# Personalized Diagnosis for Over-Constrained Problems *

**Alexander Felfernig, Monika Schubert, and Stefan Reiterer**
Graz University of Technology
Graz, Austria
{alexander.felfernig, monika.schubert, stefan.reiterer}@ist.tugraz.at

## Abstract

Constraint-based applications such as configurators, recommenders, and scheduling systems support users in complex decision making scenarios. Typically, these systems try to identify a solution that satisfies all articulated user requirements. If the requirements are inconsistent with the underlying constraint set, users have to be actively supported in finding a way out from the *no solution could be found* dilemma. In this paper we introduce techniques that support the calculation of personalized diagnoses for inconsistent constraint sets. These techniques significantly improve the diagnosis prediction quality compared to approaches based on the calculation of minimal cardinality diagnoses. In order to show the applicability of our approach we present the results of an empirical study and a corresponding performance analysis.

## 1 Introduction

Constraint-based applications such as configurators, recommenders, and scheduling systems support users in complex decision making scenarios. Interacting with constraint-based applications often means to *specify* a set of requirements (e.g., when interacting with a car configurator, required components such as *car type* and *park distance control*), to *adapt* inconsistent requirements, and to *evaluate* different alternative solutions. In this paper we focus on situations where the constraint solver is not able to identify a solution and it is difficult for the user (customer) to identify minimal sets of requirements that need to be changed such that a solution for the underlying constraint satisfaction problem (CSP) can be identified. In order to improve the prediction quality of diagnosis algorithms in such contexts, we show how to exploit personalization techniques [Felfernig *et al.*, 2007].

Existing approaches to the determination of diagnoses for inconsistent requirements are primarily focusing on *minimal-cardinality diagnoses* [Felfernig *et al.*, 2004] which are determined on the basis of breadth-first search. In the context of recommender systems [Felfernig *et al.*, 2007] the complement of a diagnosis is denoted as *maximally successful sub-query* [Godfrey, 1997; McSherry, 2004]. Such a maximally successful subquery contains maximal sets of elements (requirements) that guarantee the identification of a solution, i.e., elements which are not part of the minimal diagnosis. In the context of constraint-based systems [Tsang, 1993] diagnoses are also interpreted as a specific type of *explanation* [OSullivan *et al.*, 2007].

Especially in interactive settings the determination of all diagnoses is infeasible due to unacceptable runtimes of the underlying diagnosis algorithms [Felfernig, IJCAI 2009]. Furthermore, we are not able to guarantee that standard breadth-first search leads us to explanations that are acceptable for the user [OSullivan *et al.*, 2007]. The work of [OSullivan *et al.*, 2007] contributes to the tailoring of diagnoses in a way that makes the identification of acceptable diagnoses easier for the user – [OSullivan *et al.*, 2007] denote this type of diagnosis *representative explanations*. Representative explanations are diagnosis sets that fulfill the criteria that each element contained in at least one diagnosis is also contained in the set of diagnoses presented to the user. [Felfernig *et al.*, 2009] show how to exploit concepts of collaborative recommendation for improving diagnosis prediction quality – the concepts have been developed for a knowledge-based recommendation environment [Burke, 2000].

On the basis of this existing work, we show how to exploit different recommendation algorithms for the personalized identification of diagnoses. In our approach we exploit these algorithms for guiding best-first search in the construction of Hitting Set Directed Acyclic Graphs (HS-DAGs). The major contribution of this paper is the significant improvement of *diagnosis prediction quality* by the integration of state-of-the-art recommendation approaches (similarity-based, utility-based, probability-based, ensemble-based) with standard model-based diagnosis [Reiter, 1987; DeKleer *et al.*, 1992]. Furthermore, we provide an empirical evaluation on the basis of two configuration datasets.

The remainder of this paper is organized as follows. In Section 2 we introduce a working example from the domain of car configuration. In Section 3 we show how recommendation algorithms can be exploited for personalized model-based diagnosis. In Section 4 we present the results of evaluations conducted with two datasets. In Section 5 we discuss

*The work presented in this paper has been partially funded by the Austrian Research Promotion Agency (project 827587).

related work. The paper is concluded with Section 6.

## 2 Working Example

The configuration of cars will serve for illustration purposes throughout this paper. A configuration task can be defined as a constraint satisfaction problem (CSP) [Tsang, 1993]:[1]

**Definition 1 (Configuration Task)**. A configuration task can be defined as a CSP (V, D, C). V = $\{v_1, v_2, \ldots, v_n\}$ represents a set of finite domain variables. D = $\{$dom($v_1$), dom($v_2$), $\ldots$, dom($v_n$)$\}$ represents a set of variable domains dom($v_k$) where dom($v_k$) represents the domain of variable $v_k$. C = $C_{KB} \cup C_R$ where $C_{KB} = \{c_1, c_2, \ldots, c_q\}$ is a set of domain specific constraints (the configuration knowledge base) that restrict the possible combinations of values assigned to the variables in V. $C_R = \{c_{q+1}, c_{q+2}, \ldots, c_t\}$ is a set of customer requirements also represented as constraints.

A simple example of a configuration task is the following. The variable *type* represents the car type, *pdc* is the parc distance control feature, *fuel* represents the average fuel consumption per 100 kilometers, a *skibag* supports a convenient ski stowage inside a car, and *4-wheel* represents the actuation type (4-wheel supported or not supported). These variables are representing all possible user (customer) requirements. The possible combinations of customer requirements are restricted by $C_{KB}$ which is in our case $\{c_1, c_2, c_3, c_4, c_5\}$. Finally, we assume $C_R$ to be $\{c_6, c_7, c_8, c_9, c_{10}\}$.

- $V$ = $\{$***type***, ***fuel***, ***skibag***, ***4-wheel***, ***pdc***$\}$
- $D$ = $\{$dom(***type***)=$\{$*city*, *limo*, *combi*, *xdrive*$\}$, dom(***fuel***) = $\{$*4l*, *6l*, *10l*$\}$, dom(***skibag***)=$\{$*yes*, *no*$\}$, dom(***4-wheel***)=$\{$*yes*, *no*$\}$, dom(***pdc***)= $\{$*yes*, *no*$\}\}$
- $C_{KB}$ = $\{c_1$: ***4-wheel*** = *yes* $\Rightarrow$ ***type*** = *xdrive*, $c_2$: ***skibag*** = *yes* $\Rightarrow$ ***type*** $\neq$ *city*, $c_3$: ***fuel*** = *4l* $\Rightarrow$ ***type*** = *city*, $c_4$: ***fuel*** = *6l* $\Rightarrow$ ***type*** $\neq$ *xdrive*, $c_5$: ***type*** = *city* $\Rightarrow$ ***fuel*** $\neq$ *10l*$\}$
- $C_R$ = $\{c_6$: ***4-wheel*** = *yes*, $c_7$: ***fuel*** = *6l*, $c_8$: ***type*** = city, $c_9$: ***skibag*** = yes, $c_{10}$: ***pdc*** = yes$\}$

On the basis of this simple example of a configuration task, we can now introduce the definition of a corresponding *configuration* (solution to a configuration task).

**Definition 2 (Configuration)**. A configuration for a given configuration task (V, D, C) is an instantiation I = $\{$v$_1$=ins$_1$, v$_2$=ins$_2$, $\ldots$, v$_n$=ins$_n\}$ where ins$_k \in$ dom(v$_k$).

A solution (configuration) for a given configuration task is *consistent* if the assignments in I are consistent with the $\bigcup$c$_i$ $\in$ C. A solution is *complete* if all $v_i \in$ V are instantiated. Finally, a solution is *valid* if it is consistent and complete.

## 3 Calculating Personalized Diagnoses

Users do not want and are not able to evaluate large sets of diagnosis alternatives. For this reason we are now introducing alternative approaches that help to systematically reduce the number of diagnosis alternatives. Our goal is to identify diagnoses that are relevant for users and thus *keep the process of evaluating and selecting diagnoses as simple as possible*. The first approach to reduce the number of diagnoses (which is the only non-personalized one we consider here) is to perform *breadth first search* which returns minimal cardinality diagnoses first [Reiter, 1987]. In addition to this breadth first search approach we will discuss four approaches to the *personalized ranking of diagnoses*: *similarity-based*, *utility-based*, *probability-based*, and *ensemble-based* search.

**Cardinality-based diagnosis (not personalized).** Our example configuration task (car configuration) is defined in a way which does *not* allow the calculation of a solution, for example, the requirements $c_6$ and $c_8$ are incompatible. For identifying minimal sets of constraints which have to be deleted from the given set of customer requirements we use the concepts of Model-Based Diagnosis (MBD) [Reiter, 1987; DeKleer *et al.*, 1992]. MBD diagnosis exploits the description of a system – in our case the configuration knowledge base $C_{KB}$ which describes a set of possible configurations (solutions). If we detect that the behavior of the system conflicts with its intended behavior (at least one solution can be identified), the task of a diagnosis component is to determine components (constraints) in the given set of customer requirements ($C_R$) which, when assumed to function abnormally, sufficiently explain the discrepancy between actual and expected system behavior. An identified *minimal diagnosis* is a minimal set of faulty constraints that need to be relaxed or deleted in order to be able to calculate a configuration.

Assuming the existence of $C_{KB} = \{c_1, c_2, ..., c_q\}$ and $C_R = \{c_{q+1}, c_{q+2}, ..., c_t\}$ which is *inconsistent* with $C_{KB}$, breadth first search based diagnosis algorithms [Reiter, 1987; DeKleer *et al.*, 1992] determine minimal diagnoses $DIAGS = \{\Delta_1, \Delta_2, ..., \Delta_k\}$ in the order of their cardinality such that $\forall \Delta_i \in DIAGS : C_{KB} \cup (C_R$ - $\Delta_i)$ is consistent. A *User Requirements Diagnosis Problem (UR Diagnosis Problem)* can be defined as follows:

**Definition 3 (User Requirements (UR) Diagnosis Problem):** A UR Diagnosis Problem is defined as a tuple $(C_{KB}, C_R)$; $C_{KB}$ represents the constraints of the configuration knowledge base and $C_R$ is a set of user requirements.

Based on the definition of a UR Diagnosis Problem, a *UR Diagnosis* can be defined as follows:

**Definition 4 (UR Diagnosis):** A User Requirements Diagnosis (UR Diagnosis) for $(C_{KB}, C_R)$ is a set of constraints $\Delta \subseteq C_R$ such that $C_{KB} \cup (C_R - \Delta)$ is consistent. A diagnosis $\Delta$ is minimal iff there does not exist a diagnosis $\Delta' \subset \Delta$ s.t. $C_{KB} \cup (C_R - \Delta')$ is consistent.

Following the basic principles of Model-Based Diagnosis (MBD) [Reiter, 1987; DeKleer *et al.*, 1992], the calculation of diagnoses is based on the identification and resolution of conflict sets. A *conflict set* in $C_R$ can be defined as follows:

**Definition 5 (UR Conflict Set):** A User Requirements Conflict Set (UR Conflict Set) is defined as $CS \subseteq C_R$ s.t. $CS \cup C_{KB}$ is inconsistent. CS is minimal iff there does not exist a conflict set CS' with CS' $\subset$ CS.

In our working configuration example, $C_R = \{c_6, .., c_{10}\}$ is inconsistent with $C_{KB} = \{c_1, .., c_5\}$, i.e., there does not exist a configuration (solution) that completely fulfills the requirements in $C_R$. The *minimal conflict sets* are $CS_1 = \{c_6, c_7\}$, $CS_2 = \{c_8, c_9\}$, and $CS_3 = \{c_6, c_8\}$ since each of these

[1] Note that the presented concepts are applicable to different knowledge representations such as SAT solving [Marques-Silva and Sakallah, 1996] and description logics [Friedrich and Shchekotykhin, 2005].

conflict sets is inconsistent with $C_{KB}$ and there do not exist conflict sets $CS_1$', $CS_2$', and $CS_3$' with $CS_1' \subset CS_1$, $CS_2' \subset CS_2$, and $CS_3' \subset CS_3$.

In MBD [Reiter, 1987; DeKleer *et al.*, 1992], the standard algorithm for determining minimal diagnoses is the *hitting set directed acyclic graph* (HSDAG). UR diagnoses $\Delta_i \in DIAGS$ are determined by conflict resolution in the set of requirements $C_R$. Due to its minimality property, one conflict can simple be resolved by deleting one of the elements from the conflict set. After one element has been retracted from each of the given conflict sets, we are able to present a corresponding diagnosis. The original HSDAG approach employs breadth-first search. In our example, the diagnoses derived from $CS_1$, $CS_2$, and $CS_3$ are $DIAGS$ = $\{\Delta_1 : \{c_6, c_8\}, \Delta_2 : \{c_6, c_9\}, \Delta_3 : \{c_7, c_8\}\}$.

The HSDAG construction for our working example is shown in Figure 1. In our implementation we employ the QUICKXPLAIN conflict detection algorithm which has been developed by [Junker, 2004]. Following a strict *breadth first* search regime, we resolve the first conflict set ($CS_1$) by checking whether one of its elements already represents a diagnosis. Both alternatives ($c_6$ and $c_7$) do not lead to a diagnosis due to the inconsistency of ($C_R$ - $\{c_6\}$) $\cup$ $C_{KB}$ and ($C_R$ - $\{c_7\}$) $\cup$ $C_{KB}$. The next minimal conflict set returned by QUICKXPLAIN is $CS_2$ = $\{c_8, c_9\}$. $C_R$ - ($\{c_6\} \cup \{c_8\}$)) $\cup$ $C_{KB}$ allows the determination of a solution; consequently we have identified the first minimal diagnosis: $\Delta_1$ = $\{c_6, c_8\}$.

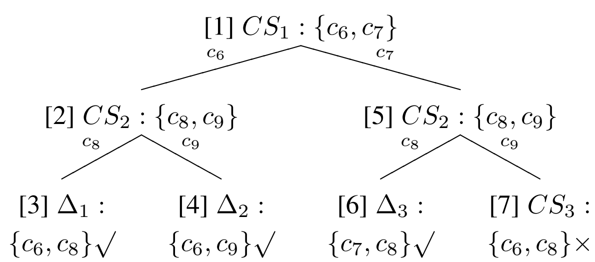


Figure 1: Cardinality-based diagnosis (breadth-first): the diagnoses ranking is $\{\Delta_1, \Delta_2, \Delta_3\}$. The expression [$\{c_6, c_8\}$ $\times$] denotes *containment*, i.e., the node can be closed.

**Similarity-based diagnosis.** The similarity-based ranking of diagnoses is based on the idea of preferring minimal diagnoses which lead to solutions (configurations) that resemble the original set of requirements as much as possible. We exploit the information contained in already existing configurations (see, e.g., Table 1). For each configuration contained in this table we determine its similarity with the given set of requirements – the similarity values of our working example are depicted in Table 3. The similarity-based determination of diagnoses is based on Algorithm 1 – a generic algorithm (PDIAG) which is applicable with different node expansion strategies (in our case, *cardinality*, *similarity*, *utility*, and *probability-based* search). This algorithm simulates the construction of a hitting set directed acyclic graph (HSDAG) [Reiter, 1987]. The function *deleteall*($\Delta'$, $H$) deletes $\Delta'$ and all supersets of $\Delta'$ from H, *delete*($\Delta'$, $H$) removes only $\Delta'$.

We determine similarity values based on three different

**Algorithm 1** PDIAG($C_R$, $C_{KB}$, $crit$, $n$): $\Delta$

```
{PDIAG returns ≤ n(n ≥ 0) personalized minimal diagnoses Δ (bag) for a given set of inconsistent user requirements (C_R) using the preference criteria defined in crit.}
{C_KB: configuration knowledge base}
{H: paths of the diagnosis search tree}
{CS: min. conflict set returned by theorem prover (TP)}
Δ = ∅; H = ∅;
repeat
  Δ' ← first(H);
  CS ← TP((C_R − Δ') ∪ C_KB);
  if isEmpty(CS) then
    Δ ← Δ ∪ Δ';
    H ← deleteall(Δ', H);
    n ← n - 1;
  else
    for all X in CS do
      H ← H ∪ {Δ' ∪ {X}};
    end for
    H ← delete(Δ', H);
    H ← sort(H, crit);
  end if
until (H = ∅ or n = 0);
return Δ;
```

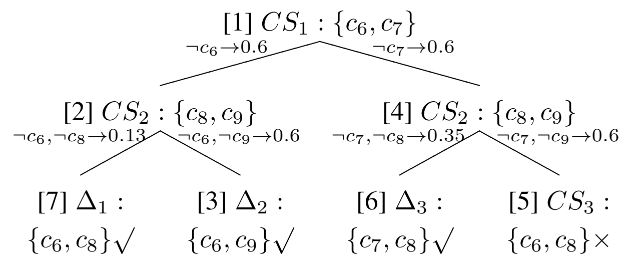


Figure 2: Similarity-based diagnosis: the diagnoses order is $\{\Delta_2, \Delta_3, \Delta_1\}$. The term $\neg c_6 \to 0.6$ denotes the fact that the highest similarity between $C_R$ and the tuples of the sessions $s_i$ in Table 1 consistent with $\neg c_6$ is 0.6 (in our case $i = 1$).

*attribute-level similarity measures* which are predominantly applied in knowledge-based recommender applications [McSherry, 2004]. The attribute-level measures determine the similarity of each attribute value $a_i$ of session $s_k$ and the corresponding requirement $c_i$ (e.g., the similarity between the attribute *type* of session $s_1$ and the corresponding requirement $c_8$). Depending on the characteristics of the attribute, one of the following attribute-level similarity measures has to be selected (see Formulae 1–3): *More-Is-Better* (MIB), *Less-Is-Better* (LIB) or *Nearer-Is-Better* (NIB) [McSherry, 2004]. The overall similiarity between $c = C_R$ and a tuple $a$ in Table 1 is defined by Formula 4. The similarity between $C_R$ ($c$) and a tuple $a$ of user interaction data is represented by the sum of weighted ($w(c_i)$ – see Table 2) attribute level similarities.[2]

For attributes such as *fuel*, the lower the value the more satisfied the user is (LIB). When the user specifies a certain car

[2]Our preference ($w(c_i)$) determination method is typically based on *multi attribute utility theory* [Winterfeldt and Edwards, 1986].

Table 1: Example user interaction data from already completed configuration sessions.

| SESSION $s_i$ | TYPE | FUEL | SKIBAG | 4-WHEEL | PDC |
|---|---|---|---|---|---|
| $s_1$ | city | 4l | no | no | yes |
| $s_2$ | city | 4l | no | no | no |
| $s_3$ | xdrive | 10l | yes | yes | yes |
| $s_4$ | limo | 6l | no | no | yes |
| $s_5$ | combi | 6l | no | no | no |
| $s_6$ | xdrive | 10l | no | yes | yes |
| $s_7$ | limo | 6l | yes | no | no |
| $s_8$ | combi | 6l | yes | no | no |

Table 2: Example importance values ($w(c_i)$ in %).

| TYPE | FUEL | SKIBAG | 4-WHEEL | PDC |
|---|---|---|---|---|
| 50.0 | 5.0 | 10.0 | 30.0 | 5.0 |

type (no intrinsic value scale), we suppose the most similar is the preferred one. In such cases, the nearer-is-better (NIB) similarity measure is used.[3]

$$MIB : s(c_i, a_i) = \frac{val(c_i) - min(a_i)}{max(a_i) - min(a_i)} \quad (1)$$

$$LIB : s(c_i, a_i) = \frac{max(a_i) - val(c_i)}{max(a_i) - min(a_i)} \quad (2)$$

$$NIB : s(c_i, a_i) = \begin{cases} 1 & if\ val(c_i) = val(a_i) \\ 0 & else \end{cases} \quad (3)$$

$$sim(c, a) = \sum_{i=1}^{n} \mathrm{s}(c_i, a_i) * w(c_i) \quad (4)$$

Table 3: Similarity ($sim(c, a)$) between requirements ($c = C_R$) and user interaction data ($a$) from configuration sessions.

| $s_i$ | TYPE | FUEL | SKIBAG | 4-WHEEL | PDC | *sim(c,a)* |
|---|---|---|---|---|---|---|
| $s_1$ | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.6 |
| $s_2$ | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.55 |
| $s_3$ | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.18 |
| $s_4$ | 0.0 | 0.67 | 0.0 | 0.0 | 1.0 | 0.08 |
| $s_5$ | 0.0 | 0.67 | 0.0 | 0.0 | 0.0 | 0.03 |
| $s_6$ | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.35 |
| $s_7$ | 0.0 | 0.67 | 1.0 | 0.0 | 0.0 | 0.13 |
| $s_8$ | 0.0 | 0.67 | 1.0 | 0.0 | 0.0 | 0.13 |

**Utility-based diagnosis.** Utility-based diagnosis prefers minimal diagnoses that are predominantly composed of requirements which are of *low importance* (a low $w(c_i)$ value) for the customer (user). Based on the concepts of *multi attribute utility theory* [Winterfeldt and Edwards, 1986], individual importance values (see Table 2) of user requirements that are part of a diagnosis are summed up – the lower this sum, the lower is the overall importance of the parameters contained in the diagnosis and the higher is the ranking of the corresponding diagnosis. The function *utility*($C \subseteq C_R$) returns a utility value for each set $C$ which is a subset of $C_R$ (see Formula 5). Note that in the case of our working example the diagnosis ranking on the basis of the utility-based approach is the same as with the discussed similarity-based approach. The corresponding results are depicted in Figure 3.

[3]For a detailed discussion of different types of similarity measures see, for example, [McSherry, 2004]. In Formulae 1 – 3, $val(c_i)$ denotes the value of user requirement $c_i$, $min(a_i)$ denotes the minimal possible value of configuration attribute $a_i$, and $max(a_i)$ denotes the maximal possible value of $a_i$.

Table 4: Example diagnoses selected by users, the individual probabilities are: p($\neg c_6$)=0.30, p($\neg c_7$)=0.50, p($\neg c_8$)=0.60, p($\neg c_9$)=0.20 where, for example, p($\neg c_6$)=0.30 denotes the probability of $c_6$ being part of a diagnosis.

| $\Delta_i$ | TYPE | FUEL | SKIBAG | 4-WHEEL | PDC |
|---|---|---|---|---|---|
| $\Delta_{log1}$ | ≠city | – | – | =no | – |
| $\Delta_{log2}$ | – | – | =no | =no | – |
| $\Delta_{log3}$ | ≠city | ≠6l | – | – | – |
| $\Delta_{log4}$ | ≠xdrive | – | – | – | – |
| $\Delta_{log5}$ | – | – | =no | =no | – |
| $\Delta_{log6}$ | ≠city | ≠6l | – | – | – |
| $\Delta_{log7}$ | ≠city | ≠6l | – | – | – |
| $\Delta_{log8}$ | ≠xdrive | ≠4l | – | =yes | – |
| $\Delta_{log9}$ | ≠city | ≠6l | – | – | – |
| $\Delta_{log10}$ | ≠city | ≠6l | – | – | – |

$$u(C \subseteq C_R) = \frac{1}{\sum_{c_i \in C} w(c_i)} \quad (5)$$

**Probability-based diagnosis.** Probability-based best first search for diagnoses prefers minimal diagnoses with a high *probability* of being selected by the user. For the determination of diagnosis probabilities we rely on joint probabilities that a particular diagnosis (or part of a diagnosis) will be selected by the user. Formula 6 is used for determining the joint probabilities for a given set of constraints $C \subseteq C_R$. Figure 4 shows the application of this approach in the context of our working example. The probabilities are determined on the basis of user-selected diagnoses (see Table 4). The assumption of *independence of failure* made here is one widely made in model-based diagnosis [DeKleer, 1990].

$$p(C \subseteq C_R) = \prod_{c_i \in C} p(c_i) \quad (6)$$

**Ensemble-based diagnosis.** In the case of cardinality-based, similarity-based, utility-based, and probability-based diagnosis search, diagnosis predictions (the rankings) are based on a single hypothesis. The idea of ensemble-based diagnosis search is to exploit a set of hypotheses (an ensemble) for making the prediction. For the ranking of diagnoses we apply a basic *majority voting* approach (see Table 5); assuming that the errors made by each individual prediction mechanism are not the same, ensemble-based methods can be very useful for improving the prediction quality (see Section 4). The determination of ensemble-based diagnoses is sketched in Algorithm 2 (ENSPDIAG); this algorithm activates PDIAG with each individual expansion criteria, collects the results, and determines a set of ensemble-based diagnoses. We do not

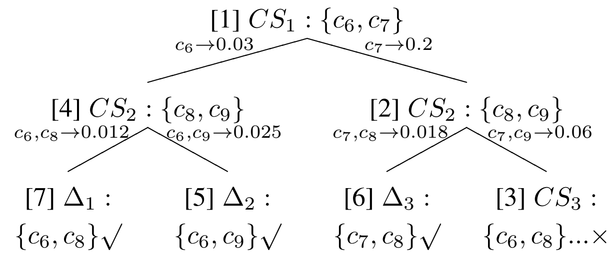


Figure 3: Utility-based diagnosis search: the order of identified diagnoses is $\{\Delta_2, \Delta_3, \Delta_1\}$.

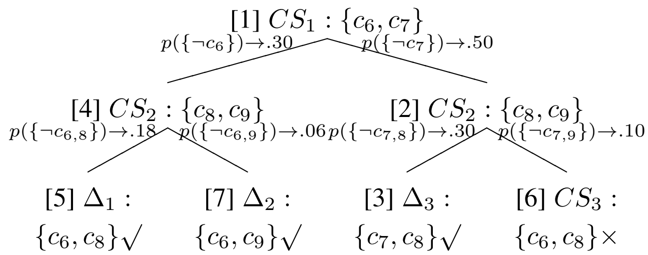


Figure 4: Probability-based diagnosis: the order of identified diagnoses is $\{\Delta_3, \Delta_1, \Delta_2\}$.

Table 5: Example diagnoses selected by ensemble method (implemented as a basic form of *majority voting*).

| METHOD / POSITION | 1 | 2 | 3 |
|---|---|---|---|
| utility-based | $\Delta_2$ | $\Delta_3$ | $\Delta_1$ |
| probability-based | $\Delta_3$ | $\Delta_1$ | $\Delta_2$ |
| similarity-based | $\Delta_2$ | $\Delta_3$ | $\Delta_1$ |
| ensemble-based | $\Delta_2$ | $\Delta_3$ | $\Delta_1$ |

provide the details of the function *CreateEnsembleDiagnoses* and refer to Table 5 which sketches the aggregation approach. Note that – since ENSPDIAG activates a HSDAG construction several times (in the for-loop) – conflicts determined with criteria $X$ can be reused in follow-up PDIAG activations.

**Algorithm 2** ENSPDIAG($C_R, C_{KB}, Crit, n$): $\Delta$

{ENSPDIAG returns $\leq n (n \geq 0)$ personalized minimal diagnoses $\Delta$ (bag) for a given set of inconsistent user requirements ($C_R$) using the ensemble-based approach.}
{$C_{KB}$: configuration knowledge base}
{$Crit$: set of expansion criteria (*card*, *sim*, *utility*, *prob*)}
{$\Gamma$: collection of diagnosis sets (bags)}
**for all** $X$ *in* $Crit$ **do**
  $\Gamma \leftarrow \Gamma \cup$ PDIAG($C_R, C_{KB}, X, n$);
**end for**
$\Delta \leftarrow$ CreateEnsembleDiagnoses($\Gamma$);
return $\Delta$;

## 4 Evaluation

**Prediction Quality.** We now demonstrate the improvements achieved by the application of our personalized diagnosis approaches on the basis of an empirical study with *two datasets*.

*Dataset 1: Computer Configuration.* This dataset has been composed on the basis of an online user experiment conducted at the Graz University of Technology. 415 subjects participated in the study (82,4% male and 17,6% female). Participants had to define their requirements ($C_R$) regarding a set of 12 *computer properties*. The task of the subjects was to define their product requirements (including requirement importance). After having completed the requirement specification phase, *each* participant was informed about the fact that no solution could be found. The configurator then presented a list of max. 50 different repair configurations (solution alternatives) – at least one property of each configuration was inconsistent with $C_R$. The configurations were extracted from *www.dell.at*. The ranking of the solution alternatives was randomized and the subjects were enabled to navigate in order to evaluate the solution alternatives regarding criteria such as *price*, *harddisk size* or *number of fulfilled requirements*. The subjects then had to select one out of the presented repair configurations that appeared to be the most acceptable one for them. Since no solution has been made available for $C_R$ (only configurations inconsistent with $C_R$ were shown) we could calculate conflicts (in $C_R$ induced by the repair configurations) and the corresponding diagnoses with our diagnosis techniques (the average number of diagnoses per $C_R$ was: 5.32 (std.dev. 1.67)). *Precision* (see Formula 7) was measured then in terms of how often a repair configuration selected by the participant was consistent with one of the top-N ranked diagnoses.

*Dataset 2: Financial Services.* This dataset belongs to a financial service recommender. In this application, inconsistent states and selected diagnoses had been recorded (N=1.703 sessions out of which in 418 sessions a diagnosis process had been activated – average number of diagnoses per $C_R$: 20.42 (std.dev. 4.51)). In the case of the financial services configurator, the importance of a user requirement ($w(c_i)$) was (is) determined on the basis of *multi attribute utility theory* [Winterfeldt and Edwards, 1986].

Based on the two datasets (*computer* and *financial services*) we evaluated our diagnosis approaches w.r.t. their *precision* (Formula 7). The idea of this precision measure is to figure out how often a diagnosis that corresponds to a *diagnosis selected by the user* or leads to a *repair configuration selected by a user* is among the top-n (N) ranked diagnoses.

$$precision = \frac{\#(correct\ predictions)}{\#(predictions)} \quad (7)$$

As can be seen in Table 6 and Table 7, in our two empirical settings the ensemble-based approach (majority voting) outperforms the other prediction methods in terms of *precision*. Breadth first search has the lowest precision. Based on a *two-sample t-test* we tried to figure out whether there exist statistically significant differences between the diagnosis approaches in terms of their *mean square error* ($\frac{1}{n}\sum_{i=1}^{n}|(1 - diagpos(i))^2|$) where $diagpos$ denotes the position of the diagnosis selected by the user and $n$=#(*diagnosis processes started*). In both datasets we detected a significant difference between breadth-first search and all other approaches (computer: $p = 2.2e^{-16}$, financial services: $p < 0.05$). Furthermore, there is a significant difference between the ensemble-based and the other personalized approaches in the case of the *computer* dataset ($p = 7.55e^{-6}$); in the case of the *financial services* dataset we can observe a tendency ($p < 0.07$).

Table 6: Predictive quality (precision) of used diagnosis selection methods for *computer* configuration dataset.

| METHOD / N | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| breadth first | 0.55 | 0.76 | 0.82 | 0.88 | 0.96 |
| utility-based | 0.66 | 0.83 | 0.94 | 0.95 | 0.98 |
| similarity-based | 0.65 | 0.81 | 0.90 | 0.93 | 0.98 |
| probability-based | 0.64 | 0.85 | 0.93 | 0.95 | 0.99 |
| ensemble-based | 0.68 | 0.86 | 0.94 | 0.96 | 0.99 |

Table 7: Predictive quality (precision) of used diagnosis selection methods for *financial services* configuration dataset.

| METHOD / N | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| breadth first | 0.12 | 0.27 | 0.39 | 0.52 | 0.62 |
| utility-based | 0.17 | 0.37 | 0.48 | 0.65 | 0.74 |
| similarity-based | 0.17 | 0.37 | 0.49 | 0.65 | 0.73 |
| probability-based | 0.15 | 0.33 | 0.47 | 0.57 | 0.74 |
| ensemble-based | 0.17 | 0.35 | 0.50 | 0.63 | 0.76 |

**Performance.** The PDIAG algorithm (Algorithm 1) has been implemented on the basis of the standard hitting set algorithm introduced in [Reiter, 1987] – it is NP-hard in the general case but is applicable for interactive configuration settings (see the following evaluation). In PDIAG minimal conflict sets are determined on the basis of QUICKXPLAIN [Junker, 2004] – the worst case complexity in terms of the number of consistency checks of QUICKXPLAIN is O($2k * log(\frac{n}{k}) + 2k$) where $k$ represents the size of the minimal conflict set and $n$ is the number of constraints.

We conducted a performance analysis in order to show the applicability of our approach (see Table 8). The tests have been executed on a standard desktop computer (*Intel®Core™2 Quad CPU Q9400* CPU with *2.66GHz* and *2GB* RAM). In addition to our datasets we evaluated our diagnosis algorithms with the Renault configuration knowledge base part of the *configuration benchmark suite*.[4] Even for complex settings (Renault benchmark) we can expect a performance acceptable for interactive settings. Note that the determination of minimal cardinality diagnoses typically takes less time due to the fact that fewer conflict sets have to be determined on an average, however, in many settings minimal cardinality diagnoses are not the preferred ones.

Table 8: Avg. runtime(*msec*) for determining the first-N diagnoses (bf = *breadth first*, pers=*all personalized approaches*).

| DATASET | APPROACH | N=1 | N=5 |
|---|---|---|---|
| computer configuration | bf | 64.1 | 70.0 |
| computer configuration | pers | 64.2 | 71.1 |
| financial services | bf | 23.3 | 36.8 |
| financial services | pers | 23.9 | 37.1 |
| car (Renault) | bf | 921.3 | 1510.1 |
| car (Renault) | pers | 952.7 | 1581.9 |

**Monotonicity Assumption.** Note that the PDIAG algorithm is able to determine the best $n$ (preferred) diagnoses under the assumption that the underlying expansion strategy fulfills the *monotonicity* property of best-first search ([Russell and Norvig, 2003]) which holds for the strategies we discussed in this paper: the *cardinality* increases, *similarity*, *utility*, and *probability* decrease with a growing expansion level of the HSDAG [Reiter, 1987].

## 5 Related Work

The increasing size and complexity of knowledge bases led to the application of model-based diagnosis [Reiter, 1987; DeKleer *et al.*, 1992] to automated knowledge base debugging [Felfernig *et al.*, 2004]. The contribution of [Felfernig *et al.*, 2004] has a special relationship to the concepts presented in this paper: [Felfernig *et al.*, 2004] identify faulty constraints in configuration knowledge bases, furthermore, they present a first approach to the identification of minimal sets of faulty user requirements (following a breadth-first search regime). An approach to personalize diagnosis has been presented in [Junker, 1994] where diagnoses are determined on the basis of a HSDAG [Reiter, 1987] given a defined preferred ordering on assumptions. In a similar line, [DeKleer, 1990] show the application of probability estimates for the identification of the most relevant diagnosis. The work presented in this paper extends existing research results by demonstrating the application of recommendation algorithms for improving the prediction quality of diagnosis algorithms. In order to identify the relevant diagnoses, we have to include intelligent estimates about preferences which is one of the major contributions of this paper. One such approach to determine personalized diagnoses for inconsistent requirements has been proposed for knowledge-based recommendation scenarios [Felfernig *et al.*, 2009] where repair proposals for inconsistent features requests are generated on the

[4] www.itu.dk/research/cla/externals/clib.

basis of the similarity between the feature requests and the items in a product table. [OSullivan *et al.*, 2007] introduce minimal exclusion sets. On the basis of such sets, [OSullivan *et al.*, 2007] discuss the concept of representative explanations which can be interpreted as sets of minimal diagnoses covering all constraints part of at least one of the existing diagnoses – this is an approach to take into account diversity in diagnosis recommendation and thus also an important aspect for our future work. In contrast to the work presented in this paper, the approach of [OSullivan *et al.*, 2007] does not explicitly take into account the preferences of the current user (customer). In knowledge-based recommendation maximally successful sub-queries [Godfrey, 1997; McSherry, 2004] represent the complement to minimal diagnoses [DeKleer *et al.*, 1992; Reiter, 1987]. Note that our work relies on the assumption of an *open configuration* (*tradeoff exploration*) based scenario where the user is free to specify preferred requirements – the configuration system then provides the corresponding feedback in terms of diagnoses.

## 6 Conclusions

We have introduced techniques that help to calculate *personalized diagnoses*. In this context we proposed different personalization strategies which can help to significantly increase prediction quality. Within the scope of an empirical study we compared five search strategies (cardinality-based, similarity-based, utility-based, probability-based, and ensemble-based). The results show clear advantages of personalized diagnosis calculation in terms of precision. Thus the results presented in this paper provide a solid basis for improving existing constraint-based applications in terms of a lower number of needed interaction cycles for the user and a lower number of needed diagnosis calculations.

## References


[Burke, 2000] R. Burke. Knowledge-based recommender systems. *Library and Inf. Systems*, 69(32):180–200, 2000.

[DeKleer *et al.*, 1992] J. DeKleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2–3):197–222, 1992.

[DeKleer, 1990] J. DeKleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45(3):381–391, 1990.

[Felfernig *et al.*, 2004] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner. Consistency-based diagnosis of configuration knowledge bases. *Artificial Intelligence*, 152(2):213–234, 2004.

[Felfernig *et al.*, 2007] A. Felfernig, G. Friedrich, and L. Schmidt-Thieme. Introduction to the ieee intelligent systems special issue: Recommender systems. *IEEE Intelligent Systems*, 22(3):18–21, 2007.

[Felfernig *et al.*, 2009] A. Felfernig, G. Friedrich, M. Schubert, M. Mandl, M. Mairitsch, and E. Teppan. Plausible repairs for inconsistent requirements. In *21st Intl. Joint Conference on Artificical Intelligence (IJCAI'09)*, pages 791–796, Pasadena, CA, 2009.

[Friedrich and Shchekotykhin, 2005] G. Friedrich and K. Shchekotykhin. A general diagnosis method for ontologies. In *4th Intl. Semantic Web Conference (ISWC05)*, number 3729 in Lecture Notes in Computer Science, pages 232–246, Galway, Ireland, 2005. Springer.

[Godfrey, 1997] P. Godfrey. Minimization in cooperative response to failing database queries. *Intl. Journal of Cooperative Information Systems*, 6(2):95–149, 1997.

[Junker, 1994] U. Junker. Preferring diagnoses using a partial order on assumptions. *Annals of Mathematics and Artificial Intelligence*, 11(1–4):169–185, 1994.

[Junker, 2004] U. Junker. Quickxplain: Preferred explanations and relaxations for over-constrained problems. In *19th National Conference on Artificial Intelligence (AAAI04)*, pages 167–172, San Jose, CA, 2004.

[Marques-Silva and Sakallah, 1996] J. Marques-Silva and K. Sakallah. Grasp: A new search algorithm for satisfiability. In *Intl. Conference on Computer-Aided Design*, pages 220–227, Santa Clara, CA, 1996.

[McSherry, 2004] D. McSherry. Maximally successful relaxations of unsuccessful queries. In *15th Conference on Artificial Intelligence and Cognitive Science*, pages 127–136, Galway, Ireland, 2004.

[OSullivan *et al.*, 2007] Barry OSullivan, A. Papdopoulos, B. Faltings, and P. Pu. Representative explanations for over-constrained problems. In *AAAI07*, pages 323–328, Vancouver, Canada, 2007.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 23(1):57–95, 1987.

[Russell and Norvig, 2003] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2003.

[Tsang, 1993] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

[Winterfeldt and Edwards, 1986] D. Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, 1986.