Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks

Zhiguo Long*
QCIS, FEIT
University of Technology Sydney
Australia
zhiguo.long@student.uts.edu.au

Michael Sioutis
CRIL-CNRS UMR 8188
University of Artois
France
sioutis@cril.fr

Sanjiang Li
UTS-AMSS Joint Lab
QCIS, FEIT
University of Technology Sydney
Australia
Sanjiang.Li@uts.edu.au

Abstract

We propose a new algorithm called **DPC**+ to enforce partial path consistency (PPC) on qualitative constraint networks. PPC restricts path consistency (PC) to a triangulation of the underlying constraint graph of a network. As PPC retains the sparseness of a constraint graph, it can make reasoning tasks such as consistency checking and minimal labelling of large qualitative constraint networks much easier to tackle than PC. For qualitative constraint networks defined over any distributive subalgebra of well-known spatio-temporal calculi, such as the Region Connection Calculus and the Interval Algebra, we show that **DPC**+ can achieve PPC very fast. Indeed, the algorithm enforces PPC on a qualitative constraint network by processing each triangle in a triangulation of its underlying constraint graph at most three times. Our experiments demonstrate significant improvements of **DPC**+ over the state-of-the-art PPC enforcing algorithm.

1 Introduction

One of the major concerns in AI is dealing with spatio-temporal information, which is involved in many AI applications, such as automatic data maintenance and visualization in Geographic Information Systems (e.g., [Wallgrün, 2012]), robotic navigation (e.g., [Wolter, 2008]), and computer-aided design (e.g., [Bhatt *et al.*, 2009]). Qualitative Spatial and Temporal Reasoning (QSTR) is devoted to processing such kind of information correctly and efficiently. In particular, the techniques in QSTR focus on finding ways to efficiently solve several fundamental reasoning problems, such as the *consistency problem*, *the minimal labelling problem*, and the *redundancy problem*. The consistency problem asks if a constraint network has a solution. The minimal labelling problem requests the *strongest* implied relations in a constraint network, i.e., the implied relations comprising only tuples that participate in a solution of the network. The redundancy problem [Li *et al.*, 2015] is a newly studied problem in QSTR that aims to simplify the representation of a constraint network by removing its redundant constraints, i.e., the constraints that can be entailed by the rest of the network. We refer to [Cohn and Renz, 2008] for more information regarding fundamental reasoning problems in QSTR.

Path consistency (PC) is an important local consistency condition for constraint networks. Given a constraint network $\mathcal{N}$, PC can be established in $\mathcal{N}$ by applying the path consistency algorithm (**PC**)[1] in [Montanari, 1974; van Beek, 1989], which makes all pairs of variables path consistent with any third variable in $\mathcal{N}$ by considering the completion of the underlying constraint graph of $\mathcal{N}$. It has long been known that **PC** can decide the consistency of maximal tractable subalgebras of several very important qualitative spatial and temporal calculi. Recently, **PC** became more useful as it was shown to be able to solve the minimal labelling problem and decisively assist in solving the redundancy problem of any *distributive subalgebra* of several well-known calculi. A subalgebra is called *distributive* if (weak) composition distributes over non-empty intersections of its relations. The concept of distributive subalgebras was recently proposed in [Li *et al.*, 2015] and further discussed in [Long and Li, 2015]; some previous related work identified certain subalgebras to be distributive as well (cf. [van Beek and Cohen, 1990]).

With the expansion of the internet, the age of Big Data has arrived. Without emphasis on scalability, the current techniques in QSTR have become less suitable to deal with the situation where tens of thousands or even millions of spatiotemporal objects are involved. For example, **PC** has a time complexity of $O(n^3)$, which already makes it hardly scalable for large real-world datasets with even less than a few thousands of variables. In the past few years, research in QSTR has focused on trying to deal with this situation. In particular, Chmeiss and Condotta [2011] and Sioutis and Koubarakis [2012] adopted the partial path consistency algorithm (**PPC**) [Bliek and Sam-Haroud, 1999], as a substitute for the original path consistency algorithm **PC**, for some fundamental reasoning tasks of certain qualitative spatial and temporal calculi. Given a constraint network $\mathcal{N}$, **PPC** enforces partial path consistency (PPC) on $\mathcal{N}$, which is PC restricted to a triangulation of the underlying constraint graph of $\mathcal{N}$. As such, **PPC** can exploit the sparseness of a constraint graph, in contrast to **PC**, and therefore be more efficient than

*Corresponding author.

[1] In what follows, we use bold sans serif font to denote an algorithm name (e.g., the PC enforcing algorithm is denoted by **PC**).

PC, especially when dealing with large and sparsely structured qualitative constraint networks. Note that when the result of a triangulation for a given constraint network $\mathcal{N}$ is a complete graph, **PPC** will enforce PC on $\mathcal{N}$. Further, it was shown by Sioutis et al. [2015b] and Long and Li [2015] that **PPC** can have the same reasoning power as **PC** on the common edges between a triangulation and the completion of the underlying constraint graph of $\mathcal{N}$. In particular, if $\mathcal{N}$ is defined over some distributive subalgebra, the relations on these edges will be minimal after enforcing PPC w.r.t. a triangulation on $\mathcal{N}$, as would be the case with PC. This kind of hybrid restrictions on both the structure of the constraint graph and the allowed relations have also been discussed in the context of constraint satisfaction problems to identify tractable subclasses (see [Cohen *et al.*, 2012]). **PPC** can also be helpful to efficiently identify the same set of non-redundant constraints in $\mathcal{N}$ as **PC**, by only considering the relations on these edges.

In addition, there is a parallel research stream in the field of the Simple Temporal Problem (STP). Xu and Choueiry [2003] realized that the STP has convexity properties analogous to those of convex CSPs and, thus, implemented a particular PPC enforcing algorithm for the STP, much like **PPC**, which was originally implemented for convex CSPs [Bliek and Sam-Haroud, 1999]. Later, Planken et al. [2008] noticed that the algorithm of Xu and Choueiry is not very efficient, as it is sometimes quadratic in the number of triangles in a triangulation of the underlying constraint graph of a given STP instance, which is also the case with **PPC** for CSPs and qualitative constraint networks. Therefore, based on an algorithm developed by Dechter et al. [1991], viz., the directional path consistency algorithm (**DPC**), Planken et al. proposed a new PPC enforcing algorithm, called $\mathbf{P^3C}$, which has a worst-case time complexity that is linear in the aforementioned number of triangles.

It is then natural to ask if there is a similar algorithm to $\mathbf{P^3C}$ for qualitative spatial and temporal calculi, as it would be quite useful for solving the associated reasoning tasks more efficiently, especially when large and sparsely structured qualitative constraint networks were involved. Our contributions with respect to answering that question are as follows. We develop a similar algorithm to $\mathbf{P^3C}$, called **DPC**+, and prove that it can correctly enforce PPC on a qualitative constraint network that is defined over some distributive subalgebra. **DPC**+ can achieve PPC (and PC when a complete graph is used as the result of a triangulation) by processing each triangle in a triangulation of the underlying constraint graph of a given qualitative constraint network no more than three times. As **PPC** (and **PC**) in general process each such triangle many more times than that, we show that **DPC**+ is more efficient than **PPC** (and **PC**) in theory. Our experimental results with both real-world and synthetic datasets also confirm in practice that **DPC**+ can be significantly more efficient than **PPC** (and **PC**).

The remainder of the paper is organized as follows. After introducing some related concepts and results in Sections 2 and 3 respectively, we present and analyse our new algorithm for achieving PPC in Section 4, and experimentally illustrate its efficiency in Section 5. Section 6 concludes the paper.

2 Preliminaries

A binary relation is a set of ordered pairs of entities. Suppose $\mathcal{U}$ is a domain of spatial or temporal entities. We write $\mathbf{Rel}(\mathcal{U})$ for the Boolean algebra of binary relations on $\mathcal{U}$. A *qualitative calculus* [Ligozat and Renz, 2004] $\mathcal{M}$ on $\mathcal{U}$ is defined as a finite Boolean subalgebra of $\mathbf{Rel}(\mathcal{U})$ that has an atom corresponding to the identity relation $\mathsf{id}_{\mathcal{U}}$ on $\mathcal{U}$, and that is closed under converse, i.e., R is in $\mathcal{M}$ iff its converse

$$R^{-1} = \{(a,b) \in \mathcal{U} \times \mathcal{U} : (b,a) \in R\}$$

is in $\mathcal{M}$. The basic relations in a qualitative calculus $\mathcal{M}$ are the atoms in $\mathcal{M}$, which form a partition of $\mathcal{U} \times \mathcal{U}$. In what follows, we denote the set of basic relations by B. A relation R in $\mathcal{M}$ is a subset of B, i.e., an element of 2^{B}. Specifically, the set of all basic relations in $\mathcal{M}$ is the universal relation, denoted by $\star$. For example, PA has the basic relations $<$, $>$, and $=$, and the universal relation $\star$ in PA is $\{<,>,=\}$.

In this paper, we are interested in the six well-known qualitative calculi of Point Algebra (PA) [Vilain and Kautz, 1986], Interval Algebra (IA) [Allen, 1983], Cardinal Relation Algebra (CRA) [Frank, 1991; Ligozat, 1998], Block Algebra (BA) [Balbiani *et al.*, 2002], and two instances of the Region Connection Calculus (RCC5/8) [Randell *et al.*, 1992].

Let $\circ$ denote the usual composition of relations, i.e., $R \circ S = \{(x,y) : \exists z \text{ s.t. } (x,z) \in R \wedge (z,y) \in S\}$. The *weak composition* of two basic relations α and β, denoted by $\alpha \diamond \beta$, is defined as $\{\gamma \in \mathsf{B} : \gamma \cap (\alpha \circ \beta) \neq \varnothing\}$. When the relations are not basic, we have $R \diamond S = \bigcup\{\alpha \diamond \beta : \alpha \in R, \beta \in S\}$.

A constraint (xRy) associates a relation $R_{ij} \in 2^{\mathsf{B}}$ with the pair of variables (x,y). For example, for two time point variables t_1 and t_2, $t_1 < t_2$ suggests that any assignment to t_1 and t_2 respectively should satisfy relation $<$.

Definition 1. A qualitative constraint network (QCN) over a qualitative calculus $\mathcal{M}$ is a tuple $(V, \mathcal{C})$, where $V = \{v_1, \ldots, v_n\}$ is a non-empty finite set of variables and $\mathcal{C}$ is a set of constraints for each pair (v_i, v_j) of $V \times V$.

In this paper, we require that the relations between variables in a QCN are symmetric, i.e., $R_{ij} = R_{ji}^{-1}$, and $R_{ii} = \mathsf{id}_{\mathcal{U}}$ for all $v_i, v_j \in V$. A *subnetwork* $\mathcal{N}' = (V', \mathcal{C}')$ of $\mathcal{N} = (V, \mathcal{C})$ is a network such that $V' = V$ and $\forall v_i, v_j \in V$ we have $R'_{ij} \subseteq R_{ij}$, where R'_{ij} is a relation in $\mathcal{C}'$ and R_{ij} a relation in $\mathcal{C}$.

The (underlying) *constraint graph* of a QCN $\mathcal{N}$, denoted by $G_{\mathcal{N}}$, is a graph that has the variables of $\mathcal{N}$ as its set of vertices, and a set of edges, denoted by $E(G_{\mathcal{N}})$, that contains an edge $\{v_i, v_j\}$ iff $R_{ij} \neq \star$ and $v_i \neq v_j$.

In graph theory, there is a special family of undirected graphs, called the *triangulated* or *chordal* graphs. An undirected graph $G = (V, E)$ is *chordal* if every cycle of length greater than 3 has a *chord*, i.e., an edge connecting two nonconsecutive vertices of the cycle [Blair and Peyton, 1993]. Note that a complete graph of any order is also a chordal graph. For each $v \in V$, the adjacency set $\mathsf{adj}(v)$ is defined as $\{w \in V : \{v,w\} \in E\}$. A vertex v is *simplicial* if $\mathsf{adj}(v)$ induces a complete graph. Every chordal graph has a simplicial vertex. Moreover, after removing a simplicial vertex and its incident edges from a chordal graph, the resulting subgraph remains chordal. The order in which simplicial vertices of sequential subgraphs are successively removed is called a *perfect elimination ordering*. Suppose that $(v_n, v_{n-1}, \ldots, v_1)$ is a perfect elimination ordering of G, then we will denote by F_k the set $\{v_j : \{v_j, v_k\} \in E \wedge j < k\}$. Note that the subgraph induced by F_k is a complete subgraph of G.

Let $\mathcal{M}$ be a qualitative calculus. A *subalgebra* $\mathcal{S}$ of $\mathcal{M}$ contains all basic relations and a subset of non-basic relations in $\mathcal{M}$, and is closed under converse, weak composition, and intersection. The concept of a *distributive subalgebra*, first proposed in [Li *et al.*, 2015] and further discussed in [Long and Li, 2015], turns out to be useful for developing efficient algorithms to accomplish reasoning tasks such as deciding the consistency or solving the minimal labelling problem of a given qualitative constraint network.

Definition 2. A subalgebra $\mathcal{S}$ of $\mathcal{M}$ is *distributive* if $R \diamond (S \cap T) = (R \diamond S) \cap (R \diamond T)$ and $(S \cap T) \diamond R = (S \diamond R) \cap (T \diamond R)$ for any $R, S, T \in \mathcal{S}$ with $S \cap T \neq \varnothing$.

Throughout the paper, we consider the calculi of PA, IA, RCC5/8, CRA, and BA, and the term "distributive subalgebra" specifically refers to a distributive subalgebra of one of these calculi. When $\mathcal{M}$ is one of these calculi, it satisfies the following property [Dylla *et al.*, 2013; Düntsch, 2005].

$$\mathcal{M} \text{ is a relation algebra.} \tag{1}$$

The *Peircean Law* (also known as the *Cycle Law*) [Dylla *et al.*, 2013; Long and Li, 2015] holds for relation algebras.

Fact 1. *For relations R, S, T of a relation algebra, the Peircean Law requires that*

$$\begin{aligned}(R \diamond S) \cap T \neq \varnothing &\Leftrightarrow (R^{-1} \diamond T) \cap S \neq \varnothing \\ &\Leftrightarrow (T \diamond S^{-1}) \cap R \neq \varnothing.\end{aligned}$$

Distributive subalgebras also have a useful property that is closely related to the well-known Helly's Theorem [Danzer *et al.*, 1963].

Definition 3. A subclass $\mathcal{S}$ of a qualitative calculus is called *Helly* if, for any finite n relations $R_1, \ldots, R_n \in \mathcal{S}$, we have

$$\bigcap_{i=1}^{n} R_i \neq \varnothing \quad \text{iff} \quad (\forall 1 \leq i \neq j \leq n)\ R_i \cap R_j \neq \varnothing.$$

Theorem 2 ([Long and Li, 2015]). *Suppose $\mathcal{M}$ is a qualitative calculus that is also a relation algebra. Let $\mathcal{S}$ be a subalgebra of $\mathcal{M}$. Then $\mathcal{S}$ is distributive iff it is Helly.*

With this property, to check if a set of relations have a non-empty intersection, one only needs to check if the intersection of each pair of relations from that set is non-empty.

3 Properties of PC and PPC

Path consistency is an important local consistency condition for several reasoning problems of CSPs. In the context of qualitative spatial and temporal calculi, the usual composition is replaced with weak composition. We have the following definition of path consistency regarding QCNs:

Definition 4. A QCN $\mathcal{N} = (V, \mathcal{C})$ is *path consistent* (PC) iff $\forall v_i, v_k, v_j \in V$ we have that $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.

Path consistency concerns all triples of variables. This could be an overkill for many reasoning tasks. The idea of enforcing path consistency on a triangulation of the constraint graph of a given CSP was first proposed by Bliek and Sam-Haroud [1999] through the partial path consistency algorithm (**PPC**). Later, Chmeiss and Condotta [2011] and Sioutis and Koubarakis [2012] adopted **PPC** for IA and RCC8 respectively, and Amaneddine et al. [2013] and Long and Li [2015] further extended this idea to all of the calculi discussed here.

Definition 5. A QCN $\mathcal{N} = (V, \mathcal{C})$ is *partially path consistent* (PPC) w.r.t. a graph $G = (V, E)$ iff $\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ we have that $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.

Note that PPC is the same as PC when the graph G in the definition of PPC is complete. In this paper, every considered calculus also satisfies the following property:

$$\text{Every atomic QCN over } \mathcal{M} \text{ that is PC is satisfiable.} \tag{2}$$

For a calculus with Properties (1) and (2), PPC has the same reasoning power as PC on the common edges between a triangulation and the completion of the constraint graph of a qualitative constraint network that is defined over a distributive subalgebra, in the following sense.

Proposition 3 ([Sioutis *et al.*, 2015b; Long and Li, 2015]). *Let $\mathcal{N} = (V, \mathcal{C})$ be a* QCN *that is defined over a distributive subalgebra of a qualitative calculus that satisfies (1) and (2), and $G = (V, E)$ a chordal graph such that $G_{\mathcal{N}} \subseteq G$. Then, enforcing* PPC *w.r.t. G on $\mathcal{N}$ decides the consistency of $\mathcal{N}$, and results in the same labelling of the relations that correspond to the edges of G as enforcing* PC.

We say that $\mathcal{N}$ is *minimal* if for each constraint (xRy) in $\mathcal{N}$, R is the minimal (or strongest) relation between x and y that is entailed by $\mathcal{N}$, where (xRy) is entailed by $\mathcal{N}$ if every solution of $\mathcal{N}$ satisfies (xRy).

Theorem 4 ([Long and Li, 2015; Li *et al.*, 2015]). *Let $\mathcal{S}$ be a distributive subalgebra of a qualitative calculus that satisfies* (1) *and* (2). *Then every path consistent* QCN *that is defined over $\mathcal{S}$ is minimal.*

The last two results show that given a QCN that is defined over a distributive subalgebra, enforcing PPC on it is able to make the relations corresponding to the edges of a triangulation of its constraint graph become minimal.

PC and PPC are also useful with regard to the redundancy problem. A constraint (xRy) in a QCN $\mathcal{N}$ is *redundant* if $\mathcal{N} \setminus \{(xRy)\}$ entails (xRy).

Definition 6. A QCN $= (V, \mathcal{C})$ is *all-different* if $\forall v_i \neq v_j \in V$, $\mathcal{N}$ does not entail $(v_i \mathsf{id}_{\mathcal{U}} v_j)$.

The following result regarding the redundancy problem was first shown for RCC5/8, but applies to any of the calculi considered here (cf. [Sioutis *et al.*, 2015b, Appendix]).

Theorem 5 ([Sioutis *et al.*, 2015b]). *Let $\mathcal{N} = (V, \mathcal{C})$ be a satisfiable all-different* QCN *that is defined over a distributive subalgebra satisfying* (1) *and* (2)*, and $G = (V, E)$ a chordal graph such that $G_{\mathcal{N}} \subseteq G$. Then, if $\mathcal{N}$ is* PPC *w.r.t. G, a constraint $(v_i R_{ij} v_j)$ is non-redundant in the* PC *subnetwork of $\mathcal{N}$ if and only if we have that $\{v_i, v_j\} \in E$ and $R_{ij} \neq \bigcap \{R_{ik} \diamond R_{kj} : \{v_i, v_k\}, \{v_k, v_j\} \in E\}$.*

Algorithm 1: DPC($\mathcal{N}$,α)

Input: A QCN $\mathcal{N} = (V, \mathcal{C})$ with n variables, and an ordering $\alpha = (v_n, \ldots, v_1)$ of V.
Output: True or False, a graph $G = (V, E)$, and an updated $\mathcal{N}$.

$G \leftarrow (V, E \leftarrow E(G_\mathcal{N}))$;
for v_k *from* v_n *to* v_1 **do**
  $F_k \leftarrow \{v_s \,:\, \{v_s, v_k\} \in E \wedge s < k\}$;
  foreach $v_i, v_j \in F_k$ *with* $i < j$ **do**
    if $\{v_i, v_j\} \notin E$ **then**
      $E \leftarrow E \cup \{\{v_i, v_j\}\}$;
    temp $\leftarrow R_{ij} \cap (R_{ik} \diamond R_{kj})$;
    if temp $\subset R_{ij}$ **then**
      $R_{ij} \leftarrow$ temp;
      $R_{ji} \leftarrow \text{temp}^{-1}$;
    if $R_{ij} = \emptyset$ **then**
      return (False,$G, \mathcal{N}$);
return (True,$G, \mathcal{N}$);

Algorithm 2: DPC+($\mathcal{N}$, α)

Input: A QCN $\mathcal{N} = (V, \mathcal{C})$ with n variables, and an ordering $\alpha = (v_n, \ldots, v_1)$ of V.
Output: True or False, a graph $G = (V, E)$, and an updated $\mathcal{N}$.

1 (result,$G, \mathcal{N}$) $\leftarrow$ **DPC**($\mathcal{N}$,α);
2 **if** result = False **then**
3   **return** (False,$G, \mathcal{N}$);
4 **for** v_k *from* v_1 *to* v_n **do**
5   **foreach** v_i *s.t.* $i < k$ *and* $\{v_i, v_k\} \in E(G)$ **do**
6     $F_k \leftarrow \{v_j : \{v_j, v_k\} \in E(G_\mathcal{N}) \wedge j < k\}$;
7     $R_{ik} \leftarrow \bigcap_{v_j \in F_k} R_{ij} \diamond R_{jk}$;
8     $R_{ki} \leftarrow R_{ik}^{-1}$;
9 **return** (True,$G, \mathcal{N}$);

Note that when G is a complete graph, this result falls back to the case where $\mathcal{N}$ is PC, as discussed in [Li *et al.*, 2015].

Because of these observations, it is clear that PC and PPC are quite useful in solving various reasoning tasks. It is then reasonable and important to develop an algorithm that can achieve PC or PPC as efficiently as possible.

4 New Algorithm for PPC

In this section, we show how to enforce PC or PPC on a QCN that is defined over a distributive subalgebra in an efficient manner. Before introducing our algorithm, we first take a look at a weaker local consistency condition that is, nevertheless, sufficient to decide the consistency of such a QCN.

4.1 Directional Path Consistency

Directional path consistency is another important local consistency condition, which can be used for deciding the consistency of a QCN that is defined over a distributive subalgebra. It was first proposed in the context of the Simple Temporal Problem (STP; [Dechter *et al.*, 1991]).

Definition 7. A QCN $\mathcal{N} = (V, \mathcal{C})$ is *directionally path consistent* (DPC) with respect to an ordering of its variables $\alpha = (v_1, \ldots, v_n)$ iff for all $v_i, v_k, v_j \in V$ with $i, j < k$ we have that $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.

As noted, DPC is already sufficient to decide the consistency of a QCN that is defined over a distributive subalgebra.

Proposition 6 ([Sioutis *et al.*, 2016])**.** *Let $\mathcal{N} = (V, \mathcal{C})$ be a* QCN *that is defined over a distributive subalgebra of a qualitative calculus that satisfies (1) and (2). Then, if $\mathcal{N}$ is* DPC *with respect to an ordering of its variables and does not contain an empty relation, $\mathcal{N}$ is satisfiable.*

DPC (Algorithm 1) achieves DPC by using the idea of variable elimination [Sioutis *et al.*, 2016]. It iterates variables with respect to an ordering, and propagates the constrainedness of the constraints involving a variable v_k, to the constraints involving only subsequent variables in the ordering, with the update rule $R_{ij} \leftarrow R_{ij} \cap (R_{ik} \diamond R_{kj})$. This is as if the variable v_k is "eliminated" from the QCN. As a by-product, **DPC** also triangulates the constraint graph of a QCN and produces a chordal graph G that has the ordering α as a perfect elimination ordering.

Theorem 7 ([Sioutis *et al.*, 2016])**.** *Let $\mathcal{N} = (V, \mathcal{C})$ be a* QCN *that is defined over a distributive subalgebra of a qualitative calculus satisfying (1) and (2), and $\alpha = (v_n, \ldots, v_1)$ an ordering of V. Then,* **DPC** *returns* (True, $G, \mathcal{N}'$) *if and only if $\mathcal{N}$ is satisfiable, where G is a chordal graph such that $G_\mathcal{N} \subseteq G$ and α is a perfect elimination ordering of it, and $\mathcal{N}'$ is the DPC w.r.t. α subnetwork of $\mathcal{N}$.*

4.2 The New Algorithm DPC+

Although enforcing DPC is efficient, it is not guaranteed to achieve PPC or PC and, thus, cannot solve the minimal labelling problem or assist in solving the redundancy problem of a given QCN. We hereby propose a new algorithm that achieves PPC (and PC with a simple modification) by building on the DPC enforcing algorithm **DPC**. Given a QCN $\mathcal{N}$ and an ordering $\alpha = (v_n, \ldots, v_1)$ of its variables, we first enforce DPC w.r.t. α on $\mathcal{N}$ using **DPC**. Then, we update relations by iterating the variables in reverse order. In particular, for a variable v_k (k is from 1 to n), we consider the set F_k of variables that are adjacent to v_k and preceded by v_k in α. The relation between each $v_i \in F_k$ and v_k is updated with $\bigcap_{v_j \in F_k} R_{ij} \diamond R_{jk}$. The detailed steps are shown in Algorithm 2. We call this new algorithm **DPC+**.

The following theorem shows that **DPC+** establishes PPC in a satisfiable QCN that is defined over a distributive subalgebra. Note that if we replace the graph G in line 1 with the complete graph of the same order, **DPC+** will achieve PC.

Theorem 8. *Let $\mathcal{N} = (V, \mathcal{C})$ be a* QCN *that is defined over a distributive subalgebra of a qualitative calculus that satisfies (1) and (2), and $\alpha = (v_n, \ldots, v_1)$ an ordering of V. Then,* **DPC+** *returns* (True, $G, \mathcal{N}'$) *if and only if $\mathcal{N}$ is satisfiable, where G is a chordal graph such that $G_\mathcal{N} \subseteq G$ and α is a perfect elimination ordering of it, and $\mathcal{N}'$ is the PPC w.r.t. G subnetwork of $\mathcal{N}$.*

Proof. After calling **DPC** in line 1, $\mathcal{N}$ becomes DPC w.r.t. α and we get a chordal graph G such that $G_{\mathcal{N}} \subseteq G$ and α is a perfect elimination ordering of it. In what follows, we denote by $\mathcal{N}^{(0)}$ the DPC network before applying the next steps of **DPC**+ and by $\mathcal{N}$ the updated network obtained afterwards.

Suppose that $\mathcal{N}_k$ is the partial network of the updated $\mathcal{N}$ restricted to variables $\{v_k, v_{k-1}, \ldots, v_1\}$, i.e., the updated partial network after considering $\{v_k, v_{k-1}, \ldots, v_1\}$ in the **for** loop in line 4 of **DPC**+. Note that $\mathcal{N}_n = \mathcal{N}$. It suffices to show that $\mathcal{N}_k$ is PPC given that $\mathcal{N}_{k-1}$ is PPC. To this end, we only need to consider F_k and show that $\forall v_i, v_j \in F_k$, we have $R_{ik} \neq \emptyset$, $R_{ij} \subseteq R_{ik} \diamond R_{kj}$, and $R_{ik} \subseteq R_{ij} \diamond R_{jk}$.

To simplify our proof, we first adjust the updating rule in line 7 of **DPC**+ from $R_{ik} \leftarrow \bigcap_{v_j \in F_k} R_{ij} \diamond R_{jk}$ to $R_{ik} \leftarrow \bigcap_{v_j \in F_k} R_{ij} \diamond R^{(0)}_{jk}$, where $R^{(0)}_{jk}$ is the relation between v_j and v_k in the original DPC network, viz., $\mathcal{N}^{(0)}$. We denote the adjusted algorithm by **DPC**+*. We will first prove that the conclusion holds for **DPC**+*.

We first show that $R_{ik} \neq \emptyset$ for all $v_i \in F_k$. Note that $\forall v_j, v_{j'} \in F_k$, by DPC of $\mathcal{N}^{(0)}$, we have $R_{jj'} \subseteq R^{(0)}_{jj'} \subseteq R^{(0)}_{jk} \diamond R^{(0)}_{kj'}$. By DPC of $\mathcal{N}_{k-1}$, we have $R_{jj'} \subseteq R_{ji} \diamond R_{ij'}$. Therefore, $\emptyset \neq R_{jj'} \subseteq R^{(0)}_{jk} \diamond R^{(0)}_{kj'} \cap R_{ji} \diamond R_{ij'}$. By the Peircean Law, we have $(R_{ij} \diamond R^{(0)}_{jk}) \cap (R_{ij'} \diamond R^{(0)}_{j'k}) \neq \emptyset$. According to the Helly property of distributive subalgebras, we have $R_{ik} = \bigcap_{v_j \in F_k} R_{ij} \diamond R^{(0)}_{jk} \neq \emptyset$.

Next, we show that $R_{ij} \subseteq R_{ik} \diamond R_{kj}$. Note that as $R_{jk} = \bigcap_{v_{j''} \in F_k} R_{jj''} \diamond R^{(0)}_{j''k}$, we have

$$R_{ik} \diamond R_{kj} = (\bigcap_{v_{j'} \in F_k} R_{ij'} \diamond R^{(0)}_{j'k}) \diamond (\bigcap_{v_{j''} \in F_k} R^{(0)}_{kj''} \diamond R_{j''j}).$$

By distribution of weak composition, we have

$$R_{ik} \diamond R_{kj} = \bigcap_{v_{j'} \in F_k} \bigcap_{v_{j''} \in F_k} R_{ij'} \diamond R^{(0)}_{j'k} \diamond R^{(0)}_{kj''} \diamond R_{j''j}.$$

As $\mathcal{N}^{(0)}$ is DPC, we have $R_{j'j''} \subseteq R^{(0)}_{j'j''} \subseteq R^{(0)}_{j'k} \diamond R^{(0)}_{kj''}$, and as $\mathcal{N}_{k-1}$ is PPC, we have $R_{ij} \subseteq R_{ij'} \diamond R_{j'j''} \diamond R_{j''j}$. Therefore $R_{ij} \subseteq R_{ij'} \diamond R^{(0)}_{j'k} \diamond R^{(0)}_{kj''} \diamond R_{j''j}, \forall v_{j'}, v_{j''} \in F_k$, and hence $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.

Next, we show that $R_{ik} \subseteq R_{ij} \diamond R_{jk}$. In fact,

$$\begin{aligned} R_{ik} = \bigcap_{v_{j'} \in F_k} R_{ij'} \diamond R^{(0)}_{j'k} &\subseteq \bigcap_{v_{j'} \in F_k} R_{ij} \diamond R_{jj'} \diamond R^{(0)}_{j'k} \\ &= R_{ij} \diamond (\bigcap_{v_{j'} \in F_k} R_{jj'} \diamond R^{(0)}_{j'k}) = R_{ij} \diamond R_{jk}. \end{aligned}$$

Since v_j is arbitrary in F_k and $R_{j'k} \subseteq R^{(0)}_{j'k}$, we also have

$$R_{ik} = \bigcap_{v_{j'} \in F_k} R_{ij'} \diamond R^{(0)}_{j'k} = \bigcap_{v_{j'} \in F_k} R_{ij'} \diamond R_{j'k}. \quad (3)$$

Therefore, under the updating rule of **DPC**+*, we have that $\mathcal{N}_k$ is PPC if N_{k-1} is PPC and, thus, **DPC**+* enforces PPC on $\mathcal{N}$. Regarding **DPC**+, its updating rule will update the first R_{ik} by using relations in $\mathcal{N}_{k-1}$ and $\mathcal{N}^{(0)}$, and then update the following $R_{i'k}$ by using the updated R_{ik}, and so on. By induction, we can prove that each updated R_{ik} is stronger than $R^{(0)}_{ik}$ and weaker than $\bigcap_{v_{j'} \in F_k} R_{ij'} \diamond R^*_{j'k}$, where $R^*_{j'k}$ is obtained by **DPC**+*. By (3), the relations obtained by **DPC**+ are the same as those obtained by **DPC**+*. As **DPC**+* enforces PPC on $\mathcal{N}$, we have that **DPC**+ also enforces PPC on $\mathcal{N}$. □

4.3 Analysis of the DPC+ Algorithm

As an adaptation of the **P^{3}C** algorithm in [Planken *et al.*, 2008] to qualitative spatial and temporal calculi, the **DPC**+ algorithm only needs to update each triangle in a graph at most three times. This means that the time complexity of **DPC**+ is linear in the number of triangles t in the graph, if we assume that t dominates the number of vertices and edges.

Theorem 9. *Let $\mathcal{N} = (V, \mathcal{C})$ be a* QCN*, and $\alpha = (v_n, \ldots, v_1)$ an ordering of V. Then,* **DPC**+ *returns* $(\mathsf{True}, G, \mathcal{N}')$ *in $\Theta(t + |V| + |E|)$ time when $\mathcal{N}$ is satisfiable, where $G = (V, E)$ is a chordal graph such that $G_{\mathcal{N}} \subseteq G$, and t is the number of triangles in G.*

Proof. The **DPC** algorithm considers each triangle in G exactly once. For each $\{v_i, v_k\} \in E$ such that $i < k$, lines 6–8 in **DPC**+ will consider all the triangles involving v_i and v_k once. Therefore, each triangle $\{v_i, v_j, v_k\}$ such that $i, j < k$ and $\{v_i, v_k\}, \{v_j, v_k\} \in E$ will be considered at most twice and, as such, by iterating v_k from v_1 to v_n, every triangle in the graph will be considered at most twice. Note that as **DPC**+ needs to scan through the vertices and edges, **DPC**+ runs in $\Theta(t + |V| + |E|)$ time when $\mathcal{N}$ is satisfiable. □

In terms of the maximum vertex degree Δ of G, with the above analysis, it is easy to see that **DPC**+ has a time complexity of $O(n\Delta^2)$, where n is the number of variables.

While **DPC**+ only needs to check each triangle in a graph at most three times, the state-of-the-art PPC enforcing algorithm **PPC** usually requires to check each such triangle many more times than that (e.g., as many times as $3|\mathsf{B}|$). This is also the case with the PC enforcing algorithm **PC**, since when a complete graph is used as the result of a triangulation, **PPC** falls back to **PC**. Therefore, we expect **DPC**+ to be more efficient than **PPC** and **PC**. Indeed, as shown in the following section, for large and sparsely structured networks, the advantage of **DPC**+ over **PPC** (and **PC**) is very significant.

5 Experimental Results

We evaluate the performance of our implementation of the **DPC**+ algorithm, against an implementation of the state-of-the-art PPC enforcing algorithm (**PPC**), for QCNs that are defined over a distributive subalgebra. We also employ an implementation of the state-of-the-art DPC enforcing algorithm (**DPC**) to pinpoint the overhead that **DPC**+ adds to **DPC**.

Technical Specifications. The experimentation was carried out on a computer with an Intel Core i7-2820QM processor with a 2.30 GHz frequency per CPU core, 8 GB of RAM, and the Trusty Tahr x86_64 OS. All algorithms were coded in Python and run with PyPy 2.2.1 [PyPy Team, 2015], which implements Python 2.7. Only one CPU core was used.

Datasets and Measures. We considered random RCC8 networks generated by the $\mathsf{BA}(n, m)$ model [Barabasi and Albert, 1999], the use of which in qualitative constraint-based spatial and temporal reasoning is well motivated in [Sioutis *et al.*, 2015a], and real-world RCC8 datasets that have been recently used in [Sioutis *et al.*, 2015b; Li *et al.*, 2015]

In particular, we used the $\mathsf{BA}(n, m)$ model to create random scale-free graphs as the constraint graphs of the RCC8 QCNs. We considered 10 satisfiable RCC8 networks of model $\mathsf{BA}(n, m)$ for each order $1000 \leq n \leq 10000$ of their constraint graphs with a 1000-vertex step and a preferential attachment value of $m = 2$. The edges of these graphs were labelled with relations from the maximal distributive subclass $\mathcal{D}_8^{64}$ of RCC8 [Li *et al.*, 2015]. Regarding real-world RCC8 datasets, we employed the ones recently used in [Sioutis *et al.*, 2015b; Li *et al.*, 2015], viz., **`nuts`** (nomenclature of territorial units)[2] with 2 235/3 176 variables/constraints (by *constraints* we mean non-universal relations), **`adm1`** (administrative geography of Great Britain) [Goodwin *et al.*, 2008] with 11 762/44 832 variables/constraints, **`gadm1`** (German administrative units)[2] with 42 749/159 600 variables/constraints, **`gadm2`** (the world's administrative areas) [GeoVocab, 2012] with 276 729/589 573 variables/constraints, **`adm2`** (the administrative geography of Greece)[2] with 1 732 999/5 236 270 variables/constraints, **`fprints`** (geographic "footprints" in the Southampton area of the UK) [Li *et al.*, 2015] with 3 470/446 847 variables/constraints, and **`stareas`** (statistical areas in Tasmania) [Li *et al.*, 2015] with 1 562/10 101 variables/constraints. These datasets are satisfiable. Each dataset comprises only relations that are contained in one of the maximal distributive subclasses of RCC8 [Li *et al.*, 2015].

The *maximum cardinality search* algorithm [Tarjan and Yannakakis, 1984] was used to obtain a variable elimination ordering for **DPC** and **DPC**$+$, and a triangulation of the constraint graph of a given QCN for **PPC** as described in [Sioutis *et al.*, 2015a]. We note that in our case any variable elimination ordering would be adequate for the evaluation to follow, as it would affect all involved algorithms proportionally and would not qualitatively distort the obtained result.

Our experimentation involves the following two measures. The first measure considers the number of *constraint checks* performed by a local consistency enforcing algorithm implementation. Given a QCN $\mathcal{N} = (V, \mathcal{C})$ and $v_i, v_k, v_j \in V$, a constraint check is performed when we compute relation $r = R_{ij} \cap (R_{ik} \diamond R_{kj})$ and check if $r \subset R_{ij}$, so that we can propagate its constrainedness. Weak compositions that yield relation $\star$ are disregarded. The second measure concerns the CPU time and is strongly correlated with the first one, as the runtime of these implementations relies heavily on the number of constraint checks performed.

Results. The experimental results for random scale-free RCC8 networks are summarized in Table 1, where a fraction $\frac{x}{y}$ denotes that an approach required x seconds of CPU time and performed y constraint checks on average per dataset of networks during its operation. It is clear that **DPC**$+$ performs significantly fewer constraint checks and is much faster than **PPC** for all considered networks, while retaining the good time complexity characteristics of **DPC**. In particular, **DPC**$+$ enforces PPC on a random scale-free RCC8 network of model $\mathsf{BA}(n = 10\,000, m = 2)$ in around 2 sec, when **PPC** requires approximately 8 times more time than that for the same task. Regarding real-world RCC8 datasets, the experimental results are summarized in Table 2, where a fraction $\frac{x}{y}$ has the same meaning as before. Again, we can see that **DPC**$+$ significantly outperforms **PPC** with regard to both the CPU time required and the number of constraint checks performed for all datasets (with the exception of **`nuts`**, whose constraint graph is almost a tree and, thus, neither of the algorithms is challenged enough), while adding only limited overhead to the performance of **DPC**. As illustration, **DPC**$+$ enforces PPC on the largest of the datasets (**`adm2`**) in 8.97 sec, when **PPC** requires over 44 times more time than that, viz., a total of 398.63 sec, for the same task. In general, we noted much more inference occurring with real-world datasets than with random networks.

n	**DPC**	**DPC+**	**PPC**
1000	$\frac{0.01s}{296.1}$	$\frac{0.02s}{4\,176.6}$	$\frac{0.06s}{6\,761.8}$
2000	$\frac{0.02s}{589.5}$	$\frac{0.08s}{9\,388.7}$	$\frac{0.35s}{19\,369.3}$
3000	$\frac{0.05s}{906.9}$	$\frac{0.16s}{16\,355.4}$	$\frac{0.97s}{45\,573.9}$
4000	$\frac{0.08s}{1224.8}$	$\frac{0.30s}{24\,235.5}$	$\frac{1.90s}{91\,585.4}$
5000	$\frac{0.12s}{1502.2}$	$\frac{0.55s}{31\,015.2}$	$\frac{3.49s}{108\,591.6}$
6000	$\frac{0.16s}{1811.4}$	$\frac{0.75s}{36\,247.9}$	$\frac{4.70s}{138\,285.6}$
7000	$\frac{0.21s}{2096.8}$	$\frac{0.90s}{41\,730.9}$	$\frac{6.68s}{145\,287.5}$
8000	$\frac{0.26s}{2425.5}$	$\frac{1.21s}{52\,081.0}$	$\frac{9.66s}{202\,238.4}$
9000	$\frac{0.34s}{2705.6}$	$\frac{1.62s}{58\,537.2}$	$\frac{13.30s}{244\,323.0}$
10000	$\frac{0.46s}{2985.6}$	$\frac{2.17s}{67\,388.3}$	$\frac{17.20s}{302\,867.4}$

Table 1: Evaluation with random scale-free RCC8 networks.

network	**DPC**	**DPC+**	**PPC**
`nuts`	$\frac{0.06s}{1\,180}$	$\frac{0.09s}{6\,080}$	$\frac{0.09s}{5\,808}$
`adm1`	$\frac{0.27s}{92\,266}$	$\frac{1.86s}{4\,183\,065}$	$\frac{80.03s}{18\,498\,096}$
`gadm1`	$\frac{0.79s}{339\,611}$	$\frac{4.07s}{6\,425\,394}$	$\frac{101.33s}{34\,140\,998}$
`gadm2`	$\frac{0.53s}{483\,377}$	$\frac{1.24s}{1\,534\,541}$	$\frac{1.61s}{1\,885\,100}$
`adm2`	$\frac{2.22s}{5\,471\,745}$	$\frac{8.97s}{29\,759\,133}$	$\frac{398.63s}{118\,799\,994}$
`fprints`	$\frac{2.86s}{14\,251\,630}$	$\frac{26.74s}{49\,726\,348}$	$\frac{162.72s}{119\,117\,222}$
`stareas`	$\frac{0.04s}{32\,408}$	$\frac{0.09s}{156\,360}$	$\frac{0.24s}{485\,082}$

Table 2: Evaluation with real-world RCC8 datasets.

6 Conclusion

We proposed a new algorithm, called **DPC**$+$, that enforces partial path consistency (PPC) on a qualitative constraint network w.r.t. a triangulation of its underlying constraint graph. In particular, we showed that **DPC**$+$ can correctly establish PPC in a qualitative constraint network that is defined over any distributive subalgebra of well-known spatio-temporal calculi, such as the Region Connection Calculus and the Interval Algebra. We also showed that **DPC**$+$ only needs to process the triangles in a given graph at most three times, and is therefore much more efficient than the state-of-the-art PPC enforcing algorithm; experimental results with both real-world and synthetic datasets confirm this.

[2]Retrieved from: http://www.linkedopendata.gr/

Acknowledgements

We thank the anonymous reviewers for their helpful suggestions. This work was supported by the ARC grants DP120104159 and FT0990811 and a PhD grant from the University of Artois and the Nord–Pas-de-Calais region.

References

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

[Amaneddine *et al.*, 2013] 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.

[Balbiani *et al.*, 2002] P. Balbiani, J.-F. Condotta, and L. F. Del Cerro. Tractability results in the block algebra. *J. Log. Comput.*, 12(5):885–909, 2002.

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

[Bhatt *et al.*, 2009] M. Bhatt, F. Dylla, and J. Hois. Spatio-terminological inference for the design of ambient environments. In *Spatial Information Theory*, pages 371–391. Springer, 2009.

[Blair and Peyton, 1993] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.

[Bliek and Sam-Haroud, 1999] C. Bliek and D. Sam-Haroud. Path consistency on triangulated constraint graphs. In *IJCAI*, 1999.

[Chmeiss and Condotta, 2011] A. Chmeiss and J.-F. Condotta. Consistency of triangulated temporal qualitative constraint networks. In *ICTAI*, 2011.

[Cohen *et al.*, 2012] D. A. Cohen, M. C. Cooper, P. Creed, D. Marx, and A. Z. Salamon. The tractability of CSP classes defined by forbidden patterns. *JAIR*, pages 47–78, 2012.

[Cohn and Renz, 2008] A. G. Cohn and J. Renz. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*. Elsevier, 2008.

[Danzer *et al.*, 1963] L. Danzer, B. Grünbaum, and V. Klee. Helly's Theorem and Its Relatives. In *Convexity*, Proceedings of Symposia in Pure Mathematics. American Mathematical Society, 1963.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *AIJ*, 49:61–95, 1991.

[Düntsch, 2005] I. Düntsch. Relation algebras and their application in temporal and spatial reasoning. *JAIR*, 23:315–357, 2005.

[Dylla *et al.*, 2013] F. Dylla, Till M., T. Schneider, and D. Wolter. Algebraic Properties of Qualitative Spatio-Temporal Calculi. In *COSIT*, 2013.

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

[GeoVocab, 2012] GeoVocab. http://gadm.geovocab.org/, 2012.

[Goodwin *et al.*, 2008] J. Goodwin, C. Dolbear, and G. Hart. Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. *T. GIS*, 12(Issue Supplement s1):19–30, 2008.

[Li *et al.*, 2015] S. Li, Z. Long, W. Liu, M. Duckham, and A. Both. On Redundant Topological Constraints. *AIJ*, 225:51–78, 2015.

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

[Ligozat, 1998] G. Ligozat. Reasoning about Cardinal Directions. *J. Vis. Lang. Comput.*, 9:23–44, 1998.

[Long and Li, 2015] Z. Long and S. Li. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*, 2015.

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

[Planken *et al.*, 2008] L. Planken, M. de Weerdt, and R. van der Krogt. P^3C: A New Algorithm for the Simple Temporal Problem. In *ICAPS*, 2008.

[PyPy Team, 2015] PyPy Team. http://pypy.org/, 2015.

[Randell *et al.*, 1992] D. A. Randell, Z. Cui, and A. G. Cohn. A Spatial Logic based on Regions and Connection. In *KR*, 1992.

[Sioutis and Koubarakis, 2012] M. Sioutis and M. Koubarakis. Consistency of Chordal RCC-8 Networks. In *ICTAI*, 2012.

[Sioutis *et al.*, 2015a] M. Sioutis, J.-F. Condotta, and M. Koubarakis. An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks. *Int. J. Artif. Intell. Tools*, 2015. In press.

[Sioutis *et al.*, 2015b] M. Sioutis, S. Li, and J.-F. Condotta. Efficiently Characterizing Non-Redundant Constraints in Large Real World Qualitative Spatial Networks. In *IJCAI*, 2015.

[Sioutis *et al.*, 2016] M. Sioutis, Z. Long, and S. Li. Efficiently Reasoning about Qualitative Constraints through Variable Elimination. In *SETN*, 2016.

[Tarjan and Yannakakis, 1984] 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.*, 13:566–579, 1984.

[van Beek and Cohen, 1990] P. van Beek and R. Cohen. Exact and approximate reasoning about temporal relations. *Computational intelligence*, 6:132–147, 1990.

[van Beek, 1989] P. van Beek. Approximation Algorithms for Temporal Reasoning. In *IJCAI*, 1989.

[Vilain and Kautz, 1986] M. Vilain and H. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *AAAI*, pages 377–382, 1986.

[Wallgrün, 2012] J. O. Wallgrün. Exploiting qualitative spatial reasoning for topological adjustment of spatial data. In *SIGSPATIAL*, 2012.

[Wolter, 2008] D. Wolter. *Spatial Representation and Reasoning for Robot Mapping: A Shape-based Approach*. Springer, 2008.

[Xu and Choueiry, 2003] L. Xu and B. Y. Choueiry. A new effcient algorithm for solving the simple temporal problem. In *TIME*, 2003.