On the Utility of Neighbourhood Singleton-style Consistencies for Qualitative Constraint-based Spatial and Temporal Reasoning

Michael Sioutis[1]
Department of Computer Science, Aalto University, Espoo, Finland
https://msioutis.gitlab.io/
michael.sioutis@aalto.fi

Anastasia Paparrizou
CRIL CNRS UMR 8188, Artois University, Lens, France
http://www.cril.univ-artois.fr/~paparrizou/
paparrizou@cril.fr

Tomi Janhunen
Department of Computer Science, Aalto University, Espoo, Finland
Tampere University, Tampere, Finland
https://users.ics.aalto.fi/ttj/
tomi.janhunen@aalto.fi

Abstract

A singleton-style consistency is a local consistency that verifies if each base relation (atom) of each constraint of a qualitative constraint network (QCN) can serve as a support with respect to the closure of that network under a (naturally) weaker local consistency. This local 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 Interval Algebra in the phase transition region to demonstrate the potential of our approach.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming; Computing methodologies → Temporal reasoning; Computing methodologies → Spatial and physical reasoning

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

Digital Object Identifier 10.4230/LIPIcs.TIME.2019.1

1 Introduction

Qualitative Spatial and Temporal Reasoning (QSTR) is a Symbolic AI approach that deals with the fundamental cognitive concepts of space and time in a qualitative, human-like, manner [28, 16]. 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. QSTR provides a concise framework that allows for rather inexpensive reasoning about entities located in

[1] Corresponding author.

26th International Symposium on Temporal Representation and Reasoning (TIME 2019).
Editors: Johann Gamper, Sophie Pinchinat, and Guido Sciavicco; Article No. 1; pp. 1:1–1:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 [7], cognitive robotics [18], deep learning [25], and qualitative model generation from video [15]. The interested reader may look into a more comprehensive review of the emerging applications, the trends, and the future directions of QSTR in [6].

The problem of representing and reasoning about qualitative spatial or temporal information can be modeled as a *qualitative constraint network* (QCN), i.e., a network of qualitative constraints corresponding to spatial or temporal relations between spatial or temporal variables respectively. Two fundamental reasoning problems associated with a given QCN $\mathcal{N}$ are the problems of *satisfiability checking* and *minimal labeling* (or *deductive closure*) [37]. In particular, the satisfiability checking problem is about deciding if there exists a valuation of the variables of $\mathcal{N}$ that satisfies its constraints, and the minimal labeling problem concerns finding the strongest implied constraints and consequently obtaining its minimal sub-network. In general, for most well-known spatio-temporal calculi the satisfiability checking problem is NP-hard [17]. Further, the minimal labeling problem is polynomial-time Turing reducible to the satisfiability checking problem [20].

Motivation

In this paper, we focus mostly on the minimal labeling problem, which, since its introduction in 1974 by Montanari [31], has been studied in the domain of both CSPs [21, 46] and QCNs [19, 30]. As noted in [21], *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 could be scheduled before a task B, or an object X could be placed on top of an object Y. Difficult problems such as the minimal labeling problem and alike are, in general, either approximated by the use of local consistencies [46] or even solved by the aid of such consistencies [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 [10, 35]). 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 [47], 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 this kind of network structures, which have received much attention over the past years [16].

Contributions

Our contributions are fourfold and are 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;

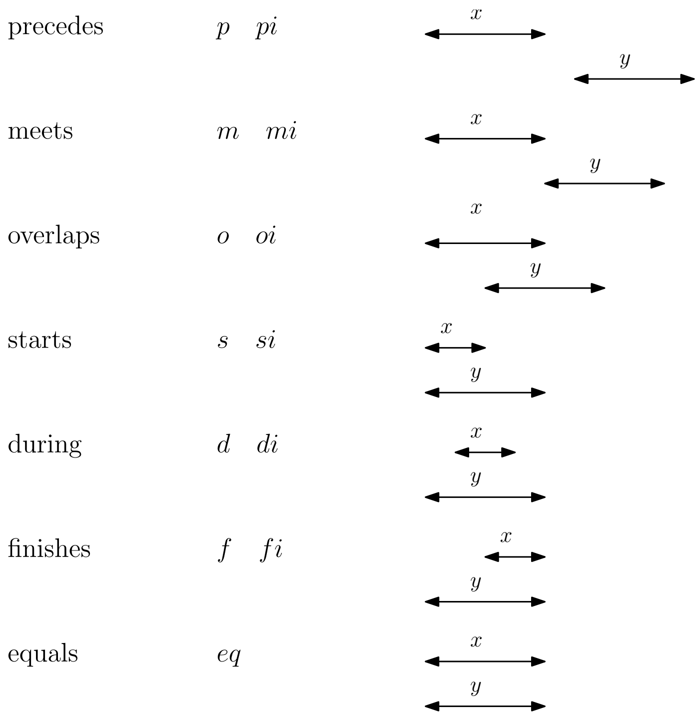


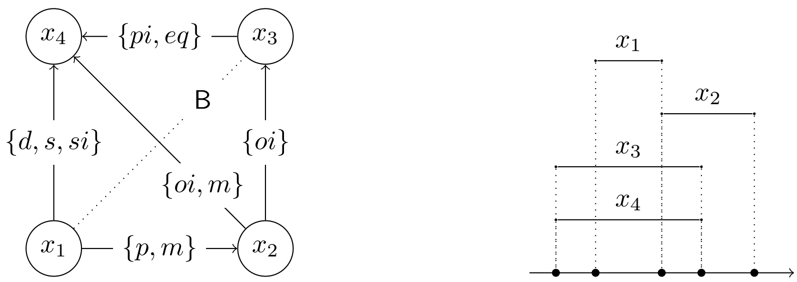
Figure 1 The base relations of IA; $\cdot i$ denotes the converse of $\cdot$

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.

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 $\sim 30\%$ faster in the phase transition region than the standard algorithms. Finally, in Section 6 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* [29], 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



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

Figure 2 Figurative examples of QCN terminology using IA

operations of union and intersection, the converse operation, and the *weak composition* operation denoted by the symbol $\diamond$ [29]. 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 $(x, z) \in b \wedge (z, y) \in b'\}$ is the (true) composition of b and b'. For all $r, r' \in 2^{\mathsf{B}}$, we have that $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r, b' \in r'\}$.

As an illustration, consider the well-known qualitative temporal constraint language of Interval Algebra (IA), introduced by Allen [1]. IA considers time intervals (as temporal entities) and the set of base relations $\mathsf{B} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$ to encode knowledge about the temporal relations between intervals on the timeline, as depicted in Figure 1. Specifically, each base relation represents a particular ordering of the four endpoints of two intervals on the timeline, and eq is the identity relation Id.

Notably, most of the well-known and well-studied qualitative constraint languages, such as Interval Algebra [1] and RCC8 [34], are in fact *relation algebras* [17]. In what follows, we restrict ourselves to such calculi in order to facilitate discussion of the consistencies and of the algorithms for enforcing them.

The problem of representing and reasoning about qualitative spatial or temporal information can be modeled as a *qualitative constraint network*, defined in the following manner:

▶ **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, each representing an entity of an infinite domain D;
- and C is a mapping $C : V \times V \to 2^{\mathsf{B}}$ such that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$ and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$, where $\bigcup \mathsf{B} = \mathsf{D} \times \mathsf{D}$.

An example of a QCN of IA is shown in Figure 2a; for clarity, converse relations as well as Id loops are not mentioned or shown in the figure.

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

- a *solution* of $\mathcal{N}$ is a mapping $\sigma : V \to \mathsf{D}$ 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 2b);
- $\mathcal{N}$ is *satisfiable* iff 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 $\mathcal{N}$ iff there exists (resp. there does not exist) a solution $\sigma : V \to \mathsf{D}$ of $\mathcal{N}$ such that $(\sigma(v), \sigma(v')) \in b$;

- $\mathcal{N}$ is *minimal* iff $\forall v, v' \in V$ 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 $\{u, v\} \in E$ iff $C(u, v) \neq \mathsf{B}$ and $u \neq v$;
- $\mathcal{N}$ is the *empty* QCN on V, denoted by $\bot^V$, iff $C(u, v) = \emptyset$ for all $u, v \in V$.

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 $C'(u, u') = C(u, u')\ \forall (u, u') \in (V \times V) \setminus \{(v, v'), (v', v)\}$.

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* iff $\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ 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 [35], and, hence, $\diamond$-consistency can be seen as a special case of ${}^{\diamond}_{G}$-consistency.

In [39] 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 [14] or quantitative temporal reasoning [13], 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 $\alpha : V \to \{0, 1, \ldots, n-1\}$, and a graph $G = (V', E)$, where $V' \subseteq V$, $\mathcal{N}$ is said to be ${}^{\overleftarrow{\diamond}}_{G}$*-consistent* iff $\forall v_i, v_j, v_k \in V$ 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. [40]), 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 $\underset{G}{\blacklozenge}$-consistent iff 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 $\underset{G}{\diamond}$-consistency on $\mathcal{N}$, the revised constraint $C(v, v')$ is always supported by $\{b\}$. Formally, $\underset{G}{\blacklozenge}$-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 $\underset{G}{\blacklozenge}$*-consistent* iff $\mathcal{N}$ is $\underset{G}{\diamond}$-consistent and $\forall \{v, v'\} \in E$ and $\forall b \in C(v, v')$ we have that $C'(v, v') = \{b\}$, where $(V, C') = \underset{G}{\diamond}(\mathcal{N}_{[v,v']/\{b\}})$.

If G is the complete graph on the variables of a given QCN, we can easily verify that $\underset{G}{\blacklozenge}$-consistency corresponds to $\underset{\mathsf{B}}{\diamond}$-consistency of the family of $\underset{f}{\diamond}$-consistencies studied in [11]. Interestingly, $\underset{G}{\blacklozenge}$-consistency for QCNs can also be seen as a counterpart of singleton arc consistency (SAC) [12] for CSPs.

Finally, in [42] the authors define a local consistency that is more restrictive than any of the *practical*[2] local consistencies known to date for QCNs, called *collective* $\underset{G}{\blacklozenge}$*-consistency*, or $\underset{G}{\blacklozenge}^{\cup}$-consistency for short. This singleton-style consistency is inspired by *k-partitioning consistency* for CSPs [5]. 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 $\underset{G}{\blacklozenge}^{\cup}$*-consistent* iff $\mathcal{N}$ is $\underset{G}{\blacklozenge}$-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 $(V, C') = \underset{G}{\diamond}(\mathcal{N}_{[u,u']/\{b'\}})$.

This underlying filtering condition of $\underset{G}{\blacklozenge}^{\cup}$-consistency is based on relation partitioning combined with $\underset{G}{\diamond}$-consistency, and allows for improved pruning capability over $\underset{G}{\blacklozenge}$-consistency [42].

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 $\underset{G}{\phi}$ and $\underset{G}{\psi}$ be two consistencies defined by some operations ϕ and ψ respectively and a graph G. Then, $\underset{G}{\phi}$ is *stronger* than $\underset{G}{\psi}$ iff whenever $\underset{G}{\phi}$ holds on a QCN $\mathcal{N}$ with respect to a graph G, $\underset{G}{\psi}$ also holds on $\mathcal{N}$ with respect to G, and $\underset{G}{\phi}$ is *strictly stronger* than $\underset{G}{\psi}$ iff $\underset{G}{\phi}$ is stronger than $\underset{G}{\psi}$ and there exists at least one QCN $\mathcal{N}$ and a graph G such that $\underset{G}{\psi}$ holds on $\mathcal{N}$ with respect to G, but $\underset{G}{\phi}$ does not hold on $\mathcal{N}$ with respect to G. (The terms *weaker* and *strictly weaker* can be defined likewise.) Finally, $\underset{G}{\phi}$ and $\underset{G}{\psi}$ are *incomparable* iff there exist QCNs $\mathcal{N}$ and $\mathcal{N}'$ such that $\underset{G}{\phi}$ is strictly stronger than $\underset{G}{\psi}$ with respect to $\mathcal{N}$ and some graph G, and $\underset{G}{\phi}$ is strictly weaker than $\underset{G}{\psi}$ with respect to $\mathcal{N}'$ and some graph G (we note that the graph G can be different in the two cases).

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 [47]. For that purpose, constraint programming researchers have proposed weaker singleton-style consistencies that localize propagation to the

[2] Clearly, in special cases notions like *k-consistency* can be defined and exploited theoretically [9], but these can be hardly implemented efficiently and are therefore not suitable for applications.

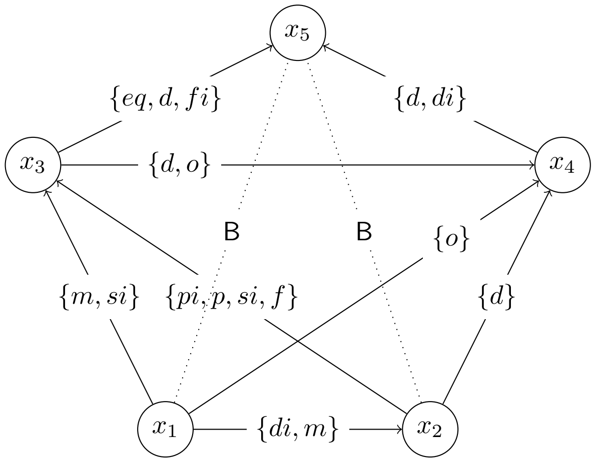


Figure 3 Given the QCN $\mathcal{N} = (V, C)$ above and the graph G that results by removing the edge $\{x_1, x_5\}$ from the complete graph on V, we have that $\mathcal{N}$ is neighbourhood $\blacklozenge_G^{\cup}$-consistent (and neighbourhood $\blacklozenge_G$-consistent), but not $\blacklozenge_G$-consistent (or $\blacklozenge_G^{\cup}$-consistent)

neighbourhood of the variable at hand [47, 33]. Neighbourhood singleton-style consistencies for CSPs, despite being strictly weaker than SAC [12] in general, can produce almost as much filtering as SAC with substantially less cost on many problems [33]. 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 and its accompanying graph shown in Figure 3. 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, iff $\mathcal{N}$ is $\lozenge_G$-consistent and $\forall\{v, v'\} \in E$ and $\forall b \in C(v, v')$ we have that $C'(v, v') = \{b\}$, where $(V, C') = \lozenge_{G_{\mathsf{N}(vv')}}(\mathcal{N}_{[v,v']/\{b\}})$.

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

▶ **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* $\blacklozenge_G^{\cup}$*-consistent*, or $\mathsf{N}\blacklozenge_G^{\cup}$-consistent for short, iff $\mathcal{N}$ 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 $\diamond_G$-consistency is restricted to the neighbourhood of the constraint at hand.

We recall the following result from [42] in our effort here to build a strength-based hierarchy among all discussed consistencies:

▶ **Proposition 1** ([42])**.** *We have that* $\blacklozenge_G^{\cup}$*-consistency is strictly stronger than* $\blacklozenge_G$*-consistency.*

In the sequel, Figure 3 will be crucial in proving the results that follow.

▶ **Proposition 2.** *We have that* $\blacklozenge_G^{\cup}$*-consistency is strictly stronger than* $\mathsf{N}\blacklozenge_G^{\cup}$*-consistency.*

Proof. Consider the QCN along with its accompanying graph depicted in Figure 3. As noted in its caption the QCN is $\mathsf{N}\blacklozenge_G^{\cup}$-consistent and $\mathsf{N}\blacklozenge_G$-consistent, but not $\blacklozenge_G$-consistent or $\blacklozenge_G^{\cup}$-consistent. Specifically, in order for the QCN to become $\blacklozenge_G$-consistent and $\blacklozenge_G^{\cup}$-consistent, the base relation mi needs to be removed from $C(x_2, x_5)$. In addition, by the definitions of $\blacklozenge_G^{\cup}$-consistency and $\mathsf{N}\blacklozenge_G^{\cup}$-consistency, we have that every $\blacklozenge_G^{\cup}$-consistent QCN is $\mathsf{N}\blacklozenge_G^{\cup}$-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 $\diamond_G$-consistent then $\mathcal{N}$ is $\diamond_{G'}$-consistent. ◀

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

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

We proceed with presenting the next result:

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

Proof. Consider the QCN along with its accompanying graph depicted in Figure 3 in [42]. It is the case that the QCN is $\mathsf{N}\blacklozenge_G$-consistent, but not $\mathsf{N}\blacklozenge_G^{\cup}$-consistent. 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.** *We have that* $\mathsf{N}\blacklozenge_G^{\cup}$*-consistency is incomparable to* $\blacklozenge_G$*-consistency.*

Proof. Consider again the QCN along with its accompanying graph depicted in Figure 3 in [42]. 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 also in the proof of Proposition 2, the QCN of Figure 3 here 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.** *We have that* $\blacklozenge_G^{\cup}$*-consistency is strictly stronger than* $\mathsf{N}\blacklozenge_G$*-consistency.*

Finally, to complete our strentgh-based hierarchy we close off with some results that involve the non-singleton-style consistencies $\diamond_G$-consistency and $\overleftarrow{\diamond}_G$-consistency.

▶ **Proposition 6.** *We have that* $\mathsf{N}\blacklozenge_G$*-consistency is strictly stronger than* $\diamond_G$*-consistency.*

Proof. Consider the QCN depicted in Figure 14 in [36], which was used to prove that $\diamond$-consistency cannot decide the minimality of a QCN in general. It is the case that the QCN is $\diamond_G$-consistent, but not $\mathsf{N}\blacklozenge_G$-consistent, with respect to the complete graph on the set of variables of that QCN. 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 $\diamond_G$-consistent. ◀

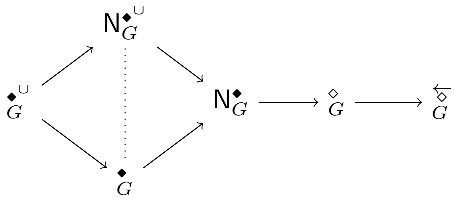


Figure 4 A strength-based hierarchy of consistencies for QCNs; an arrow denotes the (transitive) *strictly stronger* relationship and a dotted line the (symmetric) *incomparable* relationship

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

```
in   : A QCN N = (V, C), and a graph G = (V' ⊆ V, E).
out  : A sub-QCN of N.
begin
    N ← ◇_G(N);
    Q ← list(E);
    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});
        if (V, C') ⊂ N then
            flag ← False;
            foreach {u, u'} ∈ E do
                if C'(u, u') ⊂ C(u, u') then
                    C(u, u') ← C'(u, u');
                    C(u', u) ← C'(u', u);
                    flag ← True;
            if flag then Q ← list(E);
    return N;
```

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

▶ **Corollary 2.** *We have that each of the consistencies of* $\blacklozenge^{\cup}_{G}$*-consistency,* $\mathsf{N}\blacklozenge^{\cup}_{G}$*-consistency,* $\blacklozenge_{G}$*-consistency, and* $\mathsf{N}\blacklozenge_{G}$*-consistency is strictly stronger than* $\lozenge_{G}$*-consistency.*

From [40] we have the following result:

▶ **Proposition 7** ([40]). *We have that* $\lozenge_{G}$*-consistency is strictly stronger than* $\overleftarrow{\lozenge}_{G}$*-consistency.*

From Corollary 2 and Proposition 7 we obtain the following last result:

▶ **Corollary 3.** *We have that each of the consistencies of* $\blacklozenge^{\cup}_{G}$*-consistency,* $\mathsf{N}\blacklozenge^{\cup}_{G}$*-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 4.

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 $\mathcal{N}$.

```
begin
    N ← ◇_G(N);
    Q ← list(E);
    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}});
        if C'(v, v') ⊂ C(v, v') then
            C(v, v') ← C'(v, v');
            C(v', v) ← C'(v', v);
            Q ← list(E);
    return N;
```

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

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

```
begin
    N ← ◇_G(N);
    Q ← list(S ⊆ E);
    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;
```

Algorithms and Complexities

For the sake of completeness, we present here 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 [42].

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 $G = (V', E)$, where $V' \subseteq V$, we note that α is $O(\Delta|\mathsf{B}||E|)$, where Δ is the maximum vertex degree of G [10].

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 [41]. Specifically, contrary to the algorithm as it appears in [41], in Line 3 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 [41]).

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 is 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 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 intepreter under version 5.1.2, which implements Python 2.7.10. Only one CPU core was used.

Dataset

We used the dataset employed in [43]. That dataset comprises 1 000 random and structured QCNs of IA that were created using models $\mathsf{A}(\mathsf{n}, \mathsf{l}, \mathsf{d})$ [32] and $\mathsf{BA}(\mathsf{n}, \mathsf{m})$ [38] respectively. Pertaining to $\mathsf{A}(\mathsf{n}, \mathsf{l}, \mathsf{d})$, 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 $\mathsf{BA}(\mathsf{n}, \mathsf{m})$, there are 100 QCNs of IA of $\mathsf{n} = 150$ variables for all values of the constraint graph *preferential attachment* m [4] 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 [45] 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 [44]; the use of those graphs itself was inspired by [23, 22, 27] among other works, where preliminary results pertainining 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:[3]

- Solver, the state-of-the-art reasoner for checking the satisfiability of QCNs of Interval Algebra and RCC8 that was introduced in [38] (and in particular the reasoner called Phalanx▽ 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 [2];[4]
- PWC, the state-of-the-art algorithm for applying $\diamond_G$-consistency on QCNs of Interval Algebra and RCC8 that was used in [38] (which is a module of the Phalanx▽ reasoner mentioned earlier);
- DPWC, the state-of-the-art algorithm for applying $\overleftarrow{\diamond}_G$-consistency on QCNs of Interval Algebra and RCC8 that was introduced in [40] (and in particular the reasoner called Pyrrhus in that work).

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 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 Minimizer, the computational task is much more time-consuming; therefore, Minimizer has by far the worst performance among its competition.

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 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 $\mathsf{PSWC}^{\cup}_{\mathsf{N}}$ and $\mathsf{PSWC}_{\mathsf{N}}$ are around 30% faster in the phase transition region than the standard algorithms $\mathsf{PSWC}^{\cup}$ and PSWC respectively, whilst retaining much of the good performance characteristics

---

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

[4] 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., $\blacklozenge^{\cup}_G$-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 A(n = 70, l = 6.5, d) [32]

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

Table 2 Evaluation with structured IA networks that were generated using model BA(n = 150, m) [38]

m	Solver	Minimizer	$PSWC^{\cup}$	$PSWC^{\cup}_{N}$	$\ell PSWC^{\cup}$	$\ell PSWC^{\cup}_{N}$	PSWC	$PSWC_{N}$	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	$\frac{0.16s}{100}$	$\frac{0.71s}{100\%}$	$\frac{0.04s}{100 \mid 100\%}$	$\frac{0.07s}{100 \mid 100\%}$	$\frac{0.06s}{100 \mid 100\%}$	$\frac{0.06s}{100 \mid 100\%}$	$\frac{0.05s}{100 \mid 100\%}$	$\frac{0.07s}{100 \mid 100\%}$	$\frac{0.01s}{92 \mid 92.32\%}$	$\frac{0.00s}{29 \mid 28.91\%}$

(viz., unfeasible base relations elimination and satisfiability decision) of the latter. The parsimonious variants $\ell\mathsf{PSWC}^{\cup}$ and $\ell\mathsf{PSWC}_{\mathsf{N}}^{\cup}$ are up to 6 times faster in the phase transition region than $\mathsf{PSWC}^{\cup}$ and $\mathsf{PSWC}_{\mathsf{N}}^{\cup}$ respectively, but detect in general slightly fewer unsatisfiable network instances and eliminate slightly fewer unfeasible base relations 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 in Line 3 of the parsimonious variants (see Algorithm 3) corresponds to the set of edges $E(\mathsf{G}(\diamond_G(N)))$, i.e., the set of edges of the constraint graph of $\diamond_G(N)$.

Synopsis

In conclusion, and with respect to the involved datasets 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 Minimizer is provided with a minimal network instance (as it was usually the case in our evaluation due to the preprocessing with $\blacklozenge_G^{\cup}$-consistency, see again Footnote 4 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 [30]. We emphasize again the fact that the neighbourhood-focused singleton-style algorithms $\mathsf{PSWC}_{\mathsf{N}}^{\cup}$ and $\mathsf{PSWC}_{\mathsf{N}}$ were found to be around 30% faster in the phase transition region than the standard algorithms $\mathsf{PSWC}^{\cup}$ and PSWC 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. Regarding the parsimonious variants in particular, viz., $\ell\mathsf{PSWC}^{\cup}$ and $\ell\mathsf{PSWC}_{\mathsf{N}}^{\cup}$, 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.

6 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 are 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* [26], establishing adaptive constraint propagators for QCNs (see [3] 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 [24, 8]. 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 [16].

References

1. James F. Allen. Maintaining Knowledge about Temporal Intervals. *Commun. ACM*, 26:832–843, 1983.
2. Nouhad Amaneddine, Jean-François Condotta, and Michael Sioutis. Efficient Approach to Solve the Minimal Labeling Problem of Temporal and Spatial Qualitative Constraints. In *IJCAI*, 2013.
3. Amine Balafrej, Christian Bessiere, El-Houssine Bouyakhf, and Gilles Trombettoni. Adaptive Singleton-Based Consistencies. In *AAAI*, 2014.
4. Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.
5. Hachemi Bennaceur and Mohamed-Salah Affane. Partition-k-AC: An Efficient Filtering Technique Combining Domain Partition and Arc Consistency. In *CP*, 2001.
6. Mehul Bhatt, Hans Guesgen, Stefan Wölfl, and Shyamanta Hazarika. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. *Spatial Cognition & Computation*, 11:1–14, 2011.
7. Mehul Bhatt and Jan Oliver Wallgrün. Geospatial Narratives and Their Spatio-Temporal Dynamics: Commonsense Reasoning for High-Level Analyses in Geographic Information Systems. *ISPRS Int. J. Geo-Information*, 3:166–205, 2014.
8. Manuel Bodirsky, Peter Jonsson, Barnaby Martin, and Antoine Mottet. Classification Transfer for Qualitative Reasoning Problems. In *IJCAI*, 2018.
9. Manuel Bodirsky and Stefan Wölfl. RCC8 is polynomial on networks of bounded treewidth. In *IJCAI*, 2011.
10. Assef Chmeiss and Jean-François Condotta. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI*, 2011.
11. Jean-François Condotta and Christophe Lecoutre. A Class of $\diamond_f$-Consistencies for Qualitative Constraint Networks. In *KR*, 2010.
12. Romuald Debruyne and Christian Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *IJCAI*, 1997.
13. Rina Dechter, Itay Meiri, and Judea Pearl. Temporal Constraint Networks. *Artif. Intell.*, 49:61–95, 1991.
14. Rina Dechter and Judea Pearl. Network-Based Heuristics for Constraint-Satisfaction Problems. *Artif. Intell.*, 34:1–38, 1987.
15. Krishna Sandeep Reddy Dubba, Anthony G. Cohn, David C. Hogg, Mehul Bhatt, and Frank Dylla. Learning Relational Event Models from Video. *J. Artif. Intell. Res.*, 53:41–90, 2015.
16. Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André van Delden, Jasper van de Ven, and Diedrich Wolter. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Comput. Surv.*, 50:7:1–7:39, 2017.
17. Frank Dylla, Till Mossakowski, Thomas Schneider, and Diedrich Wolter. Algebraic Properties of Qualitative Spatio-Temporal Calculi. In *COSIT*, 2013.
18. Frank Dylla and Jan Oliver Wallgrün. Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. *J. Intell. Robotic Syst.*, 48:55–78, 2007.
19. Alfonso Gerevini and Alessandro Saetti. Computing the minimal relations in point-based qualitative temporal reasoning through metagraph closure. *Artif. Intell.*, 175:556–585, 2011.
20. Martin Charles Golumbic and Ron Shamir. Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach. *J. ACM*, 40:1108–1133, 1993.
21. Georg Gottlob. On minimal constraint networks. *Artif. Intell.*, 191-192:42–60, 2012.
22. Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *KR*, 2012.
23. Jinbo Huang, Jason Jingshi Li, and Jochen Renz. Decomposition and tractability in qualitative spatial and temporal reasoning. *Artif. Intell.*, 195:140–164, 2013.
24. Peter Jonsson and Victor Lagerkvist. Why are CSPs Based on Partition Schemes Computationally Hard? In *MFCS*, 2018.

25 Nikhil Krishnaswamy, Scott Friedman, and James Pustejovsky. Combining Deep Learning and Qualitative Spatial Reasoning to Learn Complex Structures from Sparse Examples with Noise. In *AAAI*, 2019.

26 Olivier Lhomme. Quick Shaving. In *AAAI*, 2005.

27 Jason Jingshi Li, Jinbo Huang, and Jochen Renz. A divide-and-conquer approach for solving interval algebra networks. In *IJCAI*, 2009.

28 Gérard Ligozat. *Qualitative Spatial and Temporal Reasoning*. Wiley, 2013.

29 Gérard Ligozat and Jochen Renz. What Is a Qualitative Calculus? A General Framework. In *PRICAI*, 2004.

30 Weiming Liu and Sanjiang Li. Solving Minimal Constraint Networks in Qualitative Spatial and Temporal Reasoning. In *CP*, 2012.

31 Ugo Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Inf. Sci.*, 7:95–132, 1974.

32 Bernhard Nebel. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. *Constraints*, 1:175–190, 1997.

33 Anastasia Paparrizou and Kostas Stergiou. On Neighborhood Singleton Consistencies. In *IJCAI*, 2017.

34 David A. Randell, Zhan Cui, and Anthony Cohn. A Spatial Logic Based on Regions & Connection. In *KR*, 1992.

35 Jochen Renz and Gérard Ligozat. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP*, 2005.

36 Jochen Renz and Bernhard Nebel. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *Artif. Intell.*, 108:69–123, 1999.

37 Jochen Renz and Bernhard Nebel. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*, pages 161–215. 2007.

38 Michael Sioutis, Jean-François Condotta, and Manolis Koubarakis. An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks. *Int. J. Artif. Intell. Tools*, 25:1–33, 2016.

39 Michael Sioutis, Zhiguo Long, and Sanjiang Li. Efficiently Reasoning about Qualitative Constraints through Variable Elimination. In *SETN*, 2016.

40 Michael Sioutis, Zhiguo Long, and Sanjiang Li. Leveraging Variable Elimination for Efficiently Reasoning about Qualitative Constraints. *Int. J. Artif. Intell. Tools*, 27:1860001, 2018.

41 Michael Sioutis, Anastasia Paparrizou, and Jean-François Condotta. A Lazy Algorithm to Efficiently Approximate Singleton Path Consistency for Qualitative Constraint Networks. In *ICTAI*, 2017.

42 Michael Sioutis, Anastasia Paparrizou, and Jean-François Condotta. Collective Singleton-Based Consistency for Qualitative Constraint Networks. In *TIME*, 2017.

43 Michael Sioutis, Anastasia Paparrizou, and Jean-François Condotta. Collective Singleton-Based Consistency for Qualitative Constraint Networks: Theory and Practice. *Theor. Comput. Sci*, 2019. In press.

44 Michael Sioutis, Yakoub Salhi, and Jean-François Condotta. Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning. *Knowl. Eng. Rev.*, 32:e4, 2016.

45 Robert E. Tarjan and Mihalis Yannakakis. Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs. *SIAM J. Comput.*, 13:566–579, 1984.

46 Peter van Beek and Rina Dechter. On the Minimality and Decomposability of Row-Convex Constraint Networks. *J. ACM*, 42:543–561, 1995.

47 Richard J. Wallace. Neighbourhood SAC: Extensions and new algorithms. *AI Commun.*, 29:249–268, 2016.