Qualitative Reasoning about Imprecise Spatial Information

Thodoros Topaloglou
Department of Computer Science,
University of Toronto,
Toronto, Ontario, M5S 1A4
thodoros@ai.toronto.edu

Abstract

This paper investigates the problem of reasoning about imprecise spatial information. As a representation scheme we assume *haze-points* that are points surrounded by an area, the *haze*, denoting the imprecision that is associated with them. Haze-points are related by *haze* and *precedence* relations. Using these notions we can build higher dimension objects and relations [Top94a]. This paper focuses on reasoning about spatial imprecision which is formalized as a constraint satisfaction problem over networks of constraints expressed in the language of haze-orders. The main contributions of this paper are (a) a set of preprocessing transformations that decrease the ambiguity which is introduced from unconditional transitivity over the haze relation, and (b) a graph-based data structure which is suitable for efficient inferencing of order relations.

1 Introduction

It is generally accepted that neither human perception nor measurement instruments are precise in the information they provide. This is especially true for spatial knowledge. The position of objects in space and the spatial relations between objects are perceived in relation to the degree of detail with which the world is viewed. The ability to reason with knowledge that is imprecise or is specified in various degrees of precision, is fundamental to human intelligence. Incorporating this ability into knowledge representation and reasoning models is a major issue for artificial intelligence research. In this paper we investigate this problem focusing on spatial knowledge.

The haze-space ontology introduced in [Top94a] sets the formal framework for this work. In the haze-space ontology, space is viewed as a totality of spatial objects connected in terms of spatial relations. There are two types of objects: points and regions. In the proposed framework, a point is not a zero-sized object but has an associated haze area, anywhere inside which the point in question may be located. Consequently, points which are "very close" to each other are indistinguishable. The notion of haze can help us to model situations where input information is not very precise such as sonar sensors output, linguistic descriptions, scientific experiment results, etc.; or situations where we are simply interested in limited degrees of precision. The size of the haze area accounts for the degree of precision.

Figure 1 shows the simple case of one-dimensional points which are depicted by small intervals that denote their positional likelihood. Points can be either in *haze* or *precedence* relations with each other; g is a parameter denoting the size of the haze. In the two-dimensional case points are surrounded by a square-shaped haze. Two points can be combined to form a rectangle, in fact a haze-rectangle, which is the only type of higher-order object supported in this representation. This can be extended to more than two dimensions. The only restrictive assumptions that exist are that the modeling space is orthogonal and discrete, which are nevertheless realistic assumptions for geographic space. The discreteness assumption implies that g, the measure of the haze size, takes integer values. The orthogonality assumption implies that points and rectangles are interpreted over the cartesian plane.

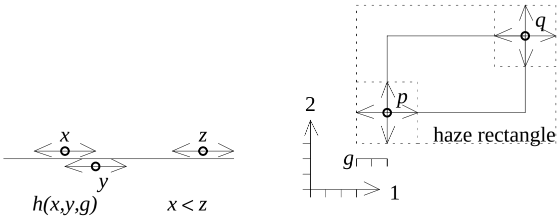
In [Top94a] we presented a set of spatial relations on points and regions that are defined in the haze-space ontology and admit approximate semantics. We have shown that the first-order theory of space based on the haze and precedence relations, for a constant g, is complete (i.e., has a unique model up to isomorphism). We have also shown that this theory of two-dimensional space is the combination of two independent one-dimensional theories. The practical advantage of this result is that reasoning about two-dimensional relations is reduced to two independent reasoning problems, one for each dimension, and the result can be translated back to the higher dimensional terms.

The definition of spatial relations in terms of the haze and precedence primitive relations is, in essence, equivalent to indexing spatial relations with an explicit argument which stands for the degree of precision. [1] Changes to the degree of precision obviously affect the truth or falsehood of stored assertions as well as that of the implied ones. The patterns according to which truth or falsehood change follow the neighbouring structure within the relations set and also depend on the direction towards which the degree of precision changes (e.g., refining, coarsening).

The tasks that spatial reasoning focuses on are, first, given a set of spatial assertions determine its consistency, and second, compute the strongest set of spatial relations

To appear in the ACM Workshop on Advances in Geographic Information Systems (GIS'94), Gaithersburg, MD, December 1-2, 1994.

[1] The terms "grain" and "degree of precision" are used interchangeably throughout this document.



(a) 1-dim haze points (b) 2-dim haze points

Figure 1: The *haze*-point ontology

implied by them. In this paper we will focus in the one dimensional case which, nevertheless, constitutes the core of our reasoning model. We then demonstrate our approach on a one-dimensional ordering inference example, and finally outline its generalization in an integrated reasoning algorithm for spatial relations.

2 Haze Orders

$\mathcal{L}_\mathcal{A}$ is the language of *haze orders*. $\mathcal{L}_\mathcal{A}$ is a sorted predicate calculus with two sorts, *Point* and *Grain*. $x, y, z, \ldots$ are *Point* variables and g is a constant of sort *Grain*. $\mathcal{L}_\mathcal{A}$ has two predicates: h, for *haze*, and $\prec_g$, for *precedence*, both of sort *Point*×*Point*×*Grain*. $\mathcal{L}_\mathcal{A}$ does not use constants to name specific points; instead, it uses unbound variables so that each particular model provides an interpretation for them. The formulas of $\mathcal{L}_\mathcal{A}$ include atomic expressions of the forms $h(x, y, g)$ and $x \prec_g y$ and conjunctions of them. The meaning of $h(x, y, g)$ is that for degree of precision g, x and y are indistinguishable, whereas $x \prec_g y$ means that at degree of precision g, x precedes y. Relation h is symmetric, reflexive and non-transitive; $\prec_g$ is asymmetric and transitive; in addition, h and $\prec_g$ are complementary, i.e., $\forall xy(x \prec_g y \lor y \prec_g x \lor h(x, y, g))$. An extensive discussion on the properties of the first order theory based on $\mathcal{L}_\mathcal{A}$ can be found in [Top94a].

Note that $\mathcal{L}_\mathcal{A}$ is a strictly qualitative formalism with a built-in parameter for imprecision. In the following we define a mapping from $\mathcal{L}_\mathcal{A}$ to the language of underlying space, L_E. L_E is used for defining the semantics of $\mathcal{L}_\mathcal{A}$. L_E also offers a framework for integrating qualitative and quantitative expressions. The qualitative reasoning model that we develop in this paper deals with $\mathcal{L}_\mathcal{A}$ expressions.

L_E is the language of exact space consisting of the subtraction operator $-$, the relations $=$ and $<$, a constant symbol c_i for each integer i and a countable number of integer variables $x, y, z, \ldots$ The set of L_E terms consists of constants, variables and $x - c_i$ terms. The atomic formulas of L_E are of the form $t = t'$ or $t < t'$ where t, t' are terms; the formulas are formed by closing the atomic formulas under $\land$, $\lor$ and $\neg$. An interpretation of L_E is a function v assigning an integer value to each of its terms and a truth value to each of its formulas. A model of L_E is an interpretation that satisfies all its formulas.

The mapping $\mathcal{L}_\mathcal{A}$ to L_E is defined as follows: For each formula $\phi \in \mathcal{L}_\mathcal{A}$, we define $\phi[g] \in L_E$ to be the same as ϕ except that every $h(x, y, g)$ expression is replaced by $|x-y| \le c_g$ and each $x \prec_g y$ expression is replaced by $x - y > c_g$.

If v is a model of L_E and g is a measure of the haze, then the pair (v, g) is a model of $\mathcal{L}_\mathcal{A}$. For a formula $\phi \in \mathcal{L}_\mathcal{A}$, we define

$$(v, \mathsf{g}) \models \phi \text{ if } v \models \phi[\mathsf{g}]$$

Let KB be a set of $\mathcal{L}_\mathcal{A}$ formulas. KB' is the exact counterpart of KB, i.e., each $\phi \in KB$ is replaced by a $\phi[g]$. KB is consistent if we can find a g such that KB' is consistent, and KB is minimaly consistent if g is minimal.

Example 1 (single g) Assume the knowledge base: $KB = \{h(y - x, 6, g),\ h(y, z, g),\ h(z, u, g),\ u \prec_g t\}$. Its exact-space counterpart is $KB' = \{6-g \le y-x \le 6+g,\ -g \le y-z \le g,\ -g \le u - z \le g,\ t - u > g\}$. Formula $h(y - x, 6, g)$ states that the length of the interval $[x, y]$ is approximately 6 units. In fact this is not a term of $\mathcal{L}_\mathcal{A}$. Its correct form should be $h(t_1, t_2, g)$ where t_1 stands for the length of the $[x, y]$ interval, and t_2 for the constant symbol "6", meaning that the interval from x to y is about 6 units. This example can convince the reader that our formalism can be used for any kind of noisy data if the necessary meaning is attached. In the present paper, we do not intend to expand further towards this direction. □

Example 2 (multiple gs) In this example each of the above formulas is specified at a different degree of precision (e.g., $g_1, g_2, \ldots$). $KB = \{h(x, y, g_1),\ h(y, z, g_2),\ h(z, u, g_3),\ u \prec_{g_4} t\}$, and $KB' = \{-g_1 \le x - y \le g_1,\ -g_2 \le y - z \le g_2,\ -g_3 \le u - z \le g_3,\ t - u > g_4\}$. □

In the next section we reduce the problem of reasoning about haze orders to a constraint reasoning problem.

3 Constraint Networks

A constraint network consists of a set of quantities $X_1, X_2, \ldots, X_n$, where D_i is the domain of each quantity, and a set of unary, $P(X_i)$, and binary, $C(X_i, X_j)$, constraints over these quantities. A solution of the network is a tuple $\{x_1, x_2, \ldots, x_n\}$ such that the assignment $X_i = x_i$ satisfies all the constraints. A network is consistent if at least one solution exists.

A constraint network is associated with a directed graph where nodes represent quantities and edges represent constraints. An edge $\langle i, j\rangle$ indicates that the constraint $C_{i,j}$ is specified between i, j. Edges are labeled.

A constraint C' is tighter than C'' if every pair of values for C' is allowed by C'', $C' \subseteq C''$. This notion is extended to networks. $\subseteq$ defines a partial order over networks. Two networks are equivalent if they have the same set of solutions. A network M is minimal if there is no equivalent network with tighter constraints.

Two interesting problems in constraint networks are: (a) given a network N, determine its consistency and (b) find the minimal solution to N.

Consider a special case of the above definition where the quantities are objects (e.g., points in space) related by binary relations. Let *Rel* be a set of labels and $C_{i,j} = \{r | r \in Rel\}$ a constraint between two nodes i, j. The following operations are defined:

addition: If $C'_{i,j}$, $C''_{i,j}$ are constraints between i, j then the combined constraint between them is

$$C_{i,j} = add(C'_{i,j}, C''_{i,j}) = C'_{i,j} \cap C''_{i,j}$$

If the intersection is empty then the arc is inconsistent [Mac77].

multiplication: If $C_{i,j}$ and $C_{j,k}$ are the constraints on two successive edges $\langle i, j\rangle$ and $\langle j, k\rangle$, then the transitive constraint on the $\langle i, k\rangle$ edge is the result of the following operation:

$$C_{i,k} = mult(C_{i,j}, C_{j,k}) = \cup_{\, r_1 \in C_{i,j}, r_2 \in C_{j,k}} \otimes (r_1, r_2)$$

where $\otimes$ is the constraint transitivity operation and depends on *Rel*.

The atomic formulas of the language of haze orders $\mathcal{L}_A$, follow one of the following types:

- $x \prec_g y$,
- $h(x, y, g)$

Both types are representable as simple binary constraints between point objects and therefore reasoning about haze orders can reduce to a constraint reasoning problem. [2]

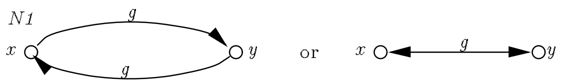
The consistency problem and the minimal network computation of a binary constraint network are extensively studied problems [KL91], [DMP91], [Mei91]. Deciding the consistency of a general network is an NP-hard problem [VKvB89]. However, there are special cases where a polynomial algorithm for deciding consistency exists, or weaker forms of consistency can be computed efficiently for the general case. The parameters that influence the complexity of the consistency checking algorithm are the domain of the variables and the type of arc labels of a constraint network.

In the next section, we investigate the consistency conditions and algorithms for determining it for haze order constraints.

4 Reasoning about Haze Orders

4.1 Qualitative Networks

In a qualitative formulation of the constraint satisfaction problem for haze orders the set of labels is assumed to be $Rel = \{<, >, g\}$; label g denotes that variables x, y are related by the haze relation. If x and y are in haze relation, the constraint network contains two directed edges from x to y and from y to x both with label g; for brevity, we mark this edge with a bi-directional arrow (see figure bellow). If $x \prec y$, then there is an edge in the constraint network emanating from node x and leading to node y. If an edge $\langle x, y\rangle$ is unconstrained then we assume that x, y are related by the most general constraint i.e., $\{<, >, g\}$.



Let us now see how the notion of consistency is conveyed in the case of haze orders. The meaning of $h(x, y, g)$ is that x and y are within distance less than g independently of their relative orientation, i.e., they can be $x - y < g$ or $x = y$ or $y - x < g$. Obviously, this expression is not expressed as a simple binary constraint although $h(x, y, g)$ is. Moreover the graph $N1$ created for $h(x, y, g)$, although trivial, does not satisfy arc-consistency in its strict meaning. $N1$ would be arc-consistent iff for all instantiations of x, say C_x, there was an instantiation C_y such that $C_{x,y}$ was satisfied; in other words $add(C^{(i)}_{x,y} \otimes C^{(i)}_y \otimes C^{(i)}_{y,x}, C^{(i)}_x) = C^{(i+1)}_x$. This requires that both x and y be instantiated by the range $[-g, g]$ so that the above consistency maintaining expression is satisfied. In other words, g-cycles are tolerated as long as nodes are instantiated by an appropriate range. On the other hand, $<$-labeled edges are unidirectional and the following well known theorem [vB90], [GS93], serves as the inconsistency detection condition.

Theorem 1 *A constraint graph is inconsistent if it contains $<$-cycles.*

[2] Note that grain, g, is treated as part of the constraint's label.

For sketching the proof we use the reduction of the $\mathcal{L}_A$ to L_E and the following argument: Assume that a $<$-cycle between x, y, z, x exists. In L_E, this means that $y - x > \mathsf{g}$ (1), $z - y > \mathsf{g}$ (2), and $x - z > \mathsf{g}$ (3). By adding (1) and (2) we obtain $z - x > 2\mathsf{g}$ which contradicts (3). Checking for inconsistency is achieved in $O(n)$ time where n is the number of nodes.

Furthermore, consistency here admits a quantitative measure, this is the length of the longest, not cyclic, g-path. For instance, graph $N1$ has maximum g-path length 1. The practical meaning of this measure is that it quantifies the increase of the degree of precision size that is necessary in order to keep the constraint network consistent. Among all consistent scenarios for a constraint network the minimal one is the one with the smallest g-path. The computation of all the implied relations for a given constraint network (the "all-to-all" problem [vB90]) is done in $O(n^3)$ time using the PC-2 algorithm [Mac77].

There is more to be said about the definition of the constraint transitivity operator, $\otimes$. Recall that, haze is a reflexive but not transitive relation. If edges $\langle i, j\rangle$ and $\langle j, k\rangle$ are both labeled with g the following can happen with regards the label of $\langle i, k\rangle$.

1. $C_{i,k} = \{<, >, g\}$

$\otimes_1$	$<$	g	$>$
$<$	$<$	$\{<, g\}$	$\{<, g, >\}$
g	$\{<, g\}$	$\{<, g, >\}$	$\{>, g\}$
$>$	$\{<, g, >\}$	$\{>, g\}$	$>$

   This transitivity table is destructive because it deteriorates the tightness of the edge labels.

2. $C_{i,k} = \{2g\}$ (i.e., $g \otimes g = g + g$)

$\otimes_2$	$<$	g	$f(g)$
$<$	$<$	$\{<, g\}$	$\{f(g) - g, <\}$
g	$\{<, g\}$	$2g$	$g + f(g)$
$f'(g)$	$\{f(g) - g, <\}$	$f'(g) + g$	$f'(g) + f(g)$

   where $f(g)$ is a linear function of g (e.g., $f(g) = 3g$). This table suggests that the size of haze increases due to transitivity.

3. For the case of multiple gs ($g_1 > g_2$) the table $\otimes_2$ is variated as follows:

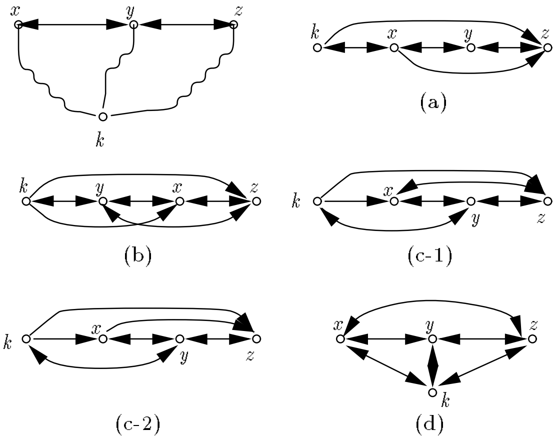
$\otimes_3$	$<_{g_1}$	g_1	$f(g_1)$
$<_{g_2}$	$<_{g_1+g_2}$	$<_{g_1-g_2}$	$\{f(g_1) - g_2, <_{f(g_1)}\}$
g_2	$\{g_1, <_{g_1}\}$	$g_1 + g_2$	$g_2 + f(g_1)$
$f'(g_2)$	$\{f(g_2) - g_1, <_{f(g_2)}\}$	$f'(g_2) + g_1$	$f'(g_2) + f(g_1)$

   The table for $\otimes_3$ is similar to that of [MS91].

In all the above cases the application of unconditional transitivity rules increases the size of the hazes and weakens the constraints on the network edges. In order to obtain a "better" solution, i.e., to achieve g-paths of smaller length , we need to disambiguate g edges before we use them in transitivity operations. For this purpose we propose the following 4-node transitivity rules and heuristic preprocessing transformations.

- *Inspecting 4-node graphs (same g).*

By checking all the four node sub-graphs in which x, y and z are involved and applying the 4-point transitivity axioms of order over the haze relation, it is possible to compute tighter labels under certain conditions ($<$ is tighter that g according to the "answer's quality" criterion stated above).



(a) If there is a node k such that k is in an order relation R with z and in haze relation with x then x is in R with z.

$\forall xyz \exists k (h(x,y,g) \wedge h(y,z,g) \wedge h(x,k,g)$
$\wedge R(k,z,g) \Rightarrow R(x,z,g))$

(b) If k is in R with z and x and in haze with y, then x is in haze with z.

$\forall xyz \exists k (h(x,y,g) \wedge h(y,z,g) \wedge h(y,k,g)$
$\wedge R(k,z,g) \wedge R(k,x,g) \Rightarrow h(x,z,g))$

(c) If k is in R with z and in haze with y and x, then x is either in haze or in R relation with z.

$\forall xyz \exists k (h(x,y,g) \wedge h(y,z,g) \wedge h(y,k,g)$
$\wedge R(k,z,g) \wedge R(k,x,g) \Rightarrow h(x,z,g) \vee R(x,z,g))$

(d) If there exist node k which is different than x, y and z, and k is in haze relation with all x, y and z nodes, then x is in haze relation with z as well.

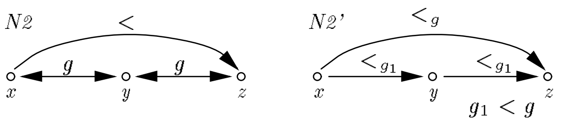
$\forall xyz \exists k (h(x,y,g) \wedge h(y,z,g) \wedge h(x,k,g)$
$\wedge h(k,z,g) \wedge h(k,y,g) \Rightarrow h(x,z,g))$

Checking 4-point graphs takes $O(n^4)$ time.

• *Refinement as a preprocessing step.*

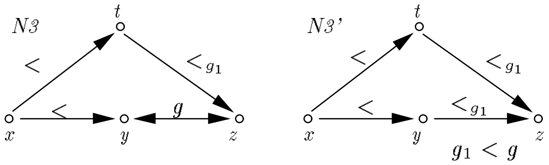
If two points are not distinguishable at haze size g, perhaps they differentiate at some $g_e < g$. In particular, if it is the case that $h(x,y,g), h(y,z,g)$ and $x \prec_g z$, then x lies on the left of y, therefore the order $x < y < z$ can be recovered for some $g_1 < g$. In the constraint network this involves to search for $N2$-type subgraphs, rewrite them as $N2'$ subgraphs and add the relationship $g_1 < g$ in the knowledge base. Then either the rules from case 3 or the unconditional transitivity operations are applied.

(a) $N2$-transformation



$\forall xyz (h(x,y,g) \wedge h(y,z,g) \wedge x \prec_g z \Rightarrow$
$x \prec_{g_1} y \wedge y \prec_{g_1} z \wedge g_1 < g)$

(b) $N3$-transformation



$\forall xyzt (h(y,z,g) \wedge x \prec_g y \wedge x \prec_g t \wedge$
$t \prec_{g_1} z \Rightarrow y \prec_{g_1} z \wedge g_1 < g)$

A naive algorithm for the $N2$ transformation requires $O(n^3)$ time, and for the $N3$ requires $O(n^4)$. With an appropriate data structure these times are further optimized. For example, an adjacency list implementation for the network reduces the cost of $N2$-transformation to $O(n^2)$.

4.2 Order Graphs

The order graph is a convenient representation of the earlier presented qualitative network, that is suitable for answering queries. Given a set of order relations between grains and a most tight haze-order graph, we want to determine the maximun length haze-order paths. To do so, we use a notion of node ranking similar to those of [GA89], [GS93].

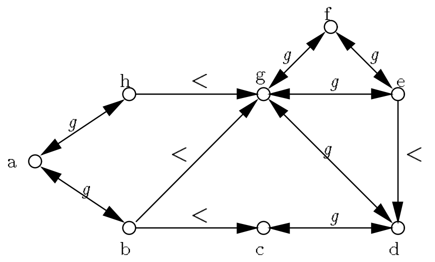
The rank of a node i is a value that each node is assigned depending on the order relations available in the haze-order graph. The distance increment between nodes is taken according to the relative order of grains, i.e., $k_m < k_n$ if $g_m < g_n$.

The nodes that are in *haze* relations are assigned the same rank value. Two nodes i, j such that $i <_{g_n} j$ get the ranking, $rank(j) = rank(i) + k_n$. The following procedure performs the ranking in $O(n + e)$ time.

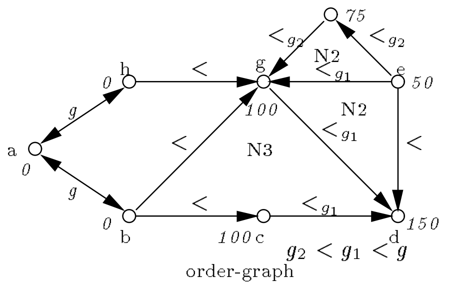
algorithm: order-graph
input: A haze-order graph, HO, and an ordered sequence of grains, G.
output: maximum length order paths.

```
N = topological sort of the nodes of HO
i = first(N); rank(i) = 0
for each node i ∈ N do
    for each node j such that edge ⟨i,j⟩ ∈ HO do
        if label(⟨i,j⟩) = g then
            rank(j) = rank(i)
        else if label(⟨i,j⟩) =<_{g_n} then
            rank(j) = rank(i) + k_m
    end-for
end-for
```
end order-graph

Example 3 The figure bellow illustrates the order graph computed for the input haze order graph (left on the figure) after the preprocessing transformations were performed. Note that this is a suitable data structure for answering order queries. The graph does not include all the implied edges, however, it can answer $h < d$ because a $<$-path between h and d exists. The ranking information can help to answer queries such as the closest smaller point to g, that is f. Furthermore, a is less than g because $rank(c) < rank(c)$, but nothing can be said about the relation between b and h.



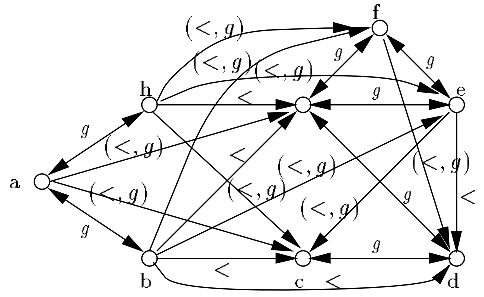
input graph



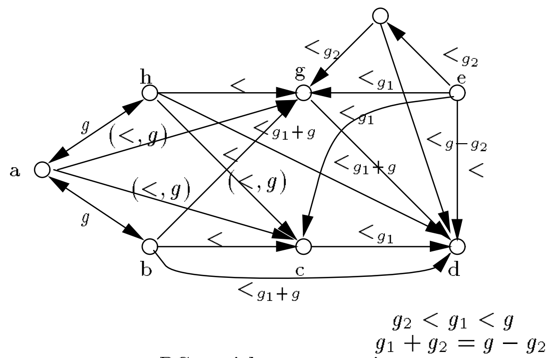
order-graph

□

Example 4 The following two figures show the result of the PC-2 algorithm which computes transitivity and check for consistency among triples of variables, with and without preprocessing (the $\{<, g, >\}$ labeled edges are not drawn for readability purposes). The network without preprocessing contains more g edges. The network with the preprocessing step performed is less ambiguous than the network produced by PC-2 without preprocessing; in fact, the former contains g-paths of length 1 whereas the latter contains g-paths of length 4.



PC-2 without preprocessing



PC-2 with preprocessing

□

5 Applications

5.1 Ordering Inference

Determining orders is acknowledged as a fundamental problem in gene mapping in the context of the Human Gemone project [GR93]. This problem is challenging because of the ambiguity and uncertainty that is associated to the ends and distances of loci (genetic intervals). Haze orders and order graphs offer the right type of model and data structure for dealing with this problem. In this section, we will simply show the applicability of our formalism in the genetic domain. As we have seen, our model is strictly qualitative at the present; an extension that will allow to express imprecise interval lengths will offer more advantage to this task.

In the following we assume that that an interval, i, is a pair of two haze points, i_L, i_G. Genomic data are assumed to be in an order graph form. Triples of the form (i, l, j) denote edges of the order graph. The following queries are a sample of interesting order inferences:

Q1. Find all intervals that are in haze relation with interval a for $g < g_q$.

$$\{i | \bigvee_{j \in \{L,G\}} (i_j, g, a_j) \wedge g < g_q\}$$

Q2. What is the best known order between a and b.

$$g = shortest\_path(a_G, b_L) \vee g = shortest\_path(b_G, a_L)$$

Q3. Is the order $a \prec_g b$ consistent with the KB?

$$\{(a_G, <_k, b_L) \wedge k \leq g\}$$

Q4. Find the g that makes $a \prec_g b$ consistent with KB.

$$\{g | (a_G, <_g, b_L)\}$$

5.2 Spatial Relations

Reasoning about spatial relations with a grain (degree of detail) parameter is a new problem which can be approached at two levels of abstraction: (a) reason in the high-dimensional space (e.g., two-dimensional) and draw inferences by computing transitivities of spatial relations ([Ege91], [Ran92]), and (b) project the objects and the relation definitions to a some low-dimensional structures with an order relation defined, make inferences using order transitivity and transfer the results to the higher dimensional structures [Roe94].

Computing transitivities of spatial relations is generally impractical [Ran92]. However, utilizing the structure of spatial relations (conceptual neighbourhood [Fre92], [Gal93]) while computing transitivities can improve the performance of certain inferences. In our case, the additional parameter of grain clearly adds more complexity to the spatial relations transitivity.

Our solution to the problem of reasoning about "approximate" spatial relations is a combination of the previous two. As we have shown in [Top94a], the defined spatial relations in our model, are decomposed into a set of order and haze relations over two one dimensional coordinate spaces. Reasoning with haze orders in one dimension was discussed in the earlier sections. The challenging part of the process is how to transfer the inferences back to spatial relations. As the degree of detail is refined or coarsened in the haze orders case, the implications should be marked in the spatial relations level. To do so, we need to maintain the structure of the spatial relation set and move along it depending on the result of the underlying one-dimensional reasoning problems. Assume, for instance, that A is *inside_tangent* to B; if the result from the underlying haze-order reasoning problem is a disambiguation of the haze relation involved, then the *inside_tangent* relation collapses to either an *overlaps* or an *inside* relation depending on the direction of the recovered order relation.

6 Discussion

In this paper we addressed the problem of reasoning with imprecise spatial information. Although the discussion was centered in the one-dimensional case, we presented an abstract algorithm that addresses this problem in the context of two-dimensional spatial relations. Detailed discussion on the latter is presented in [Top94b].

The major contribution of this work is the definition of the preprocessing transformations that disambiguate *haze* relations by using the connectivity of the haze-order graph. These transformations always guarantee tighter and less ambiguous constraints between the graph nodes. The two measures that we used to characterize the quality of the constraints on the edges of the graph are the length of g-paths and the ratio of order constraints to the total number of constraints. As a next step of this research we plan a thorough quantitative evaluation of the alternatives and experimental verification of our claims. The sensitivity of our method to the structure of the graph, makes us believe that the average performance will be better than the pessimistic worst-case theoretical analysis that we provide. Finally, we intend to use real life data from the Human Genome database to identify boundary cases of the proposed method and do further tuning.

Another important result in our work is the definition of the order graph structure which can be used as an index to support efficient inference of orders in a setting where orders of multiple grains exist. We demonstrate this principle over example queries form the Human Genome database domain. The integration of quantitative (distance) and qualitative information (orders) in the order graph is planned as future work.

Finally, our model for representing and reasoning about imprecise spatial information can be adapted to other kinds of “noisy” data, such as temperature, in qualitative reasoning tasks.

References

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49:61–95, 1991.

[Ege91] Max Egenhofer. Reasoning about Binary Topological Relations. In *Second Symposium on Large Spatial Databases (SSD'91)*, pages 143–160, 1991.

[Fre92] Cristian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227, 1992.

[GA89] M. Ghallab and M.A. Alaoui. Managing Efficiently Temporal Relations through Indexed Spanning Trees. In *Proceedings of IJCAI-89*, pages 1297–1303, 1989.

[Gal93] Antony Galton. Towards an integrated logic of space, time and motion. In *Proceedings of IJCAI-93*, pages 1550–1555, 1993.

[GR93] J. Guidi and T. Roderick. Inference of order in genetic systems. In *First International Conference on Intelligent Information Systems for Molecular Biology*, pages 163–171, 1993.

[GS93] A. Geverini and L. Schubert. Efficient temporal reasoning through timegraphs. In *IJCAI-93*, pages 648–654, 1993.

[KL91] H. Kautz and P. Ladkin. Integrating Metric and Qualitative Temporal Reasoning. In *Proceedings of AAAI-91*, pages 241–246, 1991.

[Mac77] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[Mei91] Itay Meiry. Combining Qualitative and Quantitative Constraints in Temporal Reasoning. In *Proceedings of AAAI-91*, pages 260–267, 1991.

[MS91] A. Mukerjee and T. Schnorrenberg. Tolerance Spaces: A model for multiple resolutions in space, time and perception. Dept. of Computer Science, Texas A & M University, 1991.

[Ran92] Randell, D. and Cui, Z. and Cohn, A. A spatial logic based on regions and connection. In *Proceedings of KRR-92*, 1992.

[Roe94] Ralf Roehrig. A theory for qualitative spatial reasoning based on order relations. In *Proceedings of AAAI-94*, pages 1418–1423, 1994.

[Top94a] Thodoros Topaloglou. First order theories of approximate space. In *AAAI-94 Workshop on Spatial and Temporal Reasoning*, pages 47–53, 1994.

[Top94b] Thodoros Topaloglou. Representation and reasoning in spatial knowledge bases. Thesis Proposal, Forthcoming, 1994.

[vB90] Peter van Beek. Exact and Approximate Reasoning About Qualitative Temporal Relations. Technical Report TR 90-29, Department of Computing Science, University of Alberta, August 1990.

[VKvB89] M. Vilain, H. Kautz, and P. van Beek. Constraint Propagation Algorithms for Temporal Reasoning: a Revised Report. In D.S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.