On the Use and Effect of Graph Decomposition in Qualitative Spatial and Temporal Reasoning*

Michael Sioutis
Université Lille-Nord de France, Artois
CRIL-CNRS UMR 8188
Lens, France
sioutis@cril.fr

Yakoub Salhi
Université Lille-Nord de France, Artois
CRIL-CNRS UMR 8188
Lens, France
salhi@cril.fr

Jean-François Condotta
Université Lille-Nord de France, Artois
CRIL-CNRS UMR 8188
Lens, France
condotta@cril.fr

ABSTRACT

We survey the use and effect of decomposition-based techniques in qualitative constraint-based reasoning, and clarify the notions of a tree decomposition, a chordal graph, and a partitioning graph, and their implication with a particular constraint property that has been extensively used in literature, namely, patchwork. As a consequence, we prove that a recently proposed decomposition-based approach that was presented in [*AAAI, 2014*] for checking the satisfiability of qualitative spatial constraint networks lacks soundness. Therefore, the approach becomes quite controversial as it does not seem to offer any technical advance at all, while experimental evaluation of it in a following paper presented in [*ICTAI, 2014*] becomes questionable.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic**]: Logic and constraint programming

General Terms

Theory

Keywords

Qualitative reasoning, tree decomposition, chordal graph, partitioning graph, patchwork, constraint network

1. INTRODUCTION

Qualitative Spatial and Temporal Reasoning (QSTR) is a major field of study in Artificial Intelligence, and, in particular, in Knowledge Representation. This field studies representations of space and time that abstract from numeric quantities. The concise expressiveness of the qualitative approach provides a promising framework that boosts research and applications in a plethora of areas and domains such as ambient intelligence, dynamic GIS, cognitive robotics, and spatiotemporal design [3].

The Interval Algebra (IA) [1] and a subset of the Region Connection Calculus (RCC) [19], namely RCC-8, are the dominant Artificial Intelligence approaches for representing and reasoning about qualitative temporal and topological relations respectively. These qualitative calculi use constraints to encode knowledge about the spatial or temporal relationships between entities. Thus, qualitative information can be modelled as a infinite-domain variant of a Constraint Satisfaction Problem (CSP) [15], for which we use the term Qualitative Constraint Network (QCN). For instance, there are infinitely many time points or temporal intervals on the time line and infinitely many regions in a two or three dimensional space. One way of dealing with infinite domains is using constraints over a finite set of binary relations, by employing a relation algebra [12].

Given a QCN, we are particularly interested in its satisfiability problem, that is, deciding whether there exists an interpretation of all variables of the QCN such that all constraints are satisfied by this interpretation. The satisfiability problem in IA and RCC-8 is $\mathcal{NP}$-hard in general [17, 21]. However, there exist large maximal tractable subclasses for IA and RCC-8 which can be used to make reasoning much more efficient even in the general $\mathcal{NP}$-hard case [16, 22]. In recent years, many works surfaced that use graph decomposition to significantly improve the efficiency and scalability of practical reasoning [2,6,7,11,13,18,24,25,28,29]. All these works, make use of a particular contraint property, namely, patchwork [10, 14]. Intuitively, patchwork ensures that the combination of two satisfiable QCNs that completely agree on the constraints between their common variables continues to be satisfiable.

The contribution of this paper comprises two interdependent parts: (i) we show that the approach proposed in [18] violates patchwork in two ways, namely, both in the complete agreement between two satisfiable QCNs and in the graph decomposition that is obtained, and therefore, lacks soundness, and (ii), to do so, we recall the notions of a tree decomposition, a chordal graph, and a partitioning graph that have been used in literature, and clarify the relationship between one another, but also their implication with patchwork. As such, our paper can be viewed both as a response paper to [18], and partially to [23], but also as a survey on the use and effect of graph decomposition in qualitative spatial and temporal reasoning.

The paper is organised as follows. In Section 2 we recall

*This work was funded by a PhD grant from Université d'Artois and region Nord-Pas-de-Calais.

SAC'15 April 13-17, 2015, Salamanca, Spain.

http://dx.doi.org/10.1145/2695664.2695831

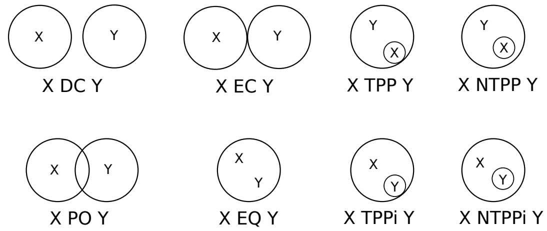

Figure 1: The RCC-8 constraint language

the definition of a QCN, along with the property of patchwork. Section 3 introduces the notions of a tree decomposition and a chordal graph, and the way they are interrelated and used in literature. In Section 4 we present the definition of a partitioning graph [18] and prove that it yields non-soundness when used with patchwork. In Section 5 we make a discussion and conclude.

2. PRELIMINARIES

A (binary) qualitative temporal or spatial constraint language is based on a finite set B of *jointly exhaustive and pairwise disjoint* (JEPD) relations defined on a domain D [12], called the set of base relations. The base relations of set B of a particular qualitative constraint language can be used to represent the definite knowledge between any two entities with respect to the given level of granularity. B contains the identity relation Id, and is closed under the converse operation ($^{-1}$). Indefinite knowledge can be specified by unions of possible base relations, and is represented by the set containing them. Hence, 2^{B} represents the total set of relations. 2^{B} is equipped with the usual set-theoretic operations (union and intersection), the converse operation, and the weak composition operation denoted by $\diamond$ [20].

The set of base relations of IA [1] is the set $\{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$. These thirteen relations represent the possible relations between *time intervals*. The set of base relations of RCC-8 [19] is the set $\{dc, ec, po, tpp, ntpp, tppi, ntppi, eq\}$. These eight relations represent the binary topological relations between *regions* that are non-empty regular subsets of some topological space, as depicted in Figure 1 (for the $2D$ case). IA and RCC-8 networks are qualitative constraint networks (QCNs), with relation eq being the identity relation in both cases.

Definition 1. *A* QCN *comprises a tuple (V, C) where V is a non empty finite set of variables and C is a mapping that associates a relation $C(v, v') \in 2^{\mathsf{B}}$ to each pair (v, v') of $V \times V$. C is such that $C(v, v) = \{\mathsf{Id}\}$ and $C(v, v') = (C(v', v))^{-1}$ for every $v, v' \in V$.*

An *atomic* QCN is a QCN where each constraint is defined by a base relation. Note that we always regard a QCN as a complete network. The constraint graph of a QCN $\mathcal{N} = (V, C)$ is a graph $G = (V, E)$, for which we have that $(v, v') \in E$ iff $C(v, v') \neq \mathsf{B}$. B corresponds to the universal relation, i.e., the non-restrictive[1] relation that contains all base relations, thus, it does not really pose a constraint. Given two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V', C')$, $\mathcal{N} \cup \mathcal{N}'$ denotes the QCN $\mathcal{N}'' = (V'', C'')$ where $V'' = V \cup V'$, $C''(u, v) = C''(v, u) = \mathsf{B}$ for all $(u, v) \in (V \setminus V') \times (V' \setminus V)$, $C''(u, v) = C(u, v) \cap C'(u, v)$ for every $u, v \in V \cap V'$, $C''(u, v) = C(u, v)$ for every $(u, v) \in (V \times V) \setminus (V' \times V')$, and $C''(u, v) = C'(u, v)$ for every $(u, v) \in (V' \times V') \setminus (V \times V)$. $C(v_i, v_j)$ will also be denoted by C_{ij}.

Checking the satisfiability of QCN of IA or RCC-8 is $\mathcal{NP}$-complete in general [17, 21]. However, there exist the large maximal tractable subclasses $\mathcal{H}_{\mathsf{IA}}$ for IA [16] and $\hat{\mathcal{H}}_8, \mathcal{C}_8$, and $\mathcal{Q}_8$ for RCC-8 [22] for which the satisfiability problem is tractable. Satisfiability checking of a QCN of IA or RCC-8 comprising only relations from a maximal tractable subclass of relations can be done in $O(|V|^3)$ time by a *path-consistency*[2] algorithm for example that iteratively performs the following operation to all triples of variables until a fixed point $\overline{C}$ is reached: $\forall i, j, k \in V$, $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \diamond C_{kj})$ [16, 22].[3]

We now recall the definition of the *patchwork* property that was originally introduced in [14] and was shown to hold for atomic QCNs of IA and RCC-8.

Definition 2 ([14]). *A constraint language has* patchwork*, if for any finite satisfiable constraint networks $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V', C')$ defined in this language such that $\forall u, v \in V \cap V'$ we have that $C(u, v) = C'(u, v)$, the constraint network $\mathcal{N} \cup \mathcal{N}'$ is satisfiable.*

Huang generalized the use of patchwork for non-atomic QCNs [10], providing us with the following proposition:

Proposition 1 ([10]). *Path-consistent* QCN*s of* IA*, or* RCC-8*, comprising relations from one of the maximal tractable subclasses $\mathcal{H}_{\mathsf{IA}}$, or $\hat{\mathcal{H}}_8, \mathcal{C}_8$, and $\mathcal{Q}_8$ resp., have patchwork.*

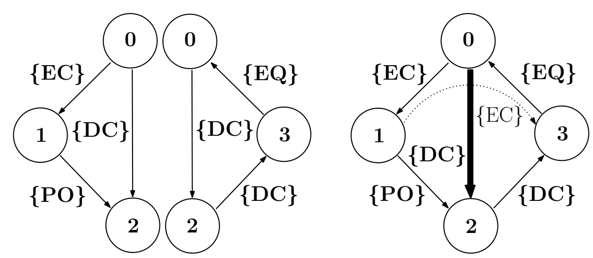

Figure 2: Patching two QCNs

Intuitively, patchwork ensures that the combination of two satisfiable constraint networks that agree on their common part, i.e., the constraints between their common variables, continues to be satisfiable. As an example, we can view the two QCNs of RCC-8 in Figure 2. The QCNs are atomic and are also path-consistent, therefore, by application of the patchwork property their union is satisfiable since they agree on the constraints between their common variables, namely, on C_{02}. (Note that it is not necessary to calculate relation C_{13} unless required by the specifics of a use case.)

[1] The result of the weak composition of any relation with the universal relation is the universal relation.

[2] The literature suggests the term *algebraic closure* [20] instead, which is equivalent to a path-consistency algorithm where the weak composition operator $\diamond$ is used [20], so we will use this more traditional term throughout the paper.

[3] Some of the cited works are based on encodings of QCNs into Boolean formulas. However, the formulas are constructed in such a way that each solution of the formula corresponds to a path-consistent QCN with relations from some maximal tractable subclass of relations, and vice versa.

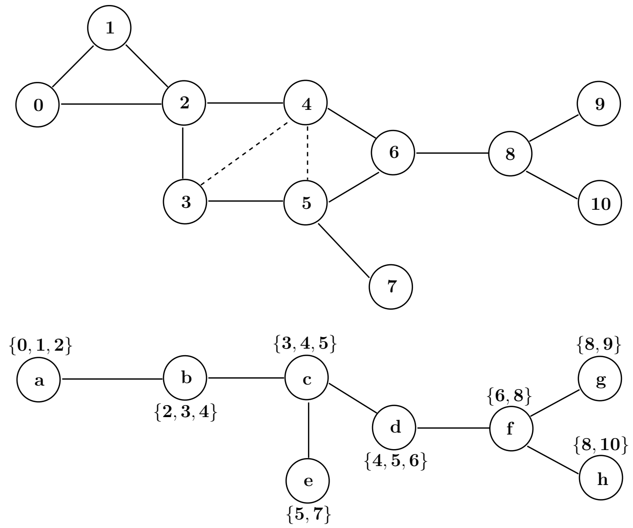

Figure 3: A graph (upper part) and its tree decomposition (lower part)

3. TREE DECOMPOSITION AND CHORDAL GRAPH

In this section we recall the notions of a tree decomposition and a chordal graph and review their use and effect in qualitative spatial and temporal reasoning in combination with patchwork[4].

A tree decomposition is formally defined as follows:

Definition 3 ([8]). *A* tree decomposition *of a graph $G = (V, E)$ is a tuple (T, X) where $T = (I, F)$ is a tree and $X = \{X_i \mid i \in I\}$ a collection of clusters (subsets of V) that satisfy the following properties:*

- *For every $v \in V$ there is at least one node $i \in I$ such that $v \in X_i$.*
- *For every $(u, v) \in E$ there exists a node $i \in I$ such that both $u, v \in X_i$.*
- *Let i_1, i_2, i_3 be three nodes in I such that i_2 lies on the path between i_1 and i_3 in T. Then, if $v \in V$ belongs to both X_{i_1} and X_{i_3}, v must also belong to X_{i_2}.*

Let us view the example presented in Figure 3. In the upper part of the figure we can view a graph $G = (V, E)$, which can be the constraint graph of a QCN. For the moment, we consider only the solid edges to be part of G and we disregard the dashed edges $(3, 4)$ and $(4, 5)$. A tree decomposition of G comprises a tree $T = (I, F)$ and a cluster X_i for every node $i \in I$ of that tree as shown in the lower part of the figure, e.g., $X_a = \{0, 1, 2\}$.

Tree decompositions have been explicitly introduced in qualitative reasoning by Condotta et al. in [7], and implicitly by Li et al. in [13] and Huang et al. in [11]. (What is presented in [11] properly contains the work in [13], thus, we will stick to the former work in what follows.)

In [7] the authors apply path-consistency on the clusters of a tree decomposition of the constraint graph of a QCN. The graphs induced by the clusters of the tree decomposition are completed with the introduction of a new set of edges, called fill edges, that correspond to the universal relation for a QCN. These fill edges for the example graph of Figure 3 are edges $(3, 4)$ and $(4, 5)$. As such, the clusters of the tree decomposition are considered to be *cliques*, namely, sets of vertices such that every two vertices in a set are connected by an edge. This is done for two reasons: (i) by definition path-consistency considers a complete graph to decide satisfiability of the corresponding constraint network, and (ii) the common vertices between any two complete graphs induce a complete graph, thus, the corresponding constraint networks will completely agree on the constraints between their common variables and the patchwork property can be used. Patchwork is then applied to patch together the path-consistent atomic QCNs that correspond to the graphs induced by the clusters of the tree decomposition in a tree-like manner and construct a satisfiable network.

In [11] the authors enlist a structure known as *dtree* (decomposition tree), which, as the name suggests, is very close to a tree decomposition. Without going further into detail, a dtree is a full binary tree where the root represents a given graph and for every non-leaf node, its two children represent a partitioning of the parent graph into two subgraphs. Thus, although a dtree is not a tree decomposition, it provides a way to construct a tree decomposition out of a given graph. A dtree and a tree decomposition are therefore equivalent in the context of qualitative reasoning, since omitting path-consistency checks across children of dtree nodes (as described in [11]) corresponds to omitting those checks across clusters of the tree decomposition into which the dtree is converted, as has been specifically pointed out in [7]. Similarly to [7], children of dtree nodes are treated as cliques, and patchwork is considered to patch together the path-consistent atomic QCNs of either IA or RCC-8 in a tree-like recursive manner and construct a satisfiable network.

The obersvant reader will note that it would be convenient to operate directly on a tree decomposition (T, X) of a given graph G, where X would be a collection of cliques. In this context *chordal* graphs become relevant. Formally, a chordal graph is defined as follows:

Definition 4 ([8]). *Let $G = (V, E)$ be an undirected graph. G is* chordal *(or triangulated) if every cycle of length greater than* 3 *has a chord, which is an edge connecting two non-adjacent nodes of the cycle.*

We then have the following proposition:

Proposition 2 ([8]). *Graph $G = (V, E)$ is chordal if and only if it has a tree decomposition $(T, \{X_1, \ldots, X_n\})$ where cluster X_i is a clique of G for every $i \in \{1, \ldots, n\}$.*

For example, the graph presented in Figure 3, with the dashed edges included, is chordal. Chordality checking can be done in (linear) $O(|V| + |E|)$ time for a given graph $G = (V, E)$ with the maximum cardinality search algorithm which also constructs an *elimination ordering* ω as a byproduct [27]. If a graph is not chordal, it can be made so by the addition of fill edges. This process is usually called *triangulation* of a given graph $G = (V, E)$ and can run as fast as in $O(|V| + (|E \bigcup F(\omega)|))$ time, where $F(\omega)$ is the set of fill edges that result by following the elimination ordering ω, eliminating the nodes one by one, and connecting all nodes in the neighborhood of each eliminated node, thus, making it simplicial in the elimination graph. If the graph is already chordal, following the elimination ordering ω means that no

[4]Some of the cited works use a property called *amalgamation*, which is equivalent to patchwork for atomic networks.

fill edges are added, i.e., ω is actually a *perfect elimination ordering* [8]. For example, a perfect elimination ordering for the chordal graph shown in Figure 3 would be the ordering $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 8 \rightarrow 10$ of its set of nodes. In general, it is desirable to achieve chordality with as few fill edges as possible. However, obtaining an optimum graph triangulation with the minimum number of fill edges is known to be $\mathcal{NP}$-hard [30]. As noted earlier, fill edges correspond to the universal relation for a QCN. As such, the *chordal constraint graph* of a given QCN is exactly its constraint graph augmented with constraints corresponding to the universal relation to make it chordal.

In the light of Propositions 1 and 2, research efforts focused on making the constraint graph of a given QCN chordal and applying path-consistency on that chordal graph, while fully utilizing maximal tractable subclasses of relations and not just the class of base relations that was typically used to only describe atomic networks. Towards this direction, we have the works of Chmeiss et al. for IA [6] and Sioutis et al. for RCC-8 [25]. These works were later combined in [2] to give the following result which is the strongest yet concerning path-consistency, patchwork, and maximal tractable subclasses of relations:

PROPOSITION 3 ([2]). *For a given* QCN $\mathcal{N} = (V, C)$ *of* RCC-*8, or* IA*, with relations from one of the maximal tractable subclasses* $\hat{\mathcal{H}}_8, \mathcal{C}_8$*, and* $\mathcal{Q}_8$*, or* $\mathcal{H}_{\mathsf{IA}}$ *resp., and for* $G = (V, E)$ *its chordal constraint graph, if* $\forall (i, j), (i, k), (j, k) \in E$ *we have that* $C_{ij} \subseteq C_{ik} \diamond C_{kj}$*, then* $\mathcal{N}$ *is satisfiable.*

Proposition 3 generalizes the results of all works that were discussed earlier in this section and make use of path-consistency as the main tool to check satisfiability, and has a great effect in the efficiency and scalability of practical reasoning. In particular, regarding native search, an algorithm based on the work of [4] was devised, called *partial path-consistency* [6], that performs path-consistency on the chordal graph $G = (V, E)$ of a given QCN $\mathcal{N}$ in $O(\delta|E|)$ time, where δ is the maximum vertex degree of G. Partial path-consistency is able to decide the satisfiability of $\mathcal{N}$ when it comprises relations from some maximal tractable subclass of relations. The search space for non-tractable QCNs was also reduced to $O(\alpha^{|E|})$ from $O(\alpha^{|V|^2})$ for a backtracking algorithm [22], where α is the branching factor provided by some maximal tractable subclass of relations (e.g., $\alpha = 1.4375$ for subclass $\hat{\mathcal{H}}_8$ for RCC-8 [22]). Regarding approaches based on encodings of QCNs into Boolean formulas, i.e., SAT-based approaches, the implication of Proposition 3 led to significant memory and speed improvements for both IA [29] and RCC-8 [28] targeted implementations.

Before closing this section with another general and strong result that concerns tree decompositions and patchwork, let us introduce the *treewidth* of a graph. The *width* of a tree decomposition $(T, \{X_1, \ldots, X_n\})$ is $\max_{1 \leq i \leq n} |X_i| - 1$. The *treewidth* of a graph G is the minimum width possible for arbitrary tree decompositions of G.

THEOREM 1 ([5, 11]). *For any k, the satisfiability problem for* QCN*s of* IA *and* RCC-*8 of treewidth at most k can be solved in polynomial time.*

An algorithm for Theorem 1 is provided both in [11] and in [5]. (The algorithm in [5] is particular to RCC-8, but it can be generalized to IA based on common properties.)

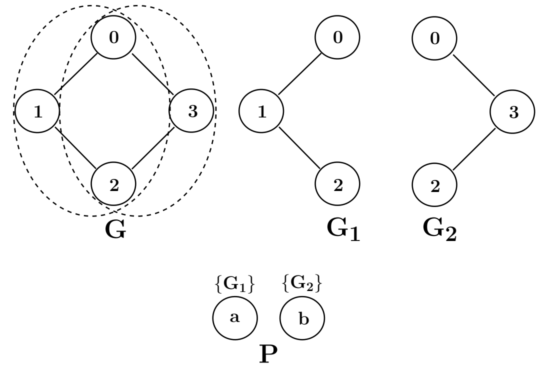

Figure 4: A graph and its partitioning graph with the parts comprising it (also contained in dashed circles in the initial graph)

4. PARTITIONING GRAPH

In this section we prove that the decomposition-based approach presented in [18] for checking the satisfiability of QCNs of RCC-8 lacks soundness, as the notion of a *partitioning graph* defined in that work is not coherent with the use of patchwork upon which it solely relies, in two ways which we enumerate and analyse in the form of issues.

Let $G = (V, E)$ be an undirected graph and k a positive integer. If $U \subseteq V$, then $G(U)$ will denote the subgraph of G that is induced by the set of vertices U. A set $\{V_i \subseteq V \mid 1 \leq i \leq k\}$ with k pairwise-disjoint elements such that $\bigcup_{i=1}^{k} V_i = V$, is called a k-way partitioning of G. Finally, let $\emptyset$ denote the empty, edgeless, graph. We now recall the definition of a partitioning graph from [18] as follows:

DEFINITION 5 ([18]). *Let $G = (V, E)$ be a graph and $\{V_1, \ldots, V_k\}$ a k-way partitioning of G for some positive integer k. A* partitioning graph *P of G is an undirected graph $(V_P, E_P, \lambda_P, G_P)$, where $V_P = \{v_1, \ldots, v_k\}$ is the set of its nodes, E_p the set of its edges, $\lambda_P : V_P \rightarrow 2^V$ a function that maps each node of P to a partition (subset of V) of G, and G_P a set of k subgraphs (parts) of G . The following conditions must be satisfied:*

- *If $G_i \in G_P$ then the set of vertices of G_i is a superset U of $\lambda_P(v_i)$ and the set of its edges is $E(G(U))$.*
- *Any edge in G should be present in at least one subgraph $G_i \in G_P$.*
- *Edge $(v_i, v_j) \in E_P$ if and only if $G_i \cap G_j \neq \emptyset$ (i.e., if and only if G_i and G_j share a common edge).*

We now enumerate the issues that lead to non-soundness and provide counter-examples for each case.

Issue 1.

The first issue has to do with the fact that a complete agreement on the constraints between the common variables of two networks is not achieved in order to allow the applicability of patchwork. Let us consider the example of Figure 4. Graph G is partitioned into two parts, namely, G_1 and G_2. The partitioning graph is shown in the lower part of the figure, and it comprises the set of nodes $\{a, b\}$ and an empty set of edges. Node a corresponds to subgraph G_1 and node b to subgraph G_2. Its set of edges E_P is empty as subgraphs G_1 and G_2 do not share a common edge, thus, the only possible edge (a, b) does not exist. In [18] the au-

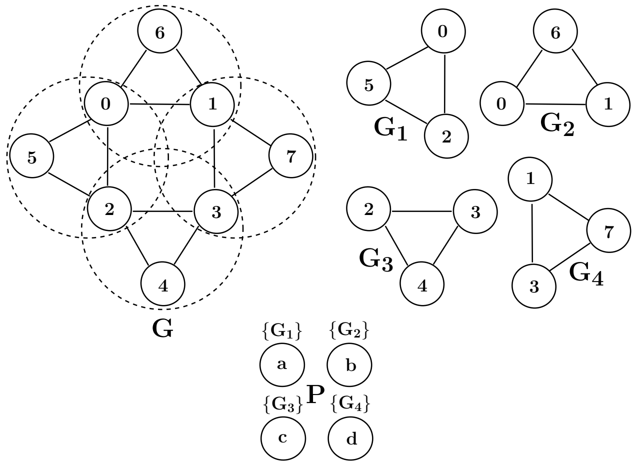

Figure 5: A graph and its partitioning graph with the parts comprising it (also contained in dashed circles in the initial graph)

thors perform path-consistency on the subgraphs of a graph separately, in a parallel fashion, and then rely on the set of edges E_P to identify the subgraphs among which a complete agreement has to be ensured (the reader is kindly asked to refer to line 7 in the function of Algorithm 2 in [18]). If, as in this example, such an edge does not exist, a complete agreement is never achieved. This can be the cause of failing to identify inconsistencies. Let us assume that graph G, as depicted in Figure 4, is the constraint graph of a given QCN comprising constraints $C_{01} = C_{12} = C_{23} = C_{30} = \{TPP\}$. This yields an inconsistent network, as it basically infers that region 0 is properly contained in region 2, and vice versa. Applying path-consistency on that network would result in the empty relation assignment for constraint C_{02} (inconsistency). However, that constraint is never checked in our example. Although the authors implicitly complete subgraphs G_1 and G_2 in order to apply path-consistency, they do not complete these subgraphs when computing their intersection as specified in the last bullet of Definition 5. Even if they did implicitly consider complete subgraphs for that part of the definition, and edge (a, b) indeed existed, line 7 in the function of Algorithm 2 in [18] still requires that an agreement should be achieved for every common edge of G_1 and G_2 (the initial non-complete subgraphs), which is none. If they implicitly considered complete subgraphs for that part of the algorithm too, then this particular issue for a 2-way partitioning would be resolved. We have also verified this issue experimentally with the implementation used in [18].

Before proceeding to the next issue, let us assume that the first issue is fixed with everything that we propose, and a 2-way partitioning is actually valid for applying patchwork. We mean to show, that the concept of a partitioning graph is beyond repair, unless it is structured in a way that it defines a tree decomposition, which beats the purpose of having to define a partitioning graph in the first place.

Issue **2.**

This issue has to do with the fact that even if the first issue is resolved, the partitioning graph can suffer from the existence of cycles that are created by subgraphs of a given graph. Let us consider the example of Figure 5. Graph G is partitioned into four parts, namely, G_1, G_2, G_3, and G_4. The partitioning graph is shown in the lower part of the figure, and the correspondance of its sets of nodes and edges with the different subgraphs should be clear up to this point. Note that all subgraphs are complete, thus, they completely overlap with each other on the common vertices. For example, graph G_1 completely overlaps with graph G_2 on the edges of the graph defined on the single common vertex 0, as their intersection yields the complete graph on singe vertex 0. Although such an overlap is trivial, as a complete graph on a single vertex (singleton graph) does not have any edges, it is sufficient to ensure the applicability of patchwork. (Our example can be easily extended to non-trivial overlaps.) However, due to the last bullet of Definition 5, the partitioning graph is unable to obtain any edges. In fact, even if such edges existed in E_P, in any possible combination or amount, the partitioning graph would still fail to capture/break the cycle that is constructed by the complete subgraphs G_1, G_2, G_3, and G_4, namely, the cycle defined by vertices 0, 1, 2, and 3. This cycle, as shown in the example of Figure 4, can harbor an inconsistency. Such a cycle exists also between vertices 3, 4, 5, and 7 in [18, Fig. 1]. Patchwork alone is only valid for tree decompositions, as they guarantee acyclicity of the graphs induced by their clusters and, thus, do not harbor cycles with inconsistencies that cannot be detected by applying path-consistency on the clusters.

From the aforementioned issues we infer the following fact:

Proposition 4. *The approach presented in [18] for checking the satisfiability of a* QCN *of* RCC-*8 lacks soundness.*

Essentially, the approach defines a partial algorithm; a given satisfiable QCN will be shown to be satisfiable, as the approach in [18] due to disregarding constraints operates on a less restrictive constraint graph of the input network where constraint propagation and consistency checks are limited, whilst an unsatisfiable QCN may be shown to be satisfiable.

4.1 Impact on Performance

The main contribution of [18] lies in the performance of its offered implementation, as it promises efficiency that goes well beyond the state-of-the-art. Computing a k-way partitioning alone is among the graph partitioning problems that fall under the category of $\mathcal{NP}$-hard problems [9], and solutions to these problems are generally derived using heuristics and approximation algorithms, such as the ones offered by the METIS[5] software employed in [18]. We leave aside any extra computational complexity that would result from needing to restrict a partitioning graph to being a tree decomposition (e.g., by identifying cycles or using some recursion as in [11]), and focus on native search. As explained in Section 3, native search in qualitative spatial and temporal reasoning is bound to the number of constraints in a given QCN, and not to its number of variables as in "traditional" constraint programming. This is because, in a sense, the constraints in a given QCN are the *true* variables for which we have to assign some relation. Indeed, the search space defined in [18] relies mainly on the number of constraints in a given QCN. As a result, the implementation in that paper benefited from a reduced search space with respect to the one that should be considered, as we showed earlier that some constraints can be disregarded. However, even in that case, a re-evaluation of the implementation used in [18] against

[5] `http://glaros.dtc.umn.edu/gkhome/views/metis`

state-of-the-art solvers showed that it performs very poorly with respect to the state-of-the-art [23]. Work in [23] does not deal with any of the issues that we dealt with in this paper as it assumes a partitioning graph to implicitly define a tree decomposition, thus, [23] presents mostly lower bounds on the performance of the implementation used in [18].

5. DISCUSSION

To conclude, we showed that the decomposition-based approach presented in [18] for checking the satisfiability of QCNs of RCC-8 lacks soundness, as the notion of a *partitioning graph* defined in that work is not coherent with the use of patchwork upon which it solely relies. Further, we showed how that notion is beyond repair, unless it is reformulated to define a tree decomposition, implicitly or explicitly, and discussed the impact of these observations on the performance of the offered implementation in [18], which was already found to be poor in [23].

We think that future efforts regarding decomposition-based approaches utilizing parallelism, such as the approach attempted in [18], should rely on chordal graphs (tree decompositions into cliques), which can both be constructed and also yield a natural tree decomposition of their cliques in linear time [8]. The cliques can then be collected at no extra cost and parallelism *might* be efficiently utilized. It is an issue that looks promising and calls for further research. Recent work suggests that for the type of networks considered in [18], even parallelism itself can be utilized cost-free with a simple, yet powerful, approach as presented in [26].

6. REFERENCES

[1] Allen, J. F. An Interval-Based Representation of Temporal Knowledge. In *IJCAI* (1981).

[2] Amaneddine, N., Condotta, J., and Sioutis, M. Efficient Approach to Solve the Minimal Labeling Problem of Temporal and Spatial Qualitative Constraints. In *IJCAI* (2013).

[3] Bhatt, M., Guesgen, H., Wölfl, S., and Hazarika, S. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. *Spatial Cognition & Computation 11* (2011), 1–14.

[4] Bliek, C., and Sam-Haroud, D. Path consistency on triangulated constraint graphs. In *IJCAI* (1999).

[5] Bodirsky, M., and Wölfl, S. RCC8 is polynomial on networks of bounded treewidth. In *IJCAI* (2011).

[6] Chmeiss, A., and Condotta, J.-F. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI* (2011).

[7] Condotta, J., and D'Almeida, D. Consistency of Qualitative Constraint Networks from Tree Decompositions. In *TIME* (2011).

[8] Diestel, R. *Graph Theory, 4th Edition*, vol. 173. Springer, 2012.

[9] Garey, M. R., Johnson, D. S., and Stockmeyer, L. J. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci. 1* (1976), 237–267.

[10] Huang, J. Compactness and its implications for qualitative spatial and temporal reasoning. In *KR* (2012).

[11] Huang, J., Li, J. J., and Renz, J. Decomposition and tractability in qualitative spatial and temporal reasoning. *AI 195* (2013), 140–164.

[12] Ladkin, P. B., and Maddux, R. D. On binary constraint problems. *JACM 41* (1994), 435–469.

[13] Li, J. J., Huang, J., and Renz, J. A divide-and-conquer approach for solving interval algebra networks. In *IJCAI* (2009).

[14] Lutz, C., and Milicic, M. A Tableau Algorithm for DLs with Concrete Domains and GCIs. *JAR 38* (2007), 227–259.

[15] Montanari, U. Networks of constraints: Fundamental properties and applications to picture processing. *Inf. Sci. 7* (1974), 95–132.

[16] Nebel, B. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. *Constraints 1* (1997), 175–190.

[17] Nebel, B., and Bürckert, H. Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *JACM 42* (1995), 43–66.

[18] Nikolaou, C., and Koubarakis, M. Fast Consistency Checking of Very Large Real-World RCC-8 Constraint Networks Using Graph Partitioning. In *AAAI* (2014).

[19] Randell, D. A., Cui, Z., and Cohn, A. A Spatial Logic Based on Regions and Connection. In *KR* (1992).

[20] Renz, J., and Ligozat, G. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP* (2005).

[21] Renz, J., and Nebel, B. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *AI 108* (1999), 69–123.

[22] Renz, J., and Nebel, B. Efficient Methods for Qualitative Spatial Reasoning. *JAIR 15* (2001), 289–318.

[23] Sioutis, M. Triangulation versus Graph Partitioning for Tackling Large Real World Qualitative Spatial Networks. In *ICTAI* (2014).

[24] Sioutis, M., and Condotta, J.-F. Tackling Large Qualitative Spatial Networks of Scale-Free-Like Structure. In *SETN* (2014).

[25] Sioutis, M., and Koubarakis, M. Consistency of Chordal RCC-8 Networks. In *ICTAI* (2012).

[26] Sioutis, M., Salhi, Y., and Condotta, J. A Simple Decomposition Scheme For Large Real World Qualitative Constraint Networks. In *FLAIRS* (2015).

[27] Tarjan, R. E., and Yannakakis, M. Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs. *SIAM J. Comput. 13* (1984), 566–579.

[28] Westphal, M., and Hué, J. A Concise Horn Theory for RCC8. In *ECAI* (2014).

[29] Westphal, M., Hué, J., and Wölfl, S. On the Propagation Strength of SAT Encodings for Qualitative Temporal Reasoning. In *ICTAI* (2013).

[30] Yannakakis, M. Computing the Minimum Fill-In is NP-Complete. *SIAM J. on Algebraic Discrete Methods 2* (1981), 77–79.