

Solving Geometric Constraint Satisfaction Problems for Spatial Planning*

Franck du Verdier
Alcatel Alsthom Recherche
route de Nozay
91460 Marcoussis
FRANCE

Email: verdier@aar.alcatel-alsthom.fr

Abstract

Like many important problems in artificial intelligence and operations research, spatial planning problems can be viewed as constraint satisfaction problems (CSP). Many techniques, such as consistency algorithms, have been designed to solve CSPs in which variables take on values from one-dimensional domains. The fact that spatial planning leads to multi-dimensional domains increases the combinatorial complexity. In this paper, we propose a new approach to solve geometric constraint satisfaction problems. A uniform symbolic representation of different geometric constraints, based on the configuration space approach, is defined in order to achieve conjunctions of constraints bearing on the same objects and arc-consistency. The geometric focus of our approach allows a rich expressivity. In terms of complexity, it takes advantage of the compactness of intentional geometric representations, with the same idea as consistency do over intervals. The proposed principles have been implemented in the ATLAS spatial planning system, and tested in the field of microwave electronic equipment design.

1 Introduction

In the face of growing importance of industrial applications in electronics, mechanics, architecture, town planning, etc. layout automation has spurred considerable interest. A spatial planning problem consists of finding a layout of a set of objects that satisfies a set of constraints among these objects.

Like many important problems in artificial intelligence and operations research, spatial planning can be viewed as a Constraint Satisfaction Problem (CSP). A CSP is defined by a finite set of variables taking on values from finite domains, and a set of constraints among these variables. In spatial planning problems, the variables are the

objects, the domains are the configuration sets (orientations + positions) of the objects, and the constraints are the geometric relations among the objects. We call such a problem a Geometric Constraint Satisfaction Problem (GCSP).

A solution to a CSP is an assignment of values to variables satisfying all constraints, and the problem is to find one or all solutions. Many improvements of the backtracking search algorithm [Haralick and Elliott, 1980], preprocessing techniques [Dechter and Pearl, 1989] and consistency algorithms [Montanari, 1974; Mackworth, 1977; Mohr and Henderson, 1986] have been designed to solve CSPs. In particular, consistency algorithms reduce the search space by removing, from the domains and constraints, values that cannot be part of a solution. Arc-consistency algorithms work on binary CSPs, and make sure that constraints are individually consistent [Mackworth, 1977]. AC-4 is an optimal arc-consistency algorithm [Mohr and Henderson, 1986] for general constraints over finite domains. Other arc-consistency algorithms have been designed to exploit internal structure of domains such as interval domains [Davis, 1987], hierarchical structured domains [Mackworth *et al.*, 1985] or specific constraints such as functional and monotonic constraints [Deville and Van Hentenryck, 1991].

In this paper, we propose a new approach to solve geometric constraint satisfaction problems. First, we represent the position set of an object in a given orientation as one multi-dimensional domain (2D or 3D). Second, we propose a uniform symbolic representation of different geometric constraints in order to make conjunctions of constraints bearing on the same objects. Third, to achieve arc-consistency, we take advantage of the ordered structure of the domains, in the same way as specific consistency algorithms do over intervals. All these principles have been implemented in the ATLAS spatial planning system, and tested in the field of microwave electronic equipment design.

Section 2 presents the representations chosen for variables and constraints. Section 3 describes the solution of the GCSP: conjunction and consistency of constraints. Section 4 reports the implementation of the proposed approach in ATLAS. Section 5 proposes a comparison with other works, and a discussion of the interest and the application of our approach. Conclusions are given in section 6.

*This work was supported in part by the National Agency of Research and Technology (ANRT), and in part by Alcatel Alsthom Recherche, in collaboration with the Claude-Bernard University of Lyon.

2 Representations

In this paragraph, we present the representations chosen for variables and constraints.

2.1 Representation of Variables

The representation of variables is based on the following hypothesis: the possible orientations of the objects are discrete and in a limited number. For example, the objects may have orientation 0 , $\pi/2$, π or $3\pi/2$. This hypothesis is not restrictive for a large variety of spatial planning applications.

The configuration set of an object (its domain) is represented by the cartesian product of k position sets, where k is the number of possible orientations. For example, in a 2D spatial planning problem with 4 possible orientations (0 , $\pi/2$, π or $3\pi/2$), figure 1 shows the decomposition of the domain D_i of an object i in several position sets $D_i(\alpha)$, for each orientation α of i . Each position set is represented by a multi-dimensional space (2D or 3D according to the application). Note that the domain representation introduces two levels: the orientation level and the position level. The position set and the object form have therefore the same topological dimension, used the same representation resulting in easier geometric operations between them.

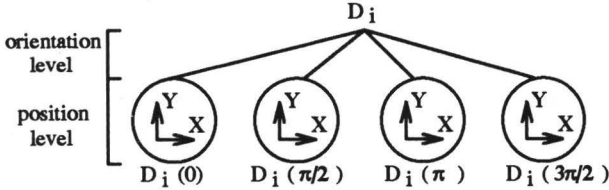


Figure 1: Representation of a configuration set.

We represent the position set of an object for a given orientation by the positions of its reference point. This representation is based on the configuration space approach used in particular in robotics for motion planning [Lozano-Perez, 1983]. Indeed, the configuration set of an object is a non injective view of the configuration set of its reference point. Note that the configuration set of an object is given by the geometric sum (Minkowski sum¹) of the configuration set of its reference point and its form. To illustrate the non injective nature of this transformation, figure 2 shows, for a given orientation of an object O , two different domains of its reference point $Dom(o(O))$ and $Dom'(o(O))$, corresponding to the same domain $Dom(O)$ of O .

Geometric constraints, such as “no-overlap”, are at the origin of the removal of values situated at the center of the domain rather than those on the boundaries. Therefore, we don’t restrict the removal of inconsistent configurations on domain boundaries, but make possible the removal of any inconsistent configuration. Note that a position set of an object may be non connected (i.e. may consist of several disjoint parts).

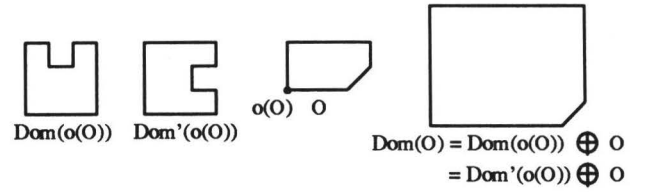


Figure 2: Example showing the non injective nature of the transformation of the configuration set of the reference point to the configuration set of the object.

2.2 Representation of Constraints

Constraints express relationships among objects. They come either from the problem specifications or from the problem solving know-how. The constraints considered here are restricted to being unary or binary. Two types are distinguished: orientation constraints and position constraints.

2.2.1 Representation of Orientation Constraints

Orientation constraints are restrictions on the orientations of one or two objects. They are represented explicitly by their relation graph. For example, a “same-orient” constraint between two objects is represented by the following set: $\{0, \pi/2, \pi \text{ or } 3\pi/2\}$.

2.2.2 Representation of Position Constraints

Position constraints are restrictions on the positions of one or two objects. They are represented implicitly by symbolic relations between two regions (“no-overlap”, “contains”, “min-distance”...). A region is a geometric entity (point, line, surface, volume) which position is defined relatively to the layout space or to the reference point of an object. For example, the form of an object is a particular region of it, like its special points. Symbolic relations considered here are the same for unary and binary constraints; a unary position constraint is a relation between a region of an object and a region of the layout space; a binary position constraint is a relation between two regions of two objects.

We take advantage of the configuration space approach [Lozano-Perez, 1983] in order to express different symbolic relations between two regions, with a uniform “belong” relation between a reference point and a region. Let $C_{i,j}$ be a position constraint between two regions R_i and R_j of the objects i and j . For example, $C_{i,j}$ may be a “no-overlap” constraint between the forms R_i and R_j of the objects i and j . We express $C_{i,j}$ with a “belong” relation between the reference point $o(i)$ of the object i and a region which is given by the deformation of the region R_j in accordance with the region R_i . Table 1 presents the uniform “belong” relation of different symbolic relation of position constraints. $R_j \oplus R_i$ denotes the geometric sum between R_j and R_i , $-R_i$ denotes the symmetric of R_i , $\overline{R_i}$ denotes the complementary of R_i and $Disk(d)$ denotes the disk of radius d . The proof of this result is given in appendix A. In [du Verdier, 1992], we give the expressions of other position constraints, such as “alignment”.

¹ $A \oplus B = \{a + b \mid a \in A, b \in B\}$

$overlap(R_i, R_j)$	$\Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$
$no-overlap(R_i, R_j)$	$\Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$
$contained-in(R_i, R_j)$	$\Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$
$contains(R_i, R_j)$	$\Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$
$max-dist(R_i, R_j, d)$	$\Leftrightarrow o(i) \in R_j \oplus Disk(d) \oplus -R_i$
$min-dist(R_i, R_j, d)$	$\Leftrightarrow o(i) \in R_j \oplus Disk(d) \oplus -R_i$

Table 1: Expression of position constraints with a uniform “belong” relation.

Note that the form of the region which is introduced in the “belong” relation depends on the object orientations. Therefore, the “belong” relation of a unary position constraint introduces as many regions as possible orientations; for binary position constraint, it introduces as many regions as possible pairs of orientations.

3 Solution

Before presenting the consistency algorithms, we describe the conjunctions of constraints bearing on the same objects, in order to improve the performance and to guarantee that there is at most one binary constraint between two variables.

3.1 Conjunction of Constraints

A significant improvement of the propagation process in a CSP consists of substituting constraints bearing on the same variables for a single constraint, conjunction of the initial constraints [Montanari, 1974]. First, because of a reduction of the back-and-forth propagation process between the same variables, inconsistent values will be detected earlier. Second, local consistency over the conjunction constraint may detect more inconsistent values than the same local consistency over the initial ones; for example, in figure 3, the detection that the value b of the variable i is inconsistent requires a path-consistency for the initial CSP (a), and only an arc-consistency for the CSP after conjunction (b).

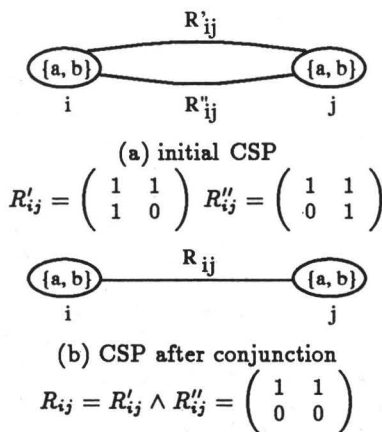


Figure 3: Example of the conjunction of two constraints bearing on the same variables.

Sections 3.1.1 and 3.1.2 describe the conjunction of

orientation and position constraints according to their proposed representation in the GCSP.

3.1.1 Conjunction of Orientation Constraints

The conjunction of orientation constraints bearing on the same variables is given by an intersection of the relations, since their representation is explicit.

3.1.2 Conjunction of Position Constraints

The conjunction of position constraints bearing on the same variables is given by a geometric intersection of the concerned regions, since they are expressed uniformly with a “belong” relation. Figure 4 gives an example of conjunction of a “no-overlap” constraint C between two regions R_i and R_j of two objects i and j and a “max-distance” constraint between the points p_i of i and p_j of j :

$$C' = no-overlap(R_i, R_j) \Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$$

$$C'' = max-dist(R_i, R_j, d) \Leftrightarrow o(i) \in R_j \oplus Disk(d) \oplus -R_i$$

$$C = C' \wedge C'' \Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i \cap R_j \oplus Disk(d) \oplus -R_i$$

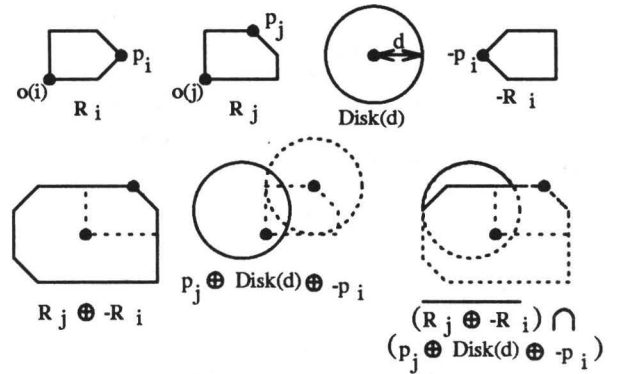


Figure 4: Example of conjunction of a “no-overlap” constraint and a “max-distance” constraint.

3.2 Consistency

For complexity reasons (time and space), we limit the degree of consistency to 2 with an arc-consistency procedure. Note that arc-consistency doesn’t achieve the complete consistency of a CSP, and that some inconsistent values may still remain in an arc-consistent CSP.

An arc-consistency algorithm is a symbolic relaxation algorithm to achieve node-consistency² over each variable and arc-consistency³ over each binary constraint. Then, the propagation process consists of only reconsidering a binary constraint between two variables i and j after a deletion from the domain D_j of j .

The arc-consistency procedure of the GCSP applies this principle, except that it operates on two kinds of constraints (orientation constraints and position constraints), and that it takes advantage of the domain representation introduced in 2.1. First, consistency of an orientation constraint operates at the orientation level of

² A variable i is node-consistent iff each value of D_i satisfies the unary constraints on i .

³ A binary constraint $C_{i,j}$ is arc-consistent iff for each value x of D_i there is at least one value y in D_j such that $C_{i,j}(x, y)$ is satisfied.

the domain representation of the objects. Second, propagation on an orientation constraint occurs only when all configurations of the position set of an orientation are removed. Therefore, propagation on a constraint concerns a given orientation.

For orientation constraints, local consistency of one node or one arc is achieved by usual algorithms ("NC" and "REVISE" procedures [Mackworth, 1977]), since the representation of these constraints is explicit. Concerning local consistency of position constraints, we propose a new specific method which is the essence of our contribution. Our approach is based on the configuration space approach [Lozano-Perez, 1983], and uses the proposed uniform symbolic representation of position constraints. The idea is to take advantage of the ordered structure of the domains, in the same way as consistency algorithms do over intervals. We present this original approach of consistency of position constraints and gives some remarks about complexity. We give the complete arc-consistency algorithm of the GCSP and analyse its complexity in [du Verdier, 1992].

3.2.1 Node-Consistency of Position Constraints

A unary position constraint limits possible configurations of an object. These configurations are directly given by the region of the layout space which is introduced by the uniform "belong" relation of the constraint. Indeed, this region defines the positions of the reference point of the object satisfying the constraint. Therefore, for a given orientation α_i , the consistency of a unary position constraint C_i on a region R_i of an object i , consists of a geometric intersection between the corresponding position set $D_i(\alpha_i)$ and the region $Region(R_i(\alpha_i))$ of the uniform "belong" relation of C_i .

Consistency of a unary position constraint
 $C_i(R_i) \Leftrightarrow o(i) \in Region(R_i)$
 for a given orientation α_i of an object i :
 $D_i(\alpha_i) \leftarrow D_i(\alpha_i) \cap Region(R_i(\alpha_i))$

3.2.2 Arc-Consistency of Position Constraints

Concerning a binary position constraint, its uniform "belong" relation introduces, for each pair of orientations, an imposed placement region for the reference point of the first object of the relation. As this region is defined wrt. to the reference point of the second object of the relation, it may "move" around the layout space in accordance with the positions of this second object. Therefore, concerning the consistency of a binary position constraint $C_{i,j}$ between the regions R_i and R_j of the objects i and j , for each pair of orientations of i and j , the position set D_i of $o(i)$ satisfying $C_{i,j}$ is given by the geometric sum between the position set D_j of $o(j)$ and the region $Region(R_i, R_j)$ which is introduced by the uniform "belong" relation of $C_{i,j}$.

Consistency of a binary position constraint
 $C_{i,j}(R_i, R_j) \Leftrightarrow o(i) \in Region(R_i, R_j)$:
 $D_i \leftarrow D_i \cap (D_j \oplus Region(R_i, R_j))$

Figure 5 shows an example of the consistency of a "no-overlap" constraint between the regions R_i and R_j of two objects i and j with domains D_i and D_j . The "belong"

relation of this "no-overlap" constraint introduces the region $R_j \oplus -R_i$. The positions of $o(i)$ satisfying the constraint are given by the geometric sum between the possible positions D_j of $o(j)$ and the region $R_j \oplus -R_i$. Then, the positions D_i of $o(i)$ which are not in the region $D_j \oplus R_j \oplus -R_i$ are removed from D_i by a geometric intersection.

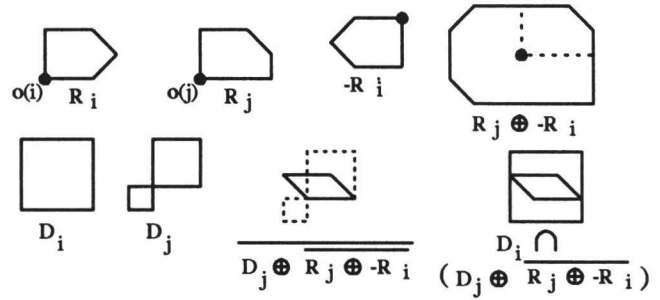


Figure 5: Example of the consistency of a "no-overlap" constraint.

This result is valid only for a given pair of orientations. In the general case, the consistency of a binary position constraint, for a given orientation of the first object of the relation, leads to a disjunction on the orientations of the second object. Instead of testing all pairs of possible orientations, it is better to test only those satisfying binary orientation constraints among the considered objects, in order to improve the efficiency of the consistency achievement in terms of both time and quality. Note that the motivation is the same for the conjunction of constraints bearing on the same variables (See section 3.1). We introduce the predicate $R-Orient(i, j, \alpha_i, \alpha_j)$, which is true for the pairs of orientations (α_i, α_j) of the objects i and j satisfying the binary orientation constraints between i and j .

Consistency of a binary position constraint
 $C_{i,j}(R_i, R_j) \Leftrightarrow o(i) \in Region(R_i, R_j)$
 for a given orientation α_i of an object i :
 $D_i(\alpha_i) \leftarrow D_i(\alpha_i) \cap \bigcup (D_j(\alpha_j) \oplus Region(R_i(\alpha_i), R_j(\alpha_j)))$
 forall α_j such that $R-Orient(i, j, \alpha_i, \alpha_j)$

3.2.3 Consistency Complexity of Position Constraints

Concerning position constraints, node-consistency of an object in a given orientation consists of only one geometric intersection, and arc-consistency of a position constraint between two objects in given orientations consists of only one geometric sum and one geometric intersection. The complexity of such geometric operations depends on the geometric representation of the regions, and on their geometric properties. For example, the geometric sum or the intersection of two convex polygons with p and q edges respectively can be computed in $O(p+q)$ time [Preparata and Shamos, 1985].

With usual consistency techniques over finite domains, the consistency of a position constraint C would consist of checking C for every possible position of the concerned objects. In opposition to our approach, such usual consistency techniques can't exploit the compactness of in-

tensional geometric representations for the position set of the object.

4 Implementation

The proposed GCSP has been implemented in the ATLAS spatial planning system. Figure 6 shows the general architecture of ATLAS. The GCSP which has been presented in this paper constitutes the kernel of ATLAS. It updates the domain of objects by consistency. To execute geometric operations, it calls a geometric representation module. For considerations about the application in particular, a geometric representation by quadrees has been implemented [Samet, 1989]. At the beginning of the solution, objects and constraints come from the problem specifications. Then, a rule-based system proposes to add constraints in order to reduce the solutions space. When a conflict arises (i.e. when the problem is over-constrained), a failure handling method applies constraint substitutions in order to solve it. More information about ATLAS may be found in [du Verdier and Tsang, 1991; du Verdier, 1992].

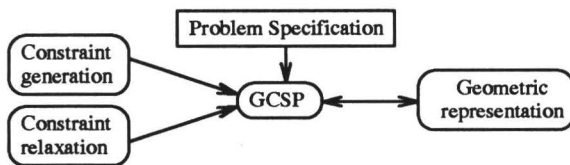


Figure 6: General architecture of ATLAS spatial planning system.

Although ATLAS has been tested in the field of microwave electronic equipment design, it is able to solve many spatial planning problems in 2D applications where each object may have 4 possible orientations (0, $\pi/2$, π or $3\pi/2$) and 2 possible forms.

To compare ATLAS with other spatial planning systems, it has been experimented on common theoretical problems. In terms of search efficiency, results substantiate the consistency approach for spatial planning problems, and show that the more the problem is constrained, the more filtering before backtracking is relevant. Moreover, in comparison with other spatial planning systems which use constraint satisfaction techniques, such as WRIGHT [Baykan and Fox, 1987], ATLAS requires less intermediate states to solve the same problems because the pruning is more effective (See section 5.1).

5 Discussion

5.1 Comparison with Other Works

Several spatial planning systems have used constraint satisfaction techniques in different application domains: WRIGHT for kitchen design [Baykan and Fox, 1987], PROTEAN to identify the 3D conformations of proteins [Hayes-Roth *et al.*, 1986] and SIGHT-PLAN for layout of construction sites [Tommelein, 1989]. These systems represent the configuration set of an object as the cartesian product of one-dimensional domains (orientations

+ coordinates) in order to use basic consistency algorithms over intervals. This type of representation is limited to parallelepipedic forms (rectangles in 2D), which thereby restricts the removal of inconsistent configurations. Moreover, a geometric constraint is defined by a disjunction of one-dimensional constraints. For example, a "no-overlap" constraint is expressed with a disjunction of "above", "under", "left-of" and "right-of" constraints.

The proposed GCSP avoids these limitations. The geometric focus of our approach allows a rich expressivity. First, we are able to solve spatial planning problems with non rectangular forms. Second, since we represent non rectangular domains, more inconsistent values may be removed; therefore, pruning is more effective and the solution requires less intermediate states.

5.2 The Geometric Representation of Regions

Although in ATLAS a geometric representation by quadrees has been implemented, the proposed approach imposes no particular geometric representations, and it can be applied for any D applications (1D, 2D, 3D...). Discrete representations as well as continuous ones are available. However, to guarantee the convergence of the consistency algorithms, note that a continuous geometric representation leads us to introduce a threshold for domain reductions, under which reductions become insignificant, and the propagation process stops. A particular geometric representation has to be chosen in accordance with geometric specifications of the problem and as well as with the efficiency of the following geometric algorithms: union, intersection, complementary, geometric sum and symmetric.

6 Conclusions

A new representation for GCSP and its solution have been described. A uniform symbolic representation of position constraints, based on the configuration space approach [Lozano-Perez, 1983], is proposed to achieve conjunctions of constraints bearing on the same objects, and arc-consistency. The geometric focus of our approach allows a rich expressivity. In terms of complexity, it takes advantage of the compactness of intentional geometric representations, with the same idea as consistency do over intervals.

Specific consistency algorithms, in opposition to general ones over finite domains [Mackworth, 1977; Mohr and Henderson, 1986], exploit the internal structure of the domains of the variables (interval domains [Davis, 1987], hierarchical structured domains [Mackworth *et al.*, 1985]) or the specificity of the constraints (functional and monotonic constraints [Deville and Van Hentenryck, 1991]). Our work contributes to this field of research in the case of geometric constraints.

Future research on this topic includes the solution of GCSPs with regions of variable forms. For example, in kitchen design, a furniture may have several possible forms. The idea is to consider region forms as additional variables of the GCSP with domains of possible values. The first difficulty consists of representing domains of possible forms, and the second one consists of taking

into account such additional variables in the solution of the GCSP.

Acknowledgments

I would like to thank particularly Jean-Patrick Tsang and Simone Pimont who helped me shape these ideas, and for their useful comments on this paper.

A Expression of position constraints with a uniform "belong" relation

Intersection Constraints: A region R_i overlaps a region R_j if the reference point $o(i)$ of i belongs to the configuration space obstacles of R_j due to R_i , denoted $CO_{R_i}(R_j)$ [Lozano-Perez, 1983]:

$overlap(R_i, R_j)$	$\Leftrightarrow o(i) \in CO_{R_i}(R_j)$
	$\Leftrightarrow o(i) \in R_j \oplus -R_i$
$no-overlap(R_i, R_j)$	$\Leftrightarrow \neg overlap(R_i, R_j)$
	$\Leftrightarrow o(i) \in \overline{CO_{R_i}(R_j)}$
	$\Leftrightarrow o(i) \in R_j \oplus -R_i$

Include Constraints: A region R_i is contained in a region R_j if the reference point $o(i)$ of i belongs to the configuration space interior of R_j due to R_i , denoted $CI_{R_i}(R_j)$ [Lozano-Perez, 1983]:

$contained-in(R_i, R_j)$	$\Leftrightarrow o(i) \in CI_{R_i}(R_j)$
	$\Leftrightarrow o(i) \in \overline{R_j} \oplus -R_i$
$contains(R_i, R_j)$	$\Leftrightarrow contained-in(\overline{R_i}, \overline{R_j})$
	$\Leftrightarrow o(i) \in \overline{CI_{\overline{R_i}}(\overline{R_j})}$
	$\Leftrightarrow o(i) \in R_j \oplus -R_i$

Distance Constraints A region R_i is at a minimal distance d to a region R_j if R_i doesn't overlap a region given by an expansion of R_j . For an euclidean distance, this expansion is the geometric sum of R_j and a disc which radius is d , denoted $Disk(d)$:

$min-dist(R_i, R_j, d)$	$\Leftrightarrow no-overlap(R_i, R_j \oplus Disk(d))$
	$\Leftrightarrow o(i) \in \overline{CO_{R_i}(R_j \oplus Disk(d))}$
	$\Leftrightarrow o(i) \in R_j \oplus Disk(d) \oplus -R_i$
$max-dist(R_i, R_j, d)$	$\Leftrightarrow \neg min-dist(R_i, R_j, d)$
	$\Leftrightarrow overlap(R_i, R_j \oplus Disk(d))$
	$\Leftrightarrow o(i) \in CO_{R_i}(R_j \oplus Disk(d))$
	$\Leftrightarrow o(i) \in R_j \oplus Disk(d) \oplus -R_i$

References

- [Baykan and Fox, 1987] Can A. Baykan and Mark S. Fox. An investigation of opportunistic constraint satisfaction in space planning. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1035-1038, Milan, Italy, August 1987.
- [Davis, 1987] Ernest Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32:281-331, 1987.
- [Dechter and Pearl, 1989] Rina Dechter and Judea Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, 38:353-366, 1989.
- [Deville and Van Hentenryck, 1991] Yves Deville and Pascal Van Hentenryck. An efficient arc consistency algorithm for a class of CSP problems. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 325-330, Sydney, Australia, August 1991.
- [du Verdier and Tsang, 1991] Franck du Verdier and Jean-Patrick Tsang. A spatial reasoning based on constraints propagation. In *Proceedings of the Eleventh International Conference on Expert Systems and their Applications*, (in french), pages 297-314, Avignon, France, Mai 1991.
- [du Verdier, 1992] Franck du Verdier. *Solving layout problems using constraint satisfaction - Validation on the layout of microwave electronic equipment*. (in french), These de Doctorat (Ph.D. Thesis), Claude-Bernard University of Lyon, July 1992.
- [Haralick and Elliott, 1980] Robert M. Haralick and Gordon L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263-313, 1980.
- [Hayes-Roth et al., 1986] Barbara Hayes-Roth, Bruce Buchanan, Olivier Lichtarge, Mike Hewett, Russ Altman and James Brinkley. PROTEAN: Deriving protein structure from constraints. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 904-909, Philadelphia, Pennsylvania, August 1986.
- [Lozano-Perez, 1983] Tomas Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, C-32(2):108-120, February 1983.
- [Mackworth, 1977] Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99-118, 1977.
- [Mackworth et al., 1985] Alan K. Mackworth, Jan A. Mulder and William S. Havens. Hierarchical arc consistency: exploiting structured domains in constraint satisfaction problems. *Computational Intelligence*, 1(3):118-126, 1985.
- [Mohr and Henderson, 1986] Roger Mohr and Thomas Henderson. Arc and path consistency revisited. *Artificial Intelligence*, 28:225-233, 1985.
- [Montanari, 1974] Ugo Montanari. Network of constraints: Fundamental properties and applications to picture processing information sciences. *Artificial Intelligence*, 7:95-132, 1974.
- [Preparata and Shamos, 1985] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, 1985.
- [Samet, 1989] Hanan Samet. *Applications of spatial data structures, Computer graphics, Image Processing and GIS*. Addison - Wesley Publishing, 1989.
- [Tommelein, 1989] Iris D. Tommelein. SightPlan: an expert system that models and augments human decision-making for designing construction site layouts. Ph.D. Thesis, Departement of Civil Engineering, Stanford University, August 1989.