On Neighbourhood Singleton-style Consistencies for Qualitative Spatial and Temporal Reasoning

Michael Sioutis[a,*], Anastasia Paparrizou[b], Tomi Janhunen[c]

[a] *Bamberg University, Faculty of Information Systems and Applied Computer Sciences, Bamberg, Germany*
[b] *Artois University, Centre de Recherche en Informatique de Lens (CRIL UMR 8188), Lens, France*
[c] *Tampere University, Faculty of Information Technology and Communication Sciences, Tampere, Finland*

Abstract

Given a qualitative constraint network (QCN), a singleton-style consistency is a local consistency that focuses on each base relation (atom) of a constraint separately, rather than the entire constraint altogether. More technically, such a consistency verifies if each base relation of each constraint of a QCN can serve as a support with respect to the closure of that network under a (naturally) weaker local consistency. This consistency is essential for tackling fundamental reasoning problems associated with QCNs, such as the satisfiability checking or the minimal labeling problem, but can suffer from redundant constraint checks, especially when those checks occur far from where the pruning usually takes place. In this paper, we propose singleton-style consistencies that are applied just on the neighbourhood of a singleton-checked constraint instead of the whole network. We make a theoretical comparison with existing consistencies and consequently prove some properties of the new ones. In addition, we propose algorithms to enforce our consistencies, as well as parsimonious variants thereof, that are more efficient in practice than the state of the art. We make an experimental evaluation with random and structured QCNs of Allen's Interval Algebra in the phase transition region to demonstrate the potential of our approach.

Keywords: Qualitative constraints, spatial and temporal reasoning, singleton-style consistencies; neighbourhood; minimal labeling problem

1. Introduction

Qualitative Spatial and Temporal Reasoning (QSTR) is a Symbolic AI approach that deals with the fundamental cognitive concepts of space and time

*Corresponding author

Email addresses: michail.sioutis@uni-bamberg.de (Michael Sioutis), paparrizou@cril.fr (Anastasia Paparrizou), tomi.janhunen@tuni.fi (Tomi Janhunen)

in a qualitative, human-like, manner [1, 2]. For instance, in natural language one uses expressions such as *inside*, *before*, and *north of* to spatially or temporally relate one object with another object or oneself, without resorting to providing quantitative information about these entities. More formally, QSTR restricts the vocabulary of rich mathematical theories that deal with spatial and temporal entities to simple qualitative constraint languages. Thus, QSTR provides a concise framework that allows for rather inexpensive reasoning about entities located in space and time and, hence, further boosts research and applications to a plethora of areas and domains that include, but are not limited to, dynamic GIS [3], cognitive robotics [4], deep learning [5], spatio-temporal design [6], qualitative model generation from video [7], ambient intelligence [8, 9], visual explanation [10] and sensemaking [11], semantic question-answering [12], qualitative simulation [13], spatio-temporal data mining [14, 15, 16], and modal logic [17, 18, 19]. The interested reader may look into a more comprehensive review of the emerging applications, the trends, and the future directions of QSTR in [20, 21]. In addition, a detailed survey of qualitative spatial and temporal calculi appears in [2].

Qualitative spatial or temporal information can be modeled as a *qualitative constraint network* (QCN), which is defined as a network where the vertices correspond to spatial or temporal entities, and the arcs are labelled with qualitative spatial or temporal relations respectively. For instance $x \leq y$ can be a temporal QCN over $\mathbb{Z}$. Two fundamental reasoning problems associated with a given QCN $\mathcal{N}$ are the problems of *satisfiability checking* and *minimal labeling* (or *deductive closure*) [22]. In particular, the satisfiability checking problem is about deciding if there exists a valuation of the variables of $\mathcal{N}$ that satisfies its constraints, such a valuation being called a *solution* of $\mathcal{N}$, and the minimal labeling problem concerns finding the strongest implied constraints and consequently obtaining its minimal sub-network. For instance, $x = 0 \wedge y = 1$ is one of the (infinitely many) solutions of the aforementioned QCN, and $x \leq y$ is already the strongest implied constraint as it is possible to have both solutions that satisfy $x < y$, e.g., $x = 0 \wedge y = 1$, and solutions that satisfy $x = y$, e.g., $x = 0 \wedge y = 0$; so, in fact, the QCN is minimal. In general, for many well known spatio-temporal calculi the satisfiability checking problem is NP-hard [23]. Further, the minimal labeling problem is polynomial-time Turing reducible to the satisfiability checking problem [24].

Motivation

In this paper, we focus mostly on the minimal labeling problem, which, since its introduction in 1974 by Montanari [25], has been studied in the domains of both CSPs [26, 27] and QCNs [28, 29]. A trivial example of how minimality applies to QCNs was presented earlier, and a more detailed one follows. As noted in [26], *a minimal network is a quite useful knowledge compilation, since it allows one to answer a number of queries in polynomial time that would otherwise be* NP*-hard*. Indeed, in the context of QSTR, for instance, one could exploit minimality of a QCN to immediately deduce whether a task A should be scheduled before a task C, or whether an object X could be placed on top of an

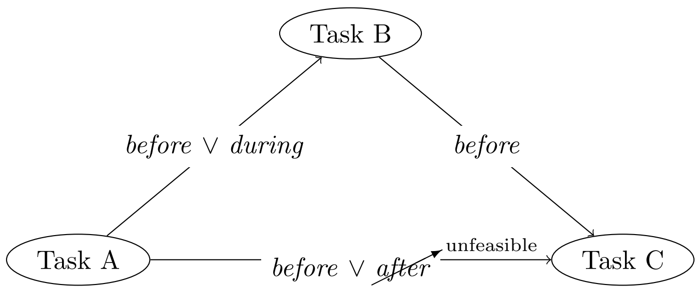


Figure 1: A QCN in simplified form

object Z. A visualization of the former example is provided in Figure 1; the initial QCN is not minimal, but becomes such by removing the base relation *after* from the constraint involving Tasks A and C, as that base relation is impossible to be satisfied by any solution. Difficult problems such as the minimal labeling problem and alike are, in general, either approximated by the use of local consistencies [27] or even solved by the aid of such consistencies [30]. In fact, in time-critical applications approximation may be the sole possibility, as solving the problem often takes significantly more time (it is NP-hard after all) and may only guarantee a marginally better result (if at all) in terms of minimal labeling (see the performance of Minimizer in Tables 1 and 2). Among the local consistencies introduced in the literature, we study *singleton-style consistencies* in the aforementioned context, which are consistencies that entail support for each base relation (atom) of the constraints of a QCN with respect to the closure of that network under a weaker local consistency (typically $\diamond_{G}$-consistency [31, 32]). Specifically, we investigate how these consistencies behave when the underlying weaker local consistency that they build upon is restricted to the neighbourhood of a singleton-checked constraint. As noted in [33], neighbourhood-based restrictions can hit the sweet spot between effectiveness and efficiency in singleton-style consistencies for CSPs; therefore, it is imperative that we introduce and study such restrictions in the context of QCNs as well, and consequently provide a foundation for further work in understanding these kinds of network structures, which have received much attention over the past years [2].

Contributions

Our contributions are fivefold and described as follows:

i) we enrich the family of consistencies for QCNs by proposing singleton-style consistencies that are applied just on the neighbourhood of the singleton-checked constraint instead of the entire network;

ii) we theoretically obtain a strength-based hierarchy among existing consistencies for QCNs and the novel ones;

iii) we present algorithms to enforce the proposed consistencies for QCNs, as well as parsimonious variants thereof;

iv) we make an experimental evaluation with random and structured QCNs of Interval Algebra to measure and compare the performance of all considered algorithms, especially in terms of how fast and how well they can independently approximate the minimal sub-network of a QCN;

v) we review the latest related work that exists in the area of traditional constraint programming, discuss similarities and differences with respect to our approach, and give some directions for future work.

Organization

The rest of the paper is organized as follows. In Section 2 we give some preliminaries on qualitative spatial and temporal reasoning. Next, in Section 3 we overview some known state-of-the-art local consistencies for QCNs. Then, in Section 4 we introduce, formally define, and thoroughly study the proposed neighbourhood-based consistencies for QCNs, and present the algorithms for enforcing these consistencies, as well as parsimonious variants thereof. In Section 5 we evaluate our approach with random and structured QCNs of Interval Algebra and comment on the outcome; one finding is that neighbourhood-focused singleton-style algorithms are around 30% faster in the phase transition region than the standard algorithms, and another one is that they exhibit an improved efficiency to effectiveness ratio of up to around 25%. Next, in Section 6 we review the latest related work that exists in the discussed direction. Finally, in Section 7 we draw some conclusive remarks and give directions for future work.

2. Preliminaries

A binary qualitative spatial or temporal constraint language, is based on a finite set B of *jointly exhaustive and pairwise disjoint* relations, called the set of *base relations* [34], that is defined over an infinite domain D. 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 D. The set B contains the identity relation 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^{B} represents the total set of relations. The set 2^{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 $\diamond$ [34]. 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., strongest) 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 $(x, z) \in b$ and $(z, y) \in b'\}$ is the (true) composition of b and b'. For all $r, r' \in 2^{\mathsf{B}}$, we have that $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r, b' \in r'\}$.

As an illustration, consider the well known qualitative temporal constraint language of Interval Algebra (IA), introduced by Allen [35]. Its domain is defined to be the set of intervals on $\mathbb{Q}$, i.e., $\mathsf{D} = \{x = (x^-, x^+) \in \mathbb{Q} \times \mathbb{Q} : x^- < x^+\}$.

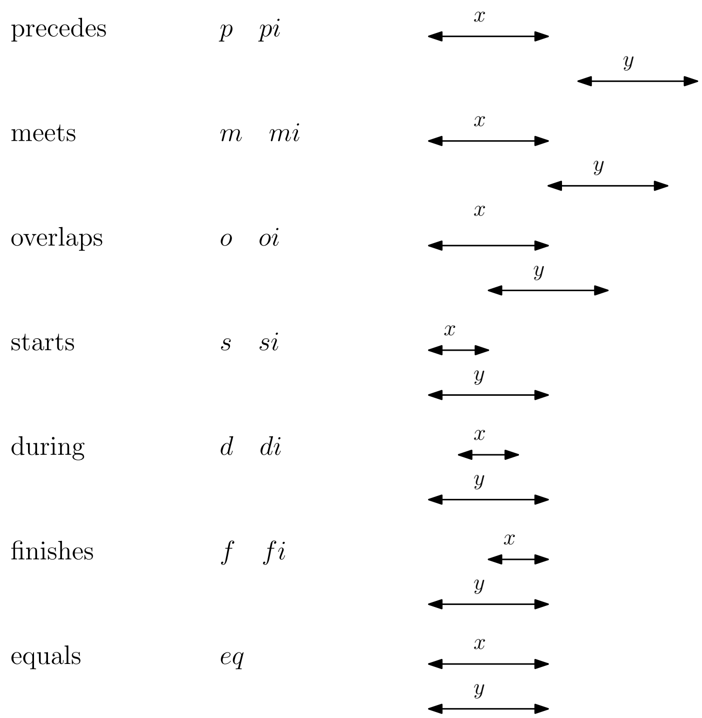
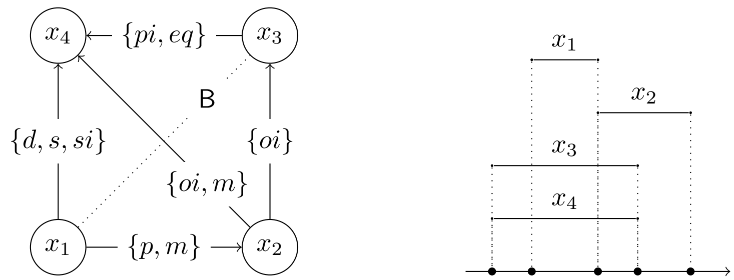


Figure 2: The base relations of IA; $\cdot i$ denotes the converse of $\cdot$

Then, IA considers such time intervals as its temporal entities, and the set of base relations $\mathsf{B} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$ as a means to encode knowledge about the temporal relations between the entities, as depicted in Figure 2. Specifically, each base relation represents a particular ordering of the four endpoints of two intervals on the timeline. For example, d, viz., *during*, is defined as $d = \{(x, y) \in \mathsf{D} \times \mathsf{D} \mid x^- > y^- \text{ and } x^+ < y^+\}$. Of those base relations, eq is the identity relation Id, for which it holds that $eq^{-1} = eq$. Typical applications of Interval Algebra involve—in addition to those listed for QSTR in the introduction—planning and scheduling [36, 37, 38, 39, 40], natural language processing [41, 42], temporal databases [43, 44], multimedia databases [45], molecular biology [24] (e.g., arrangement of DNA segments/intervals along a linear chain involves particular temporal-like problems [46]), workflow [47], and temporal diagnosis [48].

Notably, many (if not most) of the well known and well studied qualitative constraint languages, such as Interval Algebra [35] and RCC8 [49], are in fact *relation algebras* [23]. In what follows, we restrict ourselves to such calculi in order to facilitate discussion of the consistencies and of the algorithms for enforcing them.

Qualitative spatial or temporal information can be modeled as a *qualitative constraint network*, defined in the following manner:



(a) A satisfiable QCN $\mathcal{N}$ (b) A solution σ of $\mathcal{N}$

Figure 3: Figurative examples of QCN terminology using IA

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

- $V = \{v_1, \ldots, v_n\}$ is a non-empty finite set of variables;
- and C is a mapping $C : V \times V \rightarrow 2^{\mathsf{B}}$ such that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$ and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$.

An example of a QCN of IA is shown in Figure 3a; for clarity, converse relations as well as Id loops are not mentioned nor shown in the figure.

Definition 2. Let $\mathcal{N} = (V, C)$ be a QCN, then:

- a *solution* of $\mathcal{N}$ is a mapping $\sigma : V \rightarrow \mathsf{D}$ such that $\forall (u, v) \in V \times V$, $\exists b \in C(u, v)$ such that $(\sigma(u), \sigma(v)) \in b$ (see Figure 3b);
- $\mathcal{N}$ is *satisfiable* if and only if it admits a solution;
- a *sub*-QCN $\mathcal{N}'$ of $\mathcal{N}$, denoted by $\mathcal{N}' \subseteq \mathcal{N}$, is a QCN (V, C') such that $C'(u, v) \subseteq C(u, v)\ \forall u, v \in V$; if in addition $\exists u, v \in V$ such that $C'(u, v) \subset C(u, v)$, then $\mathcal{N}' \subset \mathcal{N}$;
- a base relation $b \in C(v, v')$ with $v, v' \in V$ is *feasible* (resp. *unfeasible*) in $\mathcal{N}$ if and only if there exists (resp. there does not exist) a solution $\sigma : V \rightarrow \mathsf{D}$ of $\mathcal{N}$ such that $(\sigma(v), \sigma(v')) \in b$;
- $\mathcal{N}$ is *minimal* if and only if $\forall v, v' \in V$ and $\forall b \in C(v, v')$, b is a feasible base relation in $\mathcal{N}$;
- the *constraint graph* of $\mathcal{N}$, denoted by $\mathsf{G}(\mathcal{N})$, is the graph (V, E) where $\{u, v\} \in E$ if and only if $C(u, v) \neq \mathsf{B}$ and $u \neq v$;
- $\mathcal{N}$ is the *empty* QCN on V, denoted by $\bot^V$, if and only if $C(u, v) = \emptyset$ for all $u, v \in V$ with $u \neq v$.

Let us further introduce the following operation that substitutes $C(v, v')$ with $r \in 2^{\mathsf{B}}$ in a given QCN:

- given a QCN $\mathcal{N} = (V, C)$ and $v, v' \in V$, we define that $\mathcal{N}_{[v,v']/r}$ with $r \in 2^{\mathsf{B}}$ yields the QCN $\mathcal{N}' = (V, C')$ defined by $C'(v, v') = r$, $C'(v', v) = r^{-1}$, and $\forall (u, u') \in (V \times V) \setminus \{(v, v'), (v', v)\}$, $C'(u, u') = C(u, u')$.

3. State-of-the-art Consistencies

We view a consistency ${}^{\phi}_{G}$, where ϕ is some operation (such as the *weak composition* operation) and G a graph, as a predicate on QCNs, i.e., a function that receives an input QCN and returns true or false depending on whether ${}^{\phi}_{G}$ holds on that QCN or not respectively. In what follows, given some operation ϕ (such as the weak composition operation) and a graph G, the unique $\subseteq$-maximal ${}^{\phi}_{G}$-consistent sub-QCN of $\mathcal{N}$ is called the *closure* of $\mathcal{N}$ under the consistency ${}^{\phi}_{G}$ and is denoted by ${}^{\phi}_{G}(\mathcal{N})$.

We recall the definition of ${}^{\diamond}_{G}$-consistency, which is a basic and widely used local consistency for reasoning with QCNs.

Definition 3. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be ${}^{\diamond}_{G}$-*consistent* if and only if $\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ we have that $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$.

Intuitively, ${}^{\diamond}_{G}$-consistency entails consistency for *all* triples of variables of a QCN that correspond to triangles of a given graph G. If G is the complete graph on the variables of a given QCN, then ${}^{\diamond}_{G}$-consistency becomes identical to $\diamond$-consistency [32], and, hence, $\diamond$-consistency can be seen as a special case of ${}^{\diamond}_{G}$-consistency.

In [50] the authors build upon ${}^{\diamond}_{G}$-consistency and propose a local consistency in the context of qualitative constraint-based reasoning that serves as the counterpart of *directional path consistency* in traditional constraint programming [51] or quantitative temporal reasoning [52], and is mainly distinguished by the fact that the involved consistency notions are tailored to handle infinite domains and qualitative relations. This local consistency is called ${}^{\overleftarrow{\diamond}}_{G}$-*consistency* and, in particular, it entails consistency for all *ordered* triples of variables of a QCN that correspond to triangles of a given graph G; this ordering can be specified by a bijection between the set of the variables of a QCN and a set of integers, and can be chosen randomly or via an algorithm or heuristic. We recall the formal definition of that consistency as follows:

Definition 4. Given a QCN $\mathcal{N} = (V, C)$, an ordering $(\alpha^{-1}(0), \alpha^{-1}(1), \ldots, \alpha^{-1}(n-1))$ of V defined by a bijection $\alpha : V \rightarrow \{0, 1, \ldots, n-1\}$, and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be ${}^{\overleftarrow{\diamond}}_{G}$-*consistent* if and only if $\forall v_i, v_j, v_k \in V$ such that $\{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ and $\alpha(v_i), \alpha(v_j) < \alpha(v_k)$ we have that $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$.

Since ${}^{\overleftarrow{\diamond}}_{G}$-*consistency* is basically ${}^{\diamond}_{G}$-consistency restricted to some ordering of the triples of variables of a given QCN, it is expected that it will perform worse than ${}^{\diamond}_{G}$-consistency in terms of tackling the satisfiability checking or the minimal labeling problem of that QCN, in the general case. However, that behaviour of ${}^{\overleftarrow{\diamond}}_{G}$-*consistency* in the context of the aforementioned reasoning problems for *arbitrary* QCNs has yet to be investigated (cf. [53]), and we shall use this work as an opportunity to do so (see Section 5).

We continue with the presentation of some state-of-the-art singleton-style consistencies. Given a graph $G = (V', E)$, where $V' \subseteq V$, a QCN $\mathcal{N} = (V, C)$ is $\blacklozenge_G$-consistent if and only if for every pair of variables $\{v, v'\} \in E$ and every base relation $b \in C(v, v')$, after instantiating $C(v, v')$ with $\{b\}$ as the singleton and applying $\diamond_G$-consistency on $\mathcal{N}$, the revised constraint $C(v, v')$ is always supported by $\{b\}$. Formally, $\blacklozenge_G$-consistency of a QCN is defined as follows:

Definition 5. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be $\blacklozenge_G$*-consistent* if and only if $\mathcal{N}$ is $\diamond_G$-consistent and $\forall\{v, v'\} \in E$ and $\forall b \in C(v, v')$ we have that $C'(v, v') = \{b\}$, where $(V, C') = \diamond_G(\mathcal{N}_{[v,v']/\{b\}})$.

If G is the complete graph on the variables of a given QCN, we can easily verify that $\blacklozenge_G$-consistency corresponds to $\diamond_{\mathsf{B}}$-consistency of the family of $\diamond_f$-consistencies studied in [54]. Interestingly, $\blacklozenge_G$-consistency for QCNs can also be seen as a counterpart of *Singleton Arc Consistency* (SAC) [55] for CSPs.

Finally, in [56] the authors define a local consistency that is more restrictive than any of the *practical*[1] local consistencies known to date for QCNs, called *collective* $\blacklozenge_G$*-consistency*, or $\blacklozenge^{\cup}_G$-consistency for short. This singleton-style consistency is inspired by k*-partitioning consistency* for CSPs [58]. We recall the formal definition of that consistency as follows:

Definition 6. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be $\blacklozenge^{\cup}_G$*-consistent* if and only if $\mathcal{N}$ is $\blacklozenge_G$-consistent and $\forall\{v, v'\} \in E$, $\forall b \in C(v, v')$, and $\forall\{u, u'\} \in E$ we have that $\exists b' \in C(u, u')$ such that $b \in C'(v, v')$, where $(V, C') = \diamond_G(\mathcal{N}_{[u,u']/\{b'\}})$.

This underlying filtering condition of $\blacklozenge^{\cup}_G$-consistency is based on relation partitioning combined with $\diamond_G$-consistency, and allows for improved pruning capability over $\blacklozenge_G$-consistency [56].

4. Neighbourhood Singleton-style Consistencies

In this section we propose and study a variety of singleton-style consistencies that are applied just on the neighbourhood of the singleton-checked constraint instead of the whole network.

Before doing so, let us first formally introduce a preorder in order to compare the pruning (or inference) capability of different consistencies. Let ϕ_G and ψ_G be two consistencies defined by some operations ϕ and ψ respectively and a graph G. Then, ϕ_G is *stronger* than ψ_G if and only if whenever ϕ_G holds on a QCN $\mathcal{N}$ with respect to a graph G, ψ_G also holds on $\mathcal{N}$ with respect to G, and ϕ_G is *strictly stronger* than ψ_G if and only if ϕ_G is stronger than ψ_G and there exists at least one QCN $\mathcal{N}$ and a graph G such that ψ_G holds on $\mathcal{N}$ with respect to G, but ϕ_G does

[1] Clearly, in special cases notions like k-*consistency* can be defined and exploited theoretically [57], but these can be hardly implemented efficiently and are therefore not suitable for applications.

not hold on $\mathcal{N}$ with respect to G. (The terms *weaker* and *strictly weaker* can be defined likewise.) Finally, ${}^{\phi}_{G}$ and ${}^{\psi}_{G}$ are *incomparable* if and only if there exist QCNs $\mathcal{N}$ and $\mathcal{N}'$ such that ${}^{\phi}_{G}$ is strictly stronger than ${}^{\psi}_{G}$ with respect to $\mathcal{N}$ and some graph G, and ${}^{\phi}_{G}$ is strictly weaker than ${}^{\psi}_{G}$ with respect to $\mathcal{N}'$ and some graph G (we note that the graph G can be different in the two cases). Finally, ${}^{\phi}_{G}$ and ${}^{\psi}_{G}$ are *equivalent* if and only if we have that ${}^{\phi}_{G}$ is both stronger and weaker than ${}^{\psi}_{G}$, and vice versa.

In general, standard singleton-style consistencies can make a lot of redundant checks, which consequently can slow down their efficacy. It has been observed in the domain of CSPs that the majority of constraint revisions occur close to the relation that is being singleton checked, and rarely too far from it [33]. For that purpose, constraint programming researchers have proposed weaker singleton-style consistencies that localize propagation to the neighbourhood of the variable at hand [33, 59]. Neighbourhood singleton-style consistencies for CSPs, despite being strictly weaker than SAC [55] in general, can produce almost as much filtering as SAC with substantially less computational cost on many problems [59]. In what follows, we define two neighbourhood singleton-style consistencies for QCN*s*, and then we proceed to present algorithms and parsimonious variants thereof for applying these consistencies efficiently.

In order to define the new consistencies, we first need to define what exactly is meant by "neighbourhood of a relation" in the context of QCNs. Informally, given a QCN $\mathcal{N}$ and a graph G, the neighbourhood of a relation in $\mathcal{N}$ comprises all the triangles that involve the corresponding edge in G, and all the edges among the vertices of those triangles as well. Noting that in a given graph $G = (V, E)$, for each $u \in V$ the set of adjacent vertices of u, denoted by $\mathsf{adj}(u)$, is the set $\{w \mid \{u, w\} \in E\}$, we can formally define the neighbourhood of a relation of a QCN as follows:

Definition 7. Given a QCN $\mathcal{N} = (V, C)$, a graph $G = (V', E)$, where $V' \subseteq V$, and two variables $v, v' \in V$ such that $\{v, v'\} \in E$, the *neighbourhood of* $C(v, v')$, denoted by $G_{\mathsf{N}(vv')}$, is the induced subgraph $G[S]$, where $S = (\mathsf{adj}(v) \cap \mathsf{adj}(v')) \cup \{v, v'\}$.

As an example, consider the QCN $\mathcal{N}$ and an accompanying graph as described in Figure 4. The neighbourhood of $C(x_1, x_3)$ is the induced subgraph of the set of vertices $\{x_1, x_2, x_3, x_4\}$.

With the aforementioned definition in mind, we can define the notion of *neighbourhood* ${}^{\blacklozenge}_{G}$*-consistency* as follows:

Definition 8. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be *neighbourhood* ${}^{\blacklozenge}_{G}$*-consistent*, or $\mathsf{N}{}^{\blacklozenge}_{G}$-consistent for short, if and only if $\mathcal{N}$ is ${}^{\diamond}_{G}$-consistent and $\forall \{v, v'\} \in E$ and $\forall b \in C(v, v')$ we have that $C'(v, v') = \{b\}$, where $(V, C') = {}^{\diamond}_{G_{\mathsf{N}(vv')}}(\mathcal{N}_{[v,v']/\{b\}})$.

Similarly, we can define the notion of *neighbourhood* ${}^{\blacklozenge\cup}_{G}$*-consistency* as follows:

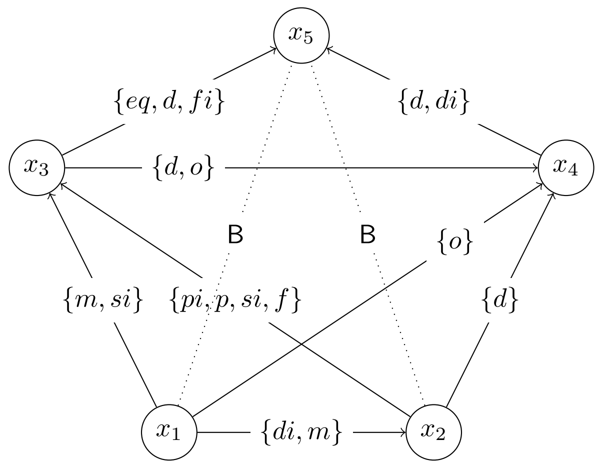


Figure 4: Given the QCN $\mathcal{N} = (V, C)$ above and the graph G that results by removing the edge $\{x_1, x_5\}$ from the complete graph on V, we have that $\mathcal{N}$ is neighbourhood $\blacklozenge^{\cup}_{G}$-consistent (and neighbourhood $\blacklozenge_{G}$-consistent), but not $\blacklozenge_{G}$-consistent (or $\blacklozenge^{\cup}_{G}$-consistent)

Definition 9. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be *neighbourhood* $\blacklozenge^{\cup}_{G}$*-consistent*, or $\mathsf{N}\blacklozenge^{\cup}_{G}$-consistent for short, if and only if $\mathcal{N}$ is $\mathsf{N}\blacklozenge_{G}$-consistent and $\forall \{v, v'\} \in E$, $\forall b \in C(v, v')$, and $\forall \{u, u'\} \in E$ we have that $\exists b' \in C(u, u')$ such that $b \in C'(v, v')$, where $(V, C') = \lozenge_{G_{\mathsf{N}(vv')}}(\mathcal{N}_{[u,u']/\{b'\}})$.

The reader can note that Definitions 8 and 9 mirror Definitions 5 and 6 respectively, the difference being that the closure under $\lozenge_{G}$-consistency is restricted to the neighbourhood of the constraint at hand.

We recall the following result from [56] in our effort here to build a strength-based hierarchy among all discussed consistencies:

Proposition 1 ([56])**.** *$\blacklozenge^{\cup}_{G}$-consistency is strictly stronger than $\blacklozenge_{G}$-consistency.*

In the sequel, Figure 4 will be crucial in proving some results that follow.

Proposition 2. *$\blacklozenge^{\cup}_{G}$-consistency is strictly stronger than $\mathsf{N}\blacklozenge^{\cup}_{G}$-consistency.*

Proof. Consider the QCN along with its accompanying graph depicted in Figure 4. As noted in the caption of the figure, the QCN is $\mathsf{N}\blacklozenge^{\cup}_{G}$-consistent and $\mathsf{N}\blacklozenge_{G}$-consistent, but not $\blacklozenge_{G}$-consistent or $\blacklozenge^{\cup}_{G}$-consistent. Specifically, in order for the QCN to become $\blacklozenge^{\cup}_{G}$-consistent and $\blacklozenge_{G}$-consistent, the base relation mi needs to be removed from $C(x_2, x_5)$. In addition, by the definitions of $\blacklozenge^{\cup}_{G}$-consistency and $\mathsf{N}\blacklozenge^{\cup}_{G}$-consistency, we have that every $\blacklozenge^{\cup}_{G}$-consistent QCN is $\mathsf{N}\blacklozenge^{\cup}_{G}$-consistent. Specifically, given a QCN $\mathcal{N}$ and two graphs G and G' such that $G' \subseteq G$, it holds that if $\mathcal{N}$ is $\lozenge_{G}$-consistent then $\mathcal{N}$ is $\lozenge_{G'}$-consistent. □

Following the same line of reasoning as that of the proof of Proposition 2, we assert the next result:

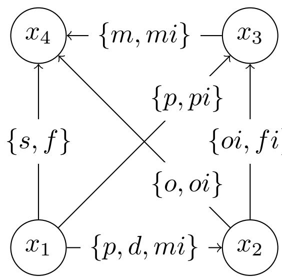


Figure 5: Given the QCN $\mathcal{N} = (V, C)$ above and the complete graph on V, we have that $\mathcal{N}$ is $\blacklozenge_G$-consistent (and neighbourhood $\blacklozenge_G$-consistent), but not neighbourhood $\blacklozenge_G^{\cup}$-consistent (or $\blacklozenge_G^{\cup}$-consistent)

Proposition 3. *$\blacklozenge_G$-consistency is strictly stronger than $\mathsf{N}\blacklozenge_G$-consistency.*

We proceed with presenting the next result:

Proposition 4. *$\mathsf{N}\blacklozenge_G^{\cup}$-consistency is strictly stronger than $\mathsf{N}\blacklozenge_G$-consistency.*

Proof. Consider the QCN along with its accompanying graph depicted in Figure 5. It is the case that the QCN is $\mathsf{N}\blacklozenge_G$-consistent, but not $\mathsf{N}\blacklozenge_G^{\cup}$-consistent. Specifically, in order for the QCN to become $\mathsf{N}\blacklozenge_G^{\cup}$-consistent, the base relation d needs to be removed from $C(x_1, x_2)$. Additionally, by definition of $\mathsf{N}\blacklozenge_G^{\cup}$-consistency, we have that every $\mathsf{N}\blacklozenge_G^{\cup}$-consistent QCN is $\mathsf{N}\blacklozenge_G$-consistent. □

We continue with another result as follows:

Proposition 5. *$\mathsf{N}\blacklozenge_G^{\cup}$-consistency is incomparable to $\blacklozenge_G$-consistency.*

Proof. Consider again the QCN along with its accompanying graph depicted in Figure 5. For the same reason remarked in the proof of Proposition 4, it is the case that the QCN is $\blacklozenge_G$-consistent, but not $\mathsf{N}\blacklozenge_G^{\cup}$-consistent. On the other hand, as noted in the proof of Proposition 2, the QCN of Figure 4 is $\mathsf{N}\blacklozenge_G^{\cup}$-consistent, but not $\blacklozenge_G$-consistent, with respect to its accompanying graph. □

From Propositions 2 and 4 (or 1 and 3) we obtain the following result:

Corollary 1. *$\blacklozenge_G^{\cup}$-consistency is strictly stronger than $\mathsf{N}\blacklozenge_G$-consistency.*

To complete our strength-based hierarchy we close off with some results that involve the non-singleton-style consistencies $\lozenge_G$-consistency and $\overleftarrow{\lozenge}_G$-consistency.

Proposition 6. *$\mathsf{N}\blacklozenge_G$-consistency is strictly stronger than $\lozenge_G$-consistency.*

Proof. Consider the QCN depicted in Figure 6. As noted in the caption of the figure, the QCN is $\lozenge_G$-consistent, but not $\mathsf{N}\blacklozenge_G$-consistent. Specifically, in order for the QCN to become $\mathsf{N}\blacklozenge_G$-consistent, the base relation eq needs to be removed from $C(x_2, x_3)$. Notably, applying $\mathsf{N}\blacklozenge_G$-consistency on that QCN makes it minimal. Additionally, by definition of $\mathsf{N}\blacklozenge_G$-consistency, we have that every $\mathsf{N}\blacklozenge_G$-consistent QCN is $\lozenge_G$-consistent. □

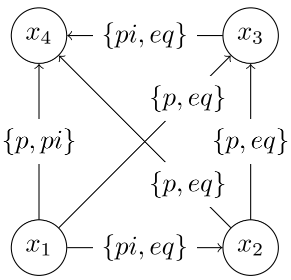


Figure 6: Given the QCN $\mathcal{N} = (V, C)$ above and the complete graph on V, we have that $\mathcal{N}$ is $\lozenge_G$-consistent, but not neighbourhood $\blacklozenge_G$-consistent

From Propositions 1, 2, 3, 4, and 6 we obtain the following result:

Corollary 2. *Each of the consistencies of* $\blacklozenge_G^{\cup}$*-consistency,* $\mathsf{N}\blacklozenge_G^{\cup}$*-consistency,* $\blacklozenge_G$*-consistency, and* $\mathsf{N}\blacklozenge_G$*-consistency is strictly stronger than* $\lozenge_G$*-consistency.*

From [53] we have the following result:

Proposition 7 ([53])**.** $\lozenge_G$*-consistency is strictly stronger than* $\overleftarrow{\lozenge}_G$*-consistency.*

From Corollary 2 and Proposition 7 we obtain the following last result with regard to our strength-based hierarchy:

Corollary 3. *Each of the consistencies of* $\blacklozenge_G^{\cup}$*-consistency,* $\mathsf{N}\blacklozenge_G^{\cup}$*-consistency,* $\blacklozenge_G$*-consistency,* $\mathsf{N}\blacklozenge_G$*-consistency, and* $\lozenge_G$*-consistency is strictly stronger than* $\overleftarrow{\lozenge}_G$*-consistency.*

A visual representation of the established strength-based hierarchy of consistencies is shown in Figure 7.

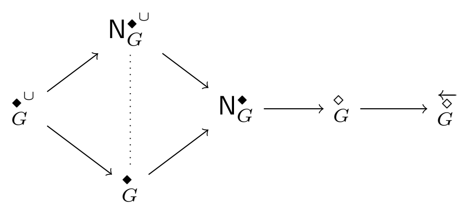


Figure 7: A strength-based hierarchy of consistencies for QCNs; an arrow denotes the (transitive) *strictly stronger* relationship and a dotted line the (symmetric) *incomparable* relationship

Finally, let the complete graph on a set of variables V be denoted by K_V, we have the following remark:

Remark. $\mathsf{N}\blacklozenge_{K_V}^{\cup}$*-consistency and* $\blacklozenge_{K_V}^{\cup}$*-consistency, respectively* $\mathsf{N}\blacklozenge_{K_V}$*-consistency and* $\blacklozenge_{K_V}$*-consistency, are equivalent.*

The above remark can facilitate the implementation of algorithms for applying the discussed neighbourhood singleton-style consistencies in the case where a complete graph is known to be used, as data structures and operations pertaining to neighbourhoods of relations need not be accounted for.

Algorithm 1: $\mathsf{PSWC}^{\cup}_{\mathsf{N}}(\mathcal{N}, G)$

in : A QCN $\mathcal{N} = (V, C)$, and a graph $G = (V' \subseteq V, E)$.
out : A sub-QCN of $\mathcal{N}$.

1 **begin**
2     $\mathcal{N} \leftarrow \diamond_G(\mathcal{N})$;
3     $Q \leftarrow list(E)$;
4   **while** $Q \neq \emptyset$ **do**
5         $\{v, v'\} \leftarrow Q.pop()$;
6         $(V, C') \leftarrow \bot^V$;
7   **foreach** $b \in C(v, v')$ **do**
8             $(V, C') \leftarrow (V, C') \cup \diamond_{G_{\mathsf{N}(vv')}}(\mathcal{N}_{[v,v']/\{b\}})$;
9   **if** $(V, C') \subset \mathcal{N}$ **then**
10             $flag \leftarrow$ False;
11   **foreach** $\{u, u'\} \in E$ **do**
12   **if** $C'(u, u') \subset C(u, u')$ **then**
13                     $C(u, u') \leftarrow C'(u, u')$;
14                     $C(u', u) \leftarrow C'(u', u)$;
15                     $flag \leftarrow$ True;
16   **if** $flag$ **then**
17                 $Q \leftarrow list(E)$
18   **return** $\mathcal{N}$;

Algorithms and Complexities

For the sake of completeness, we present in this section algorithms $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ and $\mathsf{PSWC}_{\mathsf{N}}$, shown in Algorithms 1 and 2 respectively, which given a QCN $\mathcal{N}$ and a graph G as input apply $\mathsf{N}^{\blacklozenge^{\cup}}_{G}$-consistency and $\mathsf{N}^{\blacklozenge}_{G}$-consistency on $\mathcal{N}$ respectively. By dropping the red underlined parts in the aforementioned algorithms, the reader can verify that they fall back to algorithms $\mathsf{PSWC}^{\cup}$ and PSWC respectively, which were introduced in [56]. As the latter algorithms have been proven to terminate and return $\blacklozenge^{\cup}_{G}(\mathcal{N})$ and $\blacklozenge_{G}(\mathcal{N})$ respectively given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, we can assert the following result:

Proposition 8. *Given a* QCN $\mathcal{N} = (V, C)$ *and a graph* $G = (V', E)$*, where* $V' \subseteq V$*, we have that algorithms* $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ *and* $\mathsf{PSWC}_{\mathsf{N}}$ *terminate and return* $\mathsf{N}^{\blacklozenge^{\cup}}_{G}(\mathcal{N})$ *and* $\mathsf{N}^{\blacklozenge}_{G}(\mathcal{N})$ *respectively.*

Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, the worst-case time complexity for both $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ and $\mathsf{PSWC}_{\mathsf{N}}$ is $O(\alpha|\mathsf{B}|^2|E|^2)$, where α is the worst-case time complexity for computing $\diamond_{G'}(\mathcal{N})$ with respect to the largest graph $G' \subseteq G$ that is used in Line 8 of the algorithms (as each constraint defines its own neighbourhood G'). For any given QCN $\mathcal{N} = (V, C)$ and a graph $G =$

Algorithm 2: $\mathsf{PSWC}_{\underline{\mathsf{N}}}(\mathcal{N}, G)$

in : A QCN $\mathcal{N} = (V, C)$, and a graph $G = (V' \subseteq V, E)$.
out : A sub-QCN of $\mathcal{N}$.

1 **begin**
2 $\quad \mathcal{N} \leftarrow \diamond_G(\mathcal{N})$;
3 $\quad Q \leftarrow list(E)$;
4 $\quad$ **while** $Q \neq \emptyset$ **do**
5 $\quad\quad \{v, v'\} \leftarrow Q.pop()$;
6 $\quad\quad (V, C') \leftarrow \bot^V$;
7 $\quad\quad$ **foreach** $b \in C(v, v')$ **do**
8 $\quad\quad\quad (V, C') \leftarrow (V, C') \cup \diamond_{G_{\underline{\mathsf{N}(vv')}}}(\mathcal{N}_{[v,v']/\{b\}})$;
9 $\quad\quad$ **if** $C'(v, v') \subset C(v, v')$ **then**
10 $\quad\quad\quad C(v, v') \leftarrow C'(v, v')$;
11 $\quad\quad\quad C(v', v) \leftarrow C'(v', v)$;
12 $\quad\quad\quad Q \leftarrow list(E)$;
13 $\quad$ **return** $\mathcal{N}$;

(V', E), where $V' \subseteq V$, we note that α is $O(\Delta|\mathsf{B}||E|)$, where Δ is the maximum vertex degree of G [31].

Finally, given a QCN $\mathcal{N}$ and a graph G, a parsimonious variant for approximating $\mathsf{N}^{\blacklozenge^{\cup}}_G$-consistency in $\mathcal{N}$ is algorithm $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, shown in Algorithm 3. Again, by dropping the red underlined parts in the aforementioned algorithm, the reader can verify that it falls back to a slight generalization of algorithm $\ell\mathsf{PSWC}^{\cup}$, which was introduced in [60]. Specifically, contrary to the algorithm as it appears in [60], in the input of Algorithm 3 we allow any subset S of the set of edges of the input graph to be used; this subset serves as the seed of constraints from which the singleton checks will start propagating themselves. Algorithm $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ is lazy in the sense that it relies upon previously revised constraints to allow itself to continue propagation. Therefore, depending on the subset S to be used, and the order in which the constraints are processed, the algorithm may produce different outputs for the same input (see [60]). However, we can still relate the output of $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ to the strength-based hierarchy of consistencies for QCNs presented earlier with the following result:

Proposition 9. *Given a QCN $\mathcal{N} = (V, C)$, a graph $G = (V', E)$, where $V' \subseteq V$, and a set S, where $S \subseteq E$, we have that $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ terminates and returns a sub-QCN $\mathcal{N}'$ of $\mathcal{N}$ such that $\mathsf{N}^{\blacklozenge^{\cup}}_G(\mathcal{N}) \subseteq \mathcal{N}' \subseteq \diamond_G(\mathcal{N})$ respectively.*

Proof. In line 2 of the algorithm, the original QCN $\mathcal{N}$ is made $\diamond_G$-consistent. First, we need to show that the rest of the refinement operations in the algorithm entail $\diamond_G$-consistency as well. As $\diamond_G$-consistency is closed under union (see [61] for more details), the QCN $\bigcup\{\diamond_{G_{\mathsf{N}(vv')}}(\mathcal{N}_{[v,v']/\{b\}}) \mid b \in C(v, v')\}$ that is constructed in lines 7–8 of the algorithm for each pair of variables $\{v, v'\} \in Q$, is $\diamond_G$-consistent.

Algorithm 3: $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}(\mathcal{N}, G, S)$

```
in   : A QCN 𝒩 = (V, C), a graph G = (V′ ⊆ V, E), and a set S ⊆ E.
out  : A sub-QCN of 𝒩.
begin
    𝒩 ← ◇_G(𝒩);
    Q ← list(S);
    while Q ≠ ∅ do
        {v, v′} ← Q.pop();
        (V, C′) ← ⊥^V;
        foreach b ∈ C(v, v′) do
            (V, C′) ← (V, C′) ∪ ◇_{G_N(vv′)}(𝒩_[v,v′]/{b});
        C(v, v′) ← C′(v, v′);
        if (V, C′) ⊂ 𝒩 then
            foreach {u, u′} ∈ E \ {v, v′} do
                if C′(u, u′) ⊂ C(u, u′) then
                    C(u, u′) ← C′(u, u′);
                    C(u′, u) ← C′(u′, u);
                    Q.push({u, u′});
    return 𝒩;
```

Further, since $\diamond_G(\mathcal{N})$ is the unique $\subseteq$-maximal $\diamond_G$-consistent sub-QCN of $\mathcal{N}$, it follows that $\mathcal{N}' \subseteq \diamond_G(\mathcal{N})$. Finally, the fact that $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ terminates and returns a sub-QCN $\mathcal{N}'$ of $\mathcal{N}$ such that $\mathsf{N}^{\blacklozenge\cup}_{G}(\mathcal{N}) \subseteq \mathcal{N}'$ follows directly from the structure of algorithm $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, which considers a subset of the set of neighbourhood-restricted collective singleton checks that is performed by $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$. □

The worst-case time complexity of $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ is the same as that of $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ (and $\mathsf{PSWC}_{\mathsf{N}}$), although we will see later on in Section 5 that it can be much faster in practice.

5. Experimental Evaluation

In this section we investigate the utility of the proposed neighbourhood singleton-style consistency algorithms, as well as the discussed state-of-the-art local consistency algorithms that appear in the literature, with respect to the fundamental reasoning problems of *satisfiability checking* and *minimal labeling* that are associated with QCNs. Specifically, we explore their effectiveness and efficiency in determining the satisfiability of a given network instance and in discovering the unfeasible base relations of that network instance (in regard to both CPU time and correctness of decision).

Technical specifications. The evaluation was carried out on a computer with an Intel Core i5-4570 processor, 16 GB of RAM, and the Xenial Xerus x86_64 OS (Ubuntu Linux). All algorithms were coded in Python and run using the PyPy interpreter under version 5.1.2, which implements Python 2.7.10. Only one CPU core was used.

Dataset. We used the dataset employed in [61]. That dataset comprises 1 000 random and structured QCNs of IA that were created using models $\mathsf{A(n,l,d)}$ [62] and $\mathsf{BA(n,m)}$ [63] respectively. Pertaining to $\mathsf{A(n,l,d)}$, there are 100 QCNs of IA of $\mathsf{n} = 70$ variables and with $\mathsf{l} = 6.5$ base relations per non-universal constraint on average, for all values of the average constraint graph degree d from 7 to 12 with a step of 1. Pertaining to $\mathsf{BA(n,m)}$, there are 100 QCNs of IA of $\mathsf{n} = 150$ variables for all values of the constraint graph *preferential attachment* m [64] from 2 to 5 with a step of 1. Finally, regarding the graphs that were given as input to our algorithms, the *maximum cardinality search* algorithm [65] was used to obtain triangulations of the constraint graphs of our QCNs. The choice of such chordal graphs was not only reasonable but also crucial given their important theoretical and practical implications in qualitative constraint-based spatial and temporal reasoning, as reviewed in [66]; the use of those graphs itself was inspired by [67, 68, 69] among other works, where preliminary results pertaining to tree decompositions were established.

Tools. In addition to our implementations of the algorithms that were presented in Section 4, we utilized the following four software tools:[2]

- Solver, the state-of-the-art reasoner for checking the satisfiability of QCNs of Interval Algebra and RCC8 that was introduced in [63] (and in particular the reasoner called Phalanx$\bigtriangledown$ in that work);
- Minimizer, our own implementation of the approach for solving the minimal labeling problem of QCNs of Interval Algebra and RCC8 that was first presented in [30];[3]
- PWC, the state-of-the-art algorithm for applying $\overset{\diamond}{_G}$-consistency on QCNs of Interval Algebra and RCC8 that was used in [63] (which is a module of the Phalanx$\bigtriangledown$ reasoner mentioned earlier);
- DPWC, the state-of-the-art algorithm for applying $\overset{\leftarrow}{\overset{\diamond}{_G}}$-consistency on QCNs of Interval Algebra and RCC8 that was introduced in [53] (and in particular the reasoner called Pyrrhus in that work).

[2] These software tools are available at https://msioutis.gitlab.io/software.

[3] In particular, we ported the code to Python and included all recent advances that are associated with the components that comprise that approach, such as improvements in its underlying satisfiability checking module. It must also be noted that the strongest of the local consistencies discussed here, viz., $\overset{\blacklozenge^{\cup}}{_G}$-consistency, was used as a preprocessing step to enhance the performance of Minimizer.

Table 1: Evaluation with random IA networks that were generated using model $\mathsf{A}(\mathsf{n}=70,\mathsf{l}=6.5,\mathsf{d})$ [62]

d	Solver	Minimizer	$\mathsf{PSWC}^{\cup}$	$\mathsf{PSWC}^{\cup}_{\mathsf{N}}$	$\ell\mathsf{PSWC}^{\cup}$	$\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$	PSWC	$\mathsf{PSWC}_{\mathsf{N}}$	PWC	DPWC
7	$\frac{0.16s}{2}$	$\frac{12.29s}{3.84\%}$	$\frac{2.54s}{2\|3.84\%}$	$\frac{2.27s}{2\|3.84\%}$	$\frac{0.44s}{2\|3.84\%}$	$\frac{0.40s}{2\|3.84\%}$	$\frac{3.00s}{2\|3.84\%}$	$\frac{2.72s}{2\|3.84\%}$	$\frac{0.00s}{2\|3.77\%}$	$\frac{0.00s}{2\|2.48\%}$
8	$\frac{0.17s}{5}$	$\frac{27.40s}{8.75\%}$	$\frac{9.92s}{5\|8.72\%}$	$\frac{7.80s}{5\|8.68\%}$	$\frac{1.96s}{5\|8.69\%}$	$\frac{1.58s}{5\|8.66\%}$	$\frac{11.22s}{5\|8.72\%}$	$\frac{9.64s}{5\|8.64\%}$	$\frac{0.00s}{4\|7.23\%}$	$\frac{0.00s}{3\|3.70\%}$
9	$\frac{0.29s}{6}$	$\frac{281.59s}{13.67\%}$	$\frac{24.80s}{6\|12.31\%}$	$\frac{17.47s}{6\|11.69\%}$	$\frac{4.69s}{6\|12.03\%}$	$\frac{3.41s}{6\|11.44\%}$	$\frac{28.23s}{6\|12.31\%}$	$\frac{20.96s}{6\|11.36\%}$	$\frac{0.01s}{1\|4.82\%}$	$\frac{0.00s}{0\|0.63\%}$
10	$\frac{1.77s}{55}$	$\frac{1\,541.88s}{70.57\%}$	$\frac{41.13s}{54\|64.13\%}$	$\frac{31.15s}{51\|57.06\%}$	$\frac{7.71s}{50\|57.82\%}$	$\frac{5.46s}{44\|49.93\%}$	$\frac{48.04s}{54\|64.11\%}$	$\frac{36.58s}{51\|56.45\%}$	$\frac{0.01s}{5\|10.02\%}$	$\frac{0.00s}{1\|1.92\%}$
11	$\frac{4.52s}{100}$	$\frac{5.99s}{100\%}$	$\frac{6.98s}{99\|98.97\%}$	$\frac{8.23s}{97\|97.06\%}$	$\frac{2.52s}{96\|96.13\%}$	$\frac{2.57s}{91\|91.40\%}$	$\frac{8.44s}{99\|98.97\%}$	$\frac{10.83s}{97\|97.06\%}$	$\frac{0.01s}{31\|34.93\%}$	$\frac{0.00s}{5\|6.07\%}$
12	$\frac{0.23s}{100}$	$\frac{0.96s}{100\%}$	$\frac{0.59s}{100\|100\%}$	$\frac{0.96s}{100\|100\%}$	$\frac{0.58s}{100\|100\%}$	$\frac{0.79s}{100\|100\%}$	$\frac{0.70s}{100\|100\%}$	$\frac{1.51s}{100\|100\%}$	$\frac{0.01s}{44\|47.91\%}$	$\frac{0.00s}{4\|5.09\%}$

Table 2: Evaluation with structured IA networks that were generated using model $\mathsf{BA}(\mathsf{n} = 150, \mathsf{m})$ [63]

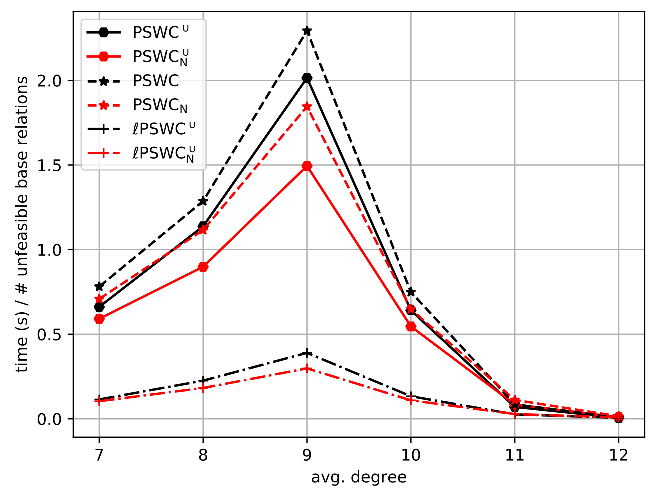
m	Solver	Minimizer	$\mathsf{PSWC}^{\cup}$	$\mathsf{PSWC}^{\cup}_{\mathsf{N}}$	$\ell\mathsf{PSWC}^{\cup}$	$\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$	PSWC	$\mathsf{PSWC}_{\mathsf{N}}$	PWC	DPWC
2	$\frac{0.15s}{2}$	$\frac{6.57s}{3.14\%}$	$\frac{0.53s}{2\mid 3.14\%}$	$\frac{0.45s}{2\mid 3.14\%}$	$\frac{0.12s}{2\mid 3.14\%}$	$\frac{0.10s}{2\mid 3.14\%}$	$\frac{0.67s}{2\mid 3.14\%}$	$\frac{0.56s}{2\mid 3.14\%}$	$\frac{0.00s}{2\mid 3.12\%}$	$\frac{0.00s}{2\mid 2.26\%}$
3	$\frac{0.17s}{7}$	$\frac{34.95s}{9.42\%}$	$\frac{9.55s}{7\mid 9.42\%}$	$\frac{7.85s}{7\mid 9.40\%}$	$\frac{2.52s}{7\mid 9.41\%}$	$\frac{1.91s}{7\mid 9.40\%}$	$\frac{12.40s}{7\mid 9.42\%}$	$\frac{10.14s}{7\mid 9.38\%}$	$\frac{0.01s}{7\mid 8.82\%}$	$\frac{0.00s}{4\mid 3.92\%}$
4	$\frac{0.23s}{60}$	$\frac{101.33s}{66.89\%}$	$\frac{95.64s}{60\mid 66.83\%}$	$\frac{69.81s}{60\mid 66.39\%}$	$\frac{26.06s}{60\mid 66.64\%}$	$\frac{19.39s}{60\mid 66.24\%}$	$\frac{126.69s}{60\mid 66.83\%}$	$\frac{94.51s}{60\mid 65.77\%}$	$\frac{0.01s}{42\mid 44.61\%}$	$\frac{0.00s}{12\mid 12.06\%}$
5	$\frac{0.16s}{100}$	$\frac{0.71s}{100\%}$	$\frac{0.04s}{100\mid 100\%}$	$\frac{0.07s}{100\mid 100\%}$	$\frac{0.06s}{100\mid 100\%}$	$\frac{0.06s}{100\mid 100\%}$	$\frac{0.05s}{100\mid 100\%}$	$\frac{0.07s}{100\mid 100\%}$	$\frac{0.01s}{92\mid 92.32\%}$	$\frac{0.00s}{29\mid 28.91\%}$

Results. The results of our experimental evaluation are detailed in Tables 1 and 2, where a fraction $\frac{x}{y}$ for Solver denotes that it required x seconds of CPU time on average per dataset of network instances during its operation and detected y such instances as being unsatisfiable in total, a fraction $\frac{x}{z}$ for $\mathsf{Minimizer}$ denotes that it determined $z\%$ of base relations to be unfeasible in total, and a fraction $\frac{x}{y|z}$ for the rest of the algorithms denotes all the previous information combined together (from the viewpoint of the respective algorithm).

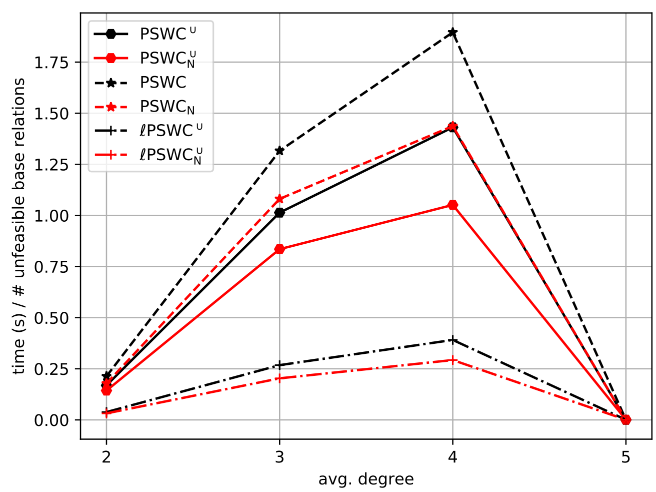
The first thing to note is that Solver has no competition whatsoever in terms of deciding the satisfiability of a network instance. This was expected, as this type of reasoner is tailored to avoid "bad" branches in the search tree and to reach a leaf (i.e., a solution) in the most efficient way possible. On the other hand, when the entire search tree needs to be taken into account, as is the case with $\mathsf{Minimizer}$, the computational task is much more time-consuming; therefore, $\mathsf{Minimizer}$ has by far the worst performance among its competitors.

Regarding the singleton-style consistency algorithms, we can note that they of course have an overhead compared to Solver, but they are much faster in general than $\mathsf{Minimizer}$ and they can, in many cases, simulate its pruning capability in an almost exact manner. It is worth mentioning that the neighbourhood-focused singleton-style algorithms $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, $\mathsf{PSWC}_{\mathsf{N}}$, and $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ are around 30% faster in the phase transition region than the standard algorithms $\mathsf{PSWC}^{\cup}$, PSWC, and $\ell\mathsf{PSWC}^{\cup}$ respectively, whilst retaining much of the good performance characteristics (viz., unfeasible base relations elimination and satisfiability decision) of the latter respectively. The parsimonious variants $\ell\mathsf{PSWC}^{\cup}$ and $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ are up to 6 times faster in the phase transition region than $\mathsf{PSWC}^{\cup}$ and $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ respectively, but detect in general slightly fewer unsatisfiable network instances and eliminate slightly fewer unfeasible base relations respectively as well. We should note that for a given QCN $\mathcal{N} = (V, C)$ and a graph $G = (V', E)$, where $V' \subseteq V$, the subset S that was used as input for the parsimonious variants (see Algorithm 3) corresponds to the set of edges $E(\mathsf{G}(\diamond_G(\mathcal{N})))$, i.e., the set of edges of the constraint graph of $\diamond_G(\mathcal{N})$.

Finally, in order to better assess how the different singleton-style consistency algorithms compare with one another, Figure 8 visualizes the efficiency to effectiveness ratios of those algorithms for the datasets considered here. In particular, the efficiency to effectiveness ratio of a singleton-style algorithm is the ratio $\frac{x}{z}$, where, as a reminder, x denotes the seconds of CPU time that were required on average per dataset of network instances during the algorithm's operation, and z denotes the % of base relations in such instances that were detected as being unfeasible in total. Clearly, the smaller the efficiency to effectiveness ratio is, the better it is, as ideally the CPU time should be kept small and the number of unfeasible base relations high. The discussion here ties in with the remark in the introduction about the *sweet spot* between effectiveness and efficiency that can be uncovered using neighbourhood-based restrictions in singleton-style consistencies. It is critical to investigate whether such ratios



(a) Efficiency to effectiveness ratio pertaining to results of Table 1



(b) Efficiency to effectiveness ratio pertaining to results of Table 2

Figure 8: Efficiency to effectiveness ratios of singleton-style consistency algorithms

are at all improved by such restrictions, and if so, by how much. As it can be observed in the graphs of Figure 8, it appears to be well worth investing in neighbourhood-based restrictions, since they improve the efficiency to effectiveness ratios of the involved standard algorithms by up to around 25% with respect to both datasets. Taking additionally into account the fact that the effectiveness of neighbourhood-focused singleton-style algorithms can only decrease with regard to the respective standard variants (but the same is not true for their efficiency in general), viewing more closely the ratios in Figure 8 suggests that neighbourhood-focused singleton-style algorithms *consistently* gain much more in efficiency than they lose in effectiveness with regard to the respective standard variants.

Synopsis. In conclusion, and with respect to the datasets involved here, we observe that the considered singleton-style consistency algorithms are not good options for just checking the satisfiability of a network instance, as they present an overhead when compared to a state-of-the-art reasoner that is tailored to this specific task. However, we also point out that they are ideal candidates for efficiently approximating and even determining in many cases the minimal labeling of a network instance; this becomes even more prominent if one considers the comparatively bad pruning capability of PWC, and the even worse one of DPWC for that matter. It should be noted that even if the state-of-the-art reasoner $\mathsf{Minimizer}$ is provided with a minimal network instance (as it was usually the case in our evaluation due to the preprocessing with $\blacklozenge^{\cup}_{G}$-consistency, see again Footnote 3 about this), it is an NP-hard problem to decide the satisfiability of that instance, and an NP-hard problem to verify its minimality as a consequence [29]. We emphasize again the fact that the neighbourhood-focused singleton-style algorithms $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, $\mathsf{PSWC}_{\mathsf{N}}$, and $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ were found to be around 30% faster in the phase transition region than the standard algorithms $\mathsf{PSWC}^{\cup}$, PSWC, and $\ell\mathsf{PSWC}^{\cup}$ respectively, for both random and structured QCNs, whilst they were able to retain much of the good performance characteristics in terms of unfeasible base relations elimination and satisfiability decision of the latter respectively. Regarding the parsimonious variants in particular, viz., $\ell\mathsf{PSWC}^{\cup}$ and $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, even though they exhibited arguably impressive performance characteristics, a major disadvantage is that they do not yield unique closures for a same QCN (see again the discussion in the previous section), which inhibits their theoretical study. Our efficiency to effectiveness ratio analysis revealed that it is well worth investing in neighbourhood-based restrictions, since they were found to improve the efficiency to effectiveness ratios of the involved standard algorithms by up to around 25% with respect to both datasets.

6. Related Works and Discussion

Singleton-based consistencies belong to the class of strong filtering techniques for both qualitative and traditional constraint-based reasoning. They have been shown to drastically reduce the search space and, thus, improve the performance of solvers for many difficult instances. However, they can suffer from a serious drawback; they are in general too expensive when applied exhaustively during the whole search. For this reason, several researchers worked on proposing either weaker variants of the classic *Singleton Arc Consistency* (SAC) [55, 70] or approximation techniques (i.e., techniques that do not reach a fixed point). Regarding the former, several weaker consistencies than SAC

have been proposed, with *Neighbourhood Singleton Arc Consistency* (NSAC) being the main representative [71]. In NSAC, the AC-based singleton checks are applied only on the sub-graph that corresponds to the neighbourhood of the variable that is being considered, instead of the full graph. This restricted form of SAC, is nearly as effective as SAC in terms of pruning, whilst requiring much less time. NSAC can be seen as a family of consistencies, since it can be generalized by a parameter k (k-NSAC) that fixes the distance from a singleton-checked variable [33]. When $k = 1$, then 1-NSAC is simply referred to as NSAC, and for $k = n$ it is the case that NSAC becomes SAC. Both weaker and stronger consistencies than NSAC have been proposed, for example, by restricting AC to a one pass application on the neighbourhood of a considered variable during a singleton check [59], or by replacing AC with a stronger consistency [72]. In this work, the presented consistencies of $\mathsf{N}^{\blacklozenge^{\cup}}_{G}$-consistency and $\mathsf{N}^{\blacklozenge}_{G}$-consistency can be seen as adaptations of NSAC to handle infinite domains and qualitative relations. Regarding $\mathsf{N}^{\blacklozenge^{\cup}}_{G}$-consistency in particular, it is an even closer adaptation for QCNs of neighbourhood-focused 1-*partitioning consistency* (POAC) [58] for CSPs, with POAC being a stronger variant of SAC (cf. [73]). Indeed, as the variables in QCNs contain infinite values, singleton checks involve the base relations that make up a (qualitative) constraint instead. Even though we currently do not parameterize on the distance from a singleton-checked constraint, i.e., on how far its neighbourhood extends away from it, we do parameterize on the graph G that is used for enforcing a given consistency. As noted in Section 5, the choice of G can have important theoretical and practical implications in qualitative constraint-based spatial and temporal reasoning [66]. Furthermore, AC always holds in a given QCN by the very definition of the latter (see also the discussion about base relations in the beginning of Section 2), and hence we typically utilize a stronger base consistency, namely, $\diamond_{G}$-consistency.

Regarding the approximation techniques of singleton-based consistencies, adaptive variants of POAC have been proposed recently [74]; in short, adaptive POAC, referred to as APOAC, is not needed to run until having proved its theoretical fixed point. Balafrej et al. in [74] propose to limit and adapt the number of times that variables are singleton checked, by measuring, during search, the stagnation in the amount of pruned values. The experiments have shown that APOAC can obtain significant speed-ups over SAC and (full) POAC. The bulk of works alternate between two or several levels of consistency to avoid the prohibitive cost of applying a strong consistency either on the entire network as a preprocessing step [75, 76, 77] or along search [78]. The parsimonious approach that we presented here, namely, $\ell\mathsf{PSWC}^{\cup}_{\mathsf{N}}$, is closer to the approach of [75], where a strong consistency is applied only on the constraints that caused a failure during search. Similarly, our method is based on observing where fruitful pruning takes place (during one call of the algorithm), thus revising only constraints whose relations were previously reduced via the elimination of unfeasible base relations.

Arguably, qualitative spatial and temporal reasoning is an area where similar consistencies play a major role in both efficiently solving existing problems and opening new directions by allowing harder problems to be defined and tackled.

7. Conclusion and Future Work

We proposed singleton-style consistencies for QCNs that are applied just on the neighbourhood of a singleton-checked constraint instead of the whole network, and attained a strength-based hierarchy among all discussed consistencies here. Further, we proposed algorithms to enforce our consistencies, as well as parsimonious variants thereof, that were shown to be much more efficient in practice than the state-of-the-art algorithms for a dataset comprising random and structured QCNs of Interval Algebra. It should be noted that our approach is generic and applies to other calculi as well, such as the spatial calculus RCC8.

Future work consists in obtaining structure-based tractability results focused on the neighbourhood of constraints, developing faster inference mechanisms that will only partially singleton-check a constraint (i.e., only some of the base relations of a constraint will be used for singleton checks), much like *quick shaving* [79], establishing adaptive constraint propagators for QCNs (see [74] for instance in the context of CSPs), and looking into prioritizing or even solely focusing on singleton checks for base relations that play a crucial role in the computational properties of a given qualitative constraint language [80, 81]. Therefore, we argue that our approach can drive both theoretical and practical future research and provide a foundation for further work in the study of QCNs, which are pertinent in Symbolic AI [2].

Funding: This work was supported by the Academy of Finland [grant numbers 251170, 327352].

References

[1] G. Ligozat, Qualitative Spatial and Temporal Reasoning, Wiley, 2013.

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

[3] M. Bhatt, J. O. Wallgrün, Geospatial Narratives and Their Spatio-Temporal Dynamics: Commonsense Reasoning for High-Level Analyses in Geographic Information Systems, ISPRS Int. J. Geo-Information 3 (2014) 166–205.

[4] F. Dylla, J. O. Wallgrün, Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control, J. Intell. Robotic Syst. 48 (2007) 55–78.

[5] N. Krishnaswamy, S. Friedman, J. Pustejovsky, Combining Deep Learning and Qualitative Spatial Reasoning to Learn Complex Structures from Sparse Examples with Noise, in: AAAI, 2019.

[6] P. A. Story, M. F. Worboys, A Design Support Environment for Spatio-Temporal Database Applications, in: COSIT, 1995.

[7] K. S. R. Dubba, A. G. Cohn, D. C. Hogg, M. Bhatt, F. Dylla, Learning Relational Event Models from Video, J. Artif. Intell. Res. 53 (2015) 41–90.

[8] M. Bhatt, F. Dylla, J. Hois, Spatio-terminological Inference for the Design of Ambient Environments, in: COSIT, 2009.

[9] M. Sioutis, M. Alirezaie, J. Renoux, A. Loutfi, Towards a Synergy of Qualitative Spatio-Temporal Reasoning and Smart Environments for Assisting the Elderly at Home, in: IJCAI Workshop on Qualitative Reasoning, 2017.

[10] J. Suchan, M. Bhatt, P. A. Walega, C. P. L. Schultz, Visual Explanation by High-Level Abduction: On Answer-Set Programming Driven Reasoning About Moving Objects, in: AAAI, 2018.

[11] J. Suchan, M. Bhatt, S. Varadarajan, Out of Sight But Not Out of Mind: An Answer Set Programming Based Online Abduction Framework for Visual Sensemaking in Autonomous Driving, in: IJCAI, 2019.

[12] J. Suchan, M. Bhatt, Semantic Question-Answering with Video and Eye-Tracking Data: AI Foundations for Human Visual Perception Driven Cognitive Film Studies, in: IJCAI, 2016.

[13] Z. Cui, A. G. Cohn, D. A. Randell, Qualitative Simulation Based on a Logical Formalism of Space and Time, in: AAAI, 1992.

[14] R. Moskovitch, Y. Shahar, Classification of multivariate time series via temporal abstraction and time intervals mining, Knowl. Inf. Syst. 45 (2015) 35–74.

[15] O. Kostakis, P. Papapetrou, On searching and indexing sequences of temporal intervals, Data Min. Knowl. Discov. 31 (2017) 809–850.

[16] O. Kostakis, N. Tatti, A. Gionis, Discovering recurring activity in temporal networks, Data Min. Knowl. Discov. 31 (2017) 1840–1871.

[17] A. Morales, I. Navarrete, G. Sciavicco, A new modal logic for reasoning about space: spatial propositional neighborhood logic, Ann. Math. Artif. Intell. 51 (2007) 1–25.

[18] E. Muñoz-Velasco, M. Pelegrín-García, P. Sala, G. Sciavicco, I. E. Stan, On coarser interval temporal logics, Artif. Intell. 266 (2019) 1–26.

[19] D. Bresolin, D. D. Monica, A. Montanari, P. Sala, G. Sciavicco, Decidability and complexity of the fragments of the modal logic of Allen's relations over the rationals, Inf. Comput. 266 (2019) 97–125.

[20] S. Hazarika, Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions, IGI Global, 2012.

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

[22] J. Renz, B. Nebel, Qualitative Spatial Reasoning Using Constraint Calculi, in: Handbook of Spatial Logics, 2007, pp. 161–215.

[23] F. Dylla, T. Mossakowski, T. Schneider, D. Wolter, Algebraic Properties of Qualitative Spatio-Temporal Calculi, in: COSIT, 2013.

[24] M. C. Golumbic, R. Shamir, Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach, J. ACM 40 (1993) 1108–1133.

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

[26] G. Gottlob, On minimal constraint networks, Artif. Intell. 191-192 (2012) 42–60.

[27] P. van Beek, R. Dechter, On the Minimality and Decomposability of Row-Convex Constraint Networks, J. ACM 42 (1995) 543–561.

[28] A. Gerevini, A. Saetti, Computing the minimal relations in point-based qualitative temporal reasoning through metagraph closure, Artif. Intell. 175 (2011) 556–585.

[29] W. Liu, S. Li, Solving Minimal Constraint Networks in Qualitative Spatial and Temporal Reasoning, in: CP, 2012.

[30] N. Amaneddine, J.-F. Condotta, M. Sioutis, Efficient Approach to Solve the Minimal Labeling Problem of Temporal and Spatial Qualitative Constraints, in: IJCAI, 2013.

[31] A. Chmeiss, J. Condotta, Consistency of Triangulated Temporal Qualitative Constraint Networks, in: ICTAI, 2011.

[32] J. Renz, G. Ligozat, Weak Composition for Qualitative Spatial and Temporal Reasoning, in: CP, 2005.

[33] R. J. Wallace, Neighbourhood SAC: Extensions and new algorithms, AI Commun. 29 (2016) 249–268.

[34] G. Ligozat, J. Renz, What Is a Qualitative Calculus? A General Framework, in: PRICAI, 2004.

[35] J. F. Allen, Maintaining Knowledge about Temporal Intervals, Commun. ACM 26 (1983) 832–843.

[36] J. F. Allen, J. A. G. M. Koomen, Planning Using a Temporal World Model, in: IJCAI, 1983.

[37] J. F. Allen, Planning as Temporal Reasoning, in: KR, 1991.

[38] R. N. Pelavin, J. F. Allen, A Model for Concurrent Actions Having Temporal Extent, in: AAAI, 1987.

[39] J. Dorn, Dependable Reactive Event-Oriented Planning, Data Knowl. Eng. 16 (1995) 27–49.

[40] L. Mudrová, N. Hawes, Task scheduling for mobile robots using interval algebra, in: ICRA, 2015.

[41] F. Song, R. Cohen, The Interpretation of Temporal Relations in Narrative, in: IJCAI, 1988.

[42] P. Denis, P. Muller, Predicting Globally-Coherent Temporal Structures from Texts via Endpoint Inference and Graph Decomposition, in: IJCAI, 2011.

[43] R. T. Snodgrass, The Temporal Query Language TQuel, ACM Trans. Database Syst. 12 (1987) 247–298.

[44] C. X. Chen, C. Zaniolo, Universal Temporal Data Languages, in: DDLP, 1998.

[45] T. D. C. Little, A. Ghafoor, Interval-Based Conceptual Models for Time-Dependent Multimedia Data, IEEE Trans. Knowl. Data Eng. 5 (1993) 551–563.

[46] S. Benzer, On the Topology of the Genetic Fine Structure, Proc. Natl. Acad. Sci. USA 45 (1959) 1607–1620.

[47] R. Lu, S. W. Sadiq, V. Padmanabhan, G. Governatori, Using a temporal constraint network for business process execution, in: ADC, 2006.

[48] J. Gamper, W. Nejdl, Abstract temporal diagnosis in medical domains, Artificial Intelligence in Medicine 10 (1997) 209–234.

[49] D. A. Randell, Z. Cui, A. Cohn, A Spatial Logic Based on Regions & Connection, in: KR, 1992.

[50] M. Sioutis, Z. Long, S. Li, Efficiently Reasoning about Qualitative Constraints through Variable Elimination, in: SETN, 2016.

[51] R. Dechter, J. Pearl, Network-Based Heuristics for Constraint-Satisfaction Problems, Artif. Intell. 34 (1987) 1–38.

[52] R. Dechter, I. Meiri, J. Pearl, Temporal Constraint Networks, Artif. Intell. 49 (1991) 61–95.

[53] M. Sioutis, Z. Long, S. Li, Leveraging Variable Elimination for Efficiently Reasoning about Qualitative Constraints, Int. J. Artif. Intell. Tools 27 (2018) 1860001.

[54] J.-F. Condotta, C. Lecoutre, A Class of $\diamond_f$-Consistencies for Qualitative Constraint Networks, in: KR, 2010.

[55] R. Debruyne, C. Bessière, Some Practicable Filtering Techniques for the Constraint Satisfaction Problem, in: IJCAI, 1997.

[56] M. Sioutis, A. Paparrizou, J. Condotta, Collective Singleton-Based Consistency for Qualitative Constraint Networks, in: TIME, 2017.

[57] M. Bodirsky, S. Wölfl, RCC8 is polynomial on networks of bounded treewidth, in: IJCAI, 2011.

[58] H. Bennaceur, M. Affane, Partition-k-AC: An Efficient Filtering Technique Combining Domain Partition and Arc Consistency, in: CP, 2001.

[59] A. Paparrizou, K. Stergiou, On Neighborhood Singleton Consistencies, in: IJCAI, 2017.

[60] M. Sioutis, A. Paparrizou, J. Condotta, A Lazy Algorithm to Efficiently Approximate Singleton Path Consistency for Qualitative Constraint Networks, in: ICTAI, 2017.

[61] M. Sioutis, A. Paparrizou, J. Condotta, Collective Singleton-Based Consistency for Qualitative Constraint Networks: Theory and Practice, Theor. Comput. Sci 797 (2019) 17–41.

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

[63] M. Sioutis, J. Condotta, M. Koubarakis, An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks, Int. J. Artif. Intell. Tools 25 (2016) 1–33.

[64] A.-L. Barabási, R. Albert, Emergence of Scaling in Random Networks, Science 286 (1999) 509–512.

[65] R. E. Tarjan, M. Yannakakis, 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.

[66] M. Sioutis, Y. Salhi, J.-F. Condotta, Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning, Knowl. Eng. Rev. 32 (2016) e4.

[67] J. Huang, J. J. Li, J. Renz, Decomposition and tractability in qualitative spatial and temporal reasoning, Artif. Intell. 195 (2013) 140–164.

[68] J. Huang, Compactness and its implications for qualitative spatial and temporal reasoning, in: KR, 2012.

[69] J. J. Li, J. Huang, J. Renz, A divide-and-conquer approach for solving interval algebra networks, in: IJCAI, 2009.

[70] C. Bessière, R. Debruyne, Theoretical analysis of singleton arc consistency and its extensions, Artif. Intell. 172 (2008) 29–41.

[71] R. J. Wallace, SAC and Neighbourhood SAC, AI Commun. 28 (2015) 345364.

[72] K. Stergiou, Neighborhood singleton consistencies, Constraints 24 (2019) 94–131.

[73] R. J. Woodward, B. Y. Choueiry, C. Bessiere, Cycle-Based Singleton Local Consistencies, in: AAAI, 2017.

[74] A. Balafrej, C. Bessiere, E. Bouyakhf, G. Trombettoni, Adaptive Singleton-Based Consistencies, in: AAAI, 2014.

[75] A. Paparrizou, K. Stergiou, Evaluating Simple Fully Automated Heuristics for Adaptive Constraint Propagation, in: ICTAI, 2012.

[76] A. Balafrej, C. Bessiere, R. Coletta, E.-H. Bouyakhf, Adaptive Parameterized Consistency, in: CP, 2013.

[77] R. J. Woodward, A. Schneider, B. Y. Choueiry, C. Bessiere, Adaptive Parameterized Consistency for Non-binary CSPs by Counting Supports, in: CP, 2014.

[78] A. Balafrej, C. Bessiere, A. Paparrizou, Multi-Armed Bandits for Adaptive Constraint Propagation, in: IJCAI, 2015.

[79] O. Lhomme, Quick Shaving, in: AAAI, 2005.

[80] P. Jonsson, V. Lagerkvist, Why are CSPs Based on Partition Schemes Computationally Hard?, in: MFCS, 2018.

[81] M. Bodirsky, P. Jonsson, B. Martin, A. Mottet, Classification Transfer for Qualitative Reasoning Problems, in: IJCAI, 2018.

Appendix A. Insight into Minimizer

In relation to Section 5, and Footnote 3 specifically, we provide insight into how one of the minimal labeling approximation consistencies, $\blacklozenge^{\cup}_{G}$-consistency, boosts the performance of Minimizer. We note that any of the singleton-style consistencies discussed here yields virtually indistinguishable results with respect to enhancing the performance of Minimizer, but we opt for $\blacklozenge^{\cup}_{G}$-consistency because it is the most effective one in characterizing unfeasible base relations, and because efficiency is not a factor in this setting (Minimizer is much slower than any of the related algorithms, see Tables 1 and 2).

Table A.3: Evaluation of Minimizer, where suffix $*$ suggests that $\blacklozenge_G^{\cup}$-consistency was *not* used as a preprocessing step and the input QCN was left untreated

(a) Evaluation with instances of Table 1

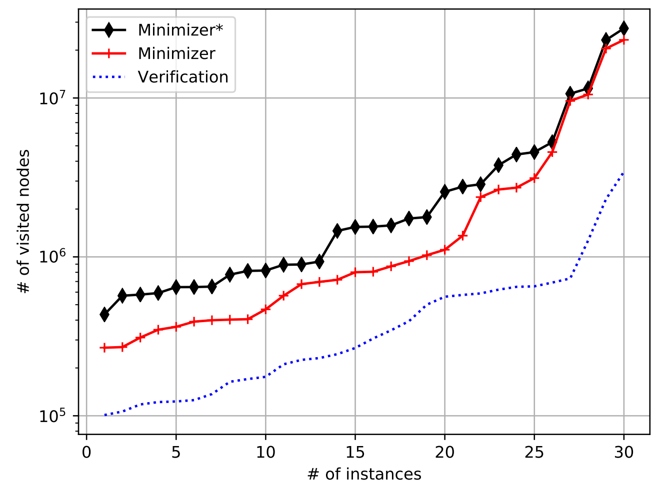
d	Minimizer$*$	Minimizer
7	$\frac{45.02k}{1.01}$	$\frac{44.90k}{1.00}$
8	$\frac{41.96k}{1.06}$	$\frac{41.64k}{1.05}$
9	$\frac{202.27k}{1.74}$	$\frac{155.68k}{1.67}$
10	$\frac{1\,096.83k}{1.99}$	$\frac{852.33k}{1.98}$
11	$\frac{1.94k}{2.00}$	$\frac{0.23k}{2.01}$
12	$\frac{0.05k}{2.07}$	$\frac{0.01k}{2.57}$

(b) Evaluation with instances of Table 2

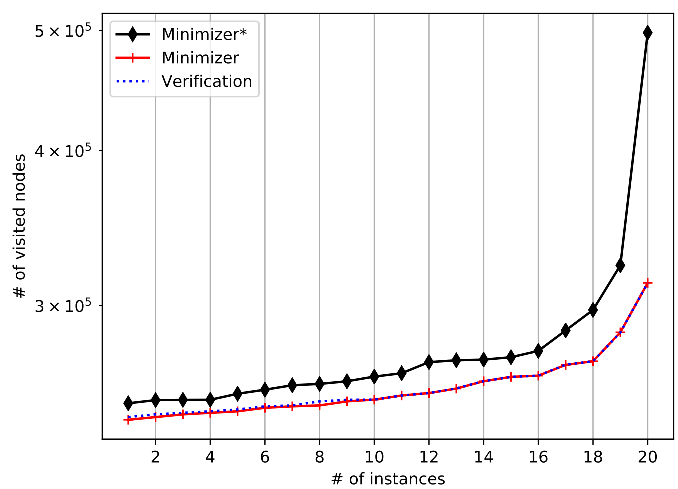
m	Minimizer$*$	Minimizer
2	$\frac{143.65k}{1.00}$	$\frac{143.64k}{1.00}$
3	$\frac{196.70k}{1.00}$	$\frac{197.18k}{1.00}$
4	$\frac{92.13k}{1.08}$	$\frac{85.29k}{1.02}$
5	$\frac{0.00k}{2.12}$	$\frac{0.00k}{3.23}$

The results of this evaluation are detailed in Table A.3, where a fraction $\frac{n}{b}$ denotes that the reasoner visited n nodes and produced a search tree with a branching factor of b on average, and in Figure A.9, where cactus plots on the most difficult instances are presented. The average CPU time is analogous to n, i.e., $x\%$ of less (or more respectively) visited nodes translates to roughly $x\%$ of less (or more respectively) CPU time. The results suggest that there is a boost of about 22% and 7% in the phase transition for instances of Table 1 ($\mathsf{d} = 10$) and Table 2 ($\mathsf{m} = 4$) respectively. These gains are maintained for the most difficult instances too, as it is demonstrated in Figure A.9; specifically, there is a gain of about 22% and 6% for the 5^{th} percentile of most difficult instances of Table 1 and Table 2 respectively (the distribution is heavy-tailed for both datasets). Finally, by viewing in particular the *verification* line in Figure A.9, we can see that gains are maxed out for instances of Table 2 (with respect to how Minimizer is in its current form), whereas there is still a lot of room for improvement for instances of Table 1. Such an improvement could be achieved via a tighter integration between one or more singleton-style consistencies and

Minimizer (e.g., during search), as the singleton-style consistency is currently only used as a preprocessing step and as a verification subroutine for the special case where it is complete for deciding minimality.



(a) Insight into the 5^{th} percentile of most difficult instances of Table 1



(b) Insight into the 5^{th} percentile of most difficult instances of Table 2

Figure A.9: Insight into the most difficult instances for Minimizer, where suffix $*$ suggests that $\blacklozenge^{\cup}_{G}$-consistency was *not* used as a preprocessing step and the input QCN was left untreated, and *verification* suggests that the input QCN was minimized beforehand (we remind the reader that verifying the minimality of a minimal QCN is already NP-hard [29])