Correlation Clustering with Global Weight Bounds

Domenico Mandaglio[1], Andrea Tagarelli[1], and Francesco Gullo[2]

[1] DIMES Dept., University of Calabria, Rende (CS), Italy
{d.mandaglio,tagarelli}@dimes.unical.it
[2] UniCredit, Rome, Italy
gullof@acm.org

Abstract. Given a set of objects and nonnegative real weights expressing "positive" and "negative" feeling of clustering any two objects together, *min-disagreement correlation clustering* partitions the input object set so as to minimize the sum of the intra-cluster negative-type weights plus the sum of the inter-cluster positive-type weights. Min-disagreement correlation clustering is **APX**-hard, but efficient constant-factor approximation algorithms exist if the weights are bounded in some way. The weight bounds so far studied in the related literature are mostly *local*, as they are required to hold for every object-pair. In this paper, we introduce the problem of min-disagreement correlation clustering with *global weight bounds*, i.e., constraints to be satisfied by the input weights altogether. Our main result is a sufficient condition that establishes when any algorithm achieving a certain approximation under the probability constraint keeps the same guarantee on an input that violates the constraint. This extends the range of applicability of the most prominent existing correlation-clustering algorithms, including the popular Pivot, thus providing benefits, both theoretical and practical. Experiments demonstrate the usefulness of our approach, in terms of both worthiness of employing existing efficient algorithms, and guidance on the definition of weights from feature vectors in a task of *fair clustering*.

1 Introduction

Correlation clustering [8] is a popular clustering formulation that has received considerable attention from both theoreticians and practitioners, and has found application in several contexts, including document clustering, duplicate detection, computational biology, image segmentation [10, 22].

The input of correlation clustering is a set V of objects, and two nonnegative, real-valued weights w^+_{uv}, w^-_{uv} for every (unordered) object pair $u, v \in V$. Any "positive" w^+_{uv} (resp. "negative" w^-_{uv}) weight expresses the benefit of clustering u and v together (resp. separately). This input can equivalently be represented as a graph G with vertex set V and edge weights w^+_{uv}, w^-_{uv}, for all $u, v \in V$, and with edge (u, v) being drawn only if at least one among w^+_{uv} and w^-_{uv} is nonzero.

The objective of correlation clustering is to partition V so as to either minimize the sum of intra-cluster negative-type weights plus the sum of inter-cluster

positive-type weights (*min-disagreement*), or maximize the sum of intra-cluster positive-type weights plus the sum of inter-cluster negative-type weights (*max-agreement*). The two formulations are equivalent in terms of exact optimization and complexity class (both **NP**-hard [8, 25]), but they have different approximation properties, with the maximization variant being easier in this respect.

Apart from being more theoretically appealing, min-disagreement correlation clustering tends to be more relevant than the maximization counterpart in practice too. The reason is twofold. First, the best known approximation algorithms for max-agreement correlation clustering either yield trivial solutions (single-cluster and all-singletons solutions are $\frac{1}{2}$-approximate solutions for complete graphs with binary weights [26]), or are inefficient and provide unpractical clusterings with a fixed number of clusters (like semidefinite-programming Swamy's algorithm for general graphs [26], which is very expensive and always yields a 6-cluster solution). Second, more importantly, among the algorithms for the minimization version is the popular Pivot [5], which provides the best tradeoff between theoretical guarantees (it achieves constant-factor expected approximation guarantee), efficiency (it takes linear time), and ease of implementation.

Correlation-clustering with local weight bounds. The seminal work by Bansal *et al.* [8] limits the input graph to be complete, with binary weights, and with exactly one nonzero weight for each weight pair (i.e., $(w^+_{uv}, w^-_{uv}) \in \{(0,1),(1,0)\}$, for all $u, v \in V$). Even for this particular input, min-disagreement correlation clustering is **APX**-hard [11], although it admits constant-factor approximation algorithms [5, 8, 11, 12, 27]. Since then, less restrictive inputs have been considered. With no constraints on the input weights, the best known approximation factor is $\mathcal{O}(\log |V|)$ [11, 16], and is unlikely to be meliorable [11, 16].

Motivated by this and the above arguments in favor of the minimization version, the research community has focused on weight bounds that go beyond Bansal *et al.*'s ones, but are still restrictive enough to allow constant-factor guarantee. In this regard, the *probability constraint* (i.e., $w^+_{uv} + w^-_{uv} = 1$, $\forall u, v \in V$) has received significant attention. Under this constraint, Pivot is recognized as a (randomized expected) 5-approximation algorithm [5]. Coupling the probability constraint with *triangle inequality* (i.e., $w^-_{uz} \leq w^-_{uv} + w^-_{vz}$, $\forall u, v, z \in V$) makes Pivot's approximation factor become 2. Further algorithms achieve a factor-4 guarantee under the probability constraint [11], and $(5 - \frac{1}{h})$-approximation for a generalization of the probability constraint (i.e., $\forall u, v \in V$, $w^+_{uv} \leq 1$, $w^-_{uv} \leq h$ for some $h \in [1, +\infty)$, and $w^+_{uv} + w^-_{uv} \geq 1$) [23]. Those two algorithms however are based on rounding the solution to a (large) linear program, thus they do not possess Pivot's nice peculiarities of efficiency and ease of implementation.

This work: correlation clustering with *global* weight bounds. Regardless of the type, the weight bounds that have been so far studied are *local bounds*, i.e., constraints that are required to hold *for every object pair in isolation*.

In this work, we are the first to consider *global weight bounds* in min-disagreement correlation clustering. We derive bounds on edge weights' aggregate functions that are sufficient to lead to proved quality guarantees. Specifically, let avg^+ and avg^- be the average of the positive-type weights and negative-

type weights over all the input vertex pairs, respectively. Let also Δ_{max} be the maximum absolute difference between the positive-type weight and the negative-type weight of a vertex pair. Our result is: if the condition $avg^+ + avg^- \geq \Delta_{max}$ holds for a graph G, then it is possible to construct a graph G' (in linear time and space) such that (i) the probability constraint holds on G', and (ii) an α-approximate clustering on G' (i.e., a clustering whose objective-function value is no more than α times G''s optimum) is an α-approximate clustering on G too.

A noteworthy consequence of this result is that, if a graph G satisfies our condition, then the Pivot algorithm can be used to get (in linear time and space) a clustering achieving a 5-approximation guarantee on G.[3] This corresponds to extending the range of validity of Pivot's guarantee beyond the probability constraint: our global-weight-bounds condition now suffices for the 5-approximation to hold. A key advantage of this finding is that our condition is milder than the probability constraint, thus more likely to be satisfied. For instance, it may happen that a bunch of edges are missing from the input graph (meaning violation of the probability constraint for at least those unlinked vertex pairs), but, if our condition holds, still one can get a 5-approximate clustering with Pivot.

We point out that our result is general and holds for *any* min-disagreement correlation-clustering algorithm achieving approximation guarantees under the probability constraint. However, the contextualization to the Pivot algorithm is relevant and worth to be emphasized, because, as said above, Pivot achieves the best tradeoff between quality guarantees, efficiency, and ease of implementation.

Benefits of our result. We believe that the findings of this work can be tremendously useful, from several perspectives.

Practical benefits. Our result can be exploited to quickly yet easily recognize whether employing probability-constraint-aware approximation algorithms is a worth choice even if the probability constraint is not met. As an example, consider a graph that violates the probability constraint. So far, that graph would have likely been handled with linear-programming (LP) algorithms [11, 16], as they achieve (factor-$\mathcal{O}(\log |V|)$) approximation guarantees on general graphs/weights (whereas algorithms like Pivot are just heuristics if the probability constraint does not hold). Instead, our condition can be used as an indicator of whether Pivot can still achieve guarantees even if the probability constraint is violated, thus being preferred over the LP algorithms. This has important practical implications, as Pivot is much faster and easier-to-implement than the LP counterparts. In our evaluation we experimentally confirm this theoretical finding, by showing that a better fulfilment of our condition corresponds to better performance of Pivot with respect to the LP algorithms, and vice versa.

A second practical exploitability of our result concerns the task of feature selection for clustering. In the context of correlation clustering this corresponds to selecting features that lead edge weights to express the best tradeoff between an accurate representation of the objects' vectors (i.e., discarding not too many

[3] In fact, a probability-constraint-compliant graph G' can be derived from G in linear time and space (statement (i) of our result). Pivot on G' yields a 5-approximate clustering [5]. A 5-approximate clustering on G' is a 5-approximate clustering on G (statement (ii) of our result).

features), and the way how the weights facilitate the downstream correlation-clustering algorithm performing well (e.g., by making it achieve approximation guarantees). Our global-weight-bounds condition can be an effective yet easy-to-use guiding principle to the achievement of this tradeoff. Being less restrictive than local weight bounds, our condition can be fulfilled more easily (e.g., in case of probability constraint, it is hard to find a subset of features leading to positive-type and negative-type weights summing *exactly* to one *for all the object pairs*). In our experiments we showcase this capability in a task of *fair clustering*.

Theoretical benefits. This work extends the validity range of the approximation guarantees of algorithms for min-disagreement correlation clustering. This extension can pave the way for more advanced theoretical results. As an example, it is not uncommon that correlation clustering is a building block of a more complex problem [17, 20, 21]. Thus, more general guarantees in correlation clustering may enable better theoretical results on those complex problems too.

Benefits for the research community. To the best of our knowledge, global weight bounds for correlation clustering have never been studied so far. We believe this work can pioneer a brand new line of research, and stimulate the community to go beyond our initial results.

Summary of contributions and outline. The contributions we achieve in this work can be summarized as follows. We focus for the first time on global weight bounds in (the minimization formulation of) correlation clustering (Section 3). We derive a sufficient condition on input weights' aggregate functions to extend the validity range of the approximation guarantees of existing correlation-clustering algorithms beyond the probability constraint (Section 4). We experimentally assess that our condition is an effective indicator of the empirical performance of existing probability-constraint-aware correlation-clustering algorithms (Section 5.1). We showcase our results in a real-world scenario of fair clustering (Section 5.2).

2 Related Work

Correlation clustering. The literature on min-disagreement correlation clustering that is functional to our work has been presented in the Introduction. As a complement, we (briefly) overview the main results on the maximization formulation (not a focus of this work), and extensions to the basic formulations.

For original Bansal *et al.*'s input of unweighted and complete graphs [8], max-agreement correlation clustering admits a PTAS [8]. On general graphs/weights, it becomes **APX**-hard [11], but admits constant-factor approximation algorithms, achieving factor-0.7664 [11] and factor-0.7666 [26] guarantees. Extensions to the basic correlation-clustering formulations include constrained/relaxed formulations, and adaptations to nonconventional types of graph or computational settings. We point the interested reader to [10, 22] for more details.

In this work we shift the attention from local to *global* weight bounds in min-disagreement correlation clustering. To the best of our knowledge, this is a completely novel perspective that has never been considered so far.

Fair clustering. Roughly speaking, the problem of fair clustering consists in partitioning a set of objects based on both clustering quality and *fairness*, i.e., limiting as much as possible the bias against/towards particular objects' subsets.

Chierichetti *et al.*'s seminal work [14] formulates fair versions of the traditional k-center and k-median problems. Since then, research has focused on generalizing those formulations [9, 24], incorporating fairness constraints into k-center [18], scalability of fair k-median [7], different fairness measures [3, 13], and fair versions of other traditional problems, i.e., k-means [1], spectral clustering [19], hierarchical clustering [2]. As for correlation clustering, Ahmadian *et al.* [4] study the problem where vertices of a *complete and unweighted* graph are assigned a *single label* representing a protected class attribute (e.g., gender, ethnicity), and every cluster is constrained to fairly represent each label.

In this work we showcase our theoretical results in a task of fair clustering. We pick a scenario where positive-type and negative-type edge weights express similarities on non-sensitive and sensitive features assigned to the input vertices, respectively. The goal is to define such weights so as to account for both an effective representation of the semantics underlying objects' features, and the peculiarities that make the downstream correlation-clustering algorithm effective. Thus, our setting differs from Ahmadian *et al.*'s one, where the graph is complete and unweighted, and vertices are not assigned feature vectors, but just a class label. In any case, the focus on fair clustering in this work is just on the application side: advancing the fair-clustering literature is beyond our scope.

3 Problem Definition

In this work we tackle the problem of *min-disagreement correlation clustering*:

*Problem 1 (*Min-CC *[5]).* Given an undirected graph $G = (V, E)$, with vertex set V and edge set $E \subseteq V \times V$, and nonnegative weights $w_e^+, w_e^- \in \mathbb{R}_0^+$ for all edges $e \in E$, find a clustering (i.e., an injective function expressing cluster-membership) $\mathcal{C} : V \to \mathbb{N}^+$ that minimizes

$$\sum_{(u,v)\in E, \mathcal{C}(u)=\mathcal{C}(v)} w_{uv}^- \quad + \sum_{(u,v)\in E, \mathcal{C}(u)\neq\mathcal{C}(v)} w_{uv}^+. \tag{1}$$

For the sake of presentation, we assume $w_e^+ = w_e^- = 0$, for all $e \notin E$, and non-trivial Min-CC instances, i.e., $w_e^+ \neq w_e^-$, for some $e \in E$.

Min-CC is **NP**-hard [8, 25] yet difficult to approximate, being it **APX**-hard even for complete graphs and edge weights $(w_e^+, w_e^-) \in \{(0, 1), (1, 0)\}$, $\forall e \in E$ [11]. For general (i.e., not necessarily complete) graphs and general (i.e., unconstrained) weights, the best known approximation factor is $\mathcal{O}(\log |V|)$ [11, 16]. This factor improves if restrictions on edge weights are imposed. A constraint that has received considerable attention is the *probability constraint* (pc):

Definition 1 (Probability constraint). *A* Min-CC *instance is said to satisfy the* probability constraint (pc) *if* $w_{uv}^+ + w_{uv}^- = 1$*, for all vertex pairs* $u, v \in V$*.*

Algorithm 1 Pivot [5]

Input: Graph $G = (V, E)$; nonnegative weights w_e^+, w_e^-, $\forall e \in E$
Output: Clustering $\mathcal{C}$ of V
1: $\mathcal{C} \leftarrow \emptyset$, $V' \leftarrow V$
2: **while** $V' \neq \emptyset$ **do**
3: pick a pivot vertex $u \in V'$ uniformly at random
4: add $\mathcal{C}_u = \{u\} \cup \{v \in V' \mid (u, v) \in E, w_{uv}^+ > w_{uv}^-\}$ to $\mathcal{C}$ and remove C_u from V'

A MIN-CC instance obeying the PC necessarily corresponds to a complete graph (otherwise, any missing edge would violate the PC). Under the PC, MIN-CC admits constant-factor guarantees. The best known approximation factor is 4, achievable – as shown in [23] – by Charikar *et al.*'s algorithm [11]. That algorithm is based on rounding the solution to a large linear program (with a number $\Omega(|V|^3)$ of constraints), thus being feasible only on small graphs.

Here, we are particularly interested in the Pivot algorithm [5], due to its theoretical properties – it achieves a factor-5 expected guarantee for MIN-CC under the PC – and practical benefits – it takes $\mathcal{O}(|E|)$ time, and is easy-to-implement. Pivot simply picks a random vertex u, builds a cluster as composed of u and all the vertices v such that an edge with $w_{uv}^+ > w_{uv}^-$ exists, and removes that cluster. The process is repeated until the graph has become empty (Algorithm 1).

4 Theoretical Results and Algorithms

Let MIN-PC-CC denote the version of MIN-CC operating on instances that satisfy the PC. The main theoretical result of this work is a *sufficient condition* – to be met *globally* by the input edge weights – on the existence of a *strict approximation-preserving* (SAP) reduction from MIN-CC to MIN-PC-CC. In the remainder of this section we detail our findings, presenting partial results (Sections 4.1–4.2), our overall result (Section 4.3), and algorithms (Section 4.4).

4.1 PC-reduction

As a first partial result, in this subsection we define the proposed reduction from MIN-CC instances to MIN-PC-CC ones, and the condition that makes it yield valid (i.e., nonnegative) edge weights. We start by recalling some basic notions, including the one of strict approximation-preserving (SAP) reduction.

Definition 2 (Minimization problem, optimum, performance ratio [6]). *A* minimization problem *Π is a triple $(\mathcal{I}, sol, obj)$, where $\mathcal{I}$ is the set of problem instances; for every $I \in \mathcal{I}$, $sol(I)$ is the set of feasible solutions of I; obj is the objective function, i.e., given $I \in \mathcal{I}$, $S \in sol(I)$, $obj(I, S)$ measures the quality of solution S to instance I.*
$OPT_\Pi(I)$ denotes the objective-function value of an optimal solution *to I.*
Given $I \in \mathcal{I}$, $S \in sol(I)$, $R_\Pi(I, S) = obj(I, S)/OPT_\Pi(I)$ denotes the performance ratio *of S with respect to I.*

Definition 3 (Reduction and SAP-reduction [15]). *Let $\Pi_1 = (\mathcal{I}_1, sol_1, obj_1)$ and $\Pi_2 = (\mathcal{I}_2, sol_2, obj_2)$ be two minimization problems.*
A reduction *from Π_1 to Π_2 is a pair (f, g) of polynomial-time-computable functions, where $f : \mathcal{I}_1 \to \mathcal{I}_2$ maps Π_1's instances to Π_2's instances, and, given $I_1 \in \mathcal{I}_1$, $g : sol_2(f(I_1)) \to sol_1(I_1)$ maps back Π_2's solutions to Π_1's solutions. A reduction is said* strict approximation-preserving *(SAP) if, for any $I_1 \in \mathcal{I}_1$, $S_2 \in sol_2(f(I_1))$, it holds that $R_{\Pi_1}(I_1, g(I_1, S_2)) \leq R_{\Pi_2}(f(I_1), S_2)$.*

The proposed reduction is as follows. To map MIN-CC instances to MIN-PC-CC ones, we adopt a function f that simply redefines edge weights, while leaving the underlying graph unchanged. Function g is set to the identity function. That is, a MIN-PC-CC solution is interpreted as a solution to the original MIN-CC instance as is. Function f makes use of two constants $M, \gamma > 0$ (which will be better discussed later), and σ_e quantities, $\forall e \in E$, which are defined as:

$$\sigma_e = \gamma \, (w_e^+ + w_e^-) - M. \tag{2}$$

We term our reduction PC-*reduction* and define it formally as follows.

Definition 4 (PC-reduction). *The* PC-reduction *is a reduction (f, g) from* MIN-CC *to* MIN-PC-CC*, where g is the identity function, while f maps a* MIN-CC *instance $\langle G = (V, E), \{w_e^+, w_e^-\}_{e \in E} \rangle$ to a* MIN-PC-CC *instance $\langle G' = (V', E'), \{\tau_e^+, \tau_e^-\}_{e \in E'} \rangle$, such that $V' = V$, $E' = V \times V$, and*

$$\tau_e^+ = \tfrac{1}{M} \left(\gamma \, w_e^+ - \tfrac{\sigma_e}{2} \right), \quad \tau_e^- = \tfrac{1}{M} \left(\gamma \, w_e^- - \tfrac{\sigma_e}{2} \right), \quad \forall e \in E'. \tag{3}$$

Note that the proposed PC-reduction is always guaranteed to yield τ_e^+, τ_e^- weights satisfying the PC (i.e., $\tau_e^+ + \tau_e^- = 1$), for any M and γ. As a particular case, recalling the assumption $w_e^+ = w_e^- = 0$ for $e \notin E$, the PC-reduction yields weights $\tau_e^+ = \tau_e^- = 0.5$ for any $e \notin E$. However, not every choice of M and γ leads to nonnegative τ_e^+, τ_e^- weights, as stated next.

Lemma 1. *The PC-reduction yields nonnegative τ_e^+, τ_e^- weights if and only if $M - \gamma \, \Delta_{max} \geq 0$, where $\Delta_{max} = \max_{e \in E} |w_e^+ - w_e^-|$.*

Proof. By simple math on the formula of τ_e^+, τ_e^- in Definition 4, it follows that $\tau_e^+, \tau_e^- \geq 0$ holds if and only if the conditions $w_e^+ - w_e^- \geq -M/\gamma$ and $w_e^+ - w_e^- \leq M/\gamma$ are simultaneously satisfied. This in turn corresponds to have $|w_e^+ - w_e^-| \leq M/\gamma$ satisfied. As the latter must hold for all $e \in E$, then the lemma. □

Constraining M and γ as in Lemma 1 is a key ingredient of our ultimate global-weight-bounds condition. We will come back to it in Section 4.3.

4.2 Preserving the approximation factor across PC-reduction

Let I be a MIN-CC instance and I' be the MIN-PC-CC instance derived from I via PC-reduction. Here, we present a further partial result, i.e., a sufficient condition according to which an approximation factor holding on I' is preserved on I. We state this result in Lemma 3. Before that, we provide the following auxiliary lemma, which shows the relationship between the objective-function values of a clustering $\mathcal{C}$ on I and on I'.

Lemma 2. *Let $I = \langle G = (V, E), \{w_e^+, w_e^-\}_{e \in E} \rangle$ be a* MIN-CC *instance, and $I' = \langle G' = (V, E' = V \times V), \{\tau_e^+, \tau_e^-\}_{e \in E'} \rangle$ be the* MIN-PC-CC *instance derived from I via* PC*-reduction. Let also $\mathcal{C}$ be a clustering of V. The following relationship holds between the objective-function value $obj(I, \mathcal{C})$ of $\mathcal{C}$ on I and the objective-function value $obj(I', \mathcal{C})$ of $\mathcal{C}$ on I':*

$$obj(I, \mathcal{C}) = \tfrac{M}{\gamma}\, obj(I', \mathcal{C}) + \tfrac{1}{2\gamma} \textstyle\sum_{u,v \in V} \sigma_{uv}. \tag{4}$$

Proof.

$$obj(I', \mathcal{C}) = \textstyle\sum_{\substack{(u,v) \in E', \\ \mathcal{C}(u) = \mathcal{C}(v)}} \tfrac{1}{M} \left(\gamma\, w_{uv}^+ - \tfrac{\sigma_{uv}}{2}\right) + \sum_{\substack{(u,v) \in E', \\ \mathcal{C}(u) = \mathcal{C}(v)}} \tfrac{1}{M} \left(\gamma\, w_{uv}^- - \tfrac{\sigma_{uv}}{2}\right) =$$

$$= \tfrac{\gamma}{M} \Big(\textstyle\sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) = \mathcal{C}(v)}} w_{uv}^+ + \sum_{\substack{(u,v) \in E, \\ \mathcal{C}(u) \neq \mathcal{C}(v)}} w_{uv}^- \Big) - \tfrac{1}{2M} \Big(\textstyle\sum_{\substack{(u,v) \in E', \\ \mathcal{C}(u) = \mathcal{C}(v)}} \sigma_{uv} + \sum_{\substack{(u,v) \in E', \\ \mathcal{C}(u) \neq \mathcal{C}(v)}} \sigma_{uv} \Big) =$$

$$= \tfrac{\gamma}{M}\, obj(I, \mathcal{C}) - \tfrac{1}{2M} \textstyle\sum_{u,v \in V} \sigma_{uv}. \qquad \square$$

Lemma 3. *Let I and I' be the two instances of Lemma 2. Let also $\mathcal{C}$ be an α-approximate solution to I', i.e., a clustering achieving objective-function value no more than α times I''s optimum, for any $\alpha > 1$. It holds that: if $\frac{1}{2\gamma} \sum_{e \in E} \sigma_e \geq 0$, then $\mathcal{C}$ is an α-approximate solution to I too.*

Proof. Let OPT and OPT' be the optima of I and I', respectively. It holds that:

$$\begin{aligned} & obj(I', \mathcal{C}) \leq \alpha\, OPT' \\ & \Rightarrow \tfrac{M}{\gamma}\, obj(I', \mathcal{C}) + \tfrac{1}{2\gamma} \textstyle\sum_{u,v \in V} \sigma_{uv} \leq \alpha \left(\tfrac{M}{\gamma}\, OPT' + \tfrac{1}{2\gamma} \textstyle\sum_{u,v \in V} \sigma_{uv} \right) \\ & \Leftrightarrow obj(I, \mathcal{C}) \leq \alpha\, OPT, \end{aligned}$$

where the second step holds because $\frac{1}{2\gamma} \sum_{e \in E} \sigma_e \geq 0$ and $\alpha > 1$ by hypothesis, while the last step holds because of Lemma 2. $\square$

4.3 Ultimate global weight bounds

With the above partial results in place, we can now present our ultimate result, i.e., a sufficient condition to guarantee that the PC-reduction is a SAP-reduction. To show our result, for a MIN-CC instance $\langle G = (V, E), \{w_e^+, w_e^-\}_{e \in E} \rangle$ we define:

$$avg^+ = \binom{|V|}{2}^{-1} \textstyle\sum_{e \in E} w_e^+, \quad avg^- = \binom{|V|}{2}^{-1} \textstyle\sum_{e \in E} w_e^-. \tag{5}$$

Theorem 1. *If $avg^+ + avg^- \geq \Delta_{max}$, then the* PC*-reduction is a* SAP*-reduction.*

Proof. Lemma 3 provides a (sufficient) condition to have an approximation factor on a MIN-PC-CC instance carried over to the original MIN-CC instance. Thus, that condition suffices to make the PC-reduction a SAP-reduction according to Definition 3. The condition in Lemma 3 has to be coupled with the one in

Algorithm 2 GlobalCC

Input: Graph $G = (V, E)$; nonnegative weights $w_e^+, w_e^-, \forall e \in E$, satisfying Theorem 1; algorithm A achieving α-approximation guarantee for Min-pc-CC
Output: Clustering $\mathcal{C}$ of V
1: choose M, γ s.t. $\frac{M}{\gamma} \in [\Delta_{max}, avg^+ + avg^-]$ {Theorem 1}
2: compute $\tau_{uv}^+, \tau_{uv}^-, \forall u, v \in V$, as in Equation (3) (using M, γ defined in Step 1)
3: $\mathcal{C} \leftarrow$ run A on Min-pc-CC instance $\langle G' = (V, V \times V), \{\tau_e^+, \tau_e^-\}_{e \in V \times V} \rangle$

Lemma 1, which guarantees nonnegativity of the edge weights of the yielded Min-pc-CC instance. To summarize, we thus require the following:

$$\begin{cases} \frac{M}{\gamma} \geq \Delta_{max}, & \{\text{Lemma 1}\} \\ \frac{1}{2\gamma} \sum_{u,v \in V} \sigma_{uv} \geq 0 \;\Leftrightarrow\; \frac{M}{\gamma} \leq avg^+ + avg^-, & \{\text{Lemma 3}\} \end{cases}$$

which corresponds to $\frac{M}{\gamma} \in [\Delta_{max}, avg^+ + avg^-]$, i.e., to $avg^+ + avg^- \geq \Delta_{max}$. □

4.4 Algorithms

According to Theorem 1, if $avg^+ + avg^- \geq \Delta_{max}$ for a Min-CC instance, then any α-approximation algorithm for Min-pc-CC can be employed – *as a black box* – to get an α-approximate solution to that Min-CC instance. The algorithm for doing so is simple: get a Min-pc-CC instance via pc-reduction, and run the black-box algorithm on it (Algorithm 2). Being the pc-reduction sap, the guarantee of this algorithm straightforwardly follows as a corollary of Theorem 1.

Corollary 1. *Let I be a* Min-CC *instance, and A be an α-approximation algorithm for* Min-pc-CC. *Algorithm 2 on input $\langle I, \mathsf{A} \rangle$ achieves factor-α guarantee on I.*

Let $T(\mathsf{A})$ be the running time of the black-box algorithm A. The time complexity of Algorithm 2 is $\mathcal{O}(\max\{|E|, T(\mathsf{A})\})$, assuming that there is no need to materialize edge weights τ_e^+, τ_e^- for missing edges $e \notin E$. This is an assumption valid in most cases: we recall that $e \notin E \Rightarrow \tau_e^+ = \tau_e^- = 0.5$, thus it is likely that their definition can safely be kept implicit. For instance, this assumption holds if Pivot [5] is used as a black-box algorithm (although with Pivot the picture is much simpler, see below). Instead, the assumption is not true for the LP algorithms in [11, 16]. In that case, however, the time complexity of Algorithm 2 would correspond to the running time of those LP algorithms nevertheless, as they take (at least) $\Omega(|V|^3)$ time to build their linear programs.

Using Pivot in Algorithm 2. It is easy to see that $w_e^+ > w_e^- \Leftrightarrow \tau_e^+ > \tau_e^-, \forall e \in E$. As Pivot makes its choices based on the condition $w_e^+ > w_e^-$ solely, the output of Algorithm 2 equipped with Pivot corresponds to the output of Pivot run directly on the input Min-CC instance. Thus, to get the 5-approximation guaranteed by Pivot, it suffices to run Pivot on the original input, without explicitly performing the pc-reduction. This finding holds in general for any algorithm whose output is determined by the condition $w_e^+ > w_e^-$ only. It does not hold for the LP algorithms: in that case, the general Algorithm 2 is still needed.

Table 1: Main characteristics of real-world graph datasets (left) and relational datasets (right) used in our evaluation stages.

	$\|V\|$	$\|E\|$	den.	a_deg	a_pl	diam	cc
Karate	34	78	0.14	4.59	2.41	5	0.26
Dolphins	62	159	0.08	5.13	3.36	8	0.31
Adjnoun	112	425	0.07	7.59	2.54	5	0.16
Football	115	613	0.09	10.66	2.51	4	0.41

	#objs.	#attrs.	fairness-aware (sensitive) attributes
Adult	32 561	7/8	race, sex, country, education, occupation, marital-status, workclass, relationship
Bank	41 188	18/3	job, marital-status, education
Credit	10 127	17/3	gender, marital-status, education-level
Student	649	28/5	sex, male_edu, female_edu, male_job, female_job

Role of M and γ. According to Theorem 1, τ_e^+, τ_e^- weights can be defined by picking any values of M and γ such that $\frac{M}{\gamma} \in [\Delta_{max}, avg^+ + avg^-]$. The condition $avg^+ + avg^- \geq \Delta_{max}$ ensures that the $[\Delta_{max}, avg^+ + avg^-]$ range is nonempty, while the assumption made in Section 3 that our input Min-CC's instances are nontrivial (thus, $\Delta_{max} > 0$) guarantees $M, \gamma > 0$.

From a theoretical point of view, all valid values of M and γ are the same. The choice of M and γ may instead have practical implications. Specifically, M and γ determine the difference between the resulting positive-type and negative-type edge weights. This may influence the empirical performance of those algorithms (e.g., the LP algorithms) for which the weight values matter. However, we remark that, in the case of Pivot — which just depends on whether the positive-type weight is more than the negative-type one — M and γ do not play any role, not even empirically. Being Pivot the main object of our practical focus, we defer a deeper investigation on M and γ to future work (see Section 6).

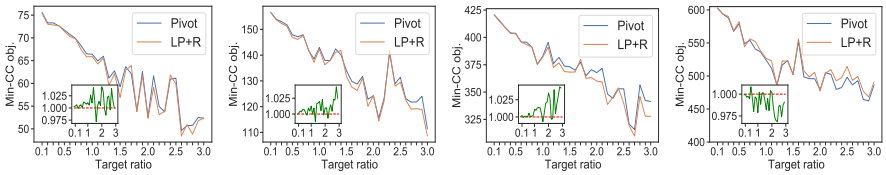
5 Experiments

5.1 Analysis of the global-weight-bounds condition

Settings. We selected four real-world graphs,[4] whose summary is reported in Table 1-(left). Note that the small size of such graphs is not an issue because this evaluation stage involves, among others, linear-programming correlation-clustering algorithms, whose time complexity ($\Omega(|V|)^3$) makes them unaffordable for graphs larger than that. We augmented these graphs with artificially-generated edge weights, to test different levels of fulfilment of our global-weight-bounds condition stated in Theorem 1. We controlled the degree of compliance of the condition by a *target ratio* parameter, defined as $t = \Delta_{max}/(avg^+ + avg^-)$. The condition is satisfied if and only if $t \in [0, 1]$, and smaller target-ratio values correspond to better fulfilment of the condition, and vice versa.

Given a desired target ratio, edge weights are generated as follows. First, all weights are drawn uniformly at random from a desired $[lb, ub]$ range. Then, the weights are adjusted in a two-step iterative fashion, until the desired target ratio is achieved: (i) keeping the maximum gap Δ_{max} fixed, the weights are changed for pairs that do not contribute to Δ_{max} so as to reflect a change in avg^+, avg^-; (ii) keeping avg^+, avg^- fixed, Δ_{max} is updated by randomly modifying pairs

[4] Publicly available at http://konect.cc/networks/



(a) Karate (b) Dolphins (c) Adjnoun (d) Football

Fig. 1: Min-CC objective by varying the target ratio.

that contribute to Δ_{max}. Once properly adjusted to meet the desired target ratio, weight pairs are randomly assigned to the edges of the input graph.

We compared the performance of Pivot (Algorithm 1 [5]) to one of the state-of-the-art algorithms achieving factor-$\mathcal{O}(\log |V|)$ guarantee on general graphs/weights [11]. We dub the latter LP+R, alluding to the fact that it rounds the solution of a linear program. We evaluated correlation-clustering objective, number of output clusters, and runtimes of these algorithms.

Results. Figure 1 shows the quality (i.e., Min-CC objective) of the clusterings produced by the selected algorithms, with the bottom-left insets reporting the ratio between the performance of Pivot and LP+R. Results refer to target ratios t varied from $[0, 3]$, with stepsize 0.1, and weights generated with $lb = 0, ub = 1$. For each target ratio, all reported measurements correspond to averages over 10 weight-generation runs, and each of such runs in turn corresponds to averages over 50 runs of the tested algorithms (being them both randomized).

The main goal here is to have experimental evidence that a better fulfilment of our global condition leads to Pivot's performance closer to LP+R's one, and vice versa. This would attest that our condition is a reliable proxy to the worthiness of employing Pivot. Figure 1 confirms this claim: in all datasets, Pivot performs more closely to LP+R as the target ratio gets smaller. In general, Pivot performs similarly to LP+R for $t \in [0, 1]$, while being outperformed for $t > 1$. This conforms with the theory: on these small graphs, factor-5 Pivot's approximation is close to factor-$\mathcal{O}(\log |V|)$ LP+R's approximation. Pivot achieves the best performance on *Football*, where it outperforms LP+R even if the condition is not met. This is motivated by *Football*'s higher clustering coefficient and average degree, which help Pivot sample vertices (and, thus, build clusters) in dense regions of the graph. This is confirmed by the number of clusters (Table 2-(right)): Pivot yields more clusters than LP+R on all datasets but *Football*.

As far as runtimes (Table 2-(left)),[5] Pivot is extremely faster than LP+R, as expected. The inefficiency of LP+R further emphasizes the importance of our result in extending the applicability of faster algorithms like Pivot.

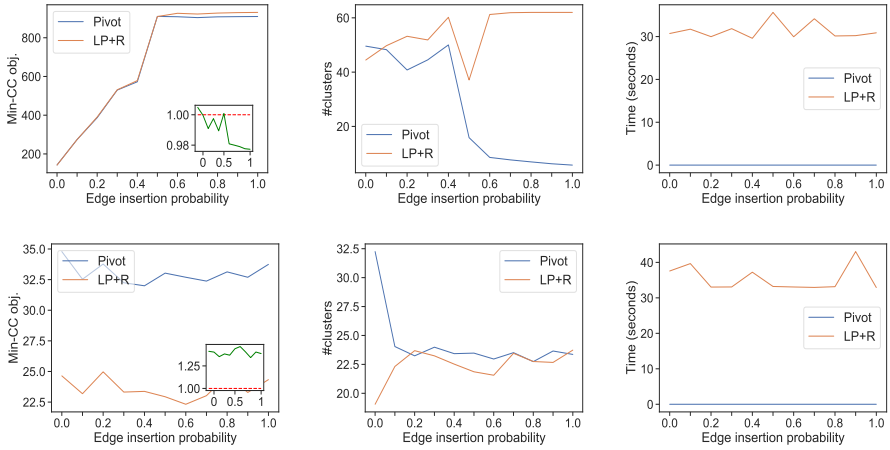
We complement this stage of evaluation by testing different graph densities. We synthetically added edges with uniform probability, ranging from 0 (no insertions) to 1 (complete graph). Figure 2 shows the results on *Dolphins* (similar

[5] Experiments were carried out on the Cresco6 cluster https://www.eneagrid.enea.it

Table 2: Running times (left) and avg. clustering-sizes for various target ratios (right).

	Pivot (secs.)	LP+R (secs.)
Karate	< 1	1.9
Dolphins	< 1	36.58
Adjnoun	< 1	775.4
Football	< 1	819.8

	0.1		0.5		1		2		3	
	Pivot	LP+R	Pivot	LP+R	Pivot	LP+R	Pivot	LP+R	Pivot	LP+R
Karate	21.75	17.18	29.61	27.93	27.22	24.66	25.55	23.82	28.17	26.81
Dolphins	49.25	50.59	45.3	38.67	49.57	44.45	47.91	48.05	48.89	43.66
Adjnoun	70.35	65.93	80.97	75.86	90.76	84.93	85.83	70.41	91.27	79.78
Football	64.43	84.91	77.14	96.43	68.35	78.72	78.65	85.31	90.87	100.31



(a) Min-CC objective (b) Number of clusters (c) Running time

Fig. 2: Varying graph density: target ratio 1 (top) and 20 (bottom), on *Dolphins*.

results are found in all the other datasets, here omitted for the sake of brevity), and for target ratios $t = 1$ (borderline satisfaction of our condition) and $t = 20$ (far fulfilment of the condition). Again, the results meet the expectations: in terms of clustering quality, Pivot performs closely to or better than LP+R for $t = 1$, while the opposite happens for $t = 20$. Denser graphs correspond to better Pivot performance. This is again motivated by the above argument that higher densities favor better Pivot's random choices. Runtimes are not affected by the differences in graph density. This is expected as well, as LP+R runtimes are dominated by the time spent in building and solving the linear program, which depends on the number of vertices only, whereas variations in the runtimes of Pivot cannot be observed due to the small size of the datasets at hand.

5.2 Application to fair clustering

Let $\mathcal{X}$ be a set of objects defined over a set of attributes $\mathcal{A}$. The latter is assumed to be divided into two sets, $\mathcal{A}^F$ and $\mathcal{A}^{\neg F}$, where $\mathcal{A}^F$ contains *fairness-aware*, or *sensitive* attributes (e.g., gender, race, religion), and $\mathcal{A}^{\neg F}$ denotes the remaining, *non-sensitive* attributes. In both cases, we assume that part of the attributes might be numerical, and the others as categorical; we will use superscripts N and C to distinguish the two types, therefore $\mathcal{A}^F = \mathcal{A}^F_N \cup \mathcal{A}^F_C$ and $\mathcal{A}^{\neg F} = \mathcal{A}^{\neg F}_N \cup \mathcal{A}^{\neg F}_C$.

We consider a twofold fair-clustering objective: cluster the objects such that (i) the intra-cluster similarity and the inter-cluster similarity are maximized and minimized, respectively, according to the non-sensitive attributes; (ii) the intra-cluster similarity and the inter-cluster similarity are minimized and maximized, respectively, according to the sensitive attributes. Pursuing this second objective would help distribute similar objects (in terms of sensitive attributes) across different clusters, thus helping the formation of diverse clusters. This is beneficial to ensure that the distribution of groups defined on sensitive attributes within each cluster approximates the distribution across the dataset.

The task of fair clustering can be mapped to a MIN-CC instance where the positive-type and negative-type weights, respectively, can be defined as follows:

$$w^+_{uv} := \psi^+\Big(\alpha^{\neg F}_N \cdot sim_{\mathcal{A}^{\neg F}_N}(u,v) + (1-\alpha^{\neg F}_N)\cdot sim_{\mathcal{A}^{\neg F}_C}(u,v)\Big) \tag{6}$$

$$w^-_{uv} := \psi^-\Big(\alpha^{F}_N \cdot sim_{\mathcal{A}^{F}_N}(u,v) + (1-\alpha^{F}_N)\cdot sim_{\mathcal{A}^{F}_C}(u,v)\Big) \tag{7}$$

where $\alpha^F_N = |\mathcal{A}^F_N|/(|\mathcal{A}^F_N| + |\mathcal{A}^F_C|)$ and $\alpha^{\neg F}_N = |\mathcal{A}^{\neg F}_N|/(|\mathcal{A}^{\neg F}_N| + |\mathcal{A}^{\neg F}_C|)$ are coefficients to weight similarities proportionally to the size of the involved set of attributes, $\psi^+ = exp(|\mathcal{A}^F|/(|\mathcal{A}^F| + |\mathcal{A}^{\neg F}|) - 1)$ and $\psi^- = exp(|\mathcal{A}^{\neg F}|/(|\mathcal{A}^F| + |\mathcal{A}^{\neg F}|) - 1)$ are smoothing factors to penalize correlation-clustering weights that are computed on a small number of attributes (which is usually the case for sensitive attributes, and hence negative-type weights), and $sim_S(\cdot)$ denotes any object similarity function defined over the subspace S of the attribute set.

Problem 2 (Attribute Selection for Fair Clustering). Given a set of objects $\mathcal{X}$ defined over the attribute sets $\mathcal{A}^F$, $\mathcal{A}^{\neg F}$, find maximal subsets $S^F \subseteq \mathcal{A}^F$ and $S^{\neg F} \subseteq \mathcal{A}^{\neg F}$, with $|S^F| \geq 1, |S^{\neg F}| \geq 1$, s.t. the correlation-clustering weights in Equations (6)–(7) satisfy the global-weight-bounds condition in Theorem 1.

Heuristics. Our first proposal to solve Problem 2 is a greedy heuristic, dubbed Greedy, which iteratively removes the attribute that leads to the correlation-clustering weights with the lowest target ratio until our global condition is satisfied. This algorithm runs in $\mathcal{O}(|\mathcal{X}|^2|\mathcal{A}|^2)$ time since, at each iteration, for each candidate attribute to be removed $\mathcal{O}(|\mathcal{X}|^2)$ similarities are computed to quantify the decrease of the target ratio. We also devised other heuristics which, like Greedy, remove one attribute at time, but exploit some easy-to-compute proxy measures to select the attribute that avoid the pairwise similarity computation for each candidate attribute. The Hlv (resp. Hmv) heuristic removes the least (resp. most) variable attribute where the variability is measured through normalized entropy for categorical attributes and with variation coefficient (capped to 1 if above 1) for numerical features. Hlv_B and Hmv_B, like the previous two heuristics, remove the least and most variable attribute, respectively, but the selection is constrained to the biggest set of features among $\mathcal{A}^F$ and $\mathcal{A}^{\neg F}$, in order to try to balance their size. Finally, Hlv_BW removes the least variable attribute from the set ($\mathcal{A}^F$ or $\mathcal{A}^{\neg F}$) which induces the highest average similarity value using the current weights, whereas Hmv_SW removes the most variable attribute from the set which induces the lowest average similarity value using the current weights. Note that all these heuristics (but Greedy) run in $\mathcal{O}(|\mathcal{X}|^2|\mathcal{A}|)$ time.

Table 3: Fair clustering results.

	#it	target ratio	$\%(w^+ > w^-)$	orig.-weights Min-CC obj.	avg. Eucl. fairness	avg. #clusts.	intra-clust $\mathcal{A}^{\neg F}$	intra-clust $\mathcal{A}^{F}$	inter-clust $\mathcal{A}^{\neg F}$	inter-clust $\mathcal{A}^{F}$	time (seconds)
Adult											
initial	–	1.086	90.34	1.1915E+08	0.082	77	0.699	0.672	0.378	0.181	–
Hlv	12	0.986	93.19	1.122659E+08	**0.031**	9	0.465	0.326	0.347	0.194	545.249
Hlv_B	12	0.765	78.09	1.119757E+08	0.039	69	0.608	0.547	0.375	0.184	529.674
Hmv	5	0.974	90.83	1.21187E+08	0.094	79	0.689	0.687	0.373	**0.203**	220.056
Hmv_B	4	0.936	87.39	1.25516E+08	0.109	905	0.963	0.96	0.377	0.199	**178.813**
Hlv_BW	5	0.963	83.17	1.343503E+08	0.152	1479	**0.969**	0.964	0.384	0.199	217.333
Hmv_SW	9	0.926	91.41	1.159874E+08	0.037	5	0.451	**0.308**	**0.329**	0.195	380.875
Greedy	2	0.967	92.36	**1.094787E+08**	0.036	32	0.668	0.654	0.361	0.195	595.610
Bank											
initial	–	1.612	98.84	7.738171E+07	0.019	9	0.593	0.466	0.413	0.083	–
Hlv	19	0.95	99.88	7.063441E+07	0.001	3	0.52	0.209	0.368	**0.082**	1289.785
Hlv_B	16	0.906	97.19	8.489668E+07	0.038	752	0.859	0.818	0.456	0.077	1223.205
Hmv	17	0.972	100.0	**7.032421E+07**	**0.0**	2	0.497	**0.136**	**0.151**	0.03	1254.341
Hmv_B	16	0.981	97.19	8.250374E+07	0.032	35	0.775	0.665	0.451	0.079	**1143.517**
Hlv_BW	17	0.984	92.87	1.163447E+08	0.095	1048	**0.997**	0.996	0.444	0.076	1212.091
Hmv_SW	17	0.972	100.0	**7.032421E+07**	**0.0**	2	0.497	**0.136**	**0.151**	0.03	1336.888
Greedy	13	0.981	99.57	7.240143E+07	0.006	3	0.508	0.371	0.381	0.076	11978.472
CreditCardCustomers											
initial	–	1.415	96.97	7.556837E+06	0.050	13	0.586	0.53	0.397	0.133	–
Hlv	18	0.935	75.51	1.234939E+07	0.121	4	0.452	**0.176**	0.402	0.114	75.252
Hlv _B	17	0.981	85.64	1.013557E+07	0.153	1210	**0.996**	0.994	0.414	0.113	78.471
Hmv	15	0.985	99.41	**6.674586E+06**	**0.002**	3	0.461	0.225	**0.343**	0.132	72.112
Hmv _B	13	0.977	97.37	7.498595E+06	0.045	12	0.601	0.559	0.402	**0.134**	**58.486**
Hlv_BW	16	0.926	85.81	9.636214E+06	0.125	571	0.986	0.982	0.409	0.123	75.484
Hmv_SW	15	0.985	99.41	**6.674586E+06**	**0.002**	3	0.461	0.225	**0.343**	0.132	72.109
Greedy	14	0.941	95.5	7.584107E+06	0.049	20	0.612	0.57	0.406	0.115	714.02
Student											
initial	–	1.042	96.79	4.307303E+04	0.034	4	0.568	0.479	0.315	0.17	–
Hlv	30	0.968	84.18	5.236701E+04	0.064	2	0.407	**0.213**	0.392	0.189	14.838
Hlv_B	22	0.967	70.09	5.828042E+04	0.143	11	0.581	0.459	0.392	0.190	10.551
Hmv	8	0.994	96.28	4.303145E+04	0.031	5	0.577	0.484	0.379	0.189	3.91
Hmv_B	8	0.974	96.94	**4.260863E+04**	**0.030**	5	**0.588**	0.507	0.364	0.184	3.809
Hlv_BW	22	0.967	70.09	5.828042E+04	0.143	11	0.581	0.459	0.392	0.190	10.923
Hmv_SW	3	0.938	94.97	4.382731E+04	0.035	4	0.561	0.446	**0.350**	0.188	**1.543**
Greedy	2	0.975	94.8	4.595454E+04	0.059	5	0.535	0.434	0.376	**0.193**	9.980

Data and results. We considered 4 real-world relational datasets: *Adult*,[6] *Bank*,[6] *CreditCardCustomers*,[7] and *Student*.[6] For each of them, we report in Table 1-(right) the number of objects, a pair of values corresponding to the count of non-sensitive and sensitive attributes, and a description of the latter.

Table 3 summarizes results achieved by each of the above heuristics, on the various datasets, according to the following criteria (columns from left to right): number of iterations at convergence, target ratio, percentage of pairs u, v having $w^+_{uv} > w^-_{uv}$; also, computed w.r.t. the full attribute space are: value of the objective function, average Euclidean fairness[8] (the lower, the better), average number of clusters, intra-cluster and inter-cluster similarities according to either the subset of sensitive attributes or the subset of non-sensitive attributes, and running time.[5] Euclidean and Jaccard similarity functions are used for numerical and categorical attributes, resp., and the overall similarity is obtained by linear

[6] https://archive.ics.uci.edu/ml/index.php

[7] https://www.kaggle.com/sakshigoyal7/credit-card-customers

[8] The average weighted by cluster-size of the per-attribute averages of the Euclidean distances between the frequency attribute vector computed over the set of objects of a cluster and the frequency attribute vector over the whole set of objects [1].

combination analogously to Eqs. (6)–(7). Note that higher values correspond to better performance for $\mathcal{A}^F$-based intra-cluster and $\mathcal{A}^{\neg F}$-based inter-cluster similarities, while the opposite holds for the other two measures. The first row in each table refers to the initial, full-attribute-space status of the relational network, as a *baseline*, whereby the global-weight-bounds condition is not satisfied.

Hlv_BW and Hlv_B tend to produce solutions that correspond to the highest (i.e., worst) value of the objective function and by far the highest clustering size; this should be ascribed to the fact that both heuristics favor the removal of the least variable attributes. By contrast, Hmv_SW and Hmv are the best performing in terms of objective function and, on average, also in terms of Euclidean fairness; moreover, they tend to produce very few clusters. Remarkably, while a higher number of clusters is found to be coupled with a worsening of the objective function, the opposite does not hold in general. Also, contrarily to the intuition that a higher percentage of pairs having $w^+ > w^-$ should favor the grouping into fewer clusters, we observed that an ordering of the clustering size is not aligned with the percentage ordering. As far as efficiency, Greedy tends to converge in less iterations, i.e., it removes fewer attributes than the other methods. In some cases (e.g., *Student*, *Adult*), this allows Greedy for compensating its expected higher cost per iteration. Hmv_B mostly provides the best time performance.

Notably, each method lowers the initial target ratio below 1 so as to satisfy the global condition, and the per-dataset best-performing method improves all intra-/inter-cluster similarities and Euclidean fairness w.r.t. the baseline.

6 Conclusions

We have studied for the first time global weight bounds in correlation clustering. We have derived a sufficient condition to extend the range of validity of approximation guarantees beyond local weight bounds, such as the probability constraint. Extensive experiments have attested the usefulness of our condition.

We believe this work offers a new perspective on correlation clustering which opens stimulating yet challenging opportunities for further research, such as investigating the role of M and γ constants, extending our results to other constraints (e.g., triangle inequality), and studying the by-product problem of feature selection guided by our condition.

For reproducibility purposes, we make source code and data available at: `https://github.com/Ralyhu/globalCC` and `http://people.dimes.unical.it/andreatagarelli/globalCC/`.

References

1. Abraham, S.S., P, D., Sundaram, S.S.: Fairness in clustering with multiple sensitive attributes. In: Proc. EDBT Conf. pp. 287–298 (2020)
2. Ahmadian, S., Epasto, A., Knittel, M., Kumar, R., Mahdian, M., Moseley, B., Pham, P., Vassilvitskii, S., Wang, Y.: Fair hierarchical clustering. In: Proc. NIPS Conf. (2020)
3. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Clustering without over-representation. In: Proc. ACM KDD Conf. pp. 267–275 (2019)

4. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Fair correlation clustering. In: Proc. AISTATS Conf. pp. 4195–4205 (2020)
5. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. JACM **55**(5), 23:1–23:27 (2008)
6. Ausiello, G., Marchetti-Spaccamela, A., Crescenzi, P., Gambosi, G., Protasi, M., Kann, V.: Complexity and approximation: combinatorial optimization problems and their approximability properties. Springer (1999)
7. Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., Wagner, T.: Scalable fair clustering. In: Proc. ICML Conf. pp. 405–413 (2019)
8. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56**(1), 89–113 (2004)
9. Bera, S.K., Chakrabarty, D., Flores, N., Negahbani, M.: Fair algorithms for clustering. In: Proc. NIPS Conf. pp. 4955–4966 (2019)
10. Bonchi, F., García-Soriano, D., Liberty, E.: Correlation clustering: from theory to practice. In: Proc. ACM KDD Conf. p. 1972 (2014)
11. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. JCSS **71**(3), 360–383 (2005)
12. Chawla, S., Makarychev, K., Schramm, T., Yaroslavtsev, G.: Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In: Proc. ACM STOC Symp. pp. 219–228 (2015)
13. Chen, X., Fain, B., Lyu, L., Munagala, K.: Proportionally fair clustering. In: Proc. ICML Conf. pp. 1032–1041 (2019)
14. Chierichetti, F., Kumar, R., Lattanzi, S., Vassilvitskii, S.: Fair clustering through fairlets. In: Proc. NIPS Conf. pp. 5029–5037 (2017)
15. Crescenzi, P.: A short guide to approximation preserving reductions. In: Proc. IEEE CCC Conf. pp. 262–273 (1997)
16. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. TCS **361**(2-3), 172–187 (2006)
17. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM TKDD **1**(1), 4 (2007)
18. Kleindessner, M., Awasthi, P., Morgenstern, J.: Fair k-center clustering for data summarization. In: Proc. ICML Conf. pp. 3448–3457 (2019)
19. Kleindessner, M., Samadi, S., Awasthi, P., Morgenstern, J.: Guarantees for spectral clustering with fairness constraints. In: Proc. ICML Conf. pp. 3458–3467 (2019)
20. Kollios, G., Potamias, M., Terzi, E.: Clustering large probabilistic graphs. IEEE TKDE **25**(2), 325–336 (2013)
21. Mandaglio, D., Tagarelli, A., Gullo, F.: In and out: Optimizing overall interaction in probabilistic graphs under clustering constraints. In: Proc. ACM KDD Conf. pp. 1371–1381 (2020)
22. Pandove, D., Goel, S., Rani, R.: Correlation clustering methodologies and their fundamental results. Expert Syst. **35**(1) (2018)
23. Puleo, G.J., Milenkovic, O.: Correlation clustering with constrained cluster sizes and extended weights bounds. SIAM J. Optim. **25**(3), 1857–1872 (2015)
24. Rösner, C., Schmidt, M.: Privacy preserving clustering with constraints. In: Proc. ICALP Colloq. vol. 107, pp. 96:1–96:14 (2018)
25. Shamir, R., Sharan, R., Tsur, D.: Cluster graph modification problems. Discret. Appl. Math. **144**(1-2), 173–182 (2004)
26. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: Proc. ACM-SIAM SODA Conf. pp. 526–527 (2004)
27. van Zuylen, A., Williamson, D.P.: Deterministic algorithms for rank aggregation and other ranking and clustering problems. In: Proc. WAOA. pp. 260–273 (2007)