Tractable Class of a Problem of Goal Satisfaction in Mutual Exclusion Network *

Pavel Surynek

Charles University
Faculty of Mathematics and Physics
Malostranské náměstí 2/25, 118 00 Praha 1, Czech Republic
surynek@ktiml.mff.cuni.cz

Abstract

In this paper we describe a class of a problem of goal satisfaction in mutual exclusion network that can be solved in polynomial time. This problem provides a common basis for reasoning about various tasks known from artificial intelligence. Namely, tasks arising during construction of concurrent solutions for planning problems and Boolean formula satisfaction can be viewed as problems of goal satisfaction. We experimentally compared a solving algorithm which exploits the defined tractable class with backtracking enhanced by maintaining consistencies on random problems and on problems arising in concurrent planning. We obtained significant speedups in both experimental setups.

Introduction and Motivation

We are dealing with a class of a problem of *goal satisfaction in mutual exclusion network*. The problem was originally introduced in (Surynek, 2007a). The author states that generally the problem is *NP*-complete. However, together with this negative result a practically efficient technique for solving this problem is proposed by the author. We are going further with this problem in this paper. We study a special class of the problem for that we design a polynomial time solving algorithm - that is, we describe a tractable class of the problem.

Briefly said, the mutual exclusion network is an undirected graph where each vertex has assigned a finite set of symbols. The interpretation of edges is that a pair of vertices connected by an edge cannot be selected together. Having a goal, which is a finite set of symbols, the task is to select a stable set of vertices in this graph such that the union of their symbols covers the given goal.

The solving process for the problem proposed in (Surynek, 2007a) is based on maintaining a special type of global consistency. The proposed consistency is trying to exploit structural information encoded in the problem. It is argued that valuable structural information in the mutual exclusion network is the knowledge of *complete subgraphs* (cliques). If we know that several vertices in the graph form a clique we also know that at most one of them can be selected into the solution. This simple property allows us to do further reasoning about the problem.

The problem of goal satisfaction in mutex network is motivated by two areas of artificial intelligence research. First it is motivated by *concurrent planning* (Blum and Furst, 1997) and second it is motivated by *Boolean formula satisfaction* (Cook, 1971).

In the basic variant of planning problems (Ghallab et al., 2004) we are searching for a sequence of actions that, when executed one by one starting in the given planning world, results into the planning world that satisfies certain goal. The concurrent planning itself represents a generalization of this basic variant. Particularly, we allow more than one action to be executed in a single time step in concurrent planning. This generalization is motivated by the fact that certain actions do not interfere with each other and they can be executed simultaneously without influencing each other. The frequently asked question which arise during solving process of algorithms for concurrent planning is "What are the sets (or is there any) of non-interfering actions that satisfies certain goal?" This problem can be reformulated as a goal satisfaction problem in mutex network.

Another motivation to study the concept of mutual exclusion networks is Boolean formula satisfiability (Surynek, 2007a). A Boolean formula satisfaction problem (SAT problem) can be modeled as a goal satisfaction problem in mutex network. The application of basic solving technique for goal satisfaction problems on SAT problems is studied in (Surynek, 2007b). Nevertheless, we do not study the goal satisfaction problem from the SAT perspective in this paper. This is just to mention another application than concurrent planning.

This paper is organized as follows. First we recall the basic concepts - mutual exclusion networks, goal satisfaction problem, and associated global consistency technique. Upon these preliminaries we describe a class of the goal satisfaction problem that can be solved in polynomial time. Next we evaluate an algorithm exploiting the tractable

Compilation copyright © 2008, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

* This work is supported by the Czech Science Foundation under the contracts 201/07/0205 and 201/05/H014. I would like to thank anonymous reviewers for many useful comments.

class of the problem in comparison with existing comparable techniques. Finally, we briefly comment some similar works and summarize our contribution.

Mutual Exclusion Networks

The following definitions formalize mutual exclusion network (shortly *mutex network*) and the associated *problem of satisfying goals* in the mutex network. We assume a finite universe of symbols S for the following definitions.

Definition 1 (Mutual exclusion network). Mutual exclusion network is an undirected graph $N=(V,M)$, where each vertex $v \in V$ has assigned a finite set of symbols $\emptyset \neq S(v) \subseteq S$. □

Definition 2 (Goal satisfaction in mutex network). Given a goal $G \subseteq S$ and a mutex network $N=(V,M)$ the problem of satisfying goal G in the mutex network N is the task of finding a stable sub-set of vertices $U \subseteq V$ such that $G \subseteq \bigcup_{u \in U} S(u)$. □

The problem of goal satisfaction in mutex network is computationally hard. It is not difficult to show that the problem is *NP*-complete. The proof is given in (Surynek, 2007a). In the light of this result there is little hope to solve the problem of goal satisfaction in mutex network effectively (in polynomial time). So, the search seems to be the only option to solve the problem systematically. However, the search may be more or less informed where the more informed search leads to the lower number to steps required for obtaining a solution.

Making the Search More Informed

In (Surynek, 2007a) the author studies the usage of maintaining consistencies to make the search more informed. Maintaining consistency means that certain type of consistency is enforced in the problem after each decision step. Enforcing consistency in the problem may reduce the size of the remaining search space and the problem can be solved faster as a consequence.

Existing consistency techniques differ in their inference strength and resource requirements. The expectable requirement is that the consistency should be strongest as possible while its resource requirements should be low. These two requirements usually go against each other. Therefore a trade-off is necessary to be found. In (Surynek, 2007a) the author studied the usage of relatively efficient local consistency - *arc-consistency* (Mackworth, 1977), and he proposed a new global consistency called *projection consistency* for solving the goal satisfaction problems in mutex network. Both consistencies satisfy the low resource requirements. However, projection consistency is more efficient on the studied problem since it exploits advantage of considering the whole problem at once while arc-consistency is considering only a small part of the problem at a time.

We further develop the concept of projection consistency here. Using projection consistency we define a class of the problem of goal satisfaction in mutex network that can be solved in polynomial time. In the following text we refer to this class of the problem as a *tractable class*.

Global Consistency for Mutex Network

The idea of projection consistency is based on a property of mutex networks arising in AI applications mentioned in the introduction. Such mutex networks are usually well structured. Specifically, they can be characterized as a relatively small number of large complete sub-graphs plus small number of edges not belonging into these cliques. The intuitive explanation of arising of such a structure is that problems typically reflect situations from the real three-dimensional world. There often appear pair-vise conflicting sets of properties of objects in the real world that induces a clique of mutual exclusions (for example no two cars can occupy the same place at the same time).

If we explicitly know the structure of the mutex network we can reason about the impact of the vertex selection on the possibility of goal satisfaction. We know that at most one vertex from each clique can be selected to contribute to the satisfaction of the goal. Hence, for each clique we can calculate the maximum number of symbols of the goal which can be covered by the vertices of the clique. When we select a vertex into the solution the necessary condition on the solution existence is that the number of symbols covered by the remaining cliques together with symbols associated with the selected vertex must not be lower than the number of symbols in the goal.

The second part of the idea of projection consistency is that if we restrict ourselves on the proper subset of the goal the set of vertices ruled out by the above counting arguments can be different. Therefore it is possible to perform filtration by the technique with respect to multiple sub-goals of the goal to achieve the maximum pruning power - these sub-goals are called *projection goals*.

For the following formal description of projection consistency we assume that a clique decomposition $V = C_1 \cup C_2 \cup \ldots \cup C_n$ of the mutex network $N=(V,M)$ was constructed (greedily, since we cannot afford optimality). Each C_i is a clique and $C_i \cap C_j = \emptyset$ for $i \neq j$. Projection consistency is defined over the above clique decomposition for a projection goal $\emptyset \neq P \subseteq G$. The fact that at most one vertex from each clique can be selected into the solution allows us to introduce the following definition.

Definition 3 (Clique contribution). A *contribution of a clique* $C \in \{C_1, C_2, \ldots, C_n\}$ to the projection goal $\emptyset \neq P \subseteq G$ is defined as $\max(|S(v) \cap P| \mid v \in C)$ and it is denoted as $c(C,P)$. □

The concept of clique contribution is helpful when we are trying to decide whether it is possible to satisfy the projection goal by selecting the vertices from the clique cover. If for instance $\sum_{i=1}^{n} c(C_i,P) < |P|$ holds then the projection goal P cannot be satisfied. Nevertheless, the projection consistency can handle a more general case as it is described in the following definitions.

Definition 4 (Supported vertex). A vertex $v \in C_i$ for $i \in \{1,2,\ldots,n\}$ is *supported* with respect to a given clique decomposition and the projection goal P if $\sum_{j=1,\, j\neq i}^{n} c(C_j, P) \geq |P - S(v)|$ holds. □

Definition 5 (Consistent problem). An instance of the problem of satisfaction of a goal G in a mutex network $N = (V, M)$ is consistent with respect to the given clique decomposition and the projection goal $\emptyset \neq P \subseteq G$ if every vertex $v \in C_i$ for $i = 1,2,\ldots,n$ is supported with respect to the given clique decomposition and the given projection goal. □

It is easy to observe that projection consistency is a necessary but not sufficient condition on existence of the solution (Surynek, 2007a).

To ensure maximum vertex filtration effect we can enforce the consistency with respect to multiple projection goals. However, it is not possible to use all the projection goals since they are too many (exactly $2^{|G|}$). Our experiments showed that projection goals consisting of symbols with the constant number of supporting vertices provide satisfactory filtration effect. The number of projection goals of this form is linear in the size of the goal G.

Tractable Case

It is possible to make projection consistency stronger by a slight reformulation of the definition of the supported vertex. The definition of the consistent problem remains the same. We will require the stronger version of consistency to be able to solve certain instances of goal satisfaction problem in polynomial time.

Definition 6 (Strongly supported vertex). A vertex $v \in C_i$ for $i \in \{1,2,\ldots,n\}$ is *strongly supported* with respect to a given clique decomposition and the projection goal P if $\sum_{j=1,\, j\neq i}^{n} c(C_j, P - S(v)) \geq |P - S(v)|$ holds. □

Let us call the projection consistency that uses the definition of strongly supported vertices a *strong projection consistency.* Observe that strong projection consistency is strictly stronger than projection consistency. That is, there exists a goal satisfaction problem which is projection consistent with respect to a projection goal P and it is not strongly projection consistent with respect to the same projection goal P.

In the following series of definitions and lemmas we will gradually develop a polynomial time algorithm for a class of the goal satisfaction problem. For the following definitions we assume that a set of symbols $S(C) = \bigcup_{v \in C} S(v)$ is assigned to each clique C.

Definition 7 (Clique intersection graph). We define a *clique intersection graph* $G_I^c = (\{C_1, C_2, \ldots, C_n\}, E_I^c)$ for the clique decomposition $V = C_1 \cup C_2 \cup \ldots \cup C_n$ as an undirected intersection graph of corresponding clique symbols. That is $E_I^c = \{\{C_i, C_j\} \mid i \neq j \;\&\; S(C_i) \cap S(C_j) \neq \emptyset\}$. □

Lemma 1 (Tractable case: clique intersection graph). Let $V = C_1 \cup C_2 \cup \ldots \cup C_n$ be a clique decomposition of the mutex network and let G be a goal we want to satisfy. Next let $G_I^c = (V_I^c, E_I^c)$ be the corresponding clique intersection graph. If the graph G_I^c is acyclic then a problem of satisfying the goal G by selecting just one vertex v_i from the clique C_i for every $i = \{1,2,\ldots,n\}$ into the solution can be solved in polynomial time after enforcing strong projection consistency. ■

Proof. We need to show that if the defined problem is strong projection consistent with respect to the certain projection goals then it is necessary to do only little to find a solution or to conclude that there is no solution. The projection goals we use are $G \cap (S(C_i) - \bigcup_{j=1,\, j\neq i}^{n} S(C_j))$ for every $i = \{1,2,\ldots,n\}$ and $G \cap S(C_i) \cap S(C_j)$ for every $\{C_i, C_j\} \in E_I^c$. If $G - \bigcup_{i=1}^{n} S(C_i) \neq \emptyset$ holds then there is obviously no solution. This condition can be checked in $O(|G||V|)$ steps.

If $G \subseteq \bigcup_{i=1}^{n} S(C_i)$ holds then arbitrary selection of just one vertex v_i from the clique C_i for every $i = \{1,2,\ldots,n\}$, which preserves relation of strong supports over the edges E_I^c, solves the problem. This selection can be carried out by starting in the root clique of G_I^c and continuing to the leaves in breadth first order. It takes $O(|G||V|)$ steps to select vertices in this way.

Consider a symbol $s \in G$. There are at most two cliques for which the symbol s is an element of their symbols. This is due to the acyclicity of the corresponding clique intersection graph G_I^c. In the case when there is just one such clique C_i a vertex $v_i \in C_i$ that satisfies s must be selected. Let $P = G \cap (S(C_i) - \bigcup_{j=1,\, j\neq i}^{n} S(C_j))$, for such P we have $s \in P$ and $\sum_{j=1,\, j\neq i}^{n} c(C_j, P - S(v_i)) \geq |P - S(v_i)|$ since the problem is strong projection consistent with respect to the projection goal P. We also have $\sum_{j=1,\, j\neq i}^{n} c(C_j, P - S(v_i)) = 0$ since the sum is empty (no other clique intersects the projection goal P by their symbols). Hence $|P - S(v_i)| = 0$ and $s \in S(v_i)$.

In the case when there are two cliques C_i and C_j for which $s \in S(C_i)$ and $s \in S(C_j)$. Suppose that a vertex v_i is selected from the clique C_i and a vertex v_j from the clique C_j. Consider the projection goal $P = G \cap S(C_i) \cap S(C_j)$, both vertices are strongly supported with respect to P. That is $\sum_{k=1,k\neq i}^{n} c(C_k, P - S(v_i)) \geq |P - S(v_i)|$ and $\sum_{k=1,k\neq j}^{n} c(C_k, P - S(v_j)) \geq |P - S(v_j)|$. Suppose that action v_i was selected before v_j. Since there are only two cliques interfering over the projection goal P, we specially have $c(C_j, P - S(v_i)) \geq |P - S(v_i)|$ after selecting v_i. Hence it is possible to select the vertex v_i such that $|P \cap S(v_j)| = c(C_j, P - S(v_i))$. Altogether we obtained that $s \in S(v_i) \cup S(v_j)$. ■

The question arises whether the strong projection consistency with respect to the projection goals mentioned in the proof of the lemma 1 can be enforced over the acyclic problem in polynomial time. The answer is positive and we can conclude that the problem from the lemma 1 can be completely solved in polynomial time.

We use the similar idea as that is commonly used to enforce arc-consistency in an acyclic constraint network (Dechter, 2003). It is possible to enforce arc-consistency in such a network by enforcing directed arc-consistency in the

direction from the root to the leaves of the network and then from the leaves to the root. Almost the same can be done for the strong projection consistency. First we enforce the consistency for the projection goals $G \cap (S(C_i) - \bigcup_{j=1,\, j\neq i}^{n} S(C_j))$ for every $i = \{1,2,\ldots,n\}$ which is easy because no interference with other cliques occurs. Then cliques of the decomposition are ordered according to the breadth first search and the strong projection consistency is enforced over the edges of the intersection graph. It is done in the direction from the root to the leaves of the clique intersection graph first and then from the leaves to the root.

The complete algorithm is shown here as algorithm 1. Observe that the algorithm can be implemented to run in polynomial time in size of the input. The illustration of the filtration effect of the algorithm is shown in figure 1.

Algorithm 1: Strong projection consistency propagation algorithm for acyclic clique intersection graph

function *enforceStrongProjectionConsistency* $(G,\{C_1,C_2,\ldots,C_n\})$: **set**
1: **let** $G_I^c = (\{C_1,C_2,\ldots,C_n\}, E_I^c)$ be the clique intersection graph
2: **for** $i = 1,2,\ldots,n$ **do**
3: $\quad P \leftarrow G \cap (S(C_i) - \bigcup_{j=1,\, j\neq i}^{n} S(C_j))$
4: $\quad \{C_1,C_2,\ldots,C_n\} \leftarrow$ *propagateSProjection* $(P,\{C_1,C_2,\ldots,C_n\})$
5: $\pi \leftarrow$ *breadthFirstSearch* (G_I^c)
6: **for** $i = 1,2,\ldots,n$ **do**
7: $\quad$ **for each** $\{C_{\pi(i)},C_{\pi(j)}\} \in E_I^c$ such that $\pi(i) < \pi(j)$ **do**
8: $\quad\quad P \leftarrow G \cap S(C_{\pi(i)}) \cap S(C_{\pi(j)})$
9: $\quad\quad \{C_1,C_2,\ldots,C_n\} \leftarrow$ *propagateSProjection* $(P,\{C_1,C_2,\ldots,C_n\})$
10: **for** $i = n,n-1,\ldots,1$ **do**
11: $\quad$ **for each** $\{C_{\pi(i)},C_{\pi(j)}\} \in E_I^c$ such that $\pi(i) > \pi(j)$ **do**
12: $\quad\quad P \leftarrow G \cap S(C_{\pi(i)}) \cap S(C_{\pi(j)})$
13: $\quad\quad \{C_1,C_2,\ldots,C_n\} \leftarrow$ *propagateSProjection* $(P,\{C_1,C_2,\ldots,C_n\})$
14: **return** $\{C_1,C_2,\ldots,C_n\}$
function *propagateSProjection* $(P,\{C_1,C_2,\ldots,C_n\})$: **set**
15: **for** $i = 1,2,\ldots,n$ **do**
16: $\quad$ **for each** $v \in C_i$ **do**
17: $\quad\quad$ **if** $\sum_{j=1,\, j\neq i}^{n} c(C_j, P - S(v)) < |P - S(v)|$ **then**
18: $\quad\quad\quad C_i \leftarrow C_i - \{v\}$
19: **return** $\{C_1,C_2,\ldots,C_n\}$
function *breadthFirstSearch* $((\{u_1,u_2,\ldots,u_m\},E))$: **permutation**
20: $\pi \leftarrow$ breadth first search ordering of vertices $u_1,u_2,\ldots,u_m$
21: **return** π

Definition 8 (Mutex intersection graph). A *mutex intersection graph* for the clique decomposition $V = C_1 \cup C_2 \cup \ldots \cup C_n$ and for the set of edges m not belonging to any clique of the decomposition is a graph $G_I^m = (\{C_1,C_2,\ldots,C_n\}, E_I^m)$, where $E_I^m = \{\{C_i,C_j\} \mid i \neq j \,\&\, (\exists v_i \in C_i, \exists v_j \in C_j)\{v_i,v_j\} \in m\}$. □

Lemma 2 (Tractable case: mutex intersection graph). Let us have a clique decomposition of the mutex network $V = C_1 \cup C_2 \cup \ldots \cup C_n$. Let $G_I^m = (V_I^m, E_I^m)$ be the corresponding mutex intersection graph. If the graph G_I^m is acyclic then a problem of selecting just one vertex v_i from the clique C_i for every $i = \{1,2,\ldots,n\}$ such that no two selected vertices are connected by an edge from m can be solved in polynomial time. ■

Sketch of proof. Actually, this is a well known result from constraint programming. If each clique of the clique decomposition $V = C_1 \cup C_2 \cup \ldots \cup C_n$ is regarded as a CSP variable and edges of the set m are regarded as constraints then the defined problem of selecting an independent set of vertices is an acyclic constraint satisfaction problem. It is sufficient to enforce arc-consistency and label the variables in breadth first order to obtain a solution. More details about this result can be found in (Dechter, 2003). The solution can be obtained in polynomial time. ■

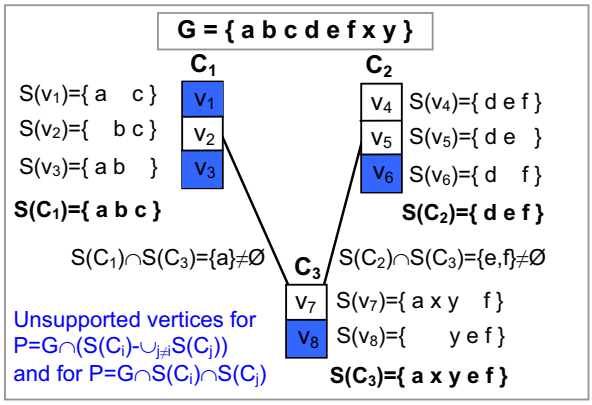

Figure 1: Illustration of strong projection consistency in a problem with acyclic clique intersection graph. Only the vertices that can be part of the solution remained.

Theorem 1 (Overall tractable case). Let us have a clique decomposition of the mutex network $V = C_1 \cup C_2 \cup \ldots \cup C_n$ and a set of edges outside the clique decomposition m. Let $G_I^c = (V_I^c, E_I^c)$ be the corresponding clique intersection graph and let $G_I^m = (V_I^m, E_I^m)$ be the corresponding mutex intersection graph. If the intersection graph $G_I = (\{C_1,C_2,\ldots,C_n\}, E_I^c \cup E_I^m)$ is acyclic then the corresponding goal satisfaction problem can be solved in polynomial time. ■

Sketch of proof. To prove the theorem we use a combination of results from lemma 1 and lemma 2. The first step consists of enforcing strong projection consistency and arc-consistency in the problem. Since it is quite easy using the above results we describe the process briefly. If the interference of cliques is through an edge from E_I^c then strong projection consistency is enforced over the intersection of corresponding clique symbols. If the interference of cliques is through an edge from E_I^m then arc-consistency with respect to m is enforced. Again this combined consistency can be enforced in polynomial time by proceeding from the root to the leaves of the graph G_I and conversely. The extraction of a solution from the consistent problem can be also done in polynomial time. The extraction procedure starts by selecting action from the root clique and proceeds to the leaves of the graph G_I while strong projection consistency and arc-consistency relations are preserved over the edges of G_I. The described solution extraction can be carried out in polynomial time. ■

We described the tractable class of the problem of goal satisfaction in mutex network in order to utilize the theo-

retical results in solving real problems. The obstacle is that not every instance of the problem belongs to the described class (not every problem induces acyclic intersection graph defined in theorem 1). However, each decision step (selecting a vertex into the solution) can be interpreted as removal of a clique from the graph. So, eventually the graph becomes acyclic. When the graph becomes acyclic it is possible to switch from search to the proposed polynomial time algorithm to solve the rest of the problem in backtrack-free manner. This is exactly the way how we exploited the tractable class in the solving algorithm.

Experimental Evaluation

Our experimental evaluation is concentrated on two aspects of the proposed tractable class. First we want to evaluate the solving procedure using the tractable class in comparison with the solving procedure using plain projection consistency. Additionally we compared both solving procedures with the solving procedure using arc-consistency. This set of experiments was done with random goal satisfaction problems in mutex networks.

Next we evaluated the usage of the proposed tractable class in producing concurrent solutions for planning problems. We carried out this evaluation by integrating the usage of tractable class into the GraphPlan based planning algorithm (Blum and Furst, 1997).

All the tested algorithms were implemented in C++ with equal coding style and were run on a machine with AMD Opteron 242 processor (1.6 GHz) and 1 GB of memory under Mandriva Linux 10.2. The code was compiled by gcc compiler version 3.4.3.

Random Goal Satisfaction Problems

Random goal satisfaction problems used in the evaluation were of the following setup motivated by the visual observation of problems arising in concurrent planning (several large cliques covered with some noise edges). In a mutex network consisting of 120 vertices we constructed several complete sub-graphs using uniform distribution with the mean value of 20.0. The size of the goal was 60 and each vertex has assigned a random set of symbols from the goal of the size generated by the normal distribution with the mean value of 16.0 and the standard deviation of 10.0. Finally we added random edges into the mutex network. More precisely, we add each possible edge into the mutex network with the probability of p where p was a variable parameter which ranged from 0.0 to 0.1. For each value of the parameter p we generated 10 goal satisfaction problems and we solved them using the tested techniques. Along the solving process we collected data such runtime, number of backtracks etc.

The common framework of all our tested solving algorithms was the standard backtracking. We compared the impact of usage of consistencies on the overall solving speed. The version using arc-consistency enforces arc-consistency in the problem after each decision step (that is we maintain consistency). The version that uses plain projection consistency is similar. It enforces projection consistency after each decision step. The version that exploits the tractable class again enforces plain projection consistency after each decision but additionally it detects acyclicity of the intersection graph. When the graph becomes acyclic the solving algorithm switches from search to the algorithm for solving the tractable case.

Part of the results obtained from these experimentations is shown in figure 2. We can observe that utilization of tractable case bring significant improvement compared to the plain projection consistency for the values of parameter p ranging from 0.0 to 0.03. For the higher probabilities of random edges there is almost no improvement. Nevertheless, our additional experimentation gives evidence that goal satisfaction problems arising in applications has similar graphical structure as random problem with low edge probability.

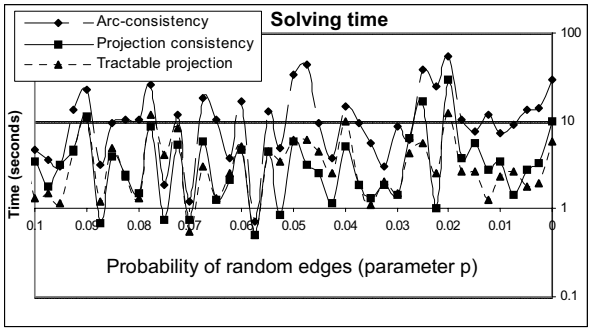

Figure 2: Runtime for random goal satisfaction problems (average of 10 problems for each value of random edge probability p). Time is shown in logarithmic scale.

Problems Arising in Concurrent Planning

We also evaluated the proposed tractable class in solving problems that arise in concurrent planning (that is in the area for which the projection consistency was designed). We used GraphPlan planning algorithm (Blum and Furst, 1997) for this evaluation. This algorithm often solves a sub-problem that can be reformulated to a goal satisfaction problem in mutex network.

In our evaluation we again used maintaining arc-consistency, projection consistency, and projection consistency with tractable case for improving the solving process of the goal satisfaction sub-problem arising within the GraphPlan algorithm. We used a set of planning problems of three domains - dock worker robots domain, towers of Hanoi domain, and refueling planes domain. The same problems were used in (Surynek and Barták, 2007). The tested problems were of various difficulties. The length of solution plans ranged from 2 to 38 actions.

The comparison of runtime of standard GraphPlan and enhanced version of the algorithm is shown in figure 3. All the planning problems used in this evaluation are available at the web site: http://ktiml.mff.cuni.cz/~surynek/research/flairs2008/ (we use our own format of planning

problems since we use non-standard representation with explicit state variables; the reason why we did not used PDDL (McDermott, 1998) is that explicit state variables were not supported by the language at the time of writing this paper).

The improvement obtained by using projection consistency combined with the tractable case is significant with respect to all other versions. Additionally, we found that goal satisfaction problems arising in these planning problems are very similar to random goal satisfaction problems with the low probability of random edges where the improvement obtained by exploiting of the tractable case is promising.

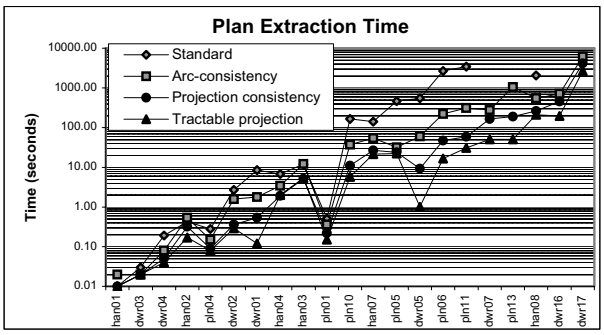

Figure 3: Concurrent plan construction time over several planning problems of various difficulties. Time is shown in logarithmic scale.

Related Works and Conclusion

Originally, projection consistency was proposed in (Surynek, 2007a). Author gives there a detailed theoretical study of the technique. Another closely related work is (Surynek and Barták, 2007). Authors studied in the paper the application of arc-consistency for solving problems of goal satisfaction in mutex network that arise in concurrent planning. In fact we developed a very specialized replacement for the usage of arc-consistency in this context. Projection consistency with tractable class should be used instead of arc-consistency with greater performance benefit.

It seems that concurrent planning is not the only application for projection consistency and the related tractable class. A study of an adaptation of projection consistency for solving difficult Boolean formula satisfaction problems is given in (Surynek, 2007b).

The ideas of using constraint programming techniques in concurrent planning are presented in (Kambhampati et al., 1997; Kambhampati, 2000). However, only local propagation techniques are studied there (contrary to our approach which is more global).

Finally to related works let us note that quite similar idea to our exploiting of structural information encoded in the problem is given in (Ryan, 2006). The author studies the problem of path planning for a group of robots that moves within a graph. The knowledge of structures such as complete sub-graphs is used to improve the solving process.

We proposed a class of the goal satisfaction problem in mutex networks based on projection consistency that can be solved in polynomial time. We built a solving algorithm exploiting the tractable class and we evaluated it in comparison with maintaining arc-consistency and with maintaining plain projection consistency. The experimentation was performed with random problems and with problems arising in concurrent planning. In both experimental setups the tractable class proved to be useful since it brings significant improvement in solving runtime.

References

Blum, A. L., Furst, M. L., 1997. *Fast Planning through planning graph analysis.* Artificial Intelligence 90, 281-300, AAAI Press.

Cook, S. A., 1971. *The Complexity of Theorem Proving Procedures.* Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 151-158, USA, ACM Press.

Dechter, R., 2003. *Constraint Processing.* Morgan Kaufmann Publishers.

Ghallab, M., Nau, D. S., Traverso, P., 2004. *Automated Planning: theory and practice.* Morgan Kaufmann Publishers.

Kambhampati, S., 2000. *Planning Graph as a (Dynamic) CSP: Exploiting EBL, DDB and other CSP Search Techniques in GraphPlan.* Journal of Artificial Intelligence Research 12 (JAIR-12), 1-34, AAAI Press.

Kambhampati, S., Parker, E., Lambrecht, E., 1997. *Understanding and Extending GraphPlan.* In Proceedings of 4th European Conference on Planning, 260-272, LNCS 1348, Springer-Verlag.

Mackworth, A. K., 1977. *Consistency in Networks of Relations.* Artificial Intelligence 8, 99-118, AAAI Press.

McDermott, D., 1998. *PDDL: the Planning Domain Definition Language.* Technical Report. Yale Center for Computational Vision and Control, Yale University, CT, USA.

Ryan, M. R. K. 2006. *Multi-robot path planning with sub-graphs.* Proceedings of the 19th Australasian Conference on Robotics and Automation, Auckland, New Zeland, Australian Robotics & Automation Association.

Surynek, P., 2007a. *A Global Filtration for Satisfying Goals in Mutual Exclusion Networks.* To appear in Recent Advances in Constraints 2007, LNCS, Springer Verlag.

Surynek, P., 2007b. *Solving Difficult SAT Instances Using Greedy Clique Decomposition.* Proceedings of the 7th SARA Symposium, Whistler, Canada. 359-374, LNCS 4612, Springer Verlag.

Surynek, P., Barták, R., 2007. *Maintaining Arc-consistency over Mutex Relations in Planning Graphs during Search.* Proceedings of the 20th FLAIRS conference, Key West, Florida, USA, 134-139, AAAI Press.