Reasoning with Lines in the Euclidean Space

Khalil Challita
Department of Computer Science
Holy Spirit University of Kaslik
khalilchallita@usek.edu.lb

Abstract

The main result of this paper is to show that the problem of instantiating a finite and path-consistent constraint network of lines in the Euclidean space is NP-complete. Indeed, we already know that reasoning with lines in the Euclidean space is NP-hard. In order to prove that this problem is NP-complete, we first establish that a particular instance of this problem can be solved by a nondeterministic polynomial-time algorithm, and then we show that solving any finite and path-consistent constraint network of lines in the Euclidean space is at most as difficult as solving that instance.

Keywords. Constraint satisfaction problems, Satisfiability, Qualitative spatial reasoning, Euclidean geometry.

1 Introduction

During the last three decades, a wide variety of formalisms concerning qualitative spatial reasoning have been proposed and extensively studied by researchers whose main interests lie in the field of artificial intelligence [Clarke, 1985; Randell *et al.*, 1992; Ligozat, 1998; Skiadopoulos and Koubarakis, 2004]. The fact is that most of the spatial information we encounter in our everyday's life can be modeled qualitatively, rather than quantitatively. An overview of qualitative spatial representation and reasoning, as well as current topics in qualitative reasoning can be found in [Cohn and Hazarika, 2001] and [Bredeweg and Struss, 2003], respectively.
Besides its natural association with qualitative temporal reasoning [Wolter and Zakharyaschev, 2000; Renz and Ligozat, 2005; Renz, 2007], qualitative spatial reasoning plays a central role in other fields such as mathematics [Liu, 1998; Moratz *et al.*, 2000] and logics [Gabelaia *et al.*, 2005; Kontchakov *et al.*, 2008]. It is worth noting that few spatial models that are related to elementary geometry have been introduced so far in the field of AI, despite the fact that we find it very interesting to study constraint satisfaction problems whose variables are elementary mathematical structures. Indeed, Balbiani and Tinchev [2007] studied lines in the Euclidean plane and showed that the satisfiability problem of any finite and path-consistent network of lines in the Euclidean plane can be solved in polynomial time. A similar problem that was partially solved [Balbiani and Challita, 2004] concerned lines in the Euclidean space, where we proved that the satisfiability problem of any finite and path-consistent network of lines in the Euclidean space is NP-hard. In this paper we determine an upper bound for that problem by proving that it is in the class NP, and thus NP-complete. In order to achieve our aim, we follow a slightly different approach than the usual one: instead of providing a nondeterministic polynomial-time algorithm that solves any instance of that problem, we first establish that a particular instance of this problem belongs to the class NP, and then we show that solving any finite and path-consistent constraint network of lines in the Euclidean space is at most as difficult as solving that instance. In other words, we only prove in this paper the *existence* of a nondeterministic polynomial-time algorithm that solves our problem by providing an upper bound on the number of possible instantiations of a networks variables.
The reason behind using this strategy is that providing a nondeterministic polynomial-time algorithm for this problem turned out to be a nontrivial task.
This paper is divided as follows. We give in section 2 the terminology needed to qualitatively compare lines in the Euclidean space, and then recall some results about lines in dimension three. In section 3, we study particular spatial networks and show that instantiating any of their variables is bounded by a polynomial. Then, in section 4, we prove that the consistency problem of these networks is NP-complete, before extending in section 5 the same result to any spatial network. Finally, we conclude in section 6.

2 Lines in dimension 3

Four basic relations are needed to compare qualitatively two lines in the Euclidean space. We denote them by PO, EQ, DC, NC, which respective meanings are: "having one point in common", "equal", "parallel and distinct", and "non coplanar". Two lines in the space, denoted by l and l', can be exactly in one of the following relations: $l\{PO\}l'$, $l\{EQ\}l'$, $l\{DC\}l'$, $l\{NC\}l'$. Let $E = \{PO, EQ, DC, NC\}$ be the set of the jointly exhaustive and pairwise disjoint relations that compare the position of any couple of lines in the Euclidean space.
The definitions needed for describing a constraint satisfaction problem (CSP) can be found in [Montanari, 1974]. We next

recall some of them.

A network of linear constraints $\mathcal{N}$ is a couple (I, C), where $I \subseteq \mathbb{N}$ is a finite set of variables, and C is a mapping from I^2 to the set of the subsets of E (i.e. 2^E). The network $\mathcal{N}$ is *atomic* if for all $i, j \in I$, $Card(C(i,j)) > 1$ then $C(i,j) = E$. We say that $\mathcal{N}$ is *path-consistent* if for all $i, j \in I$, $C(i,j) \subseteq C(i,k) \circ C(k,j)$, for every $k \in I$. A *scenario* (or an instantiation) is a function f that maps I to a set of lines in the Euclidian space. An instantiation is consistent if, for all $i, j \in I$, the relation satisfied between the lines $l_i = f(i)$ and $l_j = f(j)$ is in $C(i,j)$.

The algorithm of path consistency is explored and analyzed in [Mackworth, 1977; Mackworth and Freuder, 1985]. The constraint propagation algorithm due to Allen [1983], that replaces each constraint $C(i,j)$ by $C(i,j) \cap (C(i,k) \circ C(k,j))$, transforms in polynomial time a finite network $\mathcal{N}$ into a path-consistent one, whose set of consistent scenarios is the same as the one for $\mathcal{N}$.

For the rest of this paper, a constraint network will be denoted by $\mathcal{N} = (V, C)$, where $V = \{v_1, \ldots, v_n\}$ is a finite set of variables of $\mathcal{N}$, and C is a mapping from V^2 to 2^E. Without loss of generality, we assume in this paper that $\mathcal{N}$ is a *complete network*. This will allow us to instantiate the variables of $\mathcal{N}$ more easily, as we shall see later on in sections 3 and 4.

Definition 1 *A network of linear constraints $\mathcal{N} = (V, C)$ is complete if the following condition holds: for any couple u, v of its variables, there exists a constraint between u and v.*

Notice that any network of constraints can be transformed into an equivalent complete network, using a polynomial-time algorithm.

Indeed, let $\mathcal{N}$ be a path-consistent constraint network of lines in the Euclidean space. For each node $v \in V$ of this network, we check whether for all $u \in V$ such that $u \neq v$, there is a constraint $C(v,u)$ between v and u. If this is not the case for a particular node u, then we can add the constraint $\{PO, EQ, DC, NC\}$ between v and u.

Definition 2 *A spatial network is a finite and path-consistent linear constraint network of lines in the Euclidian space.*

We next recall two results established in [Balbiani and Challita, 2004]: the composition table given in figure 1 and proposition 1.

$\circ$	EQ	DC	PO	NC
EQ	EQ	DC	PO	NC
DC	DC	DC, EQ	PO, NC	PO, NC
PO	PO	PO, NC	PO, NC, DC,EQ	PO, NC, DC
NC	NC	PO, NC	PO, NC, DC	PO, NC, DC,EQ

Figure 1: Composition table of spatial relations.

Proposition 1 *The consistency problem of spatial networks is NP-hard.*

To prove that the problem of finding a consistent instantiation of a spatial network is NP-complete, we still have to show that it is in the class NP. Addressing the problem in its generality (i.e. when considering *any* spatial network $\mathcal{N}$) seems to be very difficult because of the multitude of possible instantiations of the variables of $\mathcal{N}$. The problem is that to find a consistent instantiation of $\mathcal{N}$, one might have to instantiate some of its variables several times which, in some cases, could require an exponential time.

***Example* 1** *Consider the case of a network where three of its variables (v_1, v_2, v_3) are constrained by the relation PO. We cannot tell in advance which instantiation of these variables would be consistent. If another variable v is in the relation PO with them, then we deduce that lines l_1, l_2, l_3 must be coplanar. On the other hand, if v is in the relation PO with two of them and NC with the third one, then lines l_1, l_2, l_3 should intersect in a single point without being all coplanar.*

3 Particular spatial networks

We consider in this section some particular spatial networks. More precisely, we determine the number of possibilities to consistently instantiate the k^{th} variable of each of these networks.

Definition 3 *For any basic relation $R \in \{EQ, PO, DC, NC\}$, let $\mathcal{N}_R^{n,n'} = (V, C)$ be a finite and complete network of constraints such that $|V| = n' \geq n$ and where the relation that holds between any couple of its first n variables is R.*

In the following subsections, we determine an upper bound for the problems of consistently instantiating spatial networks of the form $\mathcal{N}_{PO}^{n,n'}$, $\mathcal{N}_{NC}^{n,n'}$, and $\mathcal{N}_{DC}^{n,n'}$; the case of $\mathcal{N}_{EQ}^{n,n'}$ being trivial.

Note that in order to determine the number of possible instantiations of the k^{th} variable of $\mathcal{N}_R^{n,n'}$, $(1 \leq k \leq n)$, we take into account all the possible relations that might constraint v_{k+1} to the previously instantiated variables. From now on, the Euclidean plane defined by two distinct lines l_j and $l_{j'}$ will be denoted by $P_{j,j'}$.

3.1 Networks of the form $\mathcal{N}_{PO}^{n,n'}$

Question 1 *Assuming that we already instantiated $n-1$ variables of $\mathcal{N}_{PO}^{n,n'}$, how many possibilities are there to instantiate the n^{th} one?*

Since all the lines $l_1, \ldots, l_{n-1}$ should intersect with each other, we distinguish 2 cases when instantiating v_n:

1. *Case 1.* The previous lines are not included in one single plane.
   In this case all the lines must intersect in one single point. So l_n must also pass through that point, and can be included (or not) in any plane already defined by two lines $l_j, l_{j'}$, where $1 \leq j, j' \leq n-1$.
   The total number of such planes is equal to $\sum_{i=1}^{n-2} i + 1$, where the $(+1)$ denotes the choice of drawing l_n outside of the planes $P_{j,j'}$.
   Note that the case where l_n is included in a plane defined by three (or more) lines is a special case of saying that l_n is included in a plane defined by just two of those three

(or more) lines. That is why we just considered planes defined by 2 lines.

2. *Case 2*. All the previous lines are included in the same plane P. We distinguish two subcases:
   (a) All the lines intersect in one single point p. We have three possibilities for l_n:
      i. Line l_n is not coplanar with the previous lines.
      ii. Line l_n is coplanar with the previous lines and passes through p.
      iii. Line l_n is coplanar with the previous lines and does not pass through p.
   (b) They do not intersect in one single point. In this case l_n must also be included in P. Let $p_{j,j'} \in P$, where $1 \leq j, j' \leq n-1$, be the point of intersection of lines l_j and $l_{j'}$. Note that the total number of such points is at most equal to $\sum_{i=1}^{n-2} i = O(n^2)$. We distinguish three subcases:
      i. Line l_n passes through two different points: $p_{j,j'}$ and $p_{k,k'}$.
      ii. Line l_n passes through one point $p_{j,j'}$, where $1 \leq j, j' \leq n-1$.
      iii. Line l_n intersects all the previous lines without passing through any point $p_{j,j'}$ that belongs to two or more lines.

It is easy to see that the worst case happens for subcase (b) in case 2; and that the total number of different possibilities of drawing line l_n is bounded by $O(n^4)$.

Answer 1. The number of possibilities is bounded by $O(n^4)$.

3.2 Networks of the form $\mathcal{N}_{NC}^{n,n'}$

Question 2 *Assuming that we already instantiated $n-1$ variables of $\mathcal{N}_{NC}^{n,n'}$, how many possibilities are there to instantiate the n^{th} one?*

Suppose that all the previous lines have been drawn in $p \geq 1$ groups $G_1, \ldots, G_p$, where for all the lines that belong to the same group G_k, there exists a line l^k in the space that intersects them all. Note that line l^k is different from l_k, which is the instantiation of the variable v_k; and that a line l_k can, at the same time, belong to several groups $G_{\sigma_1}, \ldots, G_{\sigma_r}$ where $1 \leq \sigma_i \leq p$.
Line l_n can be included in any of these groups or it can be part of a new one. In the former case we have p possibilities, whereas in the latter case line l_n could form a new group G_{p+1} with any couple of lines amongst $l_1, \ldots, l_{n-1}$, and such that line l^{p+1} could be in the relation DC, NC, or PO with one or more of the lines $l^1, \ldots, l^p$ and $l_1, \ldots, l_{n-1}$ (this condition will help us determine the *direction* of l^{p+1}).
We know that there are $C_{n-1}^2 = O(n^2)$ ways of forming groups of two lines amongst $l_1, \ldots, l_{n-1}$. Moreover, there are $O(n^2)$ *directions* for line l^{p+1}. Assuming that $p = O(n)$, we conclude that:

Answer 2. The number of possibilities is bounded by $O(n^5)$.

3.3 Networks of the form $\mathcal{N}_{DC}^{n,n'}$

Question 3 *Assuming that we already instantiated $n-1$ variables of $\mathcal{N}_{DC}^{n,n'}$, how many possibilities are there to instantiate the n^{th} one?*

This is probably the most difficult case to deal with. Indeed, a variable v_{n+1} could be in the relation PO with $k \geq 2$ other variables of a network $\mathcal{N}'$ and with the relation NC with the remaining ones.
We next assume that the first $n-1$ lines have been drawn in p parallel planes $P_1, \ldots, P_p$; where each plane P_i contains k_i lines, for a total of $\sum_{i=1}^{p} k_i = n-1$ lines.
We next describe all the qualitative ways to draw line l_n.
Line l_n could be drawn in any plane P_i or in a new one. Moreover, one of the following cases holds:

1. Line l_n could be included in a plane $P_{j,j'}$ defined by 2 existing lines l_j and $l_{j'}$, where $1 \leq j, j' \leq n-1$;
2. Or it could be included in a plane P parallel to $P_{j,j'}$ defined by 2 existing lines l_j and $l_{j'}$, where $1 \leq j, j' \leq n-1$, and such that P contains another line $l_{j''}$ (with $1 \leq j'' \leq n-1$, and $j'' \neq j$, and $j'' \neq j'$).

It is easy to see that the number of possibilities of drawing line l_n in case 2 is greater than the one of drawing it in case 1. Indeed, after fixing a plane $P_{j,j'}$, we have $O(n)$ possibilities of selecting a third line $l_{j''}$ that satisfies the conditions of case 2.
We next determine the total number of possibilities (denoted by S) of drawing line l_n in case 1.

***Remark* 1** *Taking into account case 2 would lead us to multiply S by n.*

Let us assume that l_n is drawn in a plane P_m, where $1 \leq m \leq p$. The number of different possibilities to draw it is given by:

$$1 + \sum_{\substack{i=1 \\ i \neq m}}^{p-1} \sum_{\substack{j=i+1 \\ j \neq m}}^{p} k_i k_j \tag{1}$$

The double-sum takes into account all the possible planes $P_{j,j'}$ that might contain l_n. The $(+1)$ denotes the choice of drawing l_n outside of all the planes $P_{j,j'}$ and in a new plane P_{p+1}.
Since we can draw l_n in any of the planes $P_1, \ldots, P_p$, we deduce that the overall number of possibilities to draw it in any plane $P_{j,j'}$ is:

$$S = \underbrace{p + \sum_{m=1}^{p} \left(\sum_{\substack{i=1 \\ i \neq m}}^{p-1} \sum_{\substack{j=i+1 \\ j \neq m}}^{p} k_i k_j \right)}_{l_n \, is \, drawn \, in \, a \, plane \, P_m} + \underbrace{1 + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} k_i k_j}_{l_n \, is \, drawn \, in \, a \, new \, plane} \tag{2}$$

As we can see from equation 2, and because of the factor $k_i k_j$, the total number of possible instantiations of a constraint network $\mathcal{N}_{DC}^{n,n'}$ depends on the way the lines are distributed in the parallel planes P_i, $1 \leq i \leq p$ (see Appendix A).

Proposition 2 *The total number of possible instantiations of the n^{th} variable of $\mathcal{N}_{DC}^{n,n'}$ is bounded by $O(n^6)$.*

Proof 1 *We first notice that the complexity of the sum S is the same as its leading term S', where:*

$$S' = \sum_{m=1}^{p} \left(\sum_{\substack{i=1 \\ i \neq m}}^{p-1} \sum_{\substack{j=i+1 \\ j \neq m}}^{p} k_i k_j \right) \quad (3)$$

We then consider the worst case where we assume that the lines are distributed over p planes with $p = O(n)$, and that the number of lines in each plane P_i depends on n too (i.e. $k_i = O(n)$, for all $i \in \{1, \ldots, p\}$).
For all $i, j \in \{1, \ldots, p\}$, we have $k_i k_j = O(n^2)$. Thus, the double sum in S' is bounded by $O(n^4)$. Since S' is the leading term of S and that $p = n$, we deduce that $S = O(n^5)$; and hence conclude that the total number of possible instantiations of $\mathcal{N}_{DC}^{n,n'}$ is bounded by $nS = O(n^6)$ (refer to Remark 1).

Answer 3. The number of possibilities is bounded by $O(n^6)$.
The next proposition follows from the results established in this section.

Proposition 3 *Let $\mathcal{N}_R^{n,n'}$, where $R \in \{EQ, PO, DC, NC\}$.*
The total number of possible instantiations of the k^{th} variable of $\mathcal{N}_R^{n,n'}$, where $1 \leq k \leq n$, is bounded by $O(k^6)$.

Indeed, the worst case happens when we consider networks of the form $\mathcal{N}_{DC}^{n,n'}$.
The method we described here enlightens us on how to prove that the consistency problem of particular spatial networks is in NP, which is the subject of the next section.

4 NP-completeness results

We first prove in this section that the consistency problem of some of the particular networks (i.e. $\mathcal{N}_R^{n,n+1}$) we saw in section 3 is NP-complete, and then we extend this result to any finite spatial network.
To establish proposition 4, one must provide a nondeterministic polynomial-time algorithm that solves the consistency problem of each of the particular networks we previously described. The aim of the following algorithms is to prove proposition 4.

4.1 $\mathcal{N}_{DC}^{n,n+1}$ networks

In the below pseudo-code, p_i denotes the number of parallel planes that are used to draw lines l_1 to l_{i-1}; k_j is equal to the number of lines drawn in the plane P_j; and P_{r_1,r_2} represents the plane defined by the lines l_{r_1} and l_{r_2}.
In the below algorithms, by "randomly number the variables of $\mathcal{N}_R^{n,n}$" we mean that any numbering of the first n variables of these networks would work. Moreover, by "*guess* a number q" we mean "*nondeterministically* select a line l_q or a plane P_q", depending on the context.

Algorithm 1. *Given $\mathcal{N}_{DC}^{n,n+1}$:*

1 randomly number the variables of $\mathcal{N}_{DC}^{n,n}$;
2 **for** $i = 1$ **to** n
3 guess a number q between 0 and p_i;
4 guess a number n_0 between 1 and 2;
5 **if** $n_0 = 1$ **then**
6 {guess $q' \neq q$ and $q'' \neq q$ between 1 and p_i;
7 guess $r_1 \in \{0, 1, \ldots, k_{q'}\}$ and $r_2 \in \{0, 1, \ldots, k_{q''}\}$;
8 draw l_i in the plane P_q, where l_i
9 is included in P_{r_1,r_2};}
10 **else**
11 {guess $q' \neq q$ and $q'' \neq q$ between 1 and p_i;
12 guess $r_1 \in \{0, 1, \ldots, k_{q'}\}$ and $r_2 \in \{0, 1, \ldots, k_{q''}\}$;
13 guess a number r_0 between 1 and $n - 1$;
14 draw l_i in P_q and P_{r_0} where $P_{r_0} // P_{r_1,r_2}$;
15 draw l_{n+1};}

Intuitively, q represents the plane P_q in which to draw line l_i. If $q = 0$, then l_i is drawn in a new plane parallel to the other ones.
Furthermore, in the first case where $n_0 = 1$, we *guess* the planes (i.e. $P_{q'}$ and $P_{q''}$) that contain the lines (i.e. l_{r_1} and l_{r_2}) defining the plane that should include l_i.
Note that if $r_1 = 0$ or $r_2 = 0$, then the line l_i is drawn in plane P_q but is not included in *any* plane defined by two existing lines $l_j, l_{j'}$, where $1 \leq j, j' \leq i - 1$.
The second case, where $n_0 = 2$ is similar to the first one except that on line 13 we guess a line l_{r_0} such that P_{r_0} contains l_i and is parallel to P_{r_1,r_2}.
Note that drawing l_{n+1} is straightforward and is based on the constraints of v_{n+1} with all the previous variables of $\mathcal{N}_{DC}^{n,n+1}$.
It is easy to see that the running time of this algorithm is polynomial in the size of the spatial network $\mathcal{N}_{DC}^{n,n}$.

***Remark* 2** *During iteration i, it is worth noting that keeping track of all the planes $P_{j,j'}$ that are defined by all the couple of lines $l_j, l_{j'}$, where $1 \leq j, j' \leq i - 1$, requires polynomial space; and hence our algorithm would belong to the class $NPSPACE$.*[1]

Question 4 *A fundamental question is:*
How to draw line l_i in a plane P_q in such a way that it is not included in any plane defined by an already existing two lines $l_j, l_{j'}$, with $1 \leq j, j' \leq i - 1$, and without keeping track of all the planes $P_{j,j'}$?

Answer 4.
The trick lies in the way we actually draw line l_i in the Euclidean space, and the fact that the number of lines we already drew is finite.
We start by selecting for example the rightmost line (denoted by l_r) amongst all the lines included in the planes $(P_j)_{1 \leq j \leq p_i}$. We then draw l_i inside the plane P_q and to the right of the vertical plane to P_q that passes through l_r.
In this way we are sure to answer question 4 by just keeping track of one line (in this case the rightmost one).
Thus proposition 4 holds for the case where $R = DC$.

4.2 $\mathcal{N}_{PO}^{n,n+1}$ networks

We next provide a nondeterministic polynomial-time algorithm that solves spatial networks of the form $\mathcal{N}_{PO}^{n,n+1}$.

[1] We know by Savitch's theorem [Savitch, 1970] that $NPSPACE = PSPACE$, but this is not of help for us since we do not know whether $PSPACE \subseteq NP$.

Algorithm 2. *Given* $\mathcal{N}_{PO}^{n,n+1}$:

1 randomly number the variables of $\mathcal{N}_{PO}^{n,n}$;
2 **for** $i = 1$ **to** n
3   guess a number n_0 between 1 and 2;
4   **if** $n_0 = 1$ **then**
5   {guess two points $p_{j,j'}$ and $p_{k,k'}$ or none;
6   draw l_i such that it passes through $p_{j,j'}$
7   and $p_{k,k'}$;}
8   **else**
9   {guess $r_1 \in \{0, 1, \ldots, i-1\}$
10   and $r_2 \in \{0, 1, \ldots, i-1\}$;
11   **if** $(r_1 = 0$ or $r_2 = 0$ or $r_1 = r_2)$
12   **then** draw l_i in a new plane;
13   **else** draw l_i in the plane P_{r_1,r_2};
14 draw l_{n+1};}

Intuitively, the case where $n_0 = 1$ (resp. $n_0 = 0$) corresponds to the fact that all the lines are included in the same plane (resp. intersect in one point).
On line 5, if we guess the same point (resp. we guess none) then subcase (b) (resp. (c)) of case 2 in subsection 3.1 applies.
Moreover, during iteration i we just need to keep track of the last line that was drawn in a new plane (denoted by $l_{new,i}$). This would save us from using polynomial-space to keep track of all the possible planes that can be defined by two lines $l_j, l_{j'}$.
The idea is to draw all the new lines (that must pass through a single point) with, for example, a *slight* rotation to the right with respect to $l_{new,i}$, and in such a way not to exceed an angle of 90^0 with l_1.
It is easy to see that proposition 4 holds for the case where $R = PO$.

4.3 $\mathcal{N}_{NC}^{n,n+1}$ networks

We next provide a nondeterministic polynomial-time algorithm that solves spatial networks of the form $\mathcal{N}_{NC}^{n,n+1}$.

Algorithm 3. *Given* $\mathcal{N}_{NC}^{n,n+1}$:

1 randomly number the variables of $\mathcal{N}_{NC}^{n,n}$;
2 **for** $i = 1$ **to** n
3   guess 1 number $r_1 \in \{0, 1, \ldots, \sigma_i\}$;
4   **if** $(r_1 \neq 0)$
5   **then** draw l_i in group G_{r_1};
6   **else**
7   {guess $r_1, r_2 \in \{1, \ldots, i-1\}$;
8   guess q_0 in $\{1, \ldots, i-1\}$ or in $\{1, \ldots, \sigma_i\}$;
9   guess q'_0 in $\{0, 1, 2\}$;
10   draw l_i in a new group defined by
11   l_{r_1}, l_{r_2} and l^{σ_i+1};
12 draw l_{n+1};}

Recall that on line 5, when l_i is drawn in group G_{r_1}, this means that it intersects with l^{r_1} (cf. subsection 3.2).
On line 7, we guess 2 lines with which l_i forms a new group G_{σ_i+1}. Steps 8 and 9 help us determine the direction of l^{σ_i+1} based on its relationship with line l_{q_0}. We have 3 possibilities: DC, NC and PO. For example, $q'_0 = 0$ could mean that l^{σ_i+1} is in the relation DC with l_{q_0}.

It is easy to see that proposition 4 holds for the case where $R = NC$.

Proposition 4 *The consistency problem of* $\mathcal{N}_R^{n,n+1}$*, where* $R \in \{EQ, PO, DC, NC\}$*, is NP-complete.*

Actually, there exists a deterministic polynomial-time algorithm that solves the consistency problem of $\mathcal{N}_R^{n,n+1}$ (e.g. it suffices to start by instantiating v_{n+1}). But our idea of using a nondeterministic algorithm to instantiate the k^{th} variable of $\mathcal{N}_R^{n,n+1}$ allows us to generalize the result established in the above proposition to any network $\mathcal{N}_R^{n,n'}$.

5 Extending the result to any finite spatial network

In this section, we show that the consistency problem of any finite spatial network is NP-complete.
Recalling proposition 3, we have:

Fact 1 *Let* $\mathcal{N}' = \mathcal{N}_R^{n,n'}$*, where* $R \in \{EQ, PO, DC, NC\}$*. For any* $k \in \{1, 2, \ldots, n'\}$*, the number of different instantiations of* v_k *in* $\mathcal{N}'$ *is bounded by* $O(k^6)$*.*

Indeed, it is easy to see that the more different types of constraints a spatial network has, the less consistent instantiations its variables have.
For example, let $\mathcal{N}' = \mathcal{N}_{PO}^{3,4}$ and $\mathcal{N}'' = \mathcal{N}_{PO}^{4,4}$ be two spatial networks where the fourth variable of $\mathcal{N}'$ is in the relation DC with v'_1 and in the relation PO with v'_2 and v'_3. The constraint DC forces all the lines (l'_1 to l'_4) to belong to the same plane. We can easily see that the number of consistent instantiations of $\mathcal{N}'$ is less than that of $\mathcal{N}''$.
When instantiating the k^{th} variable of $\mathcal{N}'$, and assuming that the first $k-1$ variables are constrained with each other with the same relation R, we distinguish two cases:

1. The constraints that relate v_k to all the previous $k-1$ variables are all equal to R;
2. There exists at least one constraint that relates v_k to another variable $v_{k'}$ ($k' < k$) that is different from R;

Thus, it is easy to see that there are less possible instantiations of v_k in the latter case than in the former case because of the constraint $C_{v_k,v_{k'}}$.

Proposition 5 *Let* $\mathcal{N}$ *be a finite spatial network. The problem of finding a consistent instantiation of* $\mathcal{N}$ *is NP-complete.*

Proof 2 *The proof of this proposition follows directly from fact 1 and from noticing that any finite spatial network belongs to the set:*

$$\{\mathcal{N}_R^{n,n'} : R \in \{EQ, PO, DC, NC\}; n, n' \in \mathbb{N}^*; n' \geq n\}.$$

We conclude this section by stating that there exists a nondeterministic polynomial-time algorithm that solves the consistency problem of $\mathcal{N}$.

6 Conclusion

In this paper, we showed that the consistency problem of finite and path-consistent constraint networks of lines in the Euclidean space is NP-complete. To achieve our aim, we first proved the above result for particular spatial networks, and then extended it to any spatial network. As we already stated in the introduction, we did not provide explicitly a nondeterministic polynomial-time algorithm that solves the consistency problem of a spatial network, but rather proved the existence of such an algorithm.

Our next aim is to try to provide an explicit algorithm that solves this problem. Moreover, we intend to find an answer to the following questions: would the consistency problem of lines in the Euclidean space remain the same in the case of infinite networks? What are all the tractable subclasses of this formalism? For the time being, all we know is that if we remove the basic relation NC, the consistency problem of lines in the Euclidean plane becomes tractable.

References

[Allen, 1983] James Allen. Maintaining knowledge about temporal intervals. In *Communications of the Association for Computing Machinery*, pages 832–843, 1983.

[Balbiani and Challita, 2004] Philippe Balbiani and Khalil Challita. Solving constraints between lines in euclidean geometry. In *AIMSA*, pages 148–157. Springer Berlin/Heidelberg, 2004.

[Balbiani and Tinchev, 2007] Philippe Balbiani and Tinko Tinchev. Line-based affine reasoning in euclidean plane. *Journal of Applied Logic*, 5(3):421–434, 2007.

[Bredeweg and Struss, 2003] Bert Bredeweg and Peter Struss. Current topics in qualitative reasoning. *AI Magazine*, 24:6–13, 2003.

[Clarke, 1985] Bowman Clarke. Individuals and points. *Notre Dame Journal of Formal Logic*, 26(1):61–75, 1985.

[Cohn and Hazarika, 2001] Anthony Cohn and Shyamanta Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29, 2001.

[Gabelaia *et al.*, 2005] David Gabelaia, Roman Kontchakov, A'gnes Kurucz, and Frank Wolter. Combining spatial and temporal logics: Expressiveness vs. complexity. *Journal of Artificial Intelligence Research*, 23:167–243, 2005.

[Kontchakov *et al.*, 2008] Roman Kontchakov, Ian Pratt-Hartmann, Frank Wolter, and Michael Zakharyaschev. On the computational complexity of spatial logics with connectedness constraints. In *LPAR*, pages 574–589, 2008.

[Ligozat, 1998] Gérard Ligozat. Reasoning about cardinal directions. *Visual Languages and Computing*, pages 23–44, 1998.

[Liu, 1998] Jiming Liu. A method of spatial reasoning based on qualitative trigonometry. *Artificial Intelligence*, pages 137–168, 1998.

[Mackworth and Freuder, 1985] Alan Mackworth and Eugene Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, pages 65–74, 1985.

[Mackworth, 1977] Alan Mackworth. Consistency in networks of relations. *Artificial Intelligence*, pages 99–118, 1977.

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

[Moratz *et al.*, 2000] Reinhard Moratz, Jochen Renz, and Diedrich Wolter. Qualitative spatial reasoning about line segments. In *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, pages 234–238. Wiley, 2000.

[Randell *et al.*, 1992] David Randell, Zhan Cui, and Anthony Cohn. A spatial logic based on regions and connection. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufman, 1992.

[Renz and Ligozat, 2005] Jochen Renz and Gérard Ligozat. Weak composition for qualitative spatial and temporal reasoning. In *CP*, pages 534–548. American Association for Artificial Intelligence, 2005.

[Renz, 2007] Jochen Renz. Qualitative spatial and temporal reasoning: Efficient algorithms for everyone. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 526–531, 2007.

[Savitch, 1970] Wolter Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970.

[Skiadopoulos and Koubarakis, 2004] Spiros Skiadopoulos and Manolis Koubarakis. Composing cardinal directions relations. *Artificial Intelligence*, pages 143–171, 2004.

[Wolter and Zakharyaschev, 2000] Frank Wolter and Michael Zakharyaschev. Spatio-temporal representation and reasoning based on rcc-8. In *KR*, pages 3–14, 2000.

A Instantiating a network of the form $\mathcal{N}_{DC}^{n,n'}$

In this appendix, we give an example of an instantiation of $\mathcal{N}_{DC}^{n,n'}$ and determine the number of all possible consistent instantiations of v_n.

***Example* 2** *Suppose that $n = m^2$ and that in each plane P_i we have $m = \sqrt{n}$ lines.*
In this case $S = O(n^3\sqrt{n})$, and the overall number of possible instantiations of v_n is $O(n^4\sqrt{n})$.

Proof 3 *Notice that the number of planes is also equal to m. We then compute the leading term S' of S:*

$$\begin{aligned} S' &= m \times ((n-1) \times (m \times m) + \cdots + (1) \times (m \times m)) \\ &= m \times ((n-1)(n) + (n-2)(n) + \cdots + n) \\ &= m \times (n(n-1)/2) \times (n) \\ &= O(n^3 m) \end{aligned}$$

After multiplying the result by n, we get $O(n^4\sqrt{n})$.