On the Usefulness of Re-using Diagnostic Solutions

Luigi Portinale and **Pietro Torasso**[1]

Abstract. Recent studies on planning, comparing plan re-use and plan generation, have shown that both the above tasks may have the same degree of computational complexity, even if we deal with very similar problems. The aim of this paper is to show that the same kind of results apply also for diagnosis. We propose a theoretical complexity analysis coupled with some experimental tests, intended to evaluate the adequacy of adaptation strategies which re-use the solutions of past diagnostic problems in order to build a solution to the problem to be solved. Results of such analysis show that, even if diagnosis re-use falls into the same complexity class of diagnosis generation (they are both NP-complete problems), practical advantages can be obtained by exploiting a hybrid architecture combining case-based and model-based diagnostic problem solving in a unifying framework.

1 Introduction

Interest for Case-Based Reasoning (CBR) has recently grown since this problem solving paradigm has been shown to be able to cope with tasks like planning, design and diagnosis (see [7] for a survey). Even if CBR has been initially proposed in domains where the domain knowledge is so weak that autonomous problem solving is difficult if not impossible, CBR has been claimed to provide significant benefits from a computational point of view also for problems where a strong domain theory is available. Some CBR systems has been successfully developed in order to speed-up the reasoning process in tasks where building a solution from scratch is in general a very hard problem [12, 2, 8]. The claim that CBR is the proper approach for problem solving has been sometimes supported by means of empirical studies (particular relevant for this point are the results obtained by using PRODIGY-ANALOGY [12]). However, recent investigations on case-based planning have shown that the theoretical computational complexity of re-using plans of previously solved problems is in general at least as hard as generating the plans from scratch [9]. In particular, this may happen even if the problem for which a plan exists is very similar to the problem to be solved (intuitively, adaptation of the plan to the new problem should be easy in such a case) .

The aim of this paper is to show that the hardness of re-using (and adapting) solutions to a new case is not peculiar to planning, but it can arise also in case-based diagnosis. So , it is hard to believe that CBR can completely replace model-based approaches to diagnosis, since in general the computational complexity of re-using a diagnostic solution is not lower than the complexity of generating a diagnostic solution from scratch. However, theoretical results are general results taking into account a worst-case analysis, whose real impact has often to be verified in practice. Therefore, in the present paper we propose both a theoretical and an empirical analysis of the basic adaptation mechanisms adopted in case-based diagnosis for modifying a retrieved solution in order to adapt it to be also a solution of the new problem to be solved. In particular, we focus out attention on the adaptation strategies implemented in ADAPtER (Abductive Diagnosis through Adaptation of Past Episodes for Re-use) [10], a diagnostic system which integrates a case-based and a model-based component, by taking into account a general and formal theory of diagnosis.

[1] Dipartimento di Informatica, Universita' di Torino (Italy)

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

2 Characterization of Diagnostic Problems

In order to discuss both theoretical aspects of adaptation complexity and practical implementation, we introduce the formal framework we refer to for characterizing diagnostic problems. The framework has been initially proposed in [5] as a general approach able to unify classical approaches to model-based diagnosis (namely purely consistency-based and purely abductive diagnosis).

Definition 1 *A diagnostic problem is a tuple* $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle \rangle$ *where:*

- M *is a set of definite clauses (without recursion) representing the behavioral model of the system to be diagnosed;*
- HYP *is a finite set of ground atoms of* M*, whose predicates are called* abducibles*, representing possible diagnostic hypotheses;*
- CXT *is a finite set of ground atoms of* M *representing contextual information characterizing the diagnostic problem;*
- Ψ^+ *is a finite set of ground atoms of* M *representing the observations to be accounted for (i.e. covered) in the current case;*
- Ψ^- *is a finite set of ground atoms of* M *representing the values of observable parameters conflicting with the observations.*

In the above definition, we characterize a diagnostic problem in terms of Ψ^+ and Ψ^-. Actually, a diagnostic problem is characterized by OBS, the set of observations available for the problem under examination. While Ψ^- is uniquely determined given OBS according to the criterion $\Psi^- = \{m(y) | m(x) \in OBS \wedge x \neq y\}$, there are many possible ways for determining Ψ^+ from OBS, since we only impose that $\Psi^+ \subseteq OBS$ [2].

We assume that each predicate occurring in M has a finite set of ground instances [3]. Moreover, since we abstract from temporal aspects, we also assume that the following meta-level constraint holds for every predicate symbol p:

$$p(x) \wedge p(y) \rightarrow \bot \quad (x \neq y)$$

A set of ground atoms is consistent if and only if it does not violate the above constraint.

[2] Different choices of Ψ^+ give raise to quite different definitions of diagnosis (see [5]), ranging from purely abductive definitions ($\Psi^+ = OBS$) to purely consistency-based definitions ($\Psi^+ = \emptyset$).

[3] Therefore, M is equivalent to a propositional definite clause theory.

Let H be a set of ground atoms, we indicate as $\mathcal{P}(H)$ the set of predicate symbols mentioned in H. If H is consistent, it will be called an *assignment* to $\mathcal{P}(H)$.

If $H \subseteq HYP$ is an assignment to abducible predicates then $\hat{H} = \{H_1, \ldots, H_n\}$ is the set of assignments H_i $(1 \leq i \leq n)$ such that $H \subseteq H_i$ and $\mathcal{P}(H_i) = \mathcal{P}(HYP)$ (i.e. every $H_i \in \hat{H}$ is an assignment obtained from H by adding a ground instance for each abducible predicate not mentioned in H).

Definition 2 *Given a diagnostic problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle\rangle$, an assignment $H \subseteq HYP$ $(H \neq \emptyset)$ is a diagnosis for DP if and only if*

$$\forall m(x) \in \Psi^+ \;\; M \cup CXT \cup H \vdash m(x)$$

$$\forall m(y) \in \Psi^- \;\; M \cup CXT \cup H \nvdash m(y)$$

$$\forall H_i \in \hat{H}, H_i \text{ satisfies the two above conditions}$$

If H contains exactly one instance for each abducible predicate it is said to be a *total diagnosis*; on the contrary it is said to be a *partial diagnosis*. Partial diagnoses do not mention predicates that are irrelevant for the current case. If H is a partial diagnosis, then $\hat{H}$ is the set of total diagnoses generated from H. Notice that $H = \emptyset$ is meaningless since, in the above characterization, this case would correspond to $OBS = \emptyset$ and consequently the diagnostic problem should not arise.

3 Complexity Results

In [3] it is shown that, apart from particular restrictions, solving an abductive problem is in general an NP-hard problem. Diagnostic problems satisfying definition 1 can be viewed as a kind of problems classified in [3] as *incompatibility abduction problems*: incompatibility relations are represented by the fact that different ground instances of the same abducible predicate are incompatible. In the following, when we will refer to a diagnostic problem we will consider a problem satisfying definition 1.

Definition 3 *Given a diagnostic problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle\rangle$ and an assignment to abducibles $H \subseteq HYP$:*

- *CONCHECK is the problem of deciding whether H is consistent with the observations (i.e. whether $M \cup CXT \cup H \nvdash \Psi^-$);*
- *COVCHECK is the problem of deciding whether H covers the observations to be accounted for (i.e. whether $M \cup CXT \cup H \vdash \Psi^+$);*
- *DIAGCHECK is problem of deciding whether H is a diagnosis to DP (i.e. DIAGCHECK=CONCHECK+COVCHECK).*

Since it is well-known that verifying whether a given atom is a consequence of a set of propositional definite clauses is linear in the size of the set of clauses [6], we have the following proposition.

Proposition 1 *CONCHECK, COVCHECK and DIAGCHECK are in P*

Let us then consider the following decision problem.

Definition 4 *DIAGSAT is the decision problem consisting of determining whether an instance of a diagnostic problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle\rangle$ has a solution.*

In [3], incompatibility abduction problems have been proved to be NP-hard in general, with some special class of incompatibility problems (i.e. *independent incompatibility abduction problems*) being NP-complete. We can show that the NP-completeness property also holds for DIAGSAT.

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

Theorem 1 *DIAGSAT is NP-complete.*

Proof. First we prove that DIAGSAT is NP-hard by means of a reduction from SAT (the problem of deciding the satisfiability of a boolean formula in CNF) that is known to be NP-complete [1]. Let ϕ be a boolean formula in CNF with variables $V = \{v_1, v_2, \ldots v_n\}$ and clauses $C = \{c_1, c_2, \ldots c_m\}$. An assignment of truth values to every variable in V is called a *variable assignment*. We will construct a diagnostic problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle\rangle$ such that DP has a solution if and only if ϕ is satisfiable.[4] Without restriction, let us consider a propositional model M using the following propositional letters:

* T_i $(1 \leq i \leq n)$ meaning that v_i=true has been selected,
* F_i $(1 \leq i \leq n)$ meaning that v_i=false has been selected,
* S_i $(1 \leq i \leq n)$ meaning that a truth value for v_i has been selected,
* C_j $(1 \leq j \leq m)$ meaning that c_j is satisfied.
* $\overline{C_j}$ $(1 \leq j \leq m)$ meaning that c_j is not satisfied.

The model M is composed by the following clauses:

* $T_i \rightarrow S_i$ $(1 \leq i \leq n)$
* $F_i \rightarrow S_i$ $(1 \leq i \leq n)$
* $T_i \wedge S_1 \wedge \ldots S_n \rightarrow C_j$ if $v_i \in c_j$
* $F_i \wedge S_1 \wedge \ldots S_n \rightarrow C_j$ if $\overline{v_i} \in c_j$
* $T_{i_1} \wedge \ldots T_{i_k} \wedge F_{j_1} \wedge \ldots F_{j_h} \wedge S_1 \wedge \ldots S_n \rightarrow \overline{C_j}$ if $c_j = v_{j_1} \vee \ldots v_{j_h} \vee \overline{v_{i_1}} \vee \ldots \overline{v_{i_k}}$

We also have the following meta-level constraints:

$T_i \wedge F_i \rightarrow \bot \qquad C_j \wedge \overline{C_j} \rightarrow \bot$

We further assume $HYP = \{T_i\} \cup \{F_i\}$, $CXT = \emptyset$, $\Psi^+ = \{C_1, \ldots C_m\}$ and $\Psi^- = \{\overline{C_1}, \ldots \overline{C_m}\}$.

Notice that the last three clauses of the model M are mutually exclusive; they model the fact that C_j is satisfied (third and fourth clause) or not satisfied (fifth clause). Indeed, every assignment $H \subseteq HYP$ such that $M \cup CXT \cup H \vdash C_j$ $(1 \leq j \leq m)$ is such that $M \cup CXT \cup H \nvdash \overline{C_j}$ $(1 \leq j \leq m)$. Moreover, every assignment $H \subseteq HYP$ and such that $\mathcal{P}(H) = \mathcal{P}(HYP)$ corresponds to a variable assignment and vice versa.

If ϕ is satisfiable, the set $H \subseteq HYP$ corresponding to the satisfying variable assignment is a diagnosis for DP; indeed, since every c_j is satisfied, then every C_j will be derived from H and consequently no $\overline{C_j}$ will be derived from H. Conversely, if DP has a solution, let H be a total diagnosis for DP, then the variable assignment in V correspondent to H clearly satisfies ϕ.

Finally, DIAGSAT is in NP since the following is a non deterministic algorithm running in polynomial time (see proposition 1):
- guess a solution H to DP;
- DIAGCHECK on H and DP. □

Let us then define what we mean by diagnosis adaptation problem. We are essentially interested in studying adaptation strategies that can be classified as deletion and addition of abducibles. As also noticed in [9], case-based systems usually adopt a conservative approach trying to re-use as much as possible of the retrieved solution to be adapted; we will then consider the following problem.

Definition 5 Diagnosis Adaptation Problem. *DASAT is the decision problem defined as follows: given a diagnostic problem $DP_1 = \langle M, HYP, CXT_1, \langle \Psi_1^+, \Psi_1^- \rangle\rangle$, a diagnosis H to the problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle\rangle$ and an integer $k \leq |H|$,*

[4] We use the approach proposed for the proof of Theorem 4 in [9] as a guideline.

determine whether there exists a diagnosis H' to DP_1 containing a sub-assignment of H of cardinality at least k.

In order to transform H into H', we have to define an *adaptation strategy* $\mathcal{A}$. Such a strategy has to determine which abducibles to be deleted and which others to be added to H, in order to obtain H'; the number of abducibles that $\mathcal{A}$ deletes must be at most $|H| - k$ abducibles (in order to have H' containing a sub-assignment of H of cardinality at least k). In the following, we will use the notation DASAT_k for parametrizing the problem on the integer k.

Theorem 2 *DASAT is NP-complete.*

Proof. As for theorem 1, we show that the problem is NP-hard by using a reduction from SAT characterized by the set of boolean variables $V = \{v_1 \dots v_n\}$ and the set of clauses $C = \{c_1 \dots c_m\}$. Let us construct an instance of DASAT_0 that can be satisfied if and only if SAT can be. Consider the following diagnostic problem $DP = \langle M, HYP, CXT, \langle \Psi^+, \Psi^- \rangle \rangle$ where atoms and clauses in M and sets CXT and HYP are the same as in the proof of theorem 1. We assume that:

$\Psi^+ = \{C_1, \dots, C_{m-1}\}$; $\Psi^- = \{\overline{C_1}, \dots, \overline{C}_{m-1}\}$

Let $H = \{\alpha_1, \dots, \alpha_n\}$ be a total diagnosis for DP ($\alpha_i \in \{T_i, F_i\}$); consider now the diagnostic problem

$DP_1 = \langle M, HYP, CXT, \langle \Psi_1^+, \Psi^- \rangle \rangle$ where $\Psi_1^+ = \{C_1, \dots C_m\}$

Suppose that the SAT formula is satisfiable and that σ_V^C is a satisfying truth assignment for variables in V and clauses in C; we can build a solution H_1 to DP_1 by keeping every T_i and F_i in H corresponding to the assignment of v_i that is consistent with σ_V^C. Every other abducible of H is then substituted with the abducible corresponding to the complementary assignment of the corresponding variable (i.e. T_i is deleted and F_i is added to H or vice versa). It is clear that we delete at most n abducibles (i.e. H_1 contains a sub-assignment of H of cardinality at least $k = 0$) and that H_1 is a diagnosis for DP_1.

Conversely, if there exists an assignment to abducibles H_1 that derives Ψ_1^+ (and so it does not derive Ψ^-) adapted from H and involving at most n deletions, then the SAT formula is satisfiable by means of the truth assignment corresponding to H_1.

The inclusion in NP is simply proved by the following polynomial time non-deterministic algorithm:

- guess H_1 from H through the deletion of at most n abducibles and the addition, for each deleted atoms of its complementary atom;

- verify that $\{C_1, \dots C_m\}$ derives from H_1 (and $\{\overline{C_1}, \dots, \overline{C}_{m-1}\}$ does not derive from H_1). □

If we consider DA1SAT to be the decision problem DASAT restricted to the case where the set $\Psi_1^+ = \Psi^+ \cup \{\alpha\}$ (i.e. observations to be accounted for in the new problem differ only for one atom from those of the old problem), then we obtain the following corollary.

Corollary 1 *DA1SAT is NP-complete*

Theorem 2 and Corollary 1 show that the problem of adapting in a conservative way (i.e. by re-using as much as possible of the retrieved solution) can be as hard as generating the new solution from scratch, even if the retrieved and the current case are very similar; this corresponds to the results obtained for planning in [9].[5] However, experimental results show that we may obtain quite significant improvements with respect to the generation of diagnosis from scratch. Even if theoretical results warn us about the danger of falling into the same degree of intractability of generation, adaptation strategies may have an average complexity better than generation.

[5] Results analogous to Theorem 2 and Corollary 1 in [9] are still worse, since they show that for a planning problem for which plan generation is polynomial, plan adaptation is NP-complete.

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

4 Implementing Diagnosis Re-use: the ADAPtER system

The aim of this section is to briefly introduce the adaptation strategies we have implemented in ADAPtER (see [10] for more details). Such a system has an architecture integrating a case-based module with a model-based diagnostic module able to solve a new case from scratch. The former module is composed by a case memory storing already solved cases with their solutions, by a retrieval and matching procedure able to extract cases from the memory and by an adaptation module implementing the DASAT problem using specific adaptation strategies. The model-based module consists in the abductive inference engine of the AID system [4] and is essentially devoted to solving a new problem when the case-based component fails in its task. Adaptation mechanisms form the core strategies for re-using the solutions of retrieved cases, in order to obtain a solution for a new problem. Since ADAPtER uses the theory of diagnosis introduced in section 2 as a framework of reference, this framework is also taken into account by the adaptation strategy. More precisely, the adaptation strategy is based on the activation in sequence of the following mechanisms: *consistency checking*, *inconsistency removal* and *explanation construction*. These basic mechanisms implement the notion (discussed in section 3) of adaptation as conservative deletion and addition of abducibles.

Let us suppose that given a new diagnostic problem characterized by $\langle CXT_1, OBS_1 \rangle$, the retrieval component of ADAPtER retrieves a case $C = \langle CXT, \langle \Psi^+, \Psi^- \rangle, H \rangle$ from the case memory, where H is a diagnostic solution (according to definition 2) for the problem represented by C. The first step of adaptation strategy (*consistency checking*) corresponds to solving an instance of the CONCHECK problem. In particular, we have to verify whether

$$\forall m(y) \in \Psi_1^- \quad M \cup CXT_1 \cup H \not\vdash m(y) \tag{1}$$

where $\Psi_1^- = \{m(y) | m(x) \in OBS_1 \wedge x \neq y\}$.

This process corresponds to a form of *derivational replay* [12], since we apply the solution H of the retrieved case C in the context of the new case to be solved (i.e. by using CXT_1 and Ψ_1^-). In fact, if condition 1 holds, H is a solution for the problem characterized by $\langle CXT_1, OBS_1 \rangle$ if the user chooses a purely consistency-based characterization of diagnosis (that is $\Psi_1^+ = \emptyset$). In case the user actually requires the covering of some observations (that is $\Psi_1^+ \neq \emptyset$), H is still a diagnosis if

$$\forall m(x) \in \Psi_1^+ \quad M \cup CXT_1 \cup H \vdash m(x) \tag{2}$$

This kind of verification represents an instance of the COVCHECK problem, so the overall process takes a polynomial time, being an instance of DIAGCHECK.

If an inconsistency is pointed out, that is condition 1 does not hold, then consistency must be re-established by the step of inconsistency removal; this mechanism disproves the explanation leading to the discovered inconsistency, by removing instances of abducibles responsible for such an inconsistency (see [10] for more details).

The explanation construction mechanism builds abductive explanations for observations to be explicitly accounted for. This step has to be performed either because the retrieved solution H does not cover all data contained into Ψ_1^+ (that is condition 2 does not hold) or the inconsistency removal step has deleted from H abducibles accounting for some of the data to be covered. It stops as soon as a piece of the retrieved solution not disproved by the step of inconsistency removal is reached. In this way the amount of search for determining

an abductive explanation is reduced and a kind of conservative approach is realized. This step can clearly lead to the addition of new abducibles to the retrieved solution.

In the following, we will discuss the practical impact of the theoretical results of section 3 on adaptation strategies of ADAPtER, by means of some experimental tests. It is worth noting that, in order to be conservative, adaptation strategies of ADAPtER make use of some heuristic criteria; in particular, when more than one possibility exist for removing an inconsistency or for explaining a new datum, the choice that locally allows the maximum re-use of the solution to be adapted is made. This and other heuristic criteria, adopted for reducing the search space, may introduce incompleteness in the adaptation strategies. Therefore, the adaptation process may fail ending without a diagnosis for the case under examination. This does not imply that ADAPtER does not find a diagnosis, since in case of failure of the adaptation module, the pure model-based module (i.e. AID) is invoked for solving the problem from scratch.

5 Experimental Results

In this section, we will analyse some experiments concerning the performance of ADAPtER, with particular attention to the problem of adaptation of a retrieved solution. The results we have obtained show that, despite the hardness results coming from theoretical complexity analysis, significant advantages can be practically obtained by adapting solutions to cases similar to the current one. However, because of the intractability of the adaptation problem in general, it is not surprising to encounter cases for which adaptation leads to a complete reorganization of the retrieved solution, resulting in a process computationally harder than generating the diagnostic solution from scratch. We performed a series of experiments on two different domains: a car engine fault domain and a medical domain concerning the leprosis disease. In the following we will report on experiments performed on the latter, since it is the most significant one (results obtained for the former are very similar). All the experiments described below have been performed on a version of ADAPtER written in SICStus prolog running on a SUN Sparc Station 20 with 64 Mbytes of memory.

We tested the adaptation strategy of ADAPtER on different runs of a batch of 70 different cases, chosen in such a way to account for the relevant part of patho-physiological evolutions considered in the model. In particular, we will report on results concerning experiments we carried out by randomly storing 30 cases out of 70 and using the remaining 40 as test cases.[6] Notice that this is an extreme situation adverse to adaptation, since there is no possibility of using a stored case as a test case. A first kind of test we performed is a direct comparison between the global performance of ADAPtER (retrieval, match and adaptation) and that of AID. Since AID has been integrated into ADAPtER, they share the same kind of implementation and a direct comparison in terms of computation time is a fair comparison. We determined the percentage gain of ADAPtER vs AID on a case C defined as

$$G(C) = \frac{T^C_{AID} - T^C_{ADAPtER}}{T^C_{ADAPtER}}$$

where T^C_{AID} and $T^C_{ADAPtER}$ are the CPU time of AID and ADAPtER respectively, needed to solve the case C. Figure 1 shows the results we obtained. Among the 40 cases used for testing, 2 cases were not be solved through adaptation. The remaining 38 cases shows a considerable better performance of the adaptation approach with

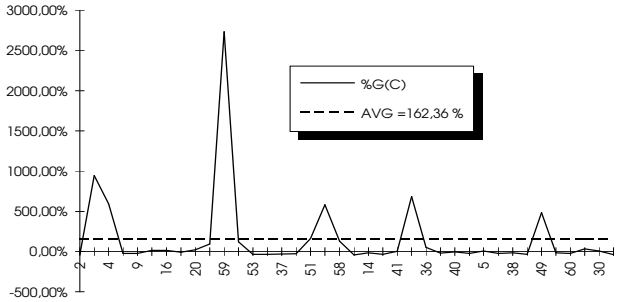

Figure 1. Percentage Gain ADAPtER vs AID

respect to the generation, with an average gain of about 162%. Indeed, even if in 22 cases out of 40, AID performed better than ADAPtER, cases that resulted advantageouses for generation showed a very small gain, while cases favourable to adaptation showed a considerable (sometimes a huge) gain.

A second kind of analysis that has been done for comparing the performance of the two systems was to consider how the typology of cases influences the case-based approach (and in particular the adaptation phase). Figure 2 shows the relative performances of AID and ADAPtER in a two-dimensional space; point in such a space represents the different cases solved by the systems and are distinguished into three different classes: *square* points represents cases solved by ADAPtER employing only consistency checking, *triangular* points are for cases requiring the explanation construction step, while *circular* points represent cases requiring both inconsistency removal and explanation steps. Points above the bisector line indicates cases for

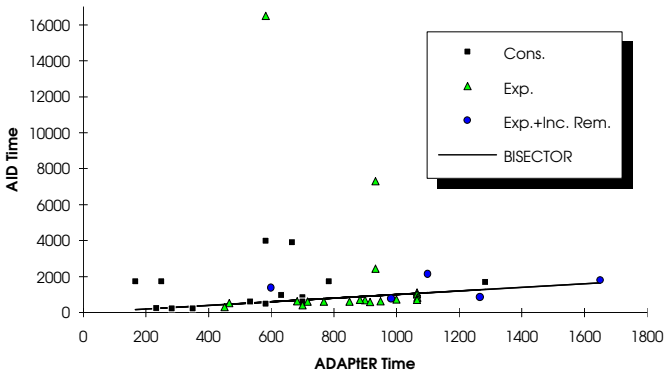

Figure 2. Case Typology Analysis

which ADAPtER performed better than AID and vice versa. As noticed above, points under the bisector line are very close to the line itself. Not very surprisingly, cases requiring only consistency checking show, in general, a considerable better performance of ADAPtER. However, significant advantages can also be obtained with cases of the other two classes, even if the major part of such cases fall under the bisector.

A last type of analysis we carried out on the experimental results for the leprosis diagnosis domain, concerned the impact of the adaptation phase on the global performance of ADAPtER; results are show in figure 3. It is possible to notice that adaptation is in general a significant part of the work performed by the case-based system, but

[6] Similar results have been obtained by considering the number of stored cases to be 20 and 40.

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

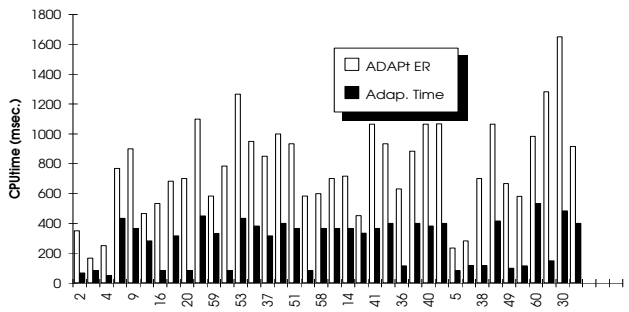

Figure 3. Adaptation vs Total Execution Time

we can also notice that more than half of the work is usually devoted to case retrieval and matching.

Finally, we tested the impact of the heuristics used in the adaptation phase of ADAPtER, with respect to the ability of adaptation to provide a solution to the current problem. In particular, we randomly generated a knowledge base having some features which makes it different from other ones considered in the experiments and not very adquate to the use of the above heuristics. In particular, in this knowledge base the number of atoms form which a given atom depends is usually quite large; this can be the cause of the failure of the adaptation phase since the heuristics removes an inconsistency on a given atom $p(a)$, by considering the removal of single atoms on which $p(a)$ depends instead of all the possible combinations of such atoms. We also tested ADAPtER vs AID on this randomly generated knowledge base by using a batch of 70 cases (40 in memory and 30 for test). Not surprisingly, the structure of such a knowledge base had a significant impact on adaptation, resulting in 13 cases out of 30 not being solved through adaptation. However, as for previous results we have discussed, the remaining cases show a significant advantage of the case-based approach with respect to diagnosis generation (see figure 4 which shows that the average gain is about 276%).

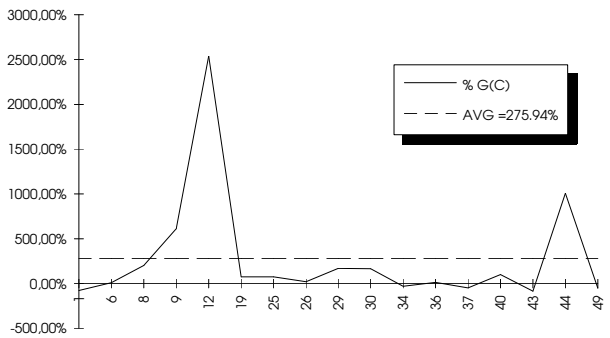

Figure 4. Results for Randomly Generated KB

6 Conclusions

In the present paper we have shown that there is no guarantee that the solution to a diagnostic problem can be easily adapted to be a solution for a very similar problem (see Theorem 2 and Corollary 1). Computational complexity results show that putting a lot on emphasis on similarity does not always pay the effort. A more principled approach, which tries to point out the sources of computational complexity, seems to be needed. In particular, we have shown that there are limited forms of adaptation (the consistency checking step in ADAPtER) that can be performed in polynomial time and that under certain conditions the CBR approach can give benefits; the savings of CPU time obtained by exploiting a case-based component are quite significant (see figure 1 and figure 4). In our opinion, the integration with model-based techniques is a key point, since an hybrid system combining case-based and model-based approaches does not require that the case memory contains a lot of cases covering all the interactions among diagnostic hypotheses. A case memory with a limited number of solved cases can be searched for in a reasonable time (note that our experiments show that adaptation time is just a fraction of the whole time taken by ADAPtER - see figure 3) and if adaptation fails (or no similar case can be retrieved) the model-based component is used for solving the problem. Opportunistic strategies for supervising the control in ADAPtER are under evaluation: some results in using a function able to predict the adaptation efforts are reported in [10], while the problem of deciding which are the "best" cases to store in the case memory has still to be investigated. Recent results described in [11] show that a right solution to this problem can provide significant benefits.

Acknowledgements

We thank C. Todaro for his help in implementing the SICStus prolog version of ADAPtER and in running the experiments. This research has been partially supported by grants from MURST 40% and CNR.

Appendix: Answers to Questionnaire

Our approach, implemented in the system ADAPtER, is based on the integration of a Case-based reasoner with a model-based one; this means that detailed background knowledge is supposed to be available for both the adaptation process and for the solution from scratch of a given problem. This can be different from other approaches where there is knowledge about adaptation available, but this is not sufficient for solving the problem. In the following we will report the answer to the questions of the two clusters we selected from the workshop questionnaire.

Structure adaptation

Why do you (not) use a structural representation rather than a flat one?
The case representation has two different levels of structure; the first is the distinction between contextual information and observations to be explained (findings), both used as features inside cases; the second is the fact that the solution stored in a case is structured. Instead of simply storing the set of diagnostic hypotheses representing the solution to the given case, we also store the whole causal path from the above hypotheses to the ultimate findings. This is crucial for adaptation, since the whole adaptation process can be viewed as a restructuring of such a causal path.

How do you match structure? How do you improve efficiency?
The structure of solutions is matched by taking into consideration the causal knowledge base used by the model-based reasoner. Efficiency is obtained by using a heuristic estimate of the adaptation effort; moreover, when removing an inconsistency during adaptation, we also avoid to consider all the possible ways of restoring consistency. This may cause the failure of the adaptation process, but it does not

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

preclude the possibility of solving the case, since the model-based reasoner is always available.

Could you match by unification?
We essentially work at the ground level.

What and how much do you transfer?
We transfer from the retrieved solutions complete pieces of causal paths.

Do you abstract from the original case representation to the structural representation? How can you translate the results back to the original format?
We do not have any abstraction and/or translation.

What about the qualitiy of the result? Can you predict it?
We can predict the adaptation effort through a heuristic estimate taking into account the different features of the retrieved and the current case and their role in the problem to be solved (see the distinction between features to be covered and those to be consistent with).

Case structure for different tasks

What is decision support? Is it an analytic or a synthetic task?
Our system is a diagnostic system. The diagnostic task is in this case more suitably characterized as a synthetic task, since it can be viewed as a kind of hypothesis assembling

What is a suitable case representation for decision support, planning, design, configuration, classification, diagnosis?
Obviously, the task to be performed can greatly influence the structure of a case. The formal characterization of the task to be solved can be essential to the definition of the case structure. the structure of a case. However, even for tasks having similar problem description, like for instance classification and diagnosis, differences may be relevant (for example the information to be stored as solution to the case).

Should problem and solution parts be distinguished? Should cases be structured into subcases? What kind of subcases?
Problem and solution part should clearly be distinguished, however cases can be hardly structured into subcases because of the interaction of faults. This may relevantly impact the adaptation process.

What preconditions allow to structure cases in these ways?
In the general case, a diagnostic problem may have a lot of interaction between faults and this preclude the subdivision of cases into independent subcases.

What are the dis/advantages?
Tha advantages are that interactions can be dealt with, the disadvantages is that (in general) we cannot take advantages of independently solving different sub-problems.

Does the case representation limit the tasks that can be solved?
Such a representation is very suitable for abductive tasks like diagnosis, but can also be useful for similar tasks like monitoring, integration of diagnosis and recovery actions, plan recognition etc....

How does case structure influence indexing, similarity and adaptation?
In our system, the availability of a deep knowledge base allows us to determine the features to be used as indeces (findings up to a given level of difficulty, contextual information), while similarity and adaptation are close related to the fact that the features listed in a case are semantically interpreted through the causal knowledge base.

What kind of similarity and indexing is required to retrieve cases without a separation?
We adopt the usual retrieval strategy adopted in discrimination (E-MOP based) networks.

© 1996 L. Portinale and P. Torasso
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

REFERENCES

[1] J.L. Balcazar, J. Diaz, and J. Gabarro, *Structural Complexity I*, Springer Verlag, 1988.

[2] B. Bartsch-Spoerl, 'Towards the integration of case-based, schema-based and model-based reasoning for supporting complex design tasks', in *LNAI 1010*, 145–156, Springer Verlag, (1995).

[3] T. Bylander, D. Allemang, M. Tanner, and J. Josephson, 'The computational complexity of abduction', *Artificial Intelligence*, **49**(1-3), 25–60, (1991).

[4] L. Console, L. Portinale, D. Theseider Dupré, and P. Torasso, 'Combining heuristic and causal reasoning in diagnostic problem solving', in *Second Generation Expert Systems*, eds., J.M. David, J.P. Krivine, and R. Simmons, 46,68, Springer Verlag, (1993).

[5] L. Console and P. Torasso, 'A spectrum of logical definitions of model-based diagnosis', *Computational Intelligence*, **7**(3), 133–141, (1991).

[6] W.F. Dowling and J.H. Gallier, 'Linear time algorithms for testing the satisfiability of propositional Horn clauses', *Journal of Logic Programming*, **1**, 267–284, (1984).

[7] J.L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, 1993.

[8] P. Koton, 'Using experience in learning and problem solving', Technical report, MIT/LCS/TR-441, (1989).

[9] B. Nebel and J. Koehler, 'Plan reuse versus plan generation: a theoretical and empirical analysis', *Artificial Intelligence*, **76**, 427–454, (1995).

[10] L. Portinale and P. Torasso, 'ADAPtER: an integrated diagnostic system combining case-based and abductive reasoning', in *LNAI 1010*, 277–288, Springer Verlag, (1995).

[11] B. Smyth and M.T. Keane, 'Remembering to forget', in *Proc. 14th IJCAI*, pp. 377–382, Montreal, (1995).

[12] M. Veloso, *Planning and learning by analogical reasoning*, LNAI 886 , Springer Verlag, 1994.