CS 666 Term Paper
Graph Isomorphism : Relationship to the Spatial Template Recognition problem
DRAFT COPY

Steven Woods
89800462

June 16, 1994

Abstract

This work describes the work of Lubiw (1981) with reference to subsequent literature from the perspective of investigating Spatial Template Recognition.

Spatial Template Recognition (STR) is an interesting problem with application to many different domains, including intelligence analysis and "high level" problems in machine vision. Essentially, a human concept with many spatial aspects and a high degree of variability is represented according to some scheme, and instances of this concept are located in a possibly "dirty" or "noisy" situation. A natural way of modeling STR is as a *Constraint Satisfaction Problem* (CSP). While CSP in general is NP-Complete, in practice certain characteristics of spatial problems have been exploited to make realistic problems solvable in reasonable amounts of time. Of concern is whether these results will "scale up" to larger templates and situations.

Graph Isomorphism (GI) is a well-studied, related problem that has yet to be shown to be a member of the class NP-Complete. It represents a class of problems known as "GI-Complete", hypothesized to be "simpler" on the basis of certain problem characteristics. In this work these characteristics are discussed with reference to similar aspects of STR, and several related problems are examined with emphasis on problem complexity. Of particular interest is the fact that GI has been shown to have both "restricted" and "relaxed" versions that are NP-Complete. A version of restricted STR based on "spatial locality restrictions" has been demonstrated to be particularly efficient, and it seems possible that this restricted problem may have a complexity related to restricted GI. This paper introduces some of the main problem aspects of GI and some other closely related problems and serves as a basis for future more detailed analysis of STR complexity.

1 Problem Introduction

Static **Spatial Template Recognition in Situation** (STR) which has been an active area of interest for practical application across various domains is defined in Section 2. Section 3 outlines an approach to solving of STR with a complete search strategy. Various formulations of the **Graph Isomorphism** (GI) problem are described in Sections 4.1,4.2, and analogies between STR and these formulations are shown in Section 4.4. In Section 4 the results shown in Lubiw (1981) are outlined, and several restrictions and generalizations of GI subsequent to Lubiw (1981) are discussed. Lubiw (1981) observes that GI may be representative of a class of problems know as GI-Complete that is in NP[1], but neither in P nor in NP-Complete. In addition, Lubiw (1981) and Booth & Colbourn (1977) describe a breadth of problems polynomially equivalent to GI, and Lubiw (1981) attempts to explore the boundaries between problems known to be NP-Complete (such as **Subgraph Isomorphism** (SGI)(Garey & Johnson 1979)) and closely related problems that appear to be members of GI-Complete. In Section 4.5 the related work of Andrew Wong in pattern matching is briefly related to STR, and this paper's main conclusions are summarized in Section 5, including some conjectures

[1]P represents the class of problems that are deterministically solvable in time polynomially related to the problem input. NP represents the class of problems solvable in nondeterministic polynomial time. Two problems are polynomially related if an encoding of either can be transformed into an encoding of the other in polynomial time. NP-Complete represents a class of problems in NP with the property that if any one NP-complete problem can be solved in polynomial time, then every problem in NP has a polynomial-time solution, that is, P = NP (Cormen, Leiserson & Rivest 1992).

about STR's complexity.

2 The Spatial Template Recognition Problem

2.1 What is STR and how complex a task is it?

STR is the problem of finding an instance of some predefined configuration of objects in a given situation, or set of object instances. An example of a non-typically rigid spatial template in a two-dimensional domain would be a "map" of a typical molecule that describes the molecule in terms of the relative positions, types and orientations of the various atoms. An example of a rather discrete, exact situation might be a two-dimensional sample containing many instances of different types of molecules discernible only by individual sub-parts. The question formed by STR for this problem instance would be: "Where are the instances of this molecular template in this situation?". In this example both the template and the elements of the situation have relatively "rigid" design - the molecular model has little variability from instance to instance, and the situation has potentially little ambiguity with respect to both identity and position of situational elements.

A more complex domain for STR is intelligence analysis. An intelligence analyst must draw on specific experiential and learned knowledge of enemy behaviour to recognize specific deployment patterns signifying organizational relationships of sighted enemy elements. One aspect of this complex task is recognizing the existence of enemy "doctrinal templates" which detail the patterns in which military units are commonly deployed for specific activities. These "templates" represent a snapshot view of how the elements of a military grouping are commonly positioned. Since military units are often strictly structured and deployed according to fixed procedures, a fairly accurate "typical" spatial representation can be built of particular formations undertaking certain activities in the field, such as "Motor Rifle Regiment (MRR) in attack". The analyst must fuse spatially and temporally disjoint messages or sightings into a coherent picture of the entire situation, including higher level formations and their activity. The individual elements of the MRR spatial template may be envisioned as "slots" in the template, and the rules or restriction regarding deployment of these elements can be represented as constraints amongst the template slots. The current set of sightings is referred to as the "situation", and each sighting as a "situation element".

One major difficulty with such a task is that these templates are embedded within an extremely complex, dynamic and incomplete "view" of the enemy[2]. Intelligence messages from the field may contain vague or conflicting composition or location information, and inaccurate messages may have been modified by experts into inaccurate conclusions. Also, any template is simply an approximation of some expert-held "ideal" concept, related work on improving these approximations with experience may be found in Woods (1994). Templates defined in terms of spatial relationships between components of specific types, sizes, activities and orientation approximate reality by allowing certain tolerances on the attributes or spatial constraints. Considering a template as a set of "stick-pins" of varying types and sizes inter-connected by a set of rubber-bands with a minimum and maximum stretch capability, the process of template matching reduces to finding appropriate situation elements in which the stick pins may fit and where the bands between the pins allow this fit.

Matching of these templates to incompletely specified situational views can result in many partial solutions where only some subset of the pins could be fit to a situation element. In these cases which of many partial solutions is more promising may possibly be determined according to predefined criteria or measures. In addition, the knowledge of the precise structure of these templates themselves is only approximate, and the templates themselves may evolve over time. Consequently, any attempt to perform the recognition task algorithmically will find approximate solutions only, however, these may form a base-point for expert utilization.

While it is important to understand how the STR problem in reality is rife with complexity, for the purposes of this paper, I will assume a simpler STR in which I am interested only in finding a set of all matches of a template in a situation. Template flexibility is modeled using range constraints between slots, and situational elements inaccuracy modeled with locations that have varying degrees of specificity.

[2]Other related work in pattern analysis such as Stacey & Wong (1988), Ghahraman, Wong & Tung (1980) and Wong, You & Chan (1990) models problems with a far greater "rigidity" in terms of the spatial template model.

2.2 STR Representation

A relatively obvious formulation of STR as a **Constraint Satisfaction Problem** (CSP) was proposed in Woods (1993*b*) and Woods (1993*a*). I initially define STR as follows: A situation consists of a set of elements each with a set of characteristics (such as color, type, etc.). A template consists of a set of variables, each possibly limited by a set of range or specific characteristic values. Additionally, each variable is possibly connected to any other variable (or possibly set of variables) by a constraint-arc or set of arcs which impose relative characteristics such as relative position, orientation, or possibly which insists upon pairwise variable characteristic assignments. A template instance is "found" in a situation if a 1:1 mapping exists from situation elements to variables such that no situation element maps to more than one variable, and based on this assignment, all constraints represented by constraint-arcs are satisfied. A decision problem based on this definition could be expressed as "Given a Template T = (set of Variables V + set of Variable-constraints VC + Arc-constraints AC), and a Situation S = (set of Situation elements), is there an instance of T in S ?". Problems specified in this fashion, where a solution is composed of a set of consistent element-to-slot assignments are well described in (Kumar 1992). An example of a template constraint graph is shown in Figure 1 with five node slots where each slot is constrained by several attribute ranges, and 11 inter-slot spatial constraints on distance, position, and orientation.

I am interested in this paper in describing the complexity of locating all possible instances of a template in a given situation. Consider a template with n slots (each possibly related to another by one or more constraints), and a situation with some number m of situation objects. We need to find all of the possible mappings from situation objects to template slots such that the "template matches" (i.e. all of the constraints are satisfied). Essentially we are trying to solve general CSP[3].

A "perfect" or total solution to a STR problem is an assignment from the situation elements to the template slots. There are n template slots to be filled with from the pool of m situation objects, with the ordering of the selection corresponding to an assignment to the slots. There are m choose n candidate solutions for our set, that is, the number of n-combinations of an

[3] CSP is NP-Complete: consider a proof by restriction to Graph 3-Coloring for instance.

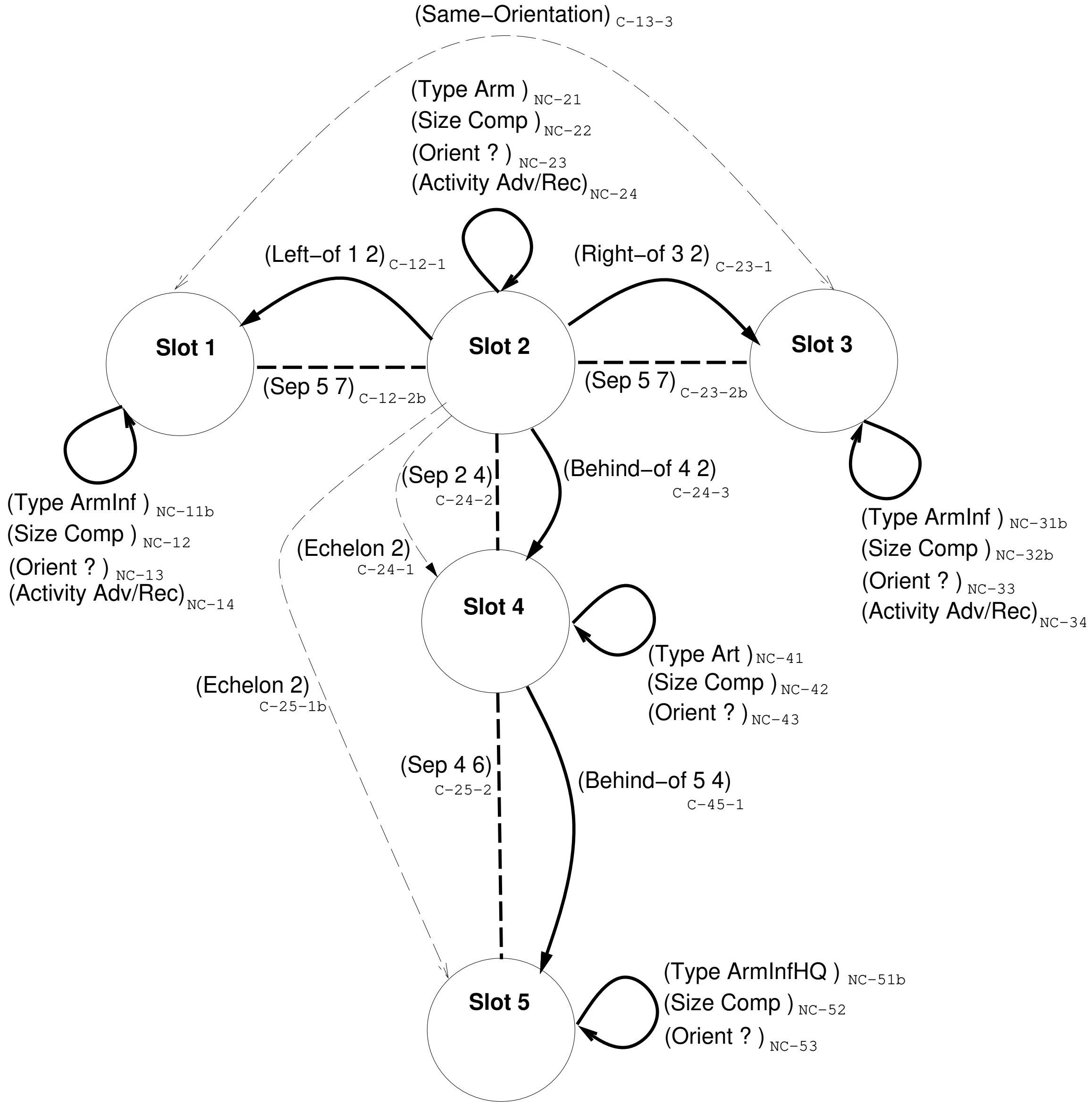

Figure 1: Whole CSP 5 variables, 11 arc constraints.

m-set:

$$NumberCandidateSolutions = m!/n!(n-m)! \tag{1}$$

Thus there are some number of solutions exponential in the size of the number of situation elements, m. Note that typically the number of situation elements far dominates the number of template slots. Naively generating these is not going to be a good solution! However, in the next section several spatial heuristics will be discussed that seem to greatly localize the problem.

3 Templates and Search : Complexity

We are interested in finding all possible instances of a particular template in a set of situation elements, and consequently in finding all solutions to a CSP modeling our template recognition problem. In the CSP literature, these problems are typically solved via some search strategy coupled with domain-dependent heuristics, via some graph-reduction method which attempts to propagate consistency throughout the problem graph, or through some hybrid approach utilizing both search and propagation. It has been shown that search strategies with "look-ahead" properties can be thought of as augmenting search with some degree of during-search constraint propagation. By defining search strategies in this way the amount of constraint propagation adopted during search can be more easily identified and adapted, and various strategies more easily compared empirically. Although Nadel (1989) indicates which type of hybrid strategy is best for some small toy domains, published results solving interesting problems using this strategy are difficult to find. The spatial template recognition problem provides an excellent example of a complex problem which can be addressed in a reasonable length of time utilizing hybrid search.

The hybrid search based approach was demonstrated to be useful even for large situation datasets as a result of effective use of specific domain heuristics such as spatial locality and general domain independent heuristics such as applying constraint propagation during search, or abstracting the CSP. The question is, how did these properties overcome the computational complexity of inherent to the problem ? Related problems are discussed later in this paper.

If we view the space of candidate solutions to an STR as being determined by the template representation, then we must ask what happened to our problem complexity as the problem is either relaxed or restricted. Section 3.1 and 3.2 discuss some relaxations and restrictions possible in the STR domain.

3.1 Relaxing STR as CSP via Partial Solutions

In the automated portion of the approach defined at length in Woods (1993*b*) and Woods (1993*a*), search is used to satisfy as many constraints in the original problem as possible. If no solutions are found for the original template specification, it is possible to "relax" some of the constraints in a systematic way so as to find instances of some "least relaxed" version of the original template. These "relaxed" templates may be thought of as abstracted views of the original spatial concept. This relaxation essentially increases the size of the potential solution set. In the absence of other factors, generating all relaxed template instances in a situation is more difficult since there are more to generate.

In other work such as that of Lapointe & Proulx (1994), a different approach is used in which the fuzzy *distance* between a perceived instance and a template is measured and is used to determine a "match" between situation and template. Once again, both domain heuristics and domain independent heuristics were used to obtain an efficient solution. In this approach, the spatial template acts less "precisely" as a direct determiner of the solution space and more like a "classifier" of the degree of fit of a particular mapping. In any event, this is once again a relaxation, except that the resulting solution set is ordered precisely by some possibilistic distance from an "ideal" represented in the template.

Both of these works make some claim to being "good enough" in the absence of a perfect solution, however, a better understanding of the conditions under which these "good enough" solutions may be obtained would be of great value in evaluating possible real applications of these algorithms.

3.2 Restricting STR as CSP via Spatial Locality

Spatial locality is an example of an important "simplification" or restriction of the general CSP nature of STR in our formulation. This heuristic is applicable to CSP problems that are grounded in spatial coordinates. Essentially, we take advantage of the fact that our spatial templates may be "precompiled" such that it is possible to encode rough boundary information for an entire solution based upon only partial variable assignment information. For instance, if a template has no two slots farther apart than some measure max, then during search if an assumption is made about the location of some template slot t, then we may conclude that no subsequent template slot can be assigned a situation element farther than max from the slot t assignment. Given that we may easily determine the distance between situation objects (perhaps through a pre-built index or similar method), it seems easy to dramatically reduce overall search complexity by pruning large numbers of situation elements as candidates for slot assignment "locally".

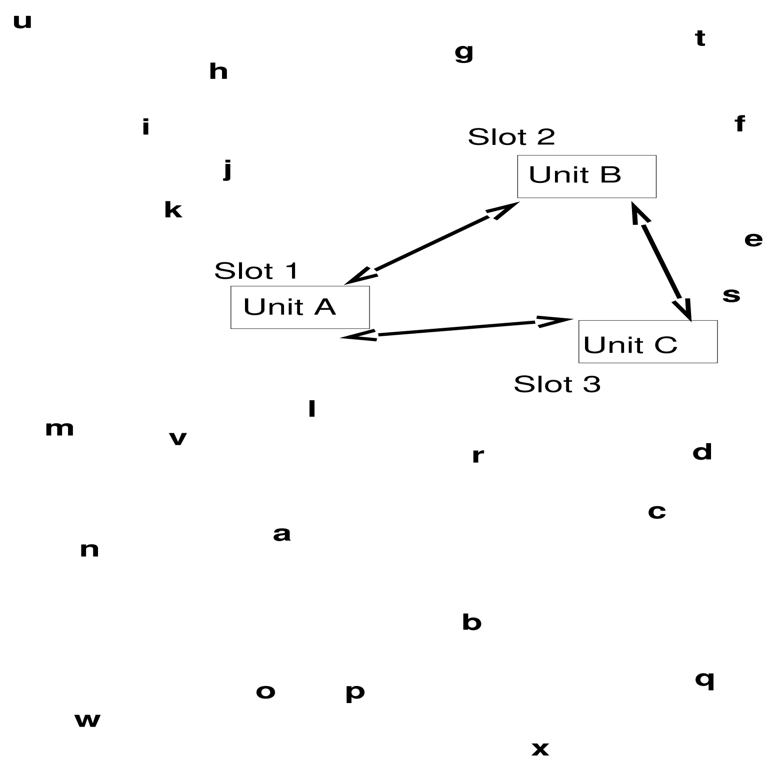

Figure 2: Situation with partial solution overdrawn.

Once assignments have been made to some variable (or set of variables), a boundary is created limiting the scope of this particular solution. Now, when subsequent variables and constraints are added making the problem more specific, the scope of the search is restricted according to the outlined area

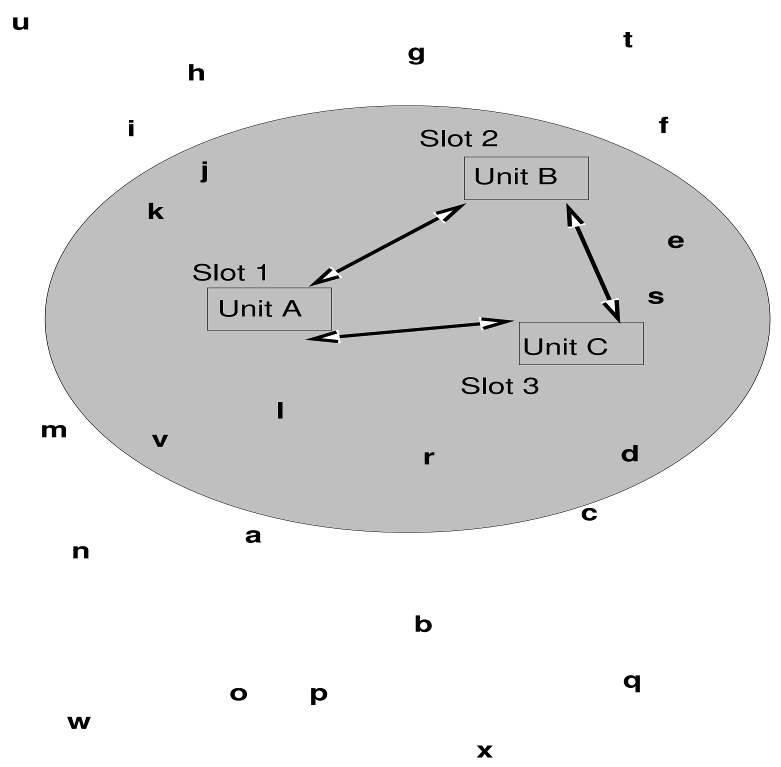

Figure 3: Situation with spatial pruning based on partial solution.

for each problem. Thus the earlier partial solution has not only been used as a basis for limiting later constraint checking, but also in quickly eliminating some domain values for each of the new-level variable additions. In our example, Figure 2 shows a partial solution where some template slots 1, 2 and 3 are matched to situation elements Units A, B and C respectively. Before search continues, we apply our spatial locality heuristic derived from A, B and C assignment positions, limiting the area of search for the remaining slots as shown in the shaded area of Figure 3. Consequently the only candidate instances in the selected area for the next level slots are *j, k, v, l, r, d s and e* , while the others outside this area are excluded.

Other restrictions are certainly possible, such as one mirroring the discussed relaxation in Section 3.1 where constraints might be tightened thus limiting the number of candidate solutions. In Woods (1994) the use of abstraction as a relaxation and specification as a restriction is discussed further.

4 Isomorphism Formulations

Given two graphs we may wish to find a correspondence between the graphs according to certain restrictions. For instance, Graph Isomorphism (GI) might be used to define the structural equivalence between two graphs, Subgraph Isomorphism (SGI) might be used to define the structural equivalence of part of one graph with another graph, and Graph Automorphism (GA) might be used to define a certain type of useful structural symmetry in a graph. GI has never been shown to be NP-Complete or a member of P, and GI along with problems that are polynomial-time reducible to GI are shown as belonging to the space GI-Complete. Figure 4 outlines some hypothetical complexity spaces in NP. Lubiw (1981) points out that earlier work has shown that if, in fact, P $\neq$ NP then there exist problems in NP which are neither in P nor NP-Complete, and this fact could possibly explain the lack of a classification of GI in P or NP, if GI belonged to such a class[4]. GI-Complete problems are shown to be "harder" than P since no GI-Complete problem has been shown to be in P, and "easier" than NP-Complete for a similar reason.

We are interested in how STR with restrictions or STR with relaxations corresponds to these problems. Certainly generalized STR as CSP is NP-Complete. But what of "STR-spatial-restrict" or "STR-partial" ?

Figure 5 details four hypothetical complexity spaces in NP: P, Graph Automorphism Complete (GA-Complete) intersect NP, Graph Isomorphism Complete (GI-Complete) intersect NP, and NP-Complete. I will outline each of these problems below and discuss the key aspects of each that places it in a class other than P or NP-Complete by discussing the boundaries or at least the changes that shift a problem across these boundaries[5]. In this figure,

[4]It should also be noted that if it is the case that problems are proved to be between P and NP-Complete such as GI-Complete implies, then there will also be problems in NP strictly between P and GI-Complete, and we have labelled these (or at least some of these) with GA-Complete in Figure 5. In addition, Hoffmann (1982) examines some generalizations of GI which appear to be harder than GI and yet are not known to be NP-complete, implying the further existence of a class between GI-Complete and NP-Complete. This is labelled on our Figure 5 as "Hoffman Space".

[5]It is important to note that these *boundaries* are not known to exist, only that they

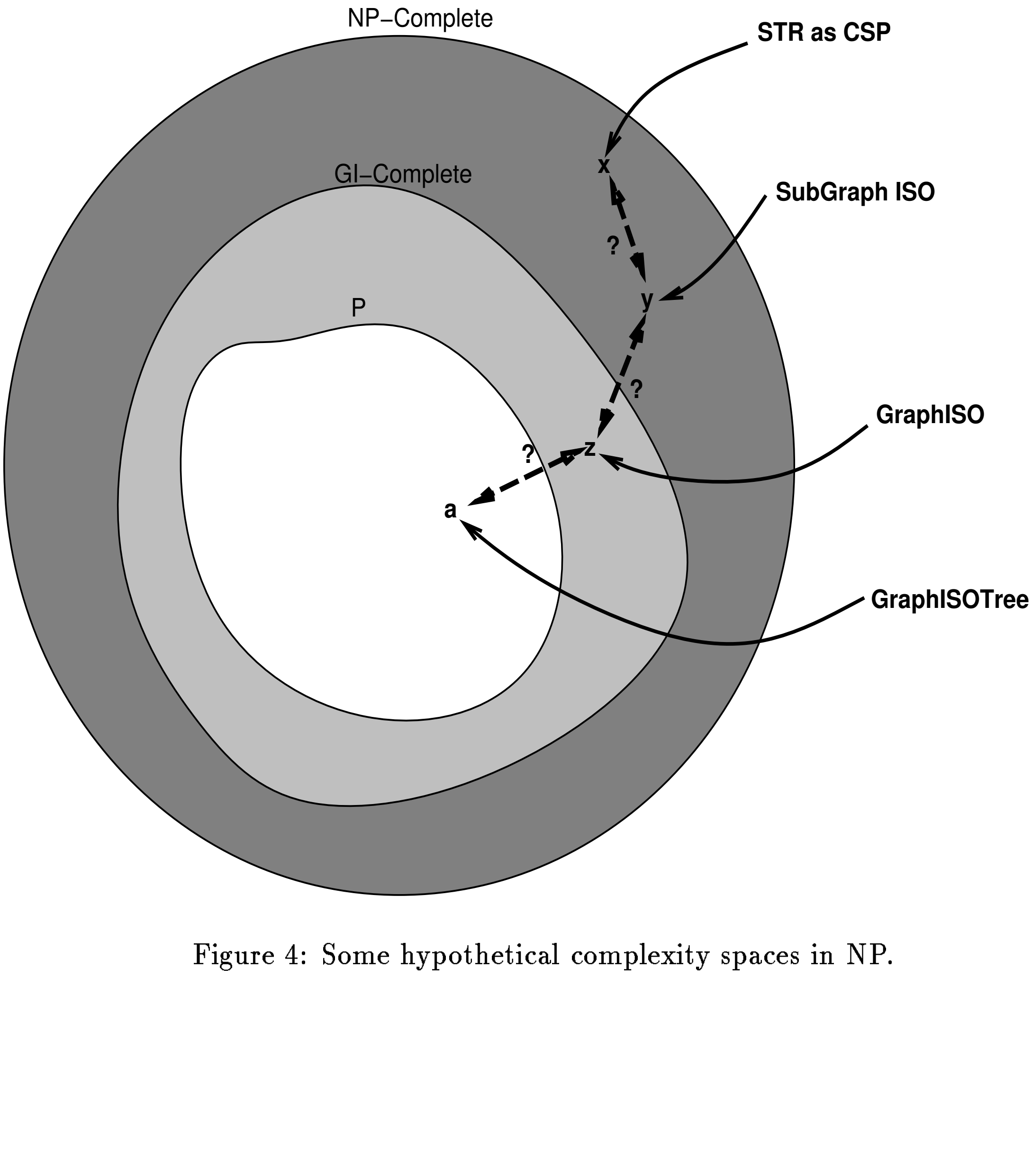

Figure 4: Some hypothetical complexity spaces in NP.

we have labelled Graph Automorphism "GA", and those problems which are polynomially reducible to GA are shown as belonging to the class GA-Complete. SGI represents Subgraph Isomorphism, while GI-bipartite, GI-regular, GI-line, GI-tree, GI-planar, and GI-interval all represent restrictions of GI where the graphs belong to the graph type specified.

4.1 Graph Isomorphism (GI)

The *Graph Isomorphism* (GI) problem is essentially to determine whether two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are isomorphic, that is, is there a mapping function (bijection) f between G and H such that (u, v) is an edge in G if and only if $(f(u), f(v)$ is an edge in H.

Graph Isomorphism has been observed (Lubiw 1981, Hoffmann 1982) to possess certain structural qualities[6] that keep it out of NP-Complete in general. Further, the destruction of these structural qualities through either certain types of relaxation or restriction results in a problem in NP-Complete. For instance, a generalization of GI to Subgraph Isomorphism (SGI) results in a destruction of GI's "structure', more freedom in mapping the initial graph to the target graph, and an NP-Complete problem. The structure "destroyed" in this case is related to the previously described equivalence relation between graph vertices - essentially the question of which vertices to map to one another is further complicated by the selection of which subgraph to attempt the mapping on.

---

may exist. Essentially then, the four classes obey the following relationship: $P <= GA - Complete <= GI - Complete <= NP - Complete$, and the inequalities could possibly be strict.

[6]Lubiw (1981) and Hoffmann (1982) point out that this structural quality is difficult to determine, but observes the following: The counting versions of all known NP-Complete problems belong to a class apparently not polynomial time equivalent to NP-Complete. However, the Automorphisms of a graph G induce and equivalence relation $==$ on the vertices of the graph. Specifically, two vertices, a and b obey $a == b$ if there is an Automorphism A of G such that $A(u) = v$. The set of equivalence classes of this relation $==$ makes up an automorphism partition of the graph, and finding this partition is known to be GI-Complete. With this fact it can be shown that the problem of counting the number of isomorphisms between two graphs is GI-Complete. Thus there are essentially two evidences that GI-Complete is not equivalent to NP-Complete: first, no know GI-Complete problem is known to belong to NP-Complete (despite much interest in the problem), and second, known NP-Complete problems have counting versions that seem to be significantly more complex than the counting versions of GI-Complete problems.

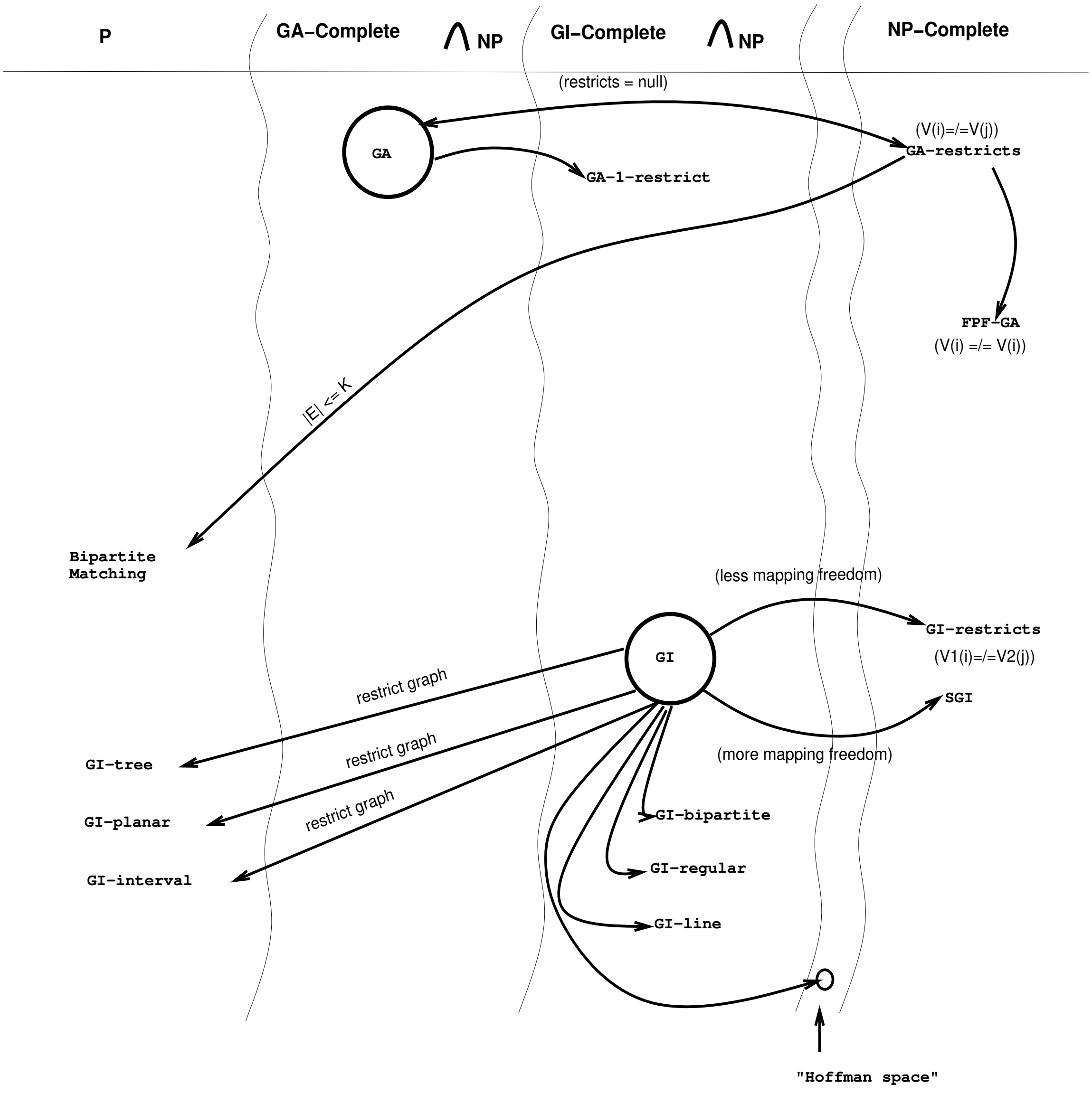

Figure 5: Some complexity spaces and problems possibly not NP-Complete.

Interestingly, GI can be made NP-Complete with either restriction and less mapping freedom amongst the graphs as in the case with GI with general restrictions allowed on the possible mappings between the two graphs, or with relaxation and more mapping freedom as in the case with SGI.

GI is clearly related to STR according to the search for an equivalence relation, but seems simpler in a sense. Specifically the fact that the graphs in a GI success are of the same "size" with respect to vertices and edges seems counter to the conception of a relatively small template in a large situation. SGI would seem to fit this conception better. Further, in a complete identification of template instances we are interested in finding *all* equivalence relations, not just determining the existence of one. It would appear that STR corresponds to the counting version of this problem. However, as we have seen, the counting version of problems in GI-Complete are GI-Complete themselves! This result is quite different from NP-Complete problems where the "counting" complexity is more than polynomially worse than NP-Complete[7]. We would certainly prefer to locate STR in some less onerous class. Unfortunately, the consideration of mapping freedom with GI-restricts seems to also parallel STR. Perhaps some form of isomorphism such as Subgraph Isomorphism with restrictions fits best, however, this problem would appear to be NP-Complete also.

Several restrictions of GI are known to be in P. Specific restriction of the structures of the graphs involved such as planar or tree have known polynomial solutions. This feature could be an important starting point for future template specification or refinement efforts[8].

[7] The complexity of counting versions of NP-Complete problems is referred to as #P-Complete.

[8] Essentially, if it was known that a graph that was more "tree-like" in a sense was easier to locate an equivalence relation for than a more general graph, then preference could be made for templates of this kind. Further, the "triggers" or cues to failure used in the polynomial time GI-tree algorithm could potentially suggest similar strategies for "tree-like" templates. Other work in pattern recognition (both parallel and serial) has focused on these critical "features" or sets of features as points of discrimination in template instance discovery, and while our problem is considerably more general than that, perhaps analogies could be exploited.

4.2 Subgraph Isomorphism (SGI)

SGI is a generalization of GI. Specifically, we are interested in asking not if two graphs are isomorphic, but rather if one graph possess a subgraph isomorphic to the other. SGI is known to be NP-Complete. As we have already stated, SGI would appear to model STR rather well if the question about the way in which constraints are mapped to arcs is ignored. If the situation reflects one graph and a template the other, then the graph represented by the situation possesses "virtual" arcs that are only evaluated during the actual equivalence determination, and in fact the "values" or even existence of these arcs is then dependent on the selection of the subgraph of the situation graph that is being matched to the template graph. For instance, if two slots "A" and "B" are joined with a distance arc, and slot "A" is mapped to a situation element "a", then the attempt to map slot "B" to some situation element "b" would only result in an equivalence (so-far) if the determined distance between "a" and "b" where "equal" to the distance arc between "A" and "B". If the arc represented some other attribute (such as type equivalence between "A" and "B") then the arc evaluation between "a" and "b" would be implicit until evaluated. The point is that we will have to consider graphs in STR that have multiple constraint arcs between nodes, and further, in solving STR, arcs will have to be evaluated dependent on the already completed element-to-slot assignments.

4.3 Graph Automorphism (GA)

Similar to Graph Isomorphism, we are searching for an equivalence relationship between a graph and itself. GA is not known to be in P, and in fact has never even been shown to be as hard as GI-Complete. Lubiw (1981) shows GA with 1 restriction[9] to be GI-Complete. GA with arbitrary restrictions is also shown in Lubiw (1981) to be NP-Complete. Additionally, a similar problem known as Fixed Point Free Automorphism (FPF-GA) where no vertex may be mapped to itself in the equivalence relation is also shown to be NP-Complete.

This problem is interesting with respect to STR based on the observation

[9]A restriction in GI or GA is essentially deemed to be a prevention of an equivalence being drawn between two vertices, one in each graph. For example a restriction might be of the form: "V_1 in V_H of graph H may not be mapped to V_6 in V_G of graph G".

made in Section 4.1 that equivalence relations are induced by automorphisms of a graph G. Prabhu & Narsingh (1984) points out that the number of isomorphisms between 2 graphs is equal to the order to the automorphism group of either one. Thus, if there exists an isomorphism I between two graphs G and H, then for any automorphism A of G, I composed with A is also an isomorphism between G and H. Essentially a symmetry is identified in G as an automorphism. This symmetry could potentially be exploited in some way in future STR formulations, either on occurrence or intentionally in advance formulation of templates, since we know that GA is possibly "easier" to solve than GI.

4.4 Isomorphism Problems and STR

The STR problem is about locating structural equivalence, however, between a particular form of attributed graph and a situation rather than between two arbitrary graphs. If we consider that the situation objects in a situation form the nodes of some complete graph, where the arcs between each node represent either the spatial relationship of the nodes (such as distance for example), or some other relevant relationship that can be measured, then, in a decision about the existence of an instance of a spatial template in a situation, we are looking at an equivalence relation between graphs. The situation "graph" arcs may in fact not be "evaluated" until such time as a check is made for equivalence with a template arc.

We know that GI for restricted graph forms (such as tree and planar) is polynomial. Consequently the problem complexity is not immediately apparent if only one of the graphs is restricted to one of these classes[10]. In such a problem, we might restrict our search to sub-parts of a template that possess one of the properties, identify these parts and re-assemble these partial solutions into larger solutions in a divide and conquer type of approach. In Woods (1993*a*), the template problem is discussed in terms of two different problem decomposition approaches: abstraction, and partition. In the partition approach, a template is divided into "key" subproblems, such as the case of an airplane being divided into two broad parts "wing structure" and "body structure". Solutions are found for each part, and a cross product

[10] If a complexity result is known for disjoint-type Graph Isomorphism, I am not aware of it.

match is attempted with all possible combinations according to additional locality restrictions. Conceivably these structures could be formed utilizing templates of restricted form, thus simplifying each subproblem.

Lubiw (1981) and Hoffmann (1982) describe several *generalizations* of GI which allow more freedom in mapping the initial graph to the target graph and Lubiw (1981) describes several *restrictions* on the possible mappings between initial and target which can result in problems that are NP-Complete. Johnson (1982) adds other examples. This suggests that an inherent structure of the GI problem is being destroyed in either related problem case, and it is precisely this structure which makes GI easier than either generalized versions such as SGI, or restricted versions such as **Isomorphism with Restrictions** (GI-restricts). The GI structure was destroyed in the relaxation to SGI by the additional complexity of selecting a subgraph to match, and also in the restriction to GI-restricts (Johnson 1982). Lubiw (1981) indicates that other similar structural destructions occur in restricting GA to GA-restricts when the equivalence relation may no longer be reflexive, symmetric, or transitive. In FPF-GA only reflexivity is destroyed. In essence, in each of these cases the "isomorphism" is eroded enough that the *counting version* of the problem in question becomes #P-Complete.

The following are typical of STR relaxations:

- Relaxation of STR as CSP through CSP relaxation as defined in Freuder & Wallace (1992) by removing template variables or slots, relaxing constraints, or removing constraints, resulting in partial solutions.

- Situational locational (or other attribute) expansion where the identified possible range of a situational element expands results in an expansion of the size of the candidate solution set[11].

The relaxations essentially seem to increase the problem complexity in general. Implicit in the solution approach of Woods (1993*a*), where abstraction is utilized in the problem representation and solution approach, is a type of *Upward Refinement Property* where a specific solution is known to have had an abstract solution, and in this way a set of abstract solutions identifies a space of specific solutions that contains all of the possible specific solutions.

[11]A candidate solution set is essentially the possible mappings exist from template to situation.

These specific "spaces" are much smaller than the entire space. It would not appear that "small doses" of these relaxations are going to benefit the solution of STR as we have formulated the problem. However, a completely relaxed constraint set would imply a trivial although huge solution set, and a very small constraint set would imply a only slightly smaller solution set.

The following are typical of STR restrictions:

- Restriction of CSP constraints resulting in a much smaller candidate set.
- Reduction in the locational (or other attribute) range in the situational elements resulting in a much smaller candidate set.
- Introducing the concept of spatial locality thus reducing the complexity of a problem after some "part" has been solved.

Spatial locality is essentially a precompilation of all constraints into a type of boundary condition for future search. It allows dramatic pruning of the search space and quick location of dead ends. It does not decrease the overall breadth of a search tree since it may only be applied to a local "partial" remaining subproblem, however, search depth may be dramatically reduced. In general, however, as the spatial "difference" between a template and a situation is seen to decrease, we see a decreasing effect. Interestingly, this size difference reduction corresponds to the case in the isomorphism problem where the larger and smaller graphs in the Subgraph Isomorphism problem approach one another and form the Graph Isomorphism problem which is likely *easier*. However, it is not obvious how to approach this problem aspect since it is not clear to me how to map a "spatial locality" heuristic to a general graph problem.

The work of Manning (1991) and Eades & Ng (1987) is particularly interesting to STR problems since they deal with visual graph representation problems that exhibit properties similar to STR in terms of spatial relationships. For example, in (Manning 1991) it is shown that detecting axial or rotational symmetry in a graph is NP-Complete. A particular "drawing" of a graph is shown that demonstrates various properties inherent in a graph (such as planarity, biconnectivity, symmetry, or diameter) while in another drawing these properties are not so "obvious". Manning (1991) is concerned with drawing graphs that display these properties, and to this end one would like to find an efficient algorithm for detecting symmetries in abstract graphs. Similarly, since it is known that STR is solvable much more readily by an expert when distinctive characteristics such as symmetry are present in a template and situation, this NP-Complete result for symmetry detection suggests that restricting STR to certain subclasses such as those with symmetry may not be any easier in general, although an optimal algorithm for all symmetry detection for points and lines in two dimensions is presented in Eades & Ng (1987). However, it is not hard to imagine an application where pattern recognition techniques could be used as a "preprocessor" based on some critical template aspects to suggest some "obvious" starting points for the actual searching for equivalence relations.

4.5 Other Related Work

Stacey & Wong (1988) present an algorithm for determining the "largest common subgraph isomorphism in attributed graphs". This work has derived from an attempt to make efficient algorithms for the pattern recognition problem which is essentially a special case of STR with exact, rigid templates. The importance of "largest subgraph" is a little uncertain (as the authors themselves point out), although it would appear that the commonality represents a kind of "best-fit" measure between situation graph and template graph. The algorithm presented is based on a depth-first exhaustive search that makes use of local contextual and structural information of the template representation graph. In this work they identify that investigating additional exploitation of attribute information (how about spatial locality ?) in search as a future enhancement. It is interesting to note that the authors observed in experiments that increasing their constraint tolerance (paralleling an STR relaxation) resulted in an expanded search space (as I predicted earlier), and no additional solutions (as would follow from situational and template rigidity and little symmetry). Additionally, the authors point out that symmetry in their situation yielded ambiguous solutions that were unresolvable without further attribute information, nicely demonstrating the effect of a graph automorphism creating additional isomorphisms.

5 Conclusions

What may we conjecture about the complexity of STR? We have made a little progress from our investigation of isomorphism problems, primarily that in general, the problem of identifying a single template instance would appear to be mappable to no better than Subgraph Isomorphism in general, and consequently would be NP-Complete for any instance, and #P-Complete for all instances. However, the ramifications of the spatial heuristic in some kind of average, typical or common case would be interesting to investigate in some future work, particularly in that since STR has immediate application potential we are far interested in cases that can possibly occur! The spatial heuristic is limited in its "effectiveness" by the difference in size between a template and a situation, where the closer the template gets in maximum extent to a situation, the less pruning may occur. It is interesting to note that in a similar graph isomorphic sense, it would seem that the closer two graphs become to one another in size the closer the problem is to GI-Complete (as the number of selections of subgraphs decreases), while the more they differ the closer the problem becomes to NP-Complete (as the number of selections of subgraphs increases). Towards efficient implementations of STR in specific domains, the properties of templates can undoubtedly be exploited - both with respect to typifying the common and unique features of templates and extending the spatial heuristic to be more accurate in specific cases, and in attempting to model templates that offer either an improved problem complexity due to their simplified structure, or allow for the structuring of intelligent heuristics based on partitioning templates into more simply structured portions in a divide and conquer strategy.

Acknowledgments

The work and investigation described in this report has been prepared specifically for the requirements of CS666, however, it follows from work undertaken by the author for the Department of National Defense(ND), Defence Research Establishment Valcartier. The author is a PhD candidate supported by an NSERC scholarship and a research contract with DND.

References

Booth, Kellogg, S. & Colbourn, Charles, J. (1977), Problems polynomially equivalent to graph isomorphism, Technical Report CS-77-04, University of Waterloo.

Cormen, T. H., Leiserson, C. E. & Rivest, Ronald, L. (1992), *Intoduction to Algorithms*, McGraw-Hill Book Company.

Eades, P. & Ng, H. C. (1987), 'An algorithm for detecting symmetries in drawings', *ARS Combinatoria* **23A**, 95–104.

Freuder, E. & Wallace, J. (1992), 'Partial constraint satisfaction', *Artificial Intelligence* **58**, 21–70.

Garey, M. R. & Johnson, D. S. (1979), *Computers and Intractability : A guide to the theory of NP-Completeness*, W. H. Freeman and Company, Bell Laboratories, Murray Hill, New Jersey.

Ghahraman, D., Wong, A. & Tung, A. (1980), 'Graph optimal monomorphism algorithms', *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-10**(4), 181–188.

Hoffmann, C. (1982), 'Subcomplete generalizations of graph isomorphism', *Journal of Computer and System Sciences* **25**, 332–359.

Johnson, D. (1982), 'The NP-completeness column - an ongoing guide', *Journal of Algorithms* **3**(3), 288–300.

Kumar, V. (1992), 'Algorithms for constraint-satisfaction problems', *AI Magazine* pp. 32–44.

Lapointe, S. & Proulx, R. (1994), Statistical identification of templates in message data, Contract Report W7701-3-0847/01-XSK, APG inc. (1994). Combat Intelligence Automation Group, Defence Research Establishment Valcartier.

Lubiw, A. (1981), 'Some NP-complete problems similar to graph isomorphism.', *SIAM Journal of Compting* **10**, 11–21.

Manning, J. (1991), 'Computational complexity of geometric symmetry detection in graphs', *Lectures in Computer Science.*

Nadel, B. A. (1989), 'Constraint satisfaction algorithms', *Computational Intelligence* **5**, 188–224.

Prabhu, G. & Narsingh, D. (1984), 'The ellipsoid algorithm and the graph isomorphism problem', *Technique et Science Informatiques* **3**, 327–333. in English.

Stacey, D. & Wong, A. (1988), A largest common subgraph isomorphism algorithm for attributed graphs, University of Waterloo, Systems Design Engineering.

Wong, A., You, M. & Chan, S. (1990), 'An algorithm for graph otpimal monomorphism', *IEEE Transactions on Systems, Man, and Cybernetics* **20**(3), 628–636.

Woods, S. (1993*a*), Interactive recognition of spatially defined model deployment templates, Research memorandum, Combat Intelligence Automation Group, Defence Research Establishment Valcartier, Courcelette, Quebec CANADA. unclassified.

Woods, S. (1993*b*), A method of interactive recognition of spatially defined model deployment templates using abstraction, *in* 'Proceedings of the 1993 DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics'.

Woods, S. (1994), Learning refinements to spatial template models based on false-positive and false-negative examples, advance copy, technical report.