



A combinatorial multi-armed bandit approach to correlation clustering

F. Gullo¹ · D. Mandaglio² · A. Tagarelli²

Received: 10 December 2021 / Accepted: 6 April 2023 / Published online: 29 June 2023
© The Author(s) 2023

Abstract

Given a graph whose edges are assigned positive-type and negative-type weights, the problem of *correlation clustering* aims at grouping the graph vertices so as to minimize (resp. maximize) the sum of negative-type (resp. positive-type) intra-cluster weights plus the sum of positive-type (resp. negative-type) inter-cluster weights. In correlation clustering, it is typically assumed that the weights are readily available. This is a rather strong hypothesis, which is unrealistic in several scenarios. To overcome this limitation, in this work we focus on the setting where edge weights of a correlation-clustering instance are unknown, and they have to be estimated in multiple rounds, while performing the clustering. The clustering solutions produced in the various rounds provide a feedback to properly adjust the weight estimates, and the goal is to maximize the cumulative quality of the clusterings. We tackle this problem by resorting to the reinforcement-learning paradigm, and, specifically, we design for the first time a Combinatorial Multi-Armed Bandit (CMAB) framework for correlation clustering. We provide a variety of contributions, namely (1) formulations of the minimization and maximization variants of correlation clustering in a CMAB setting; (2) adaptation of well-established CMAB algorithms to the correlation-clustering context; (3) regret analyses to theoretically bound the accuracy of these algorithms; (4) design of further (heuristic) algorithms to have the probability constraint satisfied at every round (key condition to soundly adopt efficient yet effective algorithms for correlation clustering as CMAB oracles); (5) extensive experimental comparison among a variety of both CMAB and non-CMAB approaches for correlation clustering.

Keywords Correlation clustering · Combinatorial multi-armed bandit · Regret analysis · Exploration and exploitation · Approximation oracle · Minimization of disagreements · Maximization of agreements · Combinatorial lower confidence bound · Probability constraint · Expected cumulative loss

Responsible editor: Albrecht Zimmermann.

Extended author information available on the last page of the article

1 Introduction

Correlation clustering is the problem of clustering the vertices of a graph whose edges are assigned *positive-type* and *negative-type* real-valued weights that express, respectively, positive and negative evidence of placing the endpoints of an edge in the same cluster (Bansal et al. 2004; Bonchi et al. 2022). Two formulations of correlation clustering exist: the minimization one (MIN-CC) aims at minimizing the sum of the negative-type intra-cluster edge weights plus the sum of the positive-type inter-cluster edge weights; in the maximization counterpart (MAX-CC), the objective is dual, i.e., to maximize the sum of the positive-type intra-cluster edge weights plus the sum of the negative-type inter-cluster edge weights. Correlation clustering has been extensively studied from a theoretical point of view, and it has been applied in numerous real-world scenarios (Bonchi et al. 2014; Pandove et al. 2018).

Correlation clustering with unknown edge weights. Traditionally, in correlation clustering it is assumed that the edge weights are all given as input; for instance, they could have been derived from, e.g., past user-interaction history, crowdsourcing, experimental trials, and so on. This has the disadvantage that clustering has to be performed after that all the weights are available, which is unfeasible in several real contexts. To overcome this, here we focus *for the first time* on a correlation-clustering setting where *edge-weight assessment is carried out while performing the clustering*.

We devise the following scenario. Edge weights are random variables whose probability distributions and means are unknown and do not change during the whole process. An estimate of the mean of the edge-weight distributions is maintained. Initial estimates are randomly generated or computed based on prior knowledge. There are multiple rounds of clustering. A clustering performed at any round gives *feedback* on how to adjust the mean estimates, so as to make them improve round after round. The rationale is that, placing a clustering, *actual* interactions among the vertices can be observed, and hence used as a *real evidence* to profitably update the mean estimates. More specifically, in MIN-CC (resp. MAX-CC) one gets feedback about the negative-type (resp. positive-type) intra-cluster edge weights and the positive-type (resp. negative-type) inter-cluster edge weights. A clustering at every round may be computed by taking into account the current mean estimates (*exploitation*) based on an (exact or approximate) *oracle*; alternatively, a clustering can be yielded without looking at the mean estimates, so as to get feedback on edge weights for which limited knowledge has been acquired so far (*exploration*). In our context, alternating between exploiting the oracle with estimated weights to determine a clustering and observing the feedbacks on the edges of the graph induced by this clustering, makes it possible to improve the estimate of the edge weights, since more observed data are collected upon which the estimates are computed.

Both exploitation and exploration have pros and cons. The former yields clusterings that rely on established—but partial—knowledge. The latter allows for expanding the current knowledge, which is supposed to yield better-quality clusterings in the next rounds, but it also may lead to possibly inaccurate clusterings in the first rounds.

Getting the best *exploration-exploitation tradeoff* is a key desideratum. The effectiveness of such a tradeoff is measured by the (expected) *cumulative quality of the clusterings produced in all the rounds*. This is the ultimate objective to be optimized, and a major challenge in the design of proper algorithms.

It should be noted that the aforementioned exploration-exploitation tradeoff refers to a reinforcement learning paradigm, which, in this work, we adopt by resorting to the *Combinatorial Multi-Armed Bandit* (CMAB) framework (Chen et al. 2016, 2018a; Kveton et al. 2015a, b; Lagrée et al. 2016; Wang and Chen 2017; Xu et al. 2020). The CMAB framework has been contextualized to several specific problems, including influence maximization (Chen et al. 2016; Vaswani and Lakshmanan 2015; Wu et al. 2019), community detection (Mandaglio and Tagarelli 2019a, b), community exploration (Chen et al. 2018b), shortest-path discovery (Talebi et al. 2017), feature selection (Liu et al. 2021). In our previous work (Mandaglio et al. 2020), we devise *non-CMAB* algorithms for a correlation-clustering problem variant in which interactions between entities are characterized by *known* input probability distributions and conditioned by external factors within the environment where the entities interact. Remarkably, none of those settings is any close to the one we consider in this work, i.e., devising a CMAB framework for (correlation) clustering.

Applications. The domain we consider in this work finds application in all those contexts where it is not preferable (or not permitted) to wait until edge weights have been produced before performing a clustering. Rather, it is desired to place clustering solutions soon, learn the weights along the way, and tolerate that the clustering quality will be less good in the initial rounds, while getting improved as the rounds go by.

For instance, we might consider a *team formation* scenario, where individuals need to be organized (clustered) into teams (Juárez et al. 2022). Individuals are associated with technical/soft skills which are required for task assignments within the teams. Any two individuals exhibiting a certain skill-level similarity should be assigned to the same team, and conversely to different teams if they are dissimilar to each other; clearly, given the variety of skills and their compatibility levels, the exact degree of matching between two individuals' skills are not a-priori known at the beginning of the team formation process, and indeed similarities should be learned through team-formation history. In this regard, individuals collaborate with both their teammates and individuals from other teams, for, e.g., general coordination purposes. A desirable goal is to establish teams so as to maximize the overall (i.e., intra-team plus inter-team) similarity between pairs of individuals. This is a problem that can easily be casted to correlation clustering, where vertices correspond to individuals, clusters correspond to teams, and the positive-type and negative-type edge weights correspond to intra-team and inter-team similarities, respectively. Note that empathy (i.e., mutable) characteristics between individuals are here discarded, since this may cause drift in the likelihood that they (dis)like each other once they are (temporarily) members of the same team, i.e., edge weights would change through the rounds. In this regard, an analogous scenario is the *task allocation* for robots, each of which is programmed to handle a number of operations. Correlation clustering would be helpful to

enable forming coalitions between robots, in order to allocate them to tasks to be completed optimally according to some efficiency requirements.

Two further example scenarios are *commercial scheduling* (e.g., Bollapragada and Garbiras 2004; Giallombardo et al. 2016) and *shelf space allocation* (e.g., Hübner et al. 2021). The former consists in optimally assigning a set of commercials to fill in each advertisement slot scheduled by a TV broadcaster, where vertices are commercials and edge weights denote marketing-driven benefits in assigning, resp. separating, any two commercials within the same, resp. to different slots; edge weights might initially be estimated by accounting for requirements provided by either the brand customers and the TV broadcaster, then the weights will be adjusted by observing the feedback provided by (online) market-surveys (e.g., delivered to targeted audience of the TV programme schedule). Shelf space allocation is to model the shelf space dimensioning and positioning for allocating selected products based on practical retail requirements, so as to maximize the product selling; here, the feedback observed from the selling outcomes, as well as from how the customers welcome or not the retailer choices, would be related to the opportunity of ensuring brand visibility, or improving customers satisfaction.

All the aforementioned scenarios correspond to well-known optimization problems in operation research and related fields; nonetheless, they are also key-enabling in emerging contexts, such as the development of smart production systems brought by Industry 4.0 (Grillo et al. 2022). However, such problems have not commonly been addressed in terms of correlation clustering; and the few existing exceptions (e.g., Dutta et al. (2019)) are far from a CMAB perspective—unlike we study in this work—which needs to be profitably adopted since correlation-clustering weights are unlikely to be a priori known.

Contributions. The one we deal with in this work is a natural reinforcement-learning scenario, which, to the best of our knowledge, *has never been considered in the context of correlation clustering*. We tackle it by designing—for the first time—a *Combinatorial Multi-Armed Bandit (CMAB)* (Chen et al. 2016) *framework for correlation clustering*. In doing so, we achieve a mix of modeling, algorithmic, technical, and empirical contributions, including principled framework design and problem formulations, design and theoretical analysis of algorithms, tricks to make the algorithms work in practice, experimental evaluation. More in details, our main contributions are as follows:

- We novelly formulate correlation clustering in a CMAB setting, by providing a contextualization of the main ingredients of a typical CMAB framework and CMAB formulations for both MIN-CC and MAX-CC (Sect. 3). Among others, a key impactful consequence of this contribution is that it enables the use of generalist CMAB approximation algorithms/heuristics for CMAB-based MIN-CC/MAX-CC with minimum customization effort. In this regard, we show how the popular *Combinatorial Upper Confidence Bound (CUCB)* method (Chen et al. 2016) can be employed in the context of MAX-CC (Appendix A.2).
- We introduce the *Combinatorial Lower Confidence Bound (CLCB)* method, which can be viewed as the counterpart of CUCB for minimization problems,

and show how to suitably customize it in order to handle MIN-CC instances (Sect. 4).

- The effectiveness of a CMAB algorithm is typically assessed in terms of *regret*, i.e., a measure of how far the (expected) cumulative quality of the solutions yielded by an algorithm is from the optimal cumulative quality. In this regard, Chen et al. (2016) provide a *regret analysis* of the CUCB method, which shows that, if the underlying combinatorial-optimization problem satisfies certain properties, CUCB is guaranteed to achieve a regret that is at most *logarithmic* in the number of clustering rounds, in presence of an *approximation oracle*. Here, we build upon Chen *et al.*'s result and show that:
 - Our CMAB formulation of MAX-CC satisfies Chen et al.'s properties, thus, CUCB achieves logarithmic regret for MAX-CC as well (Appendix A.2.1).
 - We devise a principled *regret definition* for MIN-CC. According to this definition, we also provide a regret analysis that, along the lines of Chen *et al.*'s analysis for CUCB, proves that CLCB achieves *logarithmic* regret in the number of clustering rounds for MIN-CC in presence of an approximation oracle (Sect. 4.1). Our regret definition and analysis for MIN-CC are general enough to be reused in any minimization CMAB problems with approximation oracle. *This is a per-se contribution*, as, to the best of our knowledge, no regret definitions/analyses for CMAB minimization problems (with approximation oracle) exist in the literature.
- We further investigate the applicability of the CLCB-like algorithm in practice (Sect. 4.2). A key desideratum in this regard is to employ the traditional PIVOT algorithm for MIN-CC (Ailon et al. 2008) as an (approximation) oracle within CLCB, for its efficiency, theoretical yet empirical effectiveness, and ease of implementation. A major challenge here is that, to achieve its approximation guarantees (and to provide effective solutions in practice too), PIVOT needs the input edge weights to satisfy the *probability constraint*. Unfortunately, the CLCB algorithm does not guarantee the fulfilment of this constraint at every round. Thus, we design novel variants of the basic CLCB where the correlation-clustering problem instances to be given as input to PIVOT (are close to) meet the probability constraint.
- We conduct an extensive evaluation to experimentally test the performance of CMAB correlation-clustering algorithms, including the algorithms devised in this work, as well as (correlation-clustering-customized) popular CMAB heuristics, such as *ϵ -greedy*, *pure exploitation*, and *Combinatorial Thompson Sampling* (Wang and Chen 2018) (Sects. 5–7). We consider the MIN-CC context only, due to the availability of practical approximation oracles (unlike MAX-CC). Results show superior and close accuracy of CMAB methods over non-CMAB baselines and a reference method that performs correlation clustering with the true edge weights, respectively. Also, the per-round runtime of CMAB methods is (at worst) comparable to the runtime of executing a linear-time correlation-clustering algorithm once.

Section 2 discusses background and related work. Section 8 concludes the paper.

2 Background and related work

2.1 Correlation clustering

The minimization (MIN-CC) and maximization (MAX-CC) formulations of correlation clustering aim at *minimizing disagreements* and *maximizing agreements*, respectively. They are formally defined as follows:

Problem 1 (MIN-CC (Ailon et al. 2008)) Given a graph $G = (V, E)$, and nonnegative weights $w_{uv}^+, w_{uv}^- \in \mathbb{R}_0^+$ for each edge $(u, v) \in E$, find a clustering $C^* : V \rightarrow \mathbb{N}^+$ such that:

$$C^* = \operatorname{argmin}_C f_{\min}(C) = \operatorname{argmin}_C \sum_{\substack{(u, v) \in E, \\ C(u) = C(v)}} w_{uv}^- + \sum_{\substack{(u, v) \in E, \\ C(u) \neq C(v)}} w_{uv}^+. \tag{1}$$

Problem 2 (MAX-CC (Ailon et al. 2008)) Given a graph $G = (V, E)$, and nonnegative weights $w_{uv}^+, w_{uv}^- \in \mathbb{R}_0^+$ for each edge $(u, v) \in E$, find a clustering $C^* : V \rightarrow \mathbb{N}^+$ such that:

$$C^* = \operatorname{argmax}_C f_{\max}(C) = \operatorname{argmax}_C \sum_{\substack{(u, v) \in E, \\ C(u) = C(v)}} w_{uv}^+ + \sum_{\substack{(u, v) \in E, \\ C(u) \neq C(v)}} w_{uv}^-. \tag{2}$$

In the above problems, and hereinafter, we let a clustering be represented as an injective function that expresses *cluster-membership* for the vertices in V .

MIN-CC and MAX-CC are equivalent in terms of optimality and complexity class [both NP-hard (Bansal et al. 2004; Shamir et al. 2004)], but have different approximation-guarantee properties, with the latter being easier in this regard. On general edge weights, both MIN-CC and MAX-CC are APX-hard (Charikar et al. 2005), with MAX-CC admitting constant-factor approximation algorithms (Charikar et al. 2005; Swamy 2004), and with the best known approximation factor for MIN-CC being $\mathcal{O}(\log |V|)$ (and unlikely to be meliorable) (Charikar et al. 2005; Demaine et al. 2006).

When restrictions on weights are imposed, the problems become more tractable. For instance, in the seminal work by Bansal et al. (2004), which requires the input graph to be complete, and the weights to be binary and with exactly one nonzero weight for each weight pair (i.e., $\forall u, v \in V, (w_{uv}^+, w_{uv}^-) \in \{(0, 1), (1, 0)\}$), MAX-CC admits a PTAS (Bansal et al. 2004), and MIN-CC admits constant-factor approximations (Ailon et al. 2008; Bansal et al. 2004; Charikar et al. 2005; Chawla et al. 2015; van Zuylen and Williamson 2007) [although still remaining APX-hard (Charikar et al. 2005)].

Attention has also been devoted to weight bounds that go beyond Bansal et al.'s ones, but are still restrictive enough to allow MIN-CC to achieve constant-factor

guarantee. These include weights satisfying the **probability constraint** (i.e., $w_{uv}^+ + w_{uv}^- = 1, \forall u, v \in V$) (Ailon et al. 2008), generalizations of it (i.e., $\forall u, v \in V, w_{uv}^+ \leq 1, w_{uv}^- \leq h$ for some $h \in [1, +\infty)$, and $w_{uv}^+ + w_{uv}^- \geq 1$) (Puleo and Milenkovic 2015), *triangle inequality* (i.e., $w_{uz}^- \leq w_{uv}^- + w_{vz}^-, \forall u, v, z \in V$) (Ailon et al. 2008), or *global constraints* (Mandaglio et al. 2021). The probability constraint is particularly appealing: in fact, under such a constraint, **Pivot**—a randomized algorithm for **MIN-CC** that is widely recognized for its theoretical guarantees, efficiency, and ease of implementation—achieves a 5-approximation (in expectation) (Ailon et al. 2008). Coupling the probability constraint with triangle inequality lowers **Pivot**'s (expected) approximation factor to 2 (Ailon et al. 2008).

Although considering various types of weight, all the above works still assume that *edge weights are all available as input*. In this work, we go beyond this limitative view, and focus on the context where edge weights are not available beforehand, but they have to be discovered while performing (multiple rounds of) clustering.

Beyond basic correlation clustering. Several extensions to the basic correlation-clustering formulations have been studied, including constrained/relaxed formulations (e.g., constraining the number/size of clusters, allowing overlapping clusters), and adaptations to more sophisticated types of graph (e.g., bipartite graphs, labeled graphs, multilayer graphs, hypergraphs) or nonconventional computational settings (e.g., online, parallel, streaming). We point the interested reader to Bonchi et al. (2014, 2022); Pandove et al. (2018) for more details on these advanced topics. Here, let us just discuss the problem of *query-efficient correlation clustering* (Bressan et al. 2019; García-Soriano et al. 2020), which, to our knowledge, is the only correlation-clustering extension that exhibits some (slight) similarity with the setting we study in this work. Query-efficient correlation clustering assumes that edge weights are discovered by querying an *oracle*, and the goal is to cluster the input graph by using a limited budget of Q queries ($Q \ll \mathcal{O}(|V|^2)$). Although it is still assumed that edge weights are not available beforehand (like in our setting), query-efficient correlation clustering focuses on a scenario that remains profoundly different from the one tackled in this work. In fact, it considers a hard limit Q on the number of edge weights that can be ultimately discovered, which is a restriction that is not present in our setting. Moreover, the feedback on edge weights is given by an oracle, which provides *true* edge weights for any query, at any time. Instead, in our setting, the feedback consists in a *sample of the weight distributions that are used to update the weight estimates*, and is provided by the *clustering itself* (there is no oracle). Finally, existing approaches to query-efficient correlation clustering (Bressan et al. 2019; García-Soriano et al. 2020) handle binary weights only.

2.2 Combinatorial multi-armed bandit

Combinatorial Multi-Armed Bandit (CMAB) is a popular *reinforcement-learning* framework to learn how to perform actions by exploring/exploiting the feedback from an environment (Chen et al. 2016). It extends basic *Multi-Armed Bandit* (MAB) (Berry and Fristedt 1985) so that the actions to be performed/learned correspond to combinatorial structures (*superarms*) that are defined on top of simpler, basic actions (*base arms*). Specifically, a CMAB instance consists of m *base*

arms. Each base arm i is assigned a set $\{X_{i,t} \mid 1 \leq i \leq m, 1 \leq t \leq T\}$ of random variables, where T is the number of rounds. The support of each $X_{i,t}$ —assumed to range from $[0, 1]$ —indicates the random “outcome” of playing base arm i in round t . This outcome is interpreted as a feedback from the environment and used to carry out the learning process. The random variables $\{X_{i,t}\}_{t=1}^T$ of the same arm i are *independent and identically distributed*, according to some *unknown* distribution with *unknown* expectation μ_i . Random variables of different base arms may be dependent or distributed with different laws. Estimates $\{\hat{\mu}_i\}_{i=1}^m$ of the true unknown $\{\mu_i\}_{i=1}^m$ expectations are kept (and updated) at every round.

A CMAB instance also includes a set $\mathcal{A} \subseteq 2^{[m]}$ of possible *superarms*. \mathcal{A} is typically defined as the subset of all subsets of base arms satisfying certain constraints. At each round t , a superarm $A_t \in \mathcal{A}$ is played and the outcomes of the random variables $X_{j,t}$, for all the base arms $j \in A_t$, are observed. These outcomes can be used to update the knowledge on the estimates $\{\hat{\mu}_j\}_{j \in A_t}$. Playing a superarm A_t gives a *reward* $R_t(A_t)$, which is a random variable defined as a function of the outcomes of A_t 's base arms. $R_t(A_t)$ may simply be a summation $\sum_{j \in A_t} X_{j,t}$ of the outcomes of A_t 's base arms, but more complex (possibly nonlinear) rewards are allowed. In any case, it is often assumed that the expectation $\mathbb{E}[R_t(A_t)]$ is a function of *only* A_t 's base arms and all the $\{\mu_i\}_{i=1}^m$ (true) expectations. For minimization problems, the reward can be replaced by a notion $L_t(A_t)$ of *loss*. The adaptation is straightforward.

The objective of a CMAB algorithm is to select a superarm to be played at every round, so as to *maximize the cumulative expected reward* obtained in all the rounds, i.e., $\mathbb{E}[\sum_{t=1}^T R_t(A_t)]$. With this ultimate goal in place, a superarm A_t can be chosen by either *exploiting* the knowledge acquired from the outcomes of previous rounds, or *exploring* arms that have not been played much. Here is the *exploration-exploitation* tradeoff that usually appears in reinforcement-learning scenarios: a key design principle of any CMAB algorithm consists in deciding to what extent it should pick the arms that have provided good rewards so far (exploitation) or select different arms with the aim of getting even better rewards (exploration).

As for exploitation-aware superarms, it is assumed the availability of an *oracle* that computes a superarm based on the current estimates $\{\hat{\mu}_i\}_{i=1}^m$ of the base-arm expectations and the knowledge it possesses on the specific problem at hand. The oracle can be *exact*, i.e., it outputs $A_t^* = \operatorname{argmax}_{A \subseteq \mathcal{A}} \mathbb{E}[\hat{R}_t(A)]$, or an (α, β) -*approximation* one, for some $\alpha, \beta \leq 1$, i.e., it outputs a superarm A_t so that $\Pr[\mathbb{E}[\hat{R}_t(A_t)] \geq \alpha \mathbb{E}[\hat{R}_t(A_t^*)]] \geq \beta$ (where $\hat{R}_t(\cdot)$ denotes the reward computed based on $\{\hat{\mu}_i\}_{i=1}^m$).

The effectiveness of a (C)MAB algorithm is typically measured in terms of the so-called *regret* metric, which corresponds to the difference in the cumulative expected reward between always playing the optimal arm (possibly scaled by factors α and β in case of (α, β) -approximation oracles) and playing arms according to the algorithm. A major theoretical desideratum in this regard consists in providing a suitable *regret analysis*, which guarantees that the algorithm at hand achieves a certain bounded regret. The seminal work by Chen et al. (2016) shows that it is possible to design CMAB algorithms achieving $\mathcal{O}(\log T)$ regret,

and that this is a tight bound.

Regret definitions and analyses for CMAB maximization problems exist for both exact and approximation oracles (Chen et al. 2016; Wang and Chen 2017). As for minimization problems, to the best of our knowledge, they have been devised for exact oracles only (Cesa-Bianchi and Lugosi 2012; Talebi et al. 2017). In this work, we provide for the first time a regret analysis for a minimization problem (MIN-CC) with approximation oracle. The generality of our regret definition and analysis make us believe that this is a contribution of interest for CMAB minimization problems in general, not only for (correlation) clustering.

3 Problem definition

In this section we provide the details of the proposed contextualization of CMAB to correlation clustering. As a first step, we let the weights w_e^+, w_e^- of every edge $e \in E$ be modeled as *random variables* W_e^+, W_e^- with $[0, 1]$ support,¹ and mean

$$\mu = \{\mu^+, \mu^-\}, \quad \mu^+ = \{\mu_e^+ = \mathbb{E}[W_e^+]\}_{e \in E}, \quad \mu^- = \{\mu_e^- = \mathbb{E}[W_e^-]\}_{e \in E}. \quad (3)$$

All such random variables and their means are assumed to be unknown (as typical in CMAB), and not to change in the various clustering rounds. Any CMAB algorithm keeps *estimates* of the true means, which are denoted as:

$$\hat{\mu} = \{\hat{\mu}^+, \hat{\mu}^-\}, \quad \hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}, \quad \hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}. \quad (4)$$

Let also every edge $e = (u, v) \in E$ be represented by a pair of *replicas*, e^{in} and e^{out} , which model the fact that e is an intra-cluster or inter-cluster edge (with respect to a given clustering), respectively. Let $\mathcal{S}^{in} = \{e^{in} \mid e \in E\}$ and $\mathcal{S}^{out} = \{e^{out} \mid e \in E\}$ be the sets of all intra-cluster and inter-cluster edge replicas, respectively. We make the *base arms* in CMAB correlation clustering correspond to the set $S = \mathcal{S}^{in} \cup \mathcal{S}^{out}$ of all edge replicas (thus, the number of base arms is $m = 2|E|$), and a *superarm* be identified by a set of base arms that are *consistent with the notion of clustering*. Formally, a superarm corresponds to a *clustering-compliant replica set*:

Definition 1 (*Clustering-compliant replica set*) A set $S \subseteq \mathcal{S}$ of edge replicas is *clustering-compliant* if (i) for all $e \in E$, S does not contain both e^{in} and e^{out} , and (ii) for all $e_1 = (x, y), e_2 = (y, z), e_3 = (x, z) \in E$, if $e_1^{in}, e_2^{in} \in S$, then $e_3^{in} \in S$.

In the above definition, (i) is because an edge cannot be both intra-cluster and inter-cluster, while (ii) guarantees the transitive property that if vertices x, y are within the same cluster and y, z are within the same cluster, then x, z must be in the

¹ The $[0, 1]$ -support assumption is frequently required to prove theoretical results in the CMAB setting (Chen et al. 2016), and is indeed required in our proofs too. In practice, should this assumption not hold, any $[0, 1]$ -normalization of the input edge weights can be performed beforehand.

same cluster too. Simply speaking, a superarm corresponds to a clustering. Thus, we hereinafter refer to “superarm” and “clustering” as two equivalent notions.

The *outcome* of the base arms that are triggered while playing a superarm depends on the correlation-clustering formulation. In MIN-CC, the outcome of every intra-cluster edge replica e^{in} comes from the corresponding negative-type-weight W_e^- random variable, while the outcome of every inter-cluster edge replica e^{out} comes from the corresponding positive-type-weight W_e^+ random variable. The rationale is that, in MIN-CC, the clustering quality is measured in terms of the negative-type weight of all intra-cluster edges and the positive-type weight of all the inter-cluster edges. Thus, placing a clustering (i.e., playing a superarm) is expected to give a feedback that is consistent with MIN-CC’s objective function: the outcome of e^{in} (resp. e^{out}) replicas should be used to update μ_e^- (resp. μ_e^+). Conversely, in MAX-CC, e^{in} and e^{out} are assigned (and their outcome come from) W_e^+ and W_e^- , respectively.

The *reward/loss* corresponds to the correlation-clustering objective function, hence its definition depends on the correlation-clustering formulation too. Given a superarm S , let S^{in} and S^{out} denote the intra-cluster and inter-cluster edge replicas in S , respectively. MIN-CC utilizes a disagreement-based loss $d(S)$ defined as:

$$d(S) = \sum_{e \in S^{in}} W_e^- + \sum_{e \in S^{out}} W_e^+, \tag{5}$$

while MAX-CC employs a reward $a(C)$ defined in terms of agreements as:

$$a(S) = \sum_{e \in S^{in}} W_e^+ + \sum_{e \in S^{out}} W_e^-. \tag{6}$$

The expectations of $d(\cdot)$ and $a(\cdot)$ are as follows (by linearity of the expectation):

$$\bar{d}_\mu(S) = \mathbb{E}[d(S)] = \sum_{e \in S^{in}} \mu_e^- + \sum_{e \in S^{out}} \mu_e^+, \quad \bar{a}_\mu(S) = \mathbb{E}[a(S)] = \sum_{e \in S^{in}} \mu_e^+ + \sum_{e \in S^{out}} \mu_e^-. \tag{7}$$

where the “ μ ” subscript in \bar{d}_μ and \bar{a}_μ is to emphasize that those functions depend on the *true* means μ . Denoting by C_S the clustering corresponding to superarm S , Eq. (7) can alternatively (yet equivalently) be written as:

$$\bar{d}_\mu(C_S) = \sum_{\substack{(u,v) \in E, \\ C_S(u) = C_S(v)}} \mu_{uv}^- + \sum_{\substack{(u,v) \in E, \\ C_S(u) \neq C_S(v)}} \mu_{uv}^+, \quad \bar{a}_\mu(C_S) = \sum_{\substack{(u,v) \in E, \\ C_S(u) = C_S(v)}} \mu_{uv}^+ + \sum_{\substack{(u,v) \in E, \\ C_S(u) \neq C_S(v)}} \mu_{uv}^-, \tag{8}$$

Table 1 summarizes the elements of our CMAB correlation-clustering formulation.

CMAB-MIN-CC and CMAB-MAX-CC problems. Given a graph $G = (V, E)$, we perform discrete rounds $t = 1, \dots, T$, where at each round t , a clustering C_t of the vertices in V is computed and used to update the mean estimates $\hat{\mu}^+, \hat{\mu}^-$ of the random variables modeling the positive-type and negative-type edge weights, respectively. As discussed above, in MIN-CC, the weight of an edge e between vertices within the same cluster (resp. in different clusters) is interpreted as a random sample useful to update $\hat{\mu}_e^-$ (resp. $\hat{\mu}_e^+$). In MAX-CC, the opposite holds. The ultimate

Table 1 Contextualization of CMAB to correlation clustering

		MIN-CC	MAX-CC
Base arms		Edge replicas $\mathcal{S} = \mathcal{S}^{in} \cup \mathcal{S}^{out}$	
Superarm		$\mathcal{S} \subseteq \mathcal{S}$ satisfying Definition 1	
Base-arm	Intra-cluster base arm e^{in}	$\sim W_e^-$	$\sim W_e^+$
Outcome	Inter-cluster base arm e^{out}	$\sim W_e^+$	$\sim W_e^-$
Reward/loss		Loss in Eq. (5)	Reward in Eq. (6)

objective is to minimize/maximize the *cumulative expected loss/reward of the clusterings yielded in all the rounds*. Formally, the problems we tackle in this work are:

Problem 3 (CMAB-MIN-CC) Given a graph $G = (V, E)$ and a number $T > 0$ of rounds, for every $t = 1, \dots, T$ find a clustering $\mathcal{C}_t : V \rightarrow \mathbb{N}^+$ so as to minimize

$$\mathbb{E} \left[\sum_{t=1}^T \bar{d}_\mu(\mathcal{C}_t) \right]. \tag{9}$$

Problem 4 (CMAB-MAX-CC) Given a graph $G = (V, E)$ and a number $T > 0$ of rounds, for every $t = 1, \dots, T$ find a clustering $\mathcal{C}_t : V \rightarrow \mathbb{N}^+$ so as to maximize

$$\mathbb{E} \left[\sum_{t=1}^T \bar{a}_\mu(\mathcal{C}_t) \right]. \tag{10}$$

The expectation in Eqs. (9) and (10) is taken among all the random events generating the \mathcal{C}_t clusterings (due to, e.g., possible randomization in the oracle that computes the clusterings). There is a further expectation in those equations, which is implicit in the definition of expected loss $\bar{d}_\mu(\cdot)$ and expected reward $\bar{a}_\mu(\cdot)$ (see Eq. 8).

As previously discussed, CMAB-MAX-CC (resp. CMAB-MIN-CC) requires an oracle to solve, for each round, a MAX-CC (resp. MIN-CC) instance according to the mean estimates $\hat{\mu}^+, \hat{\mu}^-$. However, oracles available for MAX-CC (Charikar et al. 2005; Swamy 2004) are both inefficient and, more importantly, poorly useful in practice, since they are not able to output more than a fixed number of clusters (i.e., six). This implies that the corresponding CMAB setting (i.e., CMAB-MAX-CC) will inherit this issue too, since the clusterings yielded at each round are obtained through these algorithms. This aspect is a showstopper in our context, as we are interested in algorithms that are effective and theoretically solid, yet capable of providing outputs whose quality is recognizable in practice too, not only theoretically. For this reason, we hereinafter focus our attention on algorithms for CMAB-MIN-CC only. For completeness, algorithms for CMAB-MAX-CC are however presented in Appendix A.2.

4 Algorithms for CMAB-MIN-CC

In this section, we present algorithms for CMAB-MIN-CC (Problem 3). We first focus on the context of *general oracles* for MIN-CC (Sect. 4.1), and, then, on the case where the employed MIN-CC oracles achieve theoretical guarantees only if the input meets certain properties (Sect. 4.2). Finally, we discuss the special case of input edge-weight distributions satisfying specific constraints (Sect. 4.3).

4.1 General MIN-CC oracles

The CC-CLCB algorithm. We devise a variant of the so-called *Combinatorial Upper Confidence Bound* (CUCB) algorithm (Chen et al. 2016) which is an extension of the UCB1 method for MAB (Auer et al. 2002). It keeps, along with the estimate of the means of the base-arm random variables, confidence intervals within which the true means fall with overwhelming probability, and plays superarms based on the upper bound of those intervals. Our proposed variant, termed *Combinatorial Lower Confidence Bound* (CLCB), is tailored for minimization problems but follows the principles of CUCB: it maintains confidence intervals where the true means fall in with high probability, but, conversely to CUCB, it plays superarms based on the confidence-interval *lower bounds*.

Our customization of CLCB to MIN-CC is termed **CC-CLCB** and outlined as Algorithm 1. CC-CLCB keeps track of the mean estimates $\hat{\mu} = \{\hat{\mu}^+, \hat{\mu}^-\}$ (Eq. 4), and of the number T_e^+ (resp. T_e^-) of times a sample from W_e^+ (resp. W_e^-) random variable has been observed until the current round, for all $e \in E$. At the beginning, $\forall e \in E : T_e^+ = T_e^- = 0$, and $\hat{\mu}$ are initialized, e.g., randomly or based on prior domain knowledge (Line 1). In every round t , the current mean estimates are adjusted with a term ρ_e^\pm (defined based on Chernoff-Hoeffding bounds (Auer et al. 2002; Chen et al. 2016)), so as to foster, to some extent, the exploration of less often played base arms (Line 3). This leads to the adjusted means $\{\tilde{\mu}_e^+, \tilde{\mu}_e^-\}_{e \in E}$ (Line 4), which are interpreted as positive-type and negative-type edge weights of a correlation-clustering instance, respectively, and are fed as input (along with G) to an oracle \mathbf{O} that computes a MIN-CC solution \mathcal{C}_t (Line 5). \mathcal{C}_t is used as a feedback to update the mean estimates (Sect. 3, Table 1). Specifically, the weight of each intra-cluster (resp. inter-cluster) edge e is interpreted as a sample of W_e^- (resp. W_e^+), and is used to update $\hat{\mu}_e^-$, T_e^- (resp. $\hat{\mu}_e^+$, T_e^+). $\hat{\mu}_e^+$ and $\hat{\mu}_e^-$ are updated so as to be equal to the average of the samples from W_e^+ and W_e^- observed so far, respectively (Lines 6–11).

Algorithm 1 CC-CLCB

Input: A graph $G = (V, E)$; an integer $T > 0$; an oracle \mathbf{O} for MIN-CC
Output: A clustering C_t of G , for all $t = 1, \dots, T$

- 1: initialize $\hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}$ and $\hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}$; $\forall e \in E: T_e^+ \leftarrow 0, T_e^- \leftarrow 0$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\forall e \in E: \rho_e^+ \leftarrow \sqrt{\frac{3 \ln t}{2T_e^+}}, \rho_e^- \leftarrow \sqrt{\frac{3 \ln t}{2T_e^-}}$ ($\rho_e^+ = 0$, if $T_e^+ = 0$; $\rho_e^- = 0$, if $T_e^- = 0$)
- 4: $\forall e \in E: \tilde{\mu}_e^+ \leftarrow \max\{\hat{\mu}_e^+ - \rho_e^+, 0\}, \tilde{\mu}_e^- \leftarrow \max\{\hat{\mu}_e^- - \rho_e^-, 0\}$
- 5: $C_t \leftarrow$ run \mathbf{O} on input $\langle G, \{\tilde{\mu}_e^+\}_{e \in E}, \{\tilde{\mu}_e^-\}_{e \in E} \rangle$
- 6: **for** $e = (u, v) \in E \mid C_t(u) = C_t(v)$ **do**
- 7: observe feedback $w^- \sim W_e^-$; $\hat{\mu}_e^- \leftarrow (\hat{\mu}_e^- T_e^- + w^-)/(T_e^- + 1)$; $T_e^- \leftarrow T_e^- + 1$;
- 8: **end for**
- 9: **for** $e = (u, v) \in E \mid C_t(u) \neq C_t(v)$ **do**
- 10: observe feedback $w^+ \sim W_e^+$; $\hat{\mu}_e^+ \leftarrow (\hat{\mu}_e^+ T_e^+ + w^+)/(T_e^+ + 1)$; $T_e^+ \leftarrow T_e^+ + 1$;
- 11: **end for**
- 12: **end for**

Regret analysis of CC-CLCB. As correlation clustering is NP-hard, it is unlikely that CC-CLCB can be equipped with an *exact* oracle \mathbf{O} for MIN-CC running in polynomial time. Hence, in analyzing the theoretical guarantees of CC-CLCB, we consider the case where \mathbf{O} is a MIN-CC(α, β)-approximation oracle:

Definition 2 (MIN-CC-(α, β)-approximation oracle) Given a MIN-CC instance $I = \langle (V, E), \{(w_e^+, w_e^-)\}_{e \in E} \rangle$, let C_I^* be the optimal solution to I . Given $\alpha, \beta \in (0, 1]$, an algorithm for MIN-CC is a *min*-(α, β)-approximation oracle if, for every input I , it yields a solution \mathcal{C} such that $\Pr[f_{\min}(\mathcal{C}) \leq \frac{1}{\alpha} f_{\min}(C_I^*)] \geq \beta$ (where $f_{\min}(\cdot)$ is MIN-CC’s objective function, Eq. (1)).

The condition in Definition 2 to recognize \mathbf{O} as a MIN-CC-(α, β)-approximation oracle needs to hold on every MIN-CC instance that is given as input to \mathbf{O} at each round. Hence, the condition has to hold on the *mean estimates*, not the true ones. Similarly to the maximization counterpart, existing MIN-CC algorithms achieving $\mathcal{O}(\log |V|)$ guarantees *in expectation* (Charikar et al. 2005; Demaine et al. 2006) can be employed as MIN-CC-(α, β)-approximation oracles. More details in Appendix A.1.

We introduce a notion of (α, β)-approximation regret, which can be viewed as the minimization counterpart of the traditional one defined in Chen et al. (2016) and used in maximization problems. Applied to the MIN-CC context, this measure is defined as follows:

Definition 3 (MIN-CC-(α, β)-approximation regret) Let C_I^* be the clustering minimizing the expected loss $\bar{d}_\mu(\cdot)$ (Eq. 7) on a CMAB-MIN-CC instance I (w.r.t. the true μ means, Eq. 3), let $\mathcal{M} = \max_{\mathcal{C} \in \mathcal{C}(I)} \bar{d}_\mu(\mathcal{C})$ (where $\mathcal{C}(I)$ is the set of all clusterings of I), and let $\{C_t\}_{t=1}^T$ be the clusterings output by an algorithm \mathbf{A} run on I . The MIN-CC-(α, β)-approximation regret of \mathbf{A} is

$$Reg_{\mu,\alpha,\beta}^A(T) = \mathbb{E} \left[\sum_{t=1}^T \bar{d}_\mu(C_t) \right] - T \left[\frac{1}{\alpha} \bar{d}_\mu(C_t^*) + (\mathcal{M} - \frac{1}{\alpha} \bar{d}_\mu(C_t^*)) (1 - \beta) \right]. \tag{11}$$

The rationale of the above definition is as follows. First, being the focus on a minimization problem, the lower the probability β of success, the higher the loss value to compare with. Moreover, to take into account possible divergences of the approximation oracle from the optimum, and recalling that here are *losses*, not rewards, we add an extra term to the $\frac{1}{\alpha} \bar{d}_\mu(C_t^*)$ loss that “interpolates” between the highest $\beta = 1$ probability (thus, we compare with $\frac{1}{\alpha} \bar{d}_\mu(C_t^*)$) and the worst $\beta = 0$ probability (thus, we compare with the maximum value \mathcal{M} of loss). Note that the $T \left[\frac{1}{\alpha} \bar{d}_\mu(C_t^*) + (\mathcal{M} - \frac{1}{\alpha} \bar{d}_\mu(C_t^*)) (1 - \beta) \right]$ term in $Reg_{\mu,\alpha,\beta}^A(T)$ is used as a comparison for the (expected) performance $\mathbb{E} \left[\sum_{t=1}^T \bar{d}_\mu(C_t) \right]$ achieved by the CMAB method at hand in the various rounds. It is defined by noticing that, in every round $t = 1, \dots, T$, a MIN-CC- (α, β) -approximation oracle yields, with probability (at least) β , a solution whose $\bar{d}_\mu(\cdot)$ value is at most $\frac{1}{\alpha}$ times the optimum (i.e., $\frac{1}{\alpha} \bar{d}_\mu(C_t^*)$), and, with probability (at most) $1 - \beta$, a solution whose $\bar{d}_\mu(\cdot)$ value is more than $\frac{1}{\alpha} \bar{d}_\mu(C_t^*)$. In the latter case, consistently with the regret definition in maximization problems (Chen et al. 2016), we assume that the $\bar{d}_\mu(\cdot)$ value of the yielded solutions is equal to an upper bound $UB = \mathcal{M}$ on \bar{d}_μ . More precisely:

$$\begin{aligned} T \left[\frac{1}{\alpha} \beta \bar{d}_\mu(C_t^*) + (1 - \beta) UB \right] &= T \left[\frac{1}{\alpha} \beta \bar{d}_\mu(C_t^*) + (1 - \beta) \mathcal{M} \right] = T \left[\frac{1}{\alpha} \beta \bar{d}_\mu(C_t^*) + \mathcal{M} - \beta \mathcal{M} \right. \\ &\left. + \frac{1}{\alpha} \bar{d}_\mu(C_t^*) - \frac{1}{\alpha} \bar{d}_\mu(C_t^*) \right] = T \left[\frac{1}{\alpha} \bar{d}_\mu(C_t^*) + (\mathcal{M} - \frac{1}{\alpha} \bar{d}_\mu(C_t^*)) (1 - \beta) \right]. \end{aligned}$$

The comparison term in $Reg_{\mu,\alpha,\beta}^A(T)$ is pessimistic in assuming that when the (α, β) -approximation oracle does not achieve approximation guarantees it yields solutions whose $\bar{d}_\mu(\cdot)$ value is equal to the upper bound \mathcal{M} . However, note that this happens with probability $1 - \beta$. In our context, $1 - \beta$ is in the order of $|V|^{-c}$ (cf. Appendix A.1), with c set to 1 in our experiments (cf. Sect. 5). This means that the pessimistic assumption arises just in a tiny minority $|V|^{-1} T$ of the rounds. Also, the comparison term still adopts the optimistic assumption that the true μ weights are known, while they are actually not for the CMAB method that is being evaluated in terms of $Reg_{\mu,\alpha,\beta}^A(T)$.

As typically required in (C)MAB, the above regret is consistent with the definition of cumulative expected reward/loss at hand (i.e., Eq. 9, in this case). Thus, minimizing that regret corresponds to solving CMAB-MIN-CC (Problem 3). A key theoretical desideratum in CMAB (and online-learning settings in general), is having a regret bounded by some function that is *sublinear in the number of rounds*. This is motivated by the fact that the overall objective is typically a summation over the number of rounds, thus a regret growing (at least) linearly in the number of rounds is considered as a straightforward result that any algorithm can easily achieve.

As shown in the next theorem, CC-CLCB achieves a regret bound that is *logarithmic* in the number of rounds:

Theorem 1 Given $\alpha, \beta \in (0, 1]$, the $\text{MIN-CC}(\alpha, \beta)$ -approximation regret (Definition 3) of the CC-CLCB algorithm (Alg. 1), when equipped with a $\text{MIN-CC}(\alpha, \beta)$ -approximation oracle \mathbf{O} (Definition 2), is upper-bounded by a function that is $\mathcal{O}(\log T)$.

Proof (sketch) The proof relies on the following main result: the $\bar{d}_\mu(\cdot)$ function (Eq. 8) satisfies the properties of monotonicity and 1-norm bounded smoothness. This triggers a (rather long yet complex) chain of further results along the line of those derived in Wang and Chen (2017) for the regret analysis of algorithms for CMAB maximization problems. The ultimate of such results attests the desired logarithmic regret bound. A more detailed proof is reported in Appendix A.3. \square

4.2 MIN-CC oracles requiring the probability constraint

The CC-CLCB algorithm makes no assumptions on the input graph or edge-weight distributions. Thus, to achieve regret guarantees, CC-CLCB needs a MIN-CC oracle whose approximation guarantees hold in general, without requiring restrictions on the input. As said above, algorithms of this kind, in the context of MIN-CC , exist (Charikar et al. 2005; Demaine et al. 2006), but they suffer from issues such as limited efficiency yet not easy implementation (they need to solve a linear program of size $\Omega(|V|^3)$), and non-constant ($\mathcal{O}(\log |V|)$) approximation factor. A much better option would be to resort to the well-established Pivot (Ailon et al. 2008), which is efficient (it takes linear time), easy to implement (it just randomly picks a vertex u and builds a cluster as composed of u and all vertices connected to u with an edge whose positive-type weight is no less than the negative-type one), and achieves constant-factor approximation. Unfortunately, the (expected factor-5) guarantees of Pivot hold only if the input graph is complete and the edge weights satisfy the probability constraint, i.e., $w_{uv}^+ + w_{uv}^- = 1$, $\forall u, v \in V$. For this purpose, here we focus on the design of heuristic variants of CC-CLCB that favor the fulfilment of the probability constraint on the MIN-CC instances to be processed by the oracle. The rationale is that the closer a MIN-CC instance is to meet the probability constraint, the closer Pivot is to its “theoretical comfort zone”, thus expected to perform better.

The PC+Exp-CLCB algorithm. Our first proposal in this regard is the PC+Exp-CLCB algorithm (where “PC+Exp” means “probability constraint + exploration”). This algorithm, outlined as Algorithm 2, follows the same scheme as CC-CLCB , but it computes $\{\tilde{\mu}_{uv}^+, \tilde{\mu}_{uv}^-\}_{u,v \in V}$ adjusted means so as to simultaneously favor some exploration and make the resulting MIN-CC instance satisfy the probability constraint.

Algorithm 2 PC+Exp-CLCB

Input: A graph $G = (V, E)$; an integer $T > 0$; an oracle \mathbf{O} for MIN-CC
Output: A clustering \mathcal{C}_t of G , for all $t = 1, \dots, T$

- 1: initialize $\hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}$ and $\hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}$; $\forall e \in E: T_e^+ \leftarrow 0, T_e^- \leftarrow 0$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\forall e \in E: \Psi_e \leftarrow 1 - (\hat{\mu}_e^+ + \hat{\mu}_e^-)$
- 4: $\forall e \in E: \gamma_e^+ \leftarrow T_e^+ / (T_e^- + T_e^+), \gamma_e^- \leftarrow T_e^- / (T_e^- + T_e^+)$ ($\gamma_e^+ = \gamma_e^- = 0.5$, if $T_e^+ = T_e^- = 0$)
- 5: $\forall u, v \in V \mid (u, v) \notin E: \tilde{\mu}_{uv}^+ \leftarrow 0, \tilde{\mu}_{uv}^- \leftarrow 1$
- 6: **for** $e = (u, v) \in E \mid \Psi_e > 0$ **do**
- 7: $\tilde{\mu}_e^+ \leftarrow \hat{\mu}_e^+ + \Psi_e \gamma_e^+, \tilde{\mu}_e^- \leftarrow \hat{\mu}_e^- + \Psi_e \gamma_e^-$ $\{\tilde{\mu}_e^+ + \tilde{\mu}_e^- = 1\}$
- 8: **end for**
- 9: **for** $e = (u, v) \in E \mid \Psi_e \leq 0$ **do**
- 10: $\tilde{\mu}_e^+ \leftarrow \hat{\mu}_e^+ + \Psi_e(1 - \gamma_e^+), \tilde{\mu}_e^- \leftarrow \hat{\mu}_e^- + \Psi_e(1 - \gamma_e^-)$ $\{\tilde{\mu}_e^+ + \tilde{\mu}_e^- = 1\}$
- 11: **end for**
- 12: $\mathcal{C}_t \leftarrow$ run \mathbf{O} on input $\langle G, \{\tilde{\mu}_{uv}^+\}_{u,v \in V}, \{\tilde{\mu}_{uv}^-\}_{u,v \in V} \rangle$
- 13: observe feedback and update $\hat{\mu}$ as in Algorithm 1, Lines 6–11
- 14: **end for**

The Global-CLCB algorithm. As our second variant of CC-CLCB, we devise an algorithm—dubbed **Global-CLCB**—which builds MIN-CC instances at each round that are as close as possible to meet a *global constraint* on the edge weights similar the one defined in Mandaglio et al. (2021). The fulfilment of this global constraint makes the probability-constraint-aware approximation guarantees still hold even if the probability constraint is locally violated. **Global-CLCB** mainly relies on the following result:

Theorem 2 *Let $I = \langle G = (V, E), \{\tilde{\mu}_{uv}^+\}_{u,v \in V}, \{\tilde{\mu}_{uv}^-\}_{u,v \in V} \rangle$ be a MIN-CC instance. If $\left(\frac{|V|}{2}\right)^{-1} \sum_{u,v \in V} (\tilde{\mu}_{uv}^+ + \tilde{\mu}_{uv}^-) \geq 1$, then any MIN-CC algorithm (e.g., Pivot) achieving (expected) factor- ε approximation in presence of the probability constraint achieves (expected) factor- ε approximation on I too.*

Proof (sketch) The result here is a special case of the one originally proved in Theorem 1 in Mandaglio et al. (2021), specifically arising for $\Delta_{max} = 1$. Therefore, the proof herein is exactly the same as the one of Theorem 1 in Mandaglio et al. (2021), with the only straightforward exception of replacing Δ_{max} with the constant 1. \square

Global-CLCB attempts to compute $\{\tilde{\mu}_{uv}^+, \tilde{\mu}_{uv}^-\}_{u,v \in V}$ adjusted means that are as close as possible to satisfy Theorem 2. **Global-CLCB** is the same as **CC-CLCB**, except for their respective Line 4. A detailed pseudocode of **Global-CLCB** is reported in Algorithm 3. We point out that **CC-CLCB**’s regret analysis does not hold for **PC+Exp-CLCB** or **Global-CLCB**. Deriving theoretical regret guarantees for these (or similar) heuristics is a challenging open question that we defer to future work.

Algorithm 3 Global-CLCB

Input: A graph $G = (V, E)$; an integer $T > 0$; an oracle \mathbf{O} for MIN-CC
Output: A clustering \mathcal{C}_t of G , for all $t = 1, \dots, T$

- 1: initialize $\hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}$ and $\hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}$; $\forall e \in E: T_e^+ \leftarrow 0, T_e^- \leftarrow 0$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\forall e \in E: \rho_e^+ \leftarrow \sqrt{\frac{3 \ln t}{2T_e^+}}, \rho_e^- \leftarrow \sqrt{\frac{3 \ln t}{2T_e^-}}$ ($\rho_e^+ = 0$, if $T_e^+ = 0$; $\rho_e^- = 0$, if $T_e^- = 0$)
- 4: $\forall e \in E: \tilde{\mu}_e^+ \leftarrow \min\{1 - (\hat{\mu}_e^- - \rho_e^-), 1\}, \tilde{\mu}_e^- \leftarrow \min\{1 - (\hat{\mu}_e^+ - \rho_e^+), 1\}$
- 5: $\mathcal{C}_t \leftarrow$ run \mathbf{O} on input $\langle G, \{\tilde{\mu}_e^+\}_{e \in E}, \{\tilde{\mu}_e^-\}_{e \in E} \rangle$
- 6: **for** $e = (u, v) \in E \mid \mathcal{C}_t(u) = \mathcal{C}_t(v)$ **do**
- 7: observe feedback $w^- \sim W_e^-$; $\hat{\mu}_e^- \leftarrow (\hat{\mu}_e^- T_e^- + w^-)/(T_e^- + 1)$; $T_e^- \leftarrow T_e^- + 1$;
- 8: **end for**
- 9: **for** $e = (u, v) \in E \mid \mathcal{C}_t(u) \neq \mathcal{C}_t(v)$ **do**
- 10: observe feedback $w^+ \sim W_e^+$; $\hat{\mu}_e^+ \leftarrow (\hat{\mu}_e^+ T_e^+ + w^+)/(T_e^+ + 1)$; $T_e^+ \leftarrow T_e^+ + 1$;
- 11: **end for**
- 12: **end for**

4.3 Special edge-weight distributions

An interesting special input is the one of *symmetric* edge-weight distributions:

Definition 4 (*Symmetric distributions*) $[0, 1]$ -support random variables W_e^+, W_e^- have symmetric distributions if and only if $W_e^+(x) = W_e^-(1 - x)$, for all $x \in [0, 1]$.

Conceptually, this is like assuming that if a similarity equal to x holds for any two vertices, a $(1 - x)$ distance implicitly holds for the same vertices as well.

CMAB-correlation-clustering instances where symmetry holds for *all* edge-weight distributions are easier to solve. In fact, symmetry in the distributions makes the instance at hand a *full-information bandit setting*: observing a sample $x \sim W_e^+$ is equivalent to observing a sample $(1 - x) \sim W_e^-$, for all $e \in E$. This corresponds to having an outcome revealed for *all the base arms, regardless of the superarm (clustering) played*. In this case, therefore, exploration is meaningless. Rather, a *full-exploitation* strategy is worth to be performed, where a clustering considering solely the current mean estimates is yielded in each round. This strategy achieves a regret bound that is constant in the number of rounds, as stated by Theorem 3:

Theorem 3 Given $\alpha, \beta \in (0, 1]$, the MIN-CC- (α, β) -approximation regret (Definition 3) of a full-exploitation strategy run on a CMAB-MIN-CC instance where all edge-weight distributions are symmetric, and equipped with a MIN-CC- (α, β) -approximation oracle (Definition 2), is upper-bounded by a function of T that is $\mathcal{O}(1)$.

Proof (sketch) The full-information bandit setting allows for simplifying some intermediate math in the regret analysis of a non-full-information setting (Theorem 1).

Fig. 1 Example CMAB-MIN-CC solutions obtained by ϵ -greedy on HighlandTribes, and by CC-CLCB on Contiguous-USA, over $T = 200$ rounds, using the linear-programming method in Charikar et al. (2005) as an oracle. Vertex colors correspond to cluster memberships, while edges are colored as black, resp. red, if there is an edge-level agreement, resp. disagreement, between the true edge weights and their estimated values; by edge-level agreement, we mean that the disequality between the positive-type weight and negative-type weight of an edge holds the same on the true weights and their estimates. Values within parentheses refer to NMI (with arithmetic mean normalization) between the CMAB-MIN-CC solution by the oracle over the graph with mean estimates and the corresponding solution over the graph with true means

These simplifications ultimately lead to a $\mathcal{O}(1)$ regret bound. A detailed proof and a pseudocode of the full-exploitation strategy are in Appendix A.4. \square

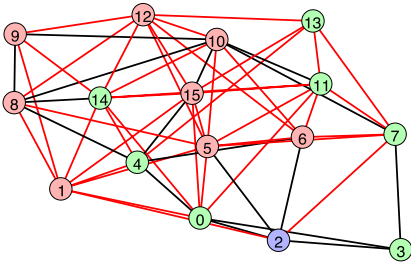
4.4 Visualization example

Figure 1 provides a visualization of the HighlandTribes and Contiguous-USA graphs, used as case in point (cf. Section 5), and their CMAB-MIN-CC clusterings produced by ϵ -greedy and CC-CLCB, respectively, using the same oracle in both cases. In particular, we show the outcomes referring to the using correlation-clustering linear-programming method in Charikar et al. (2005) as an oracle. Our goal here is to provide empirical evidence of the significance of the CMAB-MIN-CC setting and effectiveness of the CMAB-MIN-CC methods. To this purpose, we show three snapshots of execution on each graph, namely at the initial, middle and final round of a method. Besides visualizing the cluster memberships of vertices—note that vertices of one cluster share the same color at any round, but color memberships may change at different rounds—we also use black edges and red edges to distinguish between *edge-level agreements* and *disagreements*, respectively, which denote that the disequality between the positive-type weight and negative-type weight of an edge holds, or does not hold, the same on the true weights and their estimates; formally, an edge (u, v) is colored as black if $(\mu_{uv}^+ > \mu_{uv}^- \wedge \hat{\mu}_{uv}^+ > \hat{\mu}_{uv}^-) \vee (\mu_{uv}^+ \leq \mu_{uv}^- \wedge \hat{\mu}_{uv}^+ \leq \hat{\mu}_{uv}^-)$, otherwise as red.

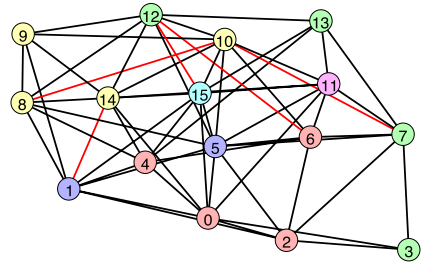
Two major remarks stand out by looking at the plots for each graph. First, the similarity measured in terms of normalized mutual information (NMI) between the CMAB-MIN-CC solution produced by the oracle over the graph with mean estimates and the corresponding solution over the graph with true means, significantly improves as more rounds are carried out; in particular, as shown for HighlandTribes (plots (a-c)), already after few early rounds, NMI approaches the maximum value reached at the final round. Second, the number of edge-level agreements also rapidly increases after few rounds, until few disagreements are left at the final round.

5 Experimental methodology

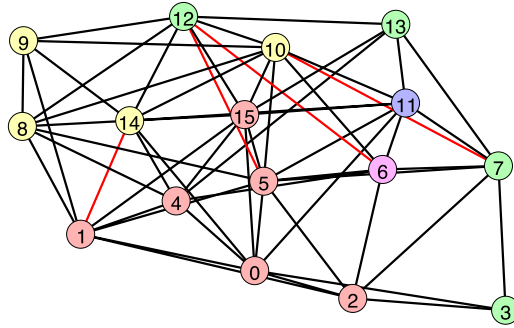
Data. We consider ten publicly-available real-world graphs, as summarized in Table 2. Each of the five networks from bottom corresponds to the flattening of a network originally represented as a set of snapshot-graphs (Galimberti et al. 2020)



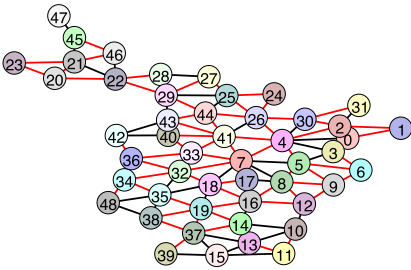
(a) HighlandTribes, initial round (0.193)



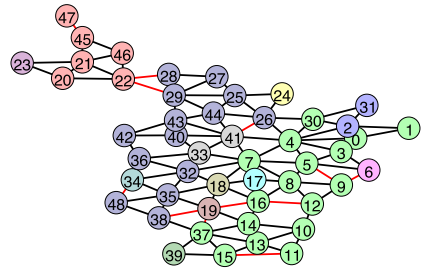
(b) HighlandTribes, middle round (0.830)



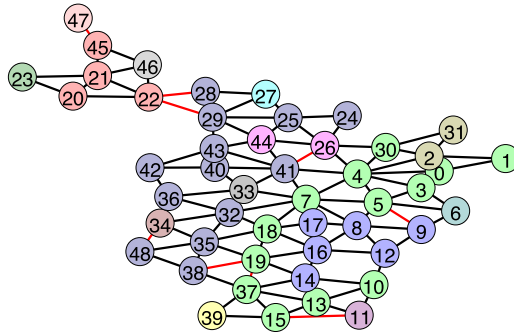
(c) HighlandTribes, final round (1.0)



(d) Contiguous-USA, initial round (0.719)



(e) Contiguous-USA, middle round (0.741)



(f) Contiguous-USA, final round (0.838)

(i.e., an edge between u and v exists in the flattened network if u and v were linked in at least one snapshot).

Edge weight distributions. The random variables W_e^+, W_e^- modeling the positive-type and negative-type edge weights in a MIN-CC instance are assumed to follow a *Bernoulli* distribution, whose means are generated according to three schemes.

In the first two schemes, termed *R-wd* and *PC-wd*, the original—possibly *incomplete*—network topology of the underlying input graph is maintained, meaning that $\mu_e^+ = \mathbb{E}[W_e^+] = \mu_e^- = \mathbb{E}[W_e^-] = 0$, for all $e \notin E$. As far as the means μ_e^+, μ_e^- for each $e \in E$, *R-wd* samples uniformly at random both μ_e^+ and μ_e^- from the $[0, 1]$ interval, independently from one another, i.e., $\mu_e^+, \mu_e^- \sim \text{Uniform}(0, 1)$, for all $e \in E$. On the other hand, *PC-wd* ensures that the probability constraint holds on the generated means, which corresponds to first sample $\mu_e^+ \sim \text{Uniform}(0, 1)$, and then set $\mu_e^- = 1 - \mu_e^+$, for all $e \in E$. As a result, for both *R-wd* and *PC-wd*, $\mu_e^+, \mu_e^- \in [0, 1]$, while the samples observed from the edge weight distributions at each CMAB round $\in \{0, 1\}$, for all $e \in E$. In particular, samples $W_e^+ = 1$ and $W_e^+ = 0$ (resp. $W_e^- = 1$ and $W_e^- = 0$) are observed with probability μ_e^+ and $1 - \mu_e^+$ (resp. μ_e^- and $1 - \mu_e^-$), respectively.

The third scheme assumes the actual network topology imposes a binary, mutually exclusive setting for *each pair of vertices*, i.e., $\mu_{uv}^+ = 1, \mu_{uv}^- = 0$, if $(u, v) \in E$, and $\mu_{uv}^+ = 0, \mu_{uv}^- = 1$, if $(u, v) \notin E$. Since this setting leads to a new *complete* graph, the scheme is referred to as *C-wd*, and it will be considered only for the smaller datasets, as it is computationally unfeasible to handle complete versions of the larger datasets. As $\mu_{uv}^+, \mu_{uv}^- \in \{0, 1\}$, for all $u, v \in V$, the underlying W_{uv}^+, W_{uv}^- distributions are actually degenerate, and every sample observed in a CMAB round from W_{uv}^+ (resp., W_{uv}^-) will be equal to 1 if $\mu_{uv}^+ = 1$ (resp., $\mu_{uv}^- = 1$), and it will be 0 if $\mu_{uv}^+ = 0$ (resp., $\mu_{uv}^- = 0$).

Assessment criteria. The means μ_e^+, μ_e^- generated via the above schemes correspond to the *true* correlation-clustering edge weights that are unknown to any CMAB method. These are used to evaluate the quality of the clusterings yielded in the various CMAB rounds via the *average expected normalized cumulative* MIN-CC *loss*, calculated until each round t :

$$f^{(t)} = \frac{1}{t} \sum_{i=1}^t \mathbb{E} \left[\frac{\bar{d}_{\mu}(C_i)}{U} \right], \tag{12}$$

where C_i is the clustering of the i -th round, $U = \sum_{u,v \in V} \max\{\mu_{uv}^+, \mu_{uv}^-\}$ is a normalization constant (equal to an upper bound on the MIN-CC objective-function value, so that $\bar{d}(C_i)/U \in [0, 1]$), and the $\mathbb{E}[\cdot]$ expectation is computed by averaging the $\bar{d}_{\mu}(\cdot)$ values obtained over all the runs of execution of the randomized MIN-CC oracles (see below). Note that Eq. (12) is a shorter and normalized version of Eq. (11). In fact, Eq. (12) is limited to the first term only in Eq. (11), as the second term is common to all the methods (under the same oracle). It is also normalized, so as to have the results on graphs of different size more easily comparable to each other.

As a second assessment criterion, we consider the error of the $\hat{\mu}_{e,t}^+, \hat{\mu}_{e,t}^-$ weight estimates at each round t , which is measured in terms of relative error norm as

Table 2 Main characteristics of the real-world datasets used in our evaluation

	$ V $	$ E $	Density	Avg. degree	Avg. path Length	Clustering Coefficient
Karate*	34	78	0.139	4.588	2.408	0.256
Dolphins*	62	159	0.084	5.129	3.357	0.309
Zebra*	27	111	0.316	8.222	1.842	0.845
Highland-Tribes*	16	58	0.483	7.250	1.542	0.527
Contiguous-USA*	49	107	0.091	4.367	4.163	0.406
Last.fm	992	369,973	0.753	745.913	1.247	0.860
Primary school**	242	8,317	0.285	68.736	1.733	0.480
Prosper loans	89,269	3,330,022	8E-04	74.607	3.239	0.003
Wikipedia	343,860	10,519,921	2E-04	61.187	3.099	0.065
DBLP	1,824,701	8,344,615	5E-06	9.146	6.514	0.169

*Available from <http://konect.cc/networks/> **Available from <http://www.sociopatterns.org/datasets/>

$$ren^{(t)} = \sqrt{\frac{\sum_{e \in E} (\mu_e^+ - \hat{\mu}_{e,t}^+)^2 + \sum_{e \in E} (\mu_e^- - \hat{\mu}_{e,t}^-)^2}{\sum_{e \in E} (\mu_e^+)^2 + \sum_{e \in E} (\mu_e^-)^2}}, \tag{13}$$

For both $f^{(t)}$ and $ren^{(t)}$ lower values correspond to better performance. Our main focus is on the $f^{(T)}$, $ren^{(T)}$ values at the final round $t = T$, as they give a compact yet general evidence of the overall performance of a method. However, we also analyze the trend of $ren^{(t)}$ over the various rounds to assess statistical significance (Sect. 6.5), and report evidence of such trends (Sect. 7).

Within this view, in the result tables presented in Sect. 6, we shall report $f^{(T)}$ and $ren^{(T)}$ values. Moreover, for the CMAB methods only, we also provide the *growth rates*, i.e., the average amount of relative change between the initial and the final round over the span T (in percentage):

$$gr_f^{\%} = \left(\frac{\bar{d}_\mu(C_T)}{\bar{d}_\mu(C_1)} - 1 \right) \times 100, \quad gr_{ren}^{\%} = \left(\frac{ren^{(T)}}{ren^{(1)}} - 1 \right) \times 100. \tag{14}$$

Finally, we are also interested in assessing the running time of the various tested methods (Sect. 6.4).

Methods. We involve methods falling into four approaches: (i) CMAB-MIN-CC methods adopting the CLCB paradigm, (ii) classic general CMAB heuristics that, in this context, are customized to work for CMAB-MIN-CC, (iii) baselines that do not follow the CMAB paradigm, and (iv) a reference method that performs clustering by utilizing the true edge weights. More specifically:

- (i) As CLCB-based methods, we include CC-CLCB (Algorithm 1), PC+Exp-CLCB (Algorithm 2), and Global-CLCB (Algorithm 3). Moreover, for both CC-CLCB and Global-CLCB, we also consider their CC-CLCB-m

- and Global-CLCB-m variants, which are less biased towards exploration. Specifically, following Wang and Chen (2018), CC-CLCB-m and Global-CLCB-m utilize uncertain terms defined as $\rho_e^\pm = \sqrt{\ln t / 2T_e^\pm}$ (instead of $\rho_e^\pm = \sqrt{3 \ln t / 2T_e^\pm}$).
- (ii) As CMAB heuristics, we involve the well-established ϵ -greedy, pure exploitation (PE), and Combinatorial Thompson Sampling (CTS) (Wang and Chen 2018). As for ϵ -greedy, we consider both a fixed exploration rate, set to 0.1, and an adaptive exploration rate, set to be proportional to t^{-1} , at each t . These variants are dubbed EG-fixed and EG, respectively.
 - (iii) As far as non-CMAB baselines, the idea is to set both types of unknown edge weights based on topological affinity of any two vertices' neighborhoods, and then run a MIN-CC algorithm [specifically, Pivot (Ailon et al. 2008) in most experiments, and the linear programming approach dubbed LP+R (Charikar et al. 2005) in the experiment in Sect. 6.3] on such an input, employing no weight learning strategy. More precisely, we resort to two well-known topological similarity measures, namely Jaccard index and Adamic-Adar index, to set the positive-type weights: for each $(u, v) \in E$, $w_{uv}^+ = |N(u) \cap N(v)| / |N(u) \cup N(v)|$ using Jaccard, or $w_{uv}^+ = |N(u) \cap N(v)|^{-1} \sum_{z \in N(u) \cap N(v)} (\log |N(z)|)^{-1}$ using (normalized) Adamic-Adar, where $N(u)$ is the set of u 's neighbors. The negative-type weights are then derived in such a way that the probability constraint holds, i.e., $w_{uv}^- = 1 - w_{uv}^+$.
 - (iv) As a reference method, we consider clustering with the actual (i.e., true) edge weights via a state-of-the-art MIN-CC algorithm (i.e., Pivot (Ailon et al. 2008) in most experiments, and LP+R (Charikar et al. 2005) in the experiment in Sect. 6.3). This method is termed *Actual-weight*.

Unless otherwise specified, all the CMAB methods are assumed to be equipped with the Pivot algorithm (Ailon et al. 2008) as an oracle for MIN-CC. Pivot is used as a reference oracle because it is more usable in practice, due to its efficiency, approximation guarantees, and ease of implementation. However, we also carry out an experiment to evaluate the impact of a different oracle, specifically the LP+R algorithm (Ailon et al. 2008) (Sect. 6.3). As LP+R takes $\Omega(|V|^3)$ time just to build the linear program, this experiment is performed on the smaller datasets only.

Since the chosen MIN-CC oracles are randomized algorithms, for every experiment, we perform $\log_2 |V|$ independent runs of the selected oracle per CMAB round (setting $\delta = 1, c = 1$, cf. Appendix A.1), and take the best solution in terms of MIN-CC objective with respect to the current weight estimates. In all the experiments, the number T of CMAB rounds is set to 500, while the number of runs of the MIN-CC oracle for every round is set to 10.

Evaluation goals. As this is the first work that investigates MIN-CC in a CMAB setting, our experiments are not really intended to assess the superiority of some proposed method(s) over the state of the art. Rather, our main objective here is to provide a comparative evaluation of a variety of CMAB heuristics, approximation algorithms, and heuristic variants of approximation algorithms in the context of

CMAB-MIN-CC, and derive experimental insights on the peculiarities of the various tested methods. Specifically, the main goals of our experimental evaluation are as follows:

- Assess the performance of the CMAB methods (CC-CLCB, EG, EG-fixed, PE, CTS) in terms of $f^{(T)}$ and $ren^{(T)}$, and compare them to non-CMAB baselines (Adamic-Adar, Jaccard) and the reference *Actual-weight* method (Sect. 6.1).
- Compare the performance of the various CLCB variants (CC-CLCB, CC-CLCB-m, PC+Exp-CLCB, Global-CLCB, Global-CLCB-m) to each other, in terms of $f^{(T)}$ and $ren^{(T)}$ (Sect. 6.2).
- Evaluate the impact of varying the MIN-CC oracle on the performance of the various CMAB methods (Sect. 6.3).
- Evaluate the efficiency of all the selected methods (Sect. 6.4).
- Perform a statistical significance analysis of the reported results (Sect. 6.5).

Further characterization. We also analyze the number of output clusters and the stability of the performance over the various rounds and runs of the tested methods (Sect. 7). This analysis is intended not really as a performance assessment, rather as an additional useful insight to better characterize the tested methods.

Implementation and testing environment. All the tested methods are implemented in Python 3.8, with some of them using external libraries. In particular, LP+R adopts the PuLP library for linear programming,² Adamic-Adar and Jaccard use, respectively, the NetworkX and python-igraph libraries to compute the topological similarity scores.³ All the experiments are carried out on the *Cresco6* cluster,⁴ a high-performance computing system running Linux Centos 7.4, and consisting of 434 nodes, where each one is equipped with two Intel(R) Xeon(R) Platinum 8160 CPU @2.10GHz x24 processor and 192GB ram.

6 Results

6.1 Performance of the CMAB methods

Quality of the clusterings (Table 3). As a first general remark, the non-CMAB baselines (Adamic-Adar, Jaccard) achieve the worst performance in all the datasets and weight settings, while *Actual-weight* is always the best method, with only a couple of exceptions. This was expected, as the non-CMAB baselines employ no strategy to learn the true weights, whereas *Actual-weight* operates on the true weights. Importantly, in most cases, the CMAB methods (CC-CLCB, EG, EG-fixed, PE, CTS) perform comparably or close to *Actual-weight*. The loss values of all the CMAB methods follow a decreasing trend over the rounds, as testified by the

² <https://coin-or.github.io/pulp/>.

³ <https://networkx.org> <https://igraph.readthedocs.io>.

⁴ <https://www.eneagrid.enea.it/CRESCOportal/>.

negative growth rates (and better shown in Fig. 2, Sect. 7). This was expected, since the CMAB algorithms learn how to cluster the vertices over time. In general, all the CMAB algorithms converge to solutions with lower growth rate in the *PC-wd* setting than the *R-wd* setting. Also, with the exception of HighlandTribes, the difference of the best loss scores is higher in the *PC-wd* setting than the *R-wd* one. This complies with the fact that the probability constraint leads to an easier MIN-CC clustering task.

Focusing on the CMAB methods, the best performance corresponds to PE. This can be explained since (i) the MIN-CC oracle therein used (i.e., *Pivot*) is a randomized algorithm, thus, although with a pure-exploitation bandit strategy, it results in some implicit exploration; (ii) due to the peculiarity of our problem, each super arm admits a feedback from half the total number of arms, thus a bandit strategy with minimal exploration would likely perform better in the long run. CC-CLCB exhibits very good performance: it is comparable or close to the best methods in most datasets and weight settings, achieving maximum and average difference in loss with respect to the best performer(s) over all the configurations of 0.038 and 0.014, respectively.

Quality of the learned edge weights (Table 4). A first general observation is that the weight estimates of all the CMAB methods improve as the rounds progress, and the relative error goes down over time, leading to a negative growth rate. This is consistent with the clustering improvement by increasing rounds observed from Table 3. As expected, the non-CMAB baselines yield the highest error values, while *Actual-weight* clearly achieves zero error everywhere. Among the CMAB methods, EG and EG-fixed yield the most accurate estimates in the *C-wd* weight setting. In the *R-wd* and *PC-wd* settings, EG-fixed is (comparable to) the best performer on the smaller datasets (Karate, Dolphins, Zebra, HighlandTribes, Contiguous-USA), while on the bigger datasets, CTS is (comparable to) the best method. CC-CLCB achieves the best performance in three datasets (Zebra, Last.fm, PrimarySchool) for *R-wd* and *PC-wd* distributions. Importantly, for some methods, the good/bad performance on the weight-estimation task does not necessarily translate into an equally good/bad performance in the clustering results discussed above. This has a twofold motivation: (1) the underlying oracle is not an exact algorithm for MIN-CC, thus it may happen that clustering with weight estimates lead to clusterings that are of better quality when evaluated in terms of the actual weights, and (2) CMAB methods like CC-CLCB adopt exploration strategies that perturb the current weight estimates before giving them to the oracle, which corresponds to perform clustering with weights that are actually different from the estimated ones.

6.2 Performance of the CLCB-based CMAB methods

Quality of the clusterings (Table 5). In general, we observe that all the CC-CLCB variants perform rather closely to each other in all the configurations. Deepening the analysis, CC-CLCB-m and Global-CLCB-m are the best methods (in all the datasets but one) in the *R-wd* weight setting. Conversely, in the *PC-wd* and *C-wd*

Table 3 Performance in terms of $f^{(T)}$ (Eq. 12) and (for the CMAB methods) gr^{gr} (Eq. 14)

(a) <i>R-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.66 -11.97%	0.656 -10.3%	0.671 -15.07%	0.583 -12.34%	0.675 -12.96%	0.749 -0.13%	0.74 -1.04%	0.747 -0.42%	0.748 -0.34%	0.717 -4.45%
EG	0.647 -15.81%	0.633 -15.3%	0.665 -7.95%	0.563 -18.24%	0.66 -13.1%	0.748 -0.27%	0.734 -2.01%	0.743 -0.84%	0.747 -0.45%	0.705 -6.02%
EG-fixed	0.652 -13.07%	0.637 -12.8%	0.67 -15.19%	0.569 -14.51%	0.668 -13.86%	0.748 -0.24%	0.735 -1.71%	0.744 -0.76%	0.747 -0.43%	0.708 -5.6%
PE	0.646 -13.85%	0.634 -13.3%	0.663 -16.19%	0.559 -15.97%	0.668 -13.85%	0.748 -0.27%	0.733 -1.92%	0.743 -0.84%	0.747 -0.46%	0.704 -6.14%
CTS	0.647 -12.03%	0.635 -14.67%	0.663 -11.11%	0.561 -23.95%	0.665 -14.26%	0.748 -0.25%	0.734 -1.81%	0.744 -0.78%	0.747 -0.46%	0.707 -5.77%
Adamic-Adar	0.758	0.759	0.764	0.648	0.778	0.75	0.748	0.75	0.75	0.75
Jaccard	0.764	0.758	0.703	0.715	0.773	0.749	0.749	0.75	0.75	0.75
Actual-weight	0.64	0.628	0.651	0.55	0.663	0.748	0.734	0.742	0.746	0.703

(b) <i>PC-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.59 -18.22%	0.54 -19.09%	0.58 -21.67%	0.51 -20.08%	0.53 -24.36%	0.67 -0.19%	0.65 -1.94%	0.66 -0.79%	0.66 -0.56%	0.62 -7.28%
EG	0.58 -13.59%	0.52 -21.31%	0.57 -14.72%	0.5 -24.44%	0.51 -24.4%	0.66 -0.36%	0.64 -3.29%	0.66 -1.32%	0.66 -0.68%	0.61 -9%
EG-fixed	0.59 -18.71%	0.52 -21.77%	0.57 -22.17%	0.5 -20.27%	0.52 -25.8%	0.66 -0.31%	0.64 -2.68%	0.66 -1.2%	0.66 -0.65%	0.61 -8.44%
PE	0.58 -19.54%	0.51 -23.45%	0.56 -24.03%	0.49 -22.9%	0.5 -28.52%	0.66 -0.34%	0.64 -3.01%	0.66 -1.34%	0.66 -0.71%	0.6 -9.41%

Table 3 (continued)

(b) <i>PC-wd weight distributions</i>													
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP			
CTS	0.58 -15.34%	0.51 -21.17%	0.56 -16.59%	0.49 -24.13%	0.51 -24.09%	0.66 -0.34%	0.64 -3.19%	0.66 -1.26%	0.66 -0.7%	0.61 -8.91%			
Adamic-Adar	0.7	0.68	0.7	0.62	0.62	0.67	0.66	0.67	0.67	0.67			
Jaccard	0.74	0.69	0.59	0.6	0.67	0.67	0.66	0.67	0.67	0.67			
Actual-weight	0.58	0.5	0.54	0.49	0.49	0.66	0.64	0.66	0.66	0.6			
(c) <i>C-wd weight distributions (smaller datasets only)</i>													
Method	Karate	Dolphins			Zebra			HighlandTribes			Contiguous-USA		
CC-CLCB	0.12	-14.04%	0.072	-14.14%	0.091	-71.36%	0.338	-30.04%	0.064	-29.4%			
EG	0.129	-67.53%	0.091	-71.88%	0.081	-81.99%	0.348	-26.08%	0.084	-80.8%			
EG-fixed	0.147	5.41%	0.108	28.09%	0.107	-66.09%	0.347	-28.14%	0.102	11.69%			
PE	0.109	-21.65%	0.07	-17.17%	0.057	-81.83%	0.334	-30.91%	0.063	-31.2%			
CTS	0.111	-73.56%	0.073	-78.55%	0.061	-86.61%	0.334	-30.29%	0.066	-81.49%			
Adamic-Adar	0.136	-	0.109	-	0.301	-	0.462	-	0.108	-			
Jaccard	0.122	-	0.075	-	0.137	-	0.38	-	0.077	-			
Actual-weight	0.112	-	0.069	-	0.054	-	0.328	-	0.061	-			

All the CMAB methods are equipped with Pivot as a Min-CC oracle. Per-dataset best scores among the CMAB methods are reported in bold

Table 4 Performance in terms of $rep^{(T)}$ (Eq. 13) and (for the CMAB methods) gr^{sc} (Eq. 14)

(a) <i>R-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.175 -80%	0.186 -78.52%	0.056 -93.37%	0.199 -78.1%	0.146 -83.17%	0.053 -93.86%	0.066 -92.44%	0.412 -52.41%	0.217 -74.99%	0.158 -81.75%
EG	0.109 -87.08%	0.132 -84.46%	0.063 -92.79%	0.127 -85.77%	0.103 -88.09%	0.057 -93.43%	0.077 -91.06%	0.396 -54.33%	0.195 -77.43%	0.167 -80.77%
EG-fixed	0.091 -89.64%	0.107 -87.69%	0.06 -92.88%	0.092 -89.83%	0.09 -89.59%	0.056 -93.49%	0.076 -91.3%	0.36 -58.4%	0.177 -79.52%	0.14 -83.8%
PE	0.302 -65.56%	0.315 -63.54%	0.173 -79.45%	0.262 -71.18%	0.332 -61.82%	0.058 -93.29%	0.086 -90.09%	0.432 -50.09%	0.237 -72.67%	0.253 -70.81%
CTS	0.102 -76.65%	0.114 -75.77%	0.067 -85.48%	0.138 -70.75%	0.088 -80.25%	0.057 -87.43%	0.076 -83.37%	0.255 -44.16%	0.181 -60.38%	0.127 -72.15%
Adamic-Adar	0.649	0.671	0.617	0.531	0.592	0.787	0.684	0.937	0.812	0.668
Jaccard	0.644	0.637	0.77	0.577	0.517	0.715	0.612	0.988	0.897	0.814
Actual-weight	0	0	0	0	0	0	0	0	0	0

(b) <i>PC-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.19 -76.88%	0.26 -68.89%	0.05 -93.14%	0.25 -71.42%	0.14 -83.15%	0.05 -93.89%	0.07 -92.29%	0.52 -39.69%	0.22 -74.07%	0.2 -77.23%
EG	0.09 -89.08%	0.13 -84.59%	0.06 -92.44%	0.09 -89.48%	0.1 -87.57%	0.06 -93.45%	0.08 -91.15%	0.41 -53.22%	0.2 -77.43%	0.17 -80.84%
EG-fixed	0.08 -89.99%	0.11 -86.82%	0.06 -92.88%	0.08 -90.95%	0.09 -89.39%	0.06 -93.51%	0.07 -91.43%	0.37 -57.49%	0.18 -79.54%	0.14 -83.82%
PE	0.34 -58.88%	0.32 -62.81%	0.15 -81.57%	0.23 -73.39%	0.27 -66.94%	0.06 -93.31%	0.09 -90.09%	0.45 -48.2%	0.24 -72.24%	0.25 -71.13%

Table 4 (continued)

(b) <i>PC-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CTS	0.09	0.13	0.06	0.12	0.09	0.06	0.08	0.27	0.18	0.13
	-79.31%	-72.69%	-86.55%	-74.09%	-80.75%	-87.46%	-83.33%	-41.24%	-59.65%	-70.49%
Adamic-Adar	0.66	0.7	0.63	0.54	0.61	0.79	0.68	0.94	0.81	0.67
Jaccard	0.72	0.64	0.73	0.53	0.56	0.72	0.61	0.99	0.9	0.81
Actual-weight	0	0	0	0	0	0	0	0	0	0
(c) <i>C-wd weight distributions (smaller datasets only)</i>										
Method	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA					
						Contiguous-USA				
CC-CLCB	0.623	-32.82%	0.824	-13.9%	0.673	-18.66%	0.158	-78.01%	0.87	-8.76%
EG	0.0	-100%	0.0	-100%	0.0	-100%	0.0	-100%	0.0	-100%
EG-fixed	0.0	-100%	0.0	-100%	0.0	-100%	0.0	-100%	0.0	-100%
PE	0.662	-28.62%	0.824	-13.87%	0.681	-17.69%	0.158	-78.01%	0.87	-8.75%
CTS	0.271	-54.9%	0.266	-55.72%	0.265	-55.97%	0.061	-89.82%	0.29	-51.8%
Adamic-Adar	0.514	-	0.446	-	0.54	-	0.706	-	0.372	-
Jaccard	0.429	-	0.294	-	0.281	-	0.596	-	0.252	-
Actual-weight	0	-	0	-	0	-	0	-	0	-

All the CMAB methods are equipped with Pivot as a Min-CC oracle. Per-dataset best scores among the CMAB methods are reported in bold

settings, the best method is PC+Exp-CLCB in most cases: specifically, it is the best performer on all the datasets (though on par with CC-CLCB-m and Global-CLCB-m on the larger ones) in *PC-wd*, and on three out of five datasets in *C-wd*. Also, Global-CLCB performs better in the *PC-wd* and *C-wd* settings than the *R-wd* one. These findings comply with the design principles of the CC-CLCB variants that favor the fulfilment of the probability constraint on the MIN-CC instances to be processed by the underlying oracle (cf. Sect. 4.2), which clearly benefit from those settings like *PC-wd* and *C-wd* where the probability constraint actually holds.

Interestingly, CLCB and Global-CLCB achieve the same results in all the configurations (and the same holds for CLCB-m vs. Global-CLCB-m). This can be explained as CC-CLCB and Global-CLCB (and the same is for CLCB-m and Global-CLCB-m) compute adjusted weight estimates (Line 4 in Algorithms 1 and 3) so that the ordering between the positive-type weight estimate and the negative-type weight estimate is likely to be the same for both algorithms. In other words, despite CC-CLCB and Global-CLCB may compute different actual values of those weight estimates, the two algorithms are mostly consistent in yielding a positive-type weight estimate that is higher/lower than the negative-type one. This leads to very similar clusterings yielded by the Pivot MIN-CC oracle in every run and every round of both CC-CLCB and Global-CLCB, as Pivot places any two vertices in the same clustering by solely checking whether the positive-type weight on the edge between those vertices is higher than the negative-type one, without looking at the specific values of those weights.

Quality of the learned edge weights (Table 6). In terms of edge weights, the picture in the *PC-wd* and *C-wd* settings is roughly consistent with what observed in terms of clustering quality. Some differences arise in the *R-wd* setting, where, unlike the clustering quality criterion, CC-CLCB-m and Global-CLCB-m are the best methods only in a few configurations (they are outperformed mostly by PC+Exp-CLCB).

As another interesting observation, here are some differences between CLCB and Global-CLCB, and between CLCB-m and Global-CLCB-m. This confirms the argument discussed above, i.e., that those methods achieve the same clustering results even though they can learn different weight estimates.

6.3 Varying the MIN-CC oracle

Table 7 shows the performance of all the competing methods when using LP+R as a MIN-CC oracle (instead of Pivot). Here, we also show the relative difference (in percentage) between the score with LP+R and the corresponding score with Pivot. Thus, the more positive (resp. negative) such a relative difference, the worse (resp. better) the performance of using LP+R than using Pivot. The general trend in terms of clustering quality (Table 7a) is that LP+R leads to an increase (resp. decrease) in performance in the *R-wd* and *PC-wd* settings (resp. *C-wd* setting). This is likely due to the fact that *R-wd* and *PC-wd* are more challenging than *C-wd*, as it is well-known that MIN-CC is easier on complete-graph input instances (Charikar et al. 2005). In fact, LP+R provides approximation guarantees at each round,

Table 5 Performance in terms of $f^{(T)}$ (Eq. 12) and g_r^{pk} (Eq. 14), for the CLCB methods (equipped with Pivot as a MIN-CC oracle)

(a) R -wtd weight distributions										
Method	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	0.66 -11.97%	0.656 -10.3%	0.671 -15.07%	0.583 -12.34%	0.675 -12.96%	0.749 -0.13%	0.74 -1.04%	0.747 -0.42%	0.748 -0.34%	0.717 -4.45%
CC-CLCB-m	0.652 -13.07%	0.646 -11.61%	0.669 -15.35%	0.573 -13.89%	0.668 -13.83%	0.749 -0.16%	0.738 -1.25%	0.747 -0.41%	0.747 -0.37%	0.713 -4.91%
PC+Exp-CLCB	0.658 -12.32%	0.649 -11.22%	0.675 -14.6%	0.58 -12.87%	0.675 -12.93%	0.749 -0.12%	0.741 -0.96%	0.746 -0.54%	0.748 -0.33%	0.716 -4.52%
Global-CLCB	0.66 -11.97%	0.656 -10.26%	0.671 -15.08%	0.584 -12.27%	0.675 -12.98%	0.749 -0.13%	0.74 -1.04%	0.747 -0.42%	0.748 -0.34%	0.717 -4.45%
Global-CLCB-m	0.652 -13.07%	0.646 -11.63%	0.669 -15.35%	0.573 -13.89%	0.668 -13.88%	0.749 -0.16%	0.738 -1.25%	0.747 -0.41%	0.747 -0.37%	0.713 -4.91%
(b) PC -wtd weight distributions										
Method	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	0.59 -18.22%	0.54 -19.09%	0.58 -21.67%	0.51 -20.08%	0.53 -24.36%	0.67 -0.19%	0.65 -1.94%	0.66 -0.79%	0.66 -0.56%	0.62 -7.28%
CC-CLCB-m	0.59 -19.06%	0.53 -21.1%	0.57 -22.36%	0.5 -21.27%	0.52 -25.82%	0.66 -0.22%	0.64 -2.28%	0.66 -0.75%	0.66 -0.6%	0.61 -7.93%
PC+Exp-CLCB	0.58 -20.02%	0.51 -23.18%	0.57 -23.15%	0.49 -22.56%	0.5 -27.67%	0.66 -0.28%	0.64 -2.77%	0.66 -1.19%	0.66 -0.68%	0.61 -8.97%
Global-CLCB	0.59 -18.22%	0.54 -19.09%	0.58 -21.65%	0.51 -20.06%	0.53 -24.39%	0.67 -0.19%	0.65 -1.94%	0.66 -0.79%	0.66 -0.56%	0.62 -7.28%

Table 5 (continued)

(b) <i>PC-wd</i> weight distributions										
Method	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA	Lastfm	Primary-School	Proser-Loans	Wikipedia	DBLP
Global-CLCB-m	0.59 -19.06%	0.53 -21.12%	0.57 -22.42%	0.5 -21.27%	0.52 -25.89%	0.66 -0.22%	0.64 -2.29%	0.66 -0.75%	0.66 -0.6%	0.61 -7.93%
(c) <i>C-wd</i> weight distributions (smaller datasets only)										
Method	Karate			Dolphins		Zebra		HighlandTribes		Contiguous-USA
CC-CLCB	0.12	-14.04%	0.072	-14.14%	0.091	-71.36%	0.338	-30.04%	0.064	-29.4%
CC-CLCB-m	0.113	-19.01%	0.071	-16.12%	0.07	-77.8%	0.335	-30.63%	0.063	-30.69%
PC+Exp-CLCB	0.11	-20.85%	0.072	-14.89%	0.059	-81.44%	0.334	-30.86%	0.065	-28.97%
Global-CLCB	0.12	-14.04%	0.072	-14.14%	0.091	-71.36%	0.338	-30.07%	0.064	-29.4%
Global-CLCB-m	0.113	-19.01%	0.071	-16.12%	0.07	-77.8%	0.335	-30.63%	0.063	-30.69%

Best scores per dataset are reported in bold

Table 6 Performance in terms of $rep^{(T)}$ (Eq. 13) and $gr_{mm}^{(k)}$ (Eq. 14), for the CLCB methods (equipped with Pivot as a MIN-CC oracle)

(a) <i>R-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.175 -80%	0.186 -78.52%	0.056 -93.37%	0.199 -78.1%	0.146 -83.17%	0.053 -93.86%	0.066 -92.44%	0.412 -52.41%	0.217 -74.99%	0.158 -81.75%
CC-CLCB-m	0.202 -76.97%	0.162 -81.25%	0.059 -92.92%	0.146 -84%	0.151 -82.62%	0.053 -93.87%	0.067 -92.28%	0.388 -55.26%	0.212 -75.47%	0.144 -83.35%
PC+Exp-CLCB	0.157 -82.12%	0.164 -81%	0.074 -91.11%	0.145 -84.07%	0.146 -83.15%	0.052 -94.04%	0.063 -92.71%	0.376 -56.64%	0.206 -76.24%	0.155 -82.15%
Global-CLCB	0.175 -80.05%	0.183 -78.82%	0.058 -93.12%	0.221 -75.79%	0.144 -83.43%	0.053 -93.86%	0.066 -92.44%	0.412 -52.41%	0.217 -74.98%	0.158 -81.74%
Global-CLCB-m	0.202 -76.97%	0.163 -81.17%	0.059 -92.92%	0.146 -84%	0.15 -82.71%	0.053 -93.87%	0.067 -92.28%	0.388 -55.26%	0.212 -75.47%	0.144 -83.35%
(b) <i>PC-wd weight distributions</i>										
Method	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	0.19 -76.88%	0.26 -68.89%	0.05 -93.14%	0.25 -71.42%	0.14 -83.15%	0.05 -93.89%	0.07 -92.29%	0.52 -39.69%	0.22 -74.07%	0.2 -77.23%
CC-CLCB-m	0.17 -80.06%	0.19 -77.17%	0.07 -91.03%	0.18 -79.08%	0.14 -82.39%	0.05 -93.84%	0.07 -91.97%	0.47 -45.91%	0.22 -74.67%	0.17 -79.9%
PC+Exp-CLCB	0.13 -84.36%	0.16 -81.36%	0.08 -90.19%	0.13 -85.14%	0.13 -83.69%	0.05 -93.73%	0.08 -91.32%	0.34 -60.23%	0.22 -74.67%	0.16 -81.98%
Global-CLCB	0.19 -76.88%	0.26 -68.89%	0.05 -93.1%	0.24 -72.5%	0.14 -83.23%	0.05 -93.89%	0.07 -92.29%	0.52 -39.69%	0.22 -74.07%	0.2 -77.22%
Global-CLCB-m	0.17 -80.06%	0.2 -76.31%	0.07 -90.97%	0.18 -79.08%	0.15 -82.31%	0.05 -93.84%	0.07 -91.97%	0.47 -45.91%	0.22 -74.67%	0.17 -79.91%

Table 6 (continued)

(c) <i>C-wd weight distributions (smaller datasets only)</i>											
CC-CLCB	0.623	-32.82%	0.824	-13.9%	0.673	-18.66%	0.158	-78.01%	0.87	-8.76%	
CC-CLCB-m	0.623	-32.82%	0.824	-13.9%	0.674	-18.45%	0.158	-78.01%	0.87	-8.76%	
PC+Exp-CLCB	0	-100%	0	-100%	0	-100%	0	-100%	0	-100%	
Global-CLCB	0.623	-32.82%	0.824	-13.9%	0.673	-18.66%	0.158	-78.01%	0.87	-8.76%	
Global-CLCB-m	0.623	-32.82%	0.824	-13.9%	0.674	-18.45%	0.158	-78.01%	0.87	-8.76%	

Best scores per dataset are reported in bold

regardless of the weights in input to the oracle. Conversely, *Pivot* provides quality guarantees if the probability constraint holds on the given input, but this is not the case in a generic round t . The C - wd setting (i.e., complete graph with probability constraint) corresponds to the most favorable scenario for *Pivot* to provide approximation guarantees.

In terms of learned edge weights (Table 7b) the advantage of using LP+R is less evident. A reason might lie in the different random choices of the two algorithms (i.e., choosing the node around which a cluster is being built in *Pivot*, and integer rounding in LP+R): the random choices of *Pivot* likely lead to more exploration, hence a better chance to discover weights close to the true ones.

6.4 Efficiency

Table 8 shows the runtimes of the tested methods on the larger datasets, averaged over the various runs and over the R - wd and PC - wd weight settings. Despite all the CMAB methods are roughly comparable with each other, *CTS* is however the slowest method, as it involves additional sampling operations with respect to the other ones.

The CMAB methods take seconds on the smaller Last.fm and PrimarySchool datasets, around one hour on ProsperLoans, and up to 3–5 h on the largest datasets, i.e., Wikipedia and DBLP. In general, however, we can conclude that all the CMAB methods are rather efficient. Even the highest runtimes on Wikipedia and DBLP are not worrying, considering that such datasets have around 10 M edges, and, more importantly, that the reported runtimes are cumulative of all the 500 CMAB rounds. In fact, the highest per-round runtime of a CMAB method is always—at worst—comparable to the runtime of *Actual-weight*, which performs *MIN-CC* clustering just once. In most cases, it is even less, likely because the time of the round-independent steps is amortized over the various rounds.

Further results are shown in Table 9 and include the use of both oracles for the smallest datasets in our collection, according to all weight settings. As it can be noted in the table, the above qualitative remarks on the relative differences between the methods are equally evident.

6.5 Statistical significance

Here we present a further step of analysis to assess the statistical significance of the performance of the CMAB-*MIN-CC* methods *CC-CLCB*, *Global-CLCB*, *PC+Exp-CLCB*, *CTS*, and *EG*, when equipped with *Pivot* as a *MIN-CC* oracle.

To this purpose, we resorted to a Friedman's test. We designed it by considering all the methods, all the datasets, and all the weight settings in one single test. More specifically, we organized the data into a matrix with 5 columns (treatments) corresponding to the methods, and 250 rows (blocks) corresponding to the number of combinations between runs (10), datasets (10), and weight settings (R - wd and PC - wd available for all 10 datasets and C - wd available for 5 datasets), where each

Table 7 Performance with LP+R as a Min-CC oracle, and relative difference (in percentage) between the score achieved with LP+R and the corresponding score achieved with Pivot

Method	Weight	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA					
<i>(a) $f^{(T)}$ criterion (Eq. 12)</i>											
CC-CLCB	<i>R-wd</i>	0.58	-12.12%	0.592	-9.76%	0.669	-0.3%	0.565	-3.09%	0.623	-7.7%
	<i>PC-wd</i>	0.458	-22.77%	0.456	-15.56%	0.562	-2.77%	0.554	9.49%	0.444	-15.59%
	<i>C-wd</i>	0.139	15.83%	0.084	16.67%	0.097	6.59%	0.483	42.9%	0.091	42.19%
EG	<i>R-wd</i>	0.567	-12.36%	0.569	-10.11%	0.658	-1.05%	0.532	-5.51%	0.607	-8.03%
	<i>PC-wd</i>	0.442	-24.32%	0.427	-17.09%	0.517	-8.82%	0.498	-0.2%	0.43	-15.52%
	<i>C-wd</i>	0.151	17.05%	0.105	15.38%	0.089	9.88%	0.484	39.08%	0.112	33.33%
EG-fixed	<i>R-wd</i>	0.57	-12.58%	0.578	-9.26%	0.666	-0.6%	0.544	-4.39%	0.614	-8.08%
	<i>PC-wd</i>	0.458	-22.37%	0.441	-15.68%	0.552	-3.83%	0.5	-0.99%	0.443	-14.15%
	<i>C-wd</i>	0.167	13.61%	0.12	11.11%	0.117	9.35%	0.484	39.48%	0.126	23.53%
PE	<i>R-wd</i>	0.602	-6.81%	0.59	-6.94%	0.688	3.77%	0.614	9.84%	0.631	-5.54%
	<i>PC-wd</i>	0.471	-19.35%	0.437	-14.48%	0.625	11.61%	0.585	19.88%	0.442	-11.07%
	<i>C-wd</i>	0.139	27.52%	0.084	20%	0.067	17.54%	0.483	44.61%	0.091	44.44%
CTS	<i>R-wd</i>	0.549	-15.15%	0.561	-11.65%	0.641	-3.32%	0.523	-6.77%	0.597	-10.23%
	<i>PC-wd</i>	0.434	-25.17%	0.415	-19.26%	0.507	-9.63%	0.483	-1.83%	0.42	-17.32%
	<i>C-wd</i>	0.111	0%	0.066	-9.59%	0.067	9.84%	0.475	42.22%	0.073	10.61%
Adamic-Adar	<i>R-wd</i>	0.746	-1.58%	0.729	-3.95%	0.78	2.09%	0.646	-0.31%	0.747	-3.98%
	<i>PC-wd</i>	0.669	-4.43%	0.68	-0.44%	0.703	0.86%	0.607	-2.25%	0.583	-5.82%
	<i>C-wd</i>	0.141	3.68%	0.089	-18.35%	0.305	1.33%	0.483	4.55%	0.111	2.78%
Jaccard	<i>R-wd</i>	0.759	-0.65%	0.75	-1.06%	0.721	2.56%	0.727	1.68%	0.765	-1.03%
	<i>PC-wd</i>	0.741	0.68%	0.683	-0.73%	0.605	2.54%	0.603	0.5%	0.678	1.19%
	<i>C-wd</i>	0.119	-2.46%	0.075	0%	0.188	37.23%	0.492	29.47%	0.123	59.74%
Actual-weight	<i>R-wd</i>	0.537	-16.09%	0.547	-12.9%	0.662	1.69%	0.506	-8%	0.583	-12.07%
	<i>PC-wd</i>	0.419	-27.63%	0.394	-21.2%	0.501	-7.73%	0.494	0.61%	0.411	-16.8%
	<i>C-wd</i>	0.132	17.86%	0.084	21.74%	0.066	22.22%	0.483	47.26%	0.091	49.18%

Table 7 (continued)

Method	Weight	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
<i>(b) $rel^{(T)}$ criterion (Eq. 13)</i>						
CC-CLCB	<i>R-wd</i>	0.087>	0.075	0.113	0.086	0.089
	<i>PC-wd</i>	0.076	0.095	0.141	0.091	0.074
EG	<i>C-wd</i>	0.928	0.957	0.47	0.719	0.953
	<i>R-wd</i>	0.156	0.151	0.128	0.143	0.149
	<i>PC-wd</i>	0.161	0.168	0.125	0.115	0.141
	<i>C-wd</i>	0	0	0	0	0
EG-fixed	<i>R-wd</i>	0.112	0.113	0.101	0.132	0.106
	<i>PC-wd</i>	0.125	0.115	0.101	0.106	0.115
	<i>C-wd</i>	0	0	0	0	0
PE	<i>R-wd</i>	0.403	0.402	0.336	0.644	0.426
	<i>PC-wd</i>	0.399	0.39	0.41	0.602	0.381
	<i>C-wd</i>	0.928	0.957	0.512	0.719	0.953
CTS	<i>R-wd</i>	0.137	0.146	0.128	0.19	0.129
	<i>PC-wd</i>	0.124	0.17	0.134	0.17	0.128
	<i>C-wd</i>	0.467	0.48	0.432	0.284	0.469
Adamic-Adar	<i>R-wd</i>	0.649	0.671	0.617	0.531	0.592
	<i>PC-wd</i>	0.661	0.695	0.627	0.542	0.609
	<i>C-wd</i>	0.514	0.446	0.54	0.706	0.372
Jaccard	<i>R-wd</i>	0.644	0.637	0.77	0.577	0.517
	<i>PC-wd</i>	0.72	0.638	0.733	0.532	0.555
	<i>C-wd</i>	0.429	0.294	0.281	0.596	0.252
Actual-weight	<i>R-wd</i>	0	0	0	0	0
	<i>PC-wd</i>	0	0	0	0	0
	<i>C-wd</i>	0	0	0	0	0

Best scores per dataset and weight distribution are reported in bold

Table 8 Running times (in secs.) on the larger datasets. Results correspond to average runtime performances in the *R-wd* and *PC-wd* distribution settings and *Pivot* as a MIN-CC oracle

Method	Last.fm	PrimarySchool	ProsperLoans	Wikipedia	DBLP
CC-CLCB	194.77	3.57	3290.48	12089.04	16294.69
CC-CLCB-m	203.89	3.57	3255.32	11988.76	16263.11
PC+Exp-CLCB	197.65	3.55	3312.86	12017.06	16371.6
Global-CLCB	194.58	3.56	3286.15	12082.44	16424.53
Global-CLCB-m	194.88	3.55	3252.51	12261.51	16354.55
EG	184.67	3.41	3233.76	11554.83	16411.67
EG-fixed	190.89	3.43	3085.8	11431.45	15992.45
PE	193.51	3.58	3469.18	12310.03	17187.58
CTS	210.58	4.4	3753.27	12966.74	17819.99
Adamic-Adar	2.93	0.1	112.1	347.45	315.22
Jaccard	2.62	0.06	105.68	337.34	305.87
Actual-weight	2.24	0.04	40.18	145.75	199.42

Table 9 Running times (in secs.)

Method	Oracle	Karate	Dolphins	HighlandTribes	Contiguous-USA	Zebra
CC-CLCB	Pivot	0.17	0.37	0.07	0.25	0.11
	LP+R	172.86	3058.17	7.66	963.88	57.41
CC-CLCB-m	Pivot	0.17	0.37	0.07	0.25	0.11
	LP+R	172.85	3013.26	7.58	962.15	56.72
PC+Exp-CLCB	Pivot	0.16	0.37	0.07	0.24	0.11
Global-CLCB	Pivot	0.17	0.38	0.07	0.29	0.11
Global-CLCB-m	Pivot	0.17	0.37	0.07	0.24	0.11
EG	Pivot	0.16	0.42	0.07	0.23	0.1
	LP+R	136.11	2403.18	5.64	743.16	48.2
EG-fixed	Pivot	0.16	0.34	0.06	0.27	0.1
	LP+R	130.89	2318.57	5.32	726.31	44.97
PE	Pivot	0.16	0.37	0.07	0.25	0.11
	LP+R	166.88	2896.21	7.93	941.58	51.64
CTS	Pivot	0.28	0.51	0.11	0.47	0.25
	LP+R	190.96	3579.58	9.14	1084.56	71.9
Adamic-Adar	Pivot	0.0	0.01	0.0	0.0	0.01
	LP+R	1.61	31.23	0.08	10.95	0.59
Jaccard	Pivot	0.0	0.01	0.0	0.0	0.0
	LP+R	1.53	29.26	0.08	9.46	0.65
Actual-weight	Pivot	0.0	0.01	0.0	0.0	0.01
	LP+R	1.82	32.6	0.08	10.08	0.72

Results correspond to average runtime performances in the *R-wd*, *PC-wd* and *C-wd* distribution settings

cell measures the average expected normalized cumulative loss (i.e., $f^{(T)}$) (Eq. 12) obtained by a particular method at the last round ($T = 500$) on a particular configuration of run, dataset, weight setting. (Note that each run corresponds to a different fixed seed for handling computation randomness.)

Our Friedman's test results indicate that there are significant differences— $\chi^2(4) = 158.1$, p-value $< 2.2\text{E-}16$ —in the average expected normalized cumulative losses in run/dataset/weight-setting blocks based on the methods, i.e., the methods have different effect on average expected normalized cumulative loss obtained on each run/dataset/weight-setting combination.

We also computed the Kendall's coefficient of concordance (Kendall's W) for measuring the effect size (degree of difference) for Friedman's test. From the result above, Kendall's W is 0.304, which indicates an effect size at the boundary of the "small" and the "moderate" effects based on Cohen's interpretation guidelines (Tomczak and Tomczak 2014).

Since Friedman's test is an omnibus test statistic, in order to know which methods are significantly different, we carried out Nemenyi's all-pairs test as a post-hoc test for pairwise comparisons of methods, where the Bonferroni correction was used to adjust the p-values for multiple hypothesis testing at a 5% cut-off. Results show p-values in the range (10^{-14} , 10^{-4}) (i.e., significant differences) for all pairs but CC-CLCB vs. Global-CLCB. It should be noted that the lack of statistical difference between CC-CLCB and Global-CLCB is not surprising: in fact, in Sect. 6.2, we already noticed and explained why CC-CLCB and Global-CLCB achieve the same performance in terms of $f^{(T)}$, in all the datasets and weight settings.

7 Additional experiments

Clustering size. Tables 10, 11 and 12 show the number of clusters yielded by the tested methods, averaged over all the CMAB rounds and runs of the MIN-CC oracle. For the CMAB methods, we also provide the difference (in percentage) between the number of clusters at the final round and the number of clusters at the first round (both averaged over the runs of the MIN-CC oracle). By inspecting such results, we notice that in the cases of *R-wd* and *PC-wd* weight distributions, the use of LP+R oracle generally corresponds to less clusters compared to the PIVOT oracle; some exceptions are observed for small-world datasets (e.g., Zebra, HighlandTribes) by most methods, especially with *R-wd*. The *PC-wd* mostly lead to less clusters than *R-wd*. Conversely, with the *C-wd* setting, the LP+R oracle leads consistently to a much larger number of clusters (at least double in many cases) than PIVOT.

Moreover, we observe that the non-CMAB methods (i.e., Adamic-Adar and Jaccard) produce a relatively small number of clusters as long as the characteristics of the input dataset are those typical of a small-world network; for instance, in Last.fm and PrimarySchool, the clustering size is about 90% and 97% of the vertex set size, respectively. This is not surprising, as the adopted approaches of (CMAB) correlation clustering are not designed to optimize some criterion function defined

Table 10 Number of clusters for *R-wd* weight setting: average over all rounds and runs, and (for the CMAB methods) relative (percentage) difference between the last and first round

Method	Oracle	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA
CC-CLCB	Pivot	17.8	29.8	9.2	6.2	20.4
		-47.65%	-51.94%	-65.93%	-61.25%	-58.37%
	LP+R	14.4	21	11.4	9.4	13.6
CC-CLCB-m	Pivot	18.2	30	8.2	6.4	22.2
		-46.47%	-51.61%	-69.63%	-60%	-54.69%
	LP+R	14	20	10.2	8.2	14.4
PC+Exp-CLCB	Pivot	18.4	31	9.4	5.2	20.4
		-45.88%	-50%	-65.19%	-67.5%	-58.37%
Global-CLCB	Pivot	18	29.8	8.8	6.2	20.8
		-47.06%	-51.94%	-67.41%	-61.25%	-57.55%
Global-CLCB-m	Pivot	18.2	30	8.2	6.4	22.6
		-46.47%	-51.61%	-69.63%	-60%	-53.88%
EG	Pivot	20	32.8	10	6.8	23.2
		21.87%	5.24%	8.86%	2.83%	5.54%
	LP+R	13.8	21	13.4	5.6	15
EG-fixed	Pivot	20.8	32.6	9.4	7.4	23.8
		-38.82%	-47.42%	-65.19%	-53.75%	-51.43%
	LP+R	13.8	18.6	12.2	5.6	15.4
PE	Pivot	21.2	37	9.6	7.6	27.6
		-37.65%	-40.32%	-64.44%	-52.5%	-43.67%
	LP+R	16.4	24.4	16.4	12	19.2
CTS	Pivot	18.4	31.8	9.2	6.6	23.2
		-10.72%	-1.06%	-13.67%	10.5%	-3.32%
	LP+R	13.6	18.8	10.8	5.2	13.8
Adamic-Adar	Pivot	23.2	40.8	17.2	9.4	15.2
		-51.76%	-60.65%	-39.26%	-25%	-60.82%
	LP+R	18	36	19	8	3
Jaccard	Pivot	23.8	49.8	7	7.2	35.2
		-37.65%	-40.32%	-64.44%	-52.5%	-43.67%
Actual-weight	Pivot	20.4	31.4	10	6.8	21.6
		-51.76%	-60.65%	-39.26%	-25%	-60.82%
LP+R	14	20.2	14	5.6	13	
Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP	
CC-CLCB	29.8	23.8	3.768E+4	2.644E+5	1.02E+6	
	-97%	-90.17%	-57.8%	-23.1%	-44.07%	

Table 10 (continued)

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB-m	33.6 -96.61%	26.8 -88.93%	3.854E+4 -56.83%	2.648E+5 -22.98%	1.027E+6 -43.7%
PC+Exp-CLCB	29.8 -97%	23.2 -90.41%	3.86E+4 -56.77%	2.638E+5 -23.27%	1.01E+6 -44.64%
Global-CLCB	29.8 -97%	23.8 -90.17%	3.768E+4 -57.79%	2.644E+5 -23.1%	1.02E+6 -44.08%
Global-CLCB-m	32.6 -96.71%	26.8 -88.93%	3.854E+4 -56.83%	2.648E+5 -22.98%	1.027E+6 -43.7%
EG	47.4 324.15%	30.8 46.71%	3.701E+4 105.86%	2.686E+5 73.5%	1.063E+6 30.98%
EG-fixed	36.6 -96.31%	29.6 -87.77%	3.617E+4 -59.48%	2.456E+5 -28.58%	1.007E+6 -44.81%
PE	41 -95.87%	30.2 -87.52%	4.32E+4 -51.61%	2.706E+5 -21.32%	1.121E+6 -38.56%
CTS	41 -4.8%	29 1.2%	3.909E+4 0.04%	2.665E+5 0.01%	1.042E+6 0.02%
Adamic-Adar	992	242	8.924E+4	3.419E+5	1.393E+6
Jaccard	66.2	114.8	8.921E+4	2.912E+5	1.172E+6
Actual-weight	43.2	28.4	3.921E+4	2.666E+5	1.042E+6

on topological properties at meso- and macroscopic level (e.g., modularity), which results in a need for refining the clustering solutions through a cluster aggregation stage.

Performance over the CMAB rounds. Figures 2 and 3 illustrate the performance—in terms of average expected normalized cumulative MIN-CC loss $f^{(t)}$ (Eq. 12)—of the tested methods over the various CMAB rounds t . As expected, the CMAB methods mostly exhibit a decreasing trend, with a decrease in loss scores that is more consistent in the first rounds, until it gets progressively vanishing as the rounds go on, meaning convergence in the weight learning process (and, thus, in the clustering quality too). A few exceptions to this strictly monotonically decreasing trend arise (e.g., with some CLCB-based methods in Last.fm *PC-wd*, ProsperLoans *R-wd*, ProsperLoans *PC-wd*, DBLP *PC-wd*). However, the minimum of the $f^{(t)}$ function in all those exceptional cases is only slightly less than the value of $f^{(t)}$ at convergence (i.e., the difference is less than 0.004). Thus, remembering also that $f^{(t)}$ is an average of all the losses computed up to round t , we can conclude that those non-monotonic trends actually correspond to the normal fluctuations of the loss values in the first CMAB rounds, when there cannot be enough knowledge on the actual edge weights to get stable clustering quality.

Table 11 Number of clusters for *PC-wd* weight setting: average over all rounds and runs, and (for the CMAB methods) relative (percentage) difference between the last and first round

Method	Oracle	Karate	Dolphins	Zebra	Highland-Tribes	Contiguous-USA
CC-CLCB	Pivot	17.2	29	9.6	5.6	22.8
		-49.41%	-53.23%	-64.44%	-65%	-53.47%
	LP+R	8.8	19.2	9.8	11.6	18.4
		-74.12%	-69.03%	-63.7%	-27.5%	-62.45%
CC-CLCB-m	Pivot	17.6	30.2	9	5.6	22.6
		-48.24%	-51.29%	-66.67%	-65%	-53.88%
	LP+R	10.8	22.4	10.4	10	17
		-68.24%	-63.87%	-61.48%	-37.5%	-65.31%
PC+Exp-CLCB	Pivot	18	31	9.8	5.8	22.2
		-47.06%	-50%	-63.7%	-63.75%	-54.69%
Global-CLCB	Pivot	17.2	29	9.6	5.6	22.8
		-49.41%	-53.23%	-64.44%	-65%	-53.47%
Global-CLCB-m	Pivot	17.6	30.6	9	5.6	22.8
		-48.24%	-50.65%	-66.67%	-65%	-53.47%
EG	Pivot	16.6	29.4	10.2	5.8	22
		-1.22%	-5.37%	10.28%	-11%	0%
	LP+R	8.8	17.2	7.8	7.2	17.2
		-42.61%	-43.16%	-9.85%	58.33%	-17.91%
EG-fixed	Pivot	19.2	31	9.6	6.2	21.8
		-43.53%	-50%	-64.44%	-61.25%	-55.51%
	LP+R	8.4	17.4	10	6.4	19
		-75.29%	-71.94%	-62.96%	-60%	-61.22%
PE	Pivot	20.4	35	10.4	5.8	23.8
		-40%	-43.55%	-61.48%	-63.75%	-51.43%
	LP+R	11.4	20.4	17.2	13.2	18
		-66.47%	-67.1%	-36.3%	-17.5%	-63.27%
CTS	Pivot	15.2	29.4	9	6	21.4
		-25.94%	-8.38%	-16%	-1%	-10.83%
	LP+R	9.4	18.4	9	7.4	17.6
		-10.14%	-6.42%	-22.81%	31%	22.67%
Adamic-Adar	Pivot	23.2	40.8	17.2	9.4	15.2
	LP+R	18	36	19	8	3
Jaccard	Pivot	23.8	49.8	7	7.2	35.2
	LP+R	25	51	10	7	32
Actual-weight	Pivot	18.2	29	9.2	6.8	22.6
	LP+R	8.4	16	8.6	6.4	18
Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP	
CC-CLCB	37.6	24	3.804E+4	2.646E+5	1.023E+6	
	-96.21%	-90.08%	-57.39%	-23.05%	-43.92%	

Table 11 (continued)

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB-m	44.6	26	3.867E+4	2.653E+5	1.032E+6
	-95.5%	-89.26%	-56.68%	-22.86%	-43.45%
PC+Exp-CLCB	44.6	27.4	3.935E+4	2.662E+5	1.041E+6
	-95.5%	-88.68%	-55.92%	-22.57%	-42.93%
Global-CLCB	37.6	24	3.804E+4	2.646E+5	1.023E+6
	-96.21%	-90.08%	-57.39%	-23.05%	-43.92%
Global-CLCB-m	44.8	25.6	3.867E+4	2.653E+5	1.032E+6
	-95.48%	-89.42%	-56.68%	-22.86%	-43.45%
EG	43	30	3.603E+4	2.675E+5	1.053E+6
	299%	42.79%	100.44%	72.82%	29.79%
EG-fixed	33.6	29.2	3.547E+4	2.45E+5	1.002E+6
	-96.61%	-87.93%	-60.27%	-28.74%	-45.06%
PE	49	29.4	4.178E+4	2.691E+5	1.091E+6
	-95.06%	-87.85%	-53.19%	-21.74%	-40.2%
CTS	43.4	28.6	3.898E+4	2.667E+5	1.042E+6
	1.15%	-0.88%	-0.23%	0.11%	0%
Adamic-Adar	992	242	8.924E+4	3.419E+5	1.393E+6
Jaccard	66.2	114.8	8.921E+4	2.912E+5	1.172E+6
Actual-weight	44.4	30.2	3.921E+4	2.666E+5	1.042E+6

Stability over the CMAB rounds. Tables 13, 14, 15, 16, 17 and 18 show the coefficient of variation (i.e., ratio between standard deviation and mean) of the scores of the tested methods in terms of $f^{(T)}$ (Eq. 12) and $ren^{(T)}$ (Eq. 13) criteria, respectively. It can be observed that the coefficients of variations of $f^{(T)}$ are typically very small for all the methods: they mostly range from $[10^{-3}, 10^{-2}]$ in the smaller datasets, and from $[10^{-6}, 10^{-4}]$ in the larger datasets, with only a very few exceptions. In terms of $ren^{(T)}$, the coefficients of variations in the *R-wd* and *PC-wd* weight settings are higher (especially in the smaller datasets), but they still remain rather small. In the *C-wd* setting, they are instead mostly equal or very close to zero. Therefore, as a general conclusion, we can state that the various tested methods exhibit high stability over the CMAB rounds and runs of the MIN-CC oracle.

8 Conclusion

We have focused on the novel setting of correlation clustering where edge weights are unknown, and they need be discovered while performing multiple rounds of clustering. We have provided a Combinatorial Multi-Armed Bandit (CMAB) framework for

Table 12 Number of clusters for *C-wd* weight setting: average over all rounds and runs, and (for the CMAB methods) relative (percentage) difference between the last and first round

Method	Oracle	Karate		Dolphins		Zebra		Highland Tribes		Contiguous-USA	
		Final	Growth	Final	Growth	Final	Growth	Final	Growth	Final	Growth
CC-CLCB	Pivot	18.6	-45.29%	26.2	-57.74%	5.8	-78.52%	3.6	-77.5%	14.6	-70.2%
	LP+R	34	0%	62	0%	9.2	-65.93%	16	0%	49	0%
CC-CLCB-m	Pivot	18.6	-45.29%	26.2	-57.74%	5.8	-78.52%	3.8	-76.25%	14.6	-70.2%
	LP+R	34	0%	62	0%	9	-66.67%	16	0%	49	0%
PC+Exp-CLCB	Pivot	18.8	-44.71%	27	-56.45%	5.8	-78.52%	3.6	-77.5%	15.2	-68.98%
Global-CLCB	Pivot	18.6	-45.29%	26.2	-57.74%	5.8	-78.52%	3.6	-77.5%	14.6	-70.2%
Global-CLCB-m	Pivot	18.6	-45.29%	26.2	-57.74%	5.8	-78.52%	3.8	-76.25%	14.6	-70.2%
EG	Pivot	17.8	196.67%	25.8	268.57%	6	20%	2.8	-44%	15.2	117.14%
	LP+R	31	416.67%	62	785.71%	9	80%	16	220%	49	600%
EG-fixed	Pivot	18.6	-45.29%	25	-59.68%	5.6	-79.26%	3.4	-78.75%	15.8	-67.76%
	LP+R	31	-8.82%	62	0%	9	-66.67%	16	0%	49	0%
PE	Pivot	18.8	-44.71%	27	-56.45%	5.8	-78.52%	3.6	-77.5%	15.2	-68.98%
	LP+R	34	0%	62	0%	9	-66.67%	16	0%	49	0%
CTS	Pivot	19	284.33%	24.8	293.33%	5.8	33%	3.6	8.33%	15.2	166.86%
	LP+R	24.4	-28.24%	42.8	-30.97%	9	-66.41%	16	103.43%	36.8	-24.9%
Adamic-Adar	Pivot	20.2	-	26	-	15.4	-	7.6	-	9.4	-
	LP+R	23	-	51	-	20	-	9	-	10	-
Jaccard	Pivot	24.8	-	49.8	-	7	-	7.2	-	35.8	-
	LP+R	25	-	50	-	10	-	7	-	33	-
<i>Actual-weight</i>	Pivot	18.8	-	25.4	-	5.6	-	3.4	-	15.6	-
	LP+R	31	-	62	-	9	-	16	-	49	-

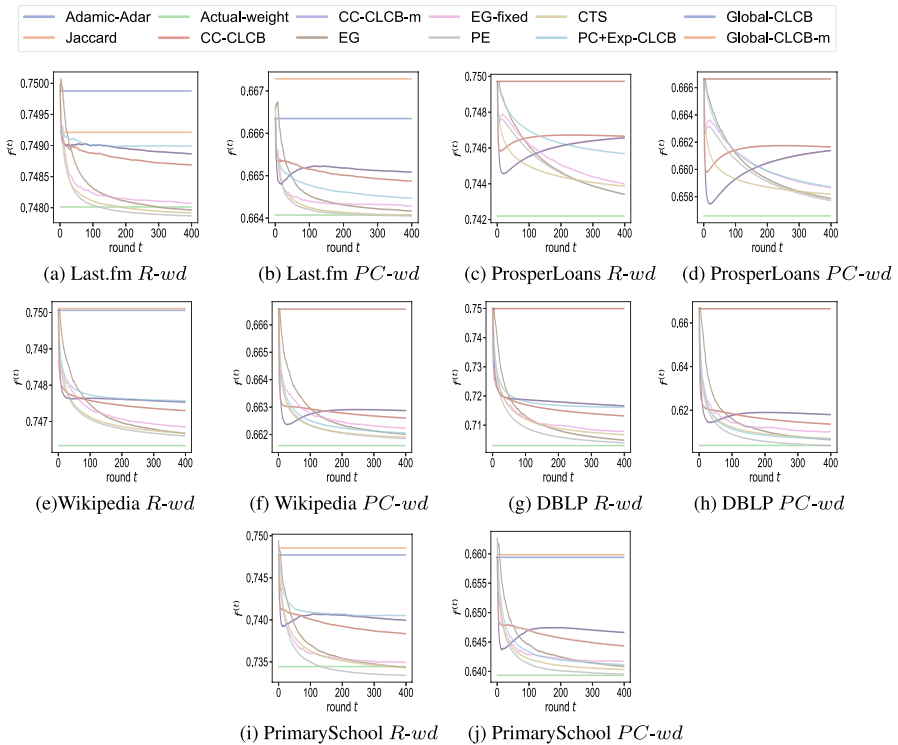


Fig. 2 Performance in terms of $f^{(t)}$ (Eq. 12), over a number $t = 1, \dots, 400$ of rounds, for the larger datasets, and R -wd and PC -wd weight distributions. All the CMAB methods are equipped with Pivot as a MIN-CC oracle

correlation clustering, algorithms for it, analyses of the theoretical guarantees of these algorithms, more practical heuristics, and extensive experiments.

In the future, we plan to investigate the theoretical properties of our heuristics, advanced CMAB settings, and clustering problems other than correlation clustering.

For reproducibility purposes, we make source code and data available at: <https://github.com/Ralyhu/CMAB-CC>, and <http://people.dimes.unical.it/andreatagarelli/CMAB-CC/>.

Appendix

A.1: From expected theoretical guarantees to (α, β) -approximation oracle

Approximation algorithms for correlation clustering are typically randomized and achieve quality guarantees *in expectation* (Ailon et al. 2008; Charikar et al. 2005; Demaine et al. 2006; Swamy 2004). Here we show how to use a correlation-clustering algorithm with expected factor- ϵ approximation as an (α, β) -approximation

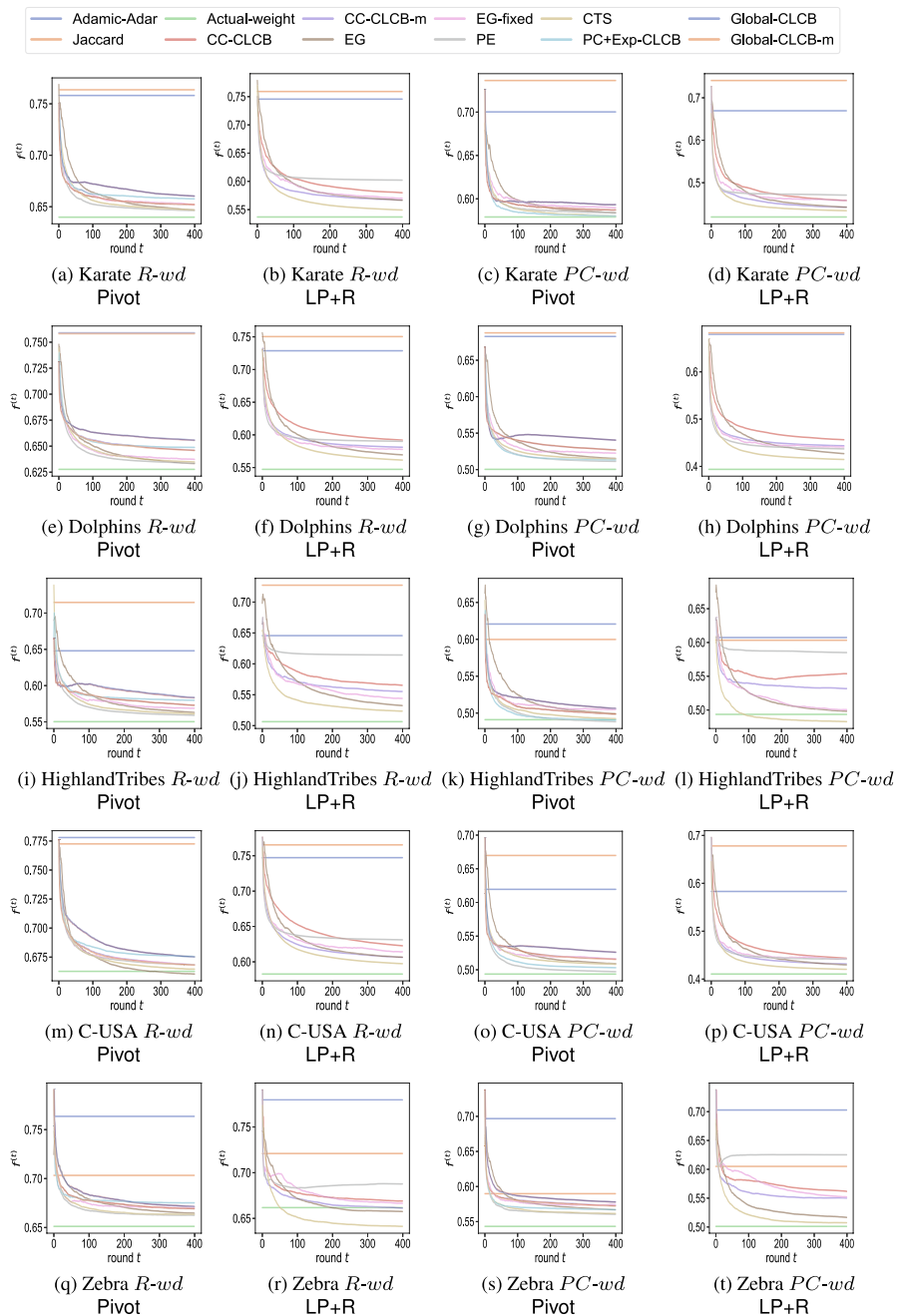


Fig. 3 Performance in terms of $f^{(t)}$ (Eq. 12), over a number $t = 1, \dots, 400$ of rounds (iterations), for the larger datasets, and R -wd and PC -wd weight distributions

Table 13 Coefficient of variation of $f^{(T)}$ (Eq. 12) over all the CMAB rounds and runs of the MIN-CC oracle, for R -wd weight setting

Method	Oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0.002	0.002	0.003	0.004	0.002
	LP+R	0.001	0.002	0.003	0.004	0.001
CC-CLCB-m	Pivot	0.004	0.002	0.004	0.005	0.001
	LP+R	0.004	0.003	0.003	0.005	0.002
PC+Exp-CLCB	Pivot	0.005	0.004	0.001	0.003	0.002
Global-CLCB	Pivot	0.002	0.002	0.003	0.002	0.002
Global-CLCB-m	Pivot	0.004	0.002	0.004	0.005	0.001
EG	Pivot	0.009	0.006	0.002	0.008	0.002
	LP+R	0.02	0.01	0.02	0.03	0.006
EG-fixed	Pivot	0.004	0.004	0.003	0.006	0.005
	LP+R	0.01	0.009	0.02	0.03	0.009
PE	Pivot	0.01	0.006	0.005	0.02	0.02
	LP+R	0.04	0.02	0.06	0.1	0.02
CTS	Pivot	0.003	0.002	0.002	0.008	0.001
	LP+R	0.004	0.003	0.01	0.02	0.004
Adamic-Adar	Pivot	0.008	0.02	0.004	0.04	0.03
	LP+R	0	0	0	0	0
Jaccard	Pivot	0.007	0.006	0	0.02	0.004
	LP+R	0	0	0	0	0
Actual-weight	Pivot	0.02	0.009	0.01	0.03	0.009
	LP+R	0	0.008	0	0.02	0

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	3E-05	0.0001	3E-06	3E-06	7E-06
CC-CLCB-m	3E-05	0.0001	5E-06	5E-06	3E-06
PC+Exp-CLCB	4E-05	0.0002	1E-05	5E-06	5E-06
Global-CLCB	3E-05	0.0001	3E-06	3E-06	5E-06
Global-CLCB-m	3E-05	0.0001	5E-06	5E-06	6E-06
EG	3E-05	0.0003	7E-05	4E-05	0.0007
EG-fixed	4E-05	0.0007	0.0001	7E-05	0.0008
PE	2E-05	0.0001	6E-06	7E-06	2E-05
CTS	2E-05	0.0001	1E-06	2E-06	6E-06
Adamic-Adar	0	0	3E-07	2E-06	4E-05
Jaccard	0.0001	0.0006	2E-07	2E-05	3E-05

Table 13 (continued)

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
Actual-weight	0.0003	0.0009	4E-05	3E-05	3E-05

oracle within CMAB correlation-clustering algorithms (where ε need not to be necessarily a constant; for instance, in the algorithms in Charikar et al. (2005); Demaine et al. (2006), ε is $\mathcal{O}(\log |V|)$). We focus on the MIN-CC context, thus on the notion of MIN-CC- (α, β) -approximation oracle (Definition 2). A similar reasoning holds for MAX-CC as well.

If $\varepsilon > 1$ is the expected approximation factor of a (randomized) MIN-CC algorithm, it means that, for every MIN-CC instance I , it holds that $\mathbb{E}[f_{\min}(\mathcal{C})] \leq \varepsilon f_{\min}(\mathcal{C}_I^*)$, where \mathcal{C} is the clustering output by the algorithm and \mathcal{C}_I^* is the optimal clustering for I . To have a MIN-CC- (α, β) -approximation oracle, we need to convert these guarantees—which hold in expectation, but not necessarily in every run—into guarantees that hold *in every run with a certain probability at least β* (with β typically in the order of $1 - (\text{poly}(|V|))^{-1}$).

More specifically, it is well-known that an algorithm with ε guarantees in expectation can be converted into an algorithm with $(1 + \delta)\varepsilon$ guarantees with high probability, for any $\delta > 0$, by exploiting Markov's inequality (Gupta 2005). In particular, running the algorithm with ε guarantees in expectation $k = c \cdot \log_{1+\delta}(|V|)$ times (for any $c > 0$) and keeping the best output among all of those trials, it yields a clustering with $(1 + \delta)\varepsilon$ quality guarantee with probability at least $1 - |V|^{-c}$. The aforementioned procedure corresponds to a MIN-CC- $(\alpha(\delta), \beta(c))$ -approximation oracle where $\alpha(\delta) = (1 + \delta)\varepsilon$ and $\beta(c) = 1 - |V|^{-c}$. Note that, given that δ is arbitrary, it means that there exist various MIN-CC- $(\alpha(\delta), \beta(c))$ -approximation oracles, each one corresponding to specific values of δ and c . Also, the worse the required approximation guarantee (i.e., higher δ , thus higher $\alpha(\delta)$), the higher the $\beta(c)$ success probability of the oracle (due to the relationship $k = c \log_{1+\delta} |V|$ between δ and c). By suitably choosing $\delta > 0$ and $c > 0$ (so as to define the number $k = c \log_{1+\delta} |V|$ of trials), one can get the desired $\alpha(\delta), \beta(c)$ values.

A.2: Algorithms for CMAB-MAX-CC

In this section, we present algorithms for CMAB-MAX-CC (Problem 4). When basic Chen et al.'s CMAB framework (Chen et al. 2016) is contextualized to a specific *maximization* problem, a major algorithmic contribution typically consists in adapting the so-called *Combinatorial Upper Confidence Bound* (CUCB) method to the context at hand, and showing how its theoretical guarantees are maintained/change while carrying out such an adaptation (Chen et al. 2018b; Liu et al. 2021; Mandaglio and Tagarelli 2019a, b; Talebi et al. 2017; Vaswani and Lakshmanan 2015; Wu et al. 2019). Here we follow that bulk of literature: we focus on the customization of

Table 14 Coefficient of variation of $f^{(T)}$ (Eq. 12) over all the CMAB rounds and runs of the MIN-CC oracle, for *PC-wd* weight setting

Method	Oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0.003	0.004	0.003	0.001	0.003
	LP+R	0.004	0.003	0.005	0.08	0.003
CC-CLCB-m	Pivot	0.002	0.003	0.003	0.001	0.004
	LP+R	0.007	0.006	0.006	0.1	0.002
PC+Exp-CLCB	Pivot	0.002	0.004	0.003	0.003	0.005
Global-CLCB	Pivot	0.003	0.004	0.004	0.001	0.003
Global-CLCB-m	Pivot	0.002	0.003	0.003	0.001	0.004
EG	Pivot	0.003	0.003	0.004	0.009	0.004
	LP+R	0.01	0.01	0.03	0.02	0.01
EG-fixed	Pivot	0.005	0.004	0.004	0.006	0.004
	LP+R	0.009	0.01	0.04	0.02	0.01
PE	Pivot	0.02	0.01	0.003	0.03	0.005
	LP+R	0.04	0.04	0.1	0.1	0.04
CTS	Pivot	0.004	0.003	0.004	0.009	0.002
	LP+R	0.004	0.01	0.02	0.02	0.003
Adamic-Adar	Pivot	0.03	0.009	0.003	0.03	0.06
	LP+R	0	0	0	0	0
Jaccard	Pivot	0.009	0.01	0	0.03	0.02
	LP+R	0	0	0	0	0
Actual-weight	Pivot	0.03	0.01	0.02	0.05	0.02
	LP+R	0.03	0	0.04	0.05	1E-16

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	3E-05	0.0002	9E-06	3E-06	1E-05
CC-CLCB-m	3E-05	0.0003	9E-06	5E-06	1E-05
PC+Exp-CLCB	3E-05	0.0003	1E-05	1E-05	1E-05
Global-CLCB	3E-05	0.0002	9E-06	3E-06	1E-05
Global-CLCB-m	4E-05	0.0003	9E-06	5E-06	1E-05
EG	4E-05	0.0003	0.0001	6E-05	0.001
EG-fixed	6E-05	0.001	0.0002	0.0001	0.001
PE	4E-05	0.0003	1E-05	7E-06	2E-05
CTS	2E-05	0.0002	2E-06	1E-06	4E-06
Adamic-Adar	0	0	2E-07	3E-06	7E-05
Jaccard	0.0002	0.001	8E-08	2E-05	3E-05
Actual-weight	0.0003	0.004	6E-05	5E-05	4E-05

Table 15 Coefficient of variation of $f^{(T)}$ (Eq. 12) over all the CMAB rounds and runs of the MIN-CC oracle, for *C-wd* weight setting

Method	Oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0.002	0.002	0.04	0.002	0.001
	LP+R	0	0	0.01	1E-16	0
CC-CLCB-m	Pivot	0.003	0.002	0.1	0.002	0.002
	LP+R	0	0	0.02	1E-16	0
PC+Exp-CLCB	Pivot	0.001	0.001	0.01	0.002	0.001
Global-CLCB	Pivot	0.002	0.002	0.04	0.002	0.001
Global-CLCB-m	Pivot	0.003	0.002	0.1	0.002	0.002
EG	Pivot	0.01	0.01	0.01	0.003	0.01
	LP+R	0.008	0.01	0.005	0.002	0.008
EG-fixed	Pivot	0.01	0.008	0.008	0.0006	0.01
	LP+R	0.005	0.009	0.009	0.003	0.008
PE	Pivot	0.002	0.001	0.01	0.002	0.001
	LP+R	0	0	0	1E-16	0
CTS	Pivot	0.003	0.004	0.01	0.002	0.003
	LP+R	0.007	0.01	0.002	0.007	0.03
Adamic-Adar	Pivot	0.07	0.05	0.02	0.1	0.06
	LP+R	0	0	0	1E-16	0
Jaccard	Pivot	0.02	0.008	0	0.02	0.009
	LP+R	0	0	0	1E-16	1E-16
Actual-weight	Pivot	0.03	0.03	0.1	0.03	0.04
	LP+R	0	0	0	1E-16	0

CUCB to the MAX-CC context, and show that the theoretical guarantees of CUCB carry over rather easily to this customization.

The CC-CUCB algorithm. CUCB (Chen et al. 2016) is an extension of the UCB1 method for MAB (Auer et al. 2002). It keeps, along with the estimate of the means of the base-arm random variables, confidence intervals within which the true means fall with overwhelming probability, and plays superarms based on the upper bound of those intervals.

Table 16 Coefficient of variation of $ren^{(T)}$ (Eq. 13) over all the CMAB rounds and runs of the MIN-CC oracle, for R - wd weight setting

Method	Oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0.02	0.03	0.04	0.2	0.02
	LP+R	0.1	0.08	0.2	0.06	0.3
CC-CLCB-m	Pivot	0.08	0.1	0.07	0.3	0.04
	LP+R	0.1	0.1	0.08	0.2	0.1
PC+Exp-CLCB	Pivot	0.2	0.1	0.4	0.3	0.2
Global-CLCB	Pivot	0.02	0.02	0.03	0.2	0.03
Global-CLCB-m	Pivot	0.08	0.1	0.07	0.3	0.04
EG	Pivot	0.02	0.2	0.1	0.2	0.2
	LP+R	0.2	0.08	0.07	0.2	0.07
EG-fixed	Pivot	0.1	0.09	0.1	0.1	0.2
	LP+R	0.07	0.06	0.1	0.1	0.09
PE	Pivot	0.1	0.08	0.3	0.2	0.1
	LP+R	0.08	0.03	0.1	0.2	0.07
CTS	Pivot	0.1	0.04	0.3	0.09	0.08
	LP+R	0.08	0.07	0.1	0.1	0.04
Adamic-Adar	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Jaccard	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Actual-weight	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	0.0008	0.007	0.002	0.02	0.002
CC-CLCB-m	0	0.008	0.002	0.02	0.002
PC+Exp-CLCB	0.001	0.006	0.0004	0.01	0.001
Global-CLCB	0.0008	0.007	0.002	0.02	0.002
Global-CLCB-m	0	0.008	0.002	0.02	0.002
EG	0.004	0.01	0.02	0.02	0.04
EG-fixed	0.006	0.02	0.02	0.03	0.03
PE	0.003	0.02	0.001	0.01	0.0007
CTS	0.004	0.004	0.0006	0.007	0.0009
Adamic-Adar	0	0	0	0	0
Jaccard	0	0	0	0	0
Actual-weight	0	0	0	0	0

Algorithm 4 CC-CUCB

Input: A graph $G = (V, E)$; an integer $T > 0$; an oracle \mathbf{O} for MAX-CC

Output: A clustering \mathcal{C}_t of G , for all $t = 1, \dots, T$

```

1: initialize  $\hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}$  and  $\hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}$ ;  $\forall e \in E: T_e^+ \leftarrow 0, T_e^- \leftarrow 0$ 
2: for  $t = 1, \dots, T$  do
3:    $\forall e \in E: \rho_e^+ \leftarrow \sqrt{\frac{3 \ln t}{2T_e^+}}, \rho_e^- \leftarrow \sqrt{\frac{3 \ln t}{2T_e^-}}$  ( $\rho_e^+ = 0$ , if  $T_e^+ = 0$ ;  $\rho_e^- = 0$ , if  $T_e^- = 0$ )
4:    $\forall e \in E: \tilde{\mu}_e^+ \leftarrow \min\{\hat{\mu}_e^+ + \rho_e^+, 1\}, \tilde{\mu}_e^- \leftarrow \min\{\hat{\mu}_e^- + \rho_e^-, 1\}$ 
5:    $\mathcal{C}_t \leftarrow$  run  $\mathbf{O}$  on input  $(G, \{\tilde{\mu}_e^+\}_{e \in E}, \{\tilde{\mu}_e^-\}_{e \in E})$ 
6:   for  $e = (u, v) \in E \mid \mathcal{C}_t(u) = \mathcal{C}_t(v)$  do
7:     observe feedback  $w^+ \sim W_e^+$ ;  $\hat{\mu}_e^+ \leftarrow (\hat{\mu}_e^+ T_e^+ + w^+) / (T_e^+ + 1)$ ;  $T_e^+ \leftarrow T_e^+ + 1$ ;
8:   end for
9:   for  $e = (u, v) \in E \mid \mathcal{C}_t(u) \neq \mathcal{C}_t(v)$  do
10:    observe feedback  $w^- \sim W_e^-$ ;  $\hat{\mu}_e^- \leftarrow (\hat{\mu}_e^- T_e^- + w^-) / (T_e^- + 1)$ ;  $T_e^- \leftarrow T_e^- + 1$ ;
11:  end for
12: end for

```

Our customization of CUCB to MAX-CC is termed **CC-CUCB** and outlined as Algorithm 4. **CC-CUCB** keeps track of the mean estimates $\hat{\mu} = \{\hat{\mu}^+, \hat{\mu}^-\}$ (Eq. 4), and of the number T_e^+ (resp. T_e^-) of times a sample from W_e^+ (resp. W_e^-) random variable has been observed until the current round, for all $e \in E$. At the beginning, $\forall e \in E: T_e^+ = T_e^- = 0$, and $\hat{\mu}$ are initialized, e.g., randomly or based on prior domain knowledge (Line 1). In every round t , the current mean estimates are adjusted with a term ρ_e^\pm (defined based on Chernoff-Hoeffding bounds (Auer et al. 2002; Chen et al. 2016)), so as to foster, to some extent, the exploration of less often played base arms (Line 3). This leads to the adjusted means $\{\tilde{\mu}_e^+, \tilde{\mu}_e^-\}_{e \in E}$ (Line 4), which are interpreted as positive-type and negative-type edge weights of a correlation-clustering instance, respectively, and are fed as input (along with G) to an oracle \mathbf{O} that computes a MAX-CC solution \mathcal{C}_t (Line 5). \mathcal{C}_t is used as a feedback to update the mean estimates (Sect. 3, Table 1). Specifically, the weight of each intra-cluster (resp. inter-cluster) edge e is interpreted as a sample of W_e^+ (resp. W_e^-), and is used to update $\hat{\mu}_e^+, T_e^+$ (resp. $\hat{\mu}_e^-, T_e^-$). $\hat{\mu}_e^+$ and $\hat{\mu}_e^-$ are updated so as to be equal to the average of the samples from W_e^+ and W_e^- observed so far, respectively (Lines 6–11).

A.2.1: Regret analysis of CC-CUCB

As correlation clustering is **NP-hard**, it is unlikely that **CC-CUCB** can be equipped with an *exact* oracle \mathbf{O} for MAX-CC running in polynomial time. Hence, in analyzing the theoretical guarantees of **CC-CUCB**, we consider the case where \mathbf{O} is a MAX-CC- (α, β) -approximation oracle:

Definition 5 (MAX-CC- (α, β) -approximation oracle) Given a MAX-CC instance $I = \langle (V, E), \{(w_e^+, w_e^-)\}_{e \in E} \rangle$, let \mathcal{C}_I^* be the optimal solution to I . Given $\alpha, \beta \in (0, 1]$, an

Table 17 Coefficient of variation of $ren^{(T)}$ (Eq. 13) over all the CMAB rounds and runs of the MIN-CC oracle, for *PC-wd* weight setting

Method	Oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0.008	0.02	0.05	0.07	0.08
	LP+R	0.09	0.2	0.01	0.5	0.05
CC-CLCB-m	Pivot	0.1	0.1	0.4	0.1	0.1
	LP+R	0.05	0.02	0.1	1	0.09
PC+Exp-CLCB	Pivot	0.05	0.07	0.1	0.05	0.04
Global-CLCB	Pivot	0.008	0.02	0.05	0.2	0.08
Global-CLCB-m	Pivot	0.1	0.1	0.4	0.1	0.1
EG	Pivot	0.2	0.1	0.1	0.2	0.05
	LP+R	0.2	0.09	0.07	0.1	0.2
EG-fixed	Pivot	0.07	0.2	0.04	0.2	0.04
	LP+R	0.1	0.1	0.1	0.09	0.09
PE	Pivot	0.2	0.1	0.3	0.1	0.1
	LP+R	0.1	0.06	0.07	0.3	0.1
CTS	Pivot	0.08	0.04	0.05	0.1	0.1
	LP+R	0.08	0.07	0.1	0.08	0.05
Adamic-Adar	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Jaccard	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Actual-weight	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0

Method	Last.fm	Primary-School	Prosper-Loans	Wikipedia	DBLP
CC-CLCB	0.0008	0.01	0.002	0.02	0.002
CC-CLCB-m	0.0009	0.009	0.002	0.02	0.002
PC+Exp-CLCB	0.002	0.002	0.0007	0.02	0.0009
Global-CLCB	0.0008	0.01	0.002	0.02	0.002
Global-CLCB-m	0.0009	0.009	0.002	0.02	0.002
EG	0.005	0.02	0.02	0.03	0.04
EG-fixed	0.006	0.02	0.02	0.03	0.03
PE	0.004	0.01	0.002	0.01	0.002
CTS	0.003	0.007	0.0007	0.006	0.0006
Adamic-Adar	0	0	0	0	0
Jaccard	0	0	0	0	0
Actual-weight	0	0	0	0	0

Table 18 Coefficient of variation of $ren^{(T)}$ (Eq. 13) over all the CMAB rounds and runs of the MIN-CC oracle, for $C-wd$ weight setting

Method	oracle	Karate	Dolphins	Zebra	HighlandTribes	Contiguous-USA
CC-CLCB	Pivot	0	1E-16	0.005	0	0
	LP+R	0	1E-16	0.2	0	0
CC-CLCB-m	Pivot	0	1E-16	0.006	0	0
	LP+R	0	1E-16	0	0	0
PC+Exp-CLCB	Pivot	0	0	0	0	0
Global-CLCB	Pivot	0	1E-16	0.005	0	0
Global-CLCB-m	Pivot	0	1E-16	0.006	0	0
EG	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
EG-fixed	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
PE	Pivot	0.03	0.0005	0.008	0	0.0002
	LP+R	0	1E-16	0	0	0
CTS	Pivot	0.01	0.01	0.03	0.1	0.009
	LP+R	0.003	0.002	0.005	0.02	0.002
Adamic-Adar	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Jaccard	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0
Actual-weight	Pivot	0	0	0	0	0
	LP+R	0	0	0	0	0

algorithm for MAX-CC is a $\max(\alpha, \beta)$ -approximation oracle if, for every input I , it yields a solution \mathcal{C} such that $\Pr[f_{\max}(\mathcal{C}) \geq \alpha f_{\max}(C_I^*)] \geq \beta$ (where $f_{\max}(\cdot)$ is MAX-CC's objective function, Eq. (2)).

When used as an oracle \mathbf{O} within CC-CUCB, the condition in Definition 5 to recognize \mathbf{O} as a MAX-CC- (α, β) -approximation oracle needs to hold on every MAX-CC instance that is given as input to \mathbf{O} at each round. Hence, the condition has to hold on the *mean estimates*, not the true ones. Existing algorithms for MAX-CC achieving constant-factor guarantees *in expectation* (Charikar et al. 2005; Swamy 2004) can be employed as MAX-CC- (α, β) -approximation oracles. We show the details of this in Appendix A.1.

When approximation oracles are used, the quality of a CMAB algorithm is typically measured in terms of the (α, β) -approximation regret metric, which is defined as the $\alpha\beta$ fraction of the expected reward of playing the best superarm in every round, minus the sum of expected reward of the superarms played by the algorithm in all the rounds (Chen et al. 2016). In the CMAB-MAX-CC setting, this metric becomes:

Definition 6 (MAX-CC-(α, β)-approximation regret) Let \mathcal{C}_t^* be the clustering maximizing $\bar{a}_\mu(\cdot)$ (Eq. 7) on a CMAB-MAX-CC instance I (w.r.t. the true μ means, Eq. (3)), and let $\{\mathcal{C}_t\}_{t=1}^T$ be the clusterings output by an algorithm **A** run on I . For any $\alpha, \beta \in (0, 1]$, the MAX-CC-(α, β)-approximation regret of **A** is

$$Reg_{\mu, \alpha, \beta}^A(T) = T \alpha \beta \bar{a}_\mu(\mathcal{C}_1^*) - \mathbb{E} \left[\sum_{t=1}^T \bar{a}_\mu(\mathcal{C}_t) \right]. \tag{15}$$

Chen et al. (2016); Wang and Chen (2017) show that, if the (expected) reward function satisfies certain properties (see Appendix A.2.2), the CUCB method indeed achieves a regret at most in the order of $\mathcal{O}(\log T)$. Here we show that this guarantee carries over to our CC-CUCB:

Theorem 4 Given $\alpha, \beta \in (0, 1]$, the MAX-CC-(α, β)-approximation regret (Definition 6) of the CC-CUCB algorithm (Alg. 4), when equipped with a MAX-CC-(α, β)-approximation oracle **O** (Definition 5), is upper-bounded by a function that is $\mathcal{O}(\log T)$.

Proof (sketch) The expected reward $\bar{a}_\mu(\cdot)$ used in CC-CUCB satisfies all the properties that Chen et al. (2016); Wang and Chen (2017) require to have a CUCB-like method achieving a logarithmic (in the number of rounds) regret bound. A detailed proof is shown in Appendix A.2.2. \square

The guarantees in Theorem 4 are on the regret computed based on the true μ means, despite the guarantees of the oracle are on the $\hat{\mu}$ estimates. This is possible thanks to the way how CC-CUCB computes the mean estimates and the properties of the reward function required by the theorem. More details on this are reported in Appendix A.2.2.

A.2.2: Proof of Theorem 4

Definition 7 (Base arms induced by a clustering (CMAB-MAX-CC)) Let \mathcal{S} be the set of all base arms and \mathcal{C} a clustering, we denote with $\mathcal{S}_\mathcal{C}$ the set of base arms corresponding to the clustering-compliant replica set induced by the clustering \mathcal{C} , i.e., $\mathcal{S}_\mathcal{C} = \{\mu_{uv}^+ \mid e = (u, v) \in e^{in}\} \cup \{\mu_{uv}^- \mid e = (u, v) \in e^{out}\}$.

Definition 8 (Bad superarm (CMAB-MAX-CC) (Wang and Chen 2017)) Let \mathcal{C}_1^* be the clustering minimizing the expected reward $\bar{a}_\mu(\cdot)$ (Eq. 7) on a CMAB-MAX-CC instance I (w.r.t. the true μ means, Eq. 3). Given a MAX-CC-(α, β)-approximation oracle, a superarm (output of the oracle) \mathcal{C} is bad if $\bar{a}_\mu(\mathcal{C}) < \alpha \bar{a}_\mu(\mathcal{C}_1^*)$.

Definition 9 (Gap (CMAB-MAX-CC) (Wang and Chen 2017)) Given a superarm \mathcal{C} , the gap $\Delta_\mathcal{C}$ of \mathcal{C} is defined as $\Delta_\mathcal{C} = \max\{0, \alpha \bar{a}_\mu(\mathcal{C}_1^*) - \bar{a}_\mu(\mathcal{C})\}$, where \mathcal{C}_1^* is the optimal solution of the MAX-CC instance at hand. Moreover, for a base arm $p \in \mathcal{S}$, we define $\Delta_{min}^p = \min_{\mathcal{C} \mid p \in \mathcal{S}_\mathcal{C}, \Delta_\mathcal{C} > 0} \Delta_\mathcal{C}$ and $\Delta_{max}^p = \max_{\mathcal{C} \mid p \in \mathcal{S}_\mathcal{C}, \Delta_\mathcal{C} > 0} \Delta_\mathcal{C}$. As a convention, if for

a base arm p there is no superarm \mathcal{C} such that $p \in \mathcal{S}_{\mathcal{C}}$ and $\Delta_{\mathcal{C}} > 0$ then $\Delta_{min}^p = +\infty$ and $\Delta_{max}^p = 0$. Also let $\Delta_{max} = \max_{p \in \mathcal{S}} \Delta_{max}^p$ and $\Delta_{min} = \min_{p \in \mathcal{S}} \Delta_{min}^p$.

Property 1 (Monotonicity (CMAB-MAX-CC)) (Wang and Chen 2017)) *The reward function is monotonically non-decreasing w.r.t. the expectations, i.e., for any two expectation vectors $\mu = (\mu_1, \dots, \mu_m)$ and $\mu' = (\mu'_1, \dots, \mu'_m)$, and for any clustering \mathcal{C} , it holds that $\bar{a}_{\mu}(\mathcal{C}) \leq \bar{a}_{\mu'}(\mathcal{C})$, if $\mu_i \leq \mu'_i$, for all $i = 1, \dots, m$.*

Property 2 (1-Norm Bounded Smoothness (CMAB-MAX-CC)) (Wang and Chen 2017)) *There exists a bounded smoothness constant $B \in \mathbb{R}^+$ such that for any two expectation vectors μ and μ' , and for any clustering \mathcal{C} , it holds that $|\bar{a}_{\mu}(\mathcal{C}) - \bar{a}_{\mu'}(\mathcal{C})| \leq B \sum_{p \in \mathcal{S}_{\mathcal{C}}} |\mu_p - \mu'_p|$.*

Lemma 1 *Property 1 holds for $\bar{a}_{\mu}(\cdot)$ (Eq. 8).*

Proof Straightforward because $\bar{a}_{\mu}(\cdot)$ is defined as a sum of the expectation values. □

Lemma 2 *Property 2 with $B = 1$ holds for $\bar{a}_{\mu}(\cdot)$ (Eq. 8).*

$$|\bar{a}_{\mu}(\mathcal{C}) - \bar{a}_{\mu'}(\mathcal{C})| = \left| \sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) = \mathcal{C}(v)}} (\mu_{uv}^+ - \mu'_{uv}^+) + \sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) \neq \mathcal{C}(v)}} (\mu_{uv}^- - \mu'_{uv}^-) \right| \leq \sum_{p \in \mathcal{S}_{\mathcal{C}}} |\mu_p - \mu'_p|.$$

Proof □

Theorem 5 *Given $\alpha, \beta \in (0, 1]$, the MAX-CC-(α, β)-approximation regret (Definition 6) of the CC-CUCB algorithm (Alg. 4), when equipped with a MAX-CC-(α, β)-approximation oracle \mathbf{O} (Definition 5), is upper-bounded as follows:*

$$Reg_{\mu, \alpha, \beta}^A(T) \leq \frac{2}{3} \pi^2 |E| \Delta_{max} + 4|E| + \sum_{p \in \mathcal{S}} \frac{48|E| \ln T}{\Delta_{min}^p}. \tag{16}$$

Proof Theorem 4 in the Appendix of Wang and Chen (2017) states that the CUCB method achieves a regret bound of the form in Eq. (16) if the expected reward satisfies Property 1 and Property 2. According to Lemmas 1 and 2, the expected reward $\bar{a}_{\mu}(\cdot)$ (Eq. 8) used in CMAB-MAX-CC satisfies both those properties. □

A.3: Regret Analysis of CC-CLCB (Proof of Theorem 1)

Next we provide several definitions which, sometimes with minor modifications, are taken from Wang and Chen (2017).

Property 3 (Monotonicity (CMAB-MIN-CC)) (Wang and Chen 2017)) *The loss function is monotonically non-decreasing w.r.t. the expectations, i.e., for any two expectation vectors $\mu = (\mu_1, \dots, \mu_m)$ and $\mu' = (\mu'_1, \dots, \mu'_m)$, and for any clustering \mathcal{C} , it holds that $\bar{d}_\mu(\mathcal{C}) \leq \bar{d}_{\mu'}(\mathcal{C})$, if $\mu_i \leq \mu'_i$, for all $i = 1, \dots, m$.*

Property 4 (1-Norm Bounded Smoothness (CMAB-MIN-CC)) (Wang and Chen 2017)) *There exists a bounded smoothness constant $B \in \mathbb{R}^+$ such that for any two expectation vectors μ and μ' , and for any clustering \mathcal{C} , it holds that $|\bar{d}_\mu(\mathcal{C}) - \bar{d}_{\mu'}(\mathcal{C})| \leq B \sum_{p \in \mathcal{S}_\mathcal{C}} |\mu_p - \mu'_p|$.*

Lemma 3 *Property 3 holds for $\bar{d}_\mu(\cdot)$ (Eq. 8).*

Proof Straightforward because $\bar{d}_\mu(\cdot)$ is defined as a sum of the expectation values. □

Lemma 4 *Property 4 with $B = 1$ holds for $\bar{d}_\mu(\cdot)$ (Eq. 8).*

Proof

$$|\bar{d}_\mu(\mathcal{C}) - \bar{d}_{\mu'}(\mathcal{C})| = \left| \sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) = \mathcal{C}(v)}} (\mu_{uv}^- - \mu'_{uv}^-) + \sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) \neq \mathcal{C}(v)}} (\mu_{uv}^+ - \mu'_{uv}^+) \right| \leq \sum_{p \in \mathcal{S}_\mathcal{C}} |\mu_p - \mu'_p|. \quad \square$$

Definition 10 (Base arms induced by a clustering (CMAB-MIN-CC)) Let \mathcal{S} be the set of all base arms and \mathcal{C} a clustering, we denote with $\mathcal{S}_\mathcal{C}$ the set of base arms corresponding to the clustering-compliant replica set induced by the clustering \mathcal{C} , i.e., $\mathcal{S}_\mathcal{C} = \{\mu_{uv}^- \mid e = (u, v) \in e^{in}\} \cup \{\mu_{uv}^+ \mid e = (u, v) \in e^{out}\}$.

Definition 11 (Bad super arm (CMAB-MIN-CC)) (Wang and Chen 2017)) Let \mathcal{C}_I^* be the clustering minimizing the expected loss $\bar{d}_\mu(\cdot)$ (Eq. 7) on a CMAB-MIN-CC instance I (w.r.t. the true μ means, Eq. 3). Given a MIN-CC- (α, β) -approximation oracle for MIN-CC, a super arm (output of the oracle) \mathcal{C} is bad if $\bar{d}_\mu(\mathcal{C}) > \frac{1}{\alpha} \cdot \bar{d}_\mu(\mathcal{C}_I^*)$.

Definition 12 (Gap (CMAB-MIN-CC)) (Wang and Chen 2017)) For a CMAB-MIN-CC instance, for a given super arm \mathcal{C} we define the gap as $\Delta_\mathcal{C} = \max\{0, \bar{d}_\mu(\mathcal{C}) - \frac{1}{\alpha} \cdot \bar{d}_\mu(\mathcal{C}_I^*)\}$, where \mathcal{C}_I^* is the optimal solution of the MIN-CC instance at hand. For each base arm $p \in \mathcal{S}$ we define $\Delta_{min}^p = \min_{\mathcal{C} \mid p \in \mathcal{S}_\mathcal{C}, \Delta_\mathcal{C} > 0} \Delta_\mathcal{C}$ and $\Delta_{max}^p = \max_{\mathcal{C} \mid p \in \mathcal{S}_\mathcal{C}, \Delta_\mathcal{C} > 0} \Delta_\mathcal{C}$. As a convention, if for a base arm p there is no super arm \mathcal{C} such that $p \in \mathcal{S}_\mathcal{C}$ and $\Delta_\mathcal{C} > 0$ then $\Delta_{min}^p = +\infty$ and $\Delta_{max}^p = 0$. Also let $\Delta_{max} = \max_{p \in \mathcal{S}} \Delta_{max}^p$ and $\Delta_{min} = \min_{p \in \mathcal{S}} \Delta_{min}^p$.

Similarly to Wang and Chen (2017), we introduce the following definition to assist our analysis.

Definition 13 (*Event-filtered regret*) Let C_I^* be the clustering minimizing the expected loss $\bar{d}_\mu(\cdot)$ (Eq. 7) on a CMAB-MIN-CC instance I (w.r.t. the true μ means, Eq. 3) and let $\{C_t\}_{t=1}^T$ be the clusterings output by an algorithm **A** run on I . For any series of events $\{\mathcal{E}\}_{t \geq 1}$ indexed by round number t , we define $Reg_{\mu,\alpha}^A(T, \{\mathcal{E}\}_{t \geq 1})$ as the regret filtered by events $\{\mathcal{E}\}_{t \geq 1}$, that is, regret is only counted in round t if \mathcal{E}_t happens in round t . Formally

$$Reg_{\mu,\alpha}^A(T, \{\mathcal{E}\}_{t \geq 1}) = \mathbb{E} \left[\sum_{t=1}^T \mathbb{1}\{\mathcal{E}_t\} \left(\bar{d}_\mu(C_t) - \frac{1}{\alpha} \cdot \bar{d}_\mu(C_I^*) \right) \right].$$

Theorem 1 Given $\alpha, \beta \in (0, 1]$, the MIN-CC- (α, β) -approximation regret (Definition 3) of the CC-CLCB algorithm (Alg. 1), when equipped with a MIN-CC- (α, β) -approximation oracle **O** (Definition 2), is upper-bounded as follows:

$$Reg_{\mu,\alpha,\beta}^A(T) \leq \frac{2}{3} \pi^2 |E| \Delta_{max} + 4|E| + \sum_{p \in \mathcal{S}} \frac{48|E| \ln T}{A_{min}^p}. \tag{17}$$

Proof The proof is very similar to the proof of Theorem 4 in Wang and Chen (2017) (Section B.2 in the Supplementary Materials), thus, here we avoid replicating all the steps. Although the adopted (event-filtered) regret definition (Definition 3) in our case is different from the one considered in Wang and Chen (2017), the derived regret bound is the same. Our proof relies on the following main result: the $\bar{d}_\mu(\cdot)$ function (Eq. 8) satisfies the properties of monotonicity (Lemma 3) and 1-norm bounded smoothness (Lemma 4). Other technical differences between our proof and the one in Wang and Chen (2017) correspond to (1) changing, in some intermediate steps, inequality signs since rewards are replaced with losses, and (2) using the argument that, for the considered regret definition, it holds that $\Delta_{max} \leq \mathcal{M} - \frac{1}{\alpha} \cdot \bar{d}_\mu(C_I^*)$. □

A.4: Details on the symmetric-distribution setting (Sect. 4.3)

Algorithm 5 Full-exploitation strategy (for symmetric distributions)

Input: A graph $G = (V, E)$; an integer $T > 0$; an oracle \mathbf{O} for MIN-CC

Output: A clustering \mathcal{C}_t of G , for all $t = 1, \dots, T$

- 1: initialize $\hat{\mu}^+ = \{\hat{\mu}_e^+\}_{e \in E}$ and $\hat{\mu}^- = \{\hat{\mu}_e^-\}_{e \in E}$; $\forall e \in E: T_e^+ \leftarrow 0, T_e^- \leftarrow 0$
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: $\mathcal{C}_t \leftarrow$ run \mathbf{O} on input $\langle G, \{\hat{\mu}_e^+\}_{e \in E}, \{\hat{\mu}_e^-\}_{e \in E} \rangle$
 - 4: **for** $e = (u, v) \in E \mid \mathcal{C}_t(u) = \mathcal{C}_t(v)$ **do**
 - 5: observe feedback $w^- \sim W_e^-$
 - 6: $\hat{\mu}_e^- \leftarrow (\hat{\mu}_e^- T_e^- + w^-)/(T_e^- + 1)$; $\hat{\mu}_e^+ \leftarrow (\hat{\mu}_e^+ T_e^+ + (1 - w^-))/(T_e^+ + 1)$
 - 7: **end for**
 - 8: **for** $e = (u, v) \in E \mid \mathcal{C}_t(u) \neq \mathcal{C}_t(v)$ **do**
 - 9: observe feedback $w^+ \sim W_e^+$
 - 10: $\hat{\mu}_e^+ \leftarrow (\hat{\mu}_e^+ T_e^+ + w^+)/(T_e^+ + 1)$; $\hat{\mu}_e^- \leftarrow (\hat{\mu}_e^- T_e^- + (1 - w^+))/(T_e^- + 1)$
 - 11: **end for**
 - 12: $T_e^+ \leftarrow T_e^+ + 1$; $T_e^- \leftarrow T_e^- + 1$
 - 13: **end for**
-

Lemma 5 (Hoeffding’s inequality (Hoeffding 1963)) *Let X_1, \dots, X_n be independent random variables, where each X_i is bounded in $[a_i, b_i]$. Define $\hat{\mu} = (X_1 + \dots + X_n)/n$ the empirical mean of these variables and let $\mu = \mathbb{E}[\hat{\mu}]$ the expected value of these mean, it holds that:*

$$Pr\left[|\hat{\mu} - \mu| \geq \delta\right] \leq 2 \exp\left(-\frac{2n^2\delta^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Theorem 3 *Given $\alpha, \beta \in (0, 1]$, the MIN-CC-(α, β)-approximation regret (Definition 3) of a full-exploitation strategy \mathbf{A} run on a CMAB-MIN-CC instance where all edge-weight distributions are symmetric, and equipped with a MIN-CC-(α, β)-approximation oracle (Definition 2), is upper-bounded as follows:*

$$Reg_{\mu, \alpha, \beta}^{\mathbf{A}}(T) \leq \left(1 + \frac{2(\frac{1}{\alpha} + 1)^2 |E|^3}{\Delta_{min}^2}\right) \Delta_{max}. \tag{18}$$

Proof Let \mathcal{F}_t be the event that the MIN-CC-(α, β)-approximation oracle fails to produce an α -approximate answer with respect to its input.

Following the reasoning of Chen et al. (2018b) (Appendix D.5), we decompose the filtered regret (Definition 13) with null event as follows:

$$Reg(\{\}) = Reg(\mathcal{F}_t) + Reg(\neg\mathcal{F}_t)$$

It can be easily shown Chen et al. (2018b) that $Reg(\mathcal{F}_t) \leq (1 - \beta)T\Delta_{max}$. Let \mathcal{C}_{μ}^* (resp. $\mathcal{C}_{\hat{\mu}_t}^*$) be the clustering minimizing the expected loss $\bar{d}_{\mu}(\cdot)$ (Eq. 7) w.r.t. the true

μ means (resp. w.r.t. the estimates at round t , denoted as $\hat{\mu}_t$). In order to bound $Reg(\neg\mathcal{F}_t)$, we first bound Δ_{C_t} when $\neg\mathcal{F}_t$ holds:

$$\begin{aligned} \Delta_{C_t} &= \max \left\{ 0, \bar{d}_\mu(C_t) - \frac{1}{\alpha} \bar{d}_\mu(C_\mu^*) \right\} \\ &= \max \left\{ 0, \bar{d}_\mu(C_t) + \bar{d}_{\hat{\mu}_t}(C_t) - \bar{d}_{\hat{\mu}_t}(C_t) - \frac{1}{\alpha} \bar{d}_\mu(C_\mu^*) \right\} \\ &\leq \max \left\{ 0, \bar{d}_\mu(C_t) + \frac{1}{\alpha} \bar{d}_{\hat{\mu}_t}(C_\mu^*) - \bar{d}_{\hat{\mu}_t}(C_t) - \frac{1}{\alpha} \bar{d}_\mu(C_\mu^*) \right\} \\ (\neg\mathcal{F}_t \Rightarrow \bar{d}_{\hat{\mu}_t}(C_t) &\leq \frac{1}{\alpha} \bar{d}_{\hat{\mu}_t}(C_\mu^*) \leq \frac{1}{\alpha} \bar{d}_{\hat{\mu}_t}(C_\mu^*)) \\ &\leq \max \left\{ 0, \left| \frac{1}{\alpha} \bar{d}_{\hat{\mu}_t}(C_\mu^*) - \frac{1}{\alpha} \bar{d}_\mu(C_\mu^*) \right| + |\bar{d}_\mu(C_t) - \bar{d}_{\hat{\mu}_t}(C_t)| \right\} \\ &\leq \frac{1}{\alpha} |\bar{d}_{\hat{\mu}_t}(C_\mu^*) - \bar{d}_\mu(C_\mu^*)| + |\bar{d}_\mu(C_t) - \bar{d}_{\hat{\mu}_t}(C_t)| \quad (1 - \text{Norm bounded smoothness of } \bar{d}_\mu(\cdot)) \\ &\leq \frac{1}{\alpha} \left(\sum_{\substack{(u,v) \in E, \\ C_\mu^*(u) = C_\mu^*(v)}} |\hat{\mu}_{uv,t}^- - \mu_{uv}^-| + \sum_{\substack{(u,v) \in E, \\ C_\mu^*(u) \neq C_\mu^*(v)}} |\hat{\mu}_{uv,t}^+ - \mu_{uv}^+| \right) + \left(\sum_{\substack{(u,v) \in E, \\ C_t(u) = C_t(v)}} |\hat{\mu}_{uv,t}^- - \mu_{uv}^-| + \sum_{\substack{(u,v) \in E, \\ C_t(u) \neq C_t(v)}} |\hat{\mu}_{uv,t}^+ - \mu_{uv}^+| \right). \end{aligned}$$

If $|\hat{\mu}_{uv,t}^\pm - \mu_{uv}^\pm| < \Delta_{min}/(\frac{1}{\alpha} + 1)|E|$ for each $(u, v) \in E$ (event denoted with B_t), then $\Delta_{C_t} < \Delta_{min}$, and, thus, $\Delta_{C_t} = 0$ by definition of Δ_{min} . This means that, when $\neg\mathcal{F}_t$ holds, Δ_{C_t} can be non-zero only when there is at least one pair of vertices for which $|\hat{\mu}_{uv,t}^\pm - \mu_{uv}^\pm| \geq \Delta_{min}/(\frac{1}{\alpha} + 1)|E|$, i.e., $\neg B_t$ holds. In other words, $Reg(\neg\mathcal{F}_t) = Reg(\neg\mathcal{F}_t \wedge B_t) + Reg(\neg\mathcal{F}_t \wedge \neg B_t)$, with $Reg(\neg\mathcal{F}_t \wedge B_t) = 0$. Next we bound $Reg(\neg\mathcal{F}_t \wedge \neg B_t)$:

$$\begin{aligned} Reg(\neg\mathcal{F}_t \wedge \neg B_t) &\leq \sum_{t=1}^T Pr[\neg\mathcal{F}_t \wedge \neg B_t] \Delta_{max} \\ &\leq \sum_{t=1}^T Pr[\neg B_t] \Delta_{max} \quad \left(\text{since } Pr[\neg\mathcal{F}_t \wedge \neg B_t] \leq Pr[\neg B_t] \right) \\ &\leq \sum_{t=1}^T \Delta_{max} \sum_{(u,v) \in E} \left(Pr\left[|\hat{\mu}_{uv,t}^+ - \mu_{uv}^-| \geq \frac{\Delta_{min}}{(\frac{1}{\alpha} + 1)|E|} \right] + Pr\left[|\hat{\mu}_{uv,t}^- - \mu_{uv}^-| \geq \frac{\Delta_{min}}{(\frac{1}{\alpha} + 1)|E|} \right] \right) \end{aligned}$$

$$\begin{aligned}
 &\leq \sum_{t=1}^T \Delta_{max} \sum_{(u,v) \in E} \left(2e^{-2T^+_{uv,t-1} \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} + 2e^{-2T^-_{uv,t} \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} \right) \quad (\text{Hoeffding's inequality}) \\
 &= \sum_{t=1}^T \Delta_{max} \sum_{(u,v) \in E} \left(2e^{-2(t-1) \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} + 2e^{-2(t-1) \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} \right) \quad (\text{full-information setting : } T_{uv,t-1}^{\pm} = t-1) \\
 &\leq \Delta_{max} + \sum_{(u,v) \in E} \Delta_{max} \sum_{t=2}^T \left(2e^{-2(t-1) \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} + 2e^{-2(t-1) \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} \right) \\
 &= \Delta_{max} + 4|E|\Delta_{max} \sum_{t=2}^T e^{-2(t-1) \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} \leq \Delta_{max} + 4|E|\Delta_{max} \int_{t=0}^{\infty} e^{-2t \frac{\Delta_{min}^2}{(\frac{1}{\alpha}+1)^2|E|^2}} dt \\
 &= \Delta_{max} + \frac{2(\frac{1}{\alpha} + 1)^2|E|^3}{\Delta_{min}^2} \Delta_{max} = \left(1 + \frac{2(\frac{1}{\alpha} + 1)^2|E|^3}{\Delta_{min}^2} \right) \Delta_{max}.
 \end{aligned}$$

Moreover, by the definition of filtered regret:

$$\begin{aligned}
 \text{Reg}_{\mu,\alpha,\beta}^A(T) &= \text{Reg}(\{\}) - T \cdot (1 - \beta) \cdot (\mathcal{M} - \frac{1}{\alpha} \cdot \bar{d}_{\mu}(C_{\mu}^*)) \\
 &= \text{Reg}(\mathcal{F}_T) + \text{Reg}(\neg\mathcal{F}_T \wedge B_T) + \text{Reg}(\neg\mathcal{F}_T \wedge \neg B_T) - T \cdot (1 - \beta) \cdot (\mathcal{M} - \frac{1}{\alpha} \cdot \bar{d}_{\mu}(C_{\mu}^*)) \\
 &\leq (1 - \beta)T\Delta_{max} + \left(1 + \frac{2(\frac{1}{\alpha} + 1)^2|E|^3}{\Delta_{min}^2} \right) \Delta_{max} - T \cdot (1 - \beta) \cdot (\mathcal{M} - \frac{1}{\alpha} \cdot \bar{d}_{\mu}(C_{\mu}^*)) \\
 &\leq \left(1 + \frac{2(\frac{1}{\alpha} + 1)^2|E|^3}{\Delta_{min}^2} \right) \Delta_{max},
 \end{aligned}$$

where, in the last step, we used the fact that $\Delta_{max} \leq \mathcal{M} - \frac{1}{\alpha} \cdot \bar{d}_{\mu}(C_{\mu}^*)$. \square

Funding Open access funding provided by Università della Calabria within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest and no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter discussed in this manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ailon N, Charikar M, Newman A (2008) Aggregating inconsistent information: ranking and clustering. *JACM* 55(5):231–2327
- Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Mach Learn* 47(2–3):235–256
- Bansal N, Blum A, Chawla S (2004) Correlation clustering. *Mach Learn* 56(1):89–113
- Berry DA, Fristedt B (1985) *Bandit problems: sequential allocation of experiments*. Chapman and Hall, London
- Bollapragada S, Garbiras M (2004) Scheduling commercials on broadcast television. *Oper Res* 52(3):337–345
- Bonchi F, García-Soriano D, Liberty E (2014) Correlation clustering: from theory to practice. In: *Proceedings of the ACM KDD conference*, pp 1972
- Bonchi F, García-Soriano D, Gullo F (2022) *Correlation clustering*. Synthesis lectures on data mining and knowledge discovery. Morgan & Claypool Publishers
- Bressan M, Cesa-Bianchi N, Paudice A, Vitale F (2019) Correlation clustering with adaptive similarity queries. In: *Proceedings of the NIPS conference*, pp. 12531–12540
- Cesa-Bianchi N, Lugosi G (2012) Combinatorial bandits. *JCSS* 78(5):1404–1422
- Charikar M, Guruswami V, Wirth A (2005) Clustering with qualitative information. *JCSS* 71(3):360–383
- Chawla S, Makarychev K, Schramm T, Yaroslavtsev G (2015) Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In: *Proceedings of the ACM STOC symposium*, pp. 219–228
- Chen L, Xu J, Lu Z (2018) Contextual combinatorial multi-armed bandits with volatile arms and sub-modular reward. In: *Proceedings of the NIPS conference*, pp 3251–3260
- Chen X, Huang W, Chen W, Lui JC (2018b) Community exploration: from offline optimization to online learning. In: *Proceedings of the NIPS conference*, pp 5474–5483
- Demaine ED, Emanuel D, Fiat A, Immorlica N (2006) Correlation clustering in general weighted graphs. *TCS* 361(2–3):172–187
- Dutta A, Ufimtsev V, Asaithambi A (2019) Correlation clustering based coalition formation for multi-robot task allocation. In: *Proceedings of the SAC symposium*, pp 906–913
- Galimberti E, Ciaperoni M, Barrat A, Bonchi F, Cattuto C, Gullo F (2020) Span-core decomposition for temporal networks: algorithms and applications. *ACM Trans Knowl Discov Data (TKDD)* 15(1):1–44
- García-Soriano D, Kutzkov K, Bonchi F, Tsourakakis C (2020) Query-efficient correlation clustering. In: *Proceedings of the WWW conference*, pp 1468–1478
- Giallombardo G, Jiang H, Miglionico G (2016) New formulations for the conflict resolution problem in the scheduling of television commercials. *Oper Res* 64(4):838–848
- Grillo H, Alemany M, Caldwell E (2022) Human resource allocation problem in the Industry 4.0: a reference framework. *Comput Ind Eng* 169:108110
- Gupta A (2005) Lecture notes—15-854: approximation algorithms. <https://www.cs.cmu.edu/afs/cs/academic/class/15854-f05/www/scribe/lec11.pdf>
- Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *JASA* 58(301):13–30
- Hübner AH, Düsterhöft T, Ostermeier M (2021) Shelf space dimensioning and product allocation in retail stores. *Eur J Oper Res* 292(1):155–171
- Juárez J, Santos CP, Brizuela CA (2022) A comprehensive review and a taxonomy proposal of team formation problems. *ACM CSUR* 54(7):15:31–153:33
- Kveton B, Szepesvári C, Wen Z, Ashkan A (2015) Cascading bandits: learning to rank in the cascade model. In: *Proceedings of the ICML conference*, pp 767–776
- Kveton B, Wen Z, Ashkan A, Szepesvári C (2015) Combinatorial cascading bandits. In: *Proceedings of the NIPS conference*, pp 1450–1458
- Lagrée P, Vernade C, Cappé O (2016) Multiple-play bandits in the position-based model. In: *Proceedings of the NIPS conference*, pp 1597–1605

- Liu K, Huang H, Zhang W, Hariri A, Fu Y, Hua KA (2021) Multi-armed bandit based feature selection. In: Proceedings of the of SIAM International conference on data mining (SDM), pp 316–323
- Mandaglio D, Tagarelli A (2019a) A combinatorial multi-armed bandit based method for dynamic consensus community detection in temporal networks. In: Proceedings of the DS conference, pp 412–427
- Mandaglio D, Tagarelli A (2019b) Dynamic consensus community detection and combinatorial multi-armed bandit. In: Proceedings of the ASONAM conference, pp 184–187
- Mandaglio D, Tagarelli A, Gullo F (2020) In and out: optimizing overall interaction in probabilistic graphs under clustering constraints. In: Proceedings of the ACM KDD conference, pp 1371–1381
- Mandaglio D, Tagarelli A, Gullo F (2021) Correlation clustering with global weight bounds. In: Proceedings of the ECML PKDD conference, pp 499–515
- Pandove D, Goel S, Rani R (2018) Correlation clustering methodologies and their fundamental results. *Expert Syst* 35(1)
- Puleo GJ, Milenkovic O (2015) Correlation clustering with constrained cluster sizes and extended weights bounds. *SIAM J Optim* 25(3):1857–1872
- Shamir R, Sharan R, Tsur D (2004) Cluster graph modification problems. *Discret Appl Math* 144(1–2):173–182
- Swamy C (2004) Correlation clustering: maximizing agreements via semidefinite programming. In: Proceedings of the ACM-SIAM SODA conference, pp 526–527
- Talebi MS, Zou Z, Combes R, Proutiere A, Johansson M (2017) Stochastic online shortest path routing: the value of feedback. *IEEE Trans Autom Control* 63(4):915–930
- Tomczak M, Tomczak E (2014) The need to report effect size estimates revisited. An overview of some recommended measures of effect size. *Trends Sport Sci* 1(21):19–25
- van Zuylen A, Williamson DP (2007) Deterministic algorithms for rank aggregation and other ranking and clustering problems. In: Proceedings of the WAOA work, pp 260–273
- Vaswani S, Lakshmanan LVS (2015) Influence maximization with bandits. [arXiv:1503.00024](https://arxiv.org/abs/1503.00024)
- Wang Q, Chen W (2017) Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In: Proceedings of the NIPS conference, pp 1161–1171
- Wang S, Chen W (2018) Thompson sampling for combinatorial semi-bandits. In: Proceedings of the ICML conference, pp 5101–5109
- Wu Q, Li Z, Wang H, Chen W, Wang H (2019) Factorization bandits for online influence maximization. In: Proceedings of the ACM KDD conference, pp 636–646
- Xu H, Liu Y, Lau WC, Li R (2020) Combinatorial multi-armed bandits with concave rewards and fairness constraints. In: Proceedings of the IJCAI conference, pp 2554–2560

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

F. Gullo¹ · D. Mandaglio² · A. Tagarelli² 

✉ A. Tagarelli
tagarelli@dimes.unical.it

F. Gullo
gullof@acm.org

D. Mandaglio
d.mandaglio@dimes.unical.it

¹ UniCredit, Rome, Italy

² Department of Computer Engineering, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, Rende, CS, Italy