A Lazy Algorithm to Efficiently Approximate Singleton Path Consistency for Qualitative Constraint Networks

Michael Sioutis*, Anastasia Paparrizou†, and Jean-François Condotta†
*Örebro universitet, MPI@AASS, Örebro, Sweden
michael.sioutis@oru.se
†Université d'Artois, CRIL-CNRS UMR 8188, Lens, France
{paparrizou,condotta}@cril.fr

***Abstract*—Partial singleton (weak) path consistency, or partial $\blacklozenge$-consistency, for a qualitative constraint network, ensures that the process of instantiating any constraint of that network with any of its base relations b and enforcing partial (weak) path consistency, or partial $\diamond$-consistency, in the updated network, yields a partially $\diamond$-consistent subnetwork where the respective constraint is still defined by b. This local consistency is essential for helping to decide the satisfiability of challenging qualitative constraint networks and has been shown to play a crucial role in tackling more demanding problems associated with a given qualitative constraint network, such as the problem of minimal labeling. One of the main downsides to using partial $\blacklozenge$-consistency, is that it is computationally expensive to enforce in a given qualitative constraint network, as, despite being a local consistency in principle, it retains a *global* scope of the network at hand. In this paper, we propose a *lazy* algorithm that restricts the singleton checks associated with partial $\blacklozenge$-consistency to constraints that are likely to lead to the removal of a base relation upon their propagation. A key feature of this algorithm is that it collectively eliminates certain unfeasible base relations by exploiting singleton checks. Further, we show that the closure that is obtained by our algorithm is incomparable to the one that is entailed by partial $\blacklozenge$-consistency and non-unique in general. We demonstrate the efficiency of our algorithm via an experimental evaluation with random Interval Algebra networks from the phase transition region of two separate models and, moreover, show that it can exhibit very similar pruning capability for such networks to the one of an algorithm for enforcing partial $\blacklozenge$-consistency.**

***Index Terms*—Qualitative constraint-based reasoning, spatial and temporal relations, partial $\blacklozenge$-consistency, approximation.**

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 [1]. QSTR abstracts from numerical quantities of space and time by using qualitative descriptions instead, e.g., *precedes*, *contains*, and *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*) [2]. An example of such a qualitative constraint language is Interval Algebra (IA), introduced by Allen [3]. 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.

Partial $\diamond$-consistency is the basic local consistency used in the literature for defining practical algorithms to efficiently reason with QCNs [4], [5], [6], [7], [8]. Given a QCN $\mathcal{N}$ and a graph G, partial $\diamond$-consistency with respect to G, denoted by ${}^{\diamond}_{G}$*-consistency*, entails (weak) consistency for all triples of variables in $\mathcal{N}$ that correspond to three-vertex cycles (triangles) in G. The literature suggests that ${}^{\diamond}_{G}$-consistency alone is sufficient in most cases to guarantee that a solution for a given QCN, should it exist, can be efficiently obtained (see also [9]). However, for the more challenging QCNs and the more demanding problems, such as the problem of *minimal labeling*, which calls for determining the strongest implied constraints in a given QCN, a stronger local consistency is typically employed that builds upon ${}^{\diamond}_{G}$-consistency, called *singleton* ${}^{\diamond}_{G}$*-consistency* and denoted by ${}^{\blacklozenge}_{G}$*-consistency* [4]. Specifically, given a QCN $\mathcal{N}$ and a graph G, it is said that ${}^{\blacklozenge}_{G}$-consistency holds on $\mathcal{N}$ if and only if after instantiating any constraint of $\mathcal{N}$ with any of its base relations b and closing the updated network $\mathcal{N}$ under ${}^{\diamond}_{G}$-consistency, the corresponding constraint in the ${}^{\diamond}_{G}$-consistent subnetwork of $\mathcal{N}$ remains defined by b.

Unfortunately, ${}^{\blacklozenge}_{G}$-consistency is computationally very expensive to enforce in a given QCN [9]; this was also verified in the experimental evaluation that took place in the context of this paper (presented in Section 4) where a state-of-the-art algorithm for enforcing ${}^{\blacklozenge}_{G}$-consistency was employed.

Therefore, we make the following contributions: (i) we propose a *lazy* algorithm to approximate $\blacklozenge_G$-consistency that restricts the singleton checks associated with $\blacklozenge_G$-consistency to constraints that are likely to lead to the removal of a base relation upon their propagation in a given QCN, and that collectively eliminates certain unfeasible base relations by exploiting singleton checks; (ii) we thoroughly study the theoretical properties of this algorithm and show, among other things, that the closure that it computes is incomparable to the one that is achieved by $\blacklozenge_G$-consistency and non-unique in general, and finally (iii) we demonstrate the efficiency of our algorithm via an experimental evaluation with random IA networks from the phase transition region of two separate models and, moreover, show that it can exhibit very similar pruning capability for such networks to the one of an algorithm for enforcing $\blacklozenge_G$-consistency.

The remainder of the paper is structured as follows. In Section 2 we give some preliminaries on qualitative constraint-based reasoning, with an emphasis on $\blacklozenge_G$-consistency. In Section 3 we introduce and thoroughly study the algorithm for approximating $\blacklozenge_G$-consistency, and in Section 4 we evaluate it against the state-of-the-art algorithm for achieving $\blacklozenge_G$-consistency. Finally, in Section 5 we conclude the paper and sketch some 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* [2]. 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 symbol $\diamond$ [2]. 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 strongest (i.e., smallest) 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 $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r, b' \in r'\}$. Finally, we assume that $r \diamond \mathsf{B} = \mathsf{B}$ for every $r \in 2^{\mathsf{B}}$.

As an illustration, consider the well known qualitative temporal constraint language of Interval Algebra (IA) introduced by Allen [3]. 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 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 [10]. 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* [11]. Other notable and well known qualitative spatial and temporal constraint languages include Point Algebra [12], Cardinal Direction Calculus [13], [14], and Block Algebra [15].

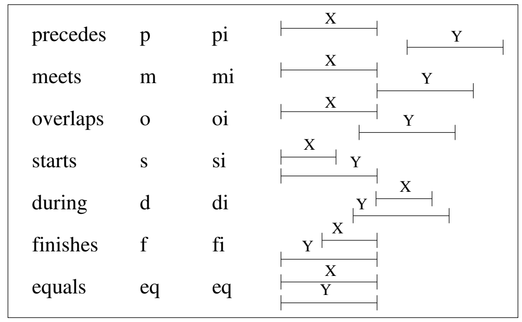


Fig. 1: The base relations of IA

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 [16].

Proposition 1 ([17])**.** *The languages of Point Algebra, Cardinal Direction Calculus, Interval Algebra, Block Algebra, and* RCC-*8 are each a relation algebra with the algebraic structure* $(2^{\mathsf{B}}$, Id, $\diamond$, $^{-1}$, $^{\complement}$, $\cup)$.

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

Definition 1. 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.

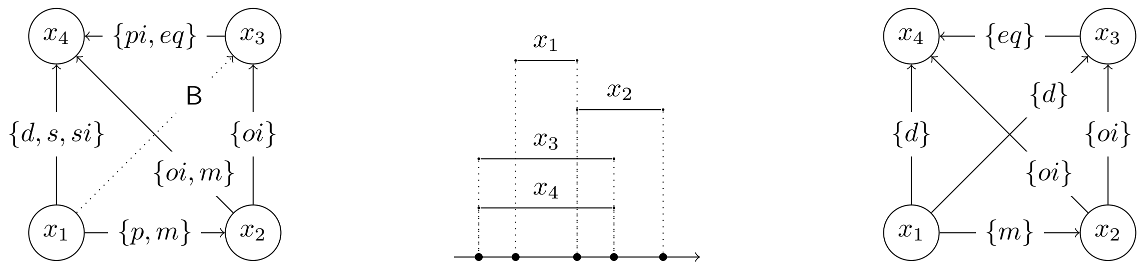
The problem of representing and reasoning about qualitative information can be modeled as a *qualitative constraint network* (QCN), defined in the following manner:

Definition 2. 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 $C : V \times V \rightarrow 2^{\mathsf{B}}$ such that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$ and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$.

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

Definition 3. Let $\mathcal{N} = (V, C)$ be a QCN, then:



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

Fig. 2: Figurative examples of QCN terminology using IA

- a *solution* of $\mathcal{N}$ is a mapping $\sigma : V \rightarrow \mathsf{D}$ such that $\forall (u, v) \in V \times V$, $\exists b \in C(u, v)$ such that $(\sigma(u), \sigma(v)) \in b$ (see Figure 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 the notation $\mathcal{N}' \subset \mathcal{N}$ is used;
- $\mathcal{N}$ 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* iff there exists 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 simple graph (V, E) where $\{u, v\} \in E$ iff $C(u, v) \neq \mathsf{B}$;
- $\mathcal{N}$ 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 reasoning with QCNs:

Definition 4. 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 that if G is complete, $\overset{\diamond}{_G}$-consistency becomes identical to $\diamond$-consistency [18], and, hence, $\diamond$-consistency is a special case of $\overset{\diamond}{_G}$-consistency. For any given QCN $\mathcal{N} = (V, C)$ and graph $G = (V, E)$, there exists a unique and largest (w.r.t. $\subseteq$) $\overset{\diamond}{_G}$-consistent sub-QCN of $\mathcal{N}$, denoted by $\overset{\diamond}{_G}(\mathcal{N})$, which is also equivalent to $\mathcal{N}$ [9].

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 5. 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 [9]. As with $\overset{\diamond}{_G}$-consistency, for any given QCN $\mathcal{N} = (V, C)$ and graph $G = (V, E)$, there exists a unique and largest (w.r.t. $\subseteq$) $\overset{\blacklozenge}{_G}$-consistent sub-QCN of $\mathcal{N}$, denoted by $\overset{\blacklozenge}{_G}(\mathcal{N})$, which is also equivalent to $\mathcal{N}$ [9]. Interestingly, $\overset{\blacklozenge}{_G}$-consistency can also be seen as a counterpart of singleton arc consistency (SAC) [19] 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 $(V, C') = \overset{\diamond}{_G}(\mathcal{N}_{[v,v']/\{b\}})$.

We recall the following result, which suggests that $\overset{\blacklozenge}{_G}$-consistency entails $\overset{\diamond}{_G}$-consistency:

Proposition 2 ([4]). *Let $\mathcal{N} = (V, C)$ be a* QCN *and $G = (V, E)$ a graph. If $\mathcal{N}$ is $\overset{\blacklozenge}{_G}$-consistent, then $\mathcal{N}$ is $\overset{\diamond}{_G}$-consistent.*

Finally, the following result shows the connection between $\overset{\blacklozenge}{_G}$-consistency and minimal QCNs:

Proposition 3 ([4]). *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 $\overset{\diamond}{_G}(\mathcal{N})$ 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 $(V, C') = \overset{\blacklozenge}{_G}(\mathcal{N})$, the base relation b*

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

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

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

Algorithm 2: $\mathsf{PSWPC}(\mathcal{N}, G)$ cf. [4]

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

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

is a feasible base relation of $\mathcal{N}$.

As a note, an interesting case where the property described in Proposition 3 can be satisfied, is the case where the considered subclass of relations is obtained from a relation algebra that has *patchwork* [20] for $\diamond_G$-consistent and not trivially inconsistent QCNs defined over that subclass, where $G = (V, E)$ is any chordal graph such that $\mathsf{G}(\mathcal{N}) \subseteq G$ for a given QCN $\mathcal{N} = (V, C)$. As a matter of fact, patchwork holds for all the qualitative constraint languages mentioned in Proposition 1 [21]. 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 certain restrictions in the structure of the constraint graph of a QCN are imposed (a trivial case being restricting a constraint graph to being a tree).

Algorithm 3: $\mathsf{PWPC}(\mathcal{N}, G, e \leftarrow \emptyset)$ [22]

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 $\mathcal{N}$.

begin
  $Q \leftarrow (e$ **if** $e \neq \emptyset$ **else** $E)$;
  while $Q \neq \emptyset$ **do**
    $\{v, v'\} \leftarrow Q.pop()$;
    foreach $v'' \in V \mid \{v, v''\}, \{v', v''\} \in E$ **do**
      $r \leftarrow C(v, v'') \cap (C(v, v') \diamond C(v', v''))$;
      if $r \subset C(v, v'')$ **then**
        $C(v, v'') \leftarrow r$;
        $C(v'', v) \leftarrow r^{-1}$;
        $Q \leftarrow Q \cup \{\{v, v''\}\}$;
      $r \leftarrow C(v'', v') \cap (C(v'', v) \diamond C(v, v'))$;
      if $r \subset C(v'', v')$ **then**
        $C(v'', v') \leftarrow r$;
        $C(v', v'') \leftarrow r^{-1}$;
        $Q \leftarrow Q \cup \{\{v'', v'\}\}$;
  return $\mathcal{N}$;

3. Efficiently Approximating $\blacklozenge_G$-Consistency

In this section, we propose an algorithm to efficiently approximate $\blacklozenge_G$-consistency for a given QCN $\mathcal{N} = (V, C)$ with respect to a graph $G = (V, E)$, presented in Algorithm 1 and called $\ell\mathsf{PSWPC}^{\cup}$ (which stands for *lazy* $\cup$-*collective* partial singleton weak path consistency).

A key feature of this algorithm is that it collectively eliminates certain unfeasible base relations by exploiting the singleton checks that are typically performed by an algorithm for enforcing $\blacklozenge_G$-consistency, such as the one presented in Algorithm 2 and called PSWPC. The operations involved in the singleton checks themselves are based on the use of an algorithm for enforcing $\diamond_G$-consistency, such as the one presented in Algorithm 3 and called PWPC (which stands for partial weak path consistency), and are in line with Definition 5 of $\blacklozenge_G$-consistency. We refer to the exploited singleton checks that are used to collectively eliminate certain unfeasible base relations as *collective singleton checks* and we define them as follows. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, a *collective singleton check* for a constraint $C(v, v')$ with $\{v, v'\} \in E$ consists of computing the QCN $\mathcal{N}' = \bigcup\{\diamond_G(\mathcal{N}_{[v,v']/\{b\}}) \mid b \in C(v, v')\}$ and checking if $\mathcal{N}' \subset \mathcal{N}$. Simply put, a collective singleton check involves successively instantiating a given constraint of a QCN with each of its base relations, computing and unifying the corresponding $\diamond_G$-consistent QCNs, and checking if there exist stronger constraints in the resulting QCN than the respective ones in the original QCN so that the latter can be updated accordingly.

Another feature of our algorithm is that during its execution it takes a *lazy* (non-exhaustive) approach and performs a collective singleton check only for a constraint that has been revised and put into the queue due to a previous collective singleton check for some other constraint. As we will see in what follows, this behavior leads to a non-unique closure being obtained in general for a given input QCN.

Further, as opposed to PSWPC, our algorithm initially takes into account only non-universal relations of $\diamond_G(\mathcal{N})$ for a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$.

In all other aspects, algorithm ℓPSWPC$^{\cup}$ can be viewed as being similar to the one for efficiently achieving $\blacklozenge_G$-consistency, namely, PSWPC, which in itself is an advancement[1] of the respective algorithm for enforcing $\blacklozenge_G$-consistency that is presented in [4], and as a non-exhaustive variant of the algorithm for enforcing a stronger notion than $\blacklozenge_G$-consistency that recently appeared in [26].

We prove the following main result regarding algorithm ℓPSWPC$^{\cup}$, which captures its major theoretical properties:

Theorem 1. *Given a* QCN $\mathcal{N} = (V, C)$ *of a relation algebra and a graph* $G = (V, E)$*, algorithm* ℓPSWPC$^{\cup}$ *terminates and returns a sub-*QCN $\mathcal{N}'$ *of* $\mathcal{N}$ *such that:*

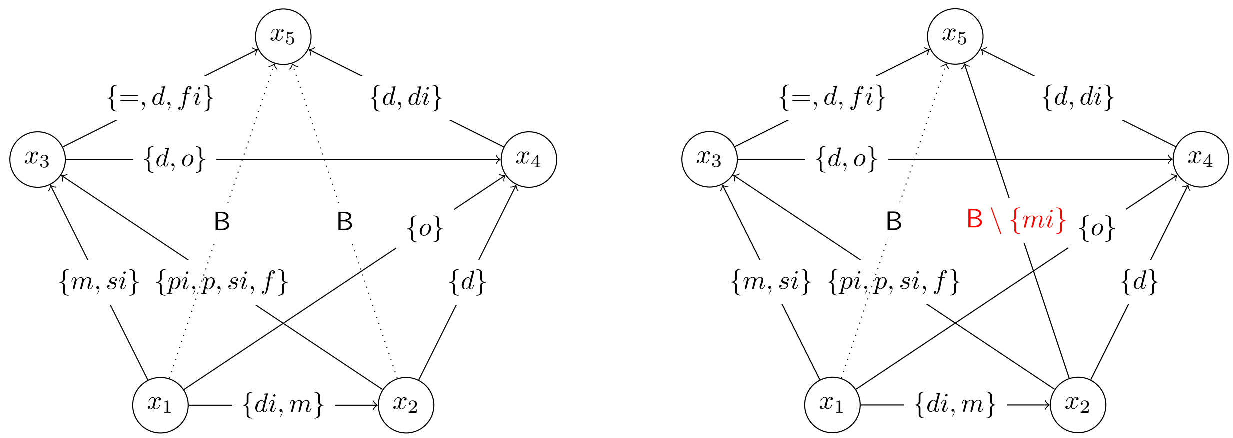
- $\mathcal{N}'$ *is* $\diamond_G$*-consistent;*
- $\mathcal{N}'$ *is equivalent to* $\mathcal{N}$*;*
- $\mathcal{N}'$ *is non-unique;*
- $\mathcal{N}'$ *is incomparable to* $\blacklozenge_G(\mathcal{N})$*;*
- $\mathcal{N}' \subseteq \diamond_G(\mathcal{N})$.

Proof. In line 2 of the algorithm, the original QCN $\mathcal{N}$ is made $\diamond_G$-consistent via a call to function PWPC; let $\mathcal{N}' = (V, C') = \diamond_G(\mathcal{N})$. We need to show that the rest of the refinement operations in the algorithm entail $\diamond_G$-consistency as well. By utilizing the incremental functionality of algorithm PWPC (see [27, Section 3]), in lines 7–8 of the algorithm, for a pair of variables $\{u, u'\} \in E$ a set of $\diamond_G$-consistent sub-QCNs of $\mathcal{N}'$ is created, namely, the set $S = \{\diamond_G(\mathcal{N}'_{[u,u']/\{b\}}) \mid b \in C'(u, u')\}$. Then, in those same lines, the operation $\bigcup S$ takes place. We show that $\bigcup S$ is $\diamond_G$-consistent. Clearly, if all the QCNs in S are trivially inconsistent, then $\bigcup S$ is $\diamond_G$-consistent. Thus, let us assume that there exist k not trivially inconsistent QCNs in S, with $k \leq |C'(u, u')|$, and hence let $\mathcal{N}_1 = (V, C_1)$, $\mathcal{N}_2 = (V, C_2)$, …, $\mathcal{N}_k = (V, C_k)$ be all the k different $\diamond_G$-consistent and not trivially inconsistent QCNs in S. We need to show that $\mathcal{N}^* = (V, C^*) = \bigcup_{i=1}^{k} \mathcal{N}_i$ is $\diamond_G$-consistent. Let us consider three variables $v, v', v'' \in V$ such that $\{v, v'\}, \{v, v''\}, \{v', v''\} \in E$, and a base relation b such that $b \in C^*(v, v')$. Then, we have that $b \in C_i(v, v')$ for some $i \in \{1, 2, \ldots, k\}$. Since $\mathcal{N}_i$ is $\diamond_G$-consistent, we have that $C_i(v, v') \subseteq C_i(v, v'') \diamond C_i(v'', v')$ and, as it is not trivially inconsistent, there exist base relations $b' \in C_i(v, v'')$ and $b'' \in C_i(v'', v')$ such that $b \in b' \diamond b''$. Therefore, we have that $b' \in C^*(v, v'')$ and $b'' \in C^*(v'', v')$. It follows that $b \in C^*(v, v'') \diamond C^*(v'', v')$ and that $\mathcal{N}^*$ is $\diamond_G$-consistent. This proves that the algorithm terminates and returns a $\diamond_G$-consistent sub-QCN of $\mathcal{N}$.

Let $\mathcal{N}' = (V, C') = \diamond_G(\mathcal{N})$ (line 2 of the algorithm). By equivalence of $\diamond_G$-consistency $\mathcal{N}'$ is equivalent to $\mathcal{N}$. Further, let $b \in C'(u, u')$ with $\{u, u'\} \in E$ be a base relation. In lines 9–14 of the algorithm, the base relation b is eliminated only 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 need to show that b is an unfeasible base relation of $\mathcal{N}'$. Let us suppose that b is a feasible base relation. Then, by definition of feasible base relations there exists a scenario $\mathcal{S} = (V, C^a)$ of $\mathcal{N}'$ such that $C^a(u, u') = \{b\}$. Further, it holds that $\diamond_G(\mathcal{S}) = \mathcal{S}$, as $\mathcal{S}$ being an atomic and satisfiable QCN is minimal and, hence, necessarily $\diamond_G$-consistent. Thus, it follows that $\forall\{v, v'\} \in E$ we have that $b \in C'''(u, u')$, where $(V, C''') = \diamond_G(\mathcal{N}'_{[v,v']/C^a(v,v')})$, as $\mathcal{S} \subseteq \mathcal{N}'_{[v,v']/C^a(v,v')}$ and, hence, $\diamond_G(\mathcal{S}) \subseteq \diamond_G(\mathcal{N}'_{[v,v']/C^a(v,v')})$ by monotonicity of $\diamond_G$-consistency. As $\mathcal{S} \subseteq \mathcal{N}'$, it follows that $\forall\{v, v'\} \in E$ we have that $C^a(v, v') \subseteq C'(v, v')$ 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 $C^a(v, v')$ 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 $b \in C''(u, u')$, which concludes our proof by contraposition. This proves that the algorithm terminates and returns a sub-QCN of $\mathcal{N}$ that is equivalent to $\mathcal{N}$.

In what follows, we give an intuition of why the order in which the constraints are processed ultimately affects the output of the algorithm. The validity of the result itself is supported by a counterexample, which is impossible to present here due to space limitations. Let $\mathcal{N}' = (V, C') = \diamond_G(\mathcal{N})$ (line 2 of the algorithm). Further, consider two different pairs of variables $\{v, v'\}, \{u, u'\} \in E$, and let $\mathcal{N}^{uu'} = (V, C^{uu'}) = \bigcup\{\diamond_G(\mathcal{N}'_{[u,u']/\{b\}}) \mid b \in C'(u, u')\}$ and $\mathcal{N}^{vv'} = (V, C^{vv'}) = \bigcup\{\diamond_G(\mathcal{N}'_{[v,v']/\{b\}}) \mid b \in C'(v, v')\}$. Then, it is entirely possible that there exist two different pairs of variables $\{y, y'\}, \{w, w'\} \in E \setminus \{\{v, v'\}, \{u, u'\}\}$ such that $C^{uu'}(y, y') \subset C^{vv'}(y, y')$ and $C^{uu'}(w, w') \supset C^{vv'}(w, w')$. (In fact, such an example can be constructed by considering two copies of the QCN of Figure 4 inside a larger QCN.) It follows that $\mathcal{N}^{uu'} \not\subseteq \mathcal{N}^{vv'}$ and $\mathcal{N}^{vv'} \not\subseteq \mathcal{N}^{uu'}$. Since both $\mathcal{N}^{uu'}$ and $\mathcal{N}^{vv'}$ are sub-QCNs of $\mathcal{N}'$, but incomparable to each other even if we only take into account constraints between pairs of variables other than $\{v, v'\}$ and $\{u, u'\}$, this result suggests that different constraints may be revised and put into the queue of the algorithm depending on the order in which $\mathcal{N}^{uu'}$ and $\mathcal{N}^{vv'}$ are calculated. As the algorithm takes a *lazy* (non-exhaustive) approach during its execution and performs a collective singleton check (in lines 7–14) only for a constraint that has been revised and put into the queue due to a previous collective singleton check for some other constraint, the order in which these collective singleton checks are performed is important and can lead to different outputs for the same input QCN.

[1] We use a queue in PSWPC that is initialized with all of the edges of a given graph G, which correspond to constraints of a given QCN, and that is also filled with all of the aforementioned edges during execution whenever any of those constraints is revised, i.e., whenever a base relation is removed. This technique is equivalent to executing a break statement in the algorithm of [4] whenever a singleton check fails and, hence, a constraint is revised, forcing the inner loop in that algorithm to stop and using the outer loop to initiate singleton checks in a fresh QCN. We have found this to work much better in practice, cutting down on the number of constraint checks performed by $\sim 20\%$. Further, using a queue allows for prioritizing certain edges, a strategy which is in line with similar techniques used in the algorithm for enforcing $\diamond_G$-consistency [23], [24], [25], but this is something that we have not yet explored and retain for future work.



(a) A QCN $\mathcal{N} = (V, C)$ of IA

(b) The QCN $\blacklozenge_G(\mathcal{N})$

Fig. 3: Given the QCN $\mathcal{N} = (V, C)$ of Figure 3a and the graph G that results by removing the edge $\{x_1, x_5\}$ from the complete graph on V, algorithm ℓPSWPC$^\cup$ is unable to eliminate the base relation mi in $C(x_2, x_5)$ for any possible order in which the constraints are processed; however, mi is not $\blacklozenge_G$-consistent for $C(x_2, x_5)$, as shown in Figure 3b

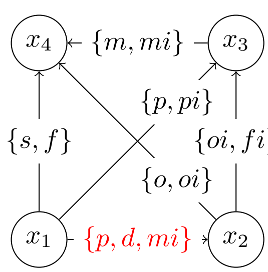


Fig. 4: A $\blacklozenge_G$-consistent QCN $\mathcal{N}$

Consider the QCN of Figure 3a and let G be the graph that results by removing the edge $\{x_1, x_5\}$ from the complete graph on the set of variables of the QCN. Given that QCN and the graph G as input, algorithm ℓPSWPC$^\cup$ is unable to eliminate any base relation in the QCN for any possible order in which the constraints are processed. However, the QCN is not $\blacklozenge_G$-consistent as shown in Figure 3b. Indeed, the base relation mi is not $\blacklozenge_G$-consistent for the constraint between variables x_2 and x_5. Therefore, $\blacklozenge_G$-consistency is able to eliminate more base relations than our algorithm in this case. Next, consider the $\blacklozenge_G$-consistent QCN of Figure 4 with respect to the complete graph G on the set of variables of the QCN. In this case, our algorithm is able to eliminate the base relation d for the constraint between variables x_1 and x_2 for any possible order in which the constraints are processed. This suggests that given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, algorithm ℓPSWPC$^\cup$ can produce an output sub-QCN $\mathcal{N}'$ of $\mathcal{N}$ such that $\blacklozenge_G(\mathcal{N}) \not\subseteq \mathcal{N}'$ and $\mathcal{N}' \not\subseteq \blacklozenge_G(\mathcal{N})$. This proves that the algorithm terminates and returns a sub-QCN $\mathcal{N}'$ of $\mathcal{N}$ that is, in general, incomparable to $\blacklozenge_G(\mathcal{N})$.

We have already established that algorithm ℓPSWPC$^\cup$ terminates and returns a $\lozenge_G$-consistent sub-QCN $\mathcal{N}'$ of $\mathcal{N}$ in the first part of this proof. By dominance of $\lozenge_G$-consistency we have that $\lozenge_G(\mathcal{N})$ is the largest (w.r.t. $\subseteq$) $\lozenge_G$-consistent sub-QCN of $\mathcal{N}$. Therefore, it follows that $\mathcal{N}' \subseteq \lozenge_G(\mathcal{N})$. $\square$

Time complexity analysis of ℓPSWPC$^\cup$. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, we have that algorithm ℓPSWPC$^\cup$ terminates 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 9–14 of the algorithm. (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 $O(\Delta \cdot |E| \cdot \mathsf{B})$ [22], a worst-case time complexity of $O(\Delta \cdot |E|^3 \cdot \mathsf{B}^3)$ can be obtained for algorithm ℓPSWPC$^\cup$. Notably, this is also the worst-case time complexity of algorithm PSWPC [4], however, in the next section we demonstrate that ℓPSWPC$^\cup$ significantly outperforms PSWPC in practice.

4. Experimental Evaluation

We evaluated the performance of an implementation of algorithm ℓPSWPC$^\cup$ against an implementation of the algorithm for enforcing partial $\blacklozenge_G$-consistency that was presented here, viz., PSWPC, with a varied dataset of QCNs of IA.

Technical specifications. The evaluation was carried out on a computer with an Intel Core i5-6200U processor, 8 GB of RAM, and the Xenial Xerus x86_64 OS (Ubuntu Linux). All algorithms were coded in Python and run using the PyPy interpreter under version 5.1.2, which implements Python 2.7.10. Only one CPU core was used.

Tab. 1: Evaluation of the computational effort of algorithms PSWPC and ℓPSWPC$^{\cup}$

(a) Evaluation with random IA networks of model $\mathsf{A}(\mathsf{n}=70, \mathsf{l}=6.5, \mathsf{d})$ [28]

	min		μ		max		σ	
d	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$
7	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{2.66s}{10k}$	$\frac{0.38s}{2k}$	$\frac{7.40s}{21k}$	$\frac{1.22s}{4k}$	$\frac{1.23s}{4k}$	$\frac{0.20s}{528}$
8	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{7.88s}{18k}$	$\frac{1.45s}{4k}$	$\frac{59.82s}{73k}$	$\frac{17.91s}{16k}$	$\frac{8.55s}{12k}$	$\frac{2.16s}{3k}$
9	$\frac{0.01s}{1}$	$\frac{0.04s}{1}$	$\frac{23.56s}{37k}$	$\frac{5.04s}{8k}$	$\frac{132.00s}{172k}$	$\frac{73.33s}{75k}$	$\frac{27.67s}{30k}$	$\frac{10.36s}{11k}$
10	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{43.56s}{36k}$	$\frac{8.86s}{9k}$	$\frac{343.99s}{205k}$	$\frac{96.71s}{90k}$	$\frac{64.98s}{52k}$	$\frac{18.60s}{16k}$
11	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{6.96s}{3k}$	$\frac{1.53s}{462}$	$\frac{126.19s}{151k}$	$\frac{16.45s}{9k}$	$\frac{16.52s}{15k}$	$\frac{3.05s}{2k}$
12	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{1.63s}{121}$	$\frac{0.43s}{39}$	$\frac{14.16s}{2k}$	$\frac{6.05s}{623}$	$\frac{2.75s}{205}$	$\frac{0.94s}{89}$

(b) Evaluation with structured random IA networks of model $\mathsf{BA}(\mathsf{n}=150, \mathsf{m})$ [6]

	min		μ		max		σ	
m	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$	PSWPC	ℓPSWPC$^{\cup}$
2	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{0.60s}{5k}$	$\frac{0.11s}{741}$	$\frac{2.21s}{12k}$	$\frac{0.49s}{3k}$	$\frac{0.30s}{2k}$	$\frac{0.06s}{266}$
3	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{11.82s}{19k}$	$\frac{2.33s}{4k}$	$\frac{73.86s}{81k}$	$\frac{18.76s}{23k}$	$\frac{11.28s}{11k}$	$\frac{2.91s}{4k}$
4	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{115.37s}{57k}$	$\frac{30.92s}{17k}$	$\frac{1423.06s}{552k}$	$\frac{575.15s}{238k}$	$\frac{205.92s}{97k}$	$\frac{73.56s}{34k}$
5	$\frac{0.00s}{1}$	$\frac{0.00s}{1}$	$\frac{0.94s}{24}$	$\frac{0.04s}{2}$	$\frac{24.28s}{585}$	$\frac{1.44s}{42}$	$\frac{3.91s}{96}$	$\frac{0.17s}{6}$

Tab. 2: Evaluation of the pruning capability of algorithm ℓPSWPC$^{\cup}$ compared to that of PSWPC; a percentage $x\%$ denotes that ℓPSWPC$^{\cup}$ removed $x\%$ more base relations

(a) Evaluation with the IA networks used in Table 1a

d	min	μ	max	med	#‖
7	−2.97%	−0.07%	0%	0%	0
8	−6.18%	−0.58%	0%	−0.09%	5
9	−36.99%	−3.59%	0.03%	−1.58%	14
10	−95.01%	−10.57%	0.18%	0%	10
11	−95.49%	−3.21%	0%	0%	0
12	0%	0%	0%	0%	0

(b) Evaluation with the IA networks used in Table 1b

m	min	μ	max	med	#‖
2	0%	0%	0%	0%	0
3	−3.94%	−0.22%	0%	0%	4
4	−74.41%	−1.36%	0.03%	0%	23
5	0%	0%	0%	0%	0

Dataset. We employed models $\mathsf{A}(\mathsf{n}, \mathsf{l}, \mathsf{d})$ [28] and $\mathsf{BA}(\mathsf{n}, \mathsf{m})$ [6] to generate random QCNs of IA. In particular, $\mathsf{A}(\mathsf{n}, \mathsf{l}, \mathsf{d})$ can generate random QCNs of n variables with an average number l of base relations per non-universal constraint and an average degree d for the respective constraint graphs, and $\mathsf{BA}(\mathsf{n}, \mathsf{m})$ can generate random QCNs of n variables with an average number $|\mathsf{B}|/2$ of base relations per non-universal constraint and by use of a *preferential attachment* [29] value m for the respective constraint graphs. Using model $\mathsf{A}(\mathsf{n}, \mathsf{l}, \mathsf{d})$, we generated 100 QCNs of IA of $\mathsf{n} = 70$ variables with $\mathsf{l} = 6.5$ base relations per non-universal constraint on average for all values of d ranging from 7 to 12 with a step of 1, as the *phase transition region* [30] for this model is observed for $8 \leq \mathsf{d} \leq 11$ [28]. Using model $\mathsf{BA}(\mathsf{n}, \mathsf{m})$, we generated 100 QCNs of IA of $\mathsf{n} = 150$ variables for all values of m ranging from 2 to 5 with a step of 1, as the phase transition region for this model is observed for $\mathsf{m} \approx 3$ or 4 [31]. Finally, regarding the graphs that were given as input to our algorithms, the *maximum cardinality search* algorithm [32] was used to obtain triangulations of the constraint graphs of our QCNs. The choice of such chordal graphs was reasonable given their extensive use in the recent literature, as reviewed in [7].

Measures. The first measure considers the number of *constraint checks per base relation removals* performed by an algorithm for meeting its objective. Given a QCN $\mathcal{N} = (V, C)$ and 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 it if that condition is satisfied. The second measure concerns the *CPU time* and is naturally correlated with the first one. The third measure compares the *pruning capability* between the evaluated algorithms, and, finally, the fourth measure keeps track of the number of cases where the algorithms yield *incomparable outputs*; this measure in particular is denoted by symbol #‖ in Table 2.

Results. The results of our experimental evalualation are detailed in Tables 1 and 2, where a fraction $\frac{x}{y}$ in Table 1 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, with respect to computational effort, Table 1 shows that $\ell\mathsf{PSWPC}^{\cup}$ had a significant advantage over PSWPC in all cases and, in particular, that $\ell\mathsf{PSWPC}^{\cup}$ was up to 5 times faster than PSWPC on average for the more difficult instances; and with respect to pruning capability, Table 2 shows that, in most cases, $\ell\mathsf{PSWPC}^{\cup}$ performed almost the same pruning in the labels of a given QCN as PSWPC. Notably, PSWPC unveiled 6 more inconsistencies than $\ell\mathsf{PSWPC}^{\cup}$ in a total of $1\,000$ QCNs, in particular, 3 more for $\mathsf{A}(70, 6.5, 10)$ and $\mathsf{A}(70, 6.5, 11)$ respectively. Further, as demonstrated by measure $\#\|$, there were 56 cases of incomparable outputs. Finally, we note that the order in which the constraints were processed was random for both algorithms. (The queue data structure in both algorithms was implemented using sets, as suggested by their compact representations in Algorithms 1 and 2; sets have no order.)

5. Conclusion and Future Work

In this paper, we proposed a lazy algorithm that restricts the singleton checks associated with partial $\overset{\bullet}{_G}$-consistency to constraints that are likely to lead to the removal of a base relation upon their propagation. A key feature of this algorithm is that it collectively eliminates certain unfeasible base relations by exploiting singleton checks. Further, we studied certain theoretical properties of this algorithm and demonstrated its efficiency against the state-of-the-art algorithm for enforcing partial $\overset{\bullet}{_G}$-consistency, with respect to both computational effort and pruning capability, via an experimental evaluation with random Interval Algebra networks from the phase transition region of two separate models. For future work we would like to explore queueing strategies that will allow our algorithm reach a fixed point faster and, possibly, even with improved pruning capability. These strategies could be coupled with certain criteria and/or a cost function that will prioritize constraints that may be propagated more efficiently based on their neighbourhood, their restrictiveness, or a combination of both.

References

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

[2] G. Ligozat and J. Renz, "What Is a Qualitative Calculus? A General Framework," in *PRICAI*, 2004.

[3] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," *Commun. ACM*, vol. 26, pp. 832–843, 1983.

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

[5] J. Huang, J. J. Li, and J. Renz, "Decomposition and tractability in qualitative spatial and temporal reasoning," *Artif. Intell.*, vol. 195, pp. 140–164, 2013.

[6] M. Sioutis, J. Condotta, and M. Koubarakis, "An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks," *Int. J. Artif. Intell. Tools*, vol. 25, pp. 1–33, 2016.

[7] M. Sioutis, Y. Salhi, and J.-F. Condotta, "Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning," *Knowl. Eng. Rev.*, vol. 32, p. e4, 2016.

[8] Z. Long, M. Sioutis, and S. Li, "Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks," in *IJCAI*, 2016.

[9] J.-F. Condotta and C. Lecoutre, "A Class of df-Consistencies for Qualitative Constraint Networks," in *KR*, 2010.

[10] D. A. Randell, Z. Cui, and A. Cohn, "A Spatial Logic Based on Regions & Connection," in *KR*, 1992.

[11] J. Renz, "A Canonical Model of the Region Connection Calculus," *J. Appl. Non-Classical Logics*, vol. 12, pp. 469–494, 2002.

[12] M. B. Vilain and H. A. Kautz, "Constraint Propagation Algorithms for Temporal Reasoning," in *AAAI*, 1986.

[13] G. Ligozat, "Reasoning about cardinal directions," *J. Vis. Lang. Comput.*, vol. 9, pp. 23–44, 1998.

[14] A. U. Frank, "Qualitative Spatial Reasoning with Cardinal Directions," in *ÖGAI*, 1991.

[15] P. Balbiani, J.-F. Condotta, and L. F. del Cerro, "Tractability Results in the Block Algebra," *J. Log. Comput.*, vol. 12, pp. 885–909, 2002.

[16] A. Tarski, "On the calculus of relations," *J. Symb. Log.*, vol. 6, pp. 73–89, 1941.

[17] F. Dylla, T. Mossakowski, T. Schneider, and D. Wolter, "Algebraic Properties of Qualitative Spatio-Temporal Calculi," in *COSIT*, 2013.

[18] J. Renz and G. Ligozat, "Weak Composition for Qualitative Spatial and Temporal Reasoning," in *CP*, 2005.

[19] R. Debruyne and C. Bessière, "Some Practicable Filtering Techniques for the Constraint Satisfaction Problem," in *IJCAI*, 1997.

[20] C. Lutz and M. Milicic, "A Tableau Algorithm for DLs with Concrete Domains and GCIs," *J. Autom. Reasoning*, vol. 38, pp. 227–259, 2007.

[21] J. Huang, "Compactness and its implications for qualitative spatial and temporal reasoning," in *KR*, 2012.

[22] A. Chmeiss and J. Condotta, "Consistency of Triangulated Temporal Qualitative Constraint Networks," in *ICTAI*, 2011.

[23] P. van Beek and D. W. Manchak, "The design and experimental analysis of algorithms for temporal reasoning," *J. Artif. Intell. Res.*, vol. 4, pp. 1–18, 1996.

[24] J. Renz and B. Nebel, "Efficient Methods for Qualitative Spatial Reasoning," *J. Artif. Intell. Res.*, vol. 15, pp. 289–318, 2001.

[25] P. B. Ladkin and A. Reinefeld, "Fast Algebraic Methods for Interval Constraint Problems," *Ann. Math. Artif. Intell.*, vol. 19, pp. 383–411, 1997.

[26] M. Sioutis, A. Paparrizou, and J.-F. Condotta, "Collective Singleton-based Local Consistency for Qualitative Constraint Networks," in *TIME*, 2017, in press.

[27] A. Gerevini, "Incremental qualitative temporal reasoning: Algorithms for the Point Algebra and the ORD-Horn class," *Artif. Intell.*, vol. 166, pp. 37–80, 2005.

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

[29] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[30] P. C. Cheeseman, B. Kanefsky, and W. M. Taylor, "Where the Really Hard Problems Are," in *IJCAI*, 1991.

[31] M. Sioutis, "Algorithmic Contributions to Qualitative Constraint-based Spatial and Temporal Reasoning," Ph.D. dissertation, Université d'Artois, 2017.

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