

Dynamic Path Consistency for Spatial Reasoning

Lamia Belouaer, Maroua Bouzid
GREYC - CNRS (UMR0672)
Université de Caen, Basse-Normandie, ENSICAEN
Boulevard du Maréchal Juin
Caen, France
Email: lamia.belouaer@info.unicaen.fr
bouzid@unicaen.fr

Malek Mouhoub
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada
Email: mouhoubm@cs.uregina.ca

Abstract—Dealing with spatial knowledge requires the consistency of spatial information. This consistency is usually enforced by constraint satisfaction techniques including constraint propagation through arc and path consistency. While these techniques often assume that spatial information are static, this is in general not the case in the real world. Our goal is to propose an approach to maintain the consistency of spatial knowledge in a dynamic environment. To our best knowledge no work in spatial reasoning has addressed this issue. In this paper we use a spatial ontology called *SpaceOntology* to describe both objects and spatial relations namely topological and distance relations between these objects. Based on a dynamic path consistency algorithm, our proposed method maintains the consistency of spatial information after adding new instances of topological relations described by *SpaceOntology* of a given environment. In order to evaluate the performance of our dynamic path consistency method, we conducted several tests on instantiations of *SpaceOntology* in addition to randomly generated spatial constraint problems. The results of these tests demonstrate the efficiency of our method to deal with large size problems in a dynamic environment.

Keywords-Dynamic qualitative spatial reasoning, dynamic temporal reasoning, constraint satisfaction problem.

I. INTRODUCTION

Several studies in Qualitative Spatial Reasoning (QSR) [6] have been dedicated to topological relations. Most of them are based on the algebra of intervals [4]. Qualitative spatial reasoning covers several applications, including Geographic Information Systems (GIS) [3], language processing [2], etc.

While a considerable literature has been dedicated to spatial reasoning, to our knowledge, there is no work that addressed the dynamic aspect of space in spatial reasoning and especially in qualitative spatial reasoning. However, in the real world, spatial knowledge is not static since the environment evolves over time. This evolution can be caused by external events or by the effects of executing some actions. In this paper, we focus on reasoning about qualitative spatial knowledge in a dynamic way which is relevant in many real world applications including reactive planning and path finding.

Let us consider the example of an accident at a nuclear plant. Generally if something goes wrong there are always

planned paths from any position in the nuclear plant to the exit according to certain security criteria. These paths are called emergency exits. In the case of a disaster that causes unexpected incidents such as earthquakes, some emergency exits may be blocked which can cause considerable damage. Moreover, in these situations the temporal dimension is very important. Indeed, the deadlines are very short. Thus, the time to compute a new path must be done very quickly. This time depends on the consistency check of spatial knowledge needed to compute a feasible path to the exit.

In this regard, several techniques based on the Constraint Satisfaction Problem (CSP) formalism have been proposed [15] [7]. To ensure the consistency of the spatial information, these latter are first represented as a CSP. The consistency of this latter is then checked. In real world problems, the major limitation of these approaches is their ability to deal with the dynamic change of information. A CSP that is consistent at a given time can become inconsistent in the future with the change of a given information. The challenge here is to maintain the consistency of the CSP anytime there is an addition or a retraction of a constraint.

In order to address this issue, we proceed as follows. (1) we use *SpaceOntology* [5] to represent the spatial knowledge (2) we adapt the dynamic path consistency algorithm (initially proposed for temporal constraints) [8] for the case of spatial relations. In order to evaluate the performance of our dynamic path consistency method, we conducted several tests on instantiations of *SpaceOntology* in addition to randomly generated spatial constraint problems. The results of these tests demonstrate the efficiency of our method to deal with large size problems in a dynamic environment.

The rest of the paper is structured as follows. In the next section we present an overview of *SpaceOntology* with a focus on topological relations. In section III we describe in details our dynamic path consistency algorithm for topological relations. A case study illustrating our proposed technique is then covered in Section IV. Section V reports the experimental tests evaluating the performance of our dynamic method. Finally, concluding remarks and future works are listed in Section VI.

$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{Y}{X}$	$\frac{Y}{X}$
$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y}$
$\frac{left}{(L)}$	$\frac{overlaps-left}{(O_L)}$	$\frac{contains}{(C_x)}$	$\frac{inside}{(I_x)}$	$\frac{overlaps-right}{(O_R)}$	$\frac{right}{(R)}$
$\frac{above}{(A)}$	$\frac{overlaps-above}{(O_A)}$	$\frac{contains}{(C_y)}$	$\frac{inside}{(I_y)}$	$\frac{overlaps-below}{(O_B)}$	$\frac{below}{(B)}$

Figure 1. ABLR relations [15]. X denotes the rectangle defining a region and the projections on each axis ((ox) and (oy)).

II. SPATIAL REPRESENTATION

To represent spatial knowledge, we use *SpaceOntology* [5] described in Figure 2.

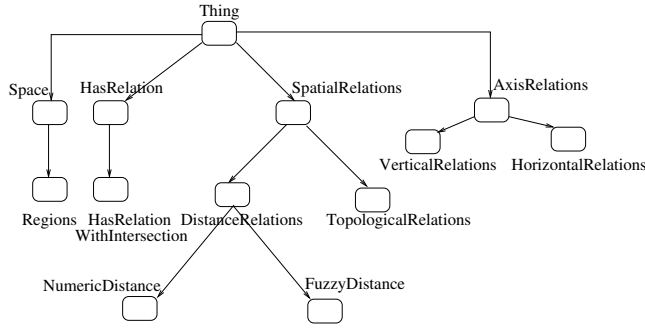


Figure 2. Graphical representation of *SpaceOntology*'s concepts.

The concept *Space* represents a global environment (a country, a city, a building, ...). The concept *Regions* is a sub-space included in a given space. A region is itself considered as a space that can be decomposed into different sub-regions. From a geometric point of view, a region is defined by the smallest rectangle corresponding to its axis-aligned bounding rectangles. A spatial relation is defined by the concept *SpatialRelations* which includes *TopologicalRelations* and *DistanceRelations* which could be numeric (for example, distance of 3 miles) or fuzzy (for example, close to a coffee machine). The details of the *SpaceOntology* definition are presented in [5].

In this paper, we focus on topological relations. In *SpaceOntology*, a topological relation is defined using the formalism ABLR depicted in Figure 1. Formally, a topological relation is a couple $\langle r_X, r_Y \rangle$, where: $r_X \in \{L, O_L, C_x, I_x, O_R, R\}$ and $r_Y \in \{A, O_A, C_y, I_y, O_B, B\}$.

SpaceOntology deduces new spatial knowledge from an initial description of space in order to complete unknown knowledge. *SpaceOntology* defines a set of rules translating the composition table of ABLR formalism in order to infer new topological relations. In general, the result of the inference mechanism gives a disjunction of topological relations, and even in some cases leads to the universal

relation. To use this new knowledge, it must be consistent with the initial knowledge.

However, to maintain spatial information consistency is made in a static way. In other words, anytime new topological relations are added/removed checking the consistency reconsiders the whole spatial description as if it were a new environment. Thus, *SpaceOntology* is not exploited effectively in real applications. In the next section, we propose a technique to maintain dynamically topological relations.

III. DYNAMIC QUALITATIVE SPATIAL REASONING

Our objective is the ability to deal with topological relations in a dynamic environment (add or remove topological relation) by adapting the technique presented in [8] for the topological relations managed in *SpaceOntology*. This technique concerns the dynamic constraint satisfaction in the case of temporal networks based on Allen's primitives. According to [8] a Dynamic Temporal Constraint Satisfaction Problem (DTCSP) is a sequence of static TCSPs (Temporal Constraint Satisfaction Problem [9]): $TCSP_0, TCSP_1, TCSP_2, \dots, TCSP_n$ each resulting from a change in the preceding one imposed by the outside world. More precisely $TCSP_{i+1}$ is obtained by performing a change on $TCSP_i$. The initial TCSP has an empty set of constraints. A particular case is when the initial constraint is equal to the disjunction of the 13 primitives. In this particular case, removing one or more Allen primitives from the universal relation is equivalent to adding a new constraint. The details of the constraint restriction and relaxation methods are presented in [8].

Similarly to [8], let us introduce the notions of *Spatial Constraint Satisfaction Problem* (SCSP) and *Dynamic Spatial Constraint Satisfaction Problem* (DSCSP).

A. SCSP and DSCSP

A Spatial Constraint Satisfaction Problem (SCSP) can be formalized as a directed graph called *spatial network*. It describes a set of variables (nodes in the spatial network) and a set of constraints (arcs in the spatial network). Each variable corresponds to a spatial entity defined in *SpaceOntology* (instances of the concepts *Regions* and *Space*).

Each constraint is a disjunction of the topological relations defined in *SpaceOntology*.

A Dynamic Spatial Constraint Satisfaction Problem (DSCSP) is a sequence of static SCSPs $P_0, P_1, P_{i+1}, \dots, P_n$ each resulting from a change of the preceding one through the addition or a retraction of a topological relation.

B. Dynamic Path Consistency Algorithm for Topological Relations

We present here an adaptation of the dynamic path consistency algorithm defined in [8]. The idea is to maintain the path consistency in the network at any time after adding or removing a disjunction of topological relations. The addition and the retraction of the topological relations defined in *SpaceOntology* is propagated to the entire spatial network using transitive closure based on the composition rules defined in *SpaceOntology*. In the case where the topological relations suppression results in an empty constraint, the spatial network is not consistent in this case (since it is not path consistent).

In the following, we focus on how to maintain dynamically the spatial consistency after adding inferred topological relations in the knowledge base of *SpaceOntology*.

The idea is to transform the initial topological knowledge described in *SpaceOntology* into a spatial network. The addition of inferred relations can be considered as a constraint restriction or a constraint relaxation [8].

1) *Constraint Restriction*: The constraint restriction is to add a new constraint (i.e a disjunction of topological relations) in a spatial network. The method for maintaining the consistency in this case is similar to the one proposed in [8].

Here, the topological relations we consider are presented on two axes. We denote by $C_{ij} = \langle r_{ij1}^H, r_{ij1}^V \rangle \cup \langle r_{ij2}^H, r_{ij2}^V \rangle, \dots, \langle r_{ijn}^H, r_{ijn}^V \rangle$ the constraint linking two vertices i and j (H for Horizontal and V for Vertical).

To adapt and optimize the algorithm described in [8], we decompose the constraint into two sub-sets of constraints according to the horizontal and vertical axes $decomp(C_{ij}) = \{C_{ij}^H, C_{ij}^V\}$ such as $C_{ij}^H = r_{ij1}^H \cup r_{ij2}^H \dots \cup r_{ijn}^H$ and $C_{ij}^V = r_{ij1}^V \cup r_{ij2}^V \dots \cup r_{ijn}^V$.

The method described by our Algorithm 1 above works as follows. Compute three intersections t^H , t^V and t . t^H is the intersection of the new sub constraint corresponding to the horizontal axis with the corresponding sub constraint, t^V is the intersection of the new sub constraint corresponding to the vertical axis with the corresponding sub constraint and t is the intersection of the new constraint with the corresponding constraint in the arc and path consistent graph. If the result of one of these intersections is an empty relation then the new constraint cannot be added otherwise it will violate the path consistency of the graph. Otherwise, we replace the current constraint of the graph by the result of the intersection. We then perform the dynamic

Algorithm 1 Restrict(i,j)

Require:

```

1:  $\{new\_c^H, new\_c^V\} \leftarrow decomp(new\_constraint)$ 
2:  $\{C_{ij}^H, C_{ij}^V\} \leftarrow decomp(C_{ij})$ 
3:  $t^H \leftarrow new\_c^H \cap C_{ij}^H$ 
4:  $t^V \leftarrow new\_c^V \cap C_{ij}^V$ 
5:  $t \leftarrow new\_constraint \cap C_{ij}$ 
6: if  $t^H = \emptyset$  or  $t^V = \emptyset$  or  $t \neq \emptyset$  then
7:   return "Constraint cannot be added"
8: else
9:    $C_{ij} \leftarrow new\_constraint$ 
10:   $update\_list \leftarrow \{new\_constraint\}$ 
11:  if  $\neg AdaptedPC(update\_list)$  or
      $\neg PC(update\_list)$  then
12:    return "Constraint cannot be added"
13:  end if
14: end if

```

path consistency (AdaptedPC 2 shown below and PC [8]) in order to propagate the update of the constraint to the rest of the graph. If the resulting graph is not path consistent then the new constraint cannot be added.

Algorithm 2 AdaptedPC

Require: $update_list$

```

1:  $PC \leftarrow false$ 
2:  $L \leftarrow update\_list$ 
3: while  $(L \neq \emptyset)$  do
4:   select and delete an  $(x, y)$  from  $L$ 
5:   for  $k \leftarrow 1$  to  $n, k \neq x$  and  $k \neq y$  do
6:      $t^H \leftarrow C_{xk}^H \cap C_{xy}^H \cdot C_{yk}^H$ 
7:      $t^V \leftarrow C_{xk}^V \cap C_{xy}^V \cdot C_{yk}^V$ 
8:     if  $t^H \neq C_{xk}^H$  or  $t^V \neq C_{xk}^V$  then
9:        $C_{xk}^H \leftarrow t^H, C_{xk}^V \leftarrow t^V$ 
10:       $C_{kx}^H \leftarrow INVERSE(t^H)$ 
11:       $C_{kx}^V \leftarrow INVERSE(t^V)$ 
12:       $L \leftarrow L \cup \{(x, k)\}$ 
13:       $updated\_list \leftarrow updated\_list \cup \{(x, k)\}$ 
14:       $t^H \leftarrow C_{ky}^H \cup C_{kx}^H \cdot C_{xy}^H, t^V \leftarrow C_{ky}^V \cup C_{kx}^V \cdot C_{xy}^V$ 
15:    end if
16:    if  $t^H \neq C_{ky}^H$  or  $t^V \neq C_{ky}^V$  then
17:       $C_{yk}^H \leftarrow INVERSE(t^H)$ 
18:       $C_{yk}^V \leftarrow INVERSE(t^V)$ 
19:       $L \leftarrow L \cup \{(k, y)\}$ 
20:       $updated\_list \leftarrow updated\_list \cup \{(y, k)\}$ 
21:    end if
22:  end for
23: end while

```

2) *Constraint Relaxation*: Similarly to the case of temporal relations in [8], the constraint relaxation corresponds to adding some topological relations to an existing constraint.

Run the dynamic path consistency algorithm starting from the triangle containing the relaxed constraint. If at least one of the other two relations belonging to the triangle is updated, we propagate in a recursive way this change to the other constraints.

IV. CASE STUDY

Let us consider the environment depicted in Figure 3.

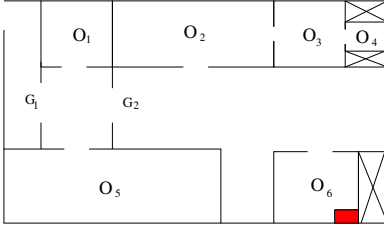


Figure 3. A Scene

There is a set of regions; offices and gates $\{O_1, O_2, \dots, O_6, G_1, G_2\}$ and the following set of topological relations between them:

- $O_1 < L, I_y > O_2$, $O_1 < L, C_y > O_4$, $O_1 < I_x, A > O_5$;
- $O_3 < R, I_y > O_2$, $O_3 < L, C_y > O_4$;
- $O_4 < O_R, A > O_6$;
- $O_5 < L, C_y > O_6$.

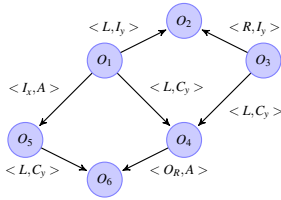


Figure 4. Spatial Network

Figure 4 shows the spatial network formalizing all the known spatial relations. We consider this network as the initial spatial state of the world.

Let us assume that, after an earthquake at a nuclear plant, a robot is sent in it in order to do some emergency tasks. The robot is guided by a human who has partial knowledge of the environment. The human implements *SpaceOntology* by the following informations.

The robot mission is to isolate the region in which the office O_6 is located. For this, it must block gateways such as G_1 and G_2 . However, with partial spatial knowledge the robot cannot make a decision. Based on the initial spatial description, we can define the space described by the graph in the upper left of Figure 5. Now, by exploiting the composition rules defined in *SpaceOntology* [5], we deduce the topological relation between O_1 and O_6 from

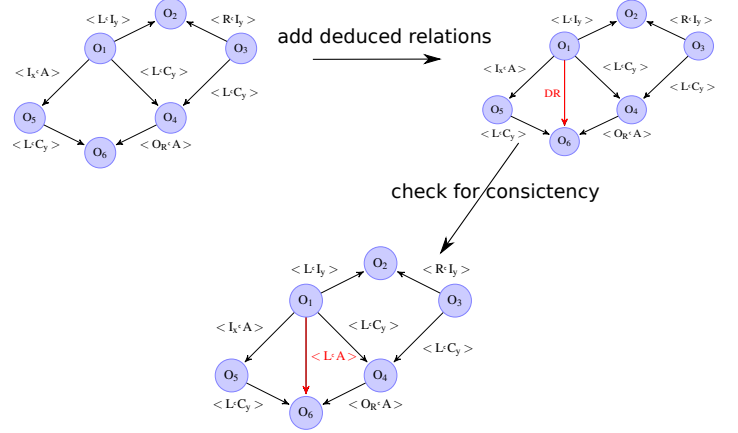


Figure 5. A scene

the composition between the couples (O_1, O_4) and (O_4, O_6) , we deduce that the relation between O_1 and O_6 is DR , where $DR = < L, C_y > \vee < L, A > \vee < L, O_A > \vee < L, I_y > \vee < O_L, C_y > \vee < O_L, A > \vee < O_L, O_A > \vee < O_L, I_y > \vee < I_x, C_y > \vee < I_x, A > \vee < I_x, O_A > \vee < I_x, I_y >$.

Therefore, the result of the composition is a disjunction that cannot be exploited to make a decision. In this disjunction some information are inconsistent. When relying on our approach we can check the consistency of this knowledge without any reformulation.

To check the consistency of topological knowledge, we proceed as follows. We add DR between O_1 and O_6 and compute the intersection between $< L, A >$ and DR . This intersection returns the following result: $< L, A >$. This latter relation is not in conflict with the network.

V. EXPERIMENTATION

This section presents comparative experimental tests evaluating the performance in time of a static method and our dynamic technique to maintain the consistency of the spatial knowledge that evolves over time. The static method enforces the path consistency from scratch anytime a constraint is added.

We have conducted two categories of experimental tests. The first category is performed on nine instantiations of *SpaceOntology*. The second one is conducted on randomly generated consistent DSCSPs. All the procedures are coded in Java.

A. Tests Performed on Instantiations of *SpaceOntology*

Each instance has a specific number of regions and topological relations as shown in Table I. For each instance new topological relations are inferred in a random order.

For each instance of *SpaceOntology* we formalize the initial spatial knowledge in an initial SCSP where the nodes are regions and arcs are disjunctions of topological relations.

Table I
STATIC METHOD VERSUS DYNAMIC METHOD

Instances	○	◇	□	△
1	15	120	562.1	210.03
2	25	98	328.5	165.2
3	35	80	281.6	98.72
4	45	72	182.9	70.32
5	55	51	90.1	62.01
6	65	38	80.9	51.07
7	75	20	60.7	42.98
8	85	9	42.1	27.02
9	95	3	12.1	10.03

○ number of regions and topological relations
◇ number of new topological relations added
□ static method
△ dynamic method

Here we compare the static and dynamic methods performances to maintain the path consistency in the network after adding inferred topological relations. The inferred relations are deduced by applying the composition rules defined in *SpaceOntology*. The results of the experimental tests are depicted in Table I. This table shows that maintaining consistency after adding a large number of topological relations (over 50 relationships) is faster with the dynamic method than with the static one. For example, if we add 120 topological relations, the dynamic method runs in 210.03 ms, instead of 562 ms for the static method.

B. Tests on Randomly Generated DSCSPs

The criteria used to evaluate the static and dynamic methods is the running time needed to maintain the consistency of the DSCSP. To generate a consistent SCSP we first randomly generate a numeric solution and then randomly add other information to it. For this, we consider the node representing a region defined by a rectangle. This latter is identified by a couple of intervals ($R = [(b_{inf}^H, b_{sup}^H), (b_{inf}^V, b_{sup}^V)]$).

Following the generation method described in [9], consistent SCSPs are randomly generated as follows.

- 1) Generation of numeric solution. Randomly pick N pairs (I, J) where $I = (b_{inf}^H, b_{sup}^H)$ and $J = (b_{inf}^V, b_{sup}^V)$ are pairs of integers such that $b_{inf}^H < b_{sup}^H$ and $b_{inf}^V < b_{sup}^V$ and $b_{inf}^H, b_{sup}^H, b_{inf}^V, b_{sup}^V \in [0, \dots, Horizon]$. This set of N pairs forms the initial solution where each couple of pairs corresponds to the projection of the rectangle on x-axis and y-axis. Horizon corresponds to a max number.
- 2) Generation of numeric constraints. For each couple (I, J) , randomly pick an interval contained within $[0, \dots, Horizon]$ and containing the couple (I, J) . This newly generated couple defines the projection of rectangle on x-axis and y-axis. From these projections, we generate the domain of the corresponding regions.
- 3) Generation of constraints. Compute the basic ABLR primitives that hold between each projection pair of

the initial solution.

Example 1: Let us assume we want to generate a consistent SCSP with $N = 3$ regions and $Horizon = 10$.

- 1) A numeric solution is generated: $S = \{([1, 4], [2, 5]), ([2, 8], [2, 6]), ([5, 7], [1, 7])\}$
- 2) Domains for the different regions are randomly generated from the numeric solution as shown in Table II.

Table II
DOMAINS DEFINITION

Projections	Domains
[1,4][2,5]	{[0, 3], ..., [4, 7], [1, 4], ..., [3, 6]}
[2,8][2,6]	{[1, 7], ..., [3, 9], [0, 4], ..., [4, 8]}
[5,7][4,7]	{[4, 6], ..., [7, 9], [0, 3], ..., [7, 10]}

- 3) ABLR primitives are computed from the pairs of projections as described in Table III.

Table III
RELATIONS DEFINITION

Pairs of Projections	Topological relations
[1,4][2,5] and [2,8][2,6] \rightarrow	$\langle O_L, O_B \rangle$
[1,4][2,5] and [5,7][4,7] \rightarrow	$\langle L, I_y \rangle$
[2,8][2,6] and [5,7][4,7] \rightarrow	$\langle C_x, I_y \rangle$

After generating the SCSP, the solving algorithm will process the list of spatial relations in an incremental way (in order to simulate a DSCSP). We start with a DSCSP having N variables and 0 constraints. Constraints are then added one by one, in arbitrary order from the randomly generated SCSP to the DSCSP until a given number of constraints C , is reached. After adding each constraint, the solving algorithm will check the path consistency of the new DSCSP.

Table IV
COMPARATIVE TESTS ON RANDOMLY GENERATED DSCSPs

N	Problem		Dynamic	Static
	C	D		
20	95	50	0.30	0.15
40	390	50	0.57	0.43
60	885	50	0.95	1.11
80	1580	50	2.56	3.02
100	2475	50	20.20	45.09
200	9950	100	157.10	200.98
300	22425	100	320.91	600.09

Table IV presents the results of tests performed on DSCSP instances defined by the number of variables N , their domain size and the number of constraints C . The running time of the two methods are comparable for small and medium problems ($N \leq 80$). For large problems, the dynamic method is more efficient.

For those real world applications such as real time systems where a solution needs to be returned within a given deadline

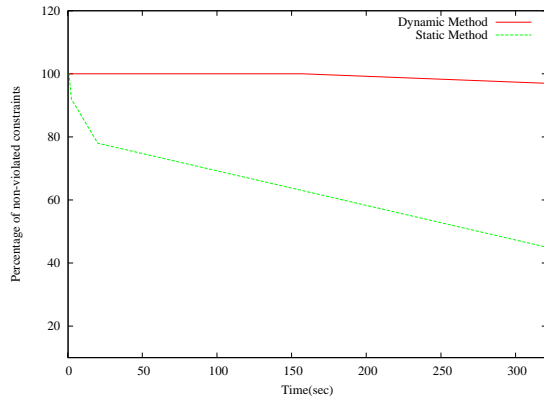


Figure 6. Solution quality given by the static and dynamic methods.

an alternative is to try to solve most of the constraints (instead of all of them). For this purpose, path consistency can be used as an incomplete method that returns a solution with a quality corresponding to the percentage of satisfied constraints.

For this purpose, we conducted tests where we randomly generate DSCSPs with a number of constraints ranging from 100 to 25000. Here, we compute the percentage of non violated constraints (quality of the solution) obtained at different time instants after applying both the static and dynamic methods.

Figure 6 presents the results of these experiments. In these tests we focus on the quality of the solution. While we compute the running time needed to return the solutions, we also check the percentage of non-violated constraints. We note that, for every time instant, the solutions returned by the dynamic method are of better quality than those returned by the static method. For example, for a computation time of 300 seconds the dynamic method was able to satisfy 97% of the constraints while the static method was successful for solving only 45%.

VI. CONCLUSION

In this paper we have proposed an approach to maintain the consistency of spatial knowledge in a dynamic environment. This approach is interest for many application where qualitative spatial information need to be managed in a dynamic environment. To our best knowledge no work in spatial reasoning has addressed this issue.

Our proposed method can be extended to the other formalisms for representing the topological relations and based on Allen's algebra such as rectangles algebra [16] as well as the method of n-intersections [11]. In this paper, we only use the topological aspect proposed by *SpaceOntology*. In the near future we plan to manage all aspects of spatial knowledge in a dynamic manner. We will also explore the case of retracting topological relations. The challenge here

is to restore back all the values that have been removed due the constraint we are relaxing.

REFERENCES

- [1] D. Hernández, *Qualitative Representation of Spatial Knowledge*, Springer, 1994.
- [2] G.Ligozat, *ESPACE ET TEMPS: Un nouveau domaine en Sciences Cognitives?*, 1998.
- [3] J. Lesbegueries, P. Loustau, *Structuration d'information spatiale qualitative pour la recherche d'information*, Relation, 2006.
- [4] J.F. Allen, *Maintaining knowledge about temporal intervals*, Vol.26(11), pages 832-843, Communications of the ACM, 1983.
- [5] L. Belouaer, M. Bouzid, A.I. Mouaddib, *Ontology Based Spatial Planning for Human-Robot Interaction*, Temporal Representation and Reasoning (TIME), 2010 17th International Symposium on, 2010.
- [6] L. Vieu, *Spatial representation and reasoning in artificial intelligence*, Spatial and temporal reasoning, 1997.
- [7] P. Balbiani, J.F. Condotta, L.F. del Cerro, *A model for reasoning about bi-dimensional temporal relations*, KR, pages 124-130, 1998.
- [8] M. Mouhoub, *Dynamic Path Consistency for Interval-based Temporal Reasoning*, Applied Informatics, pages 393-398, 2003.
- [9] M. Mouhoub, *Systematic versus non systematic techniques for solving temporal constraints in a dynamic environment*, AI Communications, IOS Press, Vol. 17(4), pages 201-211, 2004.
- [10] M.J. Egenhofer, D.M. Mark, A.U. Frank, W. Kuhn, *Spatial information theory: A theoretical basis for GIS*, International Conference COSIT, 1995.
- [11] M.J. Egenhofer, R.D. Franzosa, *Point-set topological spatial relations*, International Journal of Geographical Information Science, 1991.
- [12] C. Likitvivanavong, Y. Zhang, J. Bowen, E. Freuder, *Arc consistency in MAC: a new perspective Proceedings of CPAI*, Citeseer, 2004.
- [13] C. Bessière, J.C. Régis, R. Yap, Y. Zhang, *An optimal coarse-grained arc consistency algorithm Artificial Intelligence*, Elsevier, 165(2), pages 165-185, 2005.
- [14] M. Dib, R. Abdallah, A. Caminada, *Arc-consistency in Constraint Satisfaction Problems: A survey*, 2nd International Conference on Computational Intelligence, Modeling and Simulation Bali, Indonesia, 2010.
- [15] S. Laborie J. Euzenat N. Layaida, *Adaptation spatiale efficace de documents SMIL*, 5e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA), 2006.
- [16] P. Balbiani, J. Condotta, L. Fari as del Cerro, *A new tractable subclass of the rectangle algebra*, International Joint Conference on Artificial intelligence, 1999, 16, 442-447.