Optimal Correlation Clustering via MaxSAT

Jeremias Berg and Matti Järvisalo
HIIT & Department of Computer Science, University of Helsinki, Finland

***Abstract*—We introduce an extensible framework for correlation clustering by harnessing the Maximum satisfiability (MaxSAT) Boolean optimization paradigm. The approach is based on formulating the correlation clustering task in an exact fashion as MaxSAT, and then using a state-of-the-art MaxSAT solver for finding clusterings by solving the MaxSAT formulation. Our approach allows for finding *optimal* clusterings wrt the objective function of the problem, extends to *constrained correlation clustering*—by allowing for easy integration of user-defined domain knowledge in terms of hard constraints over the clusterings of interest—as well as *overlapping correlation clustering*. First experiments on the scalability of the approach are presented.**

I. Introduction

Correlation clustering [1] is the well-studied [2], [3], [4], [5] NP-hard task of partitioning a given set of objects into groups based on a given pairwise similarity measure between the objects. This clustering paradigm is geared towards classifying data based on qualitative information—as opposed to quantitative information—of pairs of data points, arising from various applications in biology [6], social network analysis and information retrieval [7], [8]. Correlation clustering can be also seen as a form of agnostic learning [9], [1]. For our considerations, following the original definition given in [1], the problem input consists of an undirected graph over a set of nodes (representing the data points to be clustered), with positive and negative edges, indicating that two data points are *similar* or *dissimilar*, respective. The objective is to partition (i.e., cluster) the data points into clusters, minimizing the sum of the number of positive edges between different partitions and the number of negative edges within the partitions.

We introduce an extensible framework for correlation clustering that allows for finding *cost-optimal* clusterings wrt the objective function of the problem. We formulate the correlation clustering task in an exact fashion in the Boolean optimization paradigm of Maximum satisfiability (MaxSAT), and harness a state-of-the-art MaxSAT solver for finding optimal clusterings by optimally solving the MaxSAT formulation. This is in contrast with previously proposed approximation and local-search algorithms for correlation clustering which cannot give optimality-guarantees on the produced clustering. Our approach also extends to *constrained correlation clustering*—by allowing for easy integration of user-defined domain knowledge in terms of hard constraints over the clusterings of interest—as well as *overlapping correlation clustering* [7].

The main contributions of this paper are the following.

–We present a novel and extensible MaxSAT-based approach to optimal correlation clustering. To our best knowledge this is the first practical approach to *exactly* solving correlation clustering, i.e., to finding optimal clusterings wrt to the actual objective function of the problem. In contrast, previous work on correlation clustering has mainly focused on approximation algorithms and greedy local-search techniques which cannot in general find optimal clusterings.

–At the core of the approach, we present two different MaxSAT formulations of correlation clustering, as well as performance-improving optimizations to the encodings (Sections IV–VI). We provide initial experimental results on real-world data sets, comparing our approach to both an exact integer programming formulation [2], and an in-exact spectral clustering approach specialized to clustering protein sequences [10]. The first results show that our approach (i) can provide cost-optimal solutions and (ii) scales better than the exact integer programming formulation (Section VII).

–Our approach easily extends to the task of *constrained correlation clustering*, which allows for the user to specify the clusterings of one's interest by imposing hard used-defined constraints over the search space of clusterings. While approaches to constrained clustering have been proposed previously for different clustering paradigms [11], [12], [13], [14], [15], [16], to our best knowledge, this is the first approach that allows for seamless integration of user knowledge into the task of correlation clustering. We show experimentally that added user knowledge decreases the running time of our approach, and steers the obtained clusterings fast toward a predefined ground-truth clustering (Section VII-B).

–Our approach easily extends also to *overlapping correlation clustering* [7], hence giving a novel approach to *constrained overlapping correlation clustering*. We show experimentally that our approach is able to precisely reconstruct an existing ground truth clustering for UCI data sets, which could not be achieved using recently proposed greedy local search methods for overlapping correlation clustering [7] (Section VIII). Finally, we note that our approach has the potential to similarly cover other variants of correlation clustering, such as (constrained) chromatic correlation clustering [8].

II. Correlation Clustering

A. Problem Definition

For defining the problem of correlation clustering, we follow [17], [7]. However, while the original definition [17] is restricted to complete knowledge of the pairwise (dis)similarities of data points (in other words, to complete graphs), we allow also partial (dis)similarity information.

A correlation clustering instance consists of a set $V = \{v_1, \ldots, v_N\}$ of data points, and a binary *similarity* function $s\colon E \to \{0, 1\}$ over a subset $E \subset V \times V$ of the ordered

pairs of the data points. We assume that s is symmetric, i.e., that $s(v_i, v_j) = s(v_j, v_i)$ for any two data points v_i, v_j. Two data points v_i, v_j are considered (dis)similar if $s(v_i, v_j) = 1$ (if $s(v_i, v_j) = 0$). A correlation clustering instance (V, s) can be interpreted as an undirected graph with the set V of nodes and two types of labelled edge relations: $E^+ = \{\{v_i, v_j\} \mid s(v_i, v_j) = 1\}$ (representing similar pairs of nodes) and $E^- = \{\{v_i, v_j\} \mid s(v_i, v_j) = 0\}$ (representing dissimilar pairs of nodes). Any function $cl\colon V \to \mathbb{N}$ is a solution to the correlation clustering instance, representing a clustering of the data points into clusters indexed with natural numbers. In correlation clustering, the objective is to cluster the data points in a way that correlates as well as possible with s, i.e, to find a function $cl\colon V \to \mathbb{N}$ minimizing the cost function

$$G(cl) = \sum_{\substack{(v_x, v_y) \in E \\ cl(v_x) = cl(v_y)}} (1 - s(v_x, v_y)) + \sum_{\substack{(v_x, v_y) \in E \\ cl(v_x) \neq cl(v_y)}} s(v_x, v_y). \tag{1}$$

A clustering cl of V is *optimal* iff $G(cl) \leq G(cl')$ for any clustering cl' of V.

B. Correlation Clustering as Integer Programming

Correlation clustering is an NP-hard optimization problem [4]. Previous work on algorithms for correlation clustering has focused on approximation and greedy local search algorithms which cannot in general provide optimal solutions to the problem. However, an exact integer programming formulation of correlation clustering [2] has been used as a basis of approximating the problem. Given a correlation clustering instance (V, s), the integer program uses binary indicator variables $x_{ij} \in \{0, 1\}$, where $i < j$, with the interpretation that two data points $v_i, v_j \in V$ are in the same cluster (in different clusters) if $x_{ij} = 1$ ($x_{ij} = 0$). Using these variables, the set of optimal solutions to the following integer program represent the set of optimal clusterings of V under s [2].

$$\text{MINIMIZE} \quad \sum_{s(v_i, v_j) = 0} x_{ij} - \sum_{s(v_i, v_j) = 1} x_{ij}$$

$$\text{where} \quad x_{ij} + x_{jk} \leq 1 + x_{ik} \text{ for all distinct } i, j, k$$

$$x_{ij} \in \{0, 1\} \text{ for all } i, j \text{ s.t } i \neq j.$$

The purpose of the *transitivity constraint* $x_{ij} + x_{jk} \leq 1 + x_{ik}$ is to ensure a *well-defined* clustering: For any $(v_i, v_j, v_k) \in V \times V \times V$, each of the points v_i, v_j, v_k must belong to exactly one cluster, and hence it follows that if points v_i, v_j are assigned to the same cluster and points v_j, v_k are assigned to the same cluster, by transitivity then points v_i, v_k should also be assigned to the same cluster. Stated as a linear constraint with the defined variables we require that if $x_{ij} + x_{jk} = 2$ then $x_{ik} = 1$, which is exactly what the constraint enforces.

To the best of our knowledge, attempts to optimally solve this integer program have not so far been reported. In the paper, we present experimental results for correlation clustering real-world data sets by optimally solving this integer programming formulation using the state-of-the-art integer programming solver CPLEX. Before this, we will next present two alternative declarative formulations of correlation clustering as Maximum satisfiability (MaxSAT). It turns out that our MaxSAT formulations allow for optimally clustering notably larger real-world data sets than the integer programming approach.

III. Maximum Satisfiability (MaxSAT)

Before describing our MaxSAT formulations of optimal correlation clustering, this section shortly reviews necessary basic concepts on Maximum satisfiability (see e.g. [18]).

For a Boolean variable x, there are two literals, x and $\neg x$. A clause is a disjunction ($\vee$, logical OR) of literals and a truth assignment is a function from Boolean variables to $\{0, 1\}$. A clause C is satisfied by a truth assignment τ ($\tau(C) = 1$) if $\tau(x) = 1$ for a literal x in C, or $\tau(x) = 0$ for a literal $\neg x$ in C. A set F of clauses is satisfiable if there is an assignment τ satisfying all clauses in F ($\tau(F) = 1$), and unsatisfiable ($\tau(F) = 0$ for any assignment τ) otherwise. A partial MaxSAT instance $F = (F_h, F_s)$ consists of two sets F_h, F_s of clauses. The clauses in F_h are *hard*, and the ones in F_s *soft*. Any truth assignment τ that satisfies F_h is a *solution* to F. The *cost* of τ for F, denoted by $\text{COST}(F, \tau)$, is the number of clauses in F_s not satisfied by τ. A solution τ is (globally) *optimal* for F if $\text{COST}(F, \tau) \leq \text{COST}(F, \tau')$ holds for any solution τ' to F. The cost of the optimal solutions of F is denoted by $\text{OPT}(F)$. Given a partial MaxSAT instance F, the partial MaxSAT problem asks to find an optimal solution to F. For simplicity, we will from here on drop the term "partial" when referring to partial MaxSAT instances.

MaxSAT is a viable approach to finding globally optimal solutions to various optimization problems. In general, the MaxSAT-based approach has two steps. First, the problem is encoded as a MaxSAT instance F in a way that any optimal solution to F can be mapped to an optimal solution of the original problem. Then, an off-the-shelf MaxSAT solver is used to find an optimal solution to the MaxSAT instance. In this work, we extend the application domains of MaxSAT to optimally correlation clustering real-world data. The basic idea behind both of our MaxSAT formulations of correlation clustering, is that hard clauses are used to enforce that any solution to the MaxSAT instance represents a well-defined clustering (i.e., a mapping $cl\colon V \to \mathbb{N}$; recall Sect. II-A). The set of soft clauses are then used to encode the cost function in a faithful way, so that each solution to the MaxSAT instance can be mapped into a clustering with exactly the same cost.

IV. A MaxSAT Formulation

Our first MaxSAT formulation ("Encoding 1") of correlation clustering can be viewed as a reformulation of the integer programming formulation (recall Sect. II-B) in terms of MaxSAT. However, it turns out (as we will show in the experimental evaluation) that the MaxSAT formulation allows one to optimally solve notable larger data sets than the integer programming formulation.

Similarly as in the integer programming formulation, we use indicator variables x_{ij}, where $i < j$, with the interpretation

that $x_{ij} = 1$ iff points v_i and v_j belong to the same cluster. Using these variables, Encoding 1 forms the MaxSAT instance $F^1 = (F_h^1, F_s^1)$ summarized in Figure 1.

Hard Clauses F_h^1:	$(\neg x_{ij} \vee \neg x_{jk} \vee x_{ik})$	for all $(v_i, v_j, v_k) \in V^3$ where i, j, k are distinct
Soft Clauses F_s^1:	(x_{ij})	for all $s(v_i, v_j) = 1$
	$(\neg x_{ij})$	for all $s(v_i, v_j) = 0$

Fig. 1. MaxSAT instance $F^1 = (F_h^1, F_s^1)$ produced by Encoding 1.

We next describe the different parts of F^1 in detail.

A. Hard Clauses

The hard clauses F_h^1 of Encoding 1 are a clausal formulation of the transitivity constraints $x_{ij} + x_{jk} \leq 1 + x_{ik}$ for all distinct i, j, k in the integer program. In terms of propositional logic, these can be stated as $(x_{ij} \wedge x_{jk}) \rightarrow x_{ik}$, which in clausal form corresponds to

$$\mathrm{T}(v_i, v_u, v_j) := ((\neg x_{ij} \vee \neg x_{jk} \vee x_{ik}).)\,.$$

B. Soft clauses

The soft clauses F_s^1 encode the cost function: each pair $(v_i, v_j) \in E$ for which $s(v_i, v_j) = 0$ (resp., $(v_i, v_j) = 1$) that are (not) assigned to the same cluster should correspond to exactly one soft clause being left unsatisfied. This is achieved simply by introducing the soft clause $(\neg x_{ij})$ (resp., (x_{ij})) for each such pair.

C. Constructing a Clustering from a MaxSAT Solution

Any solution τ to F^1 represents a valid clustering cl_τ of V, constructed as follows: Assign point $cl_\tau(v_1) = 1$ and $cl_\tau(v_j) = 1$ whenever $\tau(x_{1j}) = 1$. Recursively, for the smallest i for which the point v_i is not assigned yet by cl_τ, and let $cl_\tau(v_i) = 2$ and $cl_\tau(v_k) = 2$ whenever $\tau(x_{ik}) = 1$. Iterate this process until there are no unassigned points left. Since the hard transitivity constraints must be satisfied by τ, this process will not create any conflicts, and each point will be assigned into exactly one cluster. Furthermore, it follows that the optimal solutions of F^1 correspond to the optimal clusterings of V. The correctness of Encoding 1 can hence be formalized as follows; this follows from the observation of the fact that Encoding 1 is a reformulation of the integer programming formulation, and the fact that each unsatisfied clause corresponds to exactly one pair of points that incur a cost to that clustering.

Theorem 1: Given a set of data points $V = \{v_1, \ldots, v_N\}$, a subset $E \subset V \times V$, and a similarity function $s\colon E \rightarrow \{0, 1\}$, let $F^1(V, s)$ be the MaxSAT instance produced by Encoding 1. The clustering $cl_{\tau *}\colon V \rightarrow \mathbb{N}$ constructed from an optimal solution τ^* to F is an optimal clustering of V under s. ■

We note that Encoding 1 does not require a predefined number of clusters. This is avoided by the definition of the x_{ij} variables, interpreted as pairwise indicator variables for two data points v_i, v_j being assigned to the same cluster. However, Encoding 1 is not very compact, due to the fact that the number of clauses encoding the transitivity constraints is cubic in the number of data points. Next, we will present an alternative MaxSAT formulation that is more compact that Encoding 1.

V. An Alternative MaxSAT Formulation

Next we present a more compact encoding which, in contrast to Encoding 1, assumes an upper bound K on the amount of available clusters. This corresponds to a setting in which the problem is to find an optimal clustering of the data using at most K clusters. This version of correlation clustering has previously been studied in [5]. In practice, by simply solving Encoding 2 for different values of K, one can obtain optimal clustering for an arbitrary upper bound number on the number of clusters.

Encoding 2 uses $N \cdot K$ Boolean variables y_{ik}, where $i = 1..N$ (the number of data points) and $k = 1..K$ (the number of clusters). The interpretation of these variables is that $y_{ik} = 1$ iff point v_i belongs to cluster k. Furthermore, we employ two types of auxiliary variables, which are used for achieving a compact clausal encoding.
(i) A_{ijk}, where $i = 1..N$, $j = 2..N$, $i < j$, and $k = 1..K$, with the interpretation $A_{ijk} = 1$ iff points v_i and v_j are both assigned to cluster k. More formally, $A_{ijk} \leftrightarrow (y_{ik} \wedge y_{jk})$.
(ii) D_{ij}, where $i = 1..N$, $j = 2..N$, and $i < j$, with the interpretation that if $D_{ij} = 0$, then points v_i and v_j are not assigned to the same cluster. More formally, $\neg D_{ij} \rightarrow (\neg y_{ik} \vee \neg y_{jk})$ for each cluster k.

As with Encoding 1 the hard clauses limit the set of solutions to well-defined clusterings, and the soft clauses encode the cost function in a faithful way. However, the hard and soft clauses differ notable from those of Encoding 1.

Concretely, Encoding 2 forms the MaxSAT instance $F^2 = (F_h^2, F_s^2)$ summarized in Figure 2.

Hard Clauses F_h^2:	ExactlyOne(v_i)	for all $v_i \in V$
	HardSimilar(A_{ijk})	for all $s(v_i, v_j) = 1$ and $1 \leq k \leq K$
	HardDissimilar(v_i, v_j, k)	for all $s(v_i, v_j) = 0$ and $1 \leq k \leq K$
Soft Clauses F_s^2:	SoftSimilar(v_i, v_j)	for all $s(v_i, v_j) = 1$
	SoftDissimilar(v_i, v_j)	for all $s(v_i, v_j) = 0$

Fig. 2. MaxSAT instance $F^2 = (F_h^2, F_s^2)$ produced by Encoding 2.

We next describe the different parts of F^2 in detail.

A. Ensuring Well-defined Clusterings

The hard constraints ExactlyOne(v_i) enforce that each solution represents a well-defined clustering, by enforcing that each data point v_i is assigned into exactly one cluster k. In terms of the variables in the encoding, for each i exactly one of the variables $y_{i1}, \ldots, y_{iK}$ should be assigned to 1, i.e., ExactlyOne$(v_i) := \sum_{k=1}^{K} y_{ik} = 1$. A number of different encodings of this cardinality constraint as clauses have been previously developed [19]. In our experiments, we used the so-called *sequential encoding* [20].

B. Encoding Similarity

For a pair (v_i, v_j) of data points, the constraints $\text{HARDSIMILAR}(A_{ijk})$ for each $k = 1..K$ and $\text{SOFTSIMILAR}(v_i, v_j)$ together enforce the requirement that v_i, v_j are assigned to the same cluster, given that the soft constraint $\text{SOFTSIMILAR}(v_i, v_j)$ is satisfied. In terms of propositional logic, this requirement can be expressed as the formula $(y_{i1} \wedge y_{j1}) \vee (y_{i2} \wedge y_{j2}) \vee \ldots \vee (y_{iK} \wedge y_{jK})$. In order to the express this propositional formula as clauses, we employ the auxiliary variables A_{ijk}. In terms of propositional logic, the resulting hard constraint is $A_{ijk} \leftrightarrow (y_{ik} \wedge y_{jk})$, resulting in the clausal form

$$\text{HARDSIMILAR}(A_{ijk}) := (\neg A_{ijk} \vee y_{ik}) \wedge (\neg A_{ijk} \vee y_{jk}) \wedge (A_{ijk} \vee \neg y_{ik} \vee \neg y_{jk}).$$

The soft constraint, expressing that points v_i and v_j are assigned to the same cluster whenever $s(v_i, v_j) = 1$, can be encoded as a single clause:

$$\text{SOFTSIMILAR}(v_i, v_j) := (A_{ij1} \vee \cdots \vee A_{ijK}).$$

For some intuition, we note that if this clause is satisfied in a solution, then for some j, A_{ijk} is necessarily assigned to 1 in the solution. Since all hard clauses are satisfied in any solution, it follows that both points v_i and v_j will be assigned to cluster k, exactly as required. If points v_i and v_j are not assigned to the same cluster, then due to the hard constraints $A_{ijk} = 0$ for all k, and the soft clause is not satisfied.

C. Encoding Dissimilarity

For a pair (v_i, v_j) of data points, the constraints $\text{HARDDISSIMILAR}(v_i, v_j, k)$ for each $k = 1..K$ and $\text{SOFTDISSIMILAR}(v_i, v_j)$ together enforce the requirements that v_i, v_j are assigned to different clusters. This can be expressed by requiring for each cluster that at least one of v_i, v_j should not be assigned to that cluster, which in clausal form is expressed by $(\neg y_{ik} \vee \neg y_{jk})$ for a cluster k.

For the full encoding, this time we use the auxiliary variables D_{ij}, and define them in terms of propositional logic as $\neg D_{ij} \rightarrow (\neg y_{ik} \vee \neg y_{jk})$ for each cluster $k = 1..K$; that is, if $D_{ij} = 0$, then v_i and v_j are not assigned to the same cluster, which in clausal form can be expressed as

$$\text{HARDDISSIMILAR}(v_i, v_j, k) := (D_{ij} \vee \neg y_{ik} \vee \neg y_{jk}).$$

Now we can express the soft constraints that v_i and v_j should not be assigned to the same cluster simply as

$$\text{SOFTDISSIMILAR}(v_i, v_j) := (\neg D_{ij}).$$

For some intuition, we note that if the clause $(\neg D_{ij})$ is satisfied in a solution to F^2, then the clauses $(\neg y_{ik} \vee \neg y_{jk})$ have to also be satisfied for all j in the solution, and hence points v_i and v_j are not assigned to the same cluster. On the other hand, if v_i and v_j are assigned to the same cluster k, then the solution has to assign $D_{ij} = 1$ in order to satisfy the hard clause $(D_{ij} \vee \neg y_{ik} \vee \neg y_{jk})$, resulting in one unsatisfied clause, namely $(\neg D_{ij})$, exactly as required for representing the correlation clustering cost function faithfully.

D. Constructing a Clustering from a MaxSAT Solution

Given a solution τ to F^2, we can easily construct a corresponding well-defined clustering cl_τ of the data point by assigning each point v_i into the cluster k for which $\tau(y_{ik}) = 1$. Due to the hard constraints F_h^2, in any solution τ there is exactly one such k for every i. Especially, the clustering constructed from an optimal solution to F^2 will be an optimal clustering of the data, minimizing the correlation clustering objective function. This correctness of Encoding 2 can be formalized as follows.

Theorem 2: Given a set of data points $V = \{v_1, \ldots, v_N\}$, a subset $E \subset V \times V$, and a similarity function $s\colon E \rightarrow \{0, 1\}$, let $F^2(V, s)$ be the MaxSAT instance produced by Encoding 2 for a predefined number K of clusters. The clustering $cl_{\tau*}\colon V \rightarrow \{1, \ldots K\}$ constructed from an optimal solution τ^* to F is an optimal clustering of V under s over all clusterings $cl\colon V \rightarrow \{1, \ldots K\}$. ■

Intuitively, the result follows from the already discussed connections between the cost incurred by a clustering and the number of unsatisfied soft clauses in Encoding 2.

VI. Optimizations to the MaxSAT Encodings

In this section we describe specific optimizations to the MaxSAT encodings. These optimizations are not necessary for ensuring correctness of the encodings, but have a clear positive effect on the time it takes to solve especially the MaxSAT instances produced by Encoding 2, i.e., the time a MaxSAT solver needs for finding an optimal clustering.

1) Pruning Symmetric Solutions: The first optimization, specific to Encoding 2, prunes some of the symmetries within the solution space of clusterings. The solution space is highly symmetric: given any clustering cl of V of cost $G(cl)$, any permutation of the cluster indices is a well-defined clustering with the same cost $G(cl)$. Hence, for any clustering cl of V there exists another clustering cl' for which $G(cl) = G(cl')$ and $cl'(v_1) = 1$. This means that we need only search through clusterings where the first point is assigned to the first cluster. In practice we achieve this by substituting the hard constraint $\text{EXACTLYONE}(v_1)$ in F_h^2 with the following k hard clauses: (y_{11}) and $(\neg y_{1k})$ for $k = 2..K$. While this simple substitution only prunes away a small part of the symmetric solutions, it turned out that in practice it pays off in terms of solving time. Note that this optimization cannot be used for Encoding 1, since Encoding 1 does not allow to directly enforce data points being assigned to specific cluster indices.

2) Exploiting Erroneous Triangles: The second improvement is based on so-called *erroneous triangles* [17] which are triplets of points v_i, v_j, v_k for which $s(v_i, v_j) = s(v_j, v_k) = 1$ and $s(v_i, v_k) = 0$. Viewed as a subgraph, such three points form a triangle in which two edges are labeled with 1 and one with 0. An erroneous triangle represents what we call a *local conflict* in the input data: any well-defined clustering is forced to either assign points v_i, v_j or v_j, v_k into different clusters, incurring a cost of one, or to assign v_i, v_k into the same cluster, again incurring a cost of one. Hence, no matter how these three points are assigned to clusters, a cost of at

least 1 is incurred. Our key observation is that information on such local conflicts, based on a set of erroneous triangles found in the input data, can be encoded directly into the MaxSAT instance by modifying Encoding 2. This turned out to be very beneficial in terms of scalability in practice.

Intuitively, the idea is to lower the cost of the MaxSAT instance in a controlled way. This is done by explicitly encoding into the instance knowledge about the fact that each node-disjoint erroneous triangle in the input data explicitly contributes to the lower bound on the cost of optimal solutions by one. This results in improvements in the running time of a MaxSAT solver, as the solver does not need to prove the lower bound any more. We will now describe how the knowledge of erroneous triangles is used in modifying the MaxSAT instances produced by Encoding 2.

An erroneous triangle v_i, v_j, v_k corresponds to three soft clauses in the MaxSAT instance produced by Encoding 2; these clauses are $\text{SOFTSIMILAR}(v_i, v_j)$, $\text{SOFTSIMILAR}(v_j, v_k)$ and $\text{SOFTDISSIMILAR}(v_i, v_k)$. Now, we introduce fresh *relaxation variables* r_{ij}, r_{jk}, and r_{ik}, and replace in F_s^2 the soft clause $\text{SOFTSIMILAR}(v_i, v_j)$ by the (soft) clause $\text{SOFTSIMILAR}(v_i, v_j) \vee r_{ij}$, $\text{SOFTSIMILAR}(v_j, v_k)$ by $\text{SOFTSIMILAR}(v_j, v_k) \vee r_{ij}$, and $\text{SOFTDISSIMILAR}(v_i, v_k)$ by $\text{SOFTDISSIMILAR}(v_i, v_k) \vee r_{ij}$. The intuition is that, by assigning one of the three relaxation variables to 1 satisfies the corresponding original soft clause, essentially removing the local conflict due to the erroneous triangle, lowering the cost of optimal solutions by one. To ensure correctness (maintain the optimal solutions), we require that *exactly one* of the relaxation variables is assigned to one, resulting in the fact that every erroneous triangle results in lowering the cost of (optimal) solutions by exactly one. To achieve this, we add to F_h^2 a set of hard clauses which encode the cardinality constraint $r_{ij} + r_{jk} + r_{ik} = 1$.

Given a set of t *node-disjoint* erroneous triangles in the input data, Encoding 2 can be modified in the same way iteratively *for each* of the erroneous triangles, reducing the cost of (optimal) solutions by exactly t. A maximal set of node-disjoint erroneous triangles can be found in polynomial-time (in practice, in negligible time) from a given input data using a simple greedy algorithm which iteratively locates an erroneous triangle in the data (until none exist), and after that disregards the data points already contained in a found erroneous triangle. We note that erroneous triangles could also be applied to modify Encoding 1. However, we observed experimentally that mainly Encoding 2 benefits from this modification.

VII. Experimental Evaluation

In the following we provide results of an experimental evaluation of our MaxSAT-based approach (both Encoding 1 and Encoding 2). In the experiments, we used real-world data sets: clustering proteins based on their amino-acid sequences [10] and *overlapping correlation clustering* [7] of standard UCI data sets (reported in Section VIII).

Our first focus is on the protein data, for which we present extensive results comparing the scalability of the MaxSAT encodings with the integer programming formulation, and the quality of the (optimal) MaxSAT solutions found with solutions produced by the specialized protein correlation clustering algorithm SCPS [10] (based on spectral clustering). Especially, we demonstrate that our MaxSAT approach allows for scalable optimal correlation clustering under added user knowledge in the form of additional constraints over the preferred clusterings, imposed by domain experts. In Section VIII we show how our MaxSAT Encoding 2 can be easily modified to cover overlapping correlation clustering [7], and report on experimental results on the standard YEAST and EMOTION UCI data sets which show that our MaxSAT-based approach can *precisely* reconstruct an existing ground truth clustering, hence providing notable better solutions that a previously proposed greedy local search method [7] for overlapping correlation clustering, without using more computation time to find an optimal solution.

For solving the MaxSAT instances resulting from Encoding 1 and 2, we used the academic, off-the-shelf MaxSAT solver MaxHS [21]. For solving the integer programming formulation, we used the commercial state-of-the-art integer programming solver CPLEX from IBM. A timeout of 2 hours and a memory limit of 20 GB were enforced for all experiments.

A. Clustering Protein Sequences

Following [22], [10], we apply our MaxSAT-based approach to the task of clustering proteins to homologous groups under given pairwise similarities of their amino acid sequences. The data used for the experiments are data sets 1–4 from [10] (see http://www.paccanarolab.org/software/scps/). The data sets consist of 669 (data set 1 / D1) 586 (D2), 567 (D3), and 654 (D4) sequences, respectively. The similarity values provided with the data were computed using BLAST [23]. Originally, the values computed by BLAST are real numbers in the interval $[0, 1]$. In order to obtain a binary similarity function for applying our MaxSAT encodings, we round these values to the nearest number, either 0 or 1, following the OCCISECT set-intersection indicator algorithm applied in previous experimental work on correlation clustering [7]. With the data, a ground-truth clustering—a manually-crafted taxonomy of proteins—based on SCOOP [24] is provided: the ground truth of sets $D1, D2, D3, D4$ consists of $5, 6, 5, 8$ clusters respectively. In our experiments, the number of clusters in the ground truth was given as an upper bound on the number of clusters both for the SCPS algorithm and our MaxSAT Encoding 2.

1) Scalability on Unpruned Data Sets: Figure 3 reports the running time of MaxHS (for our MaxSAT encodings) and CPLEX (for the integer programming formulation) for an increasing number of points from the unpruned data set $D2$. The trends are similar for the other three data sets. Our MaxSAT approach scales clearly better than the integer programming approach. Surprisingly, this holds true even for Encoding 1, which is essentially a MaxSAT reformulation of the integer programming formulation. In fact, CPLEX runs out of memory

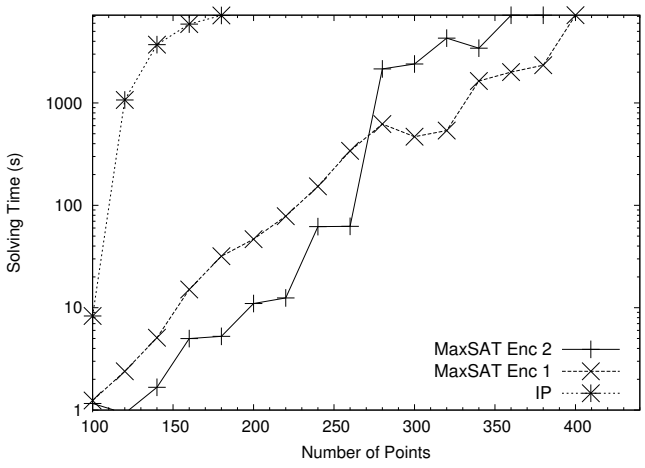

Fig. 3. Running times on different numbers of points from $D2$.

already with less than 200 data points. The figure suggests that Encoding 1 scales better than Encoding 2. However, the size of the MaxSAT instances produced by Encoding 1 limits its usability for larger data sets. Additionally, as shown in the following, Encoding 2 scales well for greater numbers of data points especially when integrating expert user knowledge as additional constraints, as well as after pruning the similarity function s in the input data.

2) Clustering Quality: Before turning to the task of integrating user knowledge, we report on a comparison of the MaxSAT approach and the SCPS algorithm, focusing on the quality of the produced clusterings. Clustering quality is measured both in terms of the standard measures precision and recall [10] (and F-score, based on precision and recall), as well as the actual fundamental correlation clustering cost function under minimization (cf. Eq. 1). Precision and recall are computed wrt a given ground-truth clustering (GT).

For a clustering cl of a set of data points V, let $P(cl) = \{(v_i, v_j) \in V^2 \mid cl(v_i) = cl(v_j)\}$ be the set of pairs of points that are assigned to the same cluster, and $g\colon V \to \mathbb{N}$ the ground-truth clustering. *Precision* $Pr(g, cl)$ and *recall* $R(g, cl)$ are defined as

$$Pr(g, cl) = \frac{|P(g) \cap P(cl)|}{|P(cl)|} \text{ and } R(g, cl) = \frac{|P(g) \cap P(cl)|}{|P(g)|}.$$

The F-score of two clusterings is the harmonic mean of precision and recall:

$$F(g, cl) = \frac{2}{\frac{1}{Pr(g,cl)} + \frac{1}{R(g,cl)}}.$$

Table I summarizes the quality of clusterings obtained for the largest number of data points still solvable in each data set and every method. We observe that according to recall and F-score, the clusterings produced by the SCPS algorithm resemble the ground-truth clustering more. On the other hand, the precision values for Encoding 1 are very good; in other words, given that a clustering produced by Encoding 1 assigns two points into the same cluster, they belong to the same cluster in the ground-truth clustering with higher probability than when using SCPS. Additionally, while both SCPS and our MaxSAT Encoding 2 were given the number of clusters in the ground clustering as the upper bound on the number of clusters, only Encoding 2 produced the same number of clusters as in the ground-truth clustering.

A clear advantage of the MaxSAT approach is observed by inspecting the actual costs of the clusterings found by the methods. Indeed, the MaxSAT approach can find *optimal* clusterings wrt the fundamental correlation clustering cost function under minimization, whereas the clusterings produced by SCPS often have notably higher cost and are thus clear worse in light of the cost function. Interestingly, the costs of ground-truth clusterings provided with the data turned out also to have rather high cost wrt the actual cost function.

We note that SCPS works on the original real-valued similarity data while the cost function Eq. 1 is defined for binary similarity values. However, we also checked the clustering reported by SCPS when given the binary similarity function as input: the costs of the produced clusterings were even higher in those cases.

Comparing Encodings 1 and 2, we observe that the clusterings produced with Encoding 1 generally have lower cost. However, Encoding 1 also produces clusterings with a high number of clusters, since the number of clusters is not bound. Clearly, the freedom to use any number of clusters results in clusterings with slightly lower cost, but at the same time, the number of clusters produced may be considered (too) high.

We next turn to the question of scalability in the presence of user knowledge and data pruning. Due to its size, Encoding 1 was not applicable for the larger numbers of data points, and hence we focus in the following on Encoding 2.

B. Integrating User Knowledge

Our exact MaxSAT approach has the advantage of easily adapting to integration user knowledge (UK), in the form of additional hard constraints on the clusterings of interest. The idea is to allow a domain expert to add information about which points should (not) be assigned to the same clusters. Being able to handle the integration of user knowledge allows for an iterative approach to correlation clustering, in which the user can iteratively refine their preferences on the characteristics of clusterings of interest. This idea of domain specific constrained clustering has been studied for other clustering settings [11], [12], [16], [13], [15], [14]. However, to the best of our knowledge, our approach is the first one for correlation clustering that easily adapts to additional user knowledge constraints.

In order to force a pair of points (v_i, v_j) to (not) be assigned to the same cluster, we simply need to add the clauses SOFTSIMILAR(v_i, v_j) (SOFTDISSIMILAR(v_i, v_j)) as *hard clauses* to the MaxSAT Encoding 2: since hard clauses are satisfied in every solution to the MaxSAT instance, the two points will (not) be assigned to the same cluster in any optimal clustering provided by the MaxSAT solver.

TABLE I
CLUSTERING QUALITY FOR LARGEST NUMBERS OF DATA POINTS SOLVABLE BY ALL METHODS. (SCPS / ENCODING 1/ ENCODING 2)

(SCPS/ Enc1/ Enc2)	# Clusters produced	# GT clusters	Precision	Recall	F-score	Cost	Cost of GT
D1 300P	5/28/5	5	0.68/**0.86**/0.6	**0.8**/0.69/0.65	**0.73**/0.72/0.63	616/**487**/491	591
D2 320P	6/21/6	6	0.81/**0.96**/0.73	**0.89**/0.74/0.74	**0.85**/0.83/0.73	883/**719**/786	869
D3 260P	4/20/5	5	0.92/**0.93**/0.78	**0.9**/0.49/0.53	**0.91**/0.64/0.63	626/**464**/470	623
D4 200P	7/28/8	8	0.68/**0.81**/0.53	**0.82**/0.48/0.48	**0.75**/0.60/0.51	184/**106**/**106**	177

In fact, in addition to integration of user knowledge within the MaxSAT approach (using Encoding 2) being simple, the scalability of the approach improves remarkably in the presence of even small amounts of user knowledge. We simulated the idea of input user knowledge by randomly sampling a percentage of all the information in the ground-truth clustering provided with the data sets. Figure 4 shows the running time of the MaxSAT solver with different amounts of UK added when clustering the data sets entirely. Already after adding 2% user knowledge, we were able to cluster the whole data sets. Furthermore, as shown in Table II, the inclusion of user knowledge dramatically improves the F-score of the clusterings obtained.

C. Scalability after Pruning

As noted for example in [7], the input data can often contain a lot of redundancy, justifying some form of pruning of the available similarity information. In this spirit, we experimented on pruned data sets, in which the similarity function (the edge relations of the input data, viewed as a graph) is pruned. Table III summarizes the results after pruning the data sets by including each of the edges in the similarity graph with 20% probability. For a comparison, even though the SCPS algorithm was given the full similarity data as input, the costs of the clusterings obtained using Encoding 2 are still

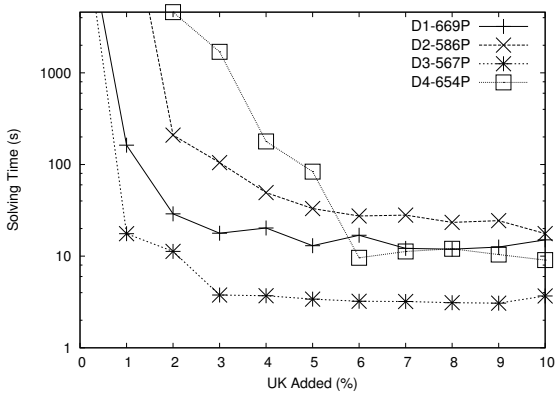

Fig. 4. MaxSAT running time with added user knowledge on unpruned data.

TABLE II
F-SCORES WITH ADDED USER KNOWLEDGE (UNPRUNED DATA SETS).

Data Set	1% UK	2% UK	3% UK	4% UK	$\geq$ 5% UK
$D1$	0.954	0.997	0.997	1	1
$D2$	–	0.97	1	1	1
$D3$	0.986	0.996	1	1	1
$D4$	–	0.987	0.991	0.998	1

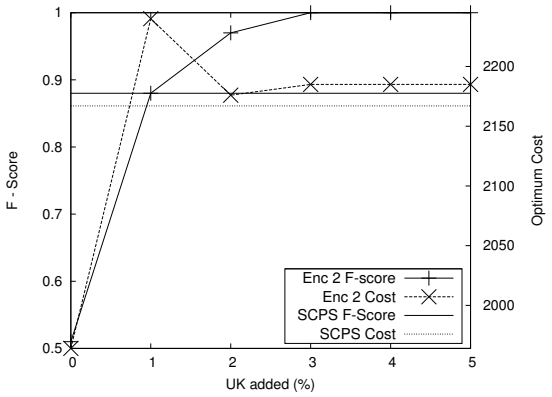

Fig. 5. Evolution of optimum cost and F-score with user knowledge (D2).

lower than those obtained using SCPS. The pruning did have a somewhat (but not drastic) negative effect on the F-score values obtained with Encoding 2. Figure 5 shows the evolution of both the cost and F-score of the clusterings obtained by Encoding 2 on the pruned data sets 2 and 4. (The results are similar for the other data sets.) Interestingly, the plots illustrate the difference between optimizing the cost function and measuring similarity between a clustering and a ground truth. We clearly see that both the cost of the clustering and the F-score is altered by the inclusion of user knowledge. The F-score improves as we steer the clustering toward the ground truth. The cost also converges fast toward the cost of the ground-truth clustering.

We note that even when working with pruned data, we still measure the cost of the clustering wrt the full data, which is why the globally optimal cost wrt the pruned data set might be higher than the cost of the ground-truth clustering wrt the entire data set. This is the case on $D4$, where adding user knowledge lowers the cost of the clustering. In $D2$ we however see the opposite. The ground-truth clustering has higher cost than the clustering produced by encoding 2, so the cost of the clustering produced rises when more UK is added.

TABLE III
COST OF CLUSTERINGS ON THE PRUNED DATA SETS.

	SCPS Cost	Encoding 2 Cost	Cost of GT
$D1$	2452	**2042**	1988
$D2$	2167	**2153**	2185
$D3$	2596	**2146**	2573
$D4$	1881	**1643**	1275

VIII. Overlapping Correlation Clustering

Finally, we demonstrate the extensibility of the MaxSAT-based approach to variants of correlation clustering, focusing on *overlapping correlation clustering* [7] in which individual data points are allow to be assigned to more than one cluster. Encoding 2 can be easily adapted for overlapping correlation clustering. As a result, our MaxSAT-based approach can *precisely* reconstruct an existing ground-truth clustering, hence providing notable better solutions that a previously proposed greedy local search method [7].

We consider a variant of overlapping correlation clustering as defined in [7]. Again, we are given a set $V = \{v_1, \ldots, v_N\}$ of data points, a binary similarity function $s\colon E \to \{0,1\}$ defined over a subset $E \subseteq V \times V$, and a set $L = \{1, \ldots, K\}$ of clusters. The objective is to find a clustering that assigns each point into a set of labels, i.e. a function $cl\colon V \to 2^L \setminus \{\emptyset\}$ minimizing the cost function $H(cl)$ defined as $\sum_{\substack{(v_i,v_j)\in E \\ cl(v_i)\cap cl(v_j)\neq\emptyset}} (1 - s(v_i, v_j)) + \sum_{\substack{(v_i,v_j)\in E \\ cl(v_i)\cap cl(v_j)=\emptyset}} s(v_i, v_j)$

It is simple to modify Encoding 2 in order to cover overlapping correlation clustering; the constraints enforcing that each data point must be assigned to *exactly one* cluster is relaxed into constraints which enforce that each data point must be assigned to *at least one* cluster. More concretely, for each data point v_i, we replace the constraint $\textsc{ExactlyOne}(v_i)$ in Encoding 2 (recall Figure 2) with the clause $(y_{i1} \vee y_{i2} \vee \cdots \vee y_{iK})$. This is the only modification needed.

In order to test the MaxSAT approach to overlapping correlation clustering using the modification of Encoding 2 just described, we consider the task of reconstructing an existing ground-truth clustering. In order to be able to directly compare our results with those reported in [7] for greedy local search methods for correlation clustering, we replicated the experimental setup of [7]. We used the well-known EMOTION and YEAST UCI data sets, containing 593 and 2417 data points, respectively. A ground-truth clustering is provided with these data sets, available at http://mulan.sourceforge.net/datasets.html. EMOTIONS has 6 clusters, YEAST 14. We obtained a binary similarity measure between the points by assigning a similarity value of 1 to each pair of point that share at least 1 cluster label. We applied Encoding 2 by setting the upper bound on the number of clusters according to the ground-truth clustering.

The solutions found with MaxSAT correspond *exactly* to the ground-truth clustering, i.e., solution was in both cases a perfect reconstruction of the original clustering, achieving a precision and recall value of 1 and a cost of 0. The EMOTIONS data set was solved optimally in 6 seconds and YEAST in just over 400 seconds. The running times reported in [7] for two methods proposed in that work for the same experiment are similar. However, neither one of their algorithms managed to perfectly reconstruct the ground-truth clustering on either of the data set, even when setting the upper bound according to the ground-truth clustering. This shows the benefits of our MaxSAT approach, guaranteeing optimal solutions.

IX. Conclusions and Future Work

We presented a MaxSAT-based approach to correlation clustering that extends to constrained and overlapping correlation clustering. To our best knowledge, this is the first approach which provides both cost-optimal clusterings and enables seamless integration of user knowledge for focusing the search on clusterings with specific properties of interest. Future work consists of developing more compact encodings, extending the approach to weighted graphs, and scaling the approach up to larger data sets. Finally, it would be interesting to apply the approach to different application domains, especially to domains in which user knowledge plays an important role.

Acknowledgements. This work is supported by Academy of Finland under grants 132812 (MJ) and 251170 (JB,MJ).

References

[1] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.
[2] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: Ranking and clustering," *J. ACM*, vol. 55, no. 5, 2008.
[3] M. Charikar, V. Guruswami, and A. Wirth, "Clustering with qualitative information," *J. Comput. Syst. Sci.*, vol. 71, no. 3, pp. 360–383, 2005.
[4] R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," *Discr. Appl. Math.*, vol. 144, no. 1-2, pp. 173–182, 2004.
[5] I. Giotis and V. Guruswami, "Correlation clustering with a fixed number of clusters," *Theory of Computing*, vol. 2, no. 1, pp. 249–266, 2006.
[6] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Comput. Biol.*, vol. 6, no. 3/4, pp. 281–297, 1999.
[7] F. Bonchi, A. Gionis, and A. Ukkonen, "Overlapping correlation clustering," in *Proc. ICDM*. IEEE, 2011, pp. 51–60.
[8] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen, "Chromatic correlation clustering," in *Proc. KDD*. ACM, 2012, pp. 1321–1329.
[9] M. J. Kearns, R. E. Schapire, and L. M. Sellie, "Toward efficient agnostic learning," in *Proc. COLT*. ACM, 1992, pp. 341–352.
[10] T. Nepusz, R. Sasidharan, and A. Paccanaro, "SCPS: a fast implementation of a spectral method for detecting protein families on a genome-wide scale," *BMC Bioinformatics*, vol. 11, p. 120, 2010.
[11] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proc. ICML*. Morgan Kaufmann, 2000, pp. 1103–1110.
[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained K-means clustering with background knowledge," in *Proc. ICML*. Morgan Kaufmann, 2001, pp. 577–584.
[13] I. Davidson and S. S. Ravi, "Intractability and clustering with constraints," in *Proc. ICML*. ACM, 2007, pp. 201–208.
[14] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *Proc. ICML*, 2002, pp. 307–314.
[15] I. Davidson and S. S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proc. SDM*, 2005.
[16] I. Davidson, S. S. Ravi, and L. Shamis, "A SAT-based framework for efficient constrained clustering," in *Proc. SDM*, 2010, pp. 94–105.
[17] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," in *Proc. FOCS*. IEEE, 2002, pp. 238–.
[18] C. Min Li and F. Manyà, "MaxSAT, hard and soft constraints," in *Handbook of Satisfiability*. IOS Press, 2009, ch. 19, pp. 613–631.
[19] S. Prestwich, "CNF encodings," in *Handbook of Satisfiability*. IOS Press, 2009, ch. 2, pp. 75–97.
[20] C. Sinz, "Towards an optimal CNF encoding of boolean cardinality constraints," in *Proc. CP*, ser. LNCS, vol. 3709, 2005, pp. 827–831.
[21] J. Davies and F. Bacchus, "Exploiting the power of MIPs solvers in Maxsat," in *Proc. SAT*, ser. LNCS, vol. 7962. Springer, 2013.
[22] A. Paccanaro, J. A. Casbon, and M. A. S. Saqi, "Spectral clustering of protein sequences," *Nucleic Acids Res.*, vol. 34, pp. 1571–1580, 2006.
[23] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990.
[24] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–540, 1995.