Diagnosing and solving over-determined constraint satisfaction problems

R.R. Bakker F. Dikker F. Tempelman P.M. Wogmim
University of Twente, Department of Computer Science,
P.O. Box 217, 7500 AE Enschede, The Netherlands,
Tel. + 31 53 89 3690, Fax. +31 53 33 9605,
E-mail: [bakker, dikker, tempelma, wognum]@cs.utwente.nl

Abstract

Constraint relaxation is a frequently used technique for managing over-determined constraint satisfaction problems. A problem in constraint relaxation is the selection of the appropriate constraints. We show that methods developed in model-based diagnosis solve this problem. The resulting method, DOC, an abbreviation for Diagnosis of Over-determined Constraint Satisfaction Problems, identifies the set of least important constraints that should be relaxed to solve the remaining constraint satisfaction problem. If the solution is not acceptable for a user, DOC selects next-best sets of least-important constraints until an acceptable solution has been generated.

The power of DOC is illustrated by a case study of scheduling the Dutch major league soccer competition. The current schedule is made using human insight and Operations Research methods. Using DOC, the 1992-1993 schedule has been improved by reducing the number and importance of the violated constraints by 56%.

The case study revealed that efficiency improvement is a major issue in order to apply this method to large-scale over-determined scheduling and constraint satisfaction problems.

1 Introduction

Making a schedule for the Dutch major league soccer competition is problematic due to a small number (110 in the 1992/1993 season) of partially contradictory constraints. The constraints origin from hooligan problems which started in the 70s and became gradually more annoying. Currently, police, city mayors, and the railways, forced to transport supporters, impose the major number of constraints on the schedule. The league (K.NVB) and the teams themselves impose additional constraints. In spite of the small number of constraints, no schedule exists that satisfies all constraints. The 1992-1993 schedule violates 7 important and 8 less important constraints.

The soccer scheduling problem is an instance of an Over-Determined Constrained Satisfaction Problem (OCSP). Especially in cases where an easy solution does not exist (e.g. withdrawal of a single constraint), it might be very difficult to identify the constraints that should be relaxed.

A strategy to solve this problem is to transform the constraint satisfaction problem into an optimization problem by splitting the problem into a set of constraints that should always be satisfied and a set of constraints that might be relaxed. For this latter set, the cost of violating each constraint have to be specified. The optimization problem then consists of finding the cheapest solution. In some cases (like linear (in)equalities and unsplit domains of variables). Operation Research methods can be applied to find such a solution. The main problem in this approach is to find a solution in case OR methods cannot be applied. An example is the soccer scheduling problem.

In this paper, we investigate the possibilities of a systematic approach to solve over-determined constraint satisfaction problems. To this end, we interpret an OCSP as a diagnostic problem that can be solved using methods developed in model-based diagnosis. Similar to the previously described approach, we assume that the costs of violating the constraints are known (if this is not the case, we assume that the unknown costs are equal). We show that model-based reasoning generates solutions to an over-determined constraint satisfaction problem in order of increasing cost. As a result, the optimal solution will be generated first.

Overview of this paper In Section 1.1, we discuss some traditional approaches to the problem. As no approach turned out to be useful for solving the case study of the soccer schedule, a more fundamental approach is required. Before we describe our approach, called DOC, in Section 3, the relation between model-based diagnosis and over-determined constrained satisfaction problems is explored in Section 2. In Section 4, we will discuss several extensions of the resulting method that are intended to cope with user wishes like cost modification and addition or removal of constraints. The application of DOC to the soccer schedule case is described in Section 5.

1.1 Related work

Most existing solution methods for Constraint Satisfaction (Meseguer [1989]) halt when no solution can be found. We know of two approaches that deal with over-determinacy in CSPs.

Borning et al. [1987] present a strategy to deal with over-determinacy by dividing all constraints in classes forming a constraint hierarchy. Each constraint in a certain class is considered to be more important than constraints in a lower class. The CSP can first be solved at the highest level in the hierarchy. The solutions found are further refined at the lower level until no solutions can be found anymore. The solutions found at the last solvable level in the hierarchy are the best that can be achieved. This approach can be refined by assigning weights to the constraints in the hierarchy. The sum of the weights of all remaining solutions can be compared to find the optimal one.

The method, however, suffers from serious drawbacks, just as similar approaches developed in the field of OR (Ravindran, Phillips & Solberg [1987]). The problem of over-determinacy can still exist at the top level in the hierarchy. In addition, the number of remaining solutions can be very large and computationally too complex to be explored completely. There is a need for a systematic approach to find the cheapest set of relaxable constraints.

Freuder has developed a method which approaches our goal (Freuder [1989]). He has introduced a partial CSP in which sophisticated possibilities for relaxing constraints in an over-determined CSP can be handled. Based on relaxation criteria an acceptable solution can be formulated and searched for.

The problem of finding the cheapest set of constraints to be relaxed has not been solved in the literature. In this paper an approach is presented that provides a simple relaxation criterion, consisting of removing constraints, and a simple classification mechanism based on weighting constraints. In this way, the optimal solution can be found.

2 OCSPs and Model-based diagnosis

In this section we will discuss the similarity between the problem of identifying faulty components in Model-Based Diagnosis (MBD) and finding a minimal set of constraints to be relaxed in an over-determined CSP.

2.1 Over-determined CSP

Over-determined CSPs (OCSP) are problems for which no solution exists without the relaxation of one or more constraints. In this paper a OCSP is defined as:

- a set of *variables* $X_1, \ldots, X_n$ of which each X_i has a *domain* D_i.
- a set of *constraints* $C_1, \ldots, C_m$. Each $C_i(X_{k_1}, \ldots, X_{k_i}) \subseteq D_{k_1} \times \ldots \times D_{k_i}$
- a set of *weights* $W_1, \ldots, W_m$, indicating the importance of the constraints.

The weight of a set constraints $\{C_1 \ldots, C_k\}$ is defined as a cost function over the individual weights $Cost(W_1, \ldots, W_k)$.

Over-determinacy means that no assignments for the variables within the domains can be found satisfying all constraints. For each assignment we call the set of violated constraints the set of overruled constraints. An optimal solution is found when the costs of relaxation arc minimal over all possible assignments.

The choice of the weights for constraints and the cost function is not arbitrary. Basic notions of preference between constraints and sets of constraints in an application domain should correspond to the representation chosen. In the example in Section 3 integers between 1 and 10 have been chosen as weights for the constraints. A higher number indicates a higher importance. The weights in the case study on the soccer scheduling problem vary between 1 and 10 for minor constraints and are 500 for major constraints. The latter value reflects that violating a major constraint is considered to be worse than violating all minor constraints together. The cost function in this paper simply adds the individual weights.

2.2 Model-based diagnosis

In model-based diagnosis, a model of a technical system is used that describes the correctly operating system. The model consists of components and connections between the components. For each component, the relation between its inputs and outputs is specified. Observations of the actual behavior of the system are used together with the system model to identify possible causes of deviating behavior of the system. Components may have weights which reflect their prior failure rate. Connections transfer signals between components. Observations consist of values of input/output signals of a system.

Diagnostic reasoning consists of constraint propagation, discovering conflicts[1], and generating diagnostic hypotheses. A diagnosis consists of a set of components that might be faulty. Usually, many diagnostic hypotheses exist; additional information is then required to identify the actual defective components. Several diagnostic methods are directed at generating the most-likely diagnoses; diagnoses with a high probability compared to the other ones.

An introduction to model-based diagnosis is given by Davis & Hamscher [1988]. The most frequently cited articles in model-based diagnosis in the context of this paper are collected by Hamscher, Console & de Kleer [1992]. The, for this paper, most relevant diagnostic technique is related to focused Sherlock (de Kleer [1991]), a method that efficiently generates the most likely diagnostic hypotheses.

2.3 Analogy between OCSPs and MBD

The basic concepts in model-based diagnosis and over-determined constraint satisfaction problems are summarized in Table 1. Components and constraints are quite similar, as is the notion of connections, weights, and likely diagnoses and low weighted overruled constraints. An observation in model-based diagnosis is similar to a partial variable assignment in constraint satisfaction problems.

As the concepts in both domains are quite similar, our initial idea was to transform an over-determined constraint satisfaction problem into a model-based diagnostic problem. Then, we could use model-based diagnosis to generate a set of least important constraints that should be relaxed.

Unfortunately, this mapping is impossible due to the problematic correspondence between observations and variable domains. Instead of investigating the consequences of a partial variable assignment (the observations) as is the case in model-based diagnosis, variable assignments in a constraint satisfaction problem are tested until either one satisfies all constraints or no such assignment exists. This difference prohibits a direct mapping.

Abstracting from the context of diagnosing technical systems, model-based diagnosis is a systematic method to identify the most-likely explanation (called diagnosis) for a set of observations. In the search for this diagnosis, conflicts will be found which in turn arc used to guide the search process

[1] A conflict is a set of components that cannot all operate correctly.

Model-based diagnosis		Over-determined constraint satisfaction problems	
Component	IO-relation	Constraint	relation
Weight	failure rate	Weight	importance
Connection	common signal	Connection	common variable
Observation	signal value	Domain	allowed variable values
Diagnosis	set of components that might cause the problems	Over-ruled constraints	set of constraints that can be relaxed to solve the problem

Table I: Basic concepts in model-based diagnosis and over-determined constraint satisfaction problems. Except for observations and domain of variables the concepts in model-based diagnosis and over-determined constraint satisfaction problems are quite similar.

(see Section 3). This guided search process does depend on conflict detection. In model-based diagnosis conflicts can be detected by constraint propagation using the model and the observations. In the context of over-determined constraint satisfaction problems the conflicts will have to be detected in a different way.

Conflict generation in the context of OCSP can be realized by using a slightly modifieated version of forward checking. Forward checking is an efficient method for solving constraint satisfaction problems, see Haralick & Elliot [1980]. The modification consists of keeping track of the constraints used in the propagations. Although we have used forward checking in our experiments, it is not required. Any method that keeps track of the constraints that have been used is fine. In the case study of the Dutch soccer competition, we have experimented with different methods for solving constraint satisfaction problems. The choice only influences the computational efficiency of DOC.

In the next section, we describe the resulting method, called DOC (Diagnosis of Over-determined Constraint satisfaction problems).

3 The DOC method

We first introduce some basic concepts. In the context of an over-determined constraint satisfaction problem, a conflict is a set of constraints that cannot be satisfied simultaneously. A diagnosis D is a set of constraints such that the remaining set of constraints C is satisfiable, with C' = C — D, where C is the original set of constraints. A minimal conflict (diagnosis) is a conflict (diagnosis) such that none of its proper subsets is a conflict (diagnosis). A diagnosis covers all conflicts, i.e., the diagnosis contains at least one constraint of each conflict. A sub-diagnosis is a set of constraints that covers the conflicts detected so far.

The diagnostic reasoning method we will use is a variant of focused Sherlock (de Kleer [1991]). This search method interleaves the generation of conflicts and sub-diagnoses. In a model-based diagnosis context, interleaving is required for efficiency reasons (an inefficient alternative is to generate all conflicts first). By interleaving, the most likely diagnosis is detected first and only a subset of all conflicts is generated. In the context of over-determined constraint satisfaction problem, interleaving is even more required because generating all conflicts will be only be possible in toy problems.

The architecture of DOC is described in Figure 1. The method boils down to (a) identifying the most likely sub-

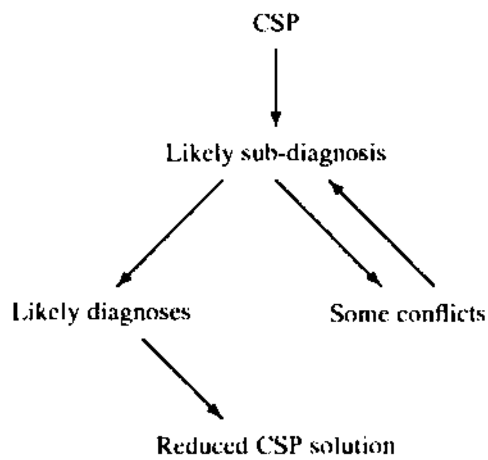

Figure 1: Architecture of DOC.

diagnosis D_{sub} given the conflicts detected so far, and (b) checking the consistency of this sub-diagnosis, i.e., checking whether the remaining constraint satisfaction problem $C' = C - D_{sub}$, with C is the original set of constraints, is solvable. Stage (b) may or may not result in consistency. In case the sub-diagnosis is not consistent, one or more conflicts are detected. We then have to make an additional iteration over stages (a) and (b). In case consistency is confirmed, we have a solution to our problem that can be presented to a user. This solution consists of the set of overruled constraints with least total cost, and, in case the solvability of CSPs is investigated by a constructive method, a solution to the remaining constraint satisfaction problem. In the rest of this section, we will elaborate on both stages.

3.1 Most likely sub-diagnosis

In stage (a) the most likely sub-diagnosis has to be identified given the conflicts discovered so far. This can be achieved by a best-first construction of a hitting-set tree of the conflicts. Generating a hitting-set tree of a set of conflicts is a method to identify all sub-diagnoses. Reiter [1987] describes a method to generate the hitting-set tree breadth-first. Given that the weight of sub-diagnosis $[c_{i1}, \ldots, c_{ik}]$ is $Cost(W_{i1}, \ldots, W_{ik})$, modifying breadth-first into minimal-weight first generation is a trivial modification of that algorithm. Instead of generating all sub-diagnoses, we stop when a least-weight sub-diagnosis is found. The partially generated hitting-set tree is stored for later use.

In Section 3.4 we describe the hitting-set tree for a small example. In case no conflicts have been discovered yet, the most likely sub-diagnosis is [], denoting that the original constraint satisfaction problem should be investigated on solvability first.

3.2 Conflict detection

The following algorithm can be used to check the consistency of a sub-diagnosis. Let CSP denote the constraint satisfaction problem that remains after removing the constraints of the sub-diagnosis. The algorithm either returns a subset of CSP that is not satisfiable, or it detects that CSP is solvable.

Conflict detection algorithm
Input: set of constraints (CSP); Output: a subset of constraints CSP' C CSP in case CSP is not satisfiable or it detects that CSP can be solved.

1. Try to solve CSP using any method for solving constraint satisfaction problems and store the constraints that are used in the set *UC.*
2. In case the CSP is solved, no conflict is identified.
3. In case the CSP is unsolvable, a conflict is the set *UC* of used constraints. Our method does not require that conflicts are minimal. For efficiency reasons, however, minimality may be desirable. A minimal conflict is constructed in the following way:
   repeat
   (a) Select a not previously chosen constraint *c* of *UC.*
   (b) Try to solve the CSP *UC -{c}.*
   In case this CSP is unsolvable, *c* is removed from *UC.*

   until all constraints in *UC* have been investigated

3.3 Efficiency considerations

DOC guarantees the identification of the set of least important constraints to be be removed. However, the order in which the constraints are processed influence the efficiency of the approach. In the example *of* the scheduling problem of the Dutch major league soccer competition, a large gain in efficiency was obtained by controlling the CSP-method such that constraints were processed in decreasing order of importance. In that way, withdrawals of important constraints were identified early on.

If the problem initially represents a strictly hierarchical ordering of constraints, it may be efficient to use this ordering. In this case, DOC can be used as a supplement to Borning's method (Borning et al. 11987]). Doc can then be used to find the solution on the hierarchical level where no solutions can be found.

The worst-case computational complexity of DOC is exponential in the number of variables and the number of constraints. It depends on the computational complexity of the selected method for solving constraint satisfaction problems (all of which have an exponential worst-case computational complexity in the number *of* variables) and the computational complexity of the construction of the hitting-set tree. The worst-case computational complexity of the latter problem is in $0(2^n)$, where n is the total number of constraints. (A worst-case situation is that a single constraint causes all inconsistencies and this constraint is considered to be more important than the combined weight of all other constraints). Although the worst-case computational complexity of DOC is very high, it is not clear whether this causes practical problems.

3.4 A simple example

We will illustrate DOC by a simple example. Consider the constraint satisfaction problem specified in Table 2. The weight are integers between 1 and 10 in which 10 is regarded as most important. The cost function for the combination of weights consists of simply adding individual weights.

Constraint	Weight	Domain
$c_1 : x < y$	$w_1 = 9$	$x \in \{1, 2, \ldots, 10\}$
$c_2 : x^2 + y^2 < 10$	$w_2 = 10$	$y \in \{1, 2, \ldots, 10\}$
$c_3 : x > 2$	$w_3 = 5$	
$c_4 : x > y$	$w_4 = 3$	

Tabic 2: Specification of a simple constraint satisfaction problem. The weights indicate that constraint c_2 is most important and constraint c_4 is least important.

This CSP is not solvable. DOC identifies the set of constraints to be relaxed in the following way. At the start, the most likely sub-diagnosis contains no constraints indicating that the full CSP should be investigated. The constraints are processed in order of their importance, leading to the conflict < c_1, C_2,C_3 > which appears to be a minimal conflict. The sub-diagnosis *[c_3* is best to explore, hence, the CSP' containing c_1, C2, and C4 is checked for solvability. The conflict < *C1*, C_2, c_4 > is discovered; minimization results in the conflict < C_1, C4 >. Sub-diagnosis [c_3,c_4] is best to investigate. The resulting CSP" can be solved, so the best solution to this over-determined constraint satisfaction problem is to remove both C_3 and c_4.

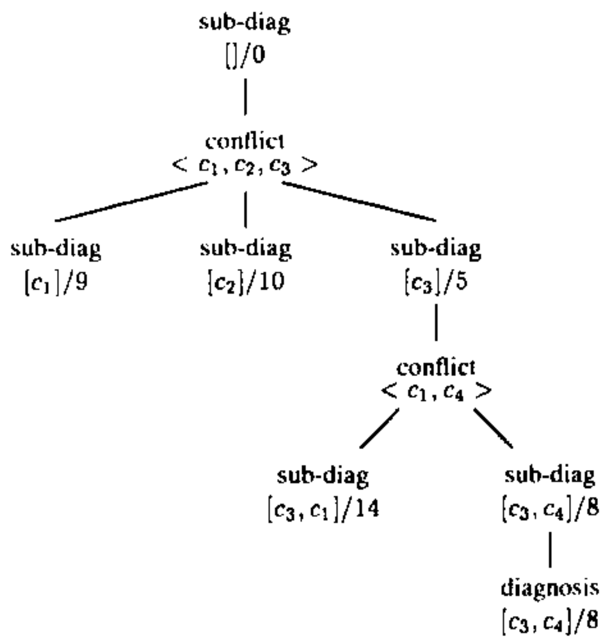

Figure 2: The partial hitting-set/sub-diagnosis tree generated to identify the best solution to the problem described in Table 2.

4 Extensions to DOC

An advantage of the method presented above is the ability to derive alternative solutions from the constructed hitting-set tree, without having to start all over again. This may become handy in case the next-best solution has to be found or in case modifications have to be made to the CSP.

If a user cannot accept the solution presented, the next best alternative can easily be found by just checking the next-best sub-diagnosis in the hitting-set tree and continue DOC. In this case, stage (a) from Section 3 has to be changed to: identifying the next-best likely sub-diagnosis throughout the use of DOC, because the most likely sub-diagnosis will remain the one belonging to the solution already found.

In case of modification of the CSP, three types of possible adjustments to the CSP can be distinguished: *changing the weight* of a constraint, *adding* a constraint and *removing* a constraint. The effects will be described below.

If the violation of a certain constraint in an optimal solution is not acceptable to a user, the weights of constraints need to be changed. In case the weight of one or more constraints is changed, the only thing we have to do is to recalculate the (combined) weights of the sub-diagnoses at the leaves of the hitting set tree. After doing so, the DOC-method can be continued with the withdrawal of the sub-diagnosis that now has become the most likely one.

When a constraint has to be added, it has to be added to the set of constraints in the CSP, and the DOC-method has to be continued with the withdrawal of the most likely sub-diagnosis. This is of course the same set of constraints that was violated in the previous optimal solution.

If a constraint is removed, remove all conflicts containing that constraint from the set of conflicts. This is necessary because if minimization of conflicts is applied, the found conflicts that contain the now removed constraint will no longer represent conflicts. If no minimization is used we might ultimately remove still valid conflicts from the hitting-set tree that have to be re-detected. DOC can now be continued with the withdrawal of the now most likely sub-diagnosis from the pruned hitting-set tree.

Obviously, in all of these cases the initially constructed hitting-set tree can still be used, thus saving effort.

5 Application of DOC

5.1 Dutch major soccer-league

We have applied the DOC method to a scheduling problem that can be classified as middle-sized: the construction of the 1992/1993 time table for the Dutch major league soccer competition: the 'PTT-Tclecompetitie' (Tempelman [1992]). In this problem, 18 playing schemes (so called 'home-away patterns' or HAPs) have to be assigned to the 18 league clubs. Additional constraints have to be considered.

A typical example of a set of home-away patterns for a hctious competition half with 6 participating clubs is shown in table 3. Assigning the patterns to the clubs results in a playing scheme in which the club that has pattern 1 assigned to it, has to play a home match in round one (hence the +) against the club that has been assigned pattern 6, and so on.

	Round				
HAP	1	2	3	4	5
1	+6	-3	+5	-2	+4
2	+5	-6	-4	+1	-3
3	-4	+1	-6	-5	+2
4	+3	-5	+2	+6	-1
5	-2	+4	-1	+3	-6
6	-1	+2	+3	-4	+5

Table 3: Example of a scheme with home-away patterns.

The constraints, a total number of 110 in the 1992/1993 season, are brought forward by municipalities, police, railways, the International Football Federation (FIFA), the clubs and television. They include demands of a security nature, e.g., a certain club cannot play a home match in a certain round because the amount of police force available is inadequate. They also include demands of a commercial nature, e.g., two clubs located in the same area do not want each other's home matches to be played in the same rounds. In the method currently used to solve this problem (Schreuder [1992]), the constraints are divided into requirements and wishes for computational reasons. The wishes are given weights between 1 and 10 by a committee in which municipalities, police and railways are represented. Requirements obtain a weight of 500 which is higher than the sum of weights of all wishes. Due to the large number of constraints in relation to the number of variables (clubs), the problem is over-determined.

An acceptable time table has been obtained for some years using techniques from Operations Research (Schreuder [1992]) and human insight to identify requirements that should be relaxed. All schemes fulfilling the remaining requirements are evaluated regarding the total weight of fulfilled wishes. This way, a strict hierarchy is created, in which one requirement is more important than all wishes together: not always an ideal situation.

The main problem is that the number of requirements, in combination with their tightness, is too big to fulfill them all. The foregoing leads to a time table for the 1992/1993 season in which 7 requirements and 8 wishes are violated. Given the encoding of the weights, the total score of the solution is 3531.

5.2 Results with DOC

In order to apply our method to the problem, first the scheduling problem has to be formulated as a CSP, which is straightforward: the notion of variables in a CSP can be mapped onto the clubs in the competition problem, and the domains of the variables consist of all possible home-away patterns. Constraints reduce the possible (combinations of) assignments of HAPs to clubs. In the example of table 3, applying a constraint like 'club A can not play a home match in round 3' has to result in a reduced solution space in which the possible HAPs for club A are: 2, 3 and 5, because in these patterns the 3rd round denotes an away match.

As in the OR approach to the same problem, weights are combined by adding individual weights. The weights of the wishes are copied from the initial problem, while the weights of requirements are, as above, set at 500, thus being higher than the weights of all wishes combined.

As described above, the DOC-method requires a CSP-method to detect conflicts. Any method that registrates the used constraints will do, but for efficiency reasons it is preferable that conflicts are found early, and that conflicts are small. We have tried a modified version of Forward Checking (Haralick & Elliot [1980]). This method has the advantage that conflicts are found early on. We have combined the method with a so-called clustering algorithm (Schreuder [1992]) in which conflicts are small because partial solutions, only pertaining to some variables and constraints, are constructed initially.

The results of DOC show that DOC's systematic analysis leads to a far better schedule. In DOC's solution, only 3 requirements and 9 wishes are violated. The total score is 1538: an improvement of 56%. An implementation in Quintus Prolog[2], solved the problem in about 25 hours with minimization of conflicts, and in roughly twice as much time without minimization. In both cases Forward Checking was used. Using the clustering algorithm, the time used was about half as much, but the required computer memory was often insufficient, due to the amount of partial results. Also, the order of the constraints appeared to influence efficiency.

The computational problems, in time as well as in available memory, are not yet properly addressed. To solve larger problems (like school scheduling problems), efficiency improvements of DOC might be necessary. A preliminary analysis indicates that the size of the conflicts is the main cause of computational problems.

6 Conclusions

In this paper a method called DOC has been described that solves over-determined constraint satisfaction problems by generating sets of constraints that should be relaxed in order of increasing costs. The application of DOC in a case study on scheduling the Dutch major league soccer competition resulted in a schedule that improves the 1992-1993 schedule by at least 56%. The Dutch major league soccer competition can be classified as a mid-sized problem. The case study indicated that DOC's computational efficiency should be improved to apply DOC to large-scale over-determined constraint satisfaction problems.

Acknowledgements

We would like to thank Dick van Soest, who provided helpful comments on an earlier version of this paper.

References

A. Borning, R. Duisberg, B. Freeman-Benson, A. Kramer & M. Woolf [Oct., 1987], "Constraint hierarchies," in OOPSLA87, Norman Meyrowitz, ed., Orlando, Florida, 48-60.

R. Davis & W. Hamscher[1988], "Model-based Reasoning: Troubleshooting," in Exploring Artificial Intelligence, H.E. Shrobe, ed., Morgan Kaufmann Publishers, San Mateo, California, 297-346.

E.C. Freuder [1989], "Partial Constraint Satisfaction;' in Proceedings Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, 20-25 August 1989, N.S. Sridharan, ed., Morgan Kaufmann Publishers, San Mateo, CA, 278-283.

[2]By no means the programs were optimized.

W. Hamscher, L. Console & J. de Kleer[1992], Readings in model-based diagnosis, Morgan Kaufman Publishers Inc., San Mateo.

R.M. Haralick & G.L. Elliot [1980], "Increasing tree search efficiency for constraint satisfaction problems," Artificial Intelligence.

J. de Kleer [1991], "Focusing on Probable Diagnoses," in Proceedings Ninth National Conference on Artificial Intelligence, Anaheim, CA, 14-19 July 1991, AAA1 Press/MIT Press, Menlo Park, CA

P. Meseguer [1989], "Constraint Satisfaction Problems: An Overview," AICOM 2, 3-17.

A. Ravindran, D.T. Phillips & J.J. Solberg[1987], Operations research : principles and practice, Wiley, New York.

R. Reiter [1987], "A Theory of Diagnosis from First Principles," Artificial Intelligence 32, 57-95.

J.A.M. Schreuder [1992], "Combinatorial aspects of construction of competition Dutch Professional Football Leagues," Discrete Applied Mathematics 35, 301-312.

F. Tempelman [1992], "Een rooster voor de PTT-Telecompetitie: Systematische aanpak van overbepaaldc problemen," University of Twente, Master Thesis, UT-KBS-92-34, Enschede, (in dutch).