Collective Singleton-based Consistency for Qualitative Constraint Networks

Michael Sioutis[1], Anastasia Paparrizou[2], and Jean-François Condotta[3]

1 **Örebro University, MPI@AASS, Örebro, Sweden**
`michael.sioutis@oru.se`

2 **Artois University, CRIL UMR 8188, Lens, France**
`paparrizou@cril.fr`

3 **Artois University, CRIL UMR 8188, Lens, France**
`condotta@cril.fr`

Abstract

Partial singleton closure under weak composition, or partial $\blacklozenge$-consistency for short, is essential for approximating satisfiability of qualitative constraints networks. Briefly put, partial $\blacklozenge$-consistency ensures that each base relation of each of the constraints of a qualitative constraint network can define a singleton relation in the corresponding partial closure of that network under weak composition, or in its corresponding partially $\diamond$-consistent subnetwork for short. In particular, partial $\blacklozenge$-consistency has been shown to play a crucial role in tackling the minimal labeling problem of a qualitative constraint network, which is the problem of finding the strongest implied constraints of that network. In this paper, we propose a stronger local consistency that couples $\blacklozenge$-consistency with the idea of collectively deleting certain unfeasible base relations by exploiting singleton checks. We then propose an efficient algorithm for enforcing this new consistency that, given a qualitative constraint network, performs fewer constraint checks than the respective algorithm for enforcing partial $\blacklozenge$-consistency in that network. We formally prove certain properties of our new local consistency, and motivate its usefulness through demonstrative examples and a preliminary experimental evaluation with qualitative constraint networks of Interval Algebra.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases Qualitative constraint network; qualitative spatial and temporal reasoning; partial singleton path-consistency; local consistency; minimal labeling problem

Digital Object Identifier 10.4230/LIPIcs.TIME.2017.19

1 Introduction

Qualitative Spatial and Temporal Reasoning (QSTR) is a major field of study in Artificial Intelligence, and in particular in Knowledge Representation & Reasoning. This field has received a lot of attention over the past decades, as it extends to a plethora of areas and domains that include ambient intelligence, dynamic GIS, cognitive robotics, and spatiotemporal design [6]. QSTR abstracts from numerical quantities of space and time by using qualitative descriptions instead (e.g., *precedes*, *contains*, *is left of*), thus providing a concise framework that allows for rather inexpensive reasoning about entities located in space and time.

The problem of representing and reasoning about qualitative information can be modeled as a qualitative constraint network (QCN) using a qualitative constraint language. Specifically, a QCN is a network of constraints corresponding to qualitative spatial or temporal relations between spatial or temporal variables respectively, and a qualitative constraint language is used to define those constraints over a finite set of binary relations, called *base relations* (or *atoms*) [19]. An example of such a qualitative constraint language is Interval Algebra (IA), introduced by Allen [1]. IA considers time intervals (as its temporal entities) and each of its base relations represents an ordering of the four endpoints of two intervals in the timeline.

The fundamental reasoning problems associated with a given QCN $\mathcal{N}$ are the problems of *satisfiability checking*, *minimal labeling* (or *deductive closure*), and *redundancy* (or *entailment*) [28]. In particular, the satisfiability checking problem is the problem of deciding if there exists a spatial or temporal valuation of the variables of $\mathcal{N}$ that satisfies its constraints, such a valuation being called a *solution* of $\mathcal{N}$, the minimal labeling problem is the problem of finding the strongest implied constraints of $\mathcal{N}$, and the redundancy problem is the problem of determining if a given constraint is entailed by the rest of $\mathcal{N}$ (that constraint being called redundant, as its removal does not change the solution set of the QCN). In general, for most qualitative constraint languages the satisfiability checking problem is NP-complete. Further, the redundancy problem, the minimal labeling problem, and the satisfiability checking problem are equivalent under polynomial Turing reductions [13].

The vast amount, if not all, of the published works that study the aforementioned reasoning problems, use *partial* $\diamond$*-consistency* as a means to define practical algorithms for efficiently tackling them [2, 30, 29, 17, 27, 23, 15]. Given a QCN $\mathcal{N}$ and a graph G, partial $\diamond$-consistency with respect to G, denoted by $\overset{\diamond}{_G}$-consistency, entails (weak) consistency for all triples of variables in $\mathcal{N}$ that correspond to three-vertex cycles (triangles) in G. We note that if G is complete, $\overset{\diamond}{_G}$-consistency becomes identical to $\diamond$-*consistency* [26]. Hence, $\diamond$-consistency is a special case of $\overset{\diamond}{_G}$-consistency. In fact, earlier works have relied solely on $\diamond$-consistency; it was not until the introduction of chordal (or triangulated) graphs in QSTR, due to some generalized theoretical results of [14], that researchers started restricting $\diamond$-consistency to a triangulation (or chordal completion) of the constraint graph of an input QCN and benefiting from better complexity properties in more recent works.

Adding to the previous paragraph, and with respect to the satisfiability checking problem in particular, the literature suggests that $\overset{\diamond}{_G}$-consistency alone is sufficient in most cases to guarantee that a solution for a given QCN, should it exist, is efficiently obtained (see also [8]). However, for the more challenging problems of minimal labeling and redundancy, a stronger local consistency is typically employed that builds upon $\overset{\diamond}{_G}$-consistency, called singleton $\overset{\diamond}{_G}$-consistency and denoted by $\overset{\blacklozenge}{_G}$-consistency [2, 30]. Given a QCN $\mathcal{N}$ and a graph G, $\overset{\blacklozenge}{_G}$-consistency holds on $\mathcal{N}$ if and only if each base relation of each of the constraints of $\mathcal{N}$ is closed under $\overset{\diamond}{_G}$-consistency, i.e., after instatiating any constraint of that network with one of its base relations b and closing the network under $\overset{\diamond}{_G}$-consistency, the corresponding constraint in the $\overset{\diamond}{_G}$-consistent subnetwork will continue being defined by b.

It is then natural to ask whether we can have an even stronger local consistency than $\overset{\blacklozenge}{_G}$-consistency (and $\overset{\diamond}{_G}$-consistency) for QCNs that can also be enforced more efficiently than $\overset{\blacklozenge}{_G}$-consistency, as a positive answer to that question would suggest an immediate improvement for any algorithm that currently employs $\overset{\blacklozenge}{_G}$-consistency. In this paper, we contribute towards obtaining such a positive answer. In particular, we enrich the family of consistencies for QCNs by proposing a new singleton style consistency inspired by k-partitioning consistency for constraint satisfaction problems (CSPs) [4]. This filtering technique is based on domain partitioning combined with a local consistency, typically *arc consistency* [5], and allows for improved pruning capability over *singleton arc consistency* [9]. Similarly to k-partitioning consistency, our new consistency, denoted by $\overset{\blacklozenge}{_G}{}^{\cup}$-consistency, combines singleton checks and $\overset{\diamond}{_G}$-consistency to present itself as a better alternative to $\overset{\blacklozenge}{_G}$-consistency. With respect to our new consistency, we also propose an algorithm for applying it on a given QCN, which turns

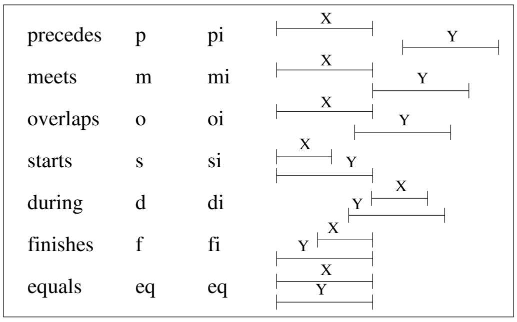


Figure 1 The base relations of IA

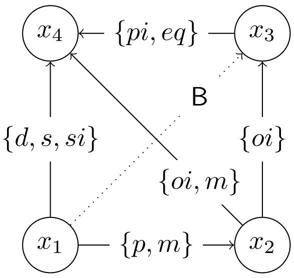
out being more efficient than the respective algorirthm for applying $\blacklozenge_G$-consistency on that same QCN. As a brief intuitive explanation of this, $\blacklozenge_G^{\cup}$-consistency allows for proactively eliminating base relations anywhere in a given QCN and not only in the set of base relations of the constraint at hand that is singleton checked. Further, we obtain several theoretical results and show, among other things, that $\blacklozenge_G^{\cup}$-consistency is *strictly stronger* than $\blacklozenge_G$-consistency and, hence, than $\diamond_G$-consistency. Finally, we present a preliminary experimental evaluation of $\blacklozenge_G^{\cup}$-consistency and $\blacklozenge_G$-consistency using QCNs of IA, in support of our argument that $\blacklozenge_G^{\cup}$-consistency can be enforced more efficiently than $\blacklozenge_G$-consistency for a given QCN.

The rest of the paper is organized as follows. In Section 2 we give some preliminaries on qualitative spatial and temporal reasoning, and in Section 3 we focus on $\diamond_G$-consistency and $\blacklozenge_G$-consistency and, in particular, recall some related result from the literature, but also provide some new results of our own. Then, in Section 4 we introduce, formally define, and thoroughly study our new local consistency, namely, $\blacklozenge_G^{\cup}$-consistency. In Section 5 we present an algorithm for efficiently applying $\blacklozenge_G^{\cup}$-consistency on a given QCN $\mathcal{N}$, and in Section 6 we evaluate this algorithm against the state-of-the-art algorithm for achieving $\blacklozenge_G$-consistency. Finally, in Section 7 we conclude the paper and give some directions for future work.

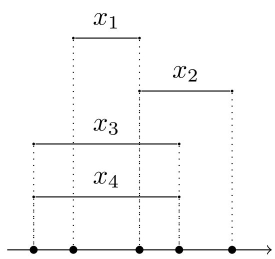
2 Preliminaries

A (binary) qualitative spatial or temporal constraint language, is based on a finite set B of *jointly exhaustive and pairwise disjoint* relations defined over an infinite domain D, which is called the set of *base relations* [19]. 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 $\diamond$ [19]. 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 $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r, b' \in r'\}$.

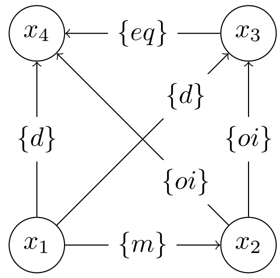
As an illustration, consider the well known qualitative temporal constraint language of Interval Algebra (IA) introduced by Allen [1]. IA considers time intervals (as its temporal entities) and the set of base relations $\mathsf{B} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$; each base



(a) A satisfiable QCN $\mathcal{N}$



(b) A solution σ of $\mathcal{N}$



(c) A scenario $\mathcal{S}$ of $\mathcal{N}$

Figure 2 Figurative examples of QCN terminology using IA

relation of IA represents a particular ordering of the four endpoints of two intervals in the timeline, as demonstrated in Figure 1. The base relation *eq* is the identity relation Id of IA. As another illustration, the Region Connection Calculus (RCC) is a first-order theory for representing and reasoning about mereotopological information [24]. The domain D of RCC comprises all possible non-empty regular subsets of some topological space. These subsets do not have to be internally connected and do not have a particular dimension, but are commonly required to be regular *closed* [25]. Other notable and well known qualitative spatial and temporal constraint languages include Point Algebra [35], Cardinal Direction Calculus [18, 11], and Block Algebra [3].

The weak composition operation $\diamond$, the converse operation $^{-1}$, the union operation $\cup$, the complement operation $^{\complement}$, and the total set of relations 2^{B} along with the identity relation Id of a qualitative constraint language, form an algebraic structure $(2^{\mathsf{B}}, \mathsf{Id}, \diamond, ^{-1}, ^{\complement}, \cup)$ that can correspond to a *relation algebra* in the sense of Tarski [33].

▶ **Proposition 1** ([10]). *The languages of Point Algebra, Cardinal Direction Calculus, Interval Algebra, Block Algebra, and* RCC*-8 are each a relation algebra with the algebraic structure* $(2^{\mathsf{B}}, \mathsf{Id}, \diamond, ^{-1}, ^{\complement}, \cup)$.

In what follows, for a qualitative constraint language that is a relation algebra with the algebraic structure $(2^{\mathsf{B}}, \mathsf{Id}, \diamond, ^{-1}, ^{\complement}, \cup)$, we will simply use the term *relation algebra*, as the algebraic structure will always be of the same format.

The problem of representing and reasoning about qualitative information can be modeled as a *qualitative constraint network* (QCN), 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;
- 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 $v, v' \in V$.

An example of a QCN of IA is shown in Figure 2a; for simplicity, 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 QCN is *equivalent* to $\mathcal{N}$ iff it admits the same set of solutions as $\mathcal{N}$;
- 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 $\mathcal{N}' \subset \mathcal{N}$;

- $\mathcal{N}$ is *atomic* iff $\forall v, v' \in V$, $C(v, v')$ is a *singleton relation*, i.e., a relation $\{b\}$ with $b \in \mathsf{B}$;
- a *scenario* $\mathcal{S}$ of $\mathcal{N}$ is an atomic satisfiable sub-QCN of $\mathcal{N}$ (see Figure 2c);
- a base relation $b \in C(v, v')$ with $v, v' \in V$ is *feasible* (resp. *unfeasible*) iff there exists (resp. there does not exist) a scenario $\mathcal{S} = (V, C')$ of $\mathcal{N}$ such that $C'(v, v') = \{b\}$;
- $\mathcal{N}$ is *minimal* iff $\forall v, v' \in V$ and $\forall b \in C(v, v')$, b is a feasible base relation of $\mathcal{N}$;
- 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 *trivially inconsistent* iff $\exists u, v \in V$ such that $C(u, v) = \emptyset$;
- $\mathcal{N}$ 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 operations with respect to QCNs:

- given a QCN $\mathcal{N} = (V, C)$ and $v, v' \in V$, we have 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 $C'(y, w) = C(y, w)$ $\forall (y, w) \in (V \times V) \setminus \{(v, v'), (v', v)\}$;
- given two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ on the same set of variables V, we have that $\mathcal{N} \cup \mathcal{N}'$ yields the QCN $\mathcal{N}'' = (V, C'')$, where $C''(v, v') = C(v, v') \cup C'(v, v')$ for all $v, v' \in V$.

We recall the following definition of $\overset{\diamond}{_G}$-consistency, which, as noted in the introduction, is the basic local consistency used in the literature for solving fundamental reasoning problems of QCNs, such as the satisfiability checking problem.

▶ **Definition 3.** Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, $\mathcal{N}$ is said to be $\overset{\diamond}{_G}$*-consistent* iff $\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ we have that $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$.

We note again that if G is complete, $\overset{\diamond}{_G}$-consistency becomes identical to $\diamond$-consistency [26], and, hence, $\diamond$-consistency is a special case of $\overset{\diamond}{_G}$-consistency.

Given a graph $G = (V, E)$, a QCN $\mathcal{N} = (V, C)$ is $\overset{\blacklozenge}{_G}$-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 $\{b\}$ and computing the closure of $\mathcal{N}$ under $\overset{\diamond}{_G}$-consistency, the revised constraint $C(v, v')$ is always defined by $\{b\}$. Formally, $\overset{\blacklozenge}{_G}$-consistency of a QCN is defined as follows:

▶ **Definition 4.** Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, $\mathcal{N}$ is said to be $\overset{\blacklozenge}{_G}$-consistent iff $\forall \{v, v'\} \in E$ and $\forall b \in C(v, v')$ we have that $\{b\} = C'(v, v')$, where $(V, C') = \overset{\diamond}{_G}(\mathcal{N}_{[v,v']/\{b\}})$.

If G is a complete graph, i.e., $G = K_V$, we can easily verify that $\overset{\blacklozenge}{_G}$-consistency corresponds to $\overset{\diamond}{_{\mathsf{B}}}$-consistency of the family of $\overset{\diamond}{_f}$-consistencies studied in [8]. Interestingly, $\overset{\blacklozenge}{_G}$-consistency can also be seen as a counterpart of singleton arc consistency (SAC) [9] for QCNs. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, for every $b \in \mathsf{B}$ and every $\{v, v'\} \in E$, we will say that b is $\overset{\blacklozenge}{_G}$-consistent for $C(v, v')$ iff $\{b\} = C'(v, v')$, where $(V, C') = \overset{\diamond}{_G}(\mathcal{N}_{[v,v']/\{b\}})$.

▶ **Definition 5.** A *subclass* of relations is a subset $\mathcal{A} \subseteq 2^{\mathsf{B}}$ that contains the singleton relations of 2^{B} and is closed under converse, intersection, and weak composition.

Given three relations r, r', $r'' \in 2^{\mathsf{B}}$, we say that weak composition distributes over intersection if we have that $r \diamond (r' \cap r'') = (r \diamond r') \cap (r \diamond r'')$ and $(r' \cap r'') \diamond r = (r' \diamond r) \cap (r'' \diamond r)$.

▶ **Definition 6.** A subclass $\mathcal{A}$ is *distributive* iff weak composition distributes over non-empty intersection $\forall r, r', r'' \in \mathcal{A}$.

Distributive subclasses of relations are defined for all of the qualitative constraint languages mentioned in Proposition 1 [20].

Finally, for the sake of simplicity in phrasing some results, in what follows we assume that all considered graphs are biconnected.

3 A closer look at $^{\diamond}_{G}$-consistency and $^{\blacklozenge}_{G}$-consistency

Let us come back to $^{\diamond}_{G}$-consistency and $^{\blacklozenge}_{G}$-consistency and recall in this section some results from the literature that will be relevant in the rest of the paper, but also provide some new results of our own.

In order to compare the pruning (or inference) capability of different consistencies, we introduce a preorder. Let $^{\phi}_{G}$ and $^{\psi}_{G}$ be two consistencies defined by some operations ϕ and ψ respectively and a graph G. Then, $^{\phi}_{G}$ is *stronger* than $^{\psi}_{G}$, denoted by $^{\phi}_{G} \trianglerighteq ^{\psi}_{G}$, iff whenever $^{\phi}_{G}$ holds on a QCN $\mathcal{N}$ with respect to a graph G, $^{\psi}_{G}$ also holds on $\mathcal{N}$ with respect to G, and $^{\phi}_{G}$ is *strictly stronger* than $^{\psi}_{G}$, denoted by $^{\phi}_{G} \triangleright ^{\psi}_{G}$, iff $^{\phi}_{G} \trianglerighteq ^{\psi}_{G}$ and there exists at least one QCN $\mathcal{N}$ and a graph G such that $^{\psi}_{G}$ holds on $\mathcal{N}$ with respect to G but $^{\phi}_{G}$ does not hold on $\mathcal{N}$ with respect to G. Finally, $^{\phi}_{G}$ and $^{\psi}_{G}$ are *equivalent*, denoted by $^{\phi}_{G} \equiv ^{\psi}_{G}$, iff we have both $^{\phi}_{G} \trianglerighteq ^{\psi}_{G}$ and $^{\psi}_{G} \trianglerighteq ^{\phi}_{G}$.

We now recall the definition of a *well-behaved* consistency [8].

▶ **Definition 7.** A consistency $^{\phi}_{G}$ is *well-behaved* iff for any QCN $\mathcal{N} = (V, E)$ and any graph $G = (V, E)$ the following properties hold:

- $^{\phi}_{G}(\mathcal{N}) \subseteq \mathcal{N}$ (viz., the $^{\phi}_{G}$-closure of $\mathcal{N}$ w.r.t. G) is the largest (w.r.t. $\subseteq$) $^{\phi}_{G}$-consistent sub-QCN of $\mathcal{N}$ *(Dominance)*;
- $^{\phi}_{G}(\mathcal{N})$ is equivalent to $\mathcal{N}$ *(Equivalence)*;
- $^{\phi}_{G}(^{\phi}_{G}(\mathcal{N})) = ^{\phi}_{G}(\mathcal{N})$ *(Idempotence)*;
- if $\mathcal{N}' \subseteq \mathcal{N}$ then $^{\phi}_{G}(\mathcal{N}') \subseteq ^{\phi}_{G}(\mathcal{N})$ *(Monotonicity)*.

It is routine to formally prove the following result for $^{\diamond}_{G}$-consistency:

▶ **Corollary 1** (cf. [8]). *We have that $^{\diamond}_{G}$-consistency is well-behaved.*

It is routine to formally prove the following result for $^{\blacklozenge}_{G}$-consistency as well:

▶ **Corollary 2** (cf. [8]). *We have that $^{\blacklozenge}_{G}$-consistency is well-behaved.*

The aforementioned two results are derived from respective results of [8] where complete graphs are used in all cases. The generalization to an arbitrary graph G is trivial.

We recall the following general result regarding the pruning capability of $^{\diamond}_{G}$-consistency in comparison with that of $^{\blacklozenge}_{G}$-consistency:

▶ **Proposition 2** ([8]). *We have that $^{\blacklozenge}_{G}$-consistency $\triangleright$ $^{\diamond}_{G}$-consistency.*

Before we proceed, we introduce the following lemma to be used in the next proposition:

▶ **Lemma 1** (cf. [20]). *Let $\mathcal{A}$ be a distributive subclass of relations of a relation algebra. Then, for any* QCN *$\mathcal{N} = (V, C)$ over $\mathcal{A}$ and any graph $G = (V, E)$, if $^{\diamond}_{G}(\mathcal{N}) = (V, C')$ is not trivially inconsistent, we have that $\forall u, v \in V$ and $\forall b \in C'(u, v)$ there exists an atomic $^{\diamond}_{G}$-consistent sub-*QCN *(V, C'') of $\mathcal{N}$ such that $\{b\} = C''(u, v)$.*

Proof. (Sketch.) The proof can be obtained by concatenating the proofs of Theorems 2 and 5 in [20] and applying that merged proof on each maximal chordal subgraph of G. The only major difference is that in those proofs the property that *any atomic* QCN *of a relation algebra that is $\diamond$-consistent is satisfiable* is used in addition to guarantee a stronger result, which is of no use to us for proving this particular lemma. ◀

Next, we introduce a result that identifies the case where $^{\diamond}_{G}$-consistency and $^{\blacklozenge}_{G}$-consistency are equivalent.

▶ **Proposition 3.** *Let $\mathcal{A}$ be a distributive subclass of relations of a relation algebra. Then, for any* QCN $\mathcal{N} = (V, C)$ *over $\mathcal{A}$ and any graph $G = (V, E)$, we have that $\diamond_G$-consistency $\equiv$ $\blacklozenge_G$-consistency.*

Proof. If $\diamond_G(\mathcal{N}) = (V, C)$ is not trivially inconsistent, then by Lemma 1 we have that $\forall u, v \in V$ and $\forall b \in C(u, v)$ there exists an atomic $\diamond_G$-consistent sub-QCN (V, C') of $\diamond_G(\mathcal{N})$ such that $\{b\} = C'(u, v)$. This suggests that $\diamond_G(\mathcal{N})$ is also $\blacklozenge_G$-consistent and, hence, that $\diamond_G$-consistency $\trianglerighteq$ $\blacklozenge_G$-consistency in this case. If $\diamond_G(\mathcal{N})$ is trivially inconsistent, then due to the closure under $\diamond_G$-consistency there is no triangle in G containing both an edge $\{v, v'\}$ such that $C(v, v') = \emptyset$ and an edge $\{u, u'\}$ such that $C(u, u') \neq \emptyset$; we can prove that $\diamond_G$-consistency $\trianglerighteq$ $\blacklozenge_G$-consistency in this case as well, by isolating the trivial inconsistencies and using the first part of the proof. Finally, by Proposition 2 we have that $\blacklozenge_G$-consistency $\trianglerighteq$ $\diamond_G$-consistency in all cases. ◀

It is interesting to note that Proposition 3 is a more general result that the respective one of [30], namely, Proposition 7 in that work. In particular, Proposition 7 in [30] requires a chordal supergraph of the constraint graph of a QCN over a distributive subclass of relations of a relation algebra to be used, along with the property that any such QCN that is $\diamond$-consistent and not trivially inconsistent is minimal, in order to prove the equivalence between $\diamond_G$-consistency and $\blacklozenge_G$-consistency for distributive subclasses of relations.

The following result shows the connection between $\diamond_G$-consistency and minimal QCNs:

▶ **Proposition 4** ([21]). *Let $\mathcal{A}$ be a distributive subclass of relations of a relation algebra with the property that any atomic* QCN *over $\mathcal{A}$ that is $\diamond$-consistent is satisfiable. Then, for any* QCN $\mathcal{N} = (V, C)$ *over $\mathcal{A}$ and any chordal graph $G = (V, E)$ such that* $\mathsf{G}(\mathcal{N}) \subseteq G$, *we have that $\forall \{u, v\} \in E$ and $\forall b \in C'(u, v)$, where $(V, C') = \diamond_G(\mathcal{N})$, the base relation b is feasible.*

The property described in Proposition 4 is satisfied by all of the qualitative constraint languages mentioned in Proposition 1 [10].

Finally, the following result shows the connection between $\blacklozenge_G$-consistency and minimal QCNs:

▶ **Proposition 5** ([2]). *Let $\mathcal{A}$ be a subclass of relations of a relation algebra with the property that for any* QCN $\mathcal{N} = (V, C)$ *over $\mathcal{A}$ there exists a graph $G = (V, E)$ such that, if $\diamond_G(\mathcal{N})$ is not trivially inconsistent, then $\mathcal{N}$ is satisfiable. Then, for any such $\mathcal{N}$ and G, we have that $\forall \{u, v\} \in E$ and $\forall b \in C'(u, v)$, where $(V, C') = \blacklozenge_G(\mathcal{N})$, the base relation b is feasible.*

As a note, an interesting case where the property described in Proposition 5 can be satisfied, is the case where the considered subclass of relations is obtained from a relation algebra that has *patchwork* [22] for $\diamond_G$-consistent and not trivially inconsistent QCNs over that subclass, where $G = (V, E)$ is any chordal graph such that $\mathsf{G}(\mathcal{N}) \subseteq G$ for a given QCN $\mathcal{N} = (V, C)$. In that case, we will indeed have that $\mathcal{N}$ is satisfiable if $\diamond_G(\mathcal{N})$ is not trivially inconsistent [2]. As a matter of fact, patchwork holds for all the qualitative constraint languages mentioned in Proposition 1 [14]. Of course, in general, the property may be satisfied in other cases as well; for instance, patchwork may not hold, but the overall property may hold for complete graphs (and, hence, when $\diamond$-consistency is used) or when constraints in the structure of the constraint graphs of the QCNs are imposed (a trivial case being restricting the constraint graphs of QCNs to being trees).

4 $\blacklozenge_G^{\cup}$-Consistency: a new local consistency for QCNs

We define a new local consistency for QCNs inspired by k-partitioning consistency for constraint satisfaction problems (CSPs), where arc consistency is used as the underlying local consistency of choice, or k-*Partition*-AC for short [4]. This technique divides a variable domain into disjoint domains, where each of them contains at most k elements. In the case of QCNs, these elements correspond to base relations. With respect to k-Partition-AC, the most common and preferred approach is dividing a domain into singleton sub-domains, which is the case where $k = 1$, otherwise many questions arise, such as what should the size of each sub-domain be, how should this size be fixed, and which elements should be considered for a given use case. Although having many questions to deal with is not necessarily bad in general, the most important aspect regarding 1-Partition-AC is that it offers the nice property that it is strictly stronger than singleton arc consistency (SAC) [9].

In this work, we adapt the aforementioned technique to QCNs using $\diamond_G$-consistency as our underlying local consistency of choice.[1] Given a QCN $\mathcal{N}$, enforcing this consistency for $k = 1$ will eliminate every base relation that is not $\blacklozenge_G$-consistent for some constraint in $\mathcal{N}$, but also every base relation that is not *supported* by some base relation in $\mathcal{N}$ through $\diamond_G$-consistency. We call this new local consistency $\blacklozenge_G^{\cup}$-consistency, and better explain it with a demonstrative example as follows. Consider the $\blacklozenge_G$-consistent QCN $\mathcal{N} = (V, C)$ of IA in Figure 3. We can see that the base relation d is $\blacklozenge_G$-consistent for $C(x_1, x_2)$, but it is not supported by any of the base relations that define constraint $C(x_1, x_3)$, namely, p and pi, through $\diamond_G$-consistency. In particular, by instatiating $C(x_1, x_3)$ with either p or pi and closing the respective QCN under $\diamond_G$-consistency, the base relation d is eliminated in $C(x_1, x_2)$. After eliminating the base relation d in $C(x_1, x_2)$, the revised QCN $\mathcal{N}$ becomes $\blacklozenge_G^{\cup}$-consistent.

Now we can formally define this consistency.

▶ **Definition 8.** Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, $\mathcal{N}$ is said to be $\blacklozenge_G^{\cup}$-consistent iff $\mathcal{N}$ 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 $(V, C') = \diamond_G(\mathcal{N}_{[u,u']/\{b'\}})$.

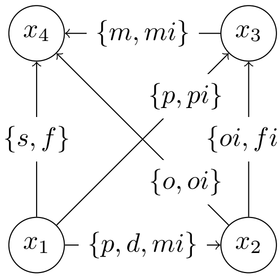
We prove the following result to be used in the sequel, which suggets that $\blacklozenge_G^{\cup}$-consistency can only eliminate unfeasible base relations:

▶ **Proposition 6.** *Let $\mathcal{N} = (V, C)$ be a* QCN*, $G = (V, E)$ a graph, and $b \in C(u, u')$ with $u, u' \in V$ a base relation. Then, if $\exists \{v, v'\} \in E$ such that $b \notin C'(u, u')$, where $(V, C') = \bigcup \{\diamond_G(\mathcal{N}_{[v,v']/\{b'\}}) \mid b' \in C(v, v')\}$, we have that b is an unfeasible base relation.*

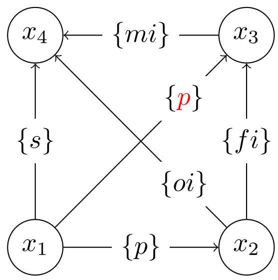
Proof. Let us assume that b is a feasible base relation. Then, by definition of feasible base relations there exists a scenario $\mathcal{S} = (V, C')$ of $\mathcal{N}$ such that $C'(u, u') = \{b\}$. Further, by the equivalence property of $\diamond_G$-consistency it holds that $\diamond_G(\mathcal{S}) = \mathcal{S}$ (as $\mathcal{S}$, being a scenario, is an atomic and satisfiable QCN and, hence, none of its base relations can be removed by application of $\diamond_G$-consistency). Thus, it follows that $\forall \{v, v'\} \in E$ we have that $b \in C''(u, u')$, where $(V, C'') = \diamond_G(\mathcal{N}_{[v,v']/C'(v,v')})$, as $\mathcal{S} \subseteq \mathcal{N}_{[v,v']/C'(v,v')}$ and, hence, $\diamond_G(\mathcal{S}) \subseteq \diamond_G(\mathcal{N}_{[v,v']/C'(v,v')})$ by the monotonicity property of $\diamond_G$-consistency. As $\mathcal{S} \subseteq \mathcal{N}$, it follows that $\forall \{v, v'\} \in E$ we have that $C'(v, v') \subseteq C(v, v')$ and, hence, that $\exists b' \in C(v, v')$ such that $b \in C'''(u, u')$, where $(V, C''') = \diamond_G(\mathcal{N}_{[v,v']/\{b'\}})$, by simply considering the base relation $b' \in C(v, v')$ to be the one of the singleton relation $C'(v, v')$ of $\mathcal{S}$. Therefore, by definition of operation $\cup$ with respect to QCNs we can derive that $\forall \{v, v'\} \in E$ it holds that $b \in C^*(u, u')$, where $(V, C^*) = \bigcup \{\diamond_G(\mathcal{N}_{[v,v']/\{b'\}}) \mid b' \in C(v, v')\}$, which concludes our proof by contraposition. ◀

We recall the following result to be used in one of our proofs later on:

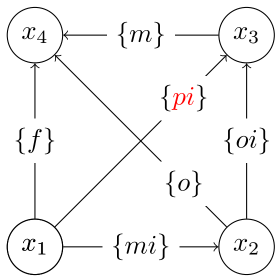
[1] The partitioning scheme can be combined with any local consistency or propagation technique. Here, the definition is restricted to $\diamond_G$-consistency as it is the most essential of local consistencies used for dealing with QCNs.



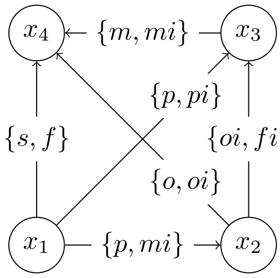
(a) A $\blacklozenge_G$-consistent QCN $\mathcal{N}$



(b) $\mathcal{N}_1 = \diamond_G(\mathcal{N}_{[x_1,x_3]/\{p\}})$



(c) $\mathcal{N}_2 = \diamond_G(\mathcal{N}_{[x_1,x_3]/\{pi\}})$



(d) $\mathcal{N}_1 \cup \mathcal{N}_2$

Figure 3 A $\blacklozenge_G$-consistent QCN $\mathcal{N}$ of IA along with a demonstration of how enforcing $\blacklozenge_G^{\cup}$-consistency can further eliminate base relations; here G is the complete graph on the set of variables of $\mathcal{N}$

▶ **Proposition 7** ([2])**.** *For any* QCN*s $\mathcal{N}_1$ and $\mathcal{N}_2$ on a set of variables V and any graph $G = (V, E)$, if $\mathcal{N}_1$ and $\mathcal{N}_2$ are $\blacklozenge_G$-consistent, then $(\mathcal{N}_1 \cup \mathcal{N}_2)$ is $\blacklozenge_G$-consistent as well.*

We note that the aforementioned result describes a sufficient property for proving dominance for a new consistency, but that property might not be necessary in general and, hence, does not solely follow from the well-behaveness of the consistency at hand. We prove the same property for $\blacklozenge_G^{\cup}$-consistency, to be used in what follows.

▶ **Proposition 8.** *For any* QCN*s $\mathcal{N}_1$ and $\mathcal{N}_2$ on a set of variables V and any graph $G = (V, E)$, if $\mathcal{N}_1$ and $\mathcal{N}_2$ are $\blacklozenge_G^{\cup}$-consistent, then $(\mathcal{N}_1 \cup \mathcal{N}_2)$ is $\blacklozenge_G^{\cup}$-consistent as well.*

Proof. Let $\mathcal{N}_1 = (V, C_1)$, $\mathcal{N}_2 = (V, C_2)$, $(\mathcal{N}_1 \cup \mathcal{N}_2) = (V, C)$, $v, v' \in V$ be two variables, and $b \in C(v, v')$ a base relation. We only need to consider the case where $b \in C_1(v, v')$, as the case where $b \in C_2(v, v')$ is symmetric. Since $\mathcal{N}_1$ is $\blacklozenge_G^{\cup}$-consistent, we have that $\mathcal{N}_1$ is $\blacklozenge_G$-consistent and $\forall\{u, u'\} \in E$ there exists $b' \in C_1(u, u')$ such that $b \in C'_1(v, v')$, where $(V, C'_1) = \diamond_G(\mathcal{N}_{1[u,u']/\{b'\}})$, by definition of $\blacklozenge_G^{\cup}$-consistency. In addition, we have that $(\mathcal{N}_1 \cup \mathcal{N}_2)$ is $\blacklozenge_G$-consistent by Proposition 7. As $\mathcal{N}_1 \subseteq (\mathcal{N}_1 \cup \mathcal{N}_2)$, we have that $\mathcal{N}_{1[u,u']/\{b'\}} \subseteq (\mathcal{N}_1 \cup \mathcal{N}_2)_{[u,u']/\{b'\}}$ $\forall\{u, u'\} \in E$ and $\forall b' \in C_1(u, u')$. Thus, we have that $\diamond_G(\mathcal{N}_{1[u,u']/\{b'\}}) \subseteq \diamond_G((\mathcal{N}_1 \cup \mathcal{N}_2)_{[u,u']/\{b'\}})$ $\forall\{u, u'\} \in E$ and $\forall b' \in C_1(u, u')$ by the monotonicity property of

$\diamond_G$-consistency. From that we can deduce that $\forall\{u,u'\} \in E$ there exists $b' \in C(u,u')$ such that $b \in C'(v,v')$, where $(V,C') = \diamond_G((\mathcal{N}_1 \cup \mathcal{N}_2)_{[u,u']/\{b'\}})$. Hence, by the assumption that $\mathcal{N}_1$ and $\mathcal{N}_2$ are $\blacklozenge^{\cup}_G$-consistent, we have proved that $(\mathcal{N}_1 \cup \mathcal{N}_2)$ is $\blacklozenge^{\cup}_G$-consistent as well. ◀

Next, we arrive to one of our main results in this work.

▶ **Theorem 2.** *We have that $\blacklozenge^{\cup}_G$-consistency is well-behaved.*

Proof. (Dominance) From Proposition 8 we can assert that, for any QCN $\mathcal{N} = (V,C)$ and any graph $G = (V,E)$, there exists a unique $\blacklozenge^{\cup}_G$-consistent QCN $\bigcup\{\mathcal{N}' \mid \mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{N}'$ is $\blacklozenge^{\cup}_G$-consistent$\}$, which by its definition is the largest (w.r.t. $\subseteq$) $\blacklozenge^{\cup}_G$-consistent sub-QCN of $\mathcal{N}$ and, hence, the closure of $\mathcal{N}$ under $\blacklozenge^{\cup}_G$-consistency. (Equivalence) Let $\mathcal{N} = (V,C)$ be a QCN, $G = (V,E)$ a graph, and $\mathcal{N}' = (V,C')$ the QCN where $\forall v,v' \in V$ and $\forall b \in \mathsf{B}$ we have that $b \in C'(v,v')$ iff there exists a solution σ of $\mathcal{N}$ such that $(\sigma(v),\sigma(v')) \in b$. Clearly, $\mathcal{N}'$ is a sub-QCN of $\mathcal{N}$ and it is necessarily $\blacklozenge^{\cup}_{K_V}$-consistent (where K_V denotes the complete graph on the set of variables V of $\mathcal{N}$), as by Proposition 6 we have that the application of $\blacklozenge^{\cup}_G$-consistency on any QCN (V,C) w.r.t. any graph $G = (V,E)$ can only remove unfeasible base relations, and not feasible ones. It follows that $\mathcal{N}' \subseteq \blacklozenge^{\cup}_G(\mathcal{N}) \subseteq \mathcal{N}$ and, as such, $\blacklozenge^{\cup}_G(\mathcal{N})$ and $\mathcal{N}$ share the same set of solutions. (Idempotence) Let $\mathcal{N} = (V,C)$ be a QCN, and $G = (V,E)$ a graph. Then, $\blacklozenge^{\cup}_G(\mathcal{N})$ is $\blacklozenge^{\cup}_G$-consistent. Now, by dominance of $\blacklozenge^{\cup}_G$-consistency the largest $\blacklozenge^{\cup}_G$-consistent sub-QCN of $\blacklozenge^{\cup}_G(\mathcal{N})$ is itself and, hence, $\blacklozenge^{\cup}_G(\blacklozenge^{\cup}_G(\mathcal{N})) = \blacklozenge^{\cup}_G(\mathcal{N})$. (Monotonicity) Let $\mathcal{N} = (V,C)$ and $\mathcal{N}' = (V,C')$ be two QCNs such that $\mathcal{N}' \subseteq \mathcal{N}$, and $G = (V,E)$ a graph. As $\mathcal{N}' \subseteq \mathcal{N}$, we have that $\blacklozenge^{\cup}_G(\mathcal{N}')$ is a $\blacklozenge^{\cup}_G$-consistent sub-QCN of $\mathcal{N}$. In addition, by dominance of $\blacklozenge^{\cup}_G$-consistency we can assert that $\blacklozenge^{\cup}_G(\mathcal{N})$ is the largest $\blacklozenge^{\cup}_G$-consistent sub-QCN of $\mathcal{N}$. Therefore, we have that $\blacklozenge^{\cup}_G(\mathcal{N}') \subseteq \blacklozenge^{\cup}_G(\mathcal{N})$. ◀

We prove the following general result regarding the pruning capability of $\blacklozenge_G$-consistency in comparison with that of $\blacklozenge^{\cup}_G$-consistency:

▶ **Proposition 9.** *We have that $\blacklozenge^{\cup}_G$-consistency $\triangleright$ $\blacklozenge_G$-consistency.*

Proof. By definition of $\blacklozenge^{\cup}_G$-consistency, we have that $\blacklozenge^{\cup}_G$-consistency $\trianglerighteq$ $\blacklozenge_G$-consistency, since, for any graph $G = (V,E)$, any QCN (V,C) that is $\blacklozenge^{\cup}_G$-consistent is already $\blacklozenge_G$-consistent. To prove strictness we use an example as follows. Consider the QCN $\mathcal{N} = (V,C)$ of Figure 3. The reader can verify that $\mathcal{N}$ is $\blacklozenge_G$-consistent, as we have that b is $\blacklozenge_G$-consistent for $C(v,v')$ $\forall\{v,v'\} \in E$ and $\forall b \in C(v,v')$. However, we have that $d \notin C'(x_1,x_2)$, where $(V,C') = \bigcup\{\diamond_G(\mathcal{N}_{[x_1,x_3]/\{b'\}}) \mid b' \in C(x_1,x_3)\}$, as demonstrated in the figure. In detail, $\diamond_G(\mathcal{N}_{[x_1,x_3]/\{p\}}) \cup \diamond_G(\mathcal{N}_{[x_1,x_3]/\{pi\}})$ is a QCN such that d is not among the base relations that define the constraint on variables x_1 and x_2. Thus, $\blacklozenge^{\cup}_G$-consistency does not hold in $\mathcal{N}$. ◀

The next result follows trivially:

▶ **Proposition 10.** *We have that $\blacklozenge^{\cup}_G$-consistency $\triangleright$ $\diamond_G$-consistency.*

Proof. A direct consequence of Propositions 2 and 9 and the transitivity of $\triangleright$. ◀

Finally, we introduce the following result that identifies the case where $\blacklozenge_G$-consistency and $\blacklozenge^{\cup}_G$-consistency are equivalent:

▶ **Proposition 11.** *Let $\mathcal{A}$ be a subclass of relations of a relation algebra with the property that for any* QCN *$\mathcal{N} = (V,C)$ over $\mathcal{A}$ there exists a graph $G = (V,E)$ such that, if $\diamond_G(\mathcal{N})$ is not trivially inconsistent, then $\mathcal{N}$ is satisfiable. Then, for any such $\mathcal{N}$ and G, we have that $\blacklozenge_G$-consistency $\equiv$ $\blacklozenge^{\cup}_G$-consistency.*

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

---

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

begin
  $\mathcal{N} \leftarrow \mathsf{PWC}(\mathcal{N}, G)$;
  $Q \leftarrow E$;
  while $Q \neq \emptyset$ **do**
    $\{v, v'\} \leftarrow Q.pop()$;
    $(V, C') \leftarrow \bot^V$;
    foreach $b \in C(v, v')$ **do**
      $(V, C') \leftarrow (V, C') \cup \mathsf{PWC}(\mathcal{N}_{[v,v']/\{b\}}, G, \{\{v, v'\}\})$;
    if $(V, C') \subset \mathcal{N}$ **then**
      foreach $\{u, u'\} \in E \mid C'(u, u') \subset C(u, u')$ **do**
        $C(u, u') \leftarrow C'(u, u')$;
        $C(u', u) \leftarrow C'(u', u)$;
      $Q \leftarrow E$;
  return $\mathcal{N}$;

Proof. We first prove that, if $\mathcal{N}$ is $\blacklozenge_G$-consistent, then $\mathcal{N}$ is also $\blacklozenge_G^{\cup}$-consistent. By Proposition 5 we have that $\forall \{u, v\} \in E$ and $\forall b \in C(u, v)$ the base relation b is feasible. In addition, by the equivalence property of $\blacklozenge_G^{\cup}$-consistency we have that the application of $\blacklozenge_G^{\cup}$-consistency on $\mathcal{N}$ can only remove unfeasible base relations and, hence, that $\blacklozenge_G^{\cup}(\mathcal{N}) = \mathcal{N}$, as every base relation $b \in C(u, v)$ $\forall \{u, v\} \in E$ is feasible. The proof that, if $\mathcal{N}$ is $\blacklozenge_G^{\cup}$-consistent, then $\mathcal{N}$ is also $\blacklozenge_G$-consistent, follows directly from the definition of $\blacklozenge_G^{\cup}$-consistency. ◀

A hasty reading of Proposition 11 might give the impression that one should always opt to apply $\blacklozenge_G$-consistency for the cases where the considered QCN and the graph G satisfy the prerequisites detailed in that proposition, as $\blacklozenge_G$-consistency, being a weaker consistency than $\blacklozenge_G^{\cup}$-consistency in general, should be "easier" to apply. However, as we will demonstrate in our experimental section, $\blacklozenge_G^{\cup}$-consistency is faster to apply. To give an intuition, any well-structured algorithm that will try to enforce $\blacklozenge_G^{\cup}$-consistency in a given QCN for some graph G, will inescapably make better use of the singleton checks than the respective algorithm for enforcing $\blacklozenge_G$-consistency. This is because the former algorithm will exploit the singleton checks (by the very definition of $\blacklozenge_G^{\cup}$-consistency) to *proactively* eliminate certain base relations that are unfeasible and, hence, possibly not $\blacklozenge_G$-consistent for the corresponding constraints.

5 An algorithm for achieving $\blacklozenge_G^{\cup}$-consistency

In this section, we propose an algorithm for efficiently applying $\blacklozenge_G^{\cup}$-consistency on a given QCN $\mathcal{N}$, called $\mathsf{PSWC}^{\cup}$ ($\cup$-*collective* partial singleton closure under weak composition) and presented in Algorithm 1. This algorithm builds on the algorithm for efficiently achieving $\blacklozenge_G$-consistency, called PSWC (partial singleton closure under weak composition) and presented in Algorithm 2, which in itself is an advancement of the respective algorithm for enforcing $\blacklozenge_G$-consistency that is presented in [2]; we explain as follows. We use a queue in both of our algorithms that is initialized with all of the edges of a given graph G that correspond to constraints of a given QCN $\mathcal{N}$. In addition, this queue is filled with all of the aforementioned edges whenever any of the constraints of $\mathcal{N}$ is revised, i.e., whenever a base relation is removed. This operation is equivalent to introducing a break statement in the algorithm of [2] whenever a singleton check fails and, hence, a constraint is revised, forcing the inner loop in

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

in : A QCN $\mathcal{N} = (V, C)$, and a graph $G = (V, E)$.
out : A sub-QCN of $\mathcal{N}$.
1 **begin**
2     $\mathcal{N} \leftarrow$ PWC($\mathcal{N}, G$);
3     $Q \leftarrow E$;
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             $(V, C') \leftarrow (V, C') \cup$ PWC($\mathcal{N}_{[v,v']/\{b\}}, G, \{\{v, v'\}\}$);
9   **if** $C'(v, v') \subset C(v, v')$ **then**
10             $C(v, v') \leftarrow C'(v, v')$;
11             $C(v', v) \leftarrow C'(v', v)$;
12             $Q \leftarrow E$;
13   **return** $\mathcal{N}$;

Algorithm 3: PWC($\mathcal{N}, G, e \leftarrow \emptyset$)

in : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$, and optionally a set e such that $e \subseteq E$.
out : A sub-QCN of $\mathcal{N}$.
1 **begin**
2     $Q \leftarrow (e$ **if** $e \neq \emptyset$ **else** $E)$;
3   **while** $Q \neq \emptyset$ **do**
4         $\{v, v'\} \leftarrow Q.pop()$;
5   **foreach** $v'' \in V \mid \{v, v''\}, \{v', v''\} \in E$ **do**
6             $r \leftarrow C(v, v'') \cap (C(v, v') \diamond C(v', v''))$;
7   **if** $r \subset C(v, v'')$ **then**
8                 $C(v, v'') \leftarrow r$;
9                 $C(v'', v) \leftarrow r^{-1}$;
10                 $Q \leftarrow Q \cup \{\{v, v''\}\}$;
11             $r \leftarrow C(v'', v') \cap (C(v'', v) \diamond C(v, v'))$;
12   **if** $r \subset C(v'', v')$ **then**
13                 $C(v'', v') \leftarrow r$;
14                 $C(v', v'') \leftarrow r^{-1}$;
15                 $Q \leftarrow Q \cup \{\{v'', v'\}\}$;
16   **return** $\mathcal{N}$;

that algorithm to stop and using the outer loop to initiate singleton checks in a fresh QCN. We have found this tactic to perform much better in practice, cutting down on the number of constraint checks by around 20%. Further, the use of a queue allows for prioritizing certain edges, a strategy which is in line with similar techniques used in the algorithm for enforcing $\overset{\diamond}{_G}$-consistency [34, 27, 16], but this is something that we have not yet explored and retain for future work. As we will also remind the reader in the experimental evaluation to follow, we use a simple FIFO (first-in, first-out) queue for our algorithms. For the sake of completeness, we also present the state-of-the-art algorithm for applying $\overset{\diamond}{_G}$-consistency on a given QCN, called PWC (partial closure under weak composition), which is utilized as a subroutine by both PSWC$^{\cup}$ and PSWC (see Algorithm 3).

The difference between algorithms PSWC$^{\cup}$ and PSWC lies solely in the way that they exploit singleton checks. In particular, note the difference between the conditions in line 9 of both algorithms; $\mathsf{PSWC}^{\cup}$ will bring up all edges in the queue for revising the entire QCN even when the constraint at hand was not revised, but another constraint somewhere in the QCN was, whereas PSWC will keep its focus solely on the constraint at hand. This is due to the fact that algorithm $\mathsf{PSWC}^{\cup}$ will use a single singleton check to eliminate base relations anywhere in the network, and not just in the constraint at hand as algorithm PSWC does. Before proving the correctness of algorithm $\mathsf{PSWC}^{\cup}$, we recall the following result regarding the correctness of algorithm PSWC:

▶ **Proposition 12** (cf. [2, 8]). *Given a* QCN $\mathcal{N} = (V, C)$ *of a relation algebra and a graph* $G = (V, E)$*, we have that algorithm* PSWC *terminates and returns* $\blacklozenge_G(\mathcal{N})$*.*

Now, we prove that algorithm $\mathsf{PSWC}^{\cup}$ is complete for applying $\blacklozenge_G^{\cup}$-consistency on a given $\mathcal{N} = (V, C)$ for a given graph $G = (V, E)$. Due to space limitations, an intuitive proof is provided, which however manages to explain the overall functionallity of algorithm $\mathsf{PSWC}^{\cup}$ in sufficient detail.

▶ **Theorem 3.** *Given a* QCN $\mathcal{N} = (V, C)$ *of a relation algebra and a graph* $G = (V, E)$*, we have that algorithm* $\mathsf{PSWC}^{\cup}$ *terminates and returns* $\blacklozenge_G^{\cup}(\mathcal{N})$*.*

Proof. (Intuition) It is easy to see that lines 9–12 in Algorithm 1 perform a superset of the operations performed in lines 9–11 in Algorithm 2. Thus, by Proposition 12 we know that given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, algorithm $\mathsf{PSWC}^{\cup}$ applies the set of operations required to make $\mathcal{N}$ $\blacklozenge_G$-consistent. We need to show that the rest of the operations maintain $\blacklozenge_G$-consistency and further achieve $\blacklozenge_G^{\cup}$-consistency. With respect to that, it is again easy to see that algorithm $\mathsf{PSWC}^{\cup}$ enforces exactly the conditions specified in Proposition 6 and, hence, removes the (unfeasible) base relations required to make $\mathcal{N}$ $\blacklozenge_G^{\cup}$-consistent. Further, since the algorithm will only terminate when b is guaranteed to have become $\blacklozenge_G$-consistent for $C(u, v)$ $\forall\{u, v\} \in E$ and $\forall b \in C(u, v)$ and no constraint is further revised to additionally achieve $\blacklozenge_G^{\cup}$-consistency, we can conclude that algorithm $\mathsf{PSWC}^{\cup}$ correctly applies $\blacklozenge_G^{\cup}$-consistency on $\mathcal{N}$. ◀

Time complexity analysis

Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, we have that algorithm $\mathsf{PSWC}^{\cup}$ applies $\blacklozenge_G^{\cup}$-consistency on $\mathcal{N}$ in $O(\Delta \cdot |E|^3 \cdot \mathsf{B}^3)$ time, where Δ is the maximum vertex degree of graph G. In particular, algorithm PWC is executed $O(|E| \cdot |B|)$ times every time a constraint is revised, and such a constraint revision can occur $O(|E| \cdot |B|)$ times. Further, we note that the unification operations that take place in line 8 of the algorithm are handled in $O(|E| \cdot |B|)$ time in total, as we keep track of the constraints that are revised by algorithm PWC and we can have a total of $O(|E| \cdot |B|)$ constraint revisions. The same argument holds for the operations that take place in lines 9–12 of the alorithm. (These details are not included in the algorithm to allow for a more compact representation.) Now, by taking into account the worst-case time complexity of algorithm PWC, which is $O(\Delta \cdot |E| \cdot \mathsf{B})$ [7], a worst-case time complexity of $O(\Delta \cdot |E|^3 \cdot \mathsf{B}^3)$ can be obtained for algorithm $\mathsf{PSWC}^{\cup}$; this is also the worst-case time complexity of algorithm PSWC [2]. It is important to note that we cannot utilize the incremental functionality of algorithm PWC (see Theorem 1 in [12, Section 3] and the surrounding text) to obtain a better bound for our algorithm, as the singleton checks are perfomed independently[2] of one another.

[2] To be more precise, the unification operations that take place in line 8 of the algorithm do not provide the level of interdependency required to tap into the incrementality of PWC.

Table 1 Evaluation with random IA networks of model S(n = 70, l = 6.5, d)

	min		μ		max		σ	
d	PSWC	$\text{PSWC}^{\cup}$	PSWC	$\text{PSWC}^{\cup}$	PSWC	$\text{PSWC}^{\cup}$	PSWC	$\text{PSWC}^{\cup}$
10	$\frac{3.19s}{9k}$	$\frac{3.11s}{9k}$	$\frac{4.25s}{14k}$	$\frac{4.02s}{13k}$	$\frac{5.63s}{22k}$	$\frac{5.43s}{21k}$	$\frac{0.50s}{2k}$	$\frac{0.52s}{2k}$
12	$\frac{5.79s}{9k}$	$\frac{5.65s}{9k}$	$\frac{7.82s}{14k}$	$\frac{7.51s}{14k}$	$\frac{11.85s}{22k}$	$\frac{10.88s}{22k}$	$\frac{1.38s}{2k}$	$\frac{1.18s}{2k}$
14	$\frac{9.92s}{12k}$	$\frac{9.63s}{12k}$	$\frac{13.81s}{19k}$	$\frac{13.08s}{18k}$	$\frac{21.26s}{38k}$	$\frac{19.24s}{28k}$	$\frac{2.68s}{5k}$	$\frac{2.40s}{4k}$
16	$\frac{14.14s}{18k}$	$\frac{13.91s}{18k}$	$\frac{30.19s}{31k}$	$\frac{27.74s}{29k}$	$\frac{96.24s}{61k}$	$\frac{99.56s}{62k}$	$\frac{14.20s}{8k}$	$\frac{14.95s}{8k}$
18	$\frac{21.39s}{26k}$	$\frac{20.94s}{26k}$	$\frac{56.16s}{45k}$	$\frac{53.17s}{44k}$	$\frac{154.14s}{88k}$	$\frac{149.98s}{88k}$	$\frac{22.92s}{13k}$	$\frac{21.22s}{12k}$
20	$\frac{54.42s}{39k}$	$\frac{52.82s}{36k}$	$\frac{100.22s}{58k}$	$\frac{89.32s}{55k}$	$\frac{192.68s}{108k}$	$\frac{188.54s}{100k}$	$\frac{30.68s}{14k}$	$\frac{26.08s}{12k}$
22	$\frac{19.66s}{48k}$	$\frac{17.09s}{46k}$	$\frac{42.51s}{67k}$	$\frac{39.83s}{63k}$	$\frac{86.09s}{108k}$	$\frac{75.55s}{108k}$	$\frac{16.67s}{14k}$	$\frac{15.19s}{14k}$

6 Experimental evaluation

We evaluated the performance of an implementation of algorithm $\text{PSWC}^{\cup}$, against an implementation of the algorithm for enforcing partial $\diamond_G$-consistency that was presented here, namely, PSWC, with a varied dataset of arbitrary QCNs of IA.

Technical specifications. The evaluation was carried out on a computer with an Intel Core i5-6200U processor (which has a max frequency of 2.7 GHz per CPU core under turbo mode[3]), 8 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; the code is available upon request. Only one CPU core was used.

Datasets and measures. We considered random IA networks generated by the S(n, l, d) model [34]. This model can randomly generate satisfiable QCNs of n variables with an average number l of base relations per non-universal constraint and an average degree d for the corresponding constraint graphs. Further, this model is typically used in the evaluation of algorithms dealing with problems associated with QCNs, with an emphasis on the minimal labelling problem [34, 2, 23]. We generated 30 QCNs of IA of n = 70 variables with $l = |B|/2 = 6.5$ base relations per non-universal constraint on average for all values of d ranging from 10 to 22 with a step of 2 (a typical range for evaluating related algorithms [2]); hence, we considered a total of 210 QCNs of IA. Finally, the *maximum cardinality search* algorithm [32] was used to obtain a triangulation of the constraint graph of a given QCN. Notice that, with respect to our evaluation, any kind of graphs would have been adequate (even complete ones), as they would have affected all involved algorithms proportionally and would not have qualitatively distorted the obtained results; however, the choice of chordal graphs was more reasonable given their extensive use in the recent literature [31].

Our evaluation involved two measures, which we describe as follows. The first measure considers the number of *constraint checks per base relation removals* performed by an algorithm for enforcing the respective local consistency. Given a QCN $\mathcal{N} = (V, C)$ and

[3] Turbo mode was maintained throughout the experimental evaluation by staying well within thermal design power (TDP) limit.

three variables $v_i, v_k, v_j \in V$, a constraint check occurs when we compute the relation $r = C(v_i, v_j) \cap (C(v_i, v_k) \diamond C(v_k, v_j))$ and check if $r \subset C(v_i, v_j)$, so that we can propagate its constrainedness if that condition is satisfied. Weak compositions that yield relation B are disregarded. The second measure concerns the CPU time and is strongly correlated with the first one, as the run-time of any proper implementation of an algorithm for enforcing a local consistency should, in principle, rely mainly on the number of constraint checks performed.

Results. The results of our experimental evalualation are detailed in Table 1, where a fraction $\frac{x}{y}$ denotes that an approach required x seconds of CPU time and performed y constraint checks per base relation removals on average per dataset of networks during its operation. In short, the advantage of $\mathsf{PSWC}^{\cup}$ over PSWC is clear across all parameters and for all settings and corresponds to around 10%. This is a promising result in terms of achieving a new stronger local consistency faster than what was possible to date even when considering a weaker local consistency. Further, we recall to the reader that we used a simple FIFO queue for our implementation; it would be interesting to explore prioritizing edges corresponding to constraints that are revised at any given step of the execution. We retain a more thorough experimental evaluation, which will also include the effect of our new stronger local consistency on backtracking-based algorithms, for future work. Here, we opted to maintain a simple configuration for our algorithms in order to obtain a first pure comparison that will serve as a basis for further evaluation.

7 Conclusion and future work

Partial singleton closure under weak composition, or partial $\blacklozenge$-consistency for short, is a local consistency that ensures that each base relation of each of the constraints of a qualitative constraint network can define a singleton relation in the corresponding partial closure of that network under weak composition, or in its corresponding partial $\diamond$-consistent subnetwork for short. This local consistency is essential for approximating satisfiability of qualitative constraints networks, and has been shown to play a crucial role in tackling the minimal labeling problem of a qualitative constraint network in particular, which is the problem of finding the strongest implied constraints of that network. In this paper, we proposed a stronger local consistency that couples $\blacklozenge$-consistency with the idea of collectively deleting certain unfeasible base relations by exploiting singleton checks. Further, we proposed an efficient algorithm for enforcing this new consistency that, given a qualitative constraint network, performs fewer constraint checks than the respective algorithm for enforcing partial $\blacklozenge$-consistency in that network. We formally proved certain properties of our new local consistency, and motivated its usefulness through demonstrative examples and a preliminary experimental evaluation with qualitative constraint networks of Interval Algebra.

There are several directions for future work. Regarding the algorithm that enforces our new consistency, we would like to explore queuing strategies such that the singleton checks are applied in a more fruitful manner. In particular, it would make sense to prioritize certain singleton checks that are more likely to eliminate base relations anywhere in the network at hand, because this could unveil certain inconsistencies faster, but also lead to fewer constraint checks overall. Such strategies have been used in the case of partial $\diamond$-consistency [34, 27, 16]. Further, regarding the new local consistency itself, we would like to define a weaker variant of it that considers singleton checks in the neighborhood of the constraint in question, instead of the entire network. Early experiments in this direction have shown really promising results with respect to constraint satisfaction problems, which is due to the fact that constraint revisions tend to propagate themselves to just neighboring constraints [36].

Acknowledgements

We would like to thank the reviewers for their helpful comments. Michael Sioutis wishes to acknowledge support from the MoveCare project (ID: 732158), which is funded by the European Commission under the H2020 framework programme for research and innovation. Finally, parts of this work were conducted during the postdoctoral research of Michael Sioutis in Artois University.

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 Philippe Balbiani, Jean-François Condotta, and Luis Fariñas del Cerro. Tractability Results in the Block Algebra. *J. Log. Comput.*, 12:885–909, 2002.
4 Hachemi Bennaceur and Mohamed-Salah Affane. Partition-k-AC: An Efficient Filtering Technique Combining Domain Partition and Arc Consistency. In *CP*, 2001.
5 Christian Bessière. Arc-Consistency and Arc-Consistency Again. *Artif. Intell.*, 65:179–190, 1994.
6 Mehul Bhatt, Hans W. Guesgen, Stefan Wölfl, and Shyamanta Moni Hazarika. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. *Spatial Cognition & Computation*, 11:1–14, 2011.
7 Assef Chmeiss and Jean-François Condotta. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI*, 2011.
8 Jean-François Condotta and Christophe Lecoutre. A Class of df-Consistencies for Qualitative Constraint Networks. In *KR*, 2010.
9 Romuald Debruyne and Christian Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *IJCAI*, 1997.
10 Frank Dylla, Till Mossakowski, Thomas Schneider, and Diedrich Wolter. Algebraic Properties of Qualitative Spatio-Temporal Calculi. In *COSIT*, 2013.
11 Andrew U. Frank. Qualitative Spatial Reasoning with Cardinal Directions. In *ÖGAI*, 1991.
12 Alfonso Gerevini. Incremental qualitative temporal reasoning: Algorithms for the Point Algebra and the ORD-Horn class. *Artif. Intell.*, 166:37–80, 2005.
13 Martin Charles Golumbic and Ron Shamir. Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach. *J. ACM*, 40:1108–1133, 1993.
14 Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *KR*, 2012.
15 Jinbo Huang, Jason Jingshi Li, and Jochen Renz. Decomposition and tractability in qualitative spatial and temporal reasoning. *Artif. Intell.*, 195:140–164, 2013.
16 Peter B. Ladkin and Alexander Reinefeld. Fast Algebraic Methods for Interval Constraint Problems. *Ann. Math. Artif. Intell.*, 19:383–411, 1997.
17 Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both. On redundant topological constraints. *Artif. Intell.*, 225:51–76, 2015.
18 Gérard Ligozat. Reasoning about cardinal directions. *J. Vis. Lang. Comput.*, 9:23–44, 1998.
19 Gérard Ligozat and Jochen Renz. What Is a Qualitative Calculus? A General Framework. In *PRICAI*, 2004.
20 Zhiguo Long and Sanjiang Li. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*, 2015.

21 Zhiguo Long, Michael Sioutis, and Sanjiang Li. Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks. In *IJCAI*, 2016.

22 Carsten Lutz and Maja Milicic. A Tableau Algorithm for DLs with Concrete Domains and GCIs. *J. Autom. Reasoning*, 38:227–259, 2007.

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

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

25 Jochen Renz. A Canonical Model of the Region Connection Calculus. *J. Appl. Non-Classical Logics*, 12:469–494, 2002.

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

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

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

29 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.

30 Michael Sioutis, Sanjiang Li, and Jean-François Condotta. Efficiently Characterizing Non-Redundant Constraints in Large Real World Qualitative Spatial Networks. In *IJCAI*, 2015.

31 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.

32 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.

33 Alfred Tarski. On the calculus of relations. *J. Symb. Log.*, 6:73–89, 1941.

34 Peter van Beek and Dennis W. Manchak. The design and experimental analysis of algorithms for temporal reasoning. *J. Artif. Intell. Res.*, 4:1–18, 1996.

35 Marc B. Vilain and Henry A. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *AAAI*, 1986.

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