ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-507

The Complexity of Reasoning with Relative Directions

Jae Hee Lee[1]

Abstract. Whether reasoning with relative directions can be performed in NP has been an open problem in qualitative spatial reasoning. Efficient reasoning with relative directions is essential, for example, in rule-compliant agent navigation. In this paper, we prove that reasoning with relative directions is $\exists\mathbb{R}$-complete. As a consequence, reasoning with relative directions is not in NP, unless NP $= \exists\mathbb{R}$.

1 INTRODUCTION

Qualitative spatial reasoning (QSR) [6, 15] enables cognitive agents to reason about space using abstract symbols. Among several aspects of space (e.g., topology, direction, distance) relative direction information is useful for agents navigating in space. Observers typically describe their environment by specifying the relative directions in which they see other objects or other people from their point of view. As such, efficient reasoning with relative directions, i.e., determining whether a given statement involving relative directions is true, can be advantageously used for applications that handle rules or requirements involving relative directions. For example, efficient reasoning with relative directions can help a bridge crew determine whether other vessels comply with the navigation regulations which can be formalized as logical statements involving relative directions [11].

Different representations have been proposed for relative directions, including $\mathcal{DCC}$ [8, 23], $\mathcal{DRA}$ [18, 7], $\mathcal{LR}$ [16, 24] and $\mathcal{OPRA}_m$ [19] (cf. Subsection 2.2).

The predominant reasoning method in the early development of QSR was the path-consistency method based on the composition operation of relations, which is a polynomial-time method originally developed for finite domain constraint satisfaction problems. Soon its underlying composition operation was superseded by the weak-composition, as many spatial constraint languages turned out to be not closed under composition, and the path-consistency method was modified to the algebraic closure (a-closure) method [21]. Though the a-closure method gives rise to an NP decision procedure for some known spatial constraint languages, it turned out that the a-closure method is not sufficient for reasoning with $\mathcal{DRA}$, $\mathcal{LR}$ and $\mathcal{OPRA}_m$ (cf. [17], [9]). Indeed, reasoning with $\mathcal{DCC}$, $\mathcal{DRA}$ and $\mathcal{LR}$ is NP-hard (cf. [17], [23], [18]) and the NP-membership has not been proven so far.

In this paper, we prove in a holistic manner that reasoning with all mentioned relative directional constraint languages, i.e., $\mathcal{LR}$, $\mathcal{DCC}$, $\mathcal{DRA}$ and $\mathcal{OPRA}_m$, is $\exists\mathbb{R}$-complete, where $\exists\mathbb{R}$ is a complexity class residing between NP and PSPACE (cf. Subsection 3.1). As a consequence, all mentioned relative directional constraint languages are equivalent to each other in terms of computational complexity, and reasoning with them cannot achieved in NP, unless NP $= \exists\mathbb{R}$. Furthermore, no NP decision procedures exist for atomic formulas of relative directional constraint languages, unless NP $= \exists\mathbb{R}$.

[1] University of Bremen, Germany, email: jay@informatik.uni-bremen.de

In [26] the authors prove the NP-hardness of reasoning with relative directions by reducing the NP-hard *realizability problem for uniform oriented matroid* (RUOM) to reasoning with relative directions. However, RUOM has not been shown to be $\exists\mathbb{R}$-hard, and therefore, one could not draw strong consequences as this paper achieves.

This work is a shortend version of [12, Chapters 2 and 3].

2 RELATIVE DIRECTIONAL CONSTRAINT LANGUAGES

2.1 Spatial Constraint Language

In what follows we will define the syntax and the semantics of a spatial constraint language (also known as qualitative calculus [6, 15]) with respect to binary spatial relations. The definition, however, extends naturally to ternary and to n-ary relations.

A spatial constraint language $\mathcal{L}$ is a quadruple $\langle D, \mathcal{R}, \iota, \mathcal{V}\rangle$, where D is the domain of spatial entities which is not empty, $\mathcal{R}$ a finite collection of relation symbols, ι the intended interpretation that maps each relation symbol $R \in \mathcal{R}$ to a relation $R^\iota \subseteq D \times D$, and $\mathcal{V}$ a countably infinite set of variables $v_1, v_2, \ldots$.

A formula of $\mathcal{L}$, or an $\mathcal{L}$-formula is defined inductively as follows:

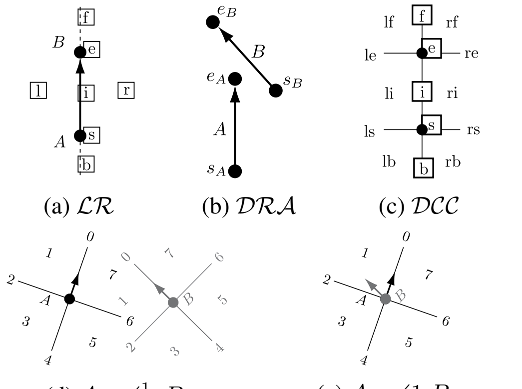
$$\varphi := \top \mid \bot \mid v_i \, R \, v_j \mid v_i \, \{R_1, \ldots, R_k\} \, v_j \mid \varphi \wedge \psi$$

where $v_i, v_j \in \mathcal{V}$, $R, R_1, \ldots, R_k \in \mathcal{R}$, $k \geq 1$ and φ and ψ are formulas.

A model of $\mathcal{L}$, which fixes the truth of $\mathcal{L}$-formulas with respect to its intended interpretation, is given by a valuation function $\nu : \mathcal{V} \to D$, which assigns to each variable v_i a value v_i^ν from the domain. The semantics of formulas are defined inductively with respect to the syntactical structure (we write $\nu \models \varphi$ to denote that valuation ν satisfies formula φ):

$\nu \models \top$		always
$\nu \models \bot$		never
$\nu \models v_i \, R \, v_j$	iff	$(v_i^\nu, v_j^\nu) \in R^\iota$
$\nu \models v_i \, \{R_1, \ldots, R_k\} \, v_j$	iff	ν satisfies some $v_i \, R_\ell \, v_j$, $1 \leq \ell \leq k$
$\nu \models \varphi \wedge \psi$	iff	$\nu \models \varphi$ and $\nu \models \psi$

An $\mathcal{L}$-formula φ is said to be satisfiable, if there is a valuation ν with $\nu \models \varphi$. The problem of deciding whether an $\mathcal{L}$-formula is satisfiable is also called the *constraint satisfaction problem* for $\mathcal{L}$, or CSP($\mathcal{L}$) for short. An $\mathcal{L}$-formula of the form $\bigwedge_{ij} v_i \, R_{ij} \, v_j$ is called *atomic*. If there is a polynomial-time decision procedure for atomic $\mathcal{L}$-formulas, then CSP($\mathcal{L}$) can be solved in NP by means of a backtracking search.



(a) $\mathcal{LR}$ (b) $\mathcal{DRA}$ (c) $\mathcal{DCC}$

(d) $A\ {}_m\angle_7^1\ B$ (e) $A\ {}_m\angle 1\ B$

Figure 1: $\mathcal{LR}$, $\mathcal{DRA}$, $\mathcal{DCC}$ and $\mathcal{OPRA}_m$ relations.

2.2 Relative Directional Constraint Languages

A *relative directional constraint language* is a spatial constraint language whose relation symbols stand for relative directions. In this subsection, we present four different relative directional constraint languages.

The most elementary language for relative directional relations is called $\mathcal{LR}$ [16, 24] (see Figure 1a). The relations of $\mathcal{LR}$ are defined based on a reference system generated by a directed line connecting two points. The position of a third point is then categorized as to be either left or right of the line (*l*, *r*), or on 5 different segments of the reference line (*f*, *e*, *i*, *s*, *b*). Two additional relations *dou* and *tri* describe degenerate cases where the first two points coincide; *dou* holds if the third point does not coincide with them, and *tri* holds if all three points coincide.

The spatial constraint language $\mathcal{DRA}$ [18, 7] has as its basic entities dipoles. A dipole A is an oriented line segment which is given by a start point s_A and an end point e_A (cp. Figure 1b). A $\mathcal{DRA}$ relation[2] between two dipoles is a quadruple of $\mathcal{LR}$ relations that hold between a dipole and the start and the end point of the other. In Figure 1b the start point s_B and the end point e_B of dipole B is respectively to the right and to the left of dipole A resulting in $\mathcal{LR}$ relations *r* and *l*, respectively. In the same way, we obtain *r* and *l* for the two $\mathcal{LR}$ relations that hold between s_A and B, and between e_A and B, respectively. A $\mathcal{DRA}$ relation records this information as *rlll* in the order the $\mathcal{LR}$ relations are presented.

The double cross calculus $\mathcal{DCC}$ [8, 23] can be regarded as a refinement of the $\mathcal{LR}$. In $\mathcal{DCC}$ the left and right plane of the reference line are further refined by two orthogonal lines passing through the reference points, which is meaningful from a cognitive point of view [8]. The refined relations are illustrated in Figure 1c.

$\mathcal{OPRA}_m$ [19] is based on the domain $\mathbb{R}^2 \times [0, 2\pi)$ of oriented points. Half-lines and angular sectors are instantiated to describe the position of one oriented point as seen from another. The relations of $\mathcal{OPRA}_m$ are defined with respect to a granularity parameter m that determines how many sectors are used ($\mathcal{OPRA}_m$ uses m lines to divide the full circle evenly, giving $2m$ angular sectors and $2m$ half-lines). Figure 1d presents an example of an $\mathcal{OPRA}_2$ relation ${}_m\angle_7^1$. In the example, B is located in sector 7 as seen from A, which, in turn, is located in sector 1 as seen from B. Symbol ${}_m\angle_7^1$ is used to denote this relation. In the degenerate case, where points A and B coincide, the sector i of A to which point B is oriented determines the relation and this is denoted by ${}_m\angle i$ (cp. Figure 1e).

[2] In this paper by $\mathcal{DRA}$ we refer to the refined version $\mathcal{DRA}_f$ in [7].

3 COMPUTATIONAL COMPLEXITY

In this section we prove the $\exists\mathbb{R}$-completeness of reasoning with relative directional constraint languages. After introducing oriented matroids and its realizability problem (ROM) which is $\exists\mathbb{R}$-complete, we reduce ROM to each of the relative directional constraint languages introduced in this paper, i.e., $\mathcal{LR}$, $\mathcal{DCC}$, $\mathcal{DRA}$ and $\mathcal{OPRA}_m$.

3.1 The Complexity Class $\exists\mathbb{R}$

The complexity class $\exists\mathbb{R}$ was first introduced in [22] to capture several well known problems which are equivalent to the existential theory of the reals.

Definition 1. *The existential theory of the reals* is the set of true sentences of the form $\exists x_1 \ldots \exists x_n\ \phi(x_1, \ldots, x_n)$, where $\phi(x)$ is a quantifier-free Boolean formula over polynomial equations or inequalities (i.e., $f(x_1, \ldots, x_n) < 0$, $g(x_1, \ldots, x_n) \leq 0$ or $h(x_1, \ldots, x_n) = 0$, f, g, h being polynomials). Here, the polynomials have rational coefficients and each variable x_i ranges over $\mathbb{R}$.

The decision problem for the existential theory of the reals (ETR) is the problem of deciding if a given sentence in the existential theory of the reals is true.

Definition 2 (The complexity class $\exists\mathbb{R}$)**.** The complexity class $\exists\mathbb{R}$ is the class of all problems that are polynomial-time reducible to ETR. A computational problem is said to be $\exists\mathbb{R}$-*hard*, if every problem in $\exists\mathbb{R}$ can be reduced to it by a polynomial-time reduction. A computational problem is said to be $\exists\mathbb{R}$-*complete*, if it is $\exists\mathbb{R}$-hard and belongs to $\exists\mathbb{R}$.

Many computational problems are identified as $\exists\mathbb{R}$-complete (e.g., stretchability of simple pseudoline arrangement, the algorithmic Steinitz problem, intersection graphs of line segments, topological inference with convexity). For more details we refer to [22].

$\exists\mathbb{R}$-complete problems are hard to solve as the following theorem states.

Theorem 3. NP $\subseteq \exists\mathbb{R} \subseteq$ PSPACE

Proof. The first inclusion NP $\subseteq \exists\mathbb{R}$ is easy to show, see for example [3]. However, the other inclusion $\exists\mathbb{R} \subseteq$ PSPACE requires advanced knowledge in real algebraic geometry and is proved in [5]. ☐

Whether NP $\supseteq \exists\mathbb{R}$ or $\exists\mathbb{R} \supseteq$ PSPACE could not be shown so far and is an open problem.

3.2 Oriented Matroids

Oriented matroids [4] can be considered as combinatorial generalizations of spatial arrangements. They provide a broad model to describe information about relative positions geometrically (Definition 4) and purely combinatorially (Definition 6). Oriented matroids can be axiomatized in several ways. From the different axiomatizations of oriented matroids, we will choose the axiomatization using the notion of *chirotopes*, which captures the aspect of relative directions. Furthermore, we will restrict ourself to chirotopes with respect to the 3-dimensional vector space. Therefore the oriented matroids dealt hereafter are of rank 3, if the rank is not mentioned explicitly.

The following definition introduces oriented matroids as a mathematical object extracted from a vector configuration. Note that a *vector configuration* in $\mathbb{R}^3$ is a finite sequence of vectors in $\mathbb{R}^3$ that span $\mathbb{R}^3$.

Definition 4 (Oriented matroid of a vector configuration)**.** Let $V = (v_1, \ldots, v_n)$ be a finite vector configuration in $\mathbb{R}^3$, sgn : $\mathbb{R} \to \{-1, 0, 1\}$ a function that returns the sign of its argument, and $\det(v_{i_1}, v_{i_2}, v_{i_3})$ the determinant of a 3×3 matrix having $v_{i_1}, v_{i_2}, v_{i_3}$ as its column vectors. The *oriented matroid* of V is given by the map $\chi_V : \{1, 2, \ldots, n\}^3 \to \{-1, 0, 1\}, (i_1, i_2, i_3) \mapsto \operatorname{sgn}(\det(v_{i_1}, v_{i_2}, v_{i_3}))$ which is called the *chirotope* of V. The map χ_V records for each vector triple the information about whether it consists of linearly dependent vectors, a positively oriented basis of $\mathbb{R}^3$, or a negatively oriented basis of $\mathbb{R}^3$ (0, 1, -1, respectively).

Example 5. The oriented matroid of $V = (v_1, v_2, v_3)$ with $v_1 = (1, 0, 0)^T$, $v_2 = (0, 1, 0)^T$, $v_3 = (0, 0, 1)^T$ is the map $\chi_V : \{1, 2, 3\}^3 \to \{-1, 0, 1\}$ with $\chi_V(1, 2, 3) = \chi_V(2, 3, 1) = \chi_V(3, 1, 2) = 1$ and $\chi_V(2, 1, 3) = \chi_V(1, 3, 2) = \chi_V(3, 2, 1) = -1$. All other triples from $\{1, 2, 3\}^3$ represent linearly dependent vector triples, and thus map to 0.

The preceding definition of oriented matroid has an underlying vector configuration. By contrast, we axiomatize in the following oriented matroids as purely combinatorial objects decoupled from a vector configuration.

Definition 6 (Oriented matroid)**.** An *oriented matroid* on $E = \{1, 2, \ldots, n\}$ with $n \geq 3$ is a map given by $\chi : E^3 \longrightarrow \{-1, 0, 1\}$, called a *chirotope*, which satisfies the following three axioms:

(C1) χ is not identically zero.
(C2) χ is alternating, i.e., $\chi(i_{\sigma(1)}, i_{\sigma(2)}, i_{\sigma(3)}) = \operatorname{sign}(\sigma)\chi(i_1, i_2, i_3)$ for all $i_1, i_2, i_3 \in E$ and every permutation σ on $\{1, 2, 3\}$, where sign(σ) stands for the signature of a permutation σ.
(C3) For all $i_1, i_2, i_3, j_1, j_2, j_3 \in E$ such that $\chi(j_1, i_2, i_3) \cdot \chi(i_1, j_2, j_3) \geq 0$, $\chi(j_2, i_2, i_3) \cdot \chi(j_1, i_1, j_3) \geq 0$, $\chi(j_3, i_2, i_3) \cdot \chi(j_1, j_2, i_1) \geq 0$ we have $\chi(i_1, i_2, i_3) \cdot \chi(j_1, j_2, j_3) \geq 0$.

We note that axiom (C2) implies $\chi(i_1, i_2, i_3) = 0$ if two of three arguments coincide. We also note that an oriented matroid χ_V of a vector configuration V as defined in Definition 4 is an oriented matroid on E, where E is the index set of V.

Example 7. The map $\chi : \{1, 2, 3\}^3 \to \{-1, 0, 1\}$ defined by $\chi(1, 2, 3) = \chi(2, 3, 1) = \chi(3, 1, 2) = 1$ and $\chi(2, 1, 3) = \chi(1, 3, 2) = \chi(3, 2, 1) = -1$, where all other triples from $\{1, 2, 3\}^3$ are mapped to 0, satisfies all three axioms in Definition 6.

Now that there is the definition of oriented matroid that is of a purely combinatorial nature, one can ask the following question:

> Given an oriented matroid χ on $E = \{1, \ldots, n\}$, is there a vector configuration $V = (v_1, \ldots, v_n)$ whose vectors span $\mathbb{R}^3$, such that V is a *realization* of χ, in other words, χ_V is equal to χ?

To exemplify this question, consider the oriented matroid from Example 7. We observe that the triple (v_1, v_2, v_3) of vectors $v_1 = (1, 0, 0)^T, v_2 = (0, 1, 0)^T, v_3 = (0, 0, 1)^T$ is a realization of χ, since $\chi(i, j, k) = \operatorname{sgn}(\det(v_i, v_j, v_k)) = \chi_V(i, j, k)$ for all $i, j, k \in \{1, 2, 3\}$. The aforementioned problem is the so-called *realizability problem for oriented matroids* (ROM) and is equivalent to the pseudoline stretchability problem which is $\exists\mathbb{R}$-complete [25, 22].

3.3 $\exists\mathbb{R}$-Completeness of Reasoning with Relative Directions

Now we establish a connection between oriented matroids and relative direction relations. This allows us to reduce the $\exists\mathbb{R}$-complete problem ROM to reasoning with relative directional constraint languages. As vector configurations are closely related to oriented matroids, we will first establish a connection between vector configurations in $\mathbb{R}^3$ and point configurations in the plane. Then we will apply the same concept used for the connection between vector configurations and point configurations to the connection between oriented matroids and relative direction relations. The following example illustrates the connection between a vector configuration and *left*/*right* relations for points in the plane.

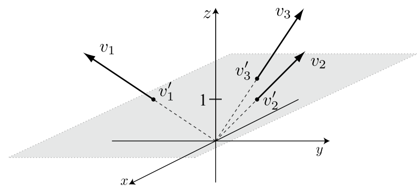


Figure 3: The connection between a vector configuration and a point configuration in the plane.

Example 8. Consider the projection $f : (x, y, z) \mapsto 1/z(x, y, z)$ shown in Figure 3, which identifies vectors v_1, v_2, v_3 with vectors v_1', v_2', v_3' in the plane $\{ (x, y, z) \in \mathbb{R}^3 \mid z = 1 \}$. Since vectors v_1, v_2, v_3 form a positively oriented basis of $\mathbb{R}^3$ (i.e., $\det(v_1, v_2, v_3) = 1$), v_3' is to the *left* of the directed line from v_1' to v_2'.

In Example 8, establishing the connection between a vector configuration in $\mathbb{R}^3$ and *left*/*right* relations for points in a plane was possible, due to the fact that all vectors are on one side of the XY-plane. Acyclic vector configurations assume this very property of vectors:

Definition 9. A vector configuration $V = (v_1, \ldots, v_n)$ in $\mathbb{R}^3$ is said to be *acyclic*, if the vectors from V are entirely contained in an open half-space induced by a plane, i.e., there is a linear map $f : \mathbb{R}^3 \to \mathbb{R}$, such that $f(v_i) > 0$ for all $i = 1, \ldots, n$.

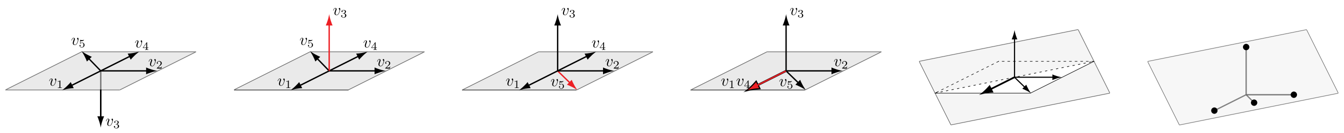
Given an acyclic vector configuration we can project the vectors v_i, $i = 1, \ldots, n$ to points in an affine plane $\mathbb{A}^2$ defined by $\mathbb{A}^2 := \{ x \in \mathbb{R}^3 \mid f(x) = 1 \}$, where we associate each vector v_i, $i = 1, \ldots, n$ with the point $1/f(v_i) \cdot v_i \in \mathbb{A}^2$.

Theorem 11 characterizes a necessary condition for a vector configuration to be acyclic, which is useful for enforcing acyclicity of a vector configuration. Hereafter, we will regard V both as a vector configuration and as a set that consists of the vectors in the vector configuration. Furthermore, v^* and v^{**} will denote two linearly independent vectors from V, and $V_1^+, V_1^-, V_2^+, V_2^-, V_3^+, V_3^-$ are sets defined as

$$
\begin{aligned}
V_1^+ &:= \{ v \in V \mid v = tv^* \text{ for a } t \in \mathbb{R}, t > 0 \} \\
V_1^- &:= \{ v \in V \mid v = tv^* \text{ for a } t \in \mathbb{R}, t < 0 \} \\
V_2^+ &:= \{ v \in V \mid v = t_1 v^* + t_2 v^{**} \text{ for } t_1, t_2 \in \mathbb{R}, t_2 > 0 \} \\
V_2^- &:= \{ v \in V \mid v = t_1 v^* + t_2 v^{**} \text{ for } t_1, t_2 \in \mathbb{R}, t_2 < 0 \} \\
V_3^+ &:= \{ v \in V \mid \det(v^*, v^{**}, v) > 0 \} \\
V_3^- &:= \{ v \in V \mid \det(v^*, v^{**}, v) < 0 \} .
\end{aligned}
$$

Lemma 10. *$V_1^+, V_1^-, V_2^+, V_2^-, V_3^+, V_3^-$ are pairwise disjoint, and jointly exhaustive, i.e., $V = V_1^+ \dot\cup V_1^- \dot\cup V_2^+ \dot\cup V_2^- \dot\cup V_3^+ \dot\cup V_3^-$.*

Proof. By definition V_i^+ and V_i^- are disjoint for $i = 1, 2, 3$. Then given a vector $v \in V$, it is from one of the pairwise disjoint



(a) The vector configuration is cyclic. (b) By switching the sign of v_3, we move v_3 from V_3^- to V_3^+. (c) By switching the sign of v_5, we move v_5 from V_2^- to V_2^+. (d) By switching the sign of v_4, we move v_4 from V_1^- to V_1^+. (e) The vector configuration is acyclic due to Theorem 11 (f) The corresponding point configuration in the affine plane $\mathbb{A}^2$.

Figure 2: Enforcing acyclicity of a vector configuration. Initially $V_1^- = \{v_4\}$, $V_2^- = \{v_5\}$, $V_3^- = \{v_3\}$, where $v^* := v_1$ and $v^{**} := v_2$.

sets $V \cap \text{span}(v^*)$ $(= V_1^+ \dot\cup V_1^-)$, $V \cap \text{span}(v^*, v^{**}) \setminus \text{span}(v^*)$ $(= V_2^+ \dot\cup V_2^-)$, or $V \cap \mathbb{R}^3 \setminus \text{span}(v^*, v^{**})$ $(= V_3^+ \dot\cup V_3^-)$ $\square$

Theorem 11. *V is acyclic, if $V = V_1^+ \cup V_2^+ \cup V_3^+$.*

Proof. Let $v^{***} \in V_3^+$. Note that V_3^+ is not empty, because V spans $\mathbb{R}^3$. Let $v^* \times v^{**}$ be the vector product of v^* and v^{**}, thus $(v^* \times v^{**})^T v = \det(v^*, v^{**}, v)$. We define a linear map $f : \mathbb{R}^3 \to \mathbb{R}$ with

$$f(v) = (v^* + \alpha(v^* \times v^{***}) + \beta(v^* \times v^{**}))^T\, v,$$

where α and β are real numbers with the properties $v^{*T} v + \alpha(v^* \times v^{***})^T v > 0$ for all $v \in V_2^+$ and $v^{*T} v + \alpha(v^* \times v^{***})^T v + \beta(v^* \times v^{**})^T v > 0$ for all $v \in V_3^+$. Such α and β exist, because $(v^* \times v^{***})^T v = \det(v^*, v^{***}, v) < 0$ for all $v \in V_2^+$ and $(v^* \times v^{**})^T v = \det(v^*, v^{**}, v) > 0$ for all $v \in V_3^+$.

Then, for all $v \in V_1^+ : f(v) = v^{*T} v > 0$, and for all $v \in V_2^+ : f(v) = v^{*T} v + \alpha(v^* \times v^{***})^T v > 0$ and for all $v \in V_3^+ : f(v) = v^{*T} v + \alpha(v^* \times v^{***})^T v + \beta(v^* \times v^{**})^T v > 0$. Thus $f(v) > 0$ for all $v \in V_1^+ \cup V_2^+ \cup V_3^+$. $\square$

Based on Theorem 11 we can devise a procedure for enforcing acyclicity of a vector configuration exclusively by changing the signs of vectors. An example is illustrated in Figure 2.

Input: A vector configuration $V = (v_1, \ldots, v_n)$.

Output: An acyclic vector configuration obtained from V by switching the signs of vectors from V

1 **begin**
2 $\quad v^* \leftarrow 0$, $v^{**} \leftarrow 0$, $v^{***} \leftarrow 0$
3 $\quad$ Choose $i, j, k \in \{1, \ldots, n\}$ such that $\det(v_i, v_j, v_k) \neq 0$ and set $v^* \leftarrow v_i$ and $v^{**} \leftarrow v_j$
4 $\quad$ **foreach** $i \in \{1, \ldots, n\}$ **do**
5 $\qquad$ **if** $\det(v^*, v^{**}, v_i) < 0$ **then** $v_i \leftarrow -v_i$
6 $\quad$ **foreach** $i \in \{1, \ldots, n\}$ **do**
7 $\qquad$ **if** $\det(v^*, v^{**}, v_i) > 0$ **then** $v^{***} \leftarrow v_i$
8 $\quad$ **foreach** $i \in \{1, \ldots, n\}$ **do**
9 $\qquad$ **if** $\det(v^*, v^{**}, v_i) = 0$ *and* $\det(v^*, v_i, v^{***}) < 0$ **then** $v_i \leftarrow -v_i$
10 $\quad$ **foreach** $i \in \{1, \ldots, n\}$ **do**
11 $\qquad$ **if** $\det(v^*, v^{**}, v_i) = 0$ *and* $\det(v^*, v_i, v^{***}) = 0$ *and* $\det(v_i, v^{**}, v^{***}) < 0$ **then** $v_i \leftarrow -v_i$
12 $\quad$ **return** V

Function EnforceAcycVC(V)

Function EnforceAcycVC implements an $O(n^3)$ algorithm for enforcing acyclicity of a vector configuration based on the idea presented in Figure 2. EnforceAcycVC moves all vectors in sets V_1^-, V_2^-, V_3^- to sets V_1^+, V_2^+, V_3^+ such that the resulting vector configuration is acyclic according to Theorem 11. This is done exclusively by changing the signs of vectors to allow for applying the underlying concept to oriented matroid setting (cf. Function EnforceAcycOM). In the following we will prove the correctness of EnforceAcycVC.

The following lemmas show that Function EnforceAcycVC detects vectors in V_1^-, V_2^-, and V_3^- by testing the signs of determinant expressions.

Lemma 12. *Let $v \in V$. Then $v \in V_3^-$, if and only if* $\det(v^*, v^{**}, v) < 0$.

Proof. The proof follows immediately from the definition of V_3^-. $\square$

For the next two lemmas we note that if V_3^- is empty, then V_3^+ is not empty, otherwise V would not span $\mathbb{R}^3$.

Lemma 13. *Let V_3^- be empty and $v^{***} \in V_3^+$. Let $v \in V$. Then $v \in V_2^-$, if and only if* $\det(v^*, v^{**}, v) = 0$ *and* $\det(v^*, v, v^{***}) < 0$.

Proof. If $v \in V_2^-$, then there are $t_1, t_2 \in \mathbb{R}$, $t_2 < 0$, such that $v = t_1 v^* + t_2 v^{**}$. Thus $\det(v^*, v^{**}, v) = t_1 \det(v^*, v^{**}, v^*) + t_2 \det(v^*, v^{**}, v^{**}) = 0$. Furthermore, $\det(v^*, v, v^{***}) = t_1 \det(v^*, v^*, v^{***}) + t_2 \det(v^*, v^{**}, v^{***}) = t_2 \det(v^*, v^{**}, v^{***}) < 0$.

For the other direction of the proof, we note that $\det(v^*, v^{**}, v) = 0$ is a necessary condition for v to be in V_2^-, since it would otherwise be in V_3^+. Now assume that $\det(v^*, v^{**}, v) = 0$ and $\det(v^*, v, v^{***}) < 0$ and $v \notin V_2^-$. Then $v \in V_1^+ \cup V_1^- \cup V_2^+$, i.e., there are $t_1, t_2 \in \mathbb{R}$, $(t_1, t_2) \neq (0, 0)$ and $t_2 \geq 0$, such that $v = t_1 v^* + t_2 v^{**}$. Then

$$\begin{aligned} 0 > \det(v^*, v, v^{***}) &= t_1 \det(v^*, v^*, v^{***}) + t_2 \det(v^*, v^{**}, v^{***}) \\ &= t_2 \det(v^*, v^{**}, v^{***}) \end{aligned}$$

Since $t_2 \geq 0$ and $\det(v^*, v^{**}, v^{***}) > 0$, the inequality is a contradiction. $\square$

Lemma 14. *Let V_3^- be empty and $v^{***} \in V_3^+$. Let $v \in V$. Then $v \in V_1^-$, if and only if* $\det(v^*, v^{**}, v) = 0$, $\det(v^*, v, v^{***}) = 0$ *and* $\det(v, v^{**}, v^{***}) < 0$.

Proof. The one direction is straight forward. Now we assume that $\det(v^*, v^{**}, v) = 0$, $\det(v^*, v, v^{***}) = 0$ and $\det(v, v^{**}, v^{***}) < 0$ and $v \notin V_1^-$. Then $v \in V_1^+ \cup V_2^+ \cup V_2^-$. However, if $v \in V_1^+$, then $\det(v, v^{**}, v^{***}) < 0$ cannot be satisfied and if $v \in V_2^+ \cup V_2^-$, then $\det(v^*, v, v^{***}) = 0$ cannot be satisfied. Thus we have a contradiction. $\square$

Theorem 15. *Function EnforceAcycVC is correct.*

Proof. Function EnforceAcycVC chooses two linear independent vectors $v^*, v^{**} \in V$ (line 3) and moves all vectors in V_3^- to V_3^+ (lines 4–5), where the vectors in V_3^- are detected by applying Lemma 12. Then it moves all vectors in V_2^- to V_2^+ (lines 8–9) and all vectors in V_1^- to V_1^+ (lines 10–11), where Lemma 13 and Lemma 14 are applied, respectively. Since $V = V_1^+ \dot\cup V_1^- \dot\cup V_2^+ \dot\cup V_2^- \dot\cup V_3^+ \dot\cup V_3^-$ by Lemma 10 and V_1^-, V_2^-, V_3^- are empty, $V = V_1^+ \cup V_2^+ \cup V_3^+$. Thus V is acyclic by Theorem 11. □

Input: An oriented matroid χ.

Output: An oriented matroid that is realizable if and only if χ is realizable. The realization is acyclic.

begin
  $i^* \leftarrow 0, i^{**} \leftarrow 0, i^{***} \leftarrow 0$
  Choose $i, j, k \in \{1, \ldots, n\}$ such that $\chi(i,j,k) \neq 0$ and set $i^* \leftarrow i$ and $i^{**} \leftarrow j$
  foreach $(i,j,k) \in \{1,\ldots,n\}^3$ **do**
    if $\chi(i,j,k) \neq 0$ **then** $i^* \leftarrow i$ and $i^{**} \leftarrow j$
  foreach $i \in \{1,\ldots,n\}$ **do**
    if $\chi(i^*, i^{**}, i) < 0$ **then** `SwitchSign`(χ, i)
  foreach $i \in \{1,\ldots,n\}$ **do**
    if $\chi(i^*, i^{**}, i) > 0$ **then** $i^{***} \leftarrow i$
  foreach $i \in \{1,\ldots,n\}$ **do**
    if $\chi(i^*, i^{**}, i) = 0$ *and* $\chi(i^*, i, i^{***}) < 0$ **then** `SwitchSign`(χ, i)
  foreach $i \in \{1,\ldots,n\}$ **do**
    if $\chi(i^*, i^{**}, i) = 0$ *and* $\chi(i^*, i, i^{***}) = 0$ *and* $\chi(i, i^{**}, i^{***}) < 0$ **then** `SwitchSign`(χ, i)
  return V

Function EnforceAcycOM(V)

We can apply the concept underlying Function EnforceAcycVC to oriented matroids, such that an oriented matroid χ can be transformed to an oriented matroid χ' which is equivalent in realizability and, if χ' is realizable, then it has an acyclic realization. The transformation is implemented by Function EnforceAcycOM which is an one-to-one translation of Function EnforceAcycVC to the oriented matroid setting. The main difference is the use of Function SwitchSign(χ, i), which modifies χ to reflect the change of the sign of vector v_i.

Function EnforceAcycOM is correct: given an oriented matroid χ with a realization V, $\chi' = \text{EnforceAcycOM}(\chi)$ is an oriented matroid with an acyclic realization $V' = \text{EnforceAcycVC}(V)$. On the other hand, if χ is not realizable, then $\chi' = \text{EnforceAcycOM}(\chi)$ is not realizable as well, because if $\chi' = \text{EnforceAcycOM}(\chi)$ were realizable with a realization V', then one would obtain a realization V of χ by reversing the operations of switching signs in EnforceAcycOM. Note that EnforceAcycOM runs in $O(n^3)$.

From the correctness of Function EnforceAcycOM we can conclude the following theorem:

Theorem 16. *Given an oriented matroid χ one can transform it in polynomial time to an oriented matroid χ', such that χ' is realizable if and only if χ is realizable, and if χ' is realizable, then the realization is acyclic.*

Theorem 17. *CSP($\mathcal{LR}$) is $\exists\mathbb{R}$-hard.*

Proof. Since ROM is $\exists\mathbb{R}$-complete, it suffices to show that ROM can be reduced to CSP($\mathcal{LR}$) in polynomial time. Let an oriented matroid $\chi : \{1,\ldots,n\}^3 \mapsto \{-1,0,1\}$ be given. Since CSP($\mathcal{LR}$) requires point configurations in the plane but the realization of an oriented matroid are vectors in the 3-dimensional space, we generate a new oriented matroid χ' which is equivalent in realizability and has an acyclic realization when realizable, such that the realization of χ' can be identified with a point configuration in an affine space. This can be accomplished in polynomial time using Function EnforceAcycOM.

Input: An oriented matroid χ and an index i.

Output: An oriented matroid that is obtained by switching all signs of χ that involve i.

begin
  for $j \leftarrow 1$ **to** n **do**
    for $k \leftarrow 1$ **to** n **do**
      $\chi(i,j,k) \leftarrow -\chi(i,j,k)$
      $\chi(j,i,k) \leftarrow -\chi(j,i,k)$
      $\chi(j,k,i) \leftarrow -\chi(j,k,i)$
      $\chi(i,k,j) \leftarrow -\chi(i,k,j)$
      $\chi(k,j,i) \leftarrow -\chi(k,j,i)$
      $\chi(k,i,j) \leftarrow -\chi(k,i,j)$
  return χ

Function SwitchSign(χ, i)

Next we translate χ' to an instance φ of CSP($\mathcal{LR}$): first, we translate the numbers $1,\ldots,n$ in the domain $\{1,\ldots,n\}^3$ of χ' to variables $v_1, v_2, \ldots, v_n$ defined on the plane $\mathbb{R}^2$. Then we generate for each triple $(i,j,k) \in \{1,\ldots,n\}^3$ a constraint $v_i\, v_j\, r\, v_k$ if $\chi'(i,j,k) = -1$, $v_i\, v_j\, l\, v_k$ if $\chi'(i,j,k) = 1$, and $v_i\, v_j\, \{f,e,i,s,b\}\, v_k$ if $\chi'(i,j,k) = 0$ (cf. Figure 4). Because the translation does not change the semantics of χ', the oriented matroid χ' is realizable, if and only if φ is satisfiable. As the translations from χ to χ' and from χ' to φ are accomplished each in polynomial time, and χ is realizable, if and only if φ is satisfiable, we have obtained a polynomial-time reduction from ROM to CSP($\mathcal{LR}$). □

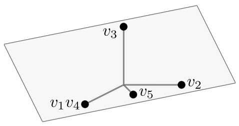


Figure 4: A realization of an acyclic oriented matroid χ with $\chi(1,2,3) = 1$, $\chi(1,2,4) = 0$, $\chi(1,3,5) = -1$, $\chi(2,3,4) = 1$ and so forth. Equivalently, we have $v_1v_2\, l\, v_3$, $v_1v_2\, \{f,e,i,s,b\}\, v_4$, $v_1v_3\, r\, v_5$, and $v_2v_3\, l\, v_4$.

Because relations in $\mathcal{DCC}$ are refinements of $\mathcal{LR}$ relations, and thus any $\mathcal{LR}$ relation can be described as a union of $\mathcal{DCC}$ relations, we have the following result.

Theorem 18. *CSP($\mathcal{DCC}$) is $\exists\mathbb{R}$-hard.*

The proof for the $\exists\mathbb{R}$-hardness of CSP($\mathcal{DRA}$) and CSP($\mathcal{OPRA}_m$) can be achieved similarly to Theorem 17. Since the proof is rather technical and gives no new insights, we omit the proof here and refer the reader instead to [12].

Theorem 19. *CSP($\mathcal{DRA}$) and CSP($\mathcal{OPRA}_m$) is $\exists\mathbb{R}$-hard.*

All in all, we have the following result:

Theorem 20. *Reasoning with relative directional constraint languages is $\exists\mathbb{R}$-hard.*

Now that CSPs for relative directional constraint languages (i.e., $\mathcal{LR}$, $\mathcal{DRA}$, $\mathcal{DCC}$ and $\mathcal{OPRA}_m$) are $\exists\mathbb{R}$-hard, we can ask whether at least the satisfiability of the *atomic* formulas of relative directional constraint languages can be decided in NP. However, this would imply the NP-membership of CSPs for relative directional constraint languages, as one can non-deterministically choose a relation in each conjunct of a formula and solve the atomic formula in NP. Therefore we have the following theorem.

Theorem 21. *Reasoning with atomic instances of a CSPfor a relative directional constraint language is not in* NP*, unless* NP $= \exists\mathbb{R}$.

That reasoning with relative directional constraint languages is in $\exists\mathbb{R}$ can be proved by translating their formulas to instances of ETR. For $\mathcal{LR}$ and $\mathcal{OPRA}_m$ this was shown in [14], and for CSP($\mathcal{DRA}$) and CSP($\mathcal{DCC}$) in [18] and [23], respectively.

Theorem 22. *Reasoning with relative directional constraint languages is $\exists\mathbb{R}$-complete.*

4 CONCLUSION

This paper proved that reasoning with any of the relative directional languages $\mathcal{LR}, \mathcal{DRA}, \mathcal{DCC}$ and $\mathcal{OPRA}_m$ is $\exists\mathbb{R}$-complete and thereby showed that reasoning with them is not in NP, unless NP $= \exists\mathbb{R}$. The same result holds even if only atomic formulas are considered. The investigation in this paper complements the investigation of topological constraint languages in [10] in that the present paper discovers the relative directional part of the complexity landscape of qualitative spatial reasoning.

As the doubly-exponential decision procedure cylindrical algebraic decomposition (CAD) [1] is more effective for ETR than those that have in theory only exponential algorithmic complexities (cf. [20, 2]), it seems unlikely that an NP decision procedure for reasoning with relative directions will be available in the near future, if at all. Indeed, empirical evaluations have shown that a modern CAD implementation was not even able to handle in reasonable time CSP($\mathcal{LR}$) instances with six or more variables [14]. Consequently, for applications involving relative directions, one should consider developing approximative algorithms or semi-decision procedures as in [13].

ACKNOWLEDGEMENTS

The author would like to thank Sanjiang Li and the anonymous ECAI reviewers for valuable comments that helped improve the paper. This work is partially supported by the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

[1] Dennis S Arnon, George E Collins, and Scott McCallum, 'Cylindrical algebraic decomposition i: the basic algorithm', *SIAM J. Comput.*, **13**(4), 865–877, (1984).
[2] Philippe Aubry, Fabrice Rouillier, and Mohab Safey El Din, 'Real solving for positive dimensional systems', *Journal of Symbolic Computation*, **34**(6), 543–560, (December 2002).
[3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy, *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics, Springer Berlin Heidelberg, 2006.
[4] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M Ziegler, *Oriented Matroids*, Cambridge University Press, 1999.
[5] John Canny, 'Some algebraic and geometric computations in PSPACE', in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, p. 460–467, New York, NY, USA, (1988). ACM.
[6] Anthony G. Cohn and Jochen Renz, 'Chapter 13 qualitative spatial representation and reasoning', in *Handbook of Constraint Programming*, 551–596, Elsevier, (2008).
[7] Frank Dylla and Reinhard Moratz, 'Exploiting qualitative spatial neighborhoods in the situation calculus', in *Spatial Cognition IV. Reasoning, Action, Interaction*, 304–322, Springer Berlin Heidelberg, (January 2005).
[8] Christian Freksa, 'Using orientation information for qualitative spatial reasoning', in *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 162–178, Springer Berlin Heidelberg, (January 1992).
[9] Lutz Frommberger, Jae Hee Lee, Jan Oliver Wallgrün, and Frank Dylla, 'Composition in OPRAm', Technical Report 013-02/2007, Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, (February 2007).
[10] Roman Kontchakov, Yavor Nenov, Ian Pratt-Hartmann, and Michael Zakharyaschev, 'On the decidability of connectedness constraints in 2D and 3D euclidean spaces', in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, p. 957–962, Barcelona, Catalonia, Spain, (2011). AAAI Press.
[11] Arne Kreutzmann, Diedrich Wolter, Frank Dylla, and Jae Hee Lee, 'Towards safe navigation by formalizing navigation rules', *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, **7**(2), 161–168, (2013).
[12] Jae Hee Lee, *Qualitative Reasoning about Relative Directions: Computational Complexity and Practical Algorithm*, Ph.D. dissertation, Universität Bremen, 2013.
[13] Jae Hee Lee, Jochen Renz, and Diedrich Wolter, 'StarVars: effective reasoning about relative directions', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, p. 976–982, Beijing, China, (2013). AAAI Press.
[14] Jae Hee Lee and Diedrich Wolter, 'A new perspective on reasoning with qualitative spatial knowledge', in *IJCAI-2011 Workshop 27*, pp. 3–8, (2011).
[15] Gérard Ligozat, *Qualitative Spatial and Temporal Reasoning*, John Wiley & Sons, May 2013.
[16] Gérard F Ligozat, 'Qualitative triangulation for spatial reasoning', in *Spatial Information Theory A Theoretical Basis for GIS*, 54–68, Springer Berlin Heidelberg, Berlin, Heidelberg, (1993).
[17] Dominik Lücke, *Qualitative Spatial Reasoning about Relative Orientation: A Question of Consistency*, Ph.D. dissertation, Universität Bremen, June 2012.
[18] Reinhard Moratz, Jochen Renz, and Diedrich Wolter, 'Qualitative spatial reasoning about line segments', in *ECAI 2000. Proceedings of the 14th European Conference on Artifical Intelligence*, p. 234–238. IOS Press, (2000).
[19] Till Mossakowski and Reinhard Moratz, 'Qualitative reasoning about relative direction of oriented points', *Artificial Intelligence*, **180–181**, 34–45, (April 2012).
[20] Grant Olney Passmore and Paul B. Jackson, 'Combined decision techniques for the existential theory of the reals', in *Intelligent Computer Mathematics*, 122–137, Springer Berlin Heidelberg, (January 2009).
[21] Jochen Renz and Gérard F Ligozat, 'Weak composition for qualitative spatial and temporal reasoning', in *Principles and Practice of Constraint Programming - CP 2005*, ed., Peter Van Beek, 534–548, Springer Berlin Heidelberg, (2005).
[22] Marcus Schaefer, 'Complexity of some geometric and topological problems', in *Graph Drawing*, 334–344, Springer Berlin Heidelberg, (January 2010).
[23] Alexander Scivos and Bernhard Nebel, 'Double-crossing: Decidability and computational complexity of a qualitative calculus for navigation', in *Spatial Information Theory*, 431–446, Springer Berlin Heidelberg, (January 2001).
[24] Alexander Scivos and Bernhard Nebel, 'The finest of its class: The natural point-based ternary calculus LR for qualitative spatial reasoning', in *Spatial Cognition IV. Reasoning, Action, Interaction*, 283–303, Springer Berlin Heidelberg, (January 2005).
[25] Peter W Shor, 'Stretchability of pseudolines is NP-hard', in *Applied Geometry and Discrete Mathematics–The Victor Klee Festschrift*, eds., P Gritzmann and B Sturmfels, 531–554, Amer. Math. Soc., (1991).
[26] Diedrich Wolter and Jae Hee Lee, 'Qualitative reasoning with directional relations', *Artificial Intelligence*, **174**(18), 1498–1507, (2010).