# A Decomposition Framework for Inconsistency Handling in Qualitative Spatial and Temporal Reasoning

**Yakoub Salhi**[1] , **Michael Sioutis**[2]
[1]CRIL UMR 8188, Université d'Artois & CNRS, France
[2]LIRMM UMR 5506, Université de Montpellier & CNRS, France
salhi@cril.fr, msioutis@lirmm.fr

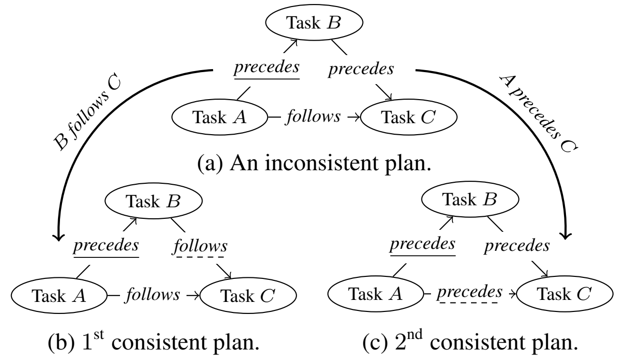
## Abstract

Decomposition can be a fundamental process for dealing with inconsistency in different domains. Among other things, it allows us to capture potential contexts, identify conflicting factors, restore consistency, and measure inconsistency. The aim of this paper is to explore the process of decomposition in qualitative spatial and temporal reasoning. We first study a problem that consists in decomposing the original inconsistent constraint network into the fewest possible consistent subnetworks (components) that share a given part. After establishing several interesting theoretical properties, such as providing bounds on the number of components in a decomposition, as well as computational complexity results, we propose two methods for solving this problem. The first method is based on a SAT encoding, while the second one corresponds to a greedy constraint-based algorithm, a variant of which involves the use of spanning trees to reduce the number of oracle calls. Secondly, we consider a version of the previous decomposition problem by focusing on maximizing the similarity between the decomposition components; the similarity in this context is represented by the common constraints among components. We then adapt our methods to solve this new problem. Thirdly, we propose two inconsistency measures that are based on our decomposition framework and show that they satisfy several desired properties. Finally, we provide implementations of our decomposition methods and perform an experimental evaluation.

## 1 Introduction

Dealing with inconsistency is a central problem in knowledge representation and reasoning. This stems from the fact that inconsistency can arise for many reasons in real-world applications, such as context dependency, multi-source information, vagueness, noisy data, etc. Among the approaches that are involved in inconsistency handling, we can mention argumentation, non-monotonic reasoning, paraconsistency, belief revision, and inconsistency measurement (e.g., see (Brewka, Dix, and Konolige 1997; Besnard and Hunter 2008; Hunter and Konieczny 2010; Tanaka et al. 2013; Thimm 2016)).

In this work, we are interested in dealing with inconsistency in Qualitative Spatio-Temporal Reasoning (QSTR) (Ligozat 2013). QSTR is an AI framework that aims to mimic, natural, human-like representation and reasoning regarding space and time. This framework is applied to a variety of domains, such as qualitative case-based reasoning and learning (Homem et al. 2020) and visual sense-making (Suchan, Bhatt, and Varadarajan 2021); see (Sioutis and Wolter 2021) for a recent survey.



(a) An inconsistent plan.

(b) 1st consistent plan.

(c) 2nd consistent plan.

Figure 1: A decomposition of an inconsistent qualitative constraint network (QCN) into consistent subnetworks (components).

**Motivation** We study the decomposition of an inconsistent constraint network into consistent subnetworks under, possible, mandatory constraints. To illustrate the interest of such a decomposition, we provide a simple example described in Figure 1. The QCN in the top corresponds to a description of an inconsistent plan. Further, assume that the constraint Task $A$ $\{precedes\}$ Task $B$ is mandatory. To handle inconsistency, this plan can be transformed into two consistent plans depicted in Figures 1b and 1c; this can be used, e.g., to capture the fact that Task $C$ must be performed twice. More generally, network decomposition can be involved in inconsistency handling in several ways. It can be used to identify potential contexts that explain the presence of inconsistent information. It can also be used to restore consistency through a compromise between the components of a decomposition (e.g., by using belief merging (Condotta et al. 2010)). In addition, QCN decomposition can be used as the basis for defining inconsistency measures.

**Contributions** The contributions of this work are manifold. First, we propose a theoretical study of a problem that consists in decomposing a possibly inconsistent QCN into a bounded number of consistent QCNs that may satisfy a specified part in the original QCN; intuitively, the required common part corresponds to the constraints that are considered necessary, if any. To this end, we provide upper bounds for the minimum number of components in a decomposition as well as computational complexity results. Secondly, we provide two methods for solving our decomposition problem. The first method corresponds to a greedy constraint-based algorithm, a variant of which involves the use of spanning trees; the basic idea of this variant is that any acyclic constraint graph in QSTR is consistent, and this can be used as a starting point for building consistent components. The second method corresponds to a SAT-based encoding: every model of this encoding is used to construct a valid decomposition. Thirdly, we consider a variant of the initial decomposition problem that focuses on maximizing the similarity between components; the similarity between two QCNs is quantified by the number of common non-universal constraints. The interest in maximizing the similarity lies mainly in the fact that it reduces the number of constraints that allow each component to be distinguished from the rest. We then adapt our previous methods to maximize the similarity. Additionally, we introduce two inconsistency measures based on QCN decomposition and show that they satisfy several desired properties in the literature. These measures can be seen as counterparts of measures for propositional knowledge bases introduced in (Thimm 2016; Ammoura et al. 2017). Finally, we provide implementations of our methods for computing decompositions and experimentally evaluate them using different metrics.

**Organization** The rest of the paper is organized as follows: In Section 2 we introduce some necessary definitions and notations for the reader; in Section 3 we introduce and study the problem of decomposing an inconsistent constraint network into consistent components, as well as certain optimization versions of this problem; in Section 4 we present diverse approaches (constraint- and SAT-based) for obtaining consistent decompositions; in Section 5 we introduce inconsistency measures in QSTR based on the proposed decomposition framework; in Section 6 we perform an experimental evaluation of our methods with constraint networks of well-known calculi in QSTR; finally, in Section 7 we conclude and provide some directions for future work.

## 2 Preliminary Definitions and Notations

In this section, we provide some necessary definitions and notations that are used in the sequel.

### 2.1 Qualitative Spatial and Temporal Reasoning

A binary qualitative spatial or temporal constraint language is based on a finite set $\mathsf{B}$ of *jointly exhaustive and pairwise disjoint* relations, called *base relations* (Ligozat 2013) and defined over an infinite domain $\mathsf{D}$ (e.g., $\mathbb{R}$). The base relations of a particular qualitative constraint language can be used to represent the definite knowledge between any two of its entities with respect to the level of granularity provided by the domain $\mathsf{D}$. The set $\mathsf{B}$ contains the identity relation $\mathsf{Id}$, and is closed under the *converse* operation ($^{-1}$). Indefinite knowledge can be specified by a union of possible base relations, and is represented by the set containing them. Hence, $2^{\mathsf{B}}$ represents the total set of relations. The set $2^{\mathsf{B}}$ is equipped with the usual set-theoretic operations of union and intersection, the converse operation, and the *weak composition* operation denoted by the symbol $\diamond$ (Ligozat 2013). For all $r \in 2^{\mathsf{B}}$, we have that $r^{-1} = \bigcup\{b^{-1} \mid b \in r\}$. The weak composition ($\diamond$) of two base relations $b, b' \in \mathsf{B}$ is defined as the smallest (i.e., most restrictive) relation $r \in 2^{\mathsf{B}}$ that includes $b \circ b'$, or, formally, $b \diamond b' = \{b'' \in \mathsf{B} \mid b'' \cap (b \circ b') \neq \emptyset\}$, where $b \circ b' = \{(x, y) \in \mathsf{D} \times \mathsf{D} \mid \exists z \in \mathsf{D}$ such that $(x, z) \in b \wedge (z, y) \in b'\}$ is the (true) composition of $b$ and $b'$. For all $r, r' \in 2^{\mathsf{B}}$, we have that $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r,\ b' \in r'\}$.

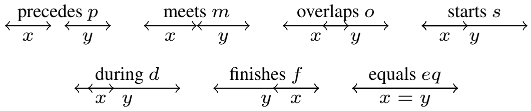


Figure 2: A representation of the 13 base relations $b$ of IA, each one relating two potential intervals $x$ and $y$ as in $x\ b\ y$; the converse of $b$, i.e., $b^{-1}$, can be denoted by $bi$ and is omitted in the figure.

As an illustration, consider the well-known qualitative temporal constraint language of Interval Algebra (IA), introduced by Allen (Allen 1983). IA considers time intervals on the real line, and the set of base relations $\mathsf{B} = \{eq\ (= \mathsf{Id})$, $p$, $pi$, $m$, $mi$, $o$, $oi$, $s$, $si$, $d$, $di$, $f$, $fi\}$ to encode knowledge about the temporal relations between such intervals, as described in Figure 2. As another example, the Region Connection Calculus (RCC8) (Randell, Cui, and Cohn 1992) considers spatial regions and the set of base relations $\mathsf{B} = \{DC$, $EC$, $EQ$, $PO$, $TPP$, $TPPi$, $NTPP$, $NTPPi\}$ to reason about topological relations between regions. Moreover, RCC5 is a fragment of RCC8 where boundaries of regions have no significance (Bennett 1994).

Finally, the challenge of representing and reasoning about qualitative spatio-temporal information can be facilitated by a *qualitative constraint network (QCN)*, defined as follows:

**Definition 1.** *A* qualitative constraint network (QCN) *is a tuple* $(V, C)$ *where:*

- $V = \{v_1, \ldots, v_n\}$ *is a finite set of variables over some infinite domain* $\mathsf{D}$ *(e.g., time points or 2D regions);*
- *and* $C$ *is a mapping* $C : V \times V \rightarrow 2^{\mathsf{B}}$ *associating a relation with each pair of variables s.t.,* $\forall v \in V$, $C(v, v) = \{\mathsf{Id}\}$, *and,* $\forall v, v' \in V$, $C(v, v') = (C(v', v))^{-1}$.

A simplified QCN is shown in Figure 1. For convenience, we often consider that the set of variables of a QCN consists of integers. Some more definitions follow.

A QCN $\mathcal{N} = (V, C)$ is *trivially inconsistent* iff $\exists v, v' \in V$ such that $C(v, v') = \emptyset$.

A *solution* of a QCN $\mathcal{N} = (V, C)$ is a mapping $f : V \rightarrow \mathsf{D}$ s.t. $\forall v, v' \in V$, $\exists b \in C(v, v')$ s.t. $(f(v), f(v')) \in b$; $\mathcal{N}$ is said to be *consistent* iff it admits a solution.

A QCN $\mathcal{N} = (V, C)$ is *atomic* iff $\forall v, v' \in V$, $C(v, v') = \{b\}$ with $b \in \mathsf{B}$.

A *scenario* of a QCN $\mathcal{N} = (V, C)$ is a consistent atomic QCN $\mathcal{S} = (V, C')$ s.t. $\forall u, v \in V$, $C'(u, v) \subseteq C(u, v)$.

Given a QSTR formalism $\mathcal{F}$, we use $\mathsf{SAT}(\mathcal{F})$ to refer to the problem of deciding whether a QCN is consistent.

The *constraint graph* of a QCN $\mathcal{N} = (V, C)$, denoted by $G_{\mathcal{N}}$, is an undirected graph $(V, E)$ where for all $i, j \in V$ with $i \neq j$, $\{i, j\} \in E$ iff $C(i, j) \neq \mathsf{B}$. We focus on qualitative formalisms where for every not trivially inconsistent QCN $\mathcal{N}$, if $G_{\mathcal{N}}$ is an acyclic graph then $\mathcal{N}$ is consistent (this is generally the case for most well-known QSTR calculi (Dylla et al. 2017)).

Given an undirected graph $G$, the set of its vertices is denoted by $V(G)$ and that of its edges by $E(G)$. In addition, we use $v(G)$ and $e(G)$ to denote the number of its vertices and that of its edges, respectively.

The following notational conventions are used throughout the paper for every QCN $\mathcal{N} = (V, C)$.

For two variables $v, v' \in V$, we use $\mathcal{N}[v, v']$ to denote the relation $C(v, v')$.

For $V' \subseteq V$, $\mathcal{N}\downarrow_{V'}$ denotes the QCN $\mathcal{N}$ restricted to $V'$.

For two variables $v, v' \in V$ and a relation $r \in 2^{\mathsf{B}}$, we use $v\ r\ v'$ to denote that $C(v, v') = r$ when there is no ambiguity about the considered QCN.

For two variables $v, v' \in V$ and a relation $r \in 2^{\mathsf{B}}$, we use $\mathcal{N}_{[v,v']/r}$ to denote the result of substituting $C(v, v')$ with $r$ in $\mathcal{N}$. Formally, $\mathcal{N}_{[v,v']/r}$ is the QCN $(V, C')$ defined by $C'(v, v') = r$, $C'(v', v) = r^{-1}$ and, $\forall (u, u') \in (V \times V) \setminus \{(v, v'), (v', v)\}$, $C'(u, u') = C(u, u')$.

Finally, we use $[\![\mathcal{N}]\!]$ to denote the set $\{(i, j) \in V \times V : i < j\}$. Furthermore, $[\![\mathcal{N}]\!]^s$ is used to denote the set $\{(i, j) \in [\![\mathcal{N}]\!] : \mathcal{N}[i, j] \neq \mathsf{B}\}$.

### 2.2 Propositional Logic and the SAT Problem

The language of propositional logic is inductively defined from a countable set of propositional variables, denoted by $\mathcal{P}$, and using the logical connectives $\wedge$, $\vee$, $\neg$ and $\rightarrow$. Notationally, we use the letters $p$, $q$ and $r$, possibly primed and/or with subscripts and/or with superscripts, to denote the propositional variables. In addition, given a formula $\phi$, we use $Var(\phi)$ to denote the set of variables occurring in $\phi$.

An *interpretation* of $\phi$ is a function $\omega : V \rightarrow \{0, 1\}$ with $Var(\phi) \subseteq V$. This function is inductively extended to the propositional formulas as usual. We say that an interpretation $\omega$ of $\phi$ is a *model* of this formula, written $\omega \models \phi$, if the identity $\omega(\phi) = 1$ holds. A formula is said to be *satisfiable* if it admits at least one model; otherwise, it is *unsatisfiable*. The *SAT problem* is a decision problem that consists in determining whether a propositional formula is satisfiable.

We also use in this paper one of the well-known optimization generalizations of the SAT problem, called Partial MaxSAT. In sum, the MaxSAT problem is the problem of finding an assignment that satisfies as many clauses of a given set of clauses as possible (Johnson 1974). Then, the Partial MaxSAT problem is an extension of the MaxSAT problem defined as follows: an instance $\Omega$ of Partial MaxSAT (Miyazaki, Iwama, and Kambayashi 1996; Cai et al. 2014) is a set of hard and soft clauses, and a solution $\omega$ of $\Omega$ is an assignment that satisfies the hard clauses and maximizes the number of satisfied soft clauses.

## 3 Consistent Decomposition

In this section, we formally introduce and study the problem of decomposing an inconsistent constraint network into consistent components, as well as certain optimization versions of this problem.

### 3.1 Problem Statement

Let us begin by defining the central object of our study.

**Definition 2.** *Let $\mathcal{N} = (V, C)$ be a QCN, $I$ a subset of $[\![\mathcal{N}]\!]$ and $\alpha$ a strictly positive integer. A* $(\alpha, I)$-decomposition *of $\mathcal{N}$ is a multiset $\{\mathcal{N}_1, \ldots, \mathcal{N}_\alpha\}$ of $\alpha$ QCNs over the same set of variables $V$ where:*

1. *for all $i, j \in V$ and for all $l \in \{1, \ldots, \alpha\}$, $\mathcal{N}_l[i, j] = \mathcal{N}[i, j]$ or $\mathcal{N}_l[i, j] = \mathsf{B}$;*
2. *for all $(i, j) \in I$ and for all $l \in \{1, \ldots, \alpha\}$, $\mathcal{N}_l[i, j] = \mathcal{N}[i, j]$;*
3. *for all $(i, j) \in [\![\mathcal{N}]\!]$, there exists $l \in \{1, \ldots, \alpha\}$ s.t. $\mathcal{N}_l[i, j] = \mathcal{N}[i, j]$;*
4. *for all $l \in \{1, \ldots, \alpha\}$, $\mathcal{N}_l$ is consistent.*

Property 1 states that each constraint in each component of the decomposition is either specified in the same way as the original QCN or universal. It allows us to avoid having a non-universal constraint that does not occur in the original QCN. Property 2 requires that the constraints corresponding to the elements of $I$ in the original QCN be satisfied by all components. Intuitively, the set $I$ is used to represent the constraints that are necessary; the other constraints can be seen as possible in a component. Property 3 says that each constraint in the original QCN must be satisfied in at least one component. Property 4 ensures that all components are consistent.

In the sequel, we use *$I$-decomposition* to refer to any $(\alpha, I)$-decomposition ($\alpha$ is arbitrary).

Given a QSTR formalism $\mathcal{F}$, we use $\mathsf{ConsDec}(\mathcal{F})$ to denote the decision problem defined as follows:

**Input:** A QCN $\mathcal{N}$, a subset $I \subseteq [\![\mathcal{N}]\!]$ and a strictly positive integer $\alpha$.

**Output:** Decide whether $\mathcal{N}$ admits a $(\alpha, I)$-decomposition.

### 3.2 Computational Complexity

Our first complexity result relates the intractability of the decomposition problem to the intractability of the consistency problem.

**Theorem 1.** *For every QSTR formalism $\mathcal{F}$, if $\mathsf{SAT}(\mathcal{F})$ is NP-hard, then $\mathsf{ConsDec}(\mathcal{F})$ is also NP-hard.*

*Proof.* It is a direct consequence of the fact that a QCN $\mathcal{N}$ is consistent iff $\mathcal{N}$ admits a $(1, \emptyset)$-decomposition. □

**Corollary 1.** *The problems $\mathsf{ConsDec}(\mathsf{RCC5})$, $\mathsf{ConsDec}(\mathsf{RCC8})$ and $\mathsf{ConsDec}(\mathsf{IA})$ are NP-complete.*

*Proof.* The proof of NP-hardness is a consequence of Theorem 1, and the fact that $\mathsf{SAT}(\mathsf{RCC5})$, $\mathsf{SAT}(\mathsf{RCC8})$ and $\mathsf{SAT}(\mathsf{IA})$ are NP-hard. Furthermore, $\mathsf{ConsDec}(\mathsf{RCC5})$, ConsDec(RCC8) and ConsDec(IA) clearly belong to NP since we can build a proof that a QCN admits a $(\alpha, I)$-decomposition that is verifiable in polynomial time. Specifically, a proof can be a set of $\alpha$ QCNs with associated scenarios, which is verifiable by checking that the properties in Definition 2 are satisfied. □

For Point Algebra (PA) (Vilain, Kautz, and van Beek 1990), with $\mathsf{B} = \{<, =, >\}$, even though the problem SAT(PA) is tractable, we show that ConsDec(PA) is not.

**Theorem 2.** *The problem ConsDec(PA) is NP-complete.*

*Proof.* The proof that ConsDec(PA) belongs to NP is similar to the one provided in Corollary 1. To prove NP-hardness, we provide an encoding of the well-known NP-hard problem of 3-coloring into ConsDec(PA). Let $G = (V, E)$ be an undirected graph s.t. $V = \{1, \ldots, n\}$. To define our encoding, we consider that each element of $V$ is a variable. Furthermore, we consider an additional variable $n + 1$. Thus, the set of variables of our encoding, denoted by $\mathcal{N}_G$, is $V \cup \{n + 1\}$. The constraints of $\mathcal{N}_G$ are defined as follows:

- for every $i \in V$, $\mathcal{N}_G[i, n + 1] = \{=\}$; and
- for every $\{i, j\} \in E$ with $i < j$, $\mathcal{N}_G[i, j] = \{<\}$.

Let $I = \{(i, j) : \{i, j\} \in E, i < j\}$. We now show that $G$ admits a 3-coloring iff $\mathcal{N}_G$ admits a $(3, I)$-decomposition. We start with the *only if part*. Let $\{V_r, V_g, V_b\}$ be a partition of $V$ that corresponds to a 3-coloring. Then, for each $c \in \{r, g, b\}$, we define the QCN $\mathcal{N}^c$ as follows:

- for every $i \in V_c$, $\mathcal{N}^c[i, n + 1] = \{=\}$;
- for every $\{i, j\} \in E$ with $i < j$, $\mathcal{N}^c[i, j] = \{<\}$; and
- for every $\{i, j\} \notin E$, $\mathcal{N}^c[i, j] = \{<, =, >\}$.

Clearly, $\mathcal{N}^c$ is inconsistent iff there exist $i, j \in V$ s.t. $\mathcal{N}^c[i, n + 1] = \{=\}$, $\mathcal{N}^c[j, n + 1] = \{=\}$ and $\{i, j\} \in E$ (which leads to $\mathcal{N}^c[i, j] = \{<\}$). Knowing that the vertices in $V_c$ can have the same color, we obtain that $\mathcal{N}^c$ is consistent for $c \in \{r, g, b\}$. It follows that $\{\mathcal{N}^r, \mathcal{N}^g, \mathcal{N}^b\}$ is a $(3, I)$-decomposition of $\mathcal{N}_G$. For the *if part*, we only need to associate a distinct color to each element of the considered $(3, I)$-decomposition: a vertex $i$ gets a color $c$ if $\mathcal{N}_G[i, n+1]$ belongs to the component associated with $c$. □

Theorem 2 may be applied to other polynomial (fragments of) calculi that embed PA, e.g., pointizable IA (Vilain, Kautz, and van Beek 1990; Ghallab and Alaoui 1989).

## 3.3 Optimization Versions

In this work, we are also interested in certain optimization versions of our original problem, viz., minimizing the number of components and maximizing the similarity among components, respectively.

**Minimizing Number of Components** We call a *minimum I-decomposition* of a QCN $\mathcal{N}$ any $(\alpha, I)$-decomposition of $\mathcal{N}$ where there is no $(\beta, I)$-decomposition s.t. $\beta < \alpha$.

The following theorem is mainly a consequence of Nash-Williams formula (Nash-Williams 1964). In particular, it provides an upper bound for the minimum number of components in the case where there is no required common part.

**Theorem 3.** *For every not trivially inconsistent QCN $\mathcal{N}$, there exists a $(\alpha, \emptyset)$-decomposition of $\mathcal{N}$ s.t. $\alpha = max\{\lceil e(G')/(v(G') - 1)\rceil : G' \in Ind(G_{\mathcal{N}})\}$, where $Ind(G_{\mathcal{N}})$ is the set of induced subgraphs of $G_{\mathcal{N}}$.*

*Proof.* Using the Nash-Williams formula, we know that the minimal number of forests covering the edges of $G_{\mathcal{N}}$ is equal to $\alpha = max\{\lceil e(G')/(v(G') - 1)\rceil : G' \in Ind(G_{\mathcal{N}})\}$. For every subgraph $H = (V, E')$ of $G_{\mathcal{N}}$, we define $\mathcal{N}_H = (V, C_H)$ as follows:

- for every $\{i, j\} \in E'$, $C_H(i, j) = \mathcal{N}[i, j]$; and
- for every $\{i, j\} \notin E'$, $C_H(i, j) = \mathsf{B}$.

Clearly, $H$ is the constraint graph of $\mathcal{N}_H$. Furthermore, using the fact that every not trivially inconsistent QCN having an acyclic constraint graph is consistent, we obtain that $\mathcal{N}_H$ is consistent for every acyclic subgraph $H$ of $G_{\mathcal{N}}$. Consequently, $\mathcal{N}$ admits a $(\alpha, \emptyset)$-decomposition. □

It is worth noting that the previous theorem gives an attainable upper bound; consider, for instance, any QCN in IA where each cycle in the constraint graph corresponds to a sequence of constraints of the form $i_1\{m\}i_2\{m\} \cdots i_{k-1}\{m\}i_k\{m\}i_1$. Indeed, in this case, a component is consistent if and only if its associated constraint graph is acyclic.

Using the fact that $max\{\lceil e(G')/(v(G') - 1)\rceil : G' \in Ind(G_{\mathcal{N}})\} \leq \lceil n/2\rceil$, we obtain the following property.

**Corollary 2.** *For every not trivially inconsistent QCN $\mathcal{N}$, there exists a $(\alpha, \emptyset)$-decomposition of $\mathcal{N}$ s.t. $\alpha \leq \lceil n/2\rceil$.*

The proposition below provides a more specific upper bound for the minimum number of components in the case of PA.

**Proposition 1.** *If $\mathcal{N}$ is a not trivially inconsistent QCN in PA, then $\mathcal{N}$ admits a $(3, \emptyset)$-decomposition.*

*Proof.* We define the decomposition $D = \{\mathcal{N}_>, \mathcal{N}_<, \mathcal{N}_=\}$ as follows: for every $b \in \{<, =, >\}$ and every $(i, j) \in [\![\mathcal{N}]\!]$, $\mathcal{N}_b[i, j] = \{b\}$. It is straightforward to show that the components of $D$ are all consistent. □

**Maximizing Similarity among Components** In this section, we focus on maximizing the similarity between the components of the decomposition. This can be seen as a way to stay close to the original QCN and to reduce the number of constraints that allow each component to be distinguished from the rest. The similarity is quantified through the common non-universal constraints: the larger the common part, the greater the similarity.

Given a $(\alpha, I)$-decomposition $D = \{\mathcal{N}_1, \ldots, \mathcal{N}_\alpha\}$ of a QCN $\mathcal{N}$, we use $\sigma(D)$ to denote the set $\{(i, j) \in [\![\mathcal{N}]\!]^s : \mathcal{N}_l[i, j] = \mathcal{N}[i, j]$ for every $l \in \{1, \ldots, \alpha\}\}$.

A $(\alpha, I)$-decomposition (resp. $I$-decomposition) $D$ is said to be *maximal* if there is no $(\alpha, I)$-decomposition (resp. $I$-decomposition) $D'$ s.t. $\sigma(D) \subsetneq \sigma(D')$. Moreover, $D$ is said to be a *maximum* $(\alpha, I)$*-decomposition* (resp. *maximum* $I$*-decomposition*) if there is no $(\alpha, I)$-decomposition (resp. $I$-decomposition) $D'$ s.t. $|\sigma(D)| < |\sigma(D')|$.

We now establish a relationship between the problem of finding a maximum decomposition and that of maximum matching in graph theory. Let us recall that a *matching* in a graph $G = (V, E)$ is a set of pairwise non-adjacent edges (i.e., no two edges share common vertices). A *maximum matching* is a matching of largest possible size. We use $\nu(G)$ to denote the matching number of a graph $G$, i.e., the size of a maximum matching. It deserves to be mentioned that the problem of finding a maximum matching is tractable using, for instance, the blossom algorithm that runs in $O(|V||E|^2)$ time (Edmonds 1965).

**Theorem 4.** *If $D$ is a maximum $\emptyset$-decomposition of a QCN $\mathcal{N}$, then $|\sigma(D)| \geq \nu(G_{\mathcal{N}})$.*

*Proof.* Let $M$ be a maximum matching of $G_{\mathcal{N}}$ and $S = \{(i,j) \in \llbracket\mathcal{N}\rrbracket : \{i,j\} \in M\}$. For every $(i,j) \in \llbracket\mathcal{N}\rrbracket^s \setminus S$, we define the QCN $\mathcal{N}^{ij}$ over the variables of $\mathcal{N}$ as follows:

- for every $(i',j') \in S \cup \{(i,j)\}$, $\mathcal{N}^{ij}[i',j'] = \mathcal{N}[i',j']$;
- for every $(i',j') \in \llbracket\mathcal{N}\rrbracket \setminus (S \cup \{(i,j)\})$, $\mathcal{N}^{ij}[i',j'] = \mathsf{B}$.

Since $M$ is a matching, we obtain for every $(i,j) \in \llbracket\mathcal{N}\rrbracket^s \setminus S$ that $G_{\mathcal{N}^{ij}}$ is an acyclic graph, and it follows that $\mathcal{N}^{ij}$ is consistent. So, $D = \{\mathcal{N}^{ij} : (i,j) \in \llbracket\mathcal{N}\rrbracket^s \setminus S\}$ is a $\emptyset$-decomposition of $\mathcal{N}$ where $|\sigma(D)| = \nu(G_{\mathcal{N}})$. □

Note that the previous theorem provides an attainable lower bound; consider, for example, a QCN $\mathcal{N}$ of four variables in $\mathsf{IA}$ where, $\forall\ (i,j) \in \llbracket\mathcal{N}\rrbracket, \mathcal{N}[i,j] = \{m\}$.

**Algorithm 1:** FINDDECOMPOSITION($\mathcal{N}, I, f$)

```
in   : A QCN $\mathcal{N} = (V, C)$, a set $I \subseteq \llbracket\mathcal{N}\rrbracket^s$, and a
       function $f \in \{\min, \max\}$
out  : A set $D$ of QCNs over $V$
$P \leftarrow \llbracket\mathcal{N}\rrbracket^s \setminus I$;
$D \leftarrow \emptyset$;
while $P \neq \emptyset$ do
    $\mathcal{N}' \leftarrow \mathcal{N}_{[i,j]/\mathsf{B}, \forall (i,j) \in \llbracket\mathcal{N}\rrbracket^s \setminus I}$ ;
    Let $(i,j) \in P$;
    $\mathcal{N}'[i,j] \leftarrow \mathcal{N}[i,j]$;
    if $\neg SAT(\mathcal{N}')$ then return $\emptyset$;
    $P' \leftarrow$ SATURATE($\mathcal{N}, \mathcal{N}', \llbracket\mathcal{N}\rrbracket^s \setminus \{(i,j)\}, f, D$);
    $P \leftarrow P \setminus (P' \cup \{(i,j)\})$;
    $D \leftarrow D \cup \{\mathcal{N}'\}$;
return $D$
```

**Algorithm 2:** FINDDECOMPOSITION($\mathcal{N}, f$)

```
in   : A QCN $\mathcal{N} = (V, C)$ and a function
       $f \in \{\min, \max\}$
out  : A set $D$ of QCNs over $V$
$P \leftarrow \llbracket\mathcal{N}\rrbracket^s$;
$D \leftarrow \emptyset$;
while $P \neq \emptyset$ do
    $\mathcal{N}' \leftarrow \top_V$;
    Let $(i,j) \in P$;
    $e_{ij} \leftarrow \{i, j, weight = |E(G_{\mathcal{N}})|\}$;
    $G \leftarrow$ MAXIMUMSPANNINGTREE($G_{\mathcal{N}} \cup \{e_{ij}\}$);
    $E \leftarrow \{(i,j) \mid \{i,j\} \in E(G) \wedge i < j\}$;
    for $(i,j) \in E$ do
        $\mathcal{N}'[i,j] \leftarrow \mathcal{N}[i,j]$;
    $P' \leftarrow$ SATURATE($\mathcal{N}, \mathcal{N}', \llbracket\mathcal{N}\rrbracket^s \setminus E, f, D$);
    $P \leftarrow P \setminus (P' \cup E)$;
    $D \leftarrow D \cup \{\mathcal{N}'\}$;
return $D$
```

**Algorithm 3:** SATURATE($\mathcal{N}, \mathcal{N}', P, f, D$)

```
in   : A QCN $\mathcal{N} = (V, C)$, a QCN $\mathcal{N}'$ with $\mathcal{N}' \supseteq \mathcal{N}$, a
       set $P \subseteq \llbracket\mathcal{N}\rrbracket^s$, a function $f \in \{\min, \max\}$, and
       a set $D$ of QCNs over $V$
out  : A set $P' \subseteq \llbracket\mathcal{N}\rrbracket^s$ s.t. $P' \subseteq P$
$P' \leftarrow \emptyset$;
while $P \neq \emptyset$ do
    Let $(i,j) \in \arg\mathrm{f}_{(i',j') \in P}(\Sigma_{\mathcal{M} \in D}[\mathcal{M}[i',j'] =
     \mathcal{N}[i',j']])$;
    $\mathcal{N}'[i,j] \leftarrow \mathcal{N}[i,j]$;
    if $SAT(\mathcal{N}')$ then
        $P' \leftarrow P' \cup \{(i,j)\}$;
    else
        $\mathcal{N}'[i,j] \leftarrow \mathsf{B}$;
    $P \leftarrow P \setminus \{(i,j)\}$;
return $P'$
```

# 4 Solving Approaches

In this section, we present our diverse approaches for obtaining consistent decompositions.

## 4.1 Greedy Constraint-based Methods

In Algorithms 1 and 2 we present two greedy approaches for tackling the problems of admitting decompositions and its optimization versions, viz., minimizing number of components and maximizing similarity among components.

Both of these algorithms operate as follows: so long as there are constraints that are not part of any component in the under-construction decomposition, a new component is created that is guaranteed to be consistent, and then it is consistently saturated with as many of the aforementioned constraints as possible. In the case of Algorithm 1, every new component is just a QCN comprising solely universal constraints (and, of course, the constraints in $I \subseteq \llbracket\mathcal{N}\rrbracket^s$, if they exist) and a single (other) non-universal constraint from the original inconsistent QCN. In the case of Algorithm 2, every new component is a QCN whose constraint graph corresponds to a spanning tree of the original QCN, computed using Kruskal's algorithm (Kruskal 1956); each spanning tree is differentiated by at least one new constraint from the original QCN. As the subset $I \subseteq \llbracket\mathcal{N}\rrbracket^s$ may in itself not form a tree (of constraints), and to maintain its simplicity, Algorithm 2 does not support the use of such a subset.

The consistent saturation of the components is performed by Algorithm 3, which iterates all constraints of the original

QCN and only keeps the ones that are characterized consistent by a SAT oracle. The most important aspect of the saturation algorithm is the function $f \in \{\min, \max\}$ used as parameter, which prioritizes the mininimization of components or the maximization of similarity among components. Specifically, this is done in the Iverson bracket in line 3 of the algorithm, which computes how many times a certain constraint appears in the decomposition; clearly, the max sum should lead to the maximization of similarity among components, and min to the minimization of their number.

## 4.2 Optimal SAT-based Encodings

First, we define our SAT encoding for deciding whether a given QCN $\mathcal{N} = (V, C)$ admits a $(\alpha, I)$-decomposition and prove its correctness. To this end, we associate $\alpha$ propositional variables $p_{ij}^{1,b}, \ldots, p_{ij}^{\alpha,b}$ with each base relation $b \in \mathsf{B}$ and each ordered pair of variables $(i,j) \in [\![\mathcal{N}]\!]$.

Our first formula says that the constraints in $I$ are satisfied by all the components in the $(\alpha, I)$-decomposition.

$$\bigwedge_{(i,j)\in I} \bigwedge_{l=1}^{\alpha} \bigvee_{b\in C(i,j)} p_{ij}^{l,b} \tag{1}$$

The formula below is used to ensure that each constraint of the original QCN occurs in at least one component in the $(\alpha, I)$-decomposition.

$$\bigwedge_{(i,j)\in[\![\mathcal{N}]\!]\setminus I} (\bigvee_{l=1}^{\alpha} \bigvee_{b\in C(i,j)} p_{ij}^{l,b}) \tag{2}$$

The following formulas are used to require that all components are consistent.

$$\bigwedge_{(i,j)\in[\![\mathcal{N}]\!]} \bigwedge_{l=1}^{\alpha} \sum_{b\in\mathsf{B}} p_{ij}^{l,b} = 1 \tag{3}$$

$$\bigwedge_{i,j,k\in V, i<j<k} \bigwedge_{l=1}^{\alpha} \bigwedge_{b,b'\in\mathsf{B}} (p_{ij}^{l,b} \wedge p_{jk}^{l,b'} \rightarrow \bigvee_{b''\in b\diamond b'} p_{ik}^{l,b''}) \tag{4}$$

Formula (3) states that a constraint can be satisfied by exactly one base relation in each component, and (4) ensures that each component admits a scenario by enforcing *algebraic closure* (Ligozat 2013) on every possible atomic QCN.

Note that the sum constraints in Formula (3) can be linearly encoded as CNF formulas in several ways (e.g., see (Sinz 2005)).

We use $\mathcal{E}(\mathcal{N}, \alpha, I)$ to denote the encoding consisting of the conjunction of Formulas (1)–(4).

For every model $\omega$ of $\mathcal{E}(\mathcal{N}, \alpha, I)$, the associated decomposition, denoted by $D_\omega$, is $\{\mathcal{N}_l : l \in \{1, \ldots, \alpha\}\}$ where for all $(i,j) \in [\![\mathcal{N}]\!]$,

$$\mathcal{N}_l[i,j] = \begin{cases} C(i,j) \text{ if } \exists b \in C(i,j) \text{ s.t. } \omega(p_{ij}^{l,b}) = 1 \\ \mathsf{B} \text{ otherwise} \end{cases}$$

**Proposition 2.** *The following properties are satisfied:*

1. *(Soundness) if $\omega \models \mathcal{E}(\mathcal{N}, \alpha, I)$, then $D_\omega$ is a $(\alpha, I)$-decomposition of $\mathcal{N}$;*
2. *(Completeness) if $\mathcal{E}(\mathcal{N}, \alpha, I)$ is unsatisfiable, then $\mathcal{N}$ does not admit any $(\alpha, I)$-decomposition.*

*Proof.*
*Soundness.* First, it is trivial that $D_\omega$ satisfies Property 1 in Definition 2. Furthermore, Property 2 is a consequence of Formula (1): for all $(i,j) \in I$ and $l \in \{1, \ldots, \alpha\}$, there is at least one base relation $b \in \mathcal{N}[i,j]$ s.t. $\omega(p_{ij}^{l,b}) = 1$, which leads to $\mathcal{N}_l[i,j] = \mathcal{N}[i,j]$. Similarly, Property 3 is a consequence of Formula (2). Formulas (3) and (4) are similar to the encoding of the consistency problem provided in (Pham, Thornton, and Sattar 2006) and shows that every element of $D_\omega$ is consistent. It follows that Property 4 is satisfied.
*Completeness.* Suppose that $\mathcal{N}$ admits a $(\alpha, I)$-decomposition $D = \{\mathcal{N}_1, \ldots, \mathcal{N}_\alpha\}$. Using Property 4 in Definition 2, each element of $D$ is consistent; we use $\mathcal{S}_l$ to refer to an arbitrary scenario of $\mathcal{N}_l$ for every $l \in \{1, \ldots, \alpha\}$. We associate with $D$ a Boolean interpretation $\omega_D$ of $\mathcal{E}(\mathcal{N}, \alpha, I)$ defined as follows: for every $l \in \{1, \ldots, \alpha\}$, every $(i,j) \in [\![\mathcal{N}]\!]$ and every $b \in \mathsf{B}$, $\omega_D(p_{ij}^{l,b}) = 1$ iff $\mathcal{S}_l[i,j] = b$. Then, using the properties in Definition 2, it is easy to check that $\omega_D$ is a model of $\mathcal{E}(\mathcal{N}, \alpha, I)$. For instance, $\omega_D$ satisfies Formula (1) thanks to Property 2: each scenario $\mathcal{S}_l$ satisfies the constraints occurring in $I$. □

Our encoding can be adapted to compute a minimum $\emptyset$-decomposition by using Partial MaxSAT. More precisely, we define the hard part as the clauses obtained from $\mathcal{E}(\mathcal{N}, \lceil n/2 \rceil, \emptyset)$ and the following formulas:

$$\bigwedge_{l=2}^{\lceil n/2 \rceil} [(\bigvee_{(i,j)\in[\![\mathcal{N}]\!]^s} \bigwedge_{l'<l} (\bigvee_{b\in\mathsf{B}\setminus C(i,j)} p_{ij}^{l',b})) \rightarrow q_l] \tag{5}$$

$$\bigwedge_{l=2}^{\lceil n/2 \rceil - 1} (\neg q_l \rightarrow \neg q_{l+1}) \tag{6}$$

where $q_1, \ldots, q_{\lceil n/2 \rceil}$ are fresh propositional variables: $q_l$ is used to decide whether the number of needed components is greater than or equal to $l$. Formula (5) states that if the first $l-1$ components do not cover all the constraints of the original QCN, then we need at least $l$ components. Formula (6) simply says that if we need less than $l$ components, then we necessarily need less than $l+1$ components. The soft part consists of the following unit clauses: $\neg q_l$ for $l = 2, \ldots, \lceil n/2 \rceil$. The unit clause $\neg q_1$ does not belong to the set of soft clauses since we need at least one component in any decomposition. Note that the bound $\lceil n/2 \rceil$ comes from Corollary 1. It is possible to obtain a smaller upper bound using Theorem 3.

More generally, to compute a minimum $I$-decomposition, we only need to use $|[\![\mathcal{N}]\!]^s| - |I \cap [\![N]\!]^s|$ instead of $\lceil n/2 \rceil$. This comes from the fact that each component must satisfy at least one non-universal constraint that does not belong to the set $I$.

In addition, to compute maximal and maximum $(\alpha, I)$-decompositions, we use SAT-based encodings where we particularly involve the problems of X-minimal model computation and Partial MaxSAT.

To define the notion of $X$-minimal model, where $X$ is a set of propositional variables, we use the preorder relation $\preceq_X$ over the Boolean interpretations defined as follows: $\omega \preceq_X \omega'$ if $\{p \in X : \omega(p) = 1\} \subseteq \{p \in X : \omega'(p) = 1\}$.

**Definition 3** (X-minimal Model (Avin and Ben-Eliyahu-Zohary 2001))**.** *Let $\phi$ be a propositional formula and $X$ a subset of propositional variables. An $X$-minimal model of $\phi$ is a model $\omega$ of $\phi$ where there is no model $\omega'$ of $\phi$ s.t. $\omega' \preceq_X \omega$ and $\omega' \not\preceq_X \omega$ .*

Our encoding to compute the maximal $(\alpha, I)$-decompositions is obtained by extending the encoding $\mathcal{E}(\mathcal{N}, \alpha, I)$. To this end, we associate with every $(i, j) \in [\![\mathcal{N}]\!]^s$ a distinct variable $r_{ij}$, which is used to capture whether or not the constraint between $i$ and $j$ in the original QCN belongs to the common part. Then, the encoding, denoted by $\mathcal{E}_m(\mathcal{N}, \alpha, I)$, is obtained by adding to $\mathcal{E}(\mathcal{N}, \alpha, I)$ the following formula:

$$\bigwedge_{(i,j)\in[\![\mathcal{N}]\!]^s} (\neg r_{ij} \rightarrow \bigwedge_{l=1}^{\alpha} \bigvee_{b\in C(i,j)} p_{ij}^{l,b}) \qquad (7)$$

This formula states that if $r_{ij}$ is false, then the constraint between $i$ and $j$ in the original QCN occurs in the common part. We use negative literals in the left-hand sides of the implications to relate maximizing similarity to minimizing the number of variables of the form $r_{ij}$ that are assigned the truth value 1.

**Proposition 3.** *Let $\mathcal{N} = (V, C)$ be a QCN and $\omega$ a model of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$. Then, $\omega$ is an $X$-minimal model of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$, with $X = \{r_{i,j} : (i, j) \in [\![\mathcal{N}]\!], \mathcal{N}[i, j] \neq \mathsf{B}\}$, iff $D_\omega$ is a maximal $(\alpha, I)$-decomposition of $\mathcal{N}$.*

*Proof.*
*Part* $\Rightarrow$. Using $\omega \models \mathcal{E}(\mathcal{N}, \alpha, I)$ and Proposition 2, we obtain that $D_\omega$ is a $(\alpha, I)$-decomposition of $\mathcal{N}$. For the sake of contradiction, assume that there exists a $(\alpha, I)$-decomposition $D = \{\mathcal{N}_1, \dots, \mathcal{N}_\alpha\}$ of $\mathcal{N}$ s.t. $\sigma(D_\omega) \subsetneq \sigma(D)$. Then, using scenarios of $\mathcal{S}_1, \dots, \mathcal{S}_\alpha$ of $\mathcal{N}_1, \dots, \mathcal{N}_\alpha$, respectively, we build a model $\omega_D$ of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$ as follows: the truth values of the variables of the form $p_{ij}^{l,b}$ are obtained in the same way as for the interpretation $\omega_D$ built in the proof of Proposition 2; and for all $r_{ij} \in X, \omega(r_{ij}) = 0$ iff $\mathcal{S}_l[i, j] \in \mathcal{N}[i, j]$ for every $l \in \{1, \dots, \alpha\}$. Hence it follows $\{p \in X : \omega_D(p) = 1\} \subsetneq \{p \in X : \omega(p) = 1\}$. Consequently, $\omega$ is not an $X$-minimal model of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$, and we get a contradiction.
*Part* $\Leftarrow$. It is mainly a consequence of the fact that if $\exists\ \omega'$ of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$ with $\{p \in X : \omega'(p) = 1\} \subsetneq \{p \in X : \omega(p) = 1\}$, then $D_{\omega'}$ is a $(\alpha, I)$-decomposition (Proposition 2) and $\sigma(D_\omega) \subsetneq \sigma(D_{\omega'})$, which means that $D_\omega$ is not a maximal $(\alpha, I)$-decomposition. □

A maximum $(\alpha, I)$-decomposition can be found using the framework of Partial MAxSAT. Indeed, the hard part consists of the clauses of $\mathcal{E}_m(\mathcal{N}, \alpha, I)$, and the soft part is simply the set of unit clauses $\neg r_{ij}$ for $(i, j) \in [\![\mathcal{N}]\!]^s$: maximizing the number of satisfied clauses of the form $\neg r_{ij}$ corresponds to maximizing the size of the common part.

To compute a maximum $I$-decomposition, we only need to consider a value of $\alpha$ that allows us to capture every possible common part between the components of a decomposition. Since the common part can correspond to only the constraints associated with the elements of $I$, a maximum $I$-decomposition can be obtained by computing a maximum $(|[\![\mathcal{N}]\!]^s| - |I \cap [\![\mathcal{N}]\!]^s|, I)$-decomposition. Moreover, using Theorem 4, a maximum $\emptyset$-decomposition is any maximum $(|[\![\mathcal{N}]\!]^s| - |\nu(G_{\mathcal{N}})|, \emptyset)$-decomposition.

**Observation** An encoding for solving the MaxQCN problem (Condotta et al. 2015) can be obtained from the one used to compute a maximum $(1, \emptyset)$-decomposition by removing Formula (2), which is used to cover all the constraints in the original QCN. However, we must point out that more efficient encodings exist if one needs to only focus on the MaxQCN problem, see (Westphal, Hué, and Wölfl 2013; Condotta, Nouaouri, and Sioutis 2016) for example.

## 5 Inconsistency Measurement

In the literature, inconsistency measures are defined as functions that associate non-negative values with knowledge bases to quantify the amount of conflict. Many proposals for measures and systems for defining them have been made using a variety of approaches (e.g., see (Grant and Hunter 2011; Grant and Hunter 2013; Ammoura et al. 2017; Bona et al. 2019)). To the best of our knowledge, there is a unique work in the literature on inconsistency measurement in qualitative spatial and temporal reasoning (Condotta, Raddaoui, and Salhi 2016). However, in the realm of temporal reasoning, a recent work has extended an inconsistency measure based on paraconsistency to linear temporal logic (Corea, Grant, and Thimm 2022). Thus, our contribution in this section adds value to the existing literature with new measures in QSTR.

The following definition is used to present rationality postulates introduced in (Condotta, Raddaoui, and Salhi 2016). These postulates are similar to that of Free Formula Independence proposed in (Hunter and Konieczny 2010).

**Definition 4** (C-Relaxation)**.** *Let $\mathcal{N} = (V, C)$ be a QCN. A* C-relaxation *of $\mathcal{N}$ is a QCN $\mathcal{N}' = (V', C')$ s.t. $V = V'$ and $\mathcal{N} \subseteq \mathcal{N}'$. A C-relaxation $\mathcal{N}'$ is said to be* trivial *if for all $i, j \in V$, if $\mathcal{N}'[i, j] \neq \mathcal{N}[i, j]$ then $\mathcal{N}'[i, j] = \mathsf{B}$. A (trivial) C-relaxation is* consistent *if it is a consistent QCN.*

A *minimal consistent C-relaxation* (resp. minimal consistent trivial C-relaxation) of a QCN $\mathcal{N}$ is a consistent C-relaxation (resp. consistent trivial C-relaxation) $\mathcal{N}'$ such that there exists no consistent C-relaxation (resp. consistent trivial C-relaxation) $\mathcal{N}''$ s.t. $\mathcal{N}'' \subsetneq \mathcal{N}'$.

Given a QCN $\mathcal{N} = (V, C)$, a pair of variables $p = \{i, j\}$ is said to be *a free constraint* (resp. *a T-free constraint*) in $\mathcal{N}$ if for every minimal consistent C-relaxation (resp. minimal consistent trivial C-relaxation) $\mathcal{N}'$, $\mathcal{N}'[i, j] = \mathcal{N}[i, j]$. We use $FreeC(\mathcal{N})$ and $TFreeC(\mathcal{N})$ to denote the set of free constraints and that of T-free constraints, respectively. It is straightforward to show that $FreeC(\mathcal{N}) \subseteq TFreeC(\mathcal{N})$.

We define an inconsistency measure as a mapping $\mathcal{I}$ from the set of QCNs to $\mathbb{R}^+_\infty$, i.e., the set of positive real num-

| $\mathcal{I}$ | **CO** | **DO** | **MO** | **FC** & **TFC** | **PY** | **SA** |
|---|---|---|---|---|---|---|
| $\mathcal{I}_1$ | ✓ | ✓ | ✓ | ✓ | X | X |
| $\mathcal{I}_{hs}$ | ✓ | ✓ | ✓ | ✓ | X | X |
| $\mathcal{I}_2$ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| $\mathcal{I}_{mcc}$ | ✓ | X | ✓ | ✓ | X | ✓ |

Table 1: Compliance of inconsistency measures with postulates.

bers augmented with a greatest element denoted by $\infty$. The definition of inconsistency measures is often driven by rationality postulates. In this section, we consider the following set of fundamental postulates that mirror those introduced in the propositional case:

- Consistency (**CO**): for every QCN $\mathcal{N} = (V, C)$, $\mathcal{I}(\mathcal{N}) = 0$ iff $\mathcal{N}$ is consistent;
- Dominance (**DO**): for all QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ with $\mathcal{N}' \subseteq \mathcal{N}$, $\mathcal{I}(\mathcal{N}) \leq \mathcal{I}(\mathcal{N}')$;
- Monotonicity (**MO**): for every QCN $\mathcal{N} = (V, C)$ and every $V' \subseteq V$, $\mathcal{I}(\mathcal{N}{\downarrow_{V'}}) \leq \mathcal{I}(\mathcal{N})$;
- Free Constraint (**FC**): for every QCN $\mathcal{N} = (V, C)$ and every $\{i, j\} \in FreeC(\mathcal{N})$, $\mathcal{I}(\mathcal{N}_{[i,j]/\mathsf{B}}) = \mathcal{I}(\mathcal{N})$;
- T-Free Constraint (**TFC**): for every QCN $\mathcal{N} = (V, C)$ and every $\{i, j\} \in TFreeC(\mathcal{N})$, $\mathcal{I}(\mathcal{N}_{[i,j]/\mathsf{B}}) = \mathcal{I}(\mathcal{N})$;
- Penalty (**PY**): for every QCN $\mathcal{N} = (V, C)$ and every $i, j \in V$ with $i \neq j$ and $\{i, j\} \notin FreeC(\mathcal{N})$, $\mathcal{I}(\mathcal{N}_{[i,j]/\mathsf{B}}) < \mathcal{I}(\mathcal{N})$;
- Super-Additivity (**SA**): for every QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V', C')$ with $V \cap V' = \emptyset$, $\mathcal{I}(\mathcal{N} \oplus \mathcal{N}') \geq \mathcal{I}(\mathcal{N}) + \mathcal{I}(\mathcal{N}')$, where $\mathcal{N} \oplus \mathcal{N}' = (V \cup V', C'')$ with $C''(i, j) = C(i, j)$ if $i, j \in V$, $C''(i, j) = C'(i, j)$ if $i, j \in V'$, and $C''(i, j) = \mathsf{B}$ in any other case.

Using $FreeC(\mathcal{N}) \subseteq TFreeC(\mathcal{N})$, we know that the property T-Free Constraint is stronger than Free Constraint.

Our decomposition-based inconsistency measures are defined as follows, where $Dec(\mathcal{N})$ is the set of $\emptyset$-decompositions of $\mathcal{N}$ and $min\ \emptyset = \infty$:

- $\mathcal{I}_1(\mathcal{N}) = min\{|D| - 1 : D \in Dec(\mathcal{N})\}$;
- $\mathcal{I}_2(\mathcal{N}) = min\{|[\![\mathcal{N}]\!]^s| - |\sigma(D)| : D \in Dec(\mathcal{N})\}$.

One can easily see that $\mathcal{I}_1$ and $\mathcal{I}_2$ can be computed through a single minimum $\emptyset$-decomposition and a single maximum $\emptyset$-decomposition, respectively.

Our two inconsistency measures can be seen as counterparts of measures introduced in the literature in the case of propositional knowledge bases. Indeed, $\mathcal{I}_1$ is similar to the measure $I_{hs}$, which is based on the minimum number of interpretations that satisfy all formulas in the knowledge base (Thimm 2016), whereas $\mathcal{I}_2$ is similar to the measure $I_{mcc}$, which is based on maximizing the number of shared formulas between maximal consistent subsets that cover the whole knowledge base (Ammoura et al. 2017).

| $d$ | $\text{Naive}^{\text{min}}_{\mathsf{IA}}$ | $\text{SpanT}^{\text{min}}_{\mathsf{IA}}$ | $\text{Naive}^{\text{max}}_{\mathsf{IA}}$ | $\text{SpanT}^{\text{max}}_{\mathsf{IA}}$ |
|---|---|---|---|---|
| 4 | $\frac{\mathbf{2} \mid 0.95}{85 \mid 1.8k}$ | $\frac{\mathbf{2} \mid 0.95}{\mathbf{67} \mid 1.7k}$ | $\frac{2.0\ (3) \mid 0.95}{85 \mid 1.8k}$ | $\frac{\mathbf{2} \mid \mathbf{0.95}}{\mathbf{67} \mid \mathbf{1.7k}}$ |
| 6 | $\frac{\mathbf{2} \mid 0.96}{128 \mid 3.2k}$ | $\frac{\mathbf{2} \mid 0.96}{\mathbf{110} \mid 3.1k}$ | $\frac{2.1\ (3) \mid \mathbf{0.97}}{132 \mid 3.0k}$ | $\frac{\mathbf{2} \mid 0.96}{\mathbf{110} \mid \mathbf{2.9k}}$ |
| 8 | $\frac{\mathbf{2} \mid \mathbf{0.95}}{167 \mid 4.0k}$ | $\frac{\mathbf{2} \mid 0.95}{\mathbf{148} \mid 4.0k}$ | $\frac{2.5\ (4) \mid 0.95}{212 \mid 4.2k}$ | $\frac{\mathbf{2} \mid 0.95}{\mathbf{148} \mid \mathbf{3.7k}}$ |
| 10 | $\frac{\mathbf{2} \mid 0.90}{202 \mid 4.5k}$ | $\frac{\mathbf{2} \mid 0.90}{\mathbf{180} \mid 4.5k}$ | $\frac{3.8\ (12) \mid \mathbf{0.90}}{392 \mid 6.5k}$ | $\frac{2.0\ (3) \mid 0.90}{185 \mid \mathbf{4.2k}}$ |
| 12 | $\frac{\mathbf{2} \mid 0.84}{237 \mid 5.0k}$ | $\frac{\mathbf{2} \mid 0.83}{\mathbf{210} \mid \mathbf{4.9k}}$ | $\frac{6.0\ (14) \mid \mathbf{0.84}}{716 \mid 10.9k}$ | $\frac{2.3\ (3) \mid 0.83}{249 \mid 5.1k}$ |
| 14 | $\frac{\mathbf{2} \mid 0.72}{277 \mid \mathbf{4.8k}}$ | $\frac{\mathbf{2} \mid 0.72}{\mathbf{245} \mid 4.8k}$ | $\frac{8.9\ (15) \mid \mathbf{0.74}}{1.2k \mid 15.6k}$ | $\frac{2.7\ (3) \mid 0.73}{346 \mid 6.1k}$ |

Table 2: Evaluation with $\mathsf{IA}$ networks of model $\mathsf{A}(n = 20, d, l = 6.5)$; we present $\frac{\text{avg. (max) \# of components} \mid \text{avg. similarity}}{\text{avg. \# of oracle calls} \mid \text{avg. \# of visited nodes}}$.

Table 1 presents the compliance of the four measures $\mathcal{I}_1$, $\mathcal{I}_{hs}$, $\mathcal{I}_2$, and $\mathcal{I}_{mcc}$ with the rationality postulates under consideration. A comparison between $\mathcal{I}_1$ and $\mathcal{I}_{hs}$ reveals their similarity. However, there is a disagreement between $\mathcal{I}_2$ and $\mathcal{I}_{mcc}$ regarding Dominance. In fact, the definition of this postulate bears resemblance to Weak Dominance rather than Dominance, and $\mathcal{I}_{mcc}$ satisfies Weak Dominance (Ammoura et al. 2017).

# 6 Experiments

With respect to the two flavours of $\emptyset$-decompositions discussed in this work, viz., *minimum* and *maximum* $\emptyset$-decompositions, for *minimizing* the number of components and *maximizing* the similarity among components, respectively, we implement and evaluate a variety of tools. Specifically, we evaluate two in-house implementations of the constraint-based FindDecomposition algorithms, one for the naive variant, viz., Naive, and one for the one utilizing spanning trees, viz., SpanT, respectively, and implementations of our SAT encodings using the PySAT toolkit (Ignatiev, Morgado, and Marques-Silva 2018) and the offered RC2 solver (Ignatiev, Morgado, and Marques-Silva 2019).

## 6.1 Dataset & Setup

We considered $\mathsf{RCC8}$ and $\mathsf{IA}$ networks generated by the standard $\mathsf{A}(n, d, l)$ model (Renz and Nebel 2001), used extensively in the literature. In short, $\mathsf{A}(n, d, l)$ creates networks of size $n$, constraint degree $d$, and an average number $l$ of base relations per constraint. We considered 100 inconsistent networks for *each* average node degree $d$ between 4 and 14 with a 2-degree step and for *each* of the calculi of $\mathsf{RCC8}$ and $\mathsf{IA}$; hence, 1 200 networks in total. For $\mathsf{RCC8}$, we have $n = 30$ and $l = 4.0$, and for $\mathsf{IA}$, $n = 20$ and $l = 6.5$. For this range of node degrees $d$, the networks of model $\mathsf{A}(n, d, l)$ are certain to lie within the *phase transition* region (Renz and Nebel 2001). The size of the networks is consistent with what has been used in the literature for similar optimization problems in order to present results that are as complete as possible, cf. (Condotta et al. 2015;

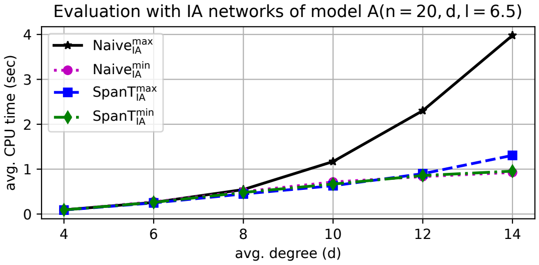


Figure 3: Runtime for the IA networks of Table 2.

Condotta, Nouaouri, and Sioutis 2016) (see also Table 3 here). For the experiments we used an Intel® Core™ CPU i7-12700H @ 4.70GHz, 16 GB of RAM, and the Ubuntu Linux 22.04 LTS OS, and one CPU core per network. All coding/running was done in Python 3.10.6; the code is available at: https://msioutis.gitlab.io/software/

### 6.2 Results & Remarks

In what follows, we only report on results pertaining to IA, as the ones for RCC8 are qualitatively similar.

The results for the constraint-based variants, along with the metrics used, are shown in Table 2 and Figure 3. Somewhat surprisingly, opting to minimize the number of components or maximize the similarity among components, i.e., setting the function $f$ to $\min$ or $\max$ in Algorithm 3, respectively, does not yield any notable difference with respect to the similarity achieved, which is already high in both cases (cf. last column in Table 3). This result is of course welcome, and it indicates that a single arbitrary iteration of the constraints of a QCN is sufficient to characterize a majority of consistent constraints. What is more, we found the use of the spanning trees to not provide any significant benefit, except in the case of maximizing similarity, where it clearly outperforms the Naive algorithm. This is because the Naive algorithm is too biased in favour of maximizing similarity by prioritizing the same (as in previous components) considered constraints in every call, whereas SpanT mitigates this issue via the use of a tree that spans over constraints guaranteed to be consistent.

The results for our SAT encodings are shown in Table 3; these are optimal with respect to *minimum* and *maximum* $\emptyset$-decompositions, and hence can be contrasted with the ones provided by the greedy constraint-based approaches. We can see that the greedy approaches already provide exceptionally good results accross all metrics, *if* optimality is of no or little concern; we only note a moderate deterioration with respect to maximizing similarity as the networks become denser, which is expected.

With respect to the given dataset, we can conclusively state that the problem of minimizing the number of components is very easy, and we conjecture that this would be the case in general too, as removing half of the constraints in any realistic constraint configuration would deem the network consistent. On the other hand, the problem of maximizing the similarity among components is very difficult, and we argue that it is also a very interesting problem to solve, in the sense that, given an inconsistent network, having many (almost) maximally consistent components to choose from lies at the heart of inconsistency resolution.

| $d$ | $\text{RC2}_{\text{IA}}^{\text{min}}$ | $\text{RC2}_{\text{IA}}^{\text{max}}$ | $\leq$ |
|---|---|---|---|
| 4 | $\mathbf{2} \mid 0.55 \bullet \mathbf{0.01s}$ | $\mathbf{2} \mid \mathbf{0.95} \bullet 0.02s$ | $\frac{0.95}{1}$ |
| 6 | $\mathbf{2} \mid 0.45 \bullet \mathbf{0.02s}$ | $2.0\ (3) \mid \mathbf{0.97} \bullet 0.12s$ | $\frac{0.97}{1.0\ (2)}$ |
| 8 | $\mathbf{2} \mid 0.47 \bullet \mathbf{0.04s}$ | $2.1\ (4) \mid \mathbf{0.97} \bullet 37.64s$ | $\frac{0.97}{1.3\ (4)}$ |
| 10 | $\mathbf{2} \mid 0.42 \bullet \mathbf{0.04s}$ | $2.4\ (4) \mid \mathbf{0.96} \bullet 131.41s\ (15)$ | $\frac{0.97}{2.5\ (9)}$ |
| 12 | $\mathbf{2} \mid 0.37 \bullet \mathbf{0.03s}$ | $2.6\ (5) \mid \mathbf{0.96} \bullet 446.67s\ (65)$ | $\frac{0.96}{4.4\ (10)}$ |
| 14 | $\mathbf{2} \mid 0.30 \bullet \mathbf{0.03s}$ | $? \mid ? \mid ? \bullet\ \text{inf}\ (100)$ | $\frac{0.93}{8.9\ (15)}$ |

Table 3: Evaluation with the IA networks of Table 2, using a $1h$ timeout per network; avg. (max) # of components | avg. similarity • avg. SAT solving time (# of timeouts), plus, in the last column, we present $\frac{\text{theoretical maximum similarity attainable}}{\text{avg. (max) \# of repairs needed (MaxQCN)}}$.

## 7 Conclusion and Perspectives

We presented and studied decomposition problems that aim to deal with inconsistency in QSTR. We provided several theoretical results on computational complexity, and bounds on the number of components and the number of common constraints. We introduced and implemented two approaches for solving the considered decomposition problems: greedy constraint-based algorithms and optimal SAT-based encodings. To demonstrate the interest of decomposition in dealing with inconsistency, we proposed two inconsistency measures, which can be seen as counterparts to measures introduced in the propositional case.

The need for inconsistency-tolerant systems is shown by the many ways inconsistency can occur in real-world situations. We believe that our work is an interesting step in this direction for QSTR-based systems and can be extended in several directions. It is worthwhile to study other versions of decompositions by tacking into account new parameters such as the number of solutions and the number of unspecified constraints in a component. Allowing inconsistent components in a decomposition by involving inconsistency measures is also an interesting perspective. Further, providing more compact SAT encodings based on tree decompositions and/or chordal graphs and/or available Horn theories could be a suitable extension. Finally, we would like to extend our toolkit with constraint-based solvers based on a notion of frozen constraints (Condotta, Ligozat, and Saade 2007), in order to fully support mandatory/structural constraints.

**Acknowledgements** The work was partially funded by the Agence Nationale de la Recherche (ANR) for the “Hybrid AI” project that is tied to the chair of Dr. Sioutis, and the I-SITE program of excellence of Université de Montpellier that complements the ANR funding.

## References

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26:832–843.

Ammoura, M.; Salhi, Y.; Oukacha, B.; and Raddaoui, B. 2017. On an MCS-based inconsistency measure. *Int. J. Approx. Reasoning* 80:443–459.

Avin, C., and Ben-Eliyahu-Zohary, R. 2001. Algorithms for Computing X-Minimal Models. In *LPNMR*.

Bennett, B. 1994. Spatial Reasoning with Propositional Logics. In *KR*.

Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

Bona, G. D.; Grant, J.; Hunter, A.; and Konieczny, S. 2019. Classifying inconsistency measures using graphs. *J. Artif. Intell. Res.* 66:937–987.

Brewka, G.; Dix, J.; and Konolige, K. 1997. *Nonmonotonic Reasoning: An Overview*, volume 73 of *CSLI Lecture Notes*. CSLI Publications, Stanford, CA.

Cai, S.; Luo, C.; Thornton, J.; and Su, K. 2014. Tailoring Local Search for Partial MaxSAT. In *AAAI*.

Condotta, J.; Kaci, S.; Marquis, P.; and Schwind, N. 2010. A Syntactical Approach to Qualitative Constraint Networks Merging. In *LPAR*, volume 6397, 233–247.

Condotta, J.; Mensi, A.; Nouaouri, I.; Sioutis, M.; and Said, L. B. 2015. A Practical Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *ICTAI*.

Condotta, J.-F.; Ligozat, G.; and Saade, M. 2007. Eligible and Frozen Constraints for Solving Temporal Qualitative Constraint Networks. In *CP*.

Condotta, J.; Nouaouri, I.; and Sioutis, M. 2016. A SAT Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *KR*.

Condotta, J.; Raddaoui, B.; and Salhi, Y. 2016. Quantifying Conflicts for Spatial and Temporal Information. In *KR*.

Corea, C.; Grant, J.; and Thimm, M. 2022. Measuring Inconsistency in Declarative Process Specifications. In *BPM*.

Dylla, F.; Lee, J. H.; Mossakowski, T.; Schneider, T.; van Delden, A.; van de Ven, J.; and Wolter, D. 2017. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Comput. Surv.* 50:7:1–7:39.

Edmonds, J. 1965. Paths, Trees, and Flowers. *Canadian Journal of Mathematics* 17:449–467.

Ghallab, M., and Alaoui, A. M. 1989. Managing Efficiently Temporal Relations Through Indexed Spanning Trees. In *IJCAI*.

Grant, J., and Hunter, A. 2011. Measuring the Good and the Bad in Inconsistent Information. In *IJCAI*.

Grant, J., and Hunter, A. 2013. Distance-Based Measures of Inconsistency. In *ECSQARU*.

Homem, T. P. D.; Santos, P. E.; Costa, A. H. R.; da Costa Bianchi, R. A.; and de Mántaras, R. L. 2020. Qualitative case-based reasoning and learning. *Artif. Intell.* 283:103258.

Hunter, A., and Konieczny, S. 2010. On the measure of conflicts: Shapley Inconsistency Values. *Artif. Intell.* 174:1007–1026.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2018. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2019. RC2: an Efficient MaxSAT Solver. *J. Satisf. Boolean Model. Comput.* 11:53–64.

Johnson, D. S. 1974. Approximation Algorithms for Combinatorial Problems. *J. Comput. Syst. Sci.* 9:256–278.

Kruskal, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.* 7:48–50.

Ligozat, G. 2013. *Qualitative Spatial and Temporal Reasoning*. ISTE. Wiley.

Miyazaki, S.; Iwama, K.; and Kambayashi, Y. 1996. Database Queries as Combinatorial Optimization Problems. In *CODAS*, 477–483.

Nash-Williams, C. S. J. A. 1964. Decomposition of Finite Graphs Into Forests. *Journal of the London Mathematical Society* s1-39:12–12.

Pham, D. N.; Thornton, J.; and Sattar, A. 2006. Towards an Efficient SAT Encoding for Temporal Reasoning. In *CP*.

Randell, D. A.; Cui, Z.; and Cohn, A. 1992. A Spatial Logic Based on Regions and Connection. In *KR*.

Renz, J., and Nebel, B. 2001. Efficient Methods for Qualitative Spatial Reasoning. *J. Artif. Intell. Res.* 15:289–318.

Sinz, C. 2005. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *CP*.

Sioutis, M., and Wolter, D. 2021. Qualitative Spatial and Temporal Reasoning: Current Status and Future Challenges. In *IJCAI*.

Suchan, J.; Bhatt, M.; and Varadarajan, S. 2021. Commonsense visual sensemaking for autonomous driving - On generalised neurosymbolic online abduction integrating vision and semantics. *Artif. Intell.* 299:103522.

Tanaka, K.; Berto, F.; Mares, E. D.; and Paoli, F., eds. 2013. *Paraconsistency: Logic and Applications*, volume 26 of *Logic, Epistemology, and the Unity of Science*. Springer.

Thimm, M. 2016. On the expressivity of inconsistency measures. *Artif. Intell.* 234:120–151.

Vilain, M.; Kautz, H.; and van Beek, P. 1990. Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers Inc. chapter Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report, 373–381.

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