Using Model-based Reasoning for Generating Explanations from Environmental Models

Franz Wotawa

Graz University of Technology, Institute for Software Technology,
8010 Graz, Inffeldgasse 16b/2, Austria
wotawa@ist.tugraz.at,
WWW home page: http://www.ist.tugraz.at/wotawa/

Abstract. The paper summarizes research for generating explanations from environmental models. We make use of the corresponding representation of the environmental models as constraint satisfaction problems. We show how this representation can be used directly to derive explanations. During the whole paper we use a coarse model which represents the relationship between the ground-level ozone concentration and its influencing quantities like traffic, industrial emissions, and meteorological conditions.

1 Introduction

The ability of explaining complex physical, chemical, or biological circumstances to others which are less educated in the respective domains is an issue of growing importance. Effects that are caused by decisions has to be known in advance in order to prevent undesired consequences. For example, constructing a hydroelectric power plant has an impact to the river's ecosystem because of changes in water flow and maybe the course of the river. In order to make a good decision which reduces negative effects while retaining the positive desired once, knowledge about consequences which originates from underlying physical, chemical, or biological knowledge is required which is hardly available. Moreover, usually decision makers do not have deep knowledge in all required areas. Instead they collect informations from experts in various fields, merge them, and come up with a decision. There are several risks attached to this process. First, there is always the problem of missing information. You can never be sure really to capture all different aspects of a certain situation. Without special knowledge about relationships between causes and potential effects there is a high possibility for failing to capture important facets of a problem. Second, merging informations requires usually a deep understanding of different domains. Moreover, there is no guarantee that the same words used by experts in different domains really have the same meaning. Hence, the merged knowledge may not capture reality.

In summary, fetching, evaluating, and merging knowledge requires a good understanding of basic principles and possible interactions behind processes, for making a good decision. One solution of this problem is to provide decision makers with complex models of the problem domain where they can interact with in order to get an understanding of causes and their corresponding effects. These models need not to capture all aspects of reality neither do they require to capture reality precisely. Instead the models should allow to extract the cause-effect relationship on an abstract level which is usually sufficient for decision making. The possibility for interacting with the model is important in order to demonstrate which actions have which effects. However, it is equal important to show why an effect occur and to identify the root causes. For example, the amount of ground-level ozone which is a poisonous gas in a region depends not only of the precursor substances which originates from emissions of industry, traffic, and domestic fuel, but also from meteorological circumstances like the degree of sunshine and wind conditions. Hence, an explanation for a high ozone concentration in the troposphere should include all causes although some of them cannot be influenced by humans. Moreover, the explanation should take care of a certain situation. If there is not traffic jam, the number of cars in an area maybe is not high and as a consequence has almost no influence on the ozone concentration. Once, the knowledge about no traffic jams is available, the explanations of the high ozone concentration should not include traffic as a cause anymore.

The purpose of this paper is to present both means for representing models in a comprehensible way and a technique for computing explanations directly from the available models. The technique for computing explanations from models automatically takes care of additional knowledge. Hence, during interacting with the model explanations change once new facts regarding the considered state are available. The paper is organized as follows. In the first part we introduce the concept of representing models as constraint satisfaction problems (CSPs). We use a course cause-effect model for the ground-level ozone concentration for illustrating the modeling. In the second part we give a short introduction into model-based diagnosis and the algorithm that are required in order to compute diagnoses which are equivalent to explanations. Related literature and open problems are discussed at the end of this article.

2 Modeling Using CSPs

In this section we introduce the basic concepts and definitions of CSPs. An introduction including a description of algorithms and improvements can be found in [1, 2] and more recently [3] which provides a good starting point for studying CSPs.

A CSP is characterized by a set of variables $V = \{V_1, \ldots, V_n\}$, each associated with its (not necessarily finite) domain D_i, $1 \leq i \leq n$, and a set of constraints $C = \{C_1, \ldots, C_k\}$. Each of the constraints C_j has an associated corresponding pair (X_j, R_j), where $X_j \subseteq V$ is a set of variables, and R_j is a relation over X_j. X_j is called the scope of constraint C_j. For convenience we assume a function $dom : V \mapsto DOM$ that maps a variable $V = i$ to its domain D_i, a function $scope : C \mapsto 2^V$ that maps a constraint to its corresponding scope, and a function $rel : C \mapsto RELATIONS$ that maps constraints to their relations.

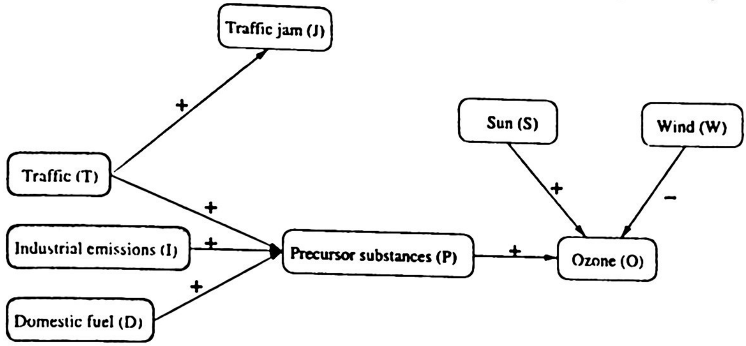

Fig. 1. The causal relationships within the ozone concentration domain

Our ground-level ozone-concentration example can be represented by a CSP. For this purpose we start using the causal relationships between the involved quantities which is depicted in Figure 1. The causal relationships are given as directed labeled arcs between the vertices. The direction of the arcs is always from the cause to the effect. A label '+' represents a positive influence, i.e., an increase of amount at the cause leads to an increase of amount at the effect, whereas a '-' is for stating a negative influence. For example, a sunny day usually leads to an increase of ozone concentration, but a day with heavy storms causes a decrease of concentration at a specific location. Of course the represented model does not capture all effects. The purpose of the model is to serve as an example to explain definitions and algorithms throughout the paper.

In order to represent the cause-effect model from Figure 1 as a CSP we have to map the arcs and vertices to a variables and relations. The extraction of variables is simple. All vertices of the graph represent variables of the CSP. Hence, the set of variables of our example comprise the emissions from traffic (T), from industry (I), and from domestic fuel (D). Moreover, we have variables for precursor substances (P), ozone (O), traffic jam (J), the amount of sunshine (S) and wind (W). The constraints between the variables are given by the arcs. The corresponding variables to the source and target vertex have to be in the same constraint. If there are several arcs which lead to the same target vertex, then all variables corresponding to the source vertices have to be in the same constraint. For our example we can identify three different constraints C_1, C_2, and C_3 with the following associated scopes X_i: $X_1 = \{T, I, D, P\}$, $X_2 = \{T, J\}$, $X_3 = \{P, S, W, O\}$. The only parts that are missing for completing the CSP representations are the relations which requires variable domains. For simplicity and because of the fact, that we are interested in stating knowledge about deviations, like saying the traffic is increasing or the amount of sunshine is stable, we associated the domain $D = \{+, 0, -\}$ to all variables, where '+' represents increasing, '-' represents decreasing, and '0' represents stable. Using this domain we can state the following relations:

$R_1(T$	I	D	$P)$
0	0	0	0
-	0	0	-
0	-	0	-
0	0	-	-
-	-	0	-
-	0	-	-
0	-	-	-
-	-	-	-
+	0	0	+
0	+	0	+
0	0	+	+
+	+	0	+
+	0	+	+
0	+	+	+
+	+	+	+
+	-	x	0
-	+	x	0
+	x	-	0
-	x	+	0
x	+	-	0
x	-	+	0
-	x	x	-
x	-	x	-
x	x	-	-
+	x	x	+
x	+	x	+
x	x	+	+

$R_2(T$	$J)$
0	0
-	-
+	+

$R_3(P$	S	W	$O)$
0	0	0	0
-	0	0	-
0	-	0	-
0	0	+	-
-	-	0	-
-	0	+	-
0	-	+	-
-	-	+	-
+	0	0	+
0	+	0	+
0	0	-	+
+	+	0	+
+	0	-	+
0	+	-	+
+	+	-	+
+	-	x	0
-	+	x	0
+	x	-	0
-	x	+	0
x	+	-	0
x	-	+	0
-	x	x	-
x	-	x	-
x	x	+	-
+	x	x	+
x	+	x	+
x	x	-	+

Note that the last lines of relation R_1 and R_2 represents other allowed combinations. For example, if one cause of precursor substances is increasing and the other is decreasing nothing can be said about the precursor substances because lack of quantitative informations. Hence, the x stands for all values of the domain D which are not explicitly given in the tables.

The CSP representation of the ground-level ozone-concentration example can now be used to answer questions. A question itself can be stated as a constraint. By adding the constraint to the CSP model, we get a new CSP from which we derive the answer. In order to distinguish the original CSP representation from the representation with additional constraints, we refer to the former by CSP_O and to the latter by CSP_O^x where x is the set of additional constraints. For example, when we want to know what causes the increase of ozone during stable weather and traffic jam conditions, we introduce new constraints C_4, C_5, C_6, and C_7 with scopes $X_4 = \{S\}$, $X_5 = \{W\}$, $X_6 = \{J\}$, $X_7 = \{O\}$ and relations $R_4 = \{(0)\}$, $R_5 = \{(0)\}$, $R_6 = \{(0)\}$, $R_7 = \{(+)\}$. Hence, we finally obtain a new CSP namely $CSP_O^{\{C_4,C_5,C_6,C_7\}}$. The answers of the question which is now represented as set of constraints are the solutions of $CSP_O^{\{C_4,C_5,C_6,C_7\}}$. This leads us directly to the question what solutions for a given CSP are? Because constraints formulate valid relationships between variables, a solution are assignments of values to variables such that all constraints are fulfilled.

When assigning a unique value to each variable from a subset of V, we get an instantiation. We further say that an instantiation satisfy a given constraint C_j if the partial assignments which correspond to the scope of C_j are element of the relation R_j of the constraint. Otherwise, we say that the constraint C_j is violated. For example, the instantiation $O =' -'$ satisfies the constraint R_3 but violates the constraint R_7 in $CSP_O^{\{C_4,C_5,C_6,C_7\}}$ because $O =' -'$ is not an element of relation R_7 (which holds only one valid tuple $O =' +'$). The notation of satisfaction and violation of constraints naturally leads to the definition of a solution for a given CSP. A solution is an instantiation of all variables V such that all constraints are satisfied. Such an instantiation is also called a legal or locally consistent instantiation. Note that there is usually not only one solution to a CSP. For $CSP_O^{\{C_4,C_5,C_6,C_7\}}$ we obtain several solutions which satisfies all constraints. For example, the assignment $T =' 0'$, $I =' +'$, $D =' 0'$, $J =' 0'$, $S =' 0'$, $W =' 0'$, $P =' +'$, $O =' +'$ satisfies all constraints. A verbal interpretation of this solution is that an increase of industrial emissions alone without an increase of emissions from households causes an increase of the ozone concentration. Another solution would include $I =' +'$ and $D =' -'$ which says that savings in emissions of households may not enough to decrease the amount of ground-level ozone when industrial exhaust fumes increase too much.

3 Computing Solutions

A simple way of computing one or all solutions of a given CSP is to test all possible variable assignment. Assignments that satisfy all constraints are added to the set of solutions. This simple approach, however, is intractable in general because it requires testing all possible assignments, which is exponential in the number of variables if we assume finite domains. Hence, a more efficient approach is necessary. One approach which is used in practice is to compute solutions by applying search algorithms. In particular depth-first search with backtracking is used. This approach can be further improved by applying heuristics which are based on criterias like number of tuples in a relation or scope size of constraints. Alternatively, there are algorithms available which can be applied for special CSP like tree-structured CSPs. In this paper we follow this approach and show how solutions can be extracted from tree-structured CSPs. Furthermore, we discuss capabilities of this approach to be of use for general CSPs.

Before introducing the solution extraction algorithm for tree-structured CSPs we have to define the term acyclicity of CSPs which is equivalent to tree-structured CSPs. For this purpose we first show how general CSPs can be compiled into its equivalent hypergraph representation and than define acyclicity of hypergraphs. A hypergraph HG for a given CSP (V, C) is defined as follows. Every variable is mapped to a vertex and the scope of every constraint $C_i \in C$ is mapped to an arc in HG. Hence, an arc not only connects two vertices as this is usually the case for graphs but connects two or more vertices. The corresponding hypergraph representation of our ozone-concentration model $CSP_O^{\{C_4,C_5,C_6,C_7\}}$ is depicted in Figure 2.

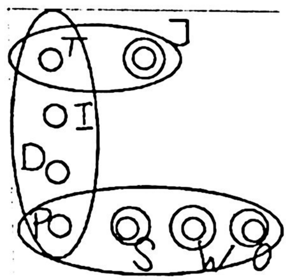

Fig. 2. The hypergraph representation of $CSP_O^{\{C_4,C_5,C_6,C_7\}}$

A graphs is said to be cyclic there exists a path through the graph which starts and ends in the same vertex. Hence, a graph is acyclic if such a path does not exists. This definition of cyclicity and acyclicity can be extended for hypergraphs. Checking acyclicity of hypergraph can be easily implemented. A simple algorithm that proves the acyclicity of hypergraphs is the following:

acyclicHG(V,A)

1. Repeatedly apply the following operations until they can not be applied:
   (a) Delete a vertex that occurs only in one arc.
   (b) Delete an arc that is contained in another arc.
2. If no vertex remains, then return **True**. Otherwise, return **False**.

In this article V denotes the set of vertices and A the set of arcs. The **acyclicHG** algorithm returns **True** if the hypergraph is acyclic and **False**, otherwise, and run in time quadratic in the size of the hypergraph. In [4] Tarjan and Yannakakis presented a linear time algorithm for testing acyclicity of hypergraphs. We now say that a CSP is cyclic (acyclic) if its corresponding hypergraph is cyclic (acyclic). Because the hypergraph from Figure 2 is acyclic the corresponding CSP $CSP_O^{\{C_4,C_5,C_6,C_7\}}$ is acyclic. .

Every acyclic hypergraph and hence every acyclic CSP can be represented as hypertree. One vertex of the hypergraph is selected as the root vertex. The other vertices are connected as given by the original hypergraph. The hypertree representation of our ozone concentration example is depicted in Figure 3.

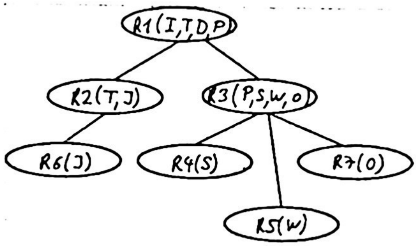

Fig. 3. The hypertree representation of $CSP_O^{\{C_4,C_5,C_6,C_7\}}$

A solution of an acyclic CSP can be easily computed. The algorithm has two phases. In the first phase we are working from the leaf vertices to the root. In the forward phase we compute new relations for a vertex by applying a semi-join operation of the relation and the relations of its child vertices. This operation is done until we reach the root vertex. The second phase or backward phase is for extracting all or one solution from the remaining relations. If we are searching only for one solution, we choose one tuple of the relation and remove all tuples of the children relations that are not compatible with the chosen tuple. By selecting the next tuples from the children we go down to the leafs. The forward phase of the algorithm works in polynomial time with respect to the size of the CSP. This holds also when there are an exponential number of possible solutions. Hence, solution extraction is might be more time consuming. If we are searching only for one solution it can also be done in polynomial time. The reason for this fast behavior of the algorithm comes from the structural properties which causes every computation to be local. Hence, we do not have to consider computations of different branches within the tree at the same time.

It is worth noting that not all CSPs are acyclic and therefore cannot be represented as a hypertree. Hence, the described CSP algorithm cannot be applied in all cases but there is solution to the problem. All CSPs can be converted into an equivalent acyclic CSP by combining different vertices, i.e.,, constraints and their relations. This conversion process can be done automatically and makes only use of the structural properties of the CSP and is usually referred as decomposition methods. There are several different composition methods described in literature, including tree-decomposition [5], hinge decomposition [4], and more recently hypertree decomposition [6–8], which subsumes the others. The number of joined constraints which is usually expressed as tree-width is an indicator of acyclicity of the original CSP. Note that every CSP can be compiled into an acyclic variant by joining all constraints together. In this case we finally get a hypertree which comprises only one vertex. Because of the fact that joining together all constraints of a CSP is intractable in general, such a compilation does not make any sense in practice. However, in many practical cases the number of constraints that have to be joined is relatively small and decomposition methods lead to CSPs which can be solved in a faster and more efficient way.

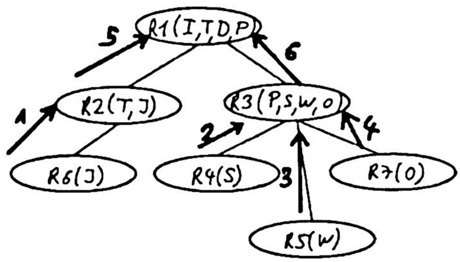

Fig. 4. Computing a solution for $CSP_O^{\{C_4,C_5,C_6,C_7\}}$

In the following we illustrate the CSP algorithm. For this purpose we use our ozone-concentration example. The computation steps of the forward phase is depicted in Figure 4. In the first step we apply a semi-join operation on the relations of constraints C_2 and C_6 which lead to a reduced relation for C_2 because only a '0' value for variable J is allowed anymore:

$R_2($	T	J	$)$
	0	0	

In steps 2, 3, and 4 the relation of constraint C_3 is reduced which is based on values of C_4, C_5, and C_7.

$R_3($	P	S	W	O	$)$
	+	0	0	+	

In the last steps we have to apply the new relations of R_2 and R_3 on relation R_1 and we finally obtain:

$R_1($	T	I	D	P	$)$
	0	+	0	+	
	0	0	+	+	
	0	+	+	+	
	0	+	-	+	
	0	-	+	+	

This 5 relations represent all possible solutions for our example. We can extract several interesting facts from the solutions. For example, it might be not enough to reduce the industrial emissions alone without considering emissions from households.

In summary computing a solution for a CSP as explained in this article include the following steps:

1. First check acyclicity of the CSP. If it is acyclic, goto step 3.
2. Convert the CSP to an equivalent acyclic CSP. For this step we require a decomposition method like hypertree decomposition.
3. Compute a solution by first going from the leafs to the root and reducing the number of tuples of the corresponding relations. This step requires the use of the semi-join operator for relations. Secondly, go from the root to the leafs and select an appropriate tuple. The step may reduce tuples in children vertices.

The proposed method comprises two steps. One offline step which converts a cyclic CSP into an acyclic one, and an online step which computes the solutions. The latter has a very good time complexity with respect to the size of the CSP.

4 Model-based Diagnosis

Although, CSP allow for representing models in a convenient way there is a shortcoming in distinguishing between different kind of variables. For example, in our ozone-concentration domain we have variables which correspond to effects and variables which correspond to causes. Effects and causes can be observed. However, once an effect is undesired we are interested in finding the causes of the effect. Note that some of the causes are associated to actions. For example, emissions can be influenced by taking actions, e.g., introducing governmental restrictions on driving or on used cars or industrial plants. Other causes are not associated with actions. For example, sunshine and wind cannot be influenced in general.

To overcome this problem we introduce the concept of model-based reasoning and in particular model-based diagnosis. In model-based diagnosis we assume to have a model of a system like our ozone-concentration CSP model, some observations, and a set of assumptions. The assumptions correspond to part of the model and therefore cause some behavior. In case of differences between the behavior and the observations, assumptions can be withdrawn until the derived behavior does not contradict the observed behavior. Model-based diagnosis has been used in different domains including ecological systems [9–11]. In this article we refer to the standard definition of model-based diagnosis [12] and adapt it.

A diagnosis problem according to Reiter [12] is a tuple (SD, OBS, AS) where SD is the model (system description), OBS is a set of observations, and AS is a set of assumptions. The assumptions are linked to the system description. They are used to select the corresponding behavior which is used to derive the overall behavior of the system. In the original article SD and OBS are sets of sentences written in first order logic (FOL). In our case SD is the original CSP with added information about the corresponding assumptions, and OBS is a set of additional constraints like in our $CSP_O^{\{C_4,C_5,C_6,C_7\}}$ model.

In order to introduce the assumption in the CSP representation we follow the following idea. Every tuple of a relation expresses a behavior. For example the first line of relation R_1 of CSP_O says that if all emissions do not change, then the amount of ozone precursor substances do not change. Hence, we can add the assumptions that all emission variables do not change in a separate column of the tuple. Hence, we can add the set $\{okT, okI, okD\}$ to this additional column where okX stands for X$\in$ $\{T, I, D\}$ does not change, e.g., okT means that the amount of traffic or to be more exact the amount of emissions coming from traffic is stable. With similar arguments we can fill the additional column for all tuples of the relations.

This filling, however, leads to a huge number of added facts about the state of causes. To overcome this problem we simplify the representation and add only negative facts, e.g., $\neg okT$ to the additional column. Hence, finally we obtain modified CSP for the ozone-concentration domain.

$R_1($ T	I	D	P $)$	Expl.
0	0	0	0	$\{\{\}\}$
-	0	0	-	$\{\{\neg okT\}\}$
0	-	0	-	$\{\{\neg okI\}\}$
0	0	-	-	$\{\{\neg okD\}\}$
-	-	0	-	$\{\{\neg okT, \neg okI\}\}$
-	0	-	-	$\{\{\neg okT, \neg okD\}\}$
0	-	-	-	$\{\{\neg okI, \neg okD\}\}$
-	-	-	-	$\{\{\neg okT, \neg okI, \neg okD\}\}$
+	0	0	+	$\{\{\neg okT\}\}$
0	+	0	+	$\{\{\neg okI\}\}$
0	0	+	+	$\{\{\neg okD\}\}$
+	+	0	+	$\{\{\neg okT, \neg okI\}\}$
+	0	+	+	$\{\{\neg okT, \neg okD\}\}$
0	+	+	+	$\{\{\neg okI, \neg okD\}\}$
+	+	+	+	$\{\{\neg okT, \neg okI, \neg okD\}\}$
⋮	⋮	⋮	⋮	⋮

$R_3($ P	S	W	O $)$	Expl.
0	0	0	0	$\{\{\}\}$
-	0	0	-	$\{\{\}\}$
0	-	0	-	$\{\{\neg okS\}\}$
0	0	+	-	$\{\{\neg okW\}\}$
-	-	0	-	$\{\{\neg okS\}\}$
-	0	+	-	$\{\{\neg okW\}\}$
0	-	+	-	$\{\{\neg okS, \neg okW\}\}$
-	-	+	-	$\{\{\neg okS, \neg okW\}\}$
+	0	0	+	$\{\{\}\}$
0	+	0	+	$\{\{\neg okS\}\}$
0	0	-	+	$\{\{\neg okW\}\}$
+	+	0	+	$\{\{\neg okS\}\}$
+	0	-	+	$\{\{\neg okW\}\}$
0	+	-	+	$\{\{\neg okW\}\}$
+	+	-	+	$\{\{\neg okS, \neg okW\}\}$
⋮	⋮	⋮	⋮	⋮

$R_2($ T	J $)$	Expl.
0	0	$\{\{\}\}$
-	-	$\{\{\neg okT\}\}$
+	+	$\{\{\neg okT\}\}$

Given the diagnosis problem stated as CSP with an additional column for explanations (Expl.), we can compute the diagnoses. A diagnosis is explanation of a tuple that remains when computing a solution. The algorithm for computing a solution for the modified remains the same. The only thing that changes is the semi-join operation which has to consider the explanation column. According to the TREE* algorithm [17] are joined together by taking all elements of the sets at the corresponding tuple and building the union. Because of the fact, that we are usually interested in explanations that comprises a given number of assumptions, we reduce the union sets to elements with a cardinality smaller or equivalent to the given number. For example if a tuple has the explanation set $\{\{a, b\}, \{c\}$ and another tuple (which can be joined) has a set $\{\{a, c\}\}$, we first build the union of all elements. We obtain a set $\{\{a, b, c\}, \{a, c\}\}$. In the second step we remove all elements with a cardinality greater than a given value, say for example 2. Hence, we finally obtain the explanation $\{\{a, c\}\}$ for the joined tuple. We could remove the first element because of its cardinality which is greater than 2.

For our modified ozone-concentration example we change the relations of the additional constraints to $R_i = \{(0, \{\{\}\})\}$ for $i \in \{4, 5, 6\}$ and $R_7 = \{(+, \{\{\}\})\}$. We now use the same algorithm for computing a solution but consider the additional explanation column of the relations. In the first step the relations of constraint C_2 and C_6 are joined which leads to the new relation:

R_2(T	J)	Expl.
0	0	{{}}

Further applying the join operation to constraints C_3, C_4, C_5, and C_7 leads to the new relation for constraint C_3:

R_3(P	S	W	O)	Expl.
+	0	0	+	{{}}

Finally, we join the relations of constraints C_1, C_2 and C_3 and get the following result:

R_1(T	I	D	P)	Expl.
0	+	0	+	$\{\{\neg okI\}\}$
0	0	+	+	$\{\{\neg okD\}\}$
0	+	+	+	$\{\{\neg okI, \neg okD\}\}$
0	+	-	+	$\{\{\neg okI, \neg okD\}\}$
0	-	+	+	$\{\{\neg okI, \neg okD\}\}$

This solution represents three different explanations for the fact that the ozone concentration is rising. One says that the industrial emissions are the root cause ($\neg okI$), one says that the household emissions ($\neg okD$) is the reason for the given observations, and the last explains the ozone concentration by assuming both emission categories to be responsible.

We can state two important differences when we compare the explanation-based solution with the solution obtained from the CSP alone without considering assumptions.

1. In the explanation-based solution, the causes of a given behavior are now explicitly stated. For example, in the first line of the resulting relation of constraint C_1 we now that only the industrial emissions are responsible.
2. The explanations can be ordered with respect to their cardinality. If we are not interested in explanations of a size 2 or more, only two explanations remain for our ozone-concentration example.

The original algorithm for computing diagnosis from tree-structured CSP can be found in [17]. There proofs of the correctness and the relationship between the CSP approach and the original model-based diagnosis definition can be found. We omitted these results and definitions because of the scope of this article which is to show how CSP techniques can be used to model environmental systems and how useful explanations can be extracted.

5 Related research

There is a lot of research at the boundaries between Artificial Intelligence and environmental modeling. When considering only work which is close to qualitative or model-based reasoning we have to mention Heller and Struss's work on modeling water ecosystems using a component-oriented modeling approach [9–11]. In their work modeling is done for the purpose of explaining undesired behavior which leads to counter measurements in order to bring the ecosystem in a desired state. Heller and Struss use the traditional model-based diagnosis approach for computing the explanations. In contrast to their work, we focus on environmental modeling using CSPs and explain how available techniques can be used to compute explanations directly from the CSPs.

Other work in environmental modeling which makes use of qualitative reasoning has been done by Bredeweg and colleagues. They use qualitative reasoning directly for expressing relationships between quantities of different ecosystems. In [13] the authors present a model which represents the interactions between populations. In [14] the qualitative modeling of stream ecosystems is explained. All the papers focus mainly on modeling and not on explanations. Moreover, they make use of causal models and not of CSPs.

One of the first algorithm which allows for computing diagnoses directly from CSPs is El Fattah and Dechter's SAB algorithm [15]. Based on the ideas behind SAB Stumptner and Wotawa [16, 17] invented the TREE* algorithm which is faster when computing minimal cardinality diagnoses. More recently Sachenbacher and Williams [18] presented a framework which explains the differences and equivalences between SAB and TREE*. Other algorithms for diagnosis like Reiter's hitting set algorithm [12, 19] or Fröhlich and Nejdl's algorithm [20] are based on logical descriptions of models and not on CSP formalizations.

Gottlob and colleagues [8] give a very good comparison of available decomposition methods including references and relationships. For a description of an algorithm which implements the hypertree decomposition we refer to [6]. The algorithm allows for an easy implementation which can be used for small and medium size CSPs. Because of memory consumption some improvements have to be done in order to extend applicability. One direction of improvements which makes use of structural pre-decomposition with limitations is explained in [21]. There the authors introduced the coupling of different decomposition methods in order to make hypertree decomposition applicable for larger CSPs.

6 Conclusion

In this paper we described modeling as formulating a corresponding constraint satisfaction problem. We showed how the resulting CSP model can be used to extract solutions. For this purpose we introduced a process comprising two parts. In the first part the CSP is compiled into an equivalent acyclic CSP. The second part is for computing solutions. We extended the CSP model by introducing assumptions which allow for distinguishing causes and effects. The ideas behind originate from model-based diagnosis. Furthermore, we showed how assumptions can be used to gain knowledge about explanations for a given situation.

References

1. Dechter, R.: From local to global consistency. Artificial Intelligence 55 (1992) 87–107.
2. Mackworth, A.: Constraint satisfaction. In Shapiro, S.C., ed.: Encyclopedia of Artificial Intelligence, John Wiley & Sons (1987) 205–211.
3. Dechter, R.: Constraint Processing. Morgan Kaufmann (2003).
4. Tarjan, R.E., Yannakakis, M.: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM Journal of Computing 13 (1984) 566–579.
5. Dechter, R., Pearl, J.: Tree clustering for constraint networks. Artificial Intelligence 38 (1989) 353–366.
6. Gottlob, G., Leone, N., Scarcello, F.: On Tractable Queries and Constraints. In: Proc. 12th International Conference on Database and Expert Systems Applications DEXA 2001, Florence, Italy (1999).
7. Gottlob, G., Leone, N., Scarcello, F.: Hypertree Decomposition and Tractable Queries. In: Proc. 18th ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS-99), Philadelphia, PA (1999) 21–32.
8. Gottlob, G., Leone, N., Scarcello, F.: A comparison of structural CSP decomposition methods. Artificial Intelligence 124 (2000) 243–282.
9. Heller, U., Struss, P.: Transformation of Qualitative Dynamic Models – Application in Hydro-Ecology. In: Proceedings of the 10^{th} International Workshop on Qualitative Reasoning, AAAI Press (1996) 83–92.
10. Heller, U., Struss, P.: Conceptual Modeling in the Environmental Domain. In: Proceedings of the 15^{th} IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics. Volume 6., Berlin, Germany (1997) 147–152.
11. Heller, U., Struss, P.: Diagnosis and therapy recognition for ecosystems - usage of model-based diagnosis techniques. In: 12th International Symposium Computer Science for Environmental Protection (UI-98), Bremen (1998).
12. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32 (1987) 57–95.
13. Salles, P., Bredeweg, B., Bensusan, N.: The ants' garden: Qualitative models of complex interactions between populations. In: Proceedings of the 17^{th} International Workshop on Qualitative Reasoning (QR), Brasilia, Brazil (2003).
14. Salles, P., Bredeweg, B., Araujo, S.: Qualitative models about stream ecosystem recovery: Exploratory studies. In: Proceedings of the 17^{th} International Workshop on Qualitative Reasoning (QR), Brasilia, Brazil (2003).
15. Fattah, Y.E., Dechter, R.: Diagnosing tree-decomposable circuits. In: Proceedings 14^{th} International Joint Conf. on Artificial Intelligence. (1995) 1742 – 1748.
16. Stumptner, M., Wotawa, F.: Diagnosing Tree-Structured Systems. In: Proceedings 15^{th} International Joint Conf. on Artificial Intelligence, Nagoya, Japan (1997).
17. Stumptner, M., Wotawa, F.: Diagnosing tree-structured systems. Artificial Intelligence 127 (2001) 1–29.
18. Sachenbacher, M., Williams, B.C.: Diagnosis as semiring-based constraint optimization. In: Proceedings of the 16^{th} European Conference on Artificial Intelligence (ECAI), Valencia, Spain (2004) 873–877.

Greiner, R., Smith, B.A., Wilkerson, R.W.: A correction to the algorithm in Reiter's theory of diagnosis. Artificial Intelligence 41 (1989) 79–88.
Fröhlich, P., Nejdl, W.: A Static Model-Based Engine for Model-Based Reasoning. In: Proceedings 15th International Joint Conf. on Artificial Intelligence, Nagoya, Japan (1997).
Gottlob, G., Hutle, M., Wotawa, F.: Combining hypertree, bicomp, and hinge decomposition. In: Proceedings of the European Conference on Artificial Intelligence (ECAI), Lyon, France (2002).