# Improved Theoretical and Practical Guarantees for Chromatic Correlation Clustering

Yael Anava*
Technion, Haifa, Israel
yaelan@tx.technion.ac.il

Noa Avigdor-Elgrabli
Yahoo Labs, Haifa, Israel
noaa@yahoo-inc.com

Iftah Gamzu
Yahoo Labs, Haifa, Israel
iftah.gamzu@yahoo.com

## ABSTRACT

We study a natural generalization of the correlation clustering problem to graphs in which the pairwise relations between objects are categorical instead of binary. This problem was recently introduced by Bonchi et al. under the name of chromatic correlation clustering, and is motivated by many real-world applications in data-mining and social networks, including community detection, link classification, and entity de-duplication.

Our main contribution is a fast and easy-to-implement constant approximation framework for the problem, which builds on a novel reduction of the problem to that of correlation clustering. This result significantly progresses the current state of knowledge for the problem, improving on a previous result that only guaranteed linear approximation in the input size. We complement the above result by developing a linear programming-based algorithm that achieves an improved approximation ratio of 4. Although this algorithm cannot be considered to be practical, it further extends our theoretical understanding of chromatic correlation clustering. We also present a fast heuristic algorithm that is motivated by real-life scenarios in which there is a ground-truth clustering that is obscured by noisy observations. We test our algorithms on both synthetic and real datasets, like social networks data. Our experiments reinforce the theoretical findings by demonstrating that our algorithms generally outperform previous approaches, both in terms of solution cost and reconstruction of an underlying ground-truth clustering.

## Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; H.2.8 [**Information Systems**]: Database Applications – Data mining

*This work was done when the author was an intern at Yahoo Labs, Haifa, Israel.


*WWW 2015,* May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741629.

## Keywords

Approximation algorithms; Categorical similarity; Clustering; Edge-labeled graphs

## 1. INTRODUCTION

Clustering is a fundamental research area in the field of data mining. The goal in clustering is to partition a collection of objects into groups, using data about the objects and their relationships, so that the objects in each group are more similar to one another than objects in other groups. The *correlation clustering* problem [6] received a great deal of attention in recent years due to its natural formulation and its applicability in a wide range of scenarios. Indeed, correlation clustering is regarded to be a basic primitive in the data mining practitioner's toolkit [15]. An instance of correlation clustering consists of an undirected graph where each edge (relation between objects) is labeled as either positive (similar) or negative (dissimilar). The goal is to partition the nodes (objects) into clusters in a way that minimizes the total number of disagreements. A disagreement happens if a positive edge becomes inter-cluster or a negative edge becomes intra-cluster.

In this paper, we study the *chromatic correlation clustering* problem which was recently introduced by Bonchi et al. [9]. This problem is a natural generalization of correlation clustering in which the pairwise similarity relations between objects are *categorical* instead of binary. The chromatic correlation clustering problem is motivated by many real-world applications in data-mining and social networks. Some of these applications extend natural applications of correlation clustering, like community detection, link classification, entity de-duplication, and more. To provide more intuition, a generic scenario in social networks follows. A social network can be modeled by a graph whose nodes represent individuals and its edges stand for the relationships between those individuals. Clearly, individuals in the social network have relationships of various types, e.g., familial, professional, social, and more. The chromatic correlation clustering paradigm can be utilized to classify those individuals into disjoint communities consisting of entities that are most similar under some relationship type. Intuitively, each such community is a cohesive group of individuals that share similar characteristics, and as such, one can leverage holistic knowledge about the community to characterize its individuals. It seems important to note that characterizing an individual based on her direct relationships of maximal occurring type may not be sufficient as the affiliated enti-

ties may only have sparse relationships between them, and hence, they do not carry meaningful joint information.

The categorical community discovery scenario described above underlies many applications of interest whose goal is to better understand the structure of networks. One example of such application is the link classification problem (see, e.g., [28, 26, 11]). The objective in this problem is to predict the category between two entities using the information provided in the rest of the network. For instance, a link classification method can be employed to infer an unknown relationship between two individuals in a social network. This classification can be later utilized by recommendation or prediction systems for various tasks, like viral marketing [31] and personalization [5]. Another application is the entity de-duplication problem (see, e.g., [17, 4, 20]). The goal in this case is to identify and potentially remove entities that are duplicates of one another according to some measure or category. Note that duplicates in the data occur in practice when the entities are obtained through independent sources. An entity de-duplication method can be used, for instance, to support a person search on next generation search engines [22]. In this case, a search query for a person should produce a cluster consisting of the most representative webpages associated with that person. There are many other relevant applications for categorical community discovery such as query refinements in web search [32] and webpage segmentation [12].

## 1.1 Problem definition

An instance of the chromatic correlation clustering problem consists of a finite set $L$ of (positive) colors, a special (negative) color $\lambda \notin L$, and a colored graph, that is, a undirected graph $G = (V, E)$ with a coloring function $\chi : E \to L \cup \{\lambda\}$ on its edges. The objective is to find a colored clustering of the graph's nodes that minimizes the overall number of disagreements. More formally, we are interested in a partition $\mathcal{C} : V \to \{C_1, C_2, \ldots\}$ of the nodes into a collection of disjoint sets and an assignment of a color to each cluster $\psi : \{C_1, C_2, \ldots\} \to L$ that minimizes

$$|\{uv \in E : \mathcal{C}(u) \neq \mathcal{C}(v) \text{ and } \chi(uv) \neq \lambda\}| + |\{uv \in E : C = \mathcal{C}(u) = \mathcal{C}(v) \text{ and } \psi(C) \neq \chi(uv)\}| \ .$$

Note that the first term counts the inter-cluster disagreements, while the latter term counts the intra-cluster disagreements. Also remark that no cluster can be assigned to the special color $\lambda$, and therefore, any $\lambda$-colored edge within a cluster implies a disagreement. Intuitively, a $c$-colored edge for some $c \in L$ indicates similarity of type $c$ between two objects, and a $\lambda$-colored edge indicates a dissimilarity between two objects with respect to all types.

Chromatic correlation clustering was introduced by Bonchi et al. [9]. One can easily notice that the correlation clustering problem is the special case of this problem when $|L| = 1$, and thus, it immediately follows that the chromatic correlation clustering problem is NP-hard [6]. The study of correlation clustering under complete graphs has been given a considerable attention in the past (see, e.g., [6, 13, 2, 30, 15, 8]). Indeed, in many applications, the underlying graph is assumed to be complete as missing edges can be interpreted as negative. In the remainder of this paper, we assume that the input graph is complete, unless we specify otherwise. Bonchi et al. developed a randomized algorithm for complete graphs whose approximation guarantee is proportional to the maximum degree of the graph induced by the $L$-edges. This can be as bad as $O(|V|)$. They also suggested several additional heuristic algorithms for the problem.

## 1.2 Our results

We establish improved theoretical and practical guarantees for the chromatic correlation clustering problem. Our contributions can be summarized as follows:

- We develop three algorithmic approaches: (1) A fast and easy-to-implement constant approximation framework, which builds on a novel reduction of the problem to that of correlation clustering. In particular, this framework implies a linear-time randomized 11-approximation algorithm by utilizing the pivot-based algorithm of [2]. This is the first constant approximation result for the problem, significantly improving over a previous $O(|V|)$-approximation [9]; (2) A deterministic linear programming-based algorithm that achieves an improved approximation ratio of 4. Even though this algorithm cannot be considered to be practical, it extends our theoretical understanding of chromatic correlation clustering, and enables us to present additional insights and observations related to the problem; (3) A linear-time heuristic algorithm that is motivated by practical scenarios in which there is a ground-truth clustering that is obscured by noisy observations.

- We present an empirical evaluation of our new algorithms and previously leading algorithms on both synthetic and real datasets. Our experiments reinforce the theoretical findings by demonstrating that our algorithms generally outperform previous approaches, both in terms of solution cost and reconstruction of an underlying ground-truth clustering.

## 1.3 Related work

The task of clustering multi-dimensional networks has received some attention recently [37, 24]. This setting models multiple relationships between entities in a network by maintaining a collection of networks, each of which captures one type of relationships. Although this setting is conceptually close to ours, its objective is semantically far. Specifically, the common goal in this setting is to find a partition of the entities that is good with respect to *all* relationship dimensions at the same time. As it turns out, this reduces the problem to effectively integrating solutions for single-dimensional networks. Finding a single (multi-dimensional) clustering that agrees as much as possible with multiple (single-dimensional) clusterings has been studied under the name of clustering aggregation [18]. Note that our chromatic correlation clustering setting requires finding a partition of the entities that takes into account all the relationships together. In particular, it is not require to have guarantees with respect to each single-dimension network.

Bansal, Blum and Chawla [6] formalized the correlation clustering problem. They developed a constant factor approximation algorithm for complete graphs, and proved that this problem is NP-hard problem. Charikar, Guruswami and Wirth [13] improved the approximation guarantee to 4, and established that the problem is APX-hard. Subsequently, Ailon, Charikar and Newman [2] presented a randomized linear-time algorithm that attains an approximation ratio of 3, and a randomized LP-based approach that

guarantees 2.5-approximation. These algorithms were derandomized by van Zuylen and Williamson [38]. Giotis and Guruswami [19], and later Karpinski and Schudy [23], developed a PTAS for the case that the number of clusters is fixed. Charikar et al. [13] and Demaine et al. [16] independently showed that the more general setting of arbitrary graphs admits an $O(\log |V|)$-approximation algorithm. The maximization variant of correlation clustering has also received considerable attention. Currently, the best known result for complete graphs is a polynomial-time approximation scheme [6], while the best result for general graphs is a 0.766-approximation [35]. Note that the latter setting is APX-hard [13]. Other papers studying correlation clustering and its variants are [3, 29, 10, 1]. The correlation clustering problem has also been studied in the context of computational biology [7, 14, 33].

# 2. ALGORITHMS

In this section, we present several efficient approximation algorithms for the chromatic correlation clustering problem. We first develop a fast and easy-to-implement constant approximation framework. This framework builds on a novel observation that reduces an underlying instance of the problem to that of correlation clustering, while losing only a small constant factor in the approximation. One can then employ any of the constant factor approximation algorithms developed for correlation clustering and attain a constant factor approximation. For example, one can utilize the fast randomized pivot-based algorithm of Ailon, Charikar and Newman [2] to obtain a linear-time randomized 11-approximation algorithm. We emphasize that this is the first constant approximation result for the chromatic correlation clustering problem, and it significantly improves over the algorithm of Bonchi et al. [9] whose approximation guarantee is $O(|V|)$.

We complement the above results by developing a linear programming-based algorithm, which applies deterministic rounding, and achieves an improved approximation ratio of 4. This algorithm builds on the approach of Charikar, Guruswami and Wirth [13] for the correlation clustering problem, augmented with additional insights and observations that enable its generalization to our chromatic setting. Although this algorithm cannot be considered to be practical, it extends our theoretical understanding of the chromatic correlation clustering problem. In fact, this approach can be adjusted to the arbitrary graph setting, attaining $O(\log |V|)$-approximation. This matches the best known approximation for the special case of correlation clustering [13, 16].

We then develop a fast heuristic algorithm that is motivated by realistic scenarios in which the input instance admits a ground-truth clustering. Indeed, in many practical settings of chromatic correlation clustering, there is an unknown but correct way to partition the objects into clusters. Note that although this underling true clustering exists, it may not be readily identifiable from the data due to noisy similarity observations. Our algorithm is specifically designed to take these into consideration. We remark that similar assumptions have been suggested and utilized in the past with respect to the correlation clustering problem (see, e.g., [21, 34, 3, 30]). Notably, noisy model assumptions led to algorithms that successfully clustered real gene expression data [7].

## 2.1 A fast constant-factor approximation

We develop a fast and easy-to-implement randomized algorithm that attains an 11-approximation ratio. This result is achieved through a generic algorithmic framework that reduces an instance of chromatic correlation clustering to that of correlation clustering. The algorithmic framework consists of two main steps:

1. The algorithm initially modifies the input graph $G$. Specifically, the algorithm goes over the nodes of the graph one after the other, and associates a positive color with each one of them. The color $c_v \in L$ that is associated with a node $v$ is the positive color that appears the most among the edges that are incident to $v$ in $G$, breaking ties arbitrarily. Then, the algorithm modifies the color of all the edges incident to each node $v$ that are different than $c_v$ to the special negative color $\lambda$. We emphasize that at the end of this step, each edge of positive color $c \in L$ in the modified graph $G'$ must be incident to two nodes whose associated color was $c$. This implies that if we only focus on positive edges and neglect all the negative $\lambda$-edges from $G'$ then each connected component have exactly the same color, henceforth referred to as the color of the connected component.

2. The algorithm continues by executing any fast algorithm for the correlation clustering problem on each of the previously-mentioned connected components of the modified graph $G'$. All the clusters in each resulting clustering are given the positive color of the underlying connected component. Note that the ground sets of nodes of the resulting clusterings are pairwise disjoint, and therefore, the algorithm proceeds by uniting all those clustering to one clustering, which is then returned as the output of the algorithm.

In what follows, we focus on the above algorithm when the algorithm utilized in its second step is the randomized pivot-based algorithm of Ailon, Charikar and Newman [2]. This latter algorithm is fast and attains a 3-approximation for correlation clustering. It is also very simple: pick a pivot node uniformly at random, create a cluster that consists of this pivot and all the adjacent nodes that are connected to it using a positive edge, remove those nodes from the graph, and repeat as long as there are still nodes that are not clustered. We like to emphasize that the properties of the pivot-based algorithm enable us to directly employ it to the entire modified graph, while treating all positive color edges identically, without the need to identify each of its connected components first. This clearly simplifies our approach. However, note that even if we were required to identify the connected components as part of the framework, the asymptotic time complexity of the algorithm would not have changed since identification of connected components in a graph can be done in linear-time. A formal description of our algorithm, Reduce-and-Cluster, appears below.

### 2.1.1 Analysis

We begin by analyzing the approximation guarantee of our algorithmic framework. We emphasize that our analysis is generic in the sense that it is independent of the concrete correlation clustering algorithm employed in the second step. Let $\mathrm{ALG}_\chi$ and $\mathrm{OPT}_\chi$ be the cost of the solution generated by our algorithm and the optimal algorithm

**Algorithm 1** Reduce-and-Cluster

**Input:** An undirected graph $G = (V, E)$ with an edge coloring $\chi : E \to L \cup \{\lambda\}$
**Output:** A clustering $\mathcal{C} : V \to \{C_1, C_2, \ldots\}$ with a coloring $\psi : \{C_1, C_2, \ldots\} \to L$

▷ Step 1: edges color modification
1: **for** $v \in V$ **do** $c_v \leftarrow \text{argmax}_{c \in L} |\{u \in V : \chi(uv) = c\}|$
2: **for** $uv \in E$ **do**
3: **if** $(\chi(uv) \neq c_u$ or $\chi(uv) \neq c_v)$ **do** $\chi(uv) \leftarrow \lambda$
▷ Step 2: pivot-based correlation clustering
4: $i \leftarrow 1$
5: **while** $V \neq \emptyset$ **do**
6: $v \leftarrow$ uniformly random node in $V$
7: $\Gamma(v) \leftarrow \{u \in V : \chi(uv) \neq \lambda\}$
8: $C_i \leftarrow \{v\} \cup \Gamma(v)$; $\psi(C_i) \leftarrow c_v$
9: $V \leftarrow V \setminus C_i$; $E \leftarrow E \setminus \{uw : u \in C_i\}$; $i \leftarrow i + 1$

with respect to the input graph $G$, respectively. Similarly, let $\text{ALG}_1$ and $\text{OPT}_1$ be the cost of the solution generated in the second step of the algorithm and the optimal correlation clustering algorithm for the graph $G'$ obtained after the edges color modification, respectively. Note that $\text{ALG}_\chi$ and $\text{ALG}_1$ correspond to the same clustering, but $\text{ALG}_\chi$ also assigns a color to each cluster. Finally, let $d_c(v)$ be the number of edges of color $c \in L$ adjacent to node $v$ in $G$, and let $d_{\min}(v) = \sum_{c \neq c_v} d_c(v)$ be the total number of edges that are incident to $v$ but have a different (positive) color than the leading color of the edges of $v$, as defined in line 1 of the algorithm.

LEMMA 2.1. $\text{ALG}_\chi \leq \text{ALG}_1 + \sum_{v \in V} d_{\min}(v)$ .

PROOF. Consider our output clustering when applied to the color modified graph $G'$. Notice that the cost of our colored clustering on $G'$ is identical to the cost of the correlation clustering solution on $G'$ when neglecting the colors of the edges (namely, $\text{ALG}_1$). This follows as the color assignment to each cluster is the same as the color of all positive edges incident to this cluster's nodes. This latter claim results as any cluster that the correlation clustering algorithm (and in particular, the pivot-based algorithm) creates is confined within a connected component of positive edges in $G'$, and all the positive edges forming a connected component have the same color which the algorithm later assigns as the color of that cluster. Thus, the clusters coloring does not incur additional cost.

The proof of the lemma is completed by noticing that $\text{ALG}_\chi$ is not greater than the cost of that colored clustering with respect to $G'$ by more than $D = \sum_{v \in V} d_{\min}(v)$. This is straight-forward since the color modification step changes the color of at most $D$ edges, and those edges may imply an additional cost of at most $D$. □

LEMMA 2.2. $\text{OPT}_1 \leq \text{OPT}_\chi + \sum_{v \in V} d_{\min}(v)$ .

PROOF. Let $\text{OPT}_1^G$ be the cost of the optimal correlation clustering for the graph $G$ when all positive colored edges are simply treated as positive. Notice that $\text{OPT}_1^G$ is at most $\text{OPT}_\chi$. In particular, one can upper bound $\text{OPT}_1^G$ by utilizing the clustering underling $\text{OPT}_\chi$ (without the color assignment), while observing that the cost may only improve as some colored intra-cluster edges may not contribute to the cost in the regular corelation clustering setting.

Recall that $\text{OPT}_1$ is the cost that the optimal correlation clustering for the modified graph $G'$. The proof of the lemma is completed by observing that $\text{OPT}_1$ can be larger than $\text{OPT}_1^G$ by no more than $D = \sum_{v \in V} d_{\min}(v)$. This is straightforward since the color modification step changes the color of at most $D$ edges, and each of those edges may imply an additional cost. □

LEMMA 2.3. $\text{OPT}_\chi \geq \sum_{v \in V} d_{\min}(v)/2$ .

PROOF. Let us concentrate on some node $v$, and suppose that the optimal solution assigns it to a cluster with a color $c^*$. This implies that all the edges that are incident on $v$ and have a positive color $c \in L$ different than $c^*$ contribute to the cost of the solution, whether they are inter-cluster or intra-cluster. Clearly, the cost of those edges is lower bounded by $d_{\min}(v)$. The lemma now follows by summing the costs implied by those incident edges across all nodes, while noticing that the cost implied by any such edge may be counted twice; once for each of its incident nodes. □

We are now ready to prove the main result of this subsection.

THEOREM 2.4. *Given an $\alpha$-approximation algorithm for the correlation clustering problem, our algorithmic framework guarantees a $(3\alpha+2)$-approximation for chromatic correlation clustering.*

PROOF. Notice that

$$\begin{aligned} \text{ALG}_\chi &\leq \text{ALG}_1 + \sum_{v \in V} d_{\min}(v) \\ &\leq \alpha \cdot \text{OPT}_1 + \sum_{v \in V} d_{\min}(v) \\ &\leq \alpha \cdot \text{OPT}_\chi + (\alpha + 1) \sum_{v \in V} d_{\min}(v) \\ &\leq (3\alpha + 2) \cdot \text{OPT}_\chi \ , \end{aligned}$$

where the first inequity is due to Lemma 2.1, the second inequality results by the approximation guarantee of the correlation clustering algorithm, the third inequality is by Lemma 2.2, and the last inequality follows from Lemma 2.3.

We like to emphasize that the fact that the correlation clustering algorithm is applied sequentially to the connected components of the modified graph $G'$, and not to the entire graph, does not change its approximation ratio. This results from the way we constructed $G'$, which guarantees that the optimal solution does not create clusters that consist of nodes residing in different connected components. Specifically, assume for the sake of contradiction that there is a cluster in the optimal clustering that consists of nodes from different components. One can split that cluster into several clusters, each consisting of the originating cluster nodes residing in different components, and reduce the overall cost. It is easy to validate that the reduction in the cost is equal to the number of (negative) edges between the underlying nodes in different components. □

We can now use this result for the analysis of algorithm Reduce-and-Cluster. Recall that this algorithm is a special case of our framework when the randomized pivot-based algorithm is employed as its second step.

COROLLARY 2.5. *Algorithm Reduce-and-Cluster is a linear-time algorithm that achieves an approximation ratio of* 11 *for the chromatic correlation clustering problem.*

PROOF. We start with analyzing the time complexity of the algorithm. The edges color modification step of the algorithm can be easily implemented in $O(|V| + |E|)$-time since each node and each edge is accessed at most a constant number of times. The second step of the algorithm can also be implemented in $O(|V| + |E|)$-time using a simple but careful design. Specifically, generating a random order of the nodes of the graph can be done once in $O(|V|)$-time using the Fisher-Yates shuffle (see, e.g., [25]), and building the clusters, including updating the random order to reflect the removal of the nodes assigned to each cluster, can be done in $O(|V| + |E|)$-time as each node and each edge is only accessed constant number of times. In particular, once a cluster is formed, the underlying nodes and edges are not considered again in the remainder of the algorithm. In conclusion, the time complexity of the algorithm is linear. The proof is now completed by recalling that the pivot-based algorithm of Ailon, Charikar and Newman [2] achieves 3-approximation. □

## 2.2 An improved approximation via LP-based approach

We design a deterministic linear programming-based algorithm that attains a 4-approximation ratio. Our algorithm and analysis extends and refines the approach of Charikar, Guruswami and Wirth [13] for the correlation clustering problem. Note that this algorithm improves over the approximation guarantee suggested by Theorem 2.4. In particular, the current best algorithm for correlation clustering has an approximation ratio of 2.5, and thus, plugging it into our framework results in a 9.5-approximation for the chromatic setting. This 2.5-approximation algorithm employs LP-based techniques, but randomly rounds the fractional solution using the pivot-based algorithm [2]. We emphasize that our algorithm applies a deterministic rounding for the linear program, which makes our entire approach deterministic. We also like to note that it is not clear if the former LP-based approach [2] can be extended to the chromatic setting.

**The algorithm.** We begin by describing a linear program (LP) that captures the chromatic correlation clustering problem:

$$\text{minimize} \quad \sum_{\substack{uv\in E:\\ c=\chi(uv)\in L}} x^c_{uv} + \sum_{\substack{uv\in E:\\ \chi(uv)=\lambda}} \sum_{c\in L} (1 - x^c_{uv})$$

$$\begin{aligned}
\text{subject to} \quad & (1) \quad \sum_{c\in L} x^c_v = |L| - 1 && \forall\, v \in V \\
& (2) \quad x^c_{uv} \geq x^c_u, x^c_v && \forall\, u, v \in V, \forall\, c \in L \\
& (3) \quad x^c_{uv} + x^c_{vw} \geq x^c_{uw} && \forall\, u, v, w \in V, \forall\, c \in L \\
& (4) \quad x^c_{uv}, x^c_v \in [0, 1] && \forall\, u, v \in V, \forall\, c \in L
\end{aligned}$$

In an integral solution, the variable $x^c_v$ indicates if node $v$ is (not) in a cluster of color $c$, that is, if $x^c_v = 0$ then the $v$ is part of a cluster of color $c \in L$. Similarly, the variable $x^c_{uv}$ indicates whether an edge $uv$ is (not) in a cluster of color $c$, namely, if $x^c_{uv} = 0$ then $uv$ is part of a cluster of color $c \in L$. Note that we regard each pair ot nodes $uv$ to be unordered, so there is no difference between $x^c_{uv}$ and $x^c_{vu}$, and we use them interchangeably. Constraint (1) guarantees that each node is assigned to one colored cluster, i.e., it is far from exactly $|L|-1$ colors. Constraint (2) ensures that the edge $uv$ is in cluster of color $c$ only if both $u$ and $v$ are in a cluster of that color. Note that constraint (1) and (2) together imply that each edge is assigned to one color. Constraint (3) is the triangle inequality, guaranteeing that if edges $uv$ and $vw$ are in the same colored cluster then the edge $uw$ must also be in the same cluster. The objective function consists of two summations: the first one captures the cost implied by positive color edges, while the latter one captures the cost of implied by negative color edges. More specifically, a negative edge does not contribute a cost only if its incident nodes are far from each other with respect to any color, that is, $x^c_{uv} = 1$ for any color $c \in L$. Similarly, a positive edge does not contribute a cost only if its incident nodes are close to each other under the edge color.

We turn to describe our rounding procedure. For ease of presentation, we first identify an important property of the fractional solution.

OBSERVATION 2.6. *Given distinct nodes $u, v \in V$ and a color $c_1 \in L$ such that $x^{c_1}_{uv} < 1/2$ then $x^{c_2}_{uw} > 1/2$ for every node $w \in V \setminus \{u\}$ and a color $c_2 \in L \setminus \{c_1\}$.*

PROOF. Notice that constraint (2) ensures that $x^{c_1}_u \leq x^{c_1}_{uv} < 1/2$. In consequence, constraint (1) implies that

$$\sum_{c_2 \in L\setminus\{c_1\}} x^{c_2}_u > |L| - \frac{3}{2} \,,$$

A simple counting argument suggests that $x^{c_2}_u > 1/2$ for every $c_2 \in L \setminus \{c_1\}$. By using constraint (2) again, we get that $x^{c_2}_{uw} \geq x^{c_2}_u > 1/2$, for every node $w \in V \setminus \{u\}$. □

Our rounding procedure works in steps. At each step, the algorithms picks an arbitrary (pivot) node $v$ that is still not clustered. It considers all the unclustered nodes that are close to $v$ with respect to some positive color. A node $u \in V$ is regraded as *close* to $v$ under some color $c \in L$ if $x^c_{uv} < 1/2$. Notice that Observation 2.6 ensures that all the nodes close to $v$ must be close under the same color. Denote this color by $c_v$. Now, if the average distance between $v$ and the nodes under consideration with respect to $c_v$ is at most $1/4$, the algorithm creates a new cluster of color $c_v$ that consists of $v$ and the underlying nodes; otherwise, the algorithm creates a new singleton cluster that only consist of $v$ (with an arbitrary color). A formal description of this rounding procedure appears below.

**Algorithm 2** LP-Rounding

**Input:** An undirected graph $G = (V, E)$ with an edge coloring $\chi : E \to L \cup \{\lambda\}$, an optimal fraction solution $x$ for the linear program (LP)
**Output:** A clustering $\mathcal{C} : V \to \{C_1, C_2, \ldots\}$ with a coloring $\psi : \{C_1, C_2, \ldots\} \to L$

```
1: i ← 1
2: while V ≠ ∅ do
3:     v ← arbitrary node in V
4:     c_v ← argmin_{c∈L}{x^c_{vu} : u ∈ V}
5:     Γ(v) ← {u ∈ V : x^{c_v}_{vu} < 1/2}
6:     if Σ_{u∈Γ(v)} x^{c_v}_{vu} ≤ |Γ(v)|/4 do
7:         C_i ← {v} ∪ Γ(v); ψ(C_i) ← c_v
8:     else
9:         C_i ← {v}; ψ(C_i) ← c_v
10:    V ← V \ C_i; E ← E \ {uw : u ∈ C_i}; i ← i + 1
```

### 2.2.1 Analysis

We present a charging scheme that associates between the cost implied by our output clustering and the cost of the fractional LP solution. We make this calculation separately for each cluster. Consider a cluster $C$ of color $c = \psi(C)$ that was created by our algorithm for a pivot node $v$ when the set of remaining unclustered nodes were $V_C$.

*Case I: $C = \{v\}$ is a singleton cluster.* Notice that all costs in this case are due to inter-cluster disagreements of positive color edges. Let $U_L = \{u \in V_C : \chi(vu) \in L\}$ be the endpoints corresponding to those edges and let $|U_L|$ denote their cost. In case $x_{vu}^{\chi(vu)} \geq 1/2$ for every $u \in U_L$ then we are done since the LP cost of the positive color edges is at least $|U_L|/2$. Hence, in what follows, we may assume that there is at least one such variable whose value is less than $1/2$. Recall that Observation 2.6 implies that all the variables of edges incident on $v$ which have a value smaller than $1/2$ must have the color $c$. Let $\Gamma(v) = \{u \in V_C : x_{vu}^c < 1/2\}$ be that endpoints corresponding to those edges. Let $L_c = L \setminus \{c\}$, and observe that the LP cost of the of the edges corresponding to $\Gamma(v)$ is

$$\begin{aligned}
\Lambda_1 &\triangleq \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)=c}} x_{vu}^c + \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)\in L_c}} x_{vu}^{\chi(vu)} + \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)=\lambda}} \sum_{\ell\in L} (1 - x_{vu}^\ell) \\
&\geq \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)=c}} x_{vu}^c + \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)\in L_c}} x_{vu}^c + \sum_{\substack{u\in\Gamma(v):\\ \chi(vu)=\lambda}} x_{vu}^c \\
&= \sum_{u\in\Gamma(v)} x_{vu}^c > \frac{1}{4}|\Gamma(v)|,
\end{aligned}$$

where the first inequality holds since $x_{vu}^{\chi(vu)} > x_{vu}^c$ for every edge $vu$ with $\chi(vu) \in L_c$ by Observation 2.6, and since $\sum_{\ell\in L}(1 - x_{vu}^\ell) \geq 1 - x_{vu}^c > x_{vu}^c$ for every edge $vu$ with $x_{vu}^c < 1/2$. The last inequality follows as the singleton cluster was created since the condition on line 6 of the rounding procedure failed. Now, notice that the LP cost over all the edges incident on $v$ is at least

$$\sum_{u\in U_L\setminus\Gamma(v)} x_{vu}^{\chi(vu)} + \Lambda_1 > \frac{1}{2}|U_L \setminus \Gamma(v)| + \frac{1}{4}|\Gamma(v)| \geq \frac{1}{4}|U_L| \ ,$$

where the first inequality follows by recalling that $x_{vu}^{\chi(vu)} \geq 1/2$ for every $u \in U_L \setminus \Gamma(v)$.

*Case II: $C = \{v\} \cup \Gamma(v)$ is a non-singleton cluster.* We analyze the cost contribution of each edge that is incident to the cluster based on its type:

*(a) Edges with color $\ell \in L\setminus\{c\}$.* Consider an edge $uw \in E$ such that $u \in C$ and $\ell = \chi(uv) \in L \setminus \{c\}$. This edge implies either an intra-cluster or an inter-cluster disagreement, and hence, its cost is 1. Note that the contribution of this edge to the LP cost is $x_{uw}^\ell$. We next argue that the value of this variable is at least $1/2$. We consider two cases: if $u$ or $w$ is the pivot node $v$ then the argument follows by Observation 2.6, while recalling that $\ell \neq c$; otherwise, the argument follows since $x_{vu}^c < 1/2$, and thus, $x_{uw}^\ell \geq 1/2$, again by Observation 2.6. Note that above proof implies that our algorithm did not pay more than twice the cost that the LP solution paid for that edge. We like to emphasize that this gives us some room for flexibility since we only need this variable to have a value of at least $1/4$ to guarantee 4-approximation. Consequently, we can use up to half of the value of each such variable in later cases.

*(b) Intra-cluster $\lambda$-edges.* Consider an edge $uw \in E$ such that $u, w \in C$ and $\chi(uw) = \lambda$. Note that this edge contributes a cost of 1 to our solution, while its LP cost is $\sum_{\ell\in L}(1 - x_{uw}^\ell)$. We analyze the cost according to the following cases:

- If $u$ or $w$ is the pivot node $v$, we get that $x_{uw}^c < 1/2$. Therefore, the LP cost of that edge is at least $1-x_{uw}^c > 1/2$.
- If $x_{vu}^c \leq 3/8$ and $x_{vw}^c \leq 3/8$ then $x_{uw}^c \leq 3/4$ by the triangle inequality in constraint (3) of the linear program. Therefore, the LP cost of that edge is $1 - x_{uw}^c > 1/4$.
- We are left to deal with the case that at least one of $x_{vw}^c, x_{vu}^c \in (3/8, 1/2)$. Let us assume without loss of generality that $x_{vw}^c \geq x_{vu}^c$. We associate the cost of the edge $uw$ to node $w$. We next demonstrate that (at least $1/4$ of) the cost assigned to each node $w \in \Gamma(v)$ having $x_{vw}^c \in (3/8, 1/2)$ can be associated with the LP cost of its neighboring edges. Formally, given a node $w$ and a set $A \subseteq L \cup \{\lambda\}$, we define $U_w^A = \{u \in \Gamma(v) : \chi(uw) \in A$ and $x_{vw}^c \geq x_{vu}^c\}$. Note that the solution cost associated with $w$ can be identified by $|U_w^{\{\lambda\}}|$. Let $L_c = L \setminus \{c\}$, we concentrate on the edges $xw$ such that $x \in U_w^{\{\lambda\}} \cup U_w^{\{c\}} \cup U_w^{L_c}$, and analyze their LP cost. Recall that we may have utilized at most half of the LP cost of the edges $yw$ such that $y \in U_w^{L_c}$ in the analysis of case (a) above. Hence, we may use half of the LP cost of those edges in a charging argument related to the current case, and still maintain a 4-approximation. Accordingly, the (un-utilized) LP cost related to $w$ is at least

$$\begin{aligned}
\Lambda_2 &\triangleq \sum_{y\in U_w^{\{\lambda\}}} (1 - x_{yw}^c) + \sum_{y\in U_w^{\{c\}}} x_{yw}^c + \frac{1}{2}\sum_{y\in U_w^{L_c}} x_{yw}^{\chi(yw)} \\
&\geq \sum_{y\in U_w^{\{\lambda\}}} (1 - x_{vw}^c - x_{vy}^c) + \sum_{y\in U_w^{\{c\}}} (x_{vw}^c - x_{vy}^c) \\
&\quad + \frac{1}{2}\sum_{y\in U_w^{L_c}} x_{yw}^{\chi(yw)} \\
&\geq |U_w^{\{\lambda\}}| \cdot (1 - x_{vw}^c) + |U_w^{\{c\}}| \cdot x_{vw}^c \\
&\quad - \sum_{y\in U_w^{\{\lambda,c\}}} x_{vy}^c + \frac{1}{4}|U_w^{L_c}| \\
&\geq \frac{1}{2}|U_w^{\{\lambda\}}| + \frac{3}{8}|U_w^{\{c\}}| - \frac{1}{4}|U_w^{L\cup\{\lambda\}}| + \frac{1}{4}|U_w^{L_c}| \\
&\geq \frac{1}{4}|U_w^{\{\lambda\}}| \ .
\end{aligned}$$

The first inequality follows from the triangle inequality in constraint (3) of the linear program. The second inequality results since $x_{yw}^{\chi(yw)} \geq 1/2$ for every $y \in U_w^{L_c}$. The third inequality holds since $x_{vw}^c \in (3/8, 1/2)$, and

$$\sum_{y\in U_w^{\{\lambda,c\}}} x_{vy}^c \leq \sum_{y\in U_w^{L\cup\{\lambda\}}} x_{vy}^c \leq \frac{1}{4}|U_w^{L\cup\{\lambda\}}| \ .$$

Here, the last inequality is due to the fact that the average value of all $x_{vy}^c$ such that $y \in \Gamma(v)$ is at most

1/4 by the condition on line 6 of the rounding procedure. As the nodes $y \in U_w^{L\cup\{\lambda\}}$ correspond to a subset of those variables with the smallest values, it follows that their average value is also smaller than 1/4.

*(c) Inter-cluster c-edges.* Consider an edge $uw \in E$ such that $u \in C$, $w \notin C$ and $\chi(uw) = c$. Note that this edge contributes a cost of 1 to our solution, while its LP cost is $x^c_{uw}$. We analyze the cost according to the following cases:

- If $u$ is the pivot node $v$, we get that $x^c_{uw} > 1/2$. Thus, the LP cost of that edge is 1/2.
- If $x^c_{vw} \geq 3/4$ then $x^c_{uw} > 1/4$ by the triangle inequality in constraint (3) and the fact that $x^c_{vw} < 1/2$. Hence, the LP cost of that edge is at least 1/4.
- We are left to deal with the case that $x^c_{vw} \in [1/2, 3/4)$. Again, we will associate the cost of the edge $uw$ to node $w$. We next demonstrate that (at least 1/4 of) the cost assigned to each such node $w$ can be associated with the LP cost of its neighboring edges. We use similar notation to that of case (b), and define the set $U^A_w = \{u \in C \setminus \{v\} : \chi(uw) \in A\}$ for a node $w$ and a set $A \subseteq L\cup\{\lambda\}$. Note that the solution cost associated with $w$ can be identified by $|U^{\{c\}}_w|$. Let $L_c = L \setminus \{c\}$, by the same arguments as in case (b), one can derive that the (un-utilized) LP cost related to $w$ is at least

$$\begin{aligned}\Lambda_3 &\triangleq |U^{\{\lambda\}}_w| \cdot (1 - x^c_{vw}) + |U^{\{c\}}_w| \cdot x^c_{vw} \\ &\quad -\frac{1}{4}|U^{L\cup\{\lambda\}}_w| + \frac{1}{4}|U^{L_c}_w| \\ &\geq \frac{1}{4}|U^{\{\lambda\}}_w| + \frac{1}{2}|U^{\{c_i\}}_w| - \frac{1}{4}|U^{L\cup\{\lambda\}}_w| + \frac{1}{4}|U^{L_c}_w| \\ &= \frac{1}{4}|U^{\{c_i\}}_w|,\end{aligned}$$

where the inequality holds since $x^c_{vw} \in [1/2, 3/4)$.

We conclude by emphasizing that it is not hard to validate that our scheme never charges the cost of an LP variable more than once. In particular, the only set of edges whose LP cost may be used in two different cases (of the non-singleton setting) are edges that have a color in $L \setminus \{c\}$. We use at most half of the LP cost of those edges in case (a), and another half of their LP cost in cases (b) and (c). Note that the set of relevant edges in cases (b) and (c) are disjoint, and thus, we do not overcharge any edge.

THEOREM 2.7. *There is a polynomial-time LP-based algorithm that attains 4-approximation for the chromatic correlation clustering problem.*

We like to note that the LP-based approach can be extended to the weighted arbitrary graph setting in which there may be object relations that are not known, that is, neither positive nor negative. This generalization requires slight refinement of the ideas suggested in [13, 16], and hence, we defer the formalities to the final version of this paper. The approximation result in this case is $O(\log |V|)$, which matches the best known result for the special case of correlation clustering.

## 2.3 A deep clustering heuristic

We develop a fast heuristic algorithm that is motivated by real-life scenarios in which there is a ground-truth clustering that is obscured by noisy observations. Algorithm Deep-Cluster, formally described below, follows a similar outline to that of algorithm Reduce-and-Cluster from Section 2.1. Initially, the algorithm modifies the input graph by negating all positive edges incident on each node $v$ whose color is different than the leading color of $v$. It continues by picking a pivot node uniformly at random, and creating a cluster that consists of this pivot and all the adjacent nodes connected to it using a positive edge (i.e., first-level nodes). The algorithm then considers each of the nodes that are connected to the first-level nodes by a positive edge (i.e., second-level nodes), and greedily adds it to the underlying cluster if it improves the post-assignment cost. Once this step is completed, all the cluster nodes are removed from the graph, and the algorithm begins a new iteration by randomly picking a new pivot node.

**Algorithm 3** Deep-Cluster

**Input:** An undirected graph $G = (V, E)$ with an edge coloring $\chi : E \to L \cup \{\lambda\}$
**Output:** A clustering $\mathcal{C} : V \to \{C_1, C_2, \ldots\}$ with a coloring $\psi : \{C_1, C_2, \ldots\} \to L$

▷ Step 1: edges color modification
1: **for** $v \in V$ **do** $c_v \leftarrow \text{argmax}_{c\in L}|\{u \in V : \chi(uv) = c\}|$
2: **for** $uv \in E$ **do**
3: **if** $(\chi(uv) \neq c_u$ or $\chi(uv) \neq c_v)$ **do** $\chi(uv) \leftarrow \lambda$
▷ Step 2: pivot-based deep clustering
4: $i \leftarrow 1$
5: **while** $V \neq \emptyset$ **do**
6: $v \leftarrow$ uniformly random node in $V$
7: $\Gamma_1(v) \leftarrow \{u \in V : \chi(uv) \neq \lambda\}$ ▷ 1st-level
8: $C_i \leftarrow \{v\} \cup \Gamma_1(v)$; $\psi(C_i) \leftarrow c_v$
9: **for** $u \in \Gamma_1(v)$ **do**
10: $\Gamma_2(u) \leftarrow \{w \in V \setminus C_i : \chi(uw) \neq \lambda\}$ ▷ 2nd-level
11: **for** $w \in \Gamma_2(u)$ **do**
12: $\deg_w \leftarrow |\{x \in V : \chi(wx) \neq \lambda\}|$
13: $\Delta_{\text{cur}} \leftarrow |\{x \in C_i : \chi(wx) \neq \lambda\}|$
14: $\Delta_{\text{inter}} \leftarrow \deg_w - \Delta_{\text{cur}}$
15: $\Delta_{\text{intra}} \leftarrow |C_i| - \Delta_{\text{cur}}$
16: **if** $\Delta_{\text{cur}} > (\Delta_{\text{inter}} + \Delta_{\text{intra}})$ **do**
17: $C_i \leftarrow C_i \cup \{w\}$
18: $V \leftarrow V \setminus C_i$; $E \leftarrow E \setminus \{uw : u \in C_i\}$; $i \leftarrow i + 1$

The intuition behind the algorithm is straight-forward. Under the assumption that a chromatic correlation clustering scenario admits a ground-truth clustering, one main difficulty in identifying this clustering is that the noisy observations obscure the underlying cliques structure. Our algorithm targets this situation by adding an extra validation step to the basic pivot-based approach which mitigates the noise effects. Recall that the basic approach creates a cluster in each iteration that consists of all the nodes similar to some pivot node. Our algorithm augments this approach by finding additional nodes that, although appearing dissimilar to the pivot due to noise, have high similarity to the other cluster nodes.

We complete the description of the heuristic algorithm by pointing out that one can implement it in linear time in the input size. Specifically, it is easy to validate that the time

complexity of the color modification step of the algorithm and the operations related to the first-level nodes is linear. This can be established along the same line of argumentations used in the analysis of algorithm Reduce-and-Cluster. Thus, we are left to bound the time complexity of the operations related to the second-level nodes. A naïve implementation in this case may take quadratic time in the input size. Indeed, it is possible that each node $w$ will be considered $E_w^L$ times as a second-level node, and in each such case, it may take $O(E_w)$-time to calculate $\Delta_{\text{cur}}$, its intersection size with the cluster under consideration. Here, $E_w$ and $E_w^L$ are the number of edges and colored edges incident of $w$, respectively. Nevertheless, one can carefully design this step to run in linear time. This can be done by initially going over all first-level nodes, and identifying all its second-level nodes. During this step, we maintain a counter for each second-level node that is incremented every time that this node is explored by an adjacent first-level node. Notice that eventually, the counter associated with each second-level node holds its intersection size with the underlying cluster. Hence, the greedy decision whether to add that node to the cluster can be done in constant time. Note that in case that a second-level node is added to the cluster, we explore its edges and increment the counters of remaining second-level nodes in a similar manner. The crucial observation is that all edges that were explored during the above process are removed from the graph once the cluster is formed. Since they are not considered again in the remainder of the algorithm, we conclude that the overall number of operations related to second-level nodes during the algorithm is no more than $O(E)$.

# 3. EXPERIMENTAL EVALUATION

We present our empirical results for the algorithms Reduce-and-Cluster and Deep-Cluster, which we refer by RC and DC, respectively. We test our algorithms on both synthetically-generated graphs, and graphs that are derived from real datasets. We compare the results with the performance of the leading algorithms developed by Bonchi et al. [9]. Specifically, we focus on the Chromatic Balls algorithm and the Lazy Chromatic Balls heuristic, abbreviated CB and LCB, respectively. The code of those algorithms was obtained directly from the authors of [9]. Informally, CB is an algorithm that iteratively builds a cluster around a randomly chosen pivot *edge*, while LCB is a heuristic that is better suited for scenarios with small number of colors, and tries to minimize the risk of selecting bad pivots. Note that we decided not to present the results of the simple baseline approach and the Alternating Minimization heuristic from [9] since in all cases, at least one of the algorithms CB or LCB outperforms them. Our experiments reinforce the theoretical findings by demonstrating that our algorithms generally outperform previous approaches, both in terms of solution cost and reconstruction of an underlying ground-truth clustering.

## 3.1 Synthetic datasets

We evaluate the algorithms on several synthetic graphs, each of which is parameterized by the number of nodes $n$, number of clusters $k$, number of positive colors $|L|$, and a noise probability $p$. Each graph is generated by the following simple process: We first partition the $n$ nodes of a complete graph into $k$ clusters uniformly in random, and assign each cluster a random color $c \in L$. Specifically, for each intra-cluster edge, assign the color of the underlying cluster, and for each inter-cluster edge, we assign a color of $\lambda$. Then, we change the color of each edge independently at random with probability $p$ to some different random color in $L \cup \{\lambda\}$. This latter step models the noisy observations that commonly happen in practice.

Notice that the first step in our generation process creates an instance that admits a chromatic correlation clustering of zero cost. This clustering is referred to as the *ground-truth clustering*. As part of our experiments, we evaluate the performance of the algorithm in retrieving this true clustering. For this purpose, we use the overall $F$-measure for clustering (see, e.g., [36]). Given a ground-truth clustering $\mathcal{C}^* = C_1^*, C_2^*, \ldots C_{k^*}^*$, and a clustering solution $\mathcal{C} = C_1, C_2, \ldots C_k$, their $F$-measure is defined as follows.

$$F(\mathcal{C}, \mathcal{C}^*) = \frac{1}{n} \sum_{i=1}^{k^*} |C_i^*| \max_{1 \le j \le k} F_{ij} \ ,$$

where $F_{ij}$ is the similarity between the clusters $C_i^*$ and $C_j$, calculated by the standard $F$-measure. Specifically, $F_{ij} = (2P_{ij}R_{ij})/(P_{ij} + R_{ij})$, where $P_{ij} = |C_i^* \cap C_j|/|C_j|$ is the precision, and $R_{ij} = |C_i^* \cap C_j|/|C_i^*|$ is the recall. Note that $F(\mathcal{C}, \mathcal{C}^*) \in [0, 1]$, and that greater overall $F$-measure values indicate higher similarity between clusterings.

Our experiments cover many datasets with varying parameters. For ease of presentation and due to space limitations, we only report the results attained for one set of base parameters. Specifically, we present the results obtained by exploring different values for each underlying parameter, while keeping the remaining parameters fixed at their base values. Our measurements indicate the average values attained for 10 runs. We like to emphasize that the trends that we identify for the setting under consideration are also evident in all other tested settings, although they vary in the intensity. In particular, we observe that our algorithms outperform previous approaches on all tested datasets, both in terms of solution cost and reconstruction of an underlying ground-truth clustering, except maybe in some extreme cases.

Most of the trends exhibited above are rather intuitive. For example, the performance of all algorithms becomes worse as the noise increases, and their performance improves as the number of colors increases. Focusing on the graphs of varying number of ground-truth clusters, one can see that the $F$-measure of all algorithms decrease as the number of clusters increase. However, the solution costs graph exhibits a more interesting phenomenon in which all the cost curves of the algorithms show a convex behavior, some even presenting a slightly increased cost for higher number of clusters. We believe that this convex behavior is related to the graph structure. Clearly, as the number of clusters increase, their average size decreases and the graph becomes sparser. As smaller clusters are easier to lose in the noise and since sparsity makes clustering problems noisier in general, the algorithms performance is expected to deteriorate at some point. The concrete point in which this happens depends on the exact properties of the algorithm. One additional observation that seems worth noting is that our new algorithms, and especially, algorithm DC, are much more robust to noise than previous algorithms, having high $F$-measure even with high levels of noise.

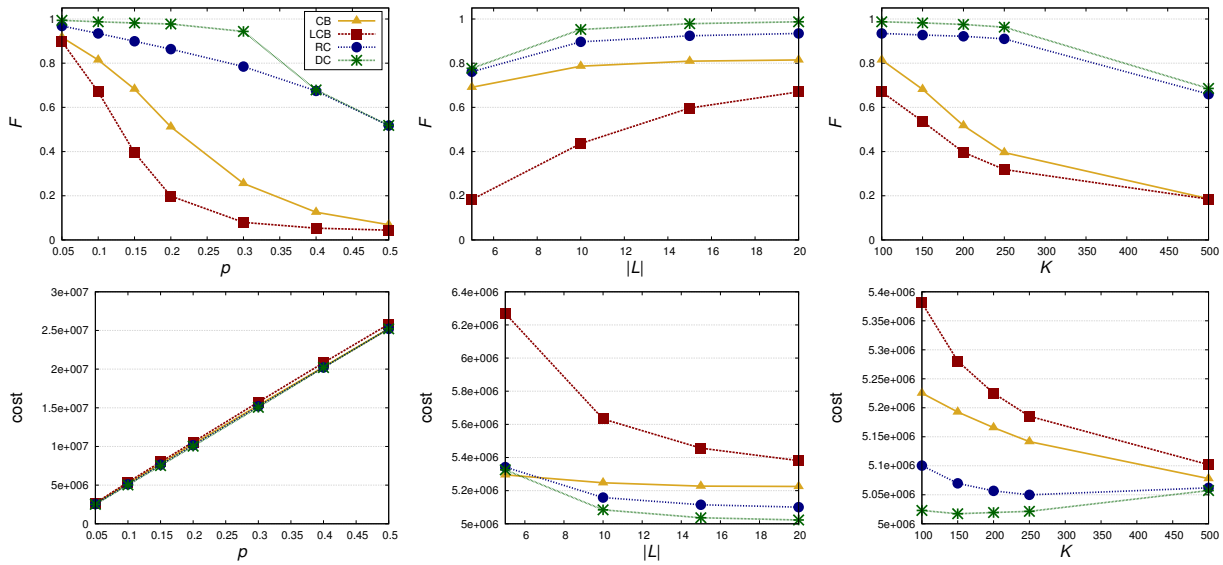


Figure 1: The results obtained for a base setting with parameters of $n = 10000$, $k = 100$, $|L| = 20$, and $p = 0.1$. The upper row shows the $F$-measure, while the lower row shows the solution cost. The results are presented with respect to varying level of noise (left most column), number of positive colors (middle column), and number of ground-truth clusters (right-most column).

## 3.2 Real datasets

We test our algorithms on four datasets that are publicly available. In particular, we experiment with two datasets (e.g., DBLP and STRING) that were identified and prepared by Bonchi et al. [9]. Note that all the datasets were processed so that the minimal node degree is at least 10. This was done to avoid sparse networks, which are of less interest. Indeed, the most interesting practical setting is when the graph is dense, and communities carry meaningful information.

| dataset | nodes | edges | colors | avg deg | max deg |
|---|---|---|---|---|---|
| DBLP | 73624 | 835414 | 100 | 22.69 | 502 |
| Facebook | 1072 | 42008 | 26 | 78.37 | 262 |
| Twitter | 4523 | 127068 | 35 | 56.18 | 623 |
| STRING | 18152 | 401582 | 4 | 44.25 | 264 |

**DBLP.** This dataset represents a co-authorship network of collaborations between researchers. It can be found at `http://dblp.uni-trier.de/xml/`. The dataset is an undirected graph in which nodes represent authors, the edges represent co-authorship of at least one paper, and there are 100 topic-based collaboration types. The topics were assigned by Bonchi et al. [9]. Specifically, they identified a collection of topics for all papers using a latent Dirichlet allocation topic-modeling, and then, assigned each edge the most prominent topic arising from the papers coauthored by the authors. A cluster of nodes in this case stands for a topic-coherent community of researchers.

**Facebook and Twitter.** These datasets are small snapshots from the social networks Facebook and Twitter. They are distributed by Stanford Network Analysis Project [27], and can be found at `http://snap.stanford.edu/data/`. Each dataset is an undirected graph in which nodes represent users, and edges represent friendship between users. There are 26 friendship types in Facebook and 35 friendship types in Twitter. We inferred those types from the users social circles. A circle is a social network feature that enable users to organize their internal networks by categorizing their friends. All users of a circle are assumed to share some mutual characteristics. A friendship edge between users is associated with some circle (category) only if both its incident nodes are identified within that circle. For edges with multiple associated categories, we picked one category arbitrarily. A cluster of nodes in this case represents a community of friends with dense relationships of specific type.

**STRING.** This dataset contains experimentally verified and computationally predicted protein-protein interactions (PPI) network. It is available at `http://string-db.org/`. The dataset is an undirected graph in which nodes represent proteins, edges represent interactions occurring between proteins, and there are 4 interaction types. A cluster of nodes in this case may designate a protein complex or a functional module. Although this dataset does not arise in the context of social networks, we decided to present it in order to highlight the applicability and effectiveness of our approach across diverse data domains.

The following table summarizes the performance of the algorithms on the real datasets. Our measurements represent average values attained for 50 runs. Similarly to the results on the synthetic data, our newly-developed algorithms, and especially, algorithm DC, generally outperform the previous algorithms. For example, algorithm DC achieves an improvement in the cost with respect to previously studied algorithms of at least 3.7% for the DBLP dataset, 11.1% for the Facebook dataset, 1.6% for the Twitter dataset, and 15.8% for the STRING dataset. Algorithm RC attains roughly the same costs as algorithm CB. As far as the runtime, we observe that all methods are very efficient, taking no more than few seconds on large graphs. Focusing on the different trends, we see that the our algorithms are slightly slower than algorithm CB, but are better than algorithm LCB as the number of colors and graph complexity increase. This

| | cost | | | | runtime (s) | | | |
|---|---|---|---|---|---|---|---|---|
| | CB | LCB | RC | DC | CB | LCB | RC | DC |
| DBLP | 617200 | 651210 | 617447 | 594292 | 1.07 | 5.71 | 1.73 | 1.81 |
| Facebook | 29979 | 29818 | 29581 | 26509 | 0.022 | 0.026 | 0.046 | 0.092 |
| Twitter | 108102 | 128495 | 111334 | 106348 | 0.092 | 0.082 | 0.163 | 0.224 |
| STRING | 160572 | 159722 | 158292 | 134464 | 0.38 | 0.36 | 0.48 | 0.63 |

is rather expected as our algorithms, and especially algorithm DC, are more complex than algorithm CB. We like to emphasize that the running time of our algorithms is linear in the input size. Finally, we believe that the running time of our algorithms can be improved by utilizing better data structures and optimizing the code.

# 4. CONCLUSIONS

We revisit the chromatic correlation clustering problem. This problem has many interesting applications, including community detection, entity de-duplication, and link classification. Our contribution is both theoretical and practical. From a theoretical point of view, our main contribution is a linear-time constant factor approximation algorithm that significantly improves over previous results. From a practical point of view, our main contribution is by demonstrating that our algorithms generally perform better than previous algorithms on both synthetic and real datasets. We think that it will be interesting and valuable to study the maximization variant of correlation clustering, as well as other correlation clustering variants, within a chromatic setting.

## Acknowledgments

We like to thank Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Antti Ukkonen [9] for kindly sharing their algorithms code and datasets.

# 5. REFERENCES

[1] N. Ailon, N. Avigdor-Elgrabli, E. Liberty, and A. van Zuylen. Improved approximation algorithms for bipartite correlation clustering. *SIAM J. Comput.*, 41(5):1110–1121, 2012.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.

[3] N. Ailon and E. Liberty. Correlation clustering revisited: The "true" cost of error minimization problems. In *36th ICALP*, pages 24–36, 2009.

[4] A. Arasu, C. Ré, and D. Suciu. Large-scale deduplication with constraints using dedupalog. In *25th ICDE*, pages 952–963, 2009.

[5] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *4th WSDM*, pages 635–644, 2011.

[6] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[7] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[8] F. Bonchi, D. García-Soriano, and E. Liberty. Correlation clustering: from theory to practice. In *20th KDD*, page 1972, 2014.

[9] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen. Chromatic correlation clustering. In *18th KDD*, pages 1321–1329, 2012.

[10] F. Bonchi, A. Gionis, and A. Ukkonen. Overlapping correlation clustering. In *11th ICDM*, pages 51–60, 2011.

[11] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. A correlation clustering approach to link classification in signed networks. In *25th COLT*, pages 34.1–34.20, 2012.

[12] D. Chakrabarti, R. Kumar, and K. Punera. A graph-theoretic approach to webpage segmentation. In *17th WWW*, pages 377–386, 2008.

[13] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.

[14] Z. Chen, T. Jiang, and G. Lin. Computing phylogenetic roots with bounded degrees and errors. *SIAM J. Comput.*, 32(4):864–879, 2003.

[15] F. Chierichetti, N. N. Dalvi, and R. Kumar. Correlation clustering in mapreduce. In *20th KDD*, pages 641–650, 2014.

[16] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2-3):172–187, 2006.

[17] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[18] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *TKDD*, 1(1), 2007.

[19] I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266, 2006.

[20] O. Hassanzadeh, F. Chiang, R. J. Miller, and H. C. Lee. Framework for evaluating clustering algorithms in duplicate detection. *PVLDB*, 2(1):1282–1293, 2009.

[21] T. Joachims and J. E. Hopcroft. Error bounds for correlation clustering. In *22nd ICML*, pages 385–392, 2005.

[22] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan. Web people search via connection analysis. *IEEE Trans. Knowl. Data Eng.*, 20(11):1550–1565, 2008.

[23] M. Karpinski and W. Schudy. Linear time approximation schemes for the gale-berlekamp game and related minimization problems. In *41st STOC*, pages 313–322, 2009.

[24] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *arXiv preprint arXiv:1309.7233*, 2013.

[25] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms, 2nd*

*Edition*. Addison-Wesley Longman Publishing Co., Inc., 1997.

[26] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Predicting positive and negative links in online social networks. In *19th WWW*, pages 641–650, 2010.

[27] J. Leskovec and A. Krevl. Snap datasets: Stanford large network dataset collection.

[28] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[29] C. Mathieu, O. Sankur, and W. Schudy. Online correlation clustering. In *27th STACS*, pages 573–584, 2010.

[30] C. Mathieu and W. Schudy. Correlation clustering with noisy input. In *21st SODA*, pages 712–728, 2010.

[31] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *8th KDD*, pages 61–70, 2002.

[32] E. Sadikov, J. Madhavan, L. Wang, and A. Y. Halevy. Clustering query refinements by user intent. In *19th WWW*, pages 841–850, 2010.

[33] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.

[34] R. Shamir and D. Tsur. Improved algorithms for the random cluster graph model. *Random Struct. Algorithms*, 31(4):418–449, 2007.

[35] C. Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *15th SODA*, pages 526–527, 2004.

[36] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[37] L. Tang, X. Wang, and H. Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.*, 25(1):1–33, 2012.

[38] A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Math. Oper. Res.*, 34(3):594–620, 2009.