Qualitative Constraint Satisfaction Problems: An Extended Framework with Landmarks ☆

Sanjiang Li[a], Weiming Liu[a,*], Shengsheng Wang[b]

[a] Centre for Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
[b] College of Computer Science and Technology, Jilin University, Changchun, China

Abstract

Dealing with spatial and temporal knowledge is an indispensable part of almost all aspects of human activity. The qualitative approach to spatial and temporal reasoning, known as Qualitative Spatial and Temporal Reasoning (QSTR), typically represents spatial/temporal knowledge in terms of qualitative relations (e.g., *to the east of*, *after*), and reasons with spatial/temporal knowledge by solving qualitative constraints.

When formulating qualitative constraint satisfaction problems (CSPs), it is usually assumed that each variable could be "*here, there and everywhere*[1]." Practical applications such as urban planning, however, often require a variable to take its value from a certain finite domain, i.e. it is required to be '*here or there, but not everywhere*'. Entities in such a finite domain often act as reference objects and are called "landmarks" in this paper. The paper extends the classical framework of qualitative CSPs by allowing variables to take values from finite domains. The computational complexity of the consistency problem in this extended framework is examined for the five most important qualitative calculi, viz. Point Algebra, Interval Algebra, Cardinal Relation Algebra, RCC5, and RCC8. We show that all these consistency problems remain in NP and provide, under practical assumptions, efficient algorithms for solving basic constraints involving landmarks for all these calculi.

☆This is an extended and revised version of two conference papers [29, 27] presented at CP-2011 and ECAI-2012.

*Corresponding Author
Email addresses: `sanjiang.li@uts.edu.au` (Sanjiang Li), `weiming.liu@uts.edu.au` (Weiming Liu), `wss@jlu.edu.cn` (Shengsheng Wang)

[1] A song by The Beatles from the album *Revolver* (1966).

Keywords: qualitative spatial and temporal reasoning; qualitative calculi; constraint satisfaction; landmarks

1. Introduction

Spatial and temporal information is pervasive and forms an increasing part of our everyday life. Many tasks in the real or virtual world require sophisticated spatial and temporal reasoning abilities. Furthermore, the rapid progress in science and technology in this century continues to present new challenges for spatial and temporal reasoning. Taking spatial information as an example, on one hand, people can now easily acquire location information with the help of GPS-enabled mobile equipment and web GISs such as Google Maps. This has greatly increased the public's demand for location-based services. On the other hand, the development of technologies such as remote sensing, medical imaging, and sensor networks has generated volumes of spatial data, which makes the phenomenon of '*rich data but poor knowledge*' particularly serious in the area of spatial knowledge management.

The qualitative approach to spatial and temporal reasoning, known as Qualitative Spatial and Temporal Reasoning (QSTR), has the potential to resolve the conflict between data and knowledge. This is because the main aims of QSTR research are to design (i) human comprehensible and cognitively plausible spatial relation models (or query languages), and (ii) efficient algorithms for consistency checking (or query preprocessing). For intelligent systems, the ability to understand and process qualitative, vague or even inconsistent (textual, graphical or speech) information collected from human beings or the Web is very important. This is because 'the input and the output of spatial processes is often qualitative rather than quantitative' [36].

QSTR represents spatial/temporal information in terms of human comprehensible qualitative relations (e.g. *partially overlaps*, *west of*, *after*) and reduces spatial/temporal reasoning to solving constraint satisfaction problems (CSPs). Qualitative relations are widely used in temporal and spatial reasoning (see e.g. [1, 38, 24]). This is partially because they are close to the way humans represent and reason about commonsense knowledge, are easy to specify, and provide a flexible way to deal with incomplete knowledge. Usually, these relations are taken from a qualitative calculus, which is a finite set of relations defined on an infinite universe U of entities [25]. Well-known qualitative calculi include among others Point Algebra (PA) [44], Interval Algebra (IA) [1], Cardinal Relation Algebra (CRA) [24], RCC5, and RCC8 [38, 24].

A central reasoning problem of QSTR is the *consistency problem*. An instance of the consistency problem is a set Γ of constraints like $(x\alpha y)$, where x, y are variables taken from a finite set V, and α is a qualitative relation. Unlike classical CSPs, the domain of a variable appearing in a qualitative constraint is usually infinite, and Hirsch [16] has shown that it may be undecidable for determining consistency for binary CSPs with infinite domains. However, for the five qualitative calculi that we have mentioned above, the consistency problems are all in NP and can be solved by using path consistency and backtracking (cf. [7, 41]).

In the past three decades, QSTR has made significant progress, and prominent qualitative calculi such as IA and RCC8 have been applied in areas such as natural language processing, geographical information systems, robotics, and content-based image retrieval (see e.g. [7]). There is a growing consensus, however, that breakthroughs are necessary to bring spatial/temporal reasoning theory closer to practical applications. One reason might be that the current qualitative reasoning scheme uses a rather restricted constraint language: constraints in a qualitative CSP are always taken from the *same* calculus and only relate variables from the same *infinite* domain. This is highly undesirable, as constraints involving restricted variables and/or multiple aspects of information frequently appear in practical tasks such as urban planning and spatial query processing.

Consider the following example. Suppose you are recommended a restaurant in Sydney by a friend who has dined there before. The spatial information about the restaurant may be similar to "it is *in* downtown and *close to* a MacDonald's, and it is to the *west of* or *southwest of* Central Station." In this example, topological, directional, and distance information appears together. While the position of the restaurant may be completely unknown, the position of Central Station is fixed as a landmark, and the position of downtown is also fixed somehow, but the position of "MacDonald's" is only *finitely fixed* because there are several branches of MacDonald's in downtown Sydney.

While some recent works have considered how to reason with qualitative constraints from different spatial or temporal calculi [13, 20, 28, 45, 21], the importance of solving constraints that involve restricted variables has been totally neglected. Cohn and Renz regarded this as a major future challenge, and commented in their chapter [7, page 578] in "*Handbook of Knowledge Representation*" that

> One problem with this [constraint-based] approach is that spatial entities are treated as variables which have to be instantiated using values of an infinite domain. How to integrate this with settings where some spatial entities are known or can only be from a small domain is still

unknown and is one of the main future challenges of constraint-based spatial reasoning.

This paper aims to address the above challenge. We say that a variable is *finitely restricted* if it can only take its value from a finite subset of the universe in a qualitative calculus. We propose to extend the qualitative CSP framework by allowing variables to be finitely restricted. In such a qualitative CSP, the constraints are taken from a fixed qualitative calculus, and the domain of each variable is either the universe of the calculus or a (nonempty) finite subset of the universe. The entities in each finite domain usually act as reference spatial/temporal objects in the constraint network. In this paper, we address these entities as "*landmarks*".

Landmarks (e.g. *Sydney Opera House* or *Big Ben*) are external, outstanding physical objects that act as reference objects. As found in many spatial discourses, landmarks play a fundamental role in cognitive spatial representations, in particular in human navigation and route planning. There are many works in geographical information science that are devoted to characterising or generating landmarks. Lynch [31] is perhaps the first such attempt, which although informal is very influential. Grabler et al. [14] developed a system to generate tourist maps enriched with landmarks. Duckham, Winter, and Robinson [11] considered how to incorporate cognitively salient landmarks in computer-generated navigation instructions. Landmarks are also used as a metaphor in automatic planning, where a landmark acts as an auxiliary sub-goal [15, 42].

In this paper, landmarks are used as reference objects for formulating constraints. This is related to but different from Allen's 'reference intervals' [1], which are used to group clusters of intervals, and the intervals in one cluster are related to intervals outside the cluster only indirectly via the reference intervals.

An important research question is, *how does this extension affect the computational complexity of deciding the consistency of qualitative CSPs*? This paper examines the effect for the five most important qualitative calculi, viz. PA, IA, CRA, RCC5 and RCC8. We show that in the extended framework the consistency problem remains in NP for each calculus. Moreover, we propose practical efficient algorithms for solving basic constraints involving landmarks for these qualitative calculi.

The remainder of this paper proceeds as follows. Section 2 introduces basic notions in qualitative constraint solving as well as the five qualitative calculi discussed in this paper. The extended qualitative CSP framework is also presented there. Section 3 discusses the computational complexity of reasoning with the point-based calculi PA, IA, and CRA, and Section 4 considers the same problem for the region-based calculi RCC5 and RCC8. The last section concludes the paper and outlines problems for future study.

2. Preliminaries

In this section, we first recall several well-known qualitative calculi and basic notions in qualitative constraint solving, and then introduce the extended qualitative CSP framework.

2.1. Qualitative calculi

The qualitative approach to spatial and temporal knowledge representation and reasoning is mainly based on qualitative calculi. In this paper, we only consider binary relations, but the extended qualitative CSP framework can be straightforwardly extended to ternary and any n-ary relations.

Suppose U is the universe of spatial or temporal entities. Write $\mathbf{Rel}(U)$ for the Boolean algebra of binary relations on U. A *qualitative calculus* on U is defined as a finite Boolean subalgebra of $\mathbf{Rel}(U)$. Let $\mathcal{M}$ be a qualitative calculus on U. A relation α in $\mathcal{M}$ is a *basic* relation if it is an atom in $\mathcal{M}$. We write $\mathcal{B}_{\mathcal{M}}$ for the set of basic relations in $\mathcal{M}$.

We next recall the well-known Point Algebra (PA) [44, 43], Cardinal Relation Algebra (CRA) [12, 24], Interval Algebra (IA) [1], and RCC5 and RCC8 [38].

Definition 1 (Point Algebra [44])**.** *Let U be the set of real numbers. The Point Algebra is the Boolean subalgebra generated by the jointly exhaustive and pairwise disjoint (JEPD) set of relations $\{<,>,=\}$, where $<,>,=$ are defined as usual.*

PA contains eight relations, viz. the three basic relations $<,>,=$, the empty relation, and the four non-basic nonempty relations $\leq,\geq,\neq,?$, where $?$ stands for the universal relation.

Definition 2 (Cardinal Relation Algebra [12, 24])**.** *Let U be the real plane. Define binary relations $NW, N, NE, W, EQ, E, SW, S, SE$ as in Table 1. The Cardinal Relation Algebra (CRA) is generated by these nine JEPD relations.*

CRA can be viewed as the Cartesian product of two PAs.

Definition 3 (Interval Algebra [1])**.** *Let U be the set of closed intervals on the real line. Thirteen binary relations between two intervals $x = [x^-, x^+]$ and $y = [y^-, y^+]$ are defined by the order of the four endpoints of x and y, see Table 2. The Interval Algebra (IA) is generated by these JEPD relations.*

Relation	Definition
NW	$x < x', y > y'$
N	$x = x', y > y'$
NW	$x > x', y > y'$
W	$x < x', y = y'$
EQ	$x = x', y = y'$
E	$x > x', y = y'$
SW	$x < x', y < y'$
S	$x = x', y < y'$
SW	$x > x', y < y'$

Table 1: Basic relations of CRA.

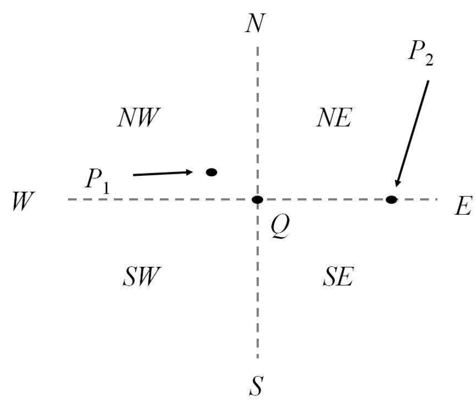


Figure 1: Examples: P_1 NW Q and P_2 E Q

Relation	Symbol	Converse	Definition
before	b	bi	$x^- < x^+ < y^- < y^+$
meets	m	mi	$x^- < x^+ = y^- < y^+$
overlaps	o	oi	$x^- < y^- < x^+ < y^+$
starts	s	si	$x^- = y^- < x^+ < y^+$
during	d	di	$y^- < x^- < x^+ < y^+$
finishes	f	fi	$y^- < x^- < x^+ = y^+$
equals	eq	eq	$x^- = y^- < x^+ = y^+$

Table 2: Basic IA relations and their converses, where $x = [x^-, x^+], y = [y^-, y^+]$ are two intervals.

Relation	Definition	Relation	Definition
DC	$a \cap b = \varnothing$	TPP	$a \subset b$, $a \not\subset b^\circ$
EC	$a \cap b \neq \varnothing$, $a^\circ \cap b^\circ = \varnothing$	NTPP	$a \subset b^\circ$
PO	$a \nsubseteq b$, $b \nsubseteq a$, $a^\circ \cap b^\circ \neq \varnothing$	EQ	$a = b$

Table 3: Topological interpretation of basic RCC8 relations in the plane, where a, b are plane regions, and a°, b° are the interiors of a, b, respectively.

Unlike the above qualitative calculi, the RCC algebras have interpretations in arbitrary topological spaces. Since applications in GIS and many other spatial reasoning tasks mainly consider objects represented in the real plane, in this paper, we only consider interpretations where regions are interpreted as nonempty regular closed sets, and two regions are connected if they somehow intersect.[2]

Definition 4 (RCC5 and RCC8 Algebras)**.** *Let U be the set of nonempty regular closed sets, or* regions, *in the real plane. The RCC8 algebra is generated by the eight topological relations*

$$\mathsf{DC}, \mathsf{EC}, \mathsf{PO}, \mathsf{EQ}, \mathsf{TPP}, \mathsf{NTPP}, \mathsf{TPPi}, \mathsf{NTPPi},$$

where DC, EC, PO, TPP and NTPP are defined in Table 3, EQ is the identity relation, and TPPi and NTPPi are the converses of TPP and NTPP respectively. See Figure 2 for illustration. It is worth mentioning that these eight relations are all definable by the connectedness relation C, which is the complement of DC and two regions are connected if they have nonempty intersection.

The RCC5 algebra is the sub-algebra of RCC8 generated by the five part-whole relations

$$\mathsf{DR}, \mathsf{PO}, \mathsf{EQ}, \mathsf{PP}, \mathsf{PPi},$$

where $\mathsf{DR} = \mathsf{DC} \cup \mathsf{EC}$, $\mathsf{PP} = \mathsf{TPP} \cup \mathsf{NTPP}$, and $\mathsf{PPi} = \mathsf{TPPi} \cup \mathsf{NTPPi}$.

While the RCC algebras defined as above using a ‘weak’ connectedness relation, we will introduce another interpretation in Section 4.4.3 based on a ‘strong’ connectedness relation.

2.2. Qualitative constraint satisfaction problem

A qualitative calculus $\mathcal{M}$ provides a constraint language by using formulas of the form $(v_i \alpha v_j)$, where α is a relation in $\mathcal{M}$ and v_i, v_j are variables taking values

[2]We note that restricting the underlying topological space may drastically change the computational properties of calculi like RCC8 [4, 39].

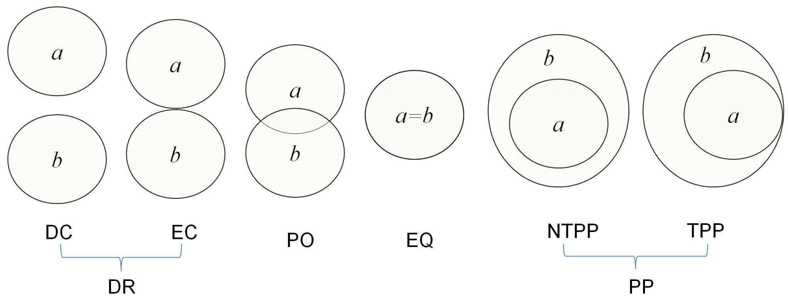


Figure 2: Illustration for basic relations in RCC5 / RCC8

from the universe of $\mathcal{M}$. Formulas of the form $(v_i \alpha v_j)$ are called *constraints* (over $\mathcal{M}$). If α is a basic relation in $\mathcal{M}$, $(v_i \alpha v_j)$ is called a *basic constraint*. The classical consistency problem over $\mathcal{M}$ can then be formulated as below.

Definition 5. *[7] Let $\mathcal{M}$ be a qualitative calculus on universe U. Suppose $\mathcal{S}$ is a subset of $\mathcal{M}$. The consistency problem* CSPSAT*($\mathcal{S}$) is defined as follows:*

> ***Instance****: A 2-tuple (V, Γ). Here V is a finite set of variables $\{v_1, v_2, \ldots, v_n\}$, and $\Gamma = \{v_i \gamma_{ij} v_j : 1 \leq i, j \leq n\}$ is a binary constraint network and each γ_{ij} is in $\mathcal{S}$.*
>
> ***Question****: Is there an instantiation $\nu : V \to U$ such that all the constraints in Γ are satisfied?*

If ν satisfies all the constraints in Γ, then we say ν is a solution of Γ and say Γ is consistent *or* satisfiable.

Notation. In this paper, we also represent an instantiation $\nu : V \to U$ as an n-tuple $(\nu(v_1), \nu(v_2), \ldots, \nu(v_n))$.

We note that each instance (V, Γ) in CSPSAT($\mathcal{S}$) is *complete* in the sense that the relation γ_{ij} between any two variables v_i, v_j is taken from $\mathcal{S}$. Given a binary constraint work $\Gamma = \{v_i \gamma_{ij} v_j : 1 \leq i, j \leq n\}$, we say Γ is a *basic constraint network* if γ_{ij} is either the universal relation or a basic relation for any two variables v_i, v_j; and say Γ is a *complete basic constraint network* if γ_{ij} is a basic relation for any two variables v_i, v_j. In other words, each complete basic constraint network is an instance of CSPSAT($\mathcal{B}_{\mathcal{M}}$), while each basic constraint network is an instance of CSPSAT($\mathcal{B}_{\mathcal{M}} \cup \{\star_{\mathcal{M}}\}$), where $\mathcal{B}_{\mathcal{M}}$ is the set of basic relations in $\mathcal{M}$, and $\star_{\mathcal{M}}$ is the universal relation of $\mathcal{M}$. [3]

The consistency problem as defined in Definition 5 has been investigated for many calculi (see e.g. [1, 43, 35, 40, 30, 26]). In particular, the consistency problem CSPSAT(PA) can be solved in $O(n^2)$ time, where n is the number of variables [43]. For most other qualitative calculi, including IA, CRA, RCC5, and RCC8, the consistency problem CSPSAT($\mathcal{M}$) is NP-complete.

When only basic constraint networks are considered, however, the consistency problem over each of these four calculi becomes tractable. In fact, it can be decided by checking whether the network is *path-consistent*. For binary relations α and β, we write $\alpha^\sim$ for the converse of α, and $\alpha \circ \beta$ for the usual composition of α and β. We say a complete basic constraint network $\Gamma = \{v_i \alpha_{ij} v_j : 1 \leq i, j \leq n\}$ is *path-consistent*, if for any three variables v_i, v_j, v_k, we have[4]

$$\alpha_{ij} = \alpha_{ji}^\sim \quad \text{and} \quad \alpha_{ij} \cap (\alpha_{ik} \circ \alpha_{kj}) \neq \varnothing \quad \text{for any} \quad 1 \leq i, j, k \leq n.$$

Note that for complete basic constraint networks, path-consistency is equivalent to saying that every subnetwork with three variables is consistent. As a local property, path-consistency can be enforced in cubic time.

We summarise the computational complexity results of these calculi in the following theorem.

Theorem 1. *[35, 24, 40] The consistency problem* CSPSAT(PA) *is in P. Let $\mathcal{M}$ be one of IA, CRA, RCC5, and RCC8. Then* CSPSAT$(\mathcal{B}_\mathcal{M})$ *is in P and* CSPSAT$(\mathcal{M})$ *is NP-complete.*

A complete basic network is *globally consistent* if any partial solution can be extended to a global solution. The following theorem can be directly proven by exploiting the density of real numbers.

Theorem 2. *Let $\mathcal{M}$ be one of PA, IA, and CRA. Then a complete basic network is globally consistent if it is path-consistent.*

We note that RCC5 and RCC8 do not have this property.

---

[3]The consistency problems CSPSAT($\mathcal{B}_\mathcal{M}$) and CSPSAT($\mathcal{B}_\mathcal{M} \cup \{\star_\mathcal{M}\}$) may have different complexities. For example, there exists a cubic algorithm for solving complete basic CDC (cardinal direction calculus) networks [30], but it is NP-hard to solve basic CDC networks [26].

[4]This definition of path-consistency is different from the same notion for finite CSPs [33, 32].

2.3. Extended qualitative CSP

By Definition 5, in the classical consistency problem, each variable can in principle take any value in the universe. In many practical applications, however, it is very common to have additional knowledge about some variables (cf. the restaurant and MacDonald's example in the Introduction), which will affect the consistency of qualitative CSPs. It is therefore necessary to extend the qualitative CSP framework to allow restricted domains of variables.

Definition 6. *Let $\mathcal{M}$ be a qualitative calculus on universe U. Suppose $\mathcal{S}$ is a subset of $\mathcal{M}$. The consistency problem* $\textsc{cspsat}_f(\mathcal{S})$ *is defined as follows, where the subscript 'f' stands for 'finite':*

> ***Instance****: A 3-tuple $(V, \Gamma, \boldsymbol{D})$. Here V is a finite set of variables $\{v_1, v_2, \ldots, v_n\}$, $\boldsymbol{D}$ is an n-tuple $(D_1, D_2, \ldots, D_n)$, where each D_i is either U or a nonempty finite subset of U, and $\Gamma = \{v_i \gamma_{ij} v_j : 1 \le i, j \le n\}$ is a binary constraint network and each γ_{ij} is in $\mathcal{S}$.*
>
> ***Question****: Is there an instantiation $\nu : V \to U$ such that $\nu(v_i) \in D_i$ for each i and all the constraints in Γ are satisfied?*

We say that a variable v_i appearing in the instance $(V, \Gamma, \boldsymbol{D})$ is finitely restricted *if its domain D_i is finite. If ν satisfies all the constraints in Γ and $\nu(v_i) \in D_i$ for each i, then we say ν is a solution of $(V, \Gamma, \boldsymbol{D})$ and say $(V, \Gamma, \boldsymbol{D})$ is* consistent *or* satisfiable. *We call elements of each finite domain D_i* landmarks *of $(V, \Gamma, \boldsymbol{D})$.*

As a special case, if each finite domain D_i is required to be a singleton, we write the corresponding consistency problem $\textsc{cspsat}_s(\mathcal{S})$, *where the subscript '$s$' denotes 'singleton'.*

An instance of $\textsc{cspsat}(\mathcal{S})$ is clearly an instance of both $\textsc{cspsat}_s(\mathcal{S})$ and $\textsc{cspsat}_f(\mathcal{S})$: we only need to let each D_i be the universe.

Proposition 1. *Suppose $\mathcal{B}_\mathcal{M}$ is the set of basic relations in a qualitative calculus $\mathcal{M}$, and $\mathcal{S}$ is a subclass of $\mathcal{M}$. Then we have*

i) $\textsc{cspsat}(\mathcal{S}) \subset \textsc{cspsat}_s(\mathcal{S}) \subset \textsc{cspsat}_f(\mathcal{S})$;

ii) $\textsc{cspsat}_f(\mathcal{M})$ *is in NP if* $\textsc{cspsat}_f(\mathcal{B}_\mathcal{M})$ *is in NP;*

iii) $\textsc{cspsat}_f(\mathcal{S})$ *is in NP if* $\textsc{cspsat}_s(\mathcal{S})$ *is in NP;*

iv) $\textsc{cspsat}_f(\mathcal{M})$ *is in NP if* $\textsc{cspsat}_s(\mathcal{B}_\mathcal{M})$ *is in NP.*

Proof. i) follows directly from the definition. As for ii), suppose we already have a nondeterministic Turing machine T_0 which solves $\textsc{cspsat}_f(\mathcal{B}_\mathcal{M})$ in polynomial time. Given a non-basic constraint network $(V, \Gamma, \mathbf{D})$, it is consistent iff there is a consistent basic constraint network Γ' that refines Γ in the sense that for each constraint $(x\alpha y)$ in Γ there exists a constraint $(x\alpha' y)$ in Γ' such that $\alpha' \subseteq \alpha$. A basic constraint network that refines Γ is often called a *scenario* of Γ. We devise a nondeterministic Turing machine T as follows. T first guesses a scenario $(V, \Gamma', \mathbf{D})$ of $(V, \Gamma, \mathbf{D})$, and then calls T_0 to decide the consistency of $(V, \Gamma', \mathbf{D})$. T' asserts the instance to be consistent if T returns consistent in any branch. It is clear that the nondeterministic Turing machine T decides the consistency of $(V, \Gamma, \mathbf{D})$ in polynomial time. Similar argument applies to iii), and iv) follows from ii) and iii) directly. □

By the above proposition, the computational complexity of $\textsc{cspsat}_f$ is in general higher than that of $\textsc{cspsat}_s$ and $\textsc{cspsat}$, as far as the same subset $\mathcal{S}$ of the same calculus is considered. In particular, recall that the classical consistency problems for CRA, IA, RCC5 and RCC8 are all NP-complete. We have the following corollary.

Corollary 1. *The consistency problem* $\textsc{cspsat}_s(\mathcal{M})$ *and* $\textsc{cspsat}_f(\mathcal{M})$ *are all NP-hard for* $\mathcal{M}$ *being any one of IA, CRA, RCC5, and RCC8.*

To determine the computational complexity of reasoning with a qualitative calculus $\mathcal{M}$, we will begin with $\textsc{cspsat}_s(\mathcal{B}_\mathcal{M})$.

Our computational complexity results are summarised in Table 4, where qualitative calculus $\mathcal{M}$ is PA, IA, CRA, RCC5 or RCC8, and $\mathcal{S}$ is either $\mathcal{B}_\mathcal{M}$ or $\mathcal{M}$ itself (i.e., we consider either complete basic networks or the most general case).

$\mathcal{M}$	PA		CRA		IA		RCC5		RCC8	
$\mathcal{S}$	$\mathcal{B}_{PA}$	PA	$\mathcal{B}_{CRA}$	CRA	$\mathcal{B}_{IA}$	IA	$\mathcal{B}_{RCC5}$	RCC5	$\mathcal{B}_{RCC8}$	RCC8
$\textsc{cspsat}(\mathcal{S})$	P	P	P	NP-C	P	NP-C	P	NP-C	P	NP-C
$\textsc{cspsat}_s(\mathcal{S})$	P	P	P	NP-C	P	NP-C	P	NP-C	NP-C	NP-C
$\textsc{cspsat}_f(\mathcal{S})$	P	NP-C	NP-C	NP-C	NP-C	NP-C	NP-C	NP-C	NP-C	NP-C

Table 4: Computational complexity results summary

In the following sections, we first consider point-based calculi PA, CRA, and IA, and then consider region-based calculi RCC5 and RCC8. Unlike point-based calculi, the *geometrical representation* (in particular, shape and location) of the landmarks may affect the existence of solutions in the plane. To make the analysis more meaningful, we assume that all the landmarks in RCC5 and RCC8 constraint networks are represented as polygons which may have different connected components and holes. This assumption is practical because polygons are the most widely used approximations of regions in spatial databases.

The NP-hardness results in Table 4 obtained in this paper are mainly achieved by designing polynomial reductions from the Graph 3-Colouring problem, which is a well-known NP-complete problem. Recall that a graph $G = (V, E)$ is *3-colourable* if there is a function $f : V \to \{0, 1, 2\}$ such that $f(v) \neq f(v')$ for each edge $(v, v') \in E$. The *Graph 3-Colouring problem* is to decide whether a graph is 3-colourable.

3. Point-based Qualitative Calculi

This section discusses the consistency problems in the extended framework for the three point-based qualitative calculi, viz. Point Algebra, Interval Algebra, and Cardinal Relation Algebra.

3.1. Some simple results

To prove the computational complexity results, we will need the following notion of a finitely restricted sub-instance.

Definition 7. *Let $\mathcal{M}$ be a qualitative calculus with universe U, and let $\mathcal{S}$ be a subclass of $\mathcal{M}$. Suppose $(V, \Gamma, \boldsymbol{D})$ is an instance of* $\textsc{cspsat}_f(\mathcal{S})$*, where $V = \{v_1, \ldots, v_n\}$, $\boldsymbol{D} = (D_1, \ldots, D_n)$ and $\Gamma = \{v_i \alpha_{ij} v_j\}_{1 \le i,j \le n}$. Let $V' = \{v_i : D_i \neq U\}$ be the set of finitely restricted variables in V. Suppose $V' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. Let $\Gamma' = \{v_{i_r} \alpha_{i_r i_s} v_{i_s}\}_{1 \le r,s \le k}$ and $\boldsymbol{D}' = (D_{i_1}, D_{i_2}, \ldots, D_{i_k})$. We call $(V', \Gamma', \boldsymbol{D}')$, which is also an instance of* $\textsc{cspsat}_f(\mathcal{S})$*, the* finitely restricted sub-instance *of $(V, \Gamma, \boldsymbol{D})$.*

For complete basic constraint networks, we have the following general result.

Lemma 1. *Let $\mathcal{M}$ be one of PA, IA, and CRA. Suppose $(V, \Gamma, \boldsymbol{D})$ is an instance of* $\textsc{cspsat}_f(\mathcal{B}_{\mathcal{M}})$*. Then $(V, \Gamma, \boldsymbol{D})$ is consistent iff Γ is path-consistent and the finitely restricted sub-instance of $(V, \Gamma, \boldsymbol{D})$ is consistent.*

Proof. The necessity is clear. We prove the sufficiency, which uses the property that any consistent basic PA (IA or CRA) network is also globally consistent.

Because the finitely restricted sub-instance $(V', \Gamma', \mathbf{D}')$ is consistent, it has a solution, say $\mathfrak{b} = (b_1, \ldots, b_k)$. Note that $\mathfrak{b}$ is a partial solution of the $\textsc{cspsat}(\mathcal{B}_{PA})$ instance (V, Γ), and thus, by Theorem 2, can be extended to a solution $\mathfrak{b}'$ of (V, Γ). It is clear that $\mathfrak{b}'$ is also a solution of $(V, \Gamma, \mathbf{D})$. Therefore $(V, \Gamma, \mathbf{D})$ is consistent.

The cases for IA and CRA can be proven in the same way. □

Using Lemma 1, we are able to show the following computational complexity results.

Theorem 3. *For PA, we have* $\textsc{cspsat}_s(\mathcal{B}_{PA})$ *and* $\textsc{cspsat}_s(PA)$ *are in P and* $\textsc{cspsat}_f(PA)$ *is in NP. Let* $\mathcal{M}$ *be IA or CRA. Then* $\textsc{cspsat}_s(\mathcal{B}_{\mathcal{M}})$ *is in P, and* $\textsc{cspsat}_s(\mathcal{M})$ *and* $\textsc{cspsat}_f(\mathcal{M})$ *are NP-complete.*

Proof. For PA, we recall that $\textsc{cspsat}(PA)$ can be solved in $O(n^2)$ time [43]. Suppose $(V, \Gamma, \mathbf{D})$ is an instance of $\textsc{cspsat}_s(PA)$. We show that the consistency of $(V, \Gamma, \mathbf{D})$ can be determined in polynomial time. For a pair of variables v_i and v_j such that $D_i = \{d_i\}$ and $D_j = \{d_j\}$ are both singletons, suppose $(v_i \alpha v_j)$ is in Γ, and β is the basic PA relation between d_i and d_j. It is clear that $(V, \Gamma, \mathbf{D})$ is inconsistent if β is not included in α. Without loss of generality, we assume α is a basic relation and $\alpha = \beta$. Under this assumption, we show that $(V, \Gamma, \mathbf{D})$ is consistent iff the $\textsc{cspsat}(PA)$ instance (V, Γ') is consistent. The necessity is clear. For the sufficiency, suppose (V, Γ) is consistent and has a consistent scenario (V, Γ_0). Note that the finitely restricted sub-instance of $(V, \Gamma_0, \mathbf{D})$ is consistent, as the constraint between any two variables with a singleton domain is the actual relation between the corresponding landmarks. By Lemma 1, we have $(V, \Gamma, \mathbf{D})$ is consistent. Because the consistency of (V, Γ) can be decided in polynomial time [43], we know that $\textsc{cspsat}_s(PA)$ is in P and consequently $\textsc{cspsat}_s(\mathcal{B}_{PA})$ is in P and $\textsc{cspsat}_f(PA)$ is in NP. [5]

For $\mathcal{M}$ being IA or CRA, suppose $(V, \Gamma, \mathbf{D})$ is an instance of $\textsc{cspsat}_s(\mathcal{B}_{\mathcal{M}})$, and $(V', \Gamma', \mathbf{D}')$ is its finitely restricted sub-instance. Assume that V has n variables and V' has $m \leq n$ variables. The path-consistency of Γ can be checked in $O(n^3)$ time. Moreover, the consistency of $(V', \Gamma', \mathbf{D}')$ can be decided in $O(m^2)$ time, as we only need to check for each pair of variables v_i and v_j in V' whether the unique landmarks specified for them satisfy the constraint between them. By Lemma 1, the consistency of $(V, \Gamma, \mathbf{D})$ can be determined in $O(n^3)$ time. Therefore, $\textsc{cspsat}_s(\mathcal{B}_{\mathcal{M}})$ is in P. By Proposition 1, we know $\textsc{cspsat}_s(\mathcal{M})$ and $\textsc{cspsat}_f(\mathcal{M})$ are all in NP. Meanwhile, the NP-completeness of $\textsc{cspsat}(\mathcal{M})$ implies that $\textsc{cspsat}_s(\mathcal{M})$ and $\textsc{cspsat}_f(\mathcal{M})$ are all NP-complete. □

The following subsections will respectively show that (i) $\textsc{cspsat}_f(\mathcal{B}_{PA})$ is in P but $\textsc{cspsat}_f(PA)$ is NP-complete, and (ii) $\textsc{cspsat}_f(\mathcal{B}_{\mathcal{M}})$ is NP-complete for $\mathcal{M}$ being IA or CRA.

[5] Suppose $\mathcal{M}$ is one of PA, IA, or CRA. Then this result can be generalised to any tractable subclass $\mathcal{S}$ of $\mathcal{M}$ that contains all basic relations.

3.2. Point Algebra

We first propose a polynomial algorithm that solves $\text{CSPSAT}_f(\mathcal{B}_{PA})$ and then provide a polynomial reduction from Graph 3-Colouring to $\text{CSPSAT}_f(PA)$.

Let $(V, \Gamma, \mathbf{D})$ be an instance of $\text{CSPSAT}_f(\mathcal{B}_{PA})$. By Lemma 1 we know that $(V, \Gamma, \mathbf{D})$ is consistent iff Γ is path-consistent and the finitely restricted sub-instance $(V', \Gamma', \mathbf{D}')$ of $(V, \Gamma, \mathbf{D})$ is consistent. Because path-consistency can be determined in cubic time, we only need to devise a polynomial algorithm for checking whether (V', Γ', D') is consistent. To this end, we show that such a consistent instance of $\text{CSPSAT}_f(\mathcal{B}_{PA})$ has a *minimal* solution in a sense.

Proposition 2. *Suppose $(V, \Gamma, \mathbf{D})$ is an instance of $\text{CSPSAT}_f(\mathcal{B}_{PA})$ such that $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ and each D_i is a finite set of real numbers. If $(V, \Gamma, \mathbf{D})$ is consistent, then there is a unique solution $(a_1, \ldots, a_n)$ such that $a_i \leq a'_i$ $(1 \leq i \leq n)$ for any other solution $(a'_1, a'_2, \ldots, a'_n)$. Furthermore, if $\Gamma = \{v_i < v_j\}_{1 \leq i < j \leq n}$, then*

- $a_1 = \min D_1$;
- $a_k = \min\{x \in D_k : x > a_{k-1}\}$ *for* $k = 2, 3, \ldots, n$.

Proof. Assume $\Gamma = \{v_i < v_j\}_{1 \leq i < j \leq n}$. This does not lose generality because we can combine variables related by the '=' constraint and then sort the variables by the '<' and '>' constraints. Every D_i is a finite set, so $(V, \Gamma, \mathbf{D})$ has at most finitely many, say k, solutions. Suppose $(a^i_1, a^i_2, \ldots, a^i_n)$ $(i = 1, 2, \ldots, k)$ enumerate all solutions. Let $a_j = \min\{a^i_j\}_{1 \leq i \leq k}$. We claim that $(a_1, a_2, \ldots, a_n)$ is the minimal solution. We only need to prove that it is a solution of $(V, \Gamma, \mathbf{D})$, i.e. to show (i) each a_j is in D_j; and (ii) $a_1 < a_2 < \ldots < a_n$. Because $a^i_j \in D_j$, we know $a_j = \min\{a^i_j\}_{1 \leq i \leq k}$ is in D_j. We next prove $a_1 < a_2$. Suppose $a_2 = a^j_2$ for some j by definition. Then $a_1 = \min\{a^i_1\}_{1 \leq i \leq k} \leq a^j_1 < a^j_2 = a_2$. By using induction, we can also prove $a_2 < a_3 < \ldots < a_n$. Therefore, $(a_1, a_2, \ldots, a_n)$ is the minimal solution of $(V, \Gamma, \mathbf{D})$. $\square$

We next propose a polynomial algorithm that solves $\text{CSPSAT}_f(\mathcal{B}_{PA})$ based on Proposition 2. For any instance $(V, \Gamma, \mathbf{D})$ of $\text{CSPSAT}_f(\mathcal{B}_{PA})$, we first check whether Γ is consistent. If it is inconsistent, then so is $(V, \Gamma, \mathbf{D})$. Otherwise, we check whether the finitely restricted sub-instance $(V', \Gamma', \mathbf{D}')$ of $(V, \Gamma, \mathbf{D})$ is consistent. To this end, we attempt to compute the minimal solution $(a_1, \ldots, a_n)$ by procedures described in Proposition 2. If in the k-th step $\{x \in D_k : x > a_{k-1}\}$ is empty, then we conclude that the sub-instance, and thus the original instance, is inconsistent. If we succeed in computing $(a_1, a_2, \ldots, a_n)$, then it is a solution of the sub-instance and can be extended to a solution of the original instance. The soundness of the algorithm is clear by the above argument.

Input: $\textsc{cspsat}_f(\mathcal{B}_{PA})$ instance $(V, \Gamma, \mathbf{D})$
Output: The consistency of $(V, \Gamma, \mathbf{D})$

if Γ *is not consistent* **then**
    return 'Inconsistent';
$(V', \Gamma', \mathbf{D}') \leftarrow$ finitely restricted sub-instance of $(V, \Gamma, \mathbf{D})$;
Sort V' to $v'_1 < \ldots < v'_{n'}$ by Γ', modify $\mathbf{D}'$ correspondingly;
$a_1 \leftarrow \min D'_1$;
for $2 \leq k \leq n'$ **do**
    if $a_{k-1} \geq \max D'_k$ **then**
        return 'Inconsistent';
    $a_k \leftarrow \min\{x \in D'_k : x > a_{k-1}\}$;
end
return 'Consistent'.

Algorithm 1: SOLVING $\textsc{cspsat}_f(\mathcal{B}_{PA})$

Theorem 4. *Algorithm 1 solves* $\textsc{cspsat}_f(\mathcal{B}_{PA})$.

We next analyse the computational complexity of the algorithm. Suppose there are n variables in V, and the sum of the cardinalities of all finite D_i is L. Then the input size is $O(n^2 + L)$ (n^2 constraints and L points). The following proposition shows the optimality of the algorithm.

Proposition 3. *The computational complexity of Algorithm 1 is* $O(n^2 + L)$.

Proof. Let $(V, \Gamma, \mathbf{D})$ be an instance of $\textsc{cspsat}_f(\mathcal{B}_{PA})$. The consistency of Γ can be computed in $O(n^2)$ time by Algorithm CSPAN proposed in [43]. Sorting V' takes $O(n \log n)$ time. Let l_i be the cardinality of D'_i. Then step '$a_1 \leftarrow \min D'_1$' takes $O(l_1)$ time, and the i-th loop body takes $O(l_{i+1})$ time ($i = 1, 2, \ldots, n' - 1$). Therefore, the computational complexity of the algorithm is $O(n^2 + n \log n + l_1 + l_2 + \ldots + l_{n'}) = O(n^2 + L)$. □

Despite the fact that both $\textsc{cspsat}(PA)$ and $\textsc{cspsat}_f(\mathcal{B}_{PA})$ are in P, the next theorem shows that $\textsc{cspsat}_f(PA)$ is NP-hard. We prove this by using a polynomial reduction from the Graph 3-Colouring problem to $\textsc{cspsat}_f(PA)$.

Theorem 5. *The consistency problem* $\textsc{cspsat}_f(PA)$ *is NP-complete.*

Proof. Let $G = (V, E)$ be a graph, where $V = \{v_0, \ldots, v_n\}$. Define a $\textsc{cspsat}_f(PA)$ instance $(U_G, \Gamma_G, \mathbf{D}_G)$ as follows:[6]

$$\begin{aligned} U_G &= \{u_0, \ldots, u_n\}, \\ \mathbf{D}_G &= \{D_{u_0}, \ldots, D_{u_n}\}, \quad \text{where } D_{u_i} = \{0, 1, 2\}, \\ \Gamma_G &= \{u_i \neq u_{i'} : (v_i, v_{i'}) \in E\}. \end{aligned}$$

That is, we construct for each vertex $v_i \in V$ a corresponding temporal variable u_i which takes value from $\{0, 1, 2\}$; and we specify for each edge $(v_i, v_{i'}) \in E$ a constraint $(u_i \neq u_{i'})$. It is clear that $G = (V, E)$ can be 3-colourable iff $(U_G, \Gamma_G, \mathbf{D}_G)$ is satisfiable. Therefore the consistency problem $\textsc{cspsat}_f(PA)$ is NP-hard, and hence NP-complete as its NP-membership has been identified in Theorem 3. □

Remark 1. *The NP-hardness of $\textsc{cspsat}_f(PA)$ is due to the uncertainty of the non-equal ($\neq$) constraints and the finiteness of the domains. It can be proven that $\textsc{cspsat}_f(\mathcal{S})$ is in P for $\mathcal{S} = \{<, =, >, \leq, \geq, ?\}$ (i.e., with $\neq$ removed from PA). A polynomial algorithm can be devised based on the observation that the concept of a minimal solution still applies. The algorithm first merges the variables which are required to be equal by the constraints (see [43]). Note the domains of the merged variables should also be revised as the intersection of their original domains. The algorithm then adopts a topological sort, during which each finitely restricted variable is assigned a value in its domain as small as possible.*

3.3. *Cardinal Relation Algebra*

To show that $\textsc{cspsat}_f(\mathcal{B}_{CRA})$ is NP-hard, we design a polynomial reduction from Graph 3-Colouring to $\textsc{cspsat}_f(\mathcal{B}_{CRA})$. Suppose $G = (V, E)$ is a graph with vertex set $V = \{v_0, \ldots, v_n\}$. We construct an instance $(U_G, \Gamma_G, \mathbf{D}_G)$ of $\textsc{cspsat}_f(\mathcal{B}_{CRA})$ such that $(U_G, \Gamma_G, \mathbf{D}_G)$ is satisfiable iff G is 3-colourable.[7]

First, for each vertex $v_i \in V$, we introduce a spatial variable u_i with domain

$$D_{u_i} = \{(3i, 3i), (3i + 1, 3i + 1), (3i + 2, 3i + 2)\}.$$

We say u_i is at position p (where $p \in \{0, 1, 2\}$), if it takes the point $(3i + p, 3i + p)$ in D_{u_i}. Second, for each edge $e_j = (v_i, v_{i'}) \in E$ (assuming $i < i'$), we introduce a spatial variable w_j with domain

$$D_{w_j} = \{(3i + p, 3i' + q) : p, q \in \{0, 1, 2\}, p \neq q\},$$

[6] We assume that the constraint between two variables is the universal constraint if it is not specified in Γ_G.

[7] We here specially thank the reviewer who suggested this elegant reduction to us.

and add two constraints $(w_j\ E\ u_i)$ and $(w_j\ S\ u_{i'})$ to Γ_G. That is to say, w_j should be to the east of u_i and to the south of $u_{i'}$. The domain of w_j is used to rule out the cases when u_i and $u_{i'}$ are at the same position (with respect to their own domains), which correspond to the requirement that vertices v_i and $v_{i'}$ cannot be coloured the same as they are connected by edge e_j.

Note that each $\mathrm{CSPSAT}_f(\mathcal{B}_{CRA})$ instance is a complete network. This means that we should specify for each pair of variables in U_G a basic CRA constraint. In above we have specified such a constraint for two spatial variables u_i and w_j when v_i is a vertex incident to edge e_j in G. There are three other cases unspecified:

- The constraint between u_i and $u_{i'}$;
- The constraint between u_i and w_j, where v_i is not incident to edge e_j in G;
- The constraint between w_j and $w_{j'}$.

In each case it is straightforward to specify a basic constraint between the two spatial variables.

Example 1. *Suppose $G = (V, E)$ is a graph, where $V = \{v_0, v_1, v_2\}$ and $E = \{(v_0, v_1), (v_1, v_2)\}$. Let $(U_G, \Gamma_G, \boldsymbol{D}_G)$ be the $\mathrm{CSPSAT}_f(\mathcal{B}_{CRA})$ instance constructed as above for G. Then $U_G = \{u_0, u_1, u_2, w_0, w_1\}$, with their domains shown in Figure 3. The constraints in Γ_G are given in Table 5, where constraints in black are those corresponding to edges in E.*

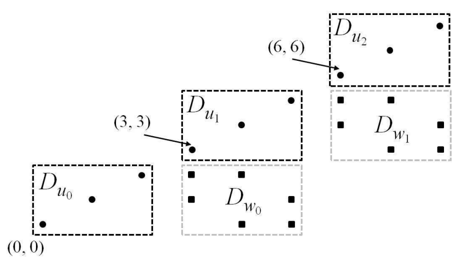


Figure 3: Domains of $(U_G, \Gamma_G, \mathbf{D}_G)$

	u_0	u_1	u_2	w_0	w_1
u_0	EQ	SW	SW	W	SW
u_1		EQ	SW	N	W
u_2			EQ	NE	N
w_0				EQ	SW
w_2					EQ

Table 5: Constraints of $(U_G, \Gamma_G, \mathbf{D}_G)$

Proposition 4. *Graph $G = (V, E)$ is 3-colourable iff $(U_G, \Gamma_G, \boldsymbol{D}_G)$ is satisfiable.*

Proof. Straightforward. $\square$

As a consequence, we have:

α_{ij}	NW	N	NE	W	EQ	E	SW	S	SE
β_{ij}	di	si	oi	fi	eq	f	o	s	d

Table 6: Translation of the constraints

Theorem 6. *The problem* CSPSAT$_f(\mathcal{B}_{CRA})$ *is NP-complete.*

Proof. Since the reduction above is polynomial, we know that CSPSAT$_f(\mathcal{B}_{CRA})$ is NP-hard. Meanwhile, the NP-membership of CSPSAT$_f(\mathcal{B}_{CRA})$ follows from Theorem 3. Therefore, CSPSAT$_f(\mathcal{B}_{CRA})$ is an NP-complete problem. □

3.4. Interval Algebra

To show that CSPSAT$_f(\mathcal{B}_{IA})$ is NP-hard, we design a polynomial reduction from CSPSAT$_f(\mathcal{B}_{CRA})$. Note that an interval $[x, y]$ corresponds to the point (x, y) on the half-plane $\{(x, y) : x < y\}$. Suppose $(V, \Gamma, \mathbf{D})$ is a CSPSAT$_f(\mathcal{B}_{CRA})$ instance, where $V = \{u_1, \ldots, u_n\}$, $\Gamma = \{u_i \alpha_{ij} u_j : 1 \leq i, j \leq n\}$, $\mathbf{D} = (D_1, \ldots, D_n)$. Note that D_i is either the universe of CRA U_{CRA} (viz. the real plane), or a finite subset of U_{CRA}. We now translate $(V, \Gamma, \mathbf{D})$ into a CSPSAT$_f(\mathcal{B}_{IA})$ instance $(V', \Gamma', \mathbf{D}')$, where Γ' is a complete basic IA network. The translation maps

- each variable u_i in V to variable u'_i in V';
- each basic CRA relation α_{ij} to a basic IA relation β_{ij} as specified in Table 6;
- each D_i to D'_i, such that if $D_i = U_{CRA}$ then D'_i is the universe of IA U_{IA}; if D_i is finite, then $D'_i = \{[x, y + \Delta] : (x, y) \in D_i\}$. Here Δ is a fixed large number such that $x < y + \Delta$ for any point (x, y) in any restricted domain D_i.

We show that the translation preserves consistency.

Proposition 5. *An instance* $(V, \Gamma, \boldsymbol{D})$ *in* CSPSAT$_f(\mathcal{B}_{CRA})$ *is consistent iff the corresponding instance* $(V', \Gamma', \boldsymbol{D}')$ *in* CSPSAT$_f(\mathcal{B}_{IA})$ *as constructed above is consistent.*

Proof. Suppose $(a_1, \ldots, a_n)$ is a solution of $(V, \Gamma, \mathbf{D})$, where $a_i = (x_i, y_i) \in D_i$. Define interval $a'_i = [x_i, y_i + \Delta] \in D'_i$. We prove that $(a'_1, \ldots, a'_i)$ is a solution of $(V', \Gamma', \mathbf{D}')$. It is clear that $a'_i \in D'_i$ by the translation from D_i to D'_i. We only need to verify that all the constraints in Γ' are satisfied by $(a'_1, \ldots, a'_i)$. This can be done by discussing each of the nine kinds of basic IA constraints in Γ'.

Suppose $(u'_i \text{ di } u'_j)$ is a constraint in Γ'. We need to prove $[x_i, y_i + \Delta]$ di $[x_j, y_j + \Delta]$, i.e., $x_i < x_j < y_j + \Delta < y_i + \Delta$. By the translation we know that $(u_i \text{ NW } u_j)$ is in Γ. Therefore (x_i, y_i) NW (x_j, y_j), i.e., $x_i < x_j$ and $y_i > y_j$. Meanwhile $x_j < y_j + \Delta$ is guaranteed by the selection of Δ, so the constraint $(u'_i \text{ di } u'_j)$ is satisfied by $(a'_1, \ldots, a'_n)$. The remaining eight cases can be proven analogously. □

Therefore we obtain the following result.

Theorem 7. *The consistency problem* $\textsc{cspsat}_f(\mathcal{B}_{IA})$ *is NP-complete.*

Proof. Since the reduction from $\textsc{cspsat}_f(\mathcal{B}_{CRA})$ to $\textsc{cspsat}_f(\mathcal{B}_{IA})$ is polynomial, we know that $\textsc{cspsat}_f(\mathcal{B}_{IA})$ is NP-hard. Moreover, by Theorem 3, we have $\textsc{cspsat}_f(\mathcal{B}_{IA})$ is in NP. This shows that $\textsc{cspsat}_f(\mathcal{B}_{IA})$ is NP-complete. □

So far, we have completed the discussion for the three point-based qualitative calculi. The next section will address region-based qualitative calculi.

4. Region-based Qualitative Calculi RCC5 and RCC8

This section discusses the consistency problems over RCC5 and RCC8 in the extended qualitative CSP framework. Note that although the universe of RCC5 (or RCC8) is the set of all regions in the plane, it is reasonable to assume that all landmarks are represented as polygons. This is because landmarks, as inputs of instances, are required to be representable in computers. In other words, they should be finitely representable. Meanwhile, general polygons (which may have holes or multiple components) are the most widely used approximations of regions: they are simple, intuitive, and expressive.[8]

Under the assumption that all landmarks are represented by general polygons, we show in this section that all these consistency problems are in NP. In particular, we show that $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is in P, but that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ and $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ are all NP-complete. It is not surprising that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ is NP-complete if we regard the finitely restricted sub-instance of each instance of $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ as a classical CSP, but the NP-hardness of $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ is quite undesirable. One way to circumvent this obstacle is to use a stronger connectedness instead of the one used in Definition 4.

The remainder of this section is organised as follows. We first introduce a simple computational complexity result in Section 4.1 showing that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ is NP-hard. Several of our results are related to computing the intersection of landmarks (represented as polygons), so we analyse its computational complexity in Section 4.2. The tractability of $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is then proven in Section 4.3. Section 4.4 shows that $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ is NP-complete if the RCC8 relations are interpreted as in Definition 4, and proves that the same problem is in P (i.e. tractable) if we adopt another interpretation that uses a stronger connectedness.

[8] Another way to represent regions is to use semi-algebraic sets, which are more expressive than polygons but the set operations are much more complicated.

4.1. *The NP-hardness of* $\text{CSPSAT}_f(\mathcal{B}_{RCC5})$

We prove the NP-hardness of $\text{CSPSAT}_f(\mathcal{B}_{RCC5})$ by designing a polynomial reduction from the Graph 3-Colouring problem.

Proposition 6. *The consistency problem* $\text{CSPSAT}_f(\mathcal{B}_{RCC5})$ *is NP-hard.*

Proof. Let $G = (V, E)$ be a graph, where $V = \{v_0, \ldots, v_n\}$ and $E = \{e_0, \ldots, e_m\}$. For each vertex $v \in V$, we introduce three regions (represented by rectangles) r_v^0, r_v^1 and r_v^2; for each edge $e \in E$, we introduce three regions (represented by rectangles) s_e^0, s_e^1 and s_e^2. These rectangles are required to be pairwise disjoint.

For any $1 \leq i \leq n$ and $0 \leq p \leq 2$, we define a landmark l_i^p as

$$l_i^p = r_{v_i}^p \cup \bigcup \{s_e^p : \text{edge } e \text{ is incident to vertex } v_i\}.$$

Because rectangles r_v^p, s_e^p are pairwise disjoint for $v \in V, e \in E$ and $p \in \{0,1,2\}$, it is clear that $l_i^p \cap l_j^q = \varnothing$ if $p \neq q$. For $i \neq j$ and $p = q$, it is also straightforward to see that $l_i^p \cap l_j^q = s_e^p$ is a rectangle if $e = (v_i, v_j) \in E$ and $l_i^p \cap l_j^q = \varnothing$ otherwise.

The $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$ instance $(V_G, \Gamma_G, \mathbf{D}_G)$ is constructed as follows.

$$\begin{aligned} V_G &= \{u_0, u_1, \ldots, u_n\}, \\ \mathbf{D}_G &= \{D_{u_0}, D_{u_1}, \ldots, D_{u_n}\}, \quad \text{where } D_{u_i} = \{l_i^0, l_i^1, l_i^2\}, \\ \Gamma_G &= \{u_i \mathsf{DR} u_j\}. \end{aligned}$$

Note that spatial variable u_i corresponds to vertex v_i, and "vertex v_i is coloured with colour p" corresponds to that "variable u_i takes value l_i^p." It is routine to show that G is 3-colourable iff $(V_G, \Gamma_G, \mathbf{D}_G)$ is consistent. Because the reduction is polynomial, we know the consistency problem $\text{CSPSAT}_f(\mathcal{B}_{RCC5})$ is NP-hard. □

4.2. *Planar subdivision and overlay computation*

In the following subsections we will see that computing the intersection of landmarks (represented as polygons) is critically important when solving the consistency problems for RCC5 and RCC8 in the extended qualitative CSP framework. To facilitate the discussion, this subsection analyses the computational complexity of computing the intersection of multiple polygons. Our discussion is based on the *doubly-connected edge list* (DCEL) structure for representing planar subdivisions (cf. e.g. [9]).

A *planar subdivision* is an embedding of a planar graph in the plane such that its edges are mapped into straight line segments. It consists of vertices, edges, and faces. *Vertices* are endpoints of line segments, *edges* are interiors of line segments, and *faces* are maximally connected subsets of the plane with all edges and vertices removed. In particular, each face is a connected open set, which may have holes. The outer face is unbounded, but every other face is bounded and its boundary consists of vertices and edges. The *complexity* of a planar subdivision is defined as the sum of the number of its vertices, the number of its edges, and the number of its faces. For example, the planar subdivision of the rectangle in Figure 4 (a) has two faces (Figure 4 (a)), four vertices (Figure 4 (b)) and four edges (Figure 4 (c)), and thus has a complexity of ten.[9]

In what follows, we write $\textsc{Face}$, $\textsc{Edge}$, and $\textsc{Vtx}$ respectively for the set of faces, the set of edges, and the set of vertices in a planar subdivision, and use lower Fraktur symbols $\mathfrak{f}, \mathfrak{e}, \mathfrak{v}$ (possibly with indices) to denote, respectively, faces, edges, and vertices in the subdivision.

The following lemma shows that the complexity of a planar subdivision is of the same order as the number of its vertices.

Lemma 2. *Let S be a planar subdivision with k vertices. Then the complexity of S is $O(k)$.*

Proof. Recall that each planar subdivision is an embedding of a planar graph in the plane. By Euler's formula (cf.[10]), if S has C connected components then

$$|\textsc{Vtx}| - |\textsc{Edge}| + |\textsc{Face}| = C + 1.$$

Furthermore, since each face is bounded by at least three edges, and each edge touches at most two faces, it is straightforward to prove that

$$|\textsc{Edge}| < 3|\textsc{Vtx}| \quad \text{and} \quad |\textsc{Face}| < 2|\textsc{Vtx}|.$$

Therefore the complexity of S is $O(k)$. □

In Computational Geometry, a planar subdivision is usually represented by the doubly-connected edge list (DCEL), where each edge is considered as two directed half-edges with opposite directions. The DCEL of a subdivision maintains a table for each vertex, each half-edge, and each face. The table allows the retrieve from an object (viz. vertex, half-edge, or faced) to its incident (or adjacent) objects efficiently. For a planar subdivision $\mathcal{S}$ with complexity k, the DCEL of $\mathcal{S}$ takes $O(k)$ space.

[9] To avoid potential confusion, when discussing the time resource it takes for computing an overlay, we always explicitly use the term *computational complexity*.

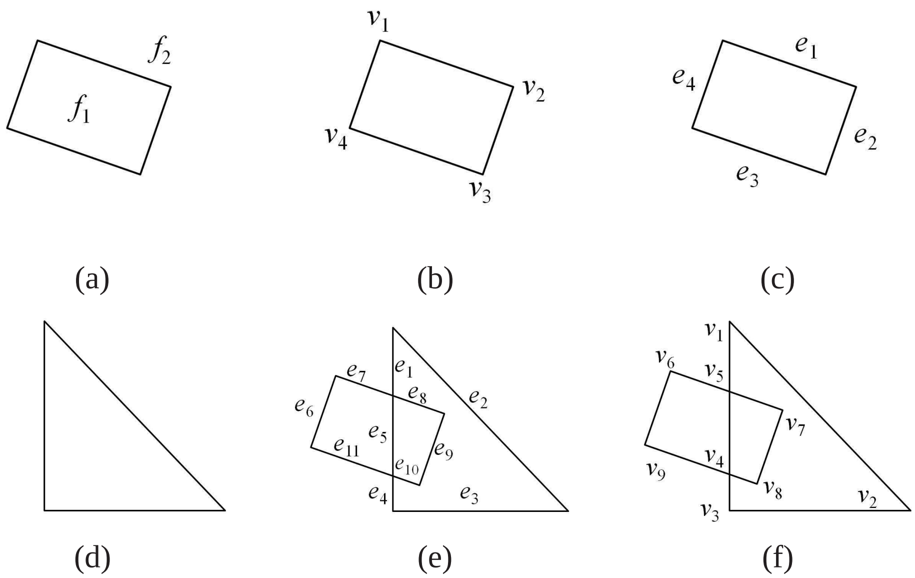


Figure 4: An example of subdivision

The *overlay* of two planar subdivisions S_1 and S_2 is the planar subdivision S induced by all edges from S_1 and S_2. Each vertex of S is either a vertex of S_1 or S_2, or the intersection point of two edges from S_1 and S_2. Each edge is either an edge of S_1 or S_2, or a part of an edge of S_1 cut by an edge of S_2, or vice versa. Similarly, each face of S is either a face of S_1 or S_2, or the intersection of two faces from S_1 and S_2. Figures 4 (e) and (f) illustrate the overlay of the rectangle in Figure 4 (a) and the triangle in Figure 4 (d), which has four faces, eleven edges and nine vertices, and hence has a complexity of 24.

We have the following result about the complexity of the overlay.

Lemma 3. *Let S_1 and S_2 be two planar subdivisions of complexity k_1 and k_2 respectively. Then the overlay of S_1 and S_2 has complexity $O(k_1k_2)$.*

Proof. Note that each vertex in the overlay is either a vertex of S_1, or a vertex of S_2, or the intersection point of two edges from different subdivisions. As the numbers of vertices and edges of S_i are less than k_i, the overlay has $O(k_1k_2)$ vertices. The complexity of the overlay then follows from Lemma 2. $\square$

The computational complexity of the overlay computation is as follows.

Proposition 7. *[9, Theorem 2.6] Let S_1 and S_2 be two planar subdivisions of complexity k_1 and k_2 respectively. Then the overlay of S_1 and S_2 can be constructed in $O((k_1 + k_2 + k)\log(k_1 + k_2))$ time, where k is the complexity of the overlay.*

Proposition 7 only considers the overlay of two subdivisions. For the consistency problems $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$ and $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$, we need to compute the overlay $\mathcal{O}$ of the subdivisions induced by landmarks $l_1, \ldots, l_m$ ($m \geq 3$). At first glance, the computational complexity seems to be very high. Suppose each landmark is represented by a polygon with k vertices. If we use Lemma 2 successively then the overlay will have complexity $O(k^m)$. As a consequence, the computational complexity of computing the overlay will be exponential if we use Proposition 7 successively.The following result shows that, however, $\mathcal{O}$ can be computed in polynomial time. The key idea is that the complexity of the overlay of the m subdivisions is, instead of $O(k^m)$, polynomial in m and k (if we assume each landmark has k vertices).

Lemma 4. *Suppose l_i is a polygon with k_i vertices for each $1 \leq i \leq m$. Let $K = \sum_{i=1}^m k_i$ and $\mathcal{O}$ be the overlay of the subdivisions induced by these polygons. Then $\mathcal{O}$ has complexity $O(K^2)$ and can be computed in $O(mK^2 \log K)$ time.*

Proof. It is clear that there are in total $O(K)$ vertices and, by Lemma 2, $O(K)$ edges in the subdivisions induced by these polygons. As each vertex in the overlay $\mathcal{O}$ is either a vertex of a subdivision, or the intersection point of two edges from different subdivisions, we know that $\mathcal{O}$ has $O(K^2)$ vertices. By Lemma 2, the complexity of $\mathcal{O}$ is also $O(K^2)$.

Write $\mathcal{O}_i$ for the overlay of the subdivisions induced by the first i polygons $l_1, \ldots, l_i$. The complexity of each $\mathcal{O}_i$ is no more than that of $\mathcal{O} = \mathcal{O}_m$. By Proposition 7 we know $\mathcal{O}_{i+1}$ can be computed in $O((K^2 + K + K^2)\log(K^2 + K)) = O(K^2 \log K)$ time from $\mathcal{O}_{i+1}$ and l_{i+1}. Therefore, the overlay $\mathcal{O}$ can be computed in $O(mK^2 \log K)$ time from $l_1, ..., l_m$. □

We note that the DCEL of $\mathcal{O}$ contains incidence and adjacency information between two elements in FACE, EDGE, and VTX. The relationship between such an element and a polygon in L, however, is not provided. For example, the DCEL does not tell us whether an edge lies inside, outside, or on the boundary of a polygon l_i. To represent the complete topological information of the polygon system L, we introduce the following functions, which can be computed by supplying a number of attributes to each object in the DCEL of the overlay.

For each polygon $l_i \in L$, we write $\mathcal{I}_{\text{FACE}}(l_i)$ ($\mathcal{E}_{\text{FACE}}(l_i)$, resp.) for the set of faces in $\mathcal{O}$ that lie in the interior (exterior, resp.) of l_i:

$$\mathcal{I}_{\text{FACE}}(l_i) = \{\mathfrak{f} \in \text{FACE} : \mathfrak{f} \subseteq l_i^\circ\}, \tag{1}$$
$$\mathcal{E}_{\text{FACE}}(l_i) = \{\mathfrak{f} \in \text{FACE} : \mathfrak{f} \cap l_i = \varnothing\}. \tag{2}$$

It is clear that $\mathcal{I}_{\text{FACE}}(l_i) \cup \mathcal{E}_{\text{FACE}}(l_i) = \text{FACE}$ and $\mathcal{I}_{\text{FACE}}(l_i) \cap \mathcal{E}_{\text{FACE}}(l_i) = \varnothing$.

For each polygon l_i, we define

$$\mathcal{I}_{\text{EDGE}}(l_i) = \{\mathfrak{e} \in \text{EDGE} : \mathfrak{e} \subseteq l_i^\circ\}, \tag{3}$$
$$\mathcal{E}_{\text{EDGE}}(l_i) = \{\mathfrak{e} \in \text{EDGE} : \mathfrak{e} \cap l_i = \varnothing\}, \tag{4}$$
$$\mathcal{B}_{\text{EDGE}}(l_i) = \{\mathfrak{e} \in \text{EDGE} : \mathfrak{e} \subseteq \partial l_i\}, \tag{5}$$

and similarly,

$$\mathcal{I}_{\text{VTX}}(l_i) = \{\mathfrak{v} \in \text{VTX} : \mathfrak{v} \in l_i^\circ\}, \tag{6}$$
$$\mathcal{E}_{\text{VTX}}(l_i) = \{\mathfrak{v} \in \text{VTX} : \mathfrak{v} \notin l_i\}, \tag{7}$$
$$\mathcal{B}_{\text{VTX}}(l_i) = \{\mathfrak{v} \in \text{VTX} : \mathfrak{v} \in \partial l_i\}. \tag{8}$$

Because each edge and each vertex is either in the interior of l_i, or in the exterior of l_i, or on the boundary of l_i, we know that $\{\mathcal{I}_{\text{EDGE}}(l_i), \mathcal{E}_{\text{EDGE}}(l_i), \mathcal{B}_{\text{EDGE}}(l_i)\}$ is a partition of EDGE, and $\{\mathcal{I}_{\text{VTX}}(l_i), \mathcal{E}_{\text{VTX}}(l_i), \mathcal{B}_{\text{VTX}}(l_i)\}$ is a partition of VTX.

We provide an example to illustrate these functions.

Example 2. *Suppose $L = \{l_1, l_2, l_3\}$ consists of the three polygons illustrated in Figure 5(a). Then we have* $\text{FACE} = \{\mathfrak{f}_0, \ldots, \mathfrak{f}_4\}$, $\text{VTX} = \{\mathfrak{v}_1, \ldots, \mathfrak{v}_{11}\}$ *and* $\text{EDGE} = \{\mathfrak{e}_1, \ldots, \mathfrak{e}_{14}\}$*, as shown in Figure 5(b-d). In particular, for landmark l_1, we have*

$\mathcal{I}_{\text{FACE}}(l_1) = \{\mathfrak{f}_1, \mathfrak{f}_2\}$, $\mathcal{I}_{\text{VTX}}(l_1) = \{\mathfrak{v}_6, \mathfrak{v}_{11}\}$, $\mathcal{I}_{\text{EDGE}}(l_1) = \{\mathfrak{e}_6, \mathfrak{e}_{10}, \mathfrak{e}_{11}\}$,
$\mathcal{E}_{\text{FACE}}(l_1) = \{\mathfrak{f}_0, \mathfrak{f}_3, \mathfrak{f}_4\}$, $\mathcal{E}_{\text{VTX}}(l_1) = \{\mathfrak{v}_3, \mathfrak{v}_8, \mathfrak{v}_9\}$, $\mathcal{E}_{\text{EDGE}}(l_1) = \{\mathfrak{e}_2, \mathfrak{e}_3, \mathfrak{e}_7, \mathfrak{e}_8, \mathfrak{e}_9\}$,
$\mathcal{B}_{\text{VTX}}(l_1) = \{\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_7, \mathfrak{v}_{10}, \mathfrak{v}_4, \mathfrak{v}_5\}$, $\mathcal{B}_{\text{EDGE}}(l_1) = \{\mathfrak{e}_1, \mathfrak{e}_{12}, \mathfrak{e}_{13}, \mathfrak{e}_{14}, \mathfrak{e}_4, \mathfrak{e}_5\}$.

Together with the functions defined in (1)-(8), the DCEL of the overlay of polygons in L completely describes the topological information of polygons in L. The following lemma shows that these functions can also be computed in polynomial time.

Lemma 5. *Suppose l_i is a polygon with $k_i > 2$ vertices for each $1 \le i \le m$. Let $\mathcal{O}$ be the overlay of all these polygons, and K be the sum of all k_i. Then the functions defined in (1)-(8) for all $1 \le i \le m$ can be computed in $O(m^2K^2)$ time in total.*

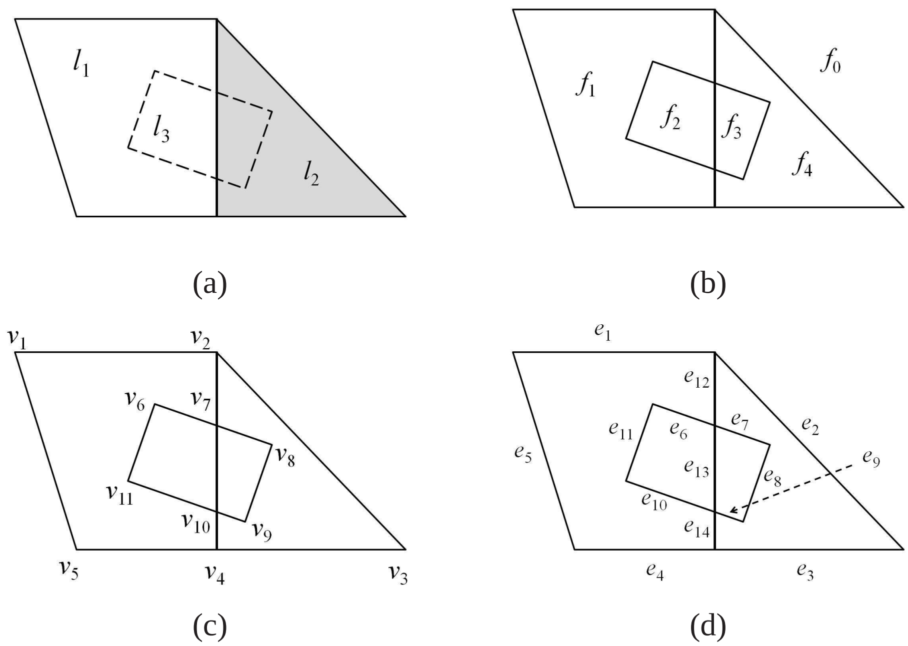


Figure 5: Example of the overlay of $L = \{l_1, l_2, l_3\}$.

Proof. As in the proof of Lemma 4, suppose $\mathcal{O}_k$ is the overlay of the first k polygons $l_1, \ldots, l_k$ and $\mathcal{O} = \mathcal{O}_m$. For each element (i.e., a face, edge or vertex) c in overlay $\mathcal{O}_i$, we introduce an additional vector to represent the relation between c and polygons $l_1, l_2, \ldots, l_i$. When updating the overlay $\mathcal{O}_i$ to $\mathcal{O}_{i+1}$, we need to update these vectors correspondingly. Note that each $\mathcal{O}_i$ has $O(K^2)$ elements. There are $O(K^2)$ vectors, each of which has $i \leq m$ indices. Therefore we need $O(mK^2)$ time to update all vectors for each overlay $\mathcal{O}_i$, and thus $O(m^2K^2)$ time in total for $\mathcal{O}$. The functions in (1)-(8) can be computed from the vectors for $\mathcal{O}$ directly in $O(mK^2)$ time. In summary, it takes an additional $O(m^2K^2)$ time to compute all the functions. □

Combined with Lemma 4, this shows that the overlay and the functions can be computed in $O(m^2K^2 \log K)$ time.

4.3. Solving basic RCC5 constraints involving polygonal landmarks

This subsection shows that $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$ is in P, provided that all landmarks are represented as polygons. We obtain this by giving a necessary and sufficient condition for deciding the consistency of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$ instances, which can be checked in polynomial time.

In what follows, we write an instance of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$ or $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ explicitly as $(V \uplus L, \Gamma)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of unrestricted variables, $L = \{l_1, l_2, \ldots, l_m\}$ is the set of uniquely restricted variables. We write, for simplicity, l_i for the only value (viz. a polygonal landmark) it takes and assume that the constraint between two landmarks is the actual relation between them.

4.3.1. A necessary and sufficient condition

Suppose $(V \uplus L, \Gamma)$ is an instance of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $L = \{l_1, l_2, \ldots, l_m\}$. Let $\mathcal{O}$ be the overlay of polygons in L. Recall that for each l_j and each face $\mathfrak{f}$ in $\mathcal{O}$, $\mathfrak{f}$ is either in $\mathcal{I}_{\text{FACE}}(l_j)$ (the set of faces contained in l_j) or in $\mathcal{E}_{\text{FACE}}(l_j)$ (the set of faces that lie outside l_j). Constraints in Γ may impose similar relationships between $\mathfrak{f}$ and the variables in V. For a variable v_i, the constraints about v_i may force $\mathfrak{f}$ to be part of v_i, or outside v_i. Precisely, $\mathfrak{f}$ is required to be part of v_i if there is a landmark l_j such that $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_j)$ and $l_j \mathsf{PP} v_i$, and $\mathfrak{f}$ is required to lie outside v_i if either $v_i \mathsf{DR} l_j$ and $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_j)$, or $v_i \mathsf{PP} l_j$ and $\mathfrak{f} \in \mathcal{E}_{\text{FACE}}(l_j)$. For each variable $v_i \in V$, we thus define $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ as follows:

$$\mathcal{I}_{\text{FACE}}(v_i) = \bigcup \{\mathcal{I}_{\text{FACE}}(l_j) : l_j \mathsf{PP} v_i\}, \tag{9}$$

$$\mathcal{E}_{\text{FACE}}(v_i) = \bigcup \{\mathcal{I}_{\text{FACE}}(l_j) : v_i \mathsf{DR} l_j\} \cup \bigcup \{\mathcal{E}_{\text{FACE}}(l_j) : v_i \mathsf{PP} l_j\}. \tag{10}$$

Example 3. *Suppose $(V \uplus L, \Gamma)$ is an instance of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$, where $V = \{v_1\}$ and $L = \{l_1, l_2, l_3\}$. Landmarks l_1, l_2, l_3 are shown in Figure 5(a). The constraints related to v_1 are specified as $l_1 \mathsf{PP} v_1, l_2 \mathsf{PP} v_1, v_1 \mathsf{PO} l_3$. Then we have*

$$\mathcal{I}_{\text{FACE}}(v_1) = \mathcal{I}_{\text{FACE}}(l_1) \cup \mathcal{I}_{\text{FACE}}(l_2) = \{\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_3, \mathfrak{f}_4\}, \quad \mathcal{E}_{\text{FACE}}(v_1) = \varnothing.$$

The following proposition asserts that no face belongs to both $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$, given that the constraint network is path-consistent.

Proposition 8. *Suppose $(V \uplus L, \Gamma)$ is an instance of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each l_i is a polygon. If Γ is path-consistent, then $\mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{E}_{\text{FACE}}(v_i) = \varnothing$.*

Proof. Assume $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{E}_{\text{FACE}}(v_i)$. By definition there exist l_j and l_k such that $l_j \mathsf{PP} v_i$ and $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_j)$, and either (i) $v_i \mathsf{DR} l_k$ and $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_k)$ or (ii) $v_i \mathsf{PP} l_k$ and $\mathfrak{f} \in \mathcal{E}_{\text{FACE}}(l_k)$. We show that both cases lead to a contradiction. For the first

Constraint	Conditions
$v_i\mathsf{PO}l_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$
$v_i\mathsf{PP}l_j$	$\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$
$l_j\mathsf{PP}v_i$	$\mathcal{E}_{\textsc{Face}}(v_i) \neq \mathcal{E}_{\textsc{Face}}(l_j)$
$v_i\mathsf{PO}v_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$
$v_i\mathsf{PP}v_j$	$\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$

Table 7: Conditions for extended RCC5 constraint network

case, we know $\mathfrak{f} \subseteq l_j \cap l_k$, while the path-consistency of Γ implies that $l_j\mathsf{DR}l_k$ since $l_j\mathsf{PP}v_i$ and $v_i\mathsf{DR}l_k$. For the second case, we have $\mathfrak{f} \subseteq l_j$ and $\mathfrak{f} \cap l_k = \varnothing$, but the path-consistency of Γ implies $l_j\mathsf{PP}l_k$ since $l_j\mathsf{PP}v_i$ and $v_i\mathsf{PP}l_k$. □

The following theorem provides a necessary and sufficient condition that decides $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$. Note that the condition only involves

$$\textsc{Face}, \mathcal{I}_{\textsc{Face}}(l_j), \mathcal{E}_{\textsc{Face}}(l_j), \mathcal{I}_{\textsc{Face}}(v_i), \mathcal{E}_{\textsc{Face}}(v_i),$$

and constraints in the network, hence it can be checked after constructing the overlay of all landmarks and computing $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$ for each v_i.

Theorem 8. *Suppose $(V \uplus L, \Gamma)$ is an instance of $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each l_i is a polygon. Then $(V \uplus L, \Gamma)$ is consistent, if and only if*

- Γ *is path-consistent.*
- *For any $v_i \in V$, $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$.*
- *All the conditions in Table 7 hold.*

Conditions in Table 7 are very natural. For instance, the three conditions for constraint $(v_i\mathsf{PO}l_j)$ guarantee, respectively, that (i) v_i is not a proper subset of l_j, (ii) v_i is not a proper superset of l_j, and (iii) v_i may overlap with l_j, i.e., not every face in $\mathcal{I}_{\textsc{Face}}(l_j)$ is excluded from v_i. Consider Example 3 again.

Example 3 (continued)
In this example, we have $\mathcal{I}_{\textsc{Face}}(v_1) = \{\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_3, \mathfrak{f}_4\}$ and $\mathcal{E}_{\textsc{Face}}(v_1) = \varnothing$. Since $\mathcal{E}_{\textsc{Face}}(l_3) = \{\mathfrak{f}_0, \mathfrak{f}_1, \mathfrak{f}_4\}$, we know $\mathcal{I}_{\textsc{Face}}(v_1) \cup \mathcal{E}_{\textsc{Face}}(l_3) = \{\mathfrak{f}_0, \mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_3, \mathfrak{f}_4\} = \textsc{Face}$. Because $(v_1 \mathsf{PO} l_3) \in \Gamma$, Row 3 of Table 7 is violated. By Theorem 8 we know this instance is inconsistent.

We prove the necessity part here and leave the sufficiency part to Appendix A.

Proof of Theorem 8 (Necessity). Suppose $(a_1, \ldots, a_n)$ is a solution of Γ, where a_i is assigned to v_i. Because each a_i has a nonempty interior, there exists at least one face $\mathfrak{f}$ such that $\mathfrak{f} \cap a_i$ is nonempty. Clearly, $\mathfrak{f} \notin \mathcal{E}_{\textsc{Face}}(v_i)$ since faces in $\mathcal{E}_{\textsc{Face}}(v_i)$ are all disjoint from a_i (otherwise a DR or PP constraint is violated). Therefore, $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$.

If $(v_i \mathsf{PO} l_j) \in \Gamma$, then by assumption we have $a_i \mathsf{PO} l_j$. By definition of PO (see Table 3), we know that a_i and l_j have a common interior point. This implies that there exists a face $\mathfrak{f}$ that contains an interior point of $a_i \cap l_j$. Face $\mathfrak{f}$ is neither in $\mathcal{E}_{\textsc{Face}}(v_i)$ nor in $\mathcal{E}_{\textsc{Face}}(l_j)$. That is, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$. Similarly, we know that neither $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j) = \textsc{Face}$ nor $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) = \textsc{Face}$.

If $(v_i \mathsf{PP} l_j) \in \Gamma$, then $a_i \mathsf{PP} l_j$. Because l_j is the regularised union of all faces it contains, i.e. $l_j = \overline{\bigcup\{\mathfrak{f} : \mathfrak{f} \in \mathcal{I}_{\textsc{Face}}(l_j)\}}$, we know there exists at least one face in $\mathcal{I}_{\textsc{Face}}(l_j)$ that is not in $\mathcal{I}_{\textsc{Face}}(v_i)$. This shows $\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$.

The remaining cases are either straightforward or similar to the above two cases. $\square$

Using Theorem 8, we are able to determine the consistency of any instance of $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ in the following procedure:

- Compute $\mathcal{I}_{\textsc{Face}}(l_j)$ and $\mathcal{E}_{\textsc{Face}}(l_j)$ for each landmark l_j (this relies on the computation of the overlay planar subdivision $\mathcal{O}$).

- Compute $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$ for each variable v_i.

- Check the conditions in Theorem 8.

Therefore the computational complexity of solving $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ consists of three parts, corresponding to (i) computing $\mathcal{I}_{\textsc{Face}}(l_j)$ and $\mathcal{E}_{\textsc{Face}}(l_j)$, (ii) computing $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$, and (iii) checking the conditions in Theorem 8. Putting them together, we come to the following theorem.

Theorem 9. *Suppose* $(V \uplus L, \Gamma)$ *is an instance of* $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$, *where* $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, *and each* l_i *is a polygon. Let* k_i *be the complexity of the planar subdivision induced by* l_i, *and let* $K = \Sigma_{i=1}^m k_i$. *Then the consistency of* $(V \uplus L, \Gamma)$ *can be decided in* $O(n^3 + n^2K^2 + m^2K^2 \log K)$ *time.*

Proof. Lemmas 4 and 5 show that $\mathcal{O}$, the overlay of all landmarks in L, together with $\mathcal{I}_{\text{FACE}}(l_j)$ and $\mathcal{E}_{\text{FACE}}(l_j)$, can be computed in $O(m^2K^2 \log K)$ time. Moreover, all $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ can be computed in $O(nmK^2)$ time by definition. For the conditions in Theorem 8, it takes $O((n+m)^3)$ time to check the path-consistency of Γ, and $O(K^2)$ time to check each of the remaining $O(n(n+m))$ conditions. Therefore, it takes $O((n+m)^3 + n(n+m)K^2)$ time to check all the conditions in Theorem 8. Summing these up, the consistency of $(V \uplus L, \Gamma)$ can be determined in $O((n+m)^3 + n(n+m)K^2 + m^2K^2 \log K)$ time. Note that $m \leq \Sigma_{i=1}^m k_i = K$. If $m \leq n$, then $O((m+n)^3) = O(n^3)$; if $n \leq m$, then $O((m+n)^3) = O(m^3)$. In both cases we have $O((m+n)^3) = O(m^3 + n^3)$. Similarly we have $O(mnK^2) = O(m^2K^2 + n^2K^2)$. Therefore,

$$\begin{aligned} & O((n+m)^3 + n(n+m)K^2 + m^2K^2 \log K) \\ = & O((m^3 + n^3) + (n^2K^2 + m^2K^2) + m^2K^2 \log K) \\ = & O(n^3 + n^2K^2 + m^2K^2 \log K) \end{aligned}$$

and the consistency of $(V \uplus L, \Gamma)$ can be decided in $O(n^3 + n^2K^2 + m^2K^2 \log K)$ time. □

As a direct consequence, we have

Theorem 10. *Assuming that all landmarks are represented by polygons, then the consistency problem* $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ *is in P, and the consistency problems* $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$, $\textsc{cspsat}_s(RCC5)$, *and* $\textsc{cspsat}_f(RCC5)$ *are all NP-complete.*

Proof. It follows directly from Theorem 9 that $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is in P. Moreover, by Proposition 1 we know that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$, $\textsc{cspsat}_s(RCC5)$, and $\textsc{cspsat}_f(RCC5)$ are all in NP. The NP-hardness of $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ is proven in Proposition 6, and the NP-hardness of $\textsc{cspsat}_s(RCC5)$ and $\textsc{cspsat}_f(RCC5)$ follows from the NP-hardness of $\textsc{cspsat}(RCC5)$. □

Although $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is in P, we show in the next subsection that the consistency problem $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ is NP-hard.

4.4. Solving basic RCC8 constraints involving landmarks

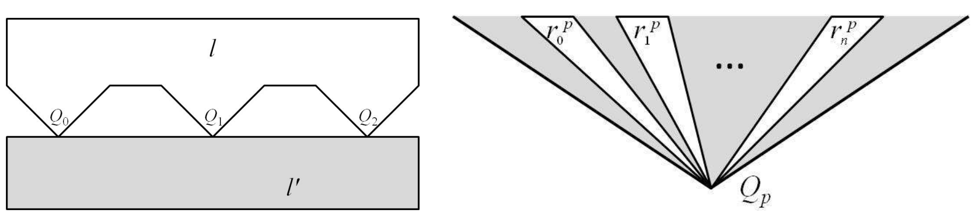
This subsection investigates the consistency problem $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$. First, we show that the problem is NP-hard by exploiting the fact that two polygons may have multiple 'meeting' points. Second, we show that the problem is still in NP by providing a polynomial nondeterministic algorithm. We then consider another interpretation of the RCC8 model by using a stronger connectedness. Under this interpretation, we show that $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ is still tractable.

4.4.1. The NP-hardness of $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$

We reduce the Graph 3-Colouring problem to the $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ problem.

Proposition 9. *Assuming that all landmarks are represented by polygons, the consistency problem $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ is NP-hard.*

Proof. Suppose $G = (V, E)$ is a graph and $V = \{v_0, \ldots, v_n\}$. We construct a $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ instance $(V_G \uplus L, \Gamma_G)$ as follows. The landmark set L is independent of the choice of G and contains the two polygons l and l' in Figure 6 (a). Note that l and l' are externally connected and have exactly three meeting points Q_0, Q_1 and Q_2, which are used to mimic the three colours in the Graph 3-Colouring problem.



(a) landmarks l and l' (b) candidate regions r_i^p $(p = 0, 1, ..., n)$

Figure 6: Illustration for the reduction for $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$

The spatial variable set V_G is defined as $\{u_0, u_1, ..., u_n\}$, where spatial variable u_i corresponds to vertex v_i in V. The constraint network Γ_G is defined as follows.

$$\Gamma_G = \{u_i \mathsf{TPP} l\} \cup \{u_i \mathsf{EC} l'\} \cup \{u_i \mathsf{DC} u_j : (v_i, v_j) \in E\} \cup \{u_i \mathsf{EC} u_j : (v_i, v_j) \notin E\}.$$

We have finished the construction of the instance. The idea behind this reduction is as follows. Because l and l' have only three meeting points (viz. Q_0, Q_1 and Q_2), each u_i can be connected to l' only via (one or more of) the three points Q_0, Q_1, Q_2. Determining which point u_i should occupy is essentially equivalent to choosing a colour for vertex v_i. For v_i and v_j, if (v_i, v_j) is an edge in E, then they cannot be coloured the same. Correspondingly, in such a case there is a constraint $u_i\mathsf{DC}u_j$, which forbids that u_i and u_j occupy the same point in $\{Q_0, Q_1, Q_2\}$.

We now prove that G is 3-colourable iff $(V_G \uplus L, \Gamma_G)$ is consistent. Suppose $\pi : V \to \{0,1,2\}$ is a valid 3-colouring of G. We choose three candidate regions r_i^0, r_i^1 and r_i^2 for each variable u_i, where r_i^p is a triangle contained in l with a vertex being Q_p. The candidate regions $r_0^p, r_1^p, \ldots, r_n^p$ are externally connected at Q_p, as illustrated in Figure 6 (b). If we assign $r_i^{\pi(v_i)}$ to u_i, then all the DC constraints are satisfied. This is because, $r_i^{\pi(v_i)}$ and $r_j^{\pi(v_j)}$ are connected iff $\pi(v_i) = \pi(v_j)$. This assignment, however, cannot fulfil all the EC constraints. For each unsatisfied EC constraint $(u_i\mathsf{EC}u_j)$, we introduce a pair of rectangles r_{ij} and r'_{ij}, which are external connected and contained in l. We require that these rectangles are small enough and disjoint from any other rectangles $r_{i'j'}$, $r'_{i'j'}$ and any triangle r_k^p. We then add r_{ij} and r'_{ij} into, respectively, the candidate regions we have selected for u_i and u_j. It is routine to verify that the modified assignment satisfies all constraints in Γ_G and hence is a solution of $(V_G \uplus L, \Gamma_G)$.

For the other direction, suppose $(a_0, \ldots, a_n)$ is a solution of $(V_G \uplus L, \Gamma_G)$. Note that each a_i occupies at least one point in $\{Q_0, Q_1, Q_2\}$. Define $\pi : V \to \{0,1,2\}$ by assigning v_i the smallest index q such that a_i occupies Q_q. The assignment π is a valid 3-colouring for graph G. In fact, suppose $\pi(v_i) = \pi(v_j) = p$. Then by definition both a_i and a_j occupies Q_p. Hence $(u_i\mathsf{DC}u_j)$ is not a constraint in Γ_G, which happens only when $(v_i, v_j) \notin E$.

The reduction given above is polynomial because there are only two landmarks and $|V|$ spatial variables in $(V_G \uplus L, \Gamma_G)$. Therefore, the consistency problem $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ is NP-hard. □

In the next subsection we show that $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$ is still in NP by designing a nondeterministic algorithm.

4.4.2. *A nondeterministic algorithm for* $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$

Suppose $(V \uplus L, \Gamma)$ is an instance of $\text{CSPSAT}_s(\mathcal{B}_{RCC8})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each l_i is a polygon. We write $\mathcal{O}$ for the overlay of all landmarks in L, and define

$$\mathcal{I}_{\text{FACE}}(l_i), \mathcal{E}_{\text{FACE}}(l_i), \mathcal{I}_{\text{EDGE}}(l_i), \mathcal{E}_{\text{EDGE}}(l_i), \mathcal{B}_{\text{EDGE}}(l_i), \mathcal{I}_{\text{VTX}}(l_i), \mathcal{E}_{\text{VTX}}(l_i), \mathcal{B}_{\text{VTX}}(l_i)$$

as in (1)-(8) for representing the topological relations between faces, edges, vertices in $\mathcal{O}$ and landmarks in L. As in the case of $\text{CSPSAT}_s(\mathcal{B}_{RCC5})$, we extend these definitions from landmarks to variables. In the following, we say an edge $\mathfrak{e}$ or a vertex $\mathfrak{v}$ in $\mathcal{O}$ is *incident* to a face $\mathfrak{f}$ in $\mathcal{O}$ if $\mathfrak{e}$ or $\mathfrak{v}$ is contained in the boundary of $\mathfrak{f}$, and write

$$S_{\text{FACE}}(\mathfrak{v}) = \{\mathfrak{f} \in \text{FACE} : \mathfrak{v} \text{ is incident to } \mathfrak{f}\}, \tag{11}$$
$$S_{\text{FACE}}(\mathfrak{e}) = \{\mathfrak{f} \in \text{FACE} : \mathfrak{e} \text{ is incident to } \mathfrak{f}\}. \tag{12}$$

Note that $S_{\text{FACE}}(\mathfrak{e})$ has exactly two faces and $S_{\text{FACE}}(\mathfrak{v})$ may have more than two faces. These two functions can be directly obtained from the DCEL of the overlay.

Similarly as in the RCC5 case, we define $\mathcal{I}_{\text{FACE}}(v_i)$ as the set of faces that should be part of v_i and define $\mathcal{E}_{\text{FACE}}(v_i)$ as the set of faces that should be excluded from v_i.

$$\mathcal{I}_{\text{FACE}}(v_i) = \bigcup\{\mathcal{I}_{\text{FACE}}(l_j) : l_j \mathsf{TPP} v_i \text{ or } l_j \mathsf{NTPP} v_i\}, \tag{13}$$
$$\begin{aligned}\mathcal{E}_{\text{FACE}}(v_i) = &\bigcup\{\mathcal{I}_{\text{FACE}}(l_j) : v_i \mathsf{DC} l_j \text{ or } v_i \mathsf{EC} l_j\} \cup \\ &\bigcup\{\mathcal{E}_{\text{FACE}}(l_j) : v_i \mathsf{TPP} l_j \text{ or } v_i \mathsf{NTPP} l_j\}.\end{aligned} \tag{14}$$

Moreover, we define $\mathcal{I}_{\text{EDGE}}(v_i)$ as the set of edges that should lie in the interior of v_i, $\mathcal{E}_{\text{EDGE}}(v_i)$ as the set of edges that should lie in the exterior of v_i, and $\mathcal{B}_{\text{EDGE}}(v_i)$ as the set of edges that are required to be parts of the boundary of v_i.

$$\mathcal{I}_{\text{EDGE}}(v_i) = \{\mathfrak{e} \in \text{EDGE} : S_{\text{FACE}}(\mathfrak{e}) \subseteq \mathcal{I}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{EDGE}}(l_j) : l_j \mathsf{NTPP} v_i\}, \tag{15}$$
$$\mathcal{E}_{\text{EDGE}}(v_i) = \{\mathfrak{e} \in \text{EDGE} : S_{\text{FACE}}(\mathfrak{e}) \subseteq \mathcal{E}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{EDGE}}(l_j) : v_i \mathsf{DC} l_j \text{ or } v_i \mathsf{NTPP} l_j\}, \tag{16}$$
$$\mathcal{B}_{\text{EDGE}}(v_i) = \{\mathfrak{e} \in \text{EDGE} : S_{\text{FACE}}(\mathfrak{e}) \cap \mathcal{I}_{\text{FACE}}(v_i) \neq \varnothing, S_{\text{FACE}}(\mathfrak{e}) \cap \mathcal{E}_{\text{FACE}}(v_i) \neq \varnothing\}. \tag{17}$$

A brief explanation for the above notions follows. For an edge $\mathfrak{e}$, if its two incident faces (i.e., faces in $S_{\text{FACE}}(\mathfrak{e})$) are both in $\mathcal{I}_{\text{FACE}}(v_i)$ ($\mathcal{E}_{\text{FACE}}(v_i)$, resp.), then $\mathfrak{e}$ itself should be in the interior (exterior, resp.) of v_i. If one incident face of $\mathfrak{e}$ is in $\mathcal{I}_{\text{FACE}}(v_i)$ while the other is in $\mathcal{E}_{\text{FACE}}(v_i)$, we know that $\mathfrak{e}$ should be on the boundary of v_i (i.e. $\mathfrak{e} \in \mathcal{B}_{\text{EDGE}}(v_i)$). Moreover, suppose $\mathfrak{e}$ is a boundary edge of l_j (i.e. $\mathfrak{e} \in \mathcal{B}_{\text{EDGE}}(l_j)$). If $l_j \mathsf{NTPP} v_i$, then $\mathfrak{e}$ should lie in the interior of v_i (i.e. $\mathfrak{e} \in \mathcal{I}_{\text{EDGE}}(v_i)$); if $v_i \mathsf{DC} l_j$ or $v_i \mathsf{NTPP} l_j$, then $\mathfrak{e}$ should lie in the exterior of v_i (i.e. $\mathfrak{e} \in \mathcal{E}_{\text{EDGE}}(v_i)$).

In the same way, we define $\mathcal{I}_{\mathrm{VTX}}(v_i)$, $\mathcal{E}_{\mathrm{VTX}}(v_i)$ and $\mathcal{B}_{\mathrm{VTX}}(v_i)$:

$$\mathcal{I}_{\mathrm{VTX}}(v_i) = \{\mathfrak{v} \in \mathrm{VTX} : S_{\mathrm{FACE}}(\mathfrak{v}) \subseteq \mathcal{I}_{\mathrm{FACE}}(v_i)\} \cup \bigcup \{\mathcal{B}_{\mathrm{VTX}}(l_j) : l_j \mathsf{NTPP} v_i\}, \tag{18}$$

$$\mathcal{E}_{\mathrm{VTX}}(v_i) = \{\mathfrak{v} \in \mathrm{VTX} : S_{\mathrm{FACE}}(\mathfrak{v}) \subseteq \mathcal{E}_{\mathrm{FACE}}(v_i)\} \cup \bigcup \{\mathcal{B}_{\mathrm{VTX}}(l_j) : v_i \mathsf{DC}\ l_j \text{or } v_i \mathsf{NTPP} l_j\}, \tag{19}$$

$$\mathcal{B}_{\mathrm{VTX}}(v_i) = \{\mathfrak{v} \in \mathrm{VTX} : S_{\mathrm{FACE}}(\mathfrak{v}) \cap \mathcal{I}_{\mathrm{FACE}}(v_i) \neq \varnothing, S_{\mathrm{FACE}}(\mathfrak{v}) \cap \mathcal{E}_{\mathrm{FACE}}(v_i) \neq \varnothing\}. \tag{20}$$

Note that $S_{\mathrm{FACE}}(\mathfrak{v})$ may contain multiple faces while $S_{\mathrm{FACE}}(\mathfrak{e})$ contains exactly two faces.

Proposition 10. *Suppose $(V \uplus L, \Gamma)$ is an instance of $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each l_i is a polygon. If Γ is path-consistent, then for each variable v_i we have*

(1) $\mathcal{I}_{\mathrm{FACE}}(v_i) \cap \mathcal{E}_{\mathrm{FACE}}(v_i) = \varnothing$.

(2) $\mathcal{I}_{\mathrm{VTX}}(v_i)$, $\mathcal{E}_{\mathrm{VTX}}(v_i)$, *and* $\mathcal{B}_{\mathrm{VTX}}(v_i)$ *are pairwise disjoint.*

(3) $\mathcal{I}_{\mathrm{EDGE}}(v_i)$, $\mathcal{E}_{\mathrm{EDGE}}(v_i)$, *and* $\mathcal{B}_{\mathrm{EDGE}}(v_i)$ *are pairwise disjoint.*

Proof. (1) can be proven in the same way as Proposition 8. The remaining two can be similarly proven. Here we only show $\mathcal{I}_{\mathrm{VTX}}(v_i) \cap \mathcal{B}_{\mathrm{VTX}}(v_i) = \varnothing$ as an example.

Suppose otherwise that there exists a vertex $\mathfrak{v} \in \mathrm{VTX}$ such that $\mathfrak{v} \in \mathcal{I}_{\mathrm{VTX}}(v_i)$ and $\mathfrak{v} \in \mathcal{B}_{\mathrm{VTX}}(v_i)$. Because $\mathfrak{v} \in \mathcal{B}_{\mathrm{VTX}}(v_i)$ we know there exist $\mathfrak{f}_1, \mathfrak{f}_2$ that are incident to $\mathfrak{v}$ and $\mathfrak{f}_1 \in \mathcal{I}_{\mathrm{FACE}}(v_i)$, $\mathfrak{f}_2 \in \mathcal{E}_{\mathrm{FACE}}(v_i)$. This implies that not all incident faces of $\mathfrak{v}$ are in $\mathcal{I}_{\mathrm{FACE}}(v_i)$. Therefore, by $\mathfrak{v} \in \mathcal{I}_{\mathrm{VTX}}(v_i)$, we know there exists a landmark l_j such that $l_j \mathsf{NTPP} v_i$ and $\mathfrak{v} \in \mathcal{B}_{\mathrm{VTX}}(l_j)$.

As $\mathfrak{f}_2 \in \mathcal{E}_{\mathrm{FACE}}(v_i)$, by definition, we know that there exists a landmark l_k such that either (i) $\mathfrak{f}_2 \in \mathcal{I}_{\mathrm{FACE}}(l_k)$ and $v_i \mathsf{DC} l_k$ or $v_i \mathsf{EC} l_k$; or (ii) $\mathfrak{f}_2 \in \mathcal{E}_{\mathrm{FACE}}(l_k)$ and $v_i \mathsf{NTPP} l_k$ or $v_i \mathsf{TPP} l_k$. Note that Γ is path-consistent. Case (i) implies that $\mathfrak{f}_2 \subset l_k$ and $l_j \mathsf{DC} l_k$. Because $\mathfrak{v}$ is incident to $\mathfrak{f}_2$, this shows that $\mathfrak{v}$ is in l_k. By $\mathfrak{v} \in \mathcal{B}_{\mathrm{VTX}}(l_j)$ we also have $\mathfrak{v} \in l_j$. This contradicts the conclusion $l_j \mathsf{DC} l_k$. In Case (ii), we have $l_j \mathsf{NTPP} l_k$ and $\mathfrak{f}_2 \cap l_k = \varnothing$. This also leads to a contradiction, because $l_j \mathsf{NTPP} l_k$ implies $\mathfrak{v}$ is in the interior of l_k, and $\mathfrak{f}_2 \cap l_k = \varnothing$ implies that $\mathfrak{v}$ is not in the interior of l_k.

Therefore, we have $\mathcal{I}_{\mathrm{VTX}}(v_i) \cap \mathcal{B}_{\mathrm{VTX}}(v_i) = \varnothing$. □

For convenience, we define

$$\mathcal{P}_{\mathrm{FACE}}(v_i) = \mathrm{FACE} - \mathcal{I}_{\mathrm{FACE}}(v_i) - \mathcal{E}_{\mathrm{FACE}}(v_i), \tag{21}$$

$$\mathcal{P}_{\mathrm{EDGE}}(v_i) = \mathrm{EDGE} - \mathcal{I}_{\mathrm{EDGE}}(v_i) - \mathcal{E}_{\mathrm{EDGE}}(v_i), \tag{22}$$

$$\mathcal{P}_{\mathrm{VTX}}(v_i) = \mathrm{VTX} - \mathcal{I}_{\mathrm{VTX}}(v_i) - \mathcal{E}_{\mathrm{VTX}}(v_i), \tag{23}$$

Constraint	Conditions
$v_i\mathsf{EC}l_j$	(25)
$v_i\mathsf{PO}l_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$
$v_i\mathsf{TPP}l_j$	$\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$ and (25)
$l_j\mathsf{TPP}v_i$	$\mathcal{E}_{\textsc{Face}}(v_i) \neq \mathcal{E}_{\textsc{Face}}(l_j)$ and (25)
$v_i\mathsf{DC}v_j$	$S_i \cap S_j = \varnothing$
$v_i\mathsf{EC}v_j$	(26)
$v_i\mathsf{PO}v_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$
$v_i\mathsf{TPP}v_j$	$\mathcal{P}_{\textsc{Face}}(v_j) \neq \varnothing$ or $\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(v_j)$, and (26)

Table 8: Conditions for extended RCC5 constraint network

where $\mathcal{P}$ denotes 'pending'. We note that while $\mathcal{B}_{\textsc{Vtx}}(v_i)$ is the set of vertices that *must* lie on the boundary of v_i, $\mathcal{P}_{\textsc{Vtx}}(v_i)$ contains all the vertices that *may* lie on the boundary of v_i. The pairwise disjointness of $\mathcal{I}_{\textsc{Vtx}}(v_i), \mathcal{E}_{\textsc{Vtx}}(v_i)$ and $\mathcal{B}_{\textsc{Vtx}}(v_i)$ implies $\mathcal{B}_{\textsc{Vtx}}(v_i) \subseteq \mathcal{P}_{\textsc{Vtx}}(v_i)$.

Suppose Γ is consistent and has a solution $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n$. Write $\bar{S}_i$ for the set of vertices on the boundary of $\bar{v}_i$, i.e., $\bar{S}_i = \{\mathfrak{v} \in \textsc{Vtx} : \mathfrak{v} \in \partial\bar{v}_i$. Then it is straightforward to show that

$$\bar{S}_i \cap \mathcal{I}_{\textsc{Vtx}}(v_i) = \varnothing,\ \bar{S}_i \cap \mathcal{E}_{\textsc{Vtx}}(v_i) = \varnothing,\ \text{and } \mathcal{B}_{\textsc{Vtx}}(v_i) \subseteq \bar{S}_i \subseteq \mathcal{P}_{\textsc{Vtx}}(v_i). \tag{24}$$

As we have seen in the reduction, determining $\bar{S}_i$ could be intractable. If all $\bar{S}_i$ are given in advance as a constraint for spatial variable v_i (i.e., we explicitly specify whether vertex $\mathfrak{e}$ in the overlay is on the boundary of v_i for all $\mathfrak{e}$ and v_i), then the existence of such a solution can be determined in polynomial time.

Lemma 6. *Suppose $(V \uplus L, \Gamma)$ is an instance of* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$*, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each l_i is a polygon. Assume furthermore that S_i is a subset of* $\textsc{Vtx}$ *for $i = 1, 2, \ldots, n$. If Γ is path-consistent, then $(V \uplus L, \Gamma)$ has a solution $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ such that $\partial\bar{v}_i \cap \textsc{Vtx} = S_i$ if and only if*

(a) $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$ *and* $\mathcal{B}_{\textsc{Vtx}}(v_i) \subseteq S_i \subseteq \mathcal{P}_{\textsc{Vtx}}(v_i)$ *for each v_i.*

(b) All the conditions in Table 8 hold, where

$$\mathcal{P}_{\mathrm{EDGE}}(v_i) \cap \mathcal{B}_{\mathrm{EDGE}}(l_j) \neq \varnothing \quad \textit{or} \quad S_i \cap \mathcal{B}_{\mathrm{VTX}}(l_j) \neq \varnothing, \tag{25}$$

$$\mathcal{P}_{\mathrm{FACE}}(v_i) \cap \mathcal{P}_{\mathrm{FACE}}(v_j) \neq \varnothing \quad \textit{or} \quad \mathcal{P}_{\mathrm{EDGE}}(v_i) \cap \mathcal{P}_{\mathrm{EDGE}}(v_j) \neq \varnothing \quad \textit{or} \quad S_i \cap S_j \neq \varnothing. \tag{26}$$

Proof. See Appendix B. □

Based on this result, we have the following theorem.

Theorem 11. *Suppose all landmarks are represented by polygons. Then the consistency problem* $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ *is NP-complete. Moreover, the consistency problems* $\mathrm{CSPSAT}_f(\mathcal{B}_{RCC8})$, $\mathrm{CSPSAT}_s(RCC8)$, *and* $\mathrm{CSPSAT}_f(RCC8)$ *are all NP-complete.*

Proof. We propose a nondeterministic algorithm which solves $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$. The algorithm first guesses a configuration of S_i and uses it as an additional constraint, then determines the consistency by Lemma 11. Note that each S_i has $O(K^2)$ points, which are polynomial in the input size. Thus guessing a configuration of S_i takes polynomial time. Meanwhile, checking all the conditions also takes polynomial time. Therefore, the extended consistency problem $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ is in NP, and hence NP-complete as its NP-hardness has been confirmed in Proposition 9.

By Proposition 1 (ii) and (iii) we know $\mathrm{CSPSAT}_f(\mathcal{B}_{RCC8})$, $\mathrm{CSPSAT}_s(RCC8)$, and $\mathrm{CSPSAT}_f(RCC8)$ are all in NP. Meanwhile, they are also NP-hard because they all contain the NP-hard problem $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ as a sub-problem. Therefore, they are all NP-complete. □

Remark 2. *Recall that in the reduction from the Graph 3-Colouring problem to* $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ *the landmark l is a concave polygon which has three meeting points with landmark l' (see Figure 6(a)). This property of landmarks plays a critical role in designing the reduction. Another reduction from the 3-SAT problem to* $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$*, given in [29], also uses concave landmarks. Note that two convex polygons cannot have multiple isolated meeting points (i.e. they either have only one meeting point or share a line segment). One may conjecture that the consistency problem* $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ *becomes tractable if all landmarks are represented as* convex *polygons. This, however, is not true.*

In fact, a polynomial reduction from 3-SAT to $\mathrm{CSPSAT}_s(\mathcal{B}_{RCC8})$ *exists even if all landmarks are represented by rectangles with edges parallel to the coordinate* *axes. The reduction is more complicated than the reduction provided in the proof of Proposition 9. The main idea is, although landmarks are all convex regions, spatial variables can be interpreted as arbitrary regions, and we can constrain a spatial variable by using these rectangular landmarks in a way such that it may have multiple meeting points with some landmark. For example, suppose* l_0, l_1, l_2 *are three rectangles as shown in Figure 7, where* l_1**TPP**l_0, l_0**EC**l_2 *and* l_1**EC**l_2. *Assume that* v *is a spatial variable and* v**TPP**l_0, v**EC**l_1 *and* v**EC**l_2. *These constraints require* v *to contain (at least) one of the two points* Q^+ *and* Q^-*, which may be used to simulate a propositional variable. Based on this obversion, a reduction from 3-SAT can be devised. Therefore, the consistency problem* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ *remains NP-hard even for rectangular landmarks.*

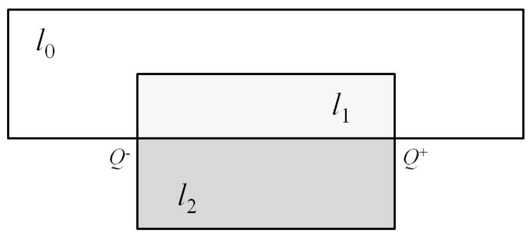


Figure 7: Illustration for simulating a propositional variable using rectangular landmarks

Remark 3. *In practice, we may reduce the problem* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ *to SAT (i.e. deciding the satisfiability of propositional formulas in conjunctive normal form). As stated in the proof of Theorem 11,* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ *is equivalent to deciding whether there exist* $S_i \subseteq \textsc{Vtx}$ *for each* i *such that all the conditions in Lemma 6 are satisfied. Note that, once the instance is given, the conditions in the lemma can be simplified (in polynomial time) into a set of conditions concerning* S_i *of the following forms:* $R \subseteq S_i \subseteq R'$, $S_i \cap R \neq \varnothing$, $S_i \cap S_j = \varnothing$, *and* $S_i \cap S_j \neq \varnothing$, *where* R *and* R' *are subsets of* $\textsc{Vtx}$ *determined by the instance. For each* S_i *and each vertex* $\mathfrak{v} \in \textsc{Vtx}$, *we introduce a propositional variable which is assigned* true *iff* $\mathfrak{v}$ *is in* S_i. *In this way, each condition in one of the above forms is transformed into a disjunction clause or a number of disjunction clauses, and thus a* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ *instance is transformed into an equivalent SAT instance. Therefore,* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ *can be reduced to SAT, which enables us to solve the problem by the well-developed SAT solvers.*

The NP-hardness of $\textsc{CSPSAT}_s(\mathcal{B}_{RCC8})$ is quite undesirable, as it is the simplest and most fundamental case of introducing landmarks to reasoning with RCC8. In the following subsection, we show that the same problem becomes tractable if we interpret RCC8 relations by using a stronger connectedness relation.

4.4.3. RCC8 model based on strong connectedness

In the standard RCC8 model, two regions are considered to be connected if they have a common point. Consequently, two externally connected (EC) regions may share one or more isolated boundary points (see Figure 6(a)). In this subsection, we turn to another interpretation of RCC8, which uses a stronger version of connectedness: two regions are considered as connected if they share a common *curve*, where a *curve* is defined as a topological embedding of the closed interval [0,1] in the plane. As a result, two non-overlapping regions are externally connected iff their boundaries share at least a curve. Formally, we have

Definition 8 (RCC8 algebra based on strong connectedness)**.** *Let U be the set of nonempty regular closed sets, or* regions*, in the real plane. The RCC8 algebra based on strong connectedness, written RCC8′, is generated by the following eight topological relations*

$$\mathsf{DC}, \mathsf{EC}, \mathsf{PO}, \mathsf{EQ}, \mathsf{TPP}, \mathsf{NTPP}, \mathsf{TPPi}, \mathsf{NTPPi},$$

where TPPi and NTPPi are the converses of TPP and NTPP respectively, and EQ is the identity relation, and for two regions a, b,

- *$a\mathsf{DC}b$ iff $a \cap b$ does not contain any curve;*
- *$a\mathsf{EC}b$ iff $a^\circ \cap b^\circ = \varnothing$ and $a \cap b$ contains at least one curve;*
- *$a\mathsf{NTPP}b$ iff $a \subset b$ and $\partial a \cap \partial b$ does not contain any curve;*
- *$a\mathsf{TPP}b$ iff $a \subset b$ and $\partial a \cap \partial b$ contains at least one curve;*
- *$a\mathsf{PO}b$ iff $a^\circ \cap b^\circ \neq \varnothing$ and $a \not\subseteq b$, $b \not\subseteq a$.*

It is easy to see that this connectedness relation (i.e. the complement of DC) is stronger than (i.e. contained in) the connectedness relation given in Definition 4.

Intuitively, the NP-hardness of $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ (for weak connectedness) is due to that there are exponentially many possibilities of S_i (the intersection of $\textsc{Vtx}$ and the boundary of v_i), since points in S_i may be evidences of EC constraints (cf. the reduction in Section 4.4.1). In the strong connectedness interpretation, however, isolated meeting points have no effects on RCC8 relations. Therefore S_i may be ignored safely and the problem $\textsc{cspsat}_s(\mathcal{B}_{RCC8'})$ becomes tractable, as shown in the following theorem.

Theorem 12. *The consistency problem* $\textsc{cspsat}_s(\mathcal{B}_{RCC8'})$ *can be decided in polynomial time.*

The computational complexity of $\textsc{cspsat}_s(\mathcal{B}_{RCC8'})$ is the same as that of $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ (see Theorem 9), as the argument for RCC5 still applies here. Precisely, the consistency of an instance of $\textsc{cspsat}_s(\mathcal{B}_{RCC8'})$ can be decided by checking the conditions in Lemma 6 and neglecting all conditions involving S_i. That is, we discard the following conditions:

- the condition $\mathcal{B}_{\mathrm{VTX}}(v_i) \subseteq S_i \subseteq \mathcal{P}_{\mathrm{VTX}}(v_i)$ in condition (a);
- the condition $S_i \cap S_j = \varnothing$ whenever $(v_i \mathsf{DC} v_j) \in \Gamma$ in Row 4 of Table 8;
- the disjunct $S_i \cap \mathcal{B}_{\mathrm{VTX}}(l_j) \neq \varnothing$ in (25);
- the disjunct $S_i \cap S_j \neq \varnothing$ in (26).

The above theorem can be proven by modifying the proof of Lemma 6 with a slightly different construction. The proof sketch is provided in Appendix C.

Remark 4. *The strong connectedness introduced above has been considered in [3, 8]. In particular, in [3], Borgo, Guarino, and Masolo argued that the classical Whiteheadian connectedness may be considered too weak in many cases. For example, "a worm cannot pass from the interior of one apple to another, which touch just at a point, without becoming visible to the exterior – so from the worm's point of view we might as well say that the apples are not 'sufficiently' connected."*

As far as consistency and realisations are concerned, Li [19] has shown that any consistent RCC8 network has a solution in any RCC model. The cubic realisation algorithm described there can be easily adapted to construct a solution in the RCC8 model based on strong connectedness. This implies in particular that an RCC8 network (without landmarks) has a solution in the RCC8 model with 'weak' connectedness iff it has a solution in the RCC8 model with 'strong' connectedness.

5. Conclusion and future work

One major difference between qualitative CSPs and classical CSPs is that the domain of a qualitative CSP is always infinite, while that of a classical CSP is usually finite. In this paper we proposed an extended framework for qualitative CSPs that supports finite domains. In the extended framework, a spatial/temporal variable could take values from a finite domain or even a singleton. This reflects demands in applications such as urban planning and spatial query processing where additional knowledge about variables may be available. We believe this extension is necessary to bring QSTR closer to real-world applications.

We then investigated the computational complexity of solving the extended consistency problem for five very important qualitative calculi, viz. PA, IA, CRA, RCC5 and RCC8. The results were summarised in Table 4, where for each calculus, we determined whether each of the four variants of the consistency problem is in P or NP-complete. Recall that the classical consistency problem is NP-complete for IA, CRA, RCC5 and RCC8. This shows that, in general, the expressiveness of the extended framework of qualitative CSP does not incur additional cost in computational complexity for these calculi. Under practical assumptions, we also provided efficient algorithms for solving basic constraints involving landmarks for all these calculi.

While this paper introduces landmarks in qualitative CSPs, there is a related work in classical CSPs. Recently, Bulatov [5] has given a full classification of computational complexity for conservative constraint satisfaction problems with finite values, in which the set of values for each individual variable can be restricted arbitrarily. The solving algorithm and the proofs given there heavily use the algebraic approach to (classical) CSP developed in [17, 6]. One interesting future research direction will be investigating the possibility of applying the solving algorithm given in [5], and, more generally, the algebraic approach, to solving qualitative CSPs involving landmarks. We refer the reader to [2] for recent progresses of applying the algebraic approach for attacking qualitative CSPs.

In this paper, we have confined ourselves to the five most important qualitative calculi, which are all binary calculi. The framework can be straightforwardly extended to any other qualitative calculus, binary or ternary, but the computational complexity has to be examined case by case. Take the ternary calculus LR [23] as an example. It has been shown that reasoning with complete basic and landmark-free LR networks is already at least NP-hard and its NP-membership is still open [46]. As a consequence, reasoning with complete basic LR networks involving landmarks is also NP-hard. Another direction of future research will be investigating the computation complexity for other well-known calculi, individually or combined together. Because most of these consistency problems are at least NP-hard, it is also necessary to develop either approximate methods or practical methods (e.g. those in [37, 18]) for solving qualitative (binary or ternary) CSPs involving landmarks.

Acknowledgements

We thank the three anonymous referees and the handling associate editor for their invaluable suggestions, and, especially, for their suggestion that using Graph 3-Colouring instead of 3-SAT for proving the NP-hardness. We also thank Steven Schockaert for suggesting us the MacDonald's example and thank Jochen Renz and Tony Cohn for fruitful discussions on our joint ARC Discovery Project "Approximate Reasoning with Qualitative Spatial Constraints Involving Landmarks."

This work was partially supported by Australian Research Council (FT0990811, DP120104159) and National Natural Science Foundation of China (61133011, 61228305).

Appendix A. Proof of Theorem 8 (Sufficiency)

The sufficiency part is proven by a realisation algorithm which generates a solution of the constraint network. The algorithm is similar to the classical realisation algorithm introduced in [19, 22]. We first construct for each variable v_i a region a_i such that $\{a_1, a_2, \ldots, a_n\}$ satisfies all except the PP constraints, and then construct regions $\{c_1, c_2, \ldots, c_n\}$ which is a solution of Γ.

For each variable v_i, we define

$$\mathcal{P}_{\textsc{Face}}(v_i) = \textsc{Face} - \mathcal{I}_{\textsc{Face}}(v_i) - \mathcal{E}_{\textsc{Face}}(v_i). \tag{A.1}$$

A number of 'base regions' are necessary in the construction of $\{a_1, a_2, \ldots, a_n\}$. Base regions are arbitrarily selected, as long as they are pairwise disjoint polygons and are so small that their union does not contain any face. We use X_i to denote the set of base regions being selected for variable v_i. The construction is as follows, where each X_i is initialised as the empty set.

1. For each face $\mathfrak{f} \in \mathcal{P}_{\textsc{Face}}(v_i)$, select a base region contained in $\mathfrak{f}$ and put it into X_i.

2. For any $i < j$ such that $(v_i \mathsf{PO} v_j) \in \Gamma$ and $\mathcal{P}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j) \neq \varnothing$, select a face $\mathfrak{f}$ in $\mathcal{P}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j)$ and a base region contained in $\mathfrak{f}$. Put the base region into both X_i and X_j.

3. For each i, let $a_i = \bigcup X_i$.

4. For each i, let $b_i = a_i \cup \bigcup\{a_j : (v_j \mathsf{PP} v_i) \in \Gamma\}$.

5. For each i, let $c_i = b_i \cup \bigcup\{l_j : (l_j \mathsf{PP} v_i) \in \Gamma\}$.

Lemma 7. *Suppose* $(V \uplus L, \Gamma)$ *is an instance of* $\textsc{CSPSAT}_s(\mathcal{B}_{RCC5})$*, where* $V = \{v_1, v_2, \ldots, v_n\}$*,* $L = \{l_1, l_2, \ldots, l_m\}$*, and each* l_i *is a polygon. Suppose* Γ *is path-consistent. Assume that* a_i, b_i, c_i *(*$1 \le i \le n$*) are as in the construction given above. Then for each face* $\mathfrak{f} \in \textsc{Face}$ *we have*

- $\mathfrak{f} \in \mathcal{I}_{\textsc{Face}}(v_i)$ *iff* $\mathfrak{f} \subseteq c_i$.
- $\mathfrak{f} \in \mathcal{E}_{\textsc{Face}}(v_i)$ *iff* $\mathfrak{f} \cap c_i = \varnothing$.
- $\mathfrak{f} \in \mathcal{P}_{\textsc{Face}}(v_i)$ *iff* $\mathfrak{f} \nsubseteq c_i$ *and* $\mathfrak{f} \cap c_i \neq \varnothing$.

Proof. We first prove the necessity part.

Suppose $\mathfrak{f} \in \mathcal{I}_{\textsc{Face}}(v_i)$. There exists a landmark l such that $\mathfrak{f} \in \mathcal{I}_{\textsc{Face}}(l)$ and $l\mathsf{PP}v_i$. Because $l \subseteq c_i$, the first statement holds directly.

Assume $\mathfrak{f} \in \mathcal{E}_{\textsc{Face}}(v_i)$. Because each base region in X_i is contained in a face in $\mathcal{P}_{\textsc{Face}}(v_i)$, we know that $\mathfrak{f} \cap a_i = \varnothing$. Suppose $(v_j\mathsf{PP}v_i) \in \Gamma$. By the definition of $\mathcal{E}_{\textsc{Face}}(v_j)$ and the path-consistency of Γ, it is direct to prove that $\mathfrak{f}$ is also in $\mathcal{E}_{\textsc{Face}}(v_j)$. Therefore we have $\mathfrak{f} \cap a_j = \varnothing$, and thus $\mathfrak{f} \cap b_i = \varnothing$ by the construction of b_i. Similarly, for any landmark l such that $(l\mathsf{PP}v_i) \in \Gamma$, we can prove that $\mathfrak{f} \cap l = \varnothing$. Therefore, we have $\mathfrak{f} \cap c_i = \varnothing$.

Now assume $\mathfrak{f} \in \mathcal{P}_{\textsc{Face}}(v_i)$. Clearly we have $\mathfrak{f} \cap a_i \neq \varnothing$ because X_i has a base region contained in $\mathfrak{f}$. We only need to prove $\mathfrak{f} \nsubseteq c_i$. By the selection of base regions, $\mathfrak{f}$ is not contained in the union of all base regions, and hence it is not contained in b_i. Moreover, for any landmark l_j, if $(l_j\mathsf{PP}v_i) \in \Gamma$, then $\mathfrak{f} \in \mathcal{E}_{\textsc{Face}}(l_j)$ (otherwise, $\mathfrak{f} \in \mathcal{I}_{\textsc{Face}}(l_j) \subseteq \mathcal{I}_{\textsc{Face}}(v_i)$). That is to say, $\mathfrak{f}$ is disjoint with l_j. Therefore, $\mathfrak{f} \nsubseteq c_i$.

The sufficiency part follows from $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{P}_{\textsc{Face}}(v_i) = \textsc{Face}$. □

Corollary 2. *Let* $(V \uplus L, \Gamma)$ *and* c_i *be as in Lemma 7. Furthermore, suppose* $(V \uplus L, \Gamma)$ *satisfies all the conditions in Theorem 8. Then* $\{c_1, c_2, \ldots, c_n\}$ *satisfies all the constraints in* Γ *of the form* $v_i \alpha l_j$*.*

Proof. Because $(V \uplus L, \Gamma)$ satisfies the conditions in Theorem 8, we know in particular that $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$ for each $1 \le i \le n$. That is, there exists a face $\mathfrak{f}$ in $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{P}_{\textsc{Face}}(v_i)$. By Lemma 7, this implies that each c_i is nonempty.

(1) If $(v_i\mathsf{PP}l_j) \in \Gamma$, then we have $\mathcal{E}_{\textsc{Face}}(l_j) \subseteq \mathcal{E}_{\textsc{Face}}(v_i)$ by (10). Lemma 7 directly implies that $c_i \subseteq l_j$. Because $\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$ (Row 2 in Table 7), there exists a face $\mathfrak{f}$ which is in $\mathcal{I}_{\textsc{Face}}(l_j)$ but not in $\mathcal{I}_{\textsc{Face}}(v_i)$. By Lemma 7, $\mathfrak{f}$ is not contained in c_i. Therefore, $c_i \subset l_j$, i.e. $c_i\mathsf{PP}l_j$.

(2) If $(l_j \mathsf{PP} v_i) \in \Gamma$, clearly we have $l_j \subseteq c_i$. Because $\mathcal{E}_{\text{FACE}}(v_i) \neq \mathcal{E}_{\text{FACE}}(l_j)$ (Row 3 in Table 7) and $\mathcal{E}_{\text{FACE}}(v_i) \subseteq \text{FACE} - \mathcal{I}_{\text{FACE}}(v_i) \subseteq \text{FACE} - \mathcal{I}_{\text{FACE}}(l_j) = \mathcal{E}_{\text{FACE}}(l_j)$, we know that $\mathcal{E}_{\text{FACE}}(v_i) \subset \mathcal{E}_{\text{FACE}}(l_j)$, i.e. there exists a face $\mathfrak{f}$ in $\mathcal{E}_{\text{FACE}}(l_j)$ but not in $\mathcal{E}_{\text{FACE}}(v_i)$. Therefore $\mathfrak{f} \cap l_j = \varnothing$ and $\mathfrak{f} \cap c_i \neq \varnothing$. That is, $l_j \subset c_i$, i.e. $l_j \mathsf{PP} c_i$.

(3) If $(v_i \mathsf{DR}\ l_j) \in \Gamma$, then we have $\mathcal{I}_{\text{FACE}}(l_j) \subseteq \mathcal{E}_{\text{FACE}}(v_i)$. Lemma 7 directly implies that $c_i \cap l_j^\circ = \varnothing$, i.e. $c_i \mathsf{DR}\ l_j$.

(4) If $(v_i \mathsf{PO} l_j) \in \Gamma$, then by Row 1 in Table 7, we know that $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(l_j) \neq \text{FACE}$. That is, there exists a face $\mathfrak{f}$ such that $\mathfrak{f} \notin \mathcal{E}_{\text{FACE}}(v_i)$ and $\mathfrak{f} \notin \mathcal{E}_{\text{FACE}}(l_j)$ (hence $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_j)$). Therefore $\mathfrak{f} \subseteq l_j$ and $\mathfrak{f} \cap c_i \neq \varnothing$ by Lemma 7, and hence c_i overlaps l_j, i.e. they have a common interior point. It can be proven that $c_i \nsubseteq l_j$ and $l_j \nsubseteq c_i$ as in the first two cases above. Therefore, $c_i \mathsf{PO} l_j$ holds. □

We next prove that $\{c_1, \ldots, c_n\}$ is a solution of Γ.

Lemma 8. *Let* $(V \uplus L, \Gamma)$ *and* c_i *be as in Corollary 2. Then* $\{c_1, \ldots, c_n\}$ *is a solution of* $(V \uplus L, \Gamma)$.

Proof. We only need to prove that constraints of the form $(v_i \alpha v_j)$ are satisfied.

(1) If $(v_i \mathsf{PP} v_j) \in \Gamma$, it can be proven that $b_i \subseteq b_j$ and $c_i \subseteq c_j$ by the path-consistency of Γ. We next prove $c_i \neq c_j$. By $\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$ (last row in Table 7), there exists a face $\mathfrak{f}$ that is in neither $\mathcal{I}_{\text{FACE}}(v_i)$ nor $\mathcal{E}_{\text{FACE}}(v_j)$. Therefore $\mathfrak{f}$ is either in $\mathcal{E}_{\text{FACE}}(v_i)$ or $\mathcal{P}_{\text{FACE}}(v_i)$. If $\mathfrak{f} \in \mathcal{E}_{\text{FACE}}(v_i)$, then $\mathfrak{f} \cap c_i = \varnothing$. By Lemma 7 and $\mathfrak{f} \notin \mathcal{E}_{\text{FACE}}(v_j)$, we also know $\mathfrak{f} \cap c_j \neq \varnothing$, and thus $c_i \neq c_j$. Now suppose $\mathfrak{f} \in \mathcal{P}_{\text{FACE}}(v_i)$. By Lemma 7 we have $\mathfrak{f} \nsubseteq c_i$. Note that $\mathfrak{f}$ is in either $\mathcal{I}_{\text{FACE}}(v_j)$ or $\mathcal{P}_{\text{FACE}}(v_j)$. In the first case, we have $\mathfrak{f} \subseteq c_j$ and thus $c_i \neq c_j$. In the second case, by the construction of X_j we know that there exists some base region r contained in $\mathfrak{f}$ that belongs to X_j only. Therefore r is disjoint with a_i and hence disjoint with b_i. Moreover, r cannot be contained in c_i. Otherwise, there must exist some landmark l such that $l \mathsf{PP} v_i$ and $r \subseteq l$. This implies that $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l)$, which further implies $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i)$, a contradiction. Therefore, we have $r \nsubseteq c_i$ and $r \subseteq c_j$ and thus $c_i \neq c_j$. In conclusion, we know $c_i \subset c_j$, i.e. $c_i \mathsf{PP} c_j$.

(2) If $(v_i \mathsf{DR} v_j) \in \Gamma$, we show $c_i \cap c_j^\circ = \varnothing$. By construction we have $a_i \cap a_j = \varnothing$, because $X_i \cap X_j = \varnothing$ unless $(v_i \mathsf{PO} v_j) \in \Gamma$. Note that $(v_k \mathsf{PP} v_i) \in \Gamma$ implies $(v_k \mathsf{DR} v_j) \in \Gamma$ by path-consistency. Therefore, we also have $a_k \cap a_j = \varnothing$. By the construction of b_i we know $b_i \cap a_j = \varnothing$. Similarly we can prove that $b_i \cap b_j = \varnothing$. In the same way, it can be further proven that $c_i \cap c_j^\circ = \varnothing$, i.e. $c_i \mathsf{DR} c_j$.

(3) If $(v_i \mathsf{PO} v_j) \in \Gamma$, we first show that c_i overlaps c_j. By $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$ (Row 6 in Table 7), there exists a face $\mathfrak{f}$ such that $\mathfrak{f} \notin \mathcal{E}_{\text{FACE}}(v_i)$ and $\mathfrak{f} \notin \mathcal{E}_{\text{FACE}}(v_j)$. In other words, we have $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{P}_{\text{FACE}}(v_i)$ and $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_j) \cup$

$\mathcal{P}_{\text{FACE}}(v_j)$. If $\mathfrak{f} \in \mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$, then by the construction of a_i and a_j there exists a base region r selected from a face in $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$ (not necessarily $\mathfrak{f}$) such that $r \in X_i \cap X_j$. Therefore, $r \subseteq a_i \cap a_j$ and hence $r \subseteq c_i \cap c_j$. If $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{I}_{\text{FACE}}(v_j)$, then $\mathfrak{f}$ is contained in both c_i and c_j. If $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$, then we know $\mathfrak{f} \subseteq c_i$ and $\mathfrak{f} \cap c_j \neq \varnothing$. Thus c_i also overlaps c_j. The last case can be proven similarly. Therefore c_i overlaps c_j. It remains to show that c_i and c_j are incomparable (i.e., one is not contained in the other). This can be proven in the same way as in the case of $(v_i \mathsf{PO} l_j) \in \Gamma$. In conclusion, we know $c_i \mathsf{PO} c_j$.

In summary, all the constraints are satisfied and $\{c_1, \ldots, c_n\}$ is a solution of Γ. □

Appendix B. Proof of Lemma 6

Appendix B.1. Necessity

Suppose $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution of Γ and $\bar{v}_i \cap \text{VTX} = S_i$ for each i. By the definitions of $\mathcal{B}_{\text{VTX}}(v_i)$ and $\mathcal{P}_{\text{VTX}}(v_i)$, it is straightforward to show that $\mathcal{B}_{\text{VTX}}(v_i) \subseteq S_i \subseteq \mathcal{P}_{\text{VTX}}(v_i)$. Similarly to the RCC5 case, we can prove that $\mathcal{E}_{\text{FACE}}(v_i) \neq \text{FACE}$ for any $v_i \in V$. We first prove the following lemmas.

Lemma 9. *Suppose $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution of Γ, then for any $\bar{v}_1$ we have*

(i) $\mathfrak{f} \subseteq (\bar{v}_i)^\circ$ *for any face* $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i)$*;*

(ii) $\mathfrak{f} \cap \bar{v}_i = \varnothing$ *for any face* $\mathfrak{f} \in \mathcal{E}_{\text{FACE}}(v_i)$*;*

(iii) $\mathfrak{e} \subseteq (\bar{v}_i)^\circ$ *for any edge* $\mathfrak{e}$ *in* $\mathcal{I}_{\text{EDGE}}(v_i)$*;*

(iv) $\mathfrak{e}' \cap \bar{v}_i = \varnothing$ *for any edge* $\mathfrak{e}'$ *in* $\mathcal{E}_{\text{EDGE}}(v_i)$*;*

(v) $\mathfrak{v} \in (\bar{v}_i)^\circ$ *for any vertex* $\mathfrak{v}$ *in* $\mathcal{I}_{\text{VTX}}(v_i)$*;*

(vi) $\mathfrak{v} \notin \bar{v}_i$ *for any vertex* $\mathfrak{v}$ *in* $\mathcal{E}_{\text{VTX}}(v_i)$*.*

Proof. For (i), by the definition of $\mathcal{I}_{\text{FACE}}(v_i)$ (see (9)), there exists a landmark l_k such that $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(l_k)$ and $l_k \mathsf{TPP} v_i$ or $l_k \mathsf{NTPP} v_i$. Thus we have $\mathfrak{f} \subseteq l_k^\circ$ and $l_k \subseteq \bar{v}_i$, and, therefore, $\mathfrak{f} \subseteq (\bar{v}_i)^\circ$. Similarly we have $\mathfrak{f} \cap \bar{v}_i = \varnothing$ for any $\mathfrak{f}$ in $\mathcal{E}_{\text{FACE}}(v_i)$.

For (iii), by the definition of $\mathcal{I}_{\text{EDGE}}(v_i)$ (see (15)), we have either $S_{\text{FACE}}(\mathfrak{e}) \subseteq \mathcal{I}_{\text{FACE}}(v_i)$, or $\mathfrak{e} \in \mathcal{B}_{\text{EDGE}}(l_k)$ for some landmark l_k with $l_k \mathsf{NTPP} v_i$. In the first case, because the two incident faces of $\mathfrak{e}$ are both in $\mathcal{I}_{\text{FACE}}(v_i)$, they are contained in the interior of $\bar{v}_i$. Because $\mathfrak{e}$ is the common boundary of its two incident faces, we know $\mathfrak{e}$ is also contained in $(\bar{v}_i)^\circ$. In the second case, we have $\mathfrak{e} \subseteq l_k \subseteq (\bar{v}_i)^\circ$. Therefore $\mathfrak{e} \subseteq (\bar{v}_i)^\circ$ holds in both cases. Similarly we have $\mathfrak{e}' \cap \bar{v}_i = \varnothing$ for any edge $\mathfrak{e}'$ in $\mathcal{E}_{\textsc{Edge}}(v_i)$.

(v) and (vi) can be proven in the same way. □

Lemma 10. *Suppose $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution of Γ, and $S_i = \bar{v}_i \cap \textsc{Vtx}$ for each i. Then for any v_i and l_j, if $(v_i \mathsf{EC} l_j)$, $(v_i \mathsf{TPP} l_j)$ or $(v_i \mathsf{TPPi} l_j)$ is a constraint in Γ, then (25) holds; for any v_i and v_j, if $(v_i \mathsf{EC} v_j)$ or $(v_i \mathsf{TPP} v_j) \in \Gamma$ is a constraint in Γ, then (26) holds.*

Proof. Suppose one of $(v_i \mathsf{EC} l_j)$, $(v_i \mathsf{TPP} l_j)$, and $(v_i \mathsf{TPPi} l_j)$ is a constraint in Γ. We show (25) holds. Because $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution, we know that $\bar{v}_i$ and l_j have a common boundary point, say P. It is clear that P is either a vertex in $\mathcal{B}_{\textsc{Vtx}}(l_j)$, or on an edge $\mathfrak{e} \in \mathcal{B}_{\textsc{Edge}}(l_j)$. In the first case, we have $P \in \partial\bar{v}_i \cap \textsc{Vtx} = S_i$. Therefore $P \in S_i \cap \mathcal{B}_{\textsc{Vtx}}(l_j)$ and thus (25) is satisfied. In the second case, because $P \in \mathfrak{e}$ and $P \in \partial\bar{v}_i$, we know edge $\mathfrak{e}$ cannot be in the interior of $\bar{v}_i$ or in the exterior of $\bar{v}_i$. By Lemma 9, $\mathfrak{e}$ is in neither $\mathcal{I}_{\textsc{Edge}}(v_i)$ nor $\mathcal{E}_{\textsc{Edge}}(v_i)$, hence $\mathfrak{e} \in \mathcal{P}_{\textsc{Edge}}(v_i)$. Therefore we have $\mathcal{P}_{\textsc{Edge}}(v_i) \cap \mathcal{B}_{\textsc{Edge}}(l_j) \neq \varnothing$ and thus (25) is also satisfied.

The other part of the lemma can be proven similarly. □

The necessity of conditions in Table 8 can then be proven straightforwardly.

Appendix B.2. Sufficiency

Suppose $(V \uplus L, \Gamma)$ and S_i $(i = 1, \ldots, n)$ satisfy the conditions in Lemma 6, we construct a solution $\{\bar{v}_1, \ldots, \bar{v}_n\}$ of Γ such that $S_i = \partial\bar{v}_i \cap \textsc{Vtx}$. The construction procedure is similar to that in [19, 22]. For each spatial variable v_i, we select a set of small triangles, denoted by X_i, in the following way.

- For each face $\mathfrak{f} \in \mathcal{P}_{\textsc{Face}}(v_i)$, select a small triangle in $\mathfrak{f}$ and put it in X_i, see Figure B.8(a).

- For each vertex $\mathfrak{v} \in S_i - \mathcal{B}_{\textsc{Vtx}}(v_i) \subseteq \mathcal{P}_{\textsc{Vtx}}(v_i) - \mathcal{B}_{\textsc{Vtx}}(v_i)$, by Proposition 10 we know that $\mathfrak{v}$ is not in $\mathcal{B}_{\textsc{Vtx}}(v_i) \cup \mathcal{I}_{\textsc{Vtx}}(v_i) \cup \mathcal{E}_{\textsc{Vtx}}(v_i)$. We have that $S_{\textsc{Face}}(\mathfrak{v}) \cap \mathcal{P}_{\textsc{Face}}(v_i) \neq \varnothing$. Otherwise, $S_{\textsc{Face}}(\mathfrak{v})$ is contained in $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_i)$, which implies that $\mathfrak{v}$ is either in $\mathcal{I}_{\textsc{Vtx}}(v_i)$, or in $\mathcal{E}_{\textsc{Vtx}}(v_i)$, or in $\mathcal{B}_{\textsc{Vtx}}(v_i)$. We select a face $\mathfrak{f}$ from $S_{\textsc{Face}}(\mathfrak{v}) \cap \mathcal{P}_{\textsc{Face}}(v_i)$, and select a small triangle in $\mathfrak{f}$ that contains $\mathfrak{v}$. Put the triangle in X_i, see Figure B.8(b).

- If $v_i\mathsf{EC}l_j$ is in Γ, then by Table 8 we have either $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{B}_{\textsc{Edge}}(l_j)\neq\varnothing$ or $S_i\cap\mathcal{B}_{\textsc{Vtx}}(l_j)\neq\varnothing$ (i.e. (25)). If $S_i\cap\mathcal{B}_{\textsc{Vtx}}(l_j)\neq\varnothing$, do nothing. Otherwise, we select an edge $\mathfrak{e}$ from $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{B}_{\textsc{Edge}}(l_j)$. Let $\mathfrak{f}$ and $\mathfrak{f}'$ be the two incident faces of $\mathfrak{e}$ such that $\mathfrak{f}\in\mathcal{I}_{\textsc{Face}}(l_j)$ and $\mathfrak{f}'\in\mathcal{E}_{\textsc{Face}}(l_j)$. By definition, we know $\mathfrak{f}\in\mathcal{E}_{\textsc{Face}}(v_i)$. We note that $\mathfrak{f}'$ cannot be in $\mathcal{E}_{\textsc{Face}}(v_i)$. This is because, otherwise, we have $S_{\textsc{Face}}(\mathfrak{e})=\{\mathfrak{f},\mathfrak{f}'\}\subseteq\mathcal{E}_{\textsc{Face}}(v_i)$ and hence $\mathfrak{e}\in\mathcal{E}_{\textsc{Edge}}(v_i)$, which contradicts the assumption that $\mathfrak{e}\in\mathcal{P}_{\textsc{Edge}}(v_i)$. If $\mathfrak{f}'\in\mathcal{I}_{\textsc{Face}}(v_i)$, do nothing. If $\mathfrak{f}'\in\mathcal{P}_{\textsc{Face}}(v_i)$, select a triangle in face $\mathfrak{f}'$ with one edge on $\mathfrak{e}$ and put it in X_i, see Figure B.8(c).

- If $v_i\mathsf{TPP}l_j$ is in Γ, then by Table 8 we also have $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{B}_{\textsc{Edge}}(l_j)\neq\varnothing$ or $S_i\cap\mathcal{B}_{\textsc{Vtx}}(l_j)\neq\varnothing$ (i.e. (25)). If $S_i\cap\mathcal{B}_{\textsc{Vtx}}(l_j)\neq\varnothing$, do nothing. Otherwise, select an edge $\mathfrak{e}$ from $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{B}_{\textsc{Edge}}(l_j)$. Let $\mathfrak{f}$ and $\mathfrak{f}'$ be the two incident faces of $\mathfrak{e}$ such that $\mathfrak{f}\in\mathcal{I}_{\textsc{Face}}(l_j)$ and $\mathfrak{f}'\in\mathcal{E}_{\textsc{Face}}(l_j)$. By definition, we know $\mathfrak{f}'\in\mathcal{E}_{\textsc{Face}}(v_i)$. Similar to the case of $v_i\mathsf{EC}l_j$, $\mathfrak{f}$ cannot be in $\mathcal{E}_{\textsc{Face}}(v_i)$. If $\mathfrak{f}\in\mathcal{I}_{\textsc{Face}}(v_i)$, do nothing. If $\mathfrak{f}\in\mathcal{P}_{\textsc{Face}}(v_i)$, select a triangle in face $\mathfrak{f}$ with one edge on $\mathfrak{e}$ and put it in X_i.

- If $v_i\mathsf{EC}v_j$ is in Γ, then by Table 8 we have $\mathcal{P}_{\textsc{Face}}(v_i)\cap\mathcal{P}_{\textsc{Face}}(v_j)\neq\varnothing$, or $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{P}_{\textsc{Edge}}(v_j)\neq\varnothing$, or $S_i\cap S_j\neq\varnothing$. If $S_i\cap S_j\neq\varnothing$, do nothing. If $S_i\cap S_j=\varnothing$ and $\mathcal{P}_{\textsc{Face}}(v_i)\cap\mathcal{P}_{\textsc{Face}}(v_j)\neq\varnothing$, select a face $\mathfrak{f}\in\mathcal{P}_{\textsc{Face}}(v_i)\cap\mathcal{P}_{\textsc{Face}}(v_j)$ and two externally connected triangles in $\mathfrak{f}$. Put one triangle in X_i and put the other in X_j, see Figure B.8(d). If $S_i\cap S_j=\varnothing$, $\mathcal{P}_{\textsc{Face}}(v_i)\cap\mathcal{P}_{\textsc{Face}}(v_j)=\varnothing$, and $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{P}_{\textsc{Edge}}(v_j)\neq\varnothing$, then select edge $\mathfrak{e}\in\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{P}_{\textsc{Edge}}(v_j)$. Suppose $\mathfrak{f}$ and $\mathfrak{f}'$ are the two incident faces of $\mathfrak{e}$. We have four subcases depending on whether $\mathfrak{e}$ is in $\mathcal{B}_{\textsc{Edge}}(v_i)$ and $\mathcal{B}_{\textsc{Edge}}(v_j)$.
  - If $\mathfrak{e}\in\mathcal{B}_{\textsc{Edge}}(v_i)$ and $\mathfrak{e}\in\mathcal{B}_{\textsc{Edge}}(v_j)$, then do nothing.
  - If $\mathfrak{e}\in\mathcal{B}_{\textsc{Edge}}(v_i)$ and $\mathfrak{e}\notin\mathcal{B}_{\textsc{Edge}}(v_j)$, suppose $\mathfrak{f}\in\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathfrak{f}'\in\mathcal{E}_{\textsc{Face}}(v_i)$. Select a triangle in $\mathfrak{f}'$ with one edge on $\mathfrak{e}$ and put it in X_j.
  - If $\mathfrak{e}\notin\mathcal{B}_{\textsc{Edge}}(v_i)$ and $\mathfrak{e}\in\mathcal{B}_{\textsc{Edge}}(v_j)$, suppose $\mathfrak{f}\in\mathcal{I}_{\textsc{Face}}(v_j)$ and $\mathfrak{f}'\in\mathcal{E}_{\textsc{Face}}(v_j)$. Select a triangle in $\mathfrak{f}'$ with one edge on $\mathfrak{e}$ and put it in X_i.
  - If $\mathfrak{e}\notin\mathcal{B}_{\textsc{Edge}}(v_i)$ and $\mathfrak{e}\notin\mathcal{B}_{\textsc{Edge}}(v_j)$, then select two triangles in $\mathfrak{f}$ and $\mathfrak{f}'$ respectively such that the triangles have a common edge on $\mathfrak{e}$, see Figure B.8(e).

- If $v_i\mathsf{TPP}v_j$ is in Γ, then by Table 8 we also have $\mathcal{P}_{\textsc{Face}}(v_i)\cap\mathcal{P}_{\textsc{Face}}(v_j)\neq\varnothing$, or $\mathcal{P}_{\textsc{Edge}}(v_i)\cap\mathcal{P}_{\textsc{Edge}}(v_j)\neq\varnothing$, or $S_i\cap S_j\neq\varnothing$. If $S_i\cap S_j\neq\varnothing$, then do

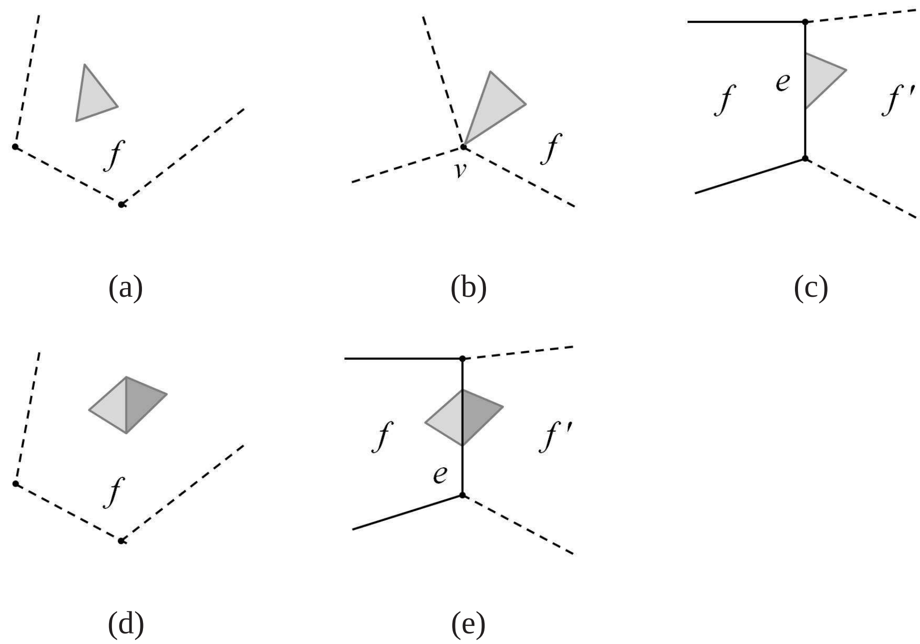
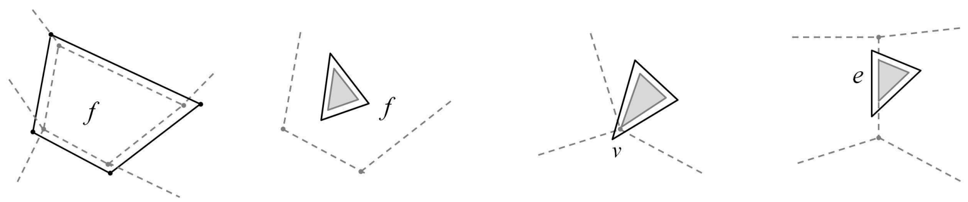


Figure B.8: Illustration of the selection of triangles

nothing. If $S_i \cap S_j = \varnothing$ and $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \varnothing$, then select a face $\mathfrak{f} \in \mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$ and one triangle in $\mathfrak{f}$. Put the triangle in both X_i and X_j. If $S_i \cap S_j = \varnothing$, $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) = \varnothing$, and $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j) \neq \varnothing$, then select an edge $\mathfrak{e} \in \mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j)$. Suppose $\mathfrak{f}$ and $\mathfrak{f}'$ are the two incident faces of $\mathfrak{e}$. At least one of $\mathfrak{f}$ and $\mathfrak{f}'$ is not in $\mathcal{E}_{\text{FACE}}(v_i)$ (otherwise $\mathfrak{e}$ is in $\mathcal{E}_{\text{EDGE}}(v_i)$). W.l.o.g., suppose $f \notin \mathcal{E}_{\text{FACE}}(v_i)$. If $\mathfrak{f} \in \mathcal{I}_{\text{FACE}}(v_i)$, then do nothing. If $\mathfrak{f} \in \mathcal{P}_{\text{FACE}}(v_i)$, we select a triangle in $\mathfrak{f}$ with one edge on $\mathfrak{e}$ and put it in X_i.

- If $v_i \mathsf{PO} v_j$ is in Γ, then by Table 8 we have $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$. There exists a face $\mathfrak{f}$ in $(\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{P}_{\text{FACE}}(v_i)) \cap (\mathcal{I}_{\text{FACE}}(v_j) \cup \mathcal{P}_{\text{FACE}}(v_j))$. If $\mathfrak{f}$ is in $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$, then select a triangle in face $\mathfrak{f}$ and put it in both X_i and X_j. Otherwise, we do nothing.

We assume that all the triangles are pairwise disjoint and are sufficiently small such that the union of all the triangles does not entirely occupy any face or any edge. Now X_i contains all the triangles we need for spatial variable v_i. For clarity, we now consider each face as its closure, and we use $(v_j \mathsf{PP} v_i) \in \Gamma$ to denote that



(a) $x \in$ FACE (b) x inside face $\mathfrak{f}$ (c) x on vertex $\mathfrak{v}$ (d) x on edge $\mathfrak{e}$

Figure B.9: Illustration of function $\mathsf{expand}(x, 1)$

$v_j \mathsf{TPP} v_i$ or $v_j \mathsf{NTPP} v_i$ is a constraint in Γ. Define a_i and b_i as follows:

$$a_i = \bigcup X_i, \tag{B.1}$$

$$b_i = a_i \cup \bigcup \mathcal{I}_{\text{FACE}}(v_i) \cup \bigcup \{a_j : (v_j \mathsf{PP} v_i) \in \Gamma\}. \tag{B.2}$$

We assert that $\{b_1, b_2 \ldots, b_n\}$ satisfies all the constraints in Γ except that some NTPP constraints may be realised as TPP. This assertion can be proven in the same way as in the proof of Lemma 8.

Let X be the union of all X_i, i.e. X is the set of all the triangles selected for spatial variables. To cope with the NTPP constraints, we introduce the expand function from $(X \cup \text{FACE}) \times \{1, 2, \ldots, n\}$ to regions in the plane such that for any $x, x' \in X \cup \text{FACE}$,

- $\mathsf{expand}(x, 1) = x$.
- $\mathsf{expand}(x, i)$ NTPP $\mathsf{expand}(x, i+1)$ for $i = 1, 2, \ldots, n-1$.
- $\mathsf{expand}(x, i)$ DC $\mathsf{expand}(x', i')$ if $x \mathsf{DC} x'$, for $i, i' = 1, 2, \ldots, n$.
- $\mathsf{expand}(x, i)$ PO $\mathsf{expand}(x', i')$ if $x \mathsf{EC} x'$, for $i, i' = 1, 2, \ldots, n$.

That is to say, $\mathsf{expand}(x, i)$ ($i = 1, 2, \ldots, n$) is a series of nested regions among which x is the innermost core. Meanwhile, $\mathsf{expand}(x, i)$ should be small enough to not touch or overlap any other regions or any other $\mathsf{expand}(x', i')$ whenever possible. Figure B.9 provides illustrations for $\mathsf{expand}(x, 1)$.

We can extend the domain of the function expand to include all b_i defined above and all landmarks by

$$\mathsf{expand}(y, i) = \bigcup \{\mathsf{expand}(x, i) : x \subseteq y, x \in X \cup \text{FACE}\}, \tag{B.3}$$

where $y \in \{b_1, \ldots, b_n, l_1, \ldots, l_m\}$ and $i = 1, 2, \ldots, n$.

Define a function $d_{\mathsf{NTPP}} : V \times (V \cup L) \to \mathbb{N}$, such that $d_{\mathsf{NTPP}}(v_i, w)$ is the length of the longest NTPPi chain from v_i to w, where w is either variable v_j or landmark l_j. Furthermore, define

$$\begin{aligned} c_i = b_i \cup \bigcup\{\mathsf{expand}(b_j, d_{\mathsf{NTPP}}(v_i, v_j)) : v_j \mathsf{NTPP} v_i\} \\ \cup \bigcup\{\mathsf{expand}(l_j, d_{\mathsf{NTPP}}(v_i, l_j)) : l_j \mathsf{NTPP} v_i\}. \end{aligned} \tag{B.4}$$

It can be proven that $\{c_1, \ldots, c_n\}$ is a solution of Γ such that $S_i = \partial c_i \cap \mathrm{V{\scriptsize TX}}$ for $i = 1, 2, \ldots, n$ in the same way as in [19, 22]. We omit the details here.

Appendix C. Proof sketch of Theorem 12

We need to adjust the construction given in the sufficiency part to cope with the strong connectedness. The only differences from the standard RCC8 interpretation are: (i) we assume $S_i = \varnothing$ for each variable v_i; (ii) although the requirements for $\mathsf{expand}(\cdot, \cdot)$ still apply, we need to modify the construction of this function to cater for the change in the interpretations of RCC8 relations. If x is a face in FACE, or a triangle in X on some vertex $\mathfrak{v}$, $\mathsf{expand}(x, 1)$ should be modified as shown in Figures C.10 (a) and (c) respectively, which can be contrasted with Figures B.9 (a) and (c). Note that in Figure C.10 (c), it holds that $x \mathsf{DC} \mathfrak{f}_1$ because their intersection is a point (not a curve). Therefore, $\mathsf{expand}(x, 1)$ is supposed to be disjoint with $\mathfrak{f}_1$ (under the strong connectedness interpretation of RCC8) due to the requirement of $\mathsf{expand}(\cdot, \cdot)$. The case in Figure C.10 (a) is similar: the boundary of the expanded face does not intersect with any face which is disjoint with the original face.

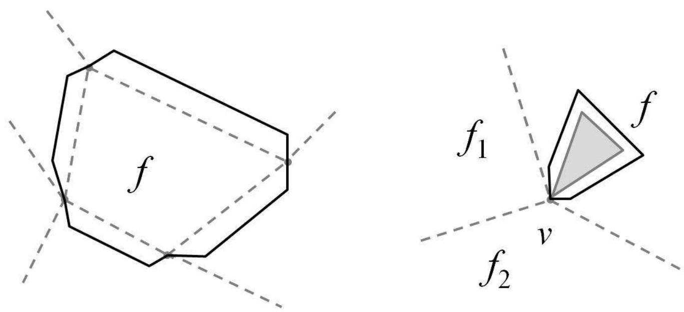
All the remaining parts of the construction, including the selection of triangles (note that $S_i = \varnothing$ here), definitions of a_i, b_i, and verification of b_i as a solution of Γ, are completely the same as in the standard interpretation of RCC8 relations.

References

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[2] Manuel Bodirsky. Complexity classification in infinite-domain constraint satisfaction. *CoRR*, abs/1201.0856, 2012.

[3] Stefano Borgo, Nicola Guarino, and Claudio Masolo. A pointless theory of space based on strong connection and congruence. In Luigia Carlucci Aiello,



(a) $x \in$ FACE (c) x on vertex $\mathfrak{v}$

Figure C.10: Illustration of function $\mathsf{expand}(x, 1)$ in the RCC model based on strong connectedness

Jon Doyle, and Stuart C. Shapiro, editors, *KR'96*, pages 220–229. Morgan Kaufmann, 1996.

[4] Mathias Broxvall and Peter Jonsson. Point algebras for temporal reasoning: Algorithms and complexity. *Artif. Intell.*, 149(2):179–220, 2003.

[5] Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Logic*, 12(4):24:1–24:66, July 2011.

[6] Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

[7] Anthony G. Cohn and Jochen Renz. Qualitative spatial reasoning. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, pages 551–596. Elsevier, 2008.

[8] Anthony G. Cohn and Achille C. Varzi. Modes of connection. In Christian Freksa and David M. Mark, editors, *COSIT*, volume 1661 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 1999.

[9] Mark de Berg, Otfried Ceong, Marc van Kreveld, and Mark Overmars. *Computational Geometry : Algorithms and Applications*. 3rd edition, Springer, 2008.

[10] Reinhard Diestel. *Graph Theory, Third Edition*. Springer, 2006.

[11] Matt Duckham, Stephan Winter, and Michelle Robinson. Including landmarks in routing instructions. *J. Location Based Services*, 4(1):28–52, 2010.

[12] Andrew U. Frank. Qualitative spatial reasoning with cardinal directions. In *ÖGAI*, pages 157–167, 1991.

[13] Alfonso Gerevini and Jochen Renz. Combining topological and size information for spatial reasoning. *Artif. Intell.*, 137(1-2):1–42, 2002.

[14] Floraine Grabler, Maneesh Agrawala, Robert W. Sumner, and Mark Pauly. Automatic generation of tourist maps. *ACM Trans. Graph.*, 27(3):100:1–100:11, 2008.

[15] Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In Alfonso Gerevini, Adele E. Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *ICAPS*. AAAI, 2009.

[16] Robin Hirsch. A finite relation algebra with undecidable network satisfaction problem. *Logic Journal of the IGPL*, 7(4):547–554, 1999.

[17] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.

[18] Jason Jingshi Li, Jinbo Huang, and Jochen Renz. A divide-and-conquer approach for solving interval algebra networks. In Craig Boutilier, editor, *IJCAI*, pages 572–577, 2009.

[19] Sanjiang Li. On topological consistency and realization. *Constraints*, 11(1):31–51, 2006.

[20] Sanjiang Li. Combining topological and directional information for spatial reasoning. In Manuela M. Veloso, editor, *IJCAI*, pages 435–440, 2007.

[21] Sanjiang Li and Anthony G. Cohn. Reasoning with topological and directional spatial information. *Computational Intelligence*, 28(4):579–616, 2012.

[22] Sanjiang Li and Huaiqing Wang. RCC8 binary constraint network can be consistently extended. *Artif. Intell.*, 170(1):1–18, 2006.

[23] Gérard Ligozat. Qualitative triangulation for spatial reasoning. In *Spatial Information Theory: A Theoretical Basis for GIS (COSIT'93)*, pages 54–68, 1993.

[24] Gérard Ligozat. Reasoning about cardinal directions. *J. Vis. Lang. Comput.*, 9(1):23–44, 1998.

[25] Gérard Ligozat and Jochen Renz. What is a qualitative calculus? A general framework. In Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors, *PRICAI*, pages 53–64. Springer, 2004.

[26] Weiming Liu and Sanjiang Li. Reasoning about cardinal directions between extended objects: The NP-hardness result. *Artif. Intell.*, 175(18):2155–2169, 2011.

[27] Weiming Liu and Sanjiang Li. Here, there, but not everywhere: An extended framework for qualitative constraint satisfaction. In Luc De Raedt et al., editors, *ECAI*, pages 552–557, 2012.

[28] Weiming Liu, Sanjiang Li, and Jochen Renz. Combining RCC-8 with qualitative direction calculi: Algorithms and complexity. In Craig Boutilier, editor, *IJCAI*, pages 854–859, 2009.

[29] Weiming Liu, Shengsheng Wang, Sanjiang Li, and Dayou Liu. Solving qualitative constraints involving landmarks. In Jimmy Ho-Man Lee, editor, *CP*, pages 523–537, 2011.

[30] Weiming Liu, Xiaotong Zhang, Sanjiang Li, and Mingsheng Ying. Reasoning about cardinal directions between extended objects. *Artif. Intell.*, 174(12-13):951–983, 2010.

[31] Kevin Lynch. *The Image of the City*. MIT, 1960.

[32] Alan K. Mackworth. Constraint satisfaction, in S.C. Shapiro editor, *Encyclopedia of AI*, Vol. 1, 2nd edition, Wiley, 1992, pp. 285–293.

[33] Ugo Montanari. Networks of constraints: Fundamental properties and application to picture processing, *Information Science* **7**:95–132, 1974.

[34] Bernhard Nebel. Computational properties of qualitative spatial reasoning: First results. In Ipke Wachsmuth, Claus-Rainer Rollinger, and Wilfried Brauer, editors, *KI*, pages 233–244. Springer, 1995.

[35] Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *J. ACM*, 42(1):43–66, 1995.

[36] Dimitris Papadias and Timos Sellis. Qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal*, 3(4):479–516, 1994.

[37] Duc Nghia Pham, John Thornton, and Abdul Sattar. Modelling and solving temporal reasoning as propositional satisfiability. *Artif. Intell.*, 172(15):1752–1782, 2008.

[38] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *KR'92*, pages 165–176, 1992.

[39] Jochen Renz. A canonical model of the region connection calculus. *Journal of Applied Non-Classical Logics*, 12(3-4):469–494, 2002.

[40] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1-2):69–123, 1999.

[41] Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In Marco Aiello, Ian Pratt-Hartmann, and Johan van Benthem, editors, *Handbook of Spatial Logics*, pages 161–215. Springer, 2007.

[42] Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *J. Artif. Intell. Res. (JAIR)*, 39:127–177, 2010.

[43] Peter van Beek. Reasoning about qualitative temporal information. *Artif. Intell.*, 58(1-3):297–326, 1992.

[44] Marc B. Vilain and Henry A. Kautz. Constraint propagation algorithms for temporal reasoning. In *AAAI*, pages 377–382, 1986.

[45] Stefan Wölfl and Matthias Westphal. On combinations of binary qualitative constraint calculi. In Craig Boutilier, editor, *IJCAI*, pages 967–973, 2009.

[46] Diedrich Wolter and J. H. Lee. Qualitative reasoning with directional relations. *Artif. Intell.*, 174(18):1498–1507, 2010.