ degree (cf. line 4 in Algorithm 2) can be implemented with a priority queue Q with priorities $d_{V'}(u) \times rnd$, where rnd is a random number. At the beginning, all vertices V are added to Q. Pivots are sampled following the order upon which vertices are extracted from Q, discarding vertices that have been assigned to clusters beforehand. Initializing Q and extracting all vertices from it takes $O(|E| + |V| \log |V|)$ time. Building all the clusters takes $O(|E|)$ time overall, as each cluster requires accessing the neighbors of the pivot $O(1)$ times. Updating the degrees and the priorities of vertices in Q after a cluster has been built (and removed) takes $O(|E| \log |V|)$ time. As a result, the overall time complexity of D-MIL is $O(|E| \log |V|)$.

Proposition 3: Computing Equation (16) for all vertices takes $O(|V| + |E|)$ time, as, for every vertex u, it requires processing u’s neighbors only. Hence, denoting by I the number of iterations of the process, the overall time complexity of Algorithm 3 is $O(I(|V| + |E|))$.

E DATA AND SETTINGS

Table 3 summarizes main structural characteristics of the real-world networks used in our evaluation. Each of the input temporal networks is treated as a sequence of undirected snapshot graphs $\langle G_1, \ldots, G_T \rangle$, where each $G_t = (V, E_t)$ $(t = 1..T)$ models the vertex interactions at time t. We defined the interaction graphs $\mathcal{G}^+ = (V, E, P^+)$ and $\mathcal{G}^- = (V, E, P^-)$ as follows. The topology of the two graphs was derived by “flattening” the temporal network, i.e., $(u, v) \in E$ if u and v are linked in at least one graph from $\langle G_1, \ldots, G_T \rangle$; For each pair $u, v \in V$, if $(u, v) \notin E$ we assume that the two vertices will have no interaction with probability one, otherwise (i.e., $(u, v) \in E$) we define the distributions p^+_{uv}, p^-_{uv} as:

$$p^+_{uv}(w) = \frac{\Pr[w_G(u, v) = w \wedge C(u) = C(v)]}{\Pr[C(u) = C(v)]} \tag{18}$$

$$p^-_{uv}(w) = \frac{\Pr[w_G(u, v) = w \wedge C(u) \neq C(v)]}{\Pr[C(u) \neq C(v)]} \tag{19}$$

for $w \in \mathcal{D}(p_{uv})$, with $G \sqsubseteq \mathcal{G}_C$ possible world induced by C from $\mathcal{G}$.

To estimate the above probabilities, we first derived a clustering solution on each graph from $\langle G_1, \ldots, G_T \rangle$, by initially assigning each vertex to a singleton cluster (i.e., starting from a solution totally biased towards the distributions in P^-), then iteratively performing agglomerative hierarchical clustering based on the minimization of a criterion function defined as the absolute value of the difference between the sum of the number of edges internal to each cluster and the sum of the number of edges external to each cluster. Although simple, this criterion function is better suited to our setting than classic community-detection approaches, such as *modularity*-based optimization criteria, which compares the actual within-community connectivity with the expected one based on a null model.

Once obtained the clustering solution on each G_t, we finally estimated $p^+_{uv}(w)$, resp. $p^-_{uv}(w)$, as the fraction of the timestamped graphs where u and v shared the same cluster, resp. were not in the same cluster, that corresponds to the interaction strength equal to w. The intuition for the definition of $p^+_{uv}(w)$ is that the more frequently u and v were grouped into the same cluster and their observed strength of interaction was w, the higher the probability that they will interact with strength w if they would be assigned to the same group; analogously for the functions p^-_{uv}. In our evaluation, we considered binary distribution functions; in this regard, note that the last column in Table 3 denotes the percentage of edges (u, v), in each network, such that $\mathbb{E}[p^+_{uv}] > \mathbb{E}[p^-_{uv}]$.

For both BA and WS models, we generated networks with 1000 vertices. For the BA model, we varied m from 0 to 1000, with steps of 5, for a total of 200 BA networks generated; analogously, for the WS model, we varied *neigh* from 0 to 1000/2=500, for a total of 100 WS networks generated. The expected values of interaction were randomly generated (uniformly) between 0 and 1.

F COMPETITORS ON SYNTHETIC DATA

Figures 3(a)-(b) show the percentage loss decrease of D-MIL_R against each competitor, which is always positive. Both CPM and GJA produce fewer clusters than the other methods, whereas CPMap, except for low m and *neigh*, yields the highest number of clusters (c.f. Figs. 3(c)-(d)). CPMap is the fastest method among competitors, followed by CPM and GJA (c.f. Figs. 3(e)-(f)). All competitors are anyway outperformed by MIL_R and D-MIL_R.