Extended Spectrum Based Plan Diagnosis for Plan-Repair

Shekhar Gupta , Bob Price and Johan de Kleer
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto CA 94304 USA
Email: {shekhar.gupta,bprice,dekleer}@parc.com

Nico Roos
Masstricht University
The Netherlands
Email: roos@maastrichtuniversity.nl

Cees Witteveen
Department of Software Technology
Delft University of Technology, The Netherlands
Email: c.witteveen@tudelft.nl

Abstract

In dynamic environments unexpected malfunctions or conditions can cause plan failure. Research has shown that plan-repair on failure can be more efficient than building complete conditional plans from scratch to handle all contingencies. The effectiveness of replanning depends on knowledge of exactly which plan actions failed and why. Conventional Model Based Diagnosis (MBD) can be used to detect such faulty components but the modeling cost (to generate the fault model) outweighs the benefits of MBD. In this paper, we propose an Extended Spectrum Based Diagnosis approach that efficiently pinpoints failed actions and does not require the fault models. Our approach first computes the likelihood of an action being faulty and subsequently proposes optimal probe locations to refine the diagnosis. We also exploit knowledge of plan steps that are instances of the same plan operator to optimize the selection of the most informative diagnostic probes. This reduces costs and improves the accuracy of diagnoses.

Introduction

Classical planning assumes that the world is deterministic so that every action produces the intended effects. However, this assumption is not true in real world planning domains where actions can fail because of unexpected events. Execution of a plan will lead to an unexpected goal state if one or more actions are behaving abnormally. When such incidents happen one possible way to achieve the desired goal state is repairing the original plan by adding/removing some actions (van der Krogt and de Weerdt, 2005). For example, consider a multimodal freight logistics system, where a planner such as TIMIPLAN generates optimal plans to deliver goods from one location to another (Flórez et al., 2011). TIMIPLAN has a plan monitoring component checking whether the execution of plans deviates from the expected outcome and triggers a replanning module if needed. To avoid replanning from scratch, the planner uses plan repair to change the initial plan as little as possible. For example, a damaged truck is replaced by a new truck and TIMIPLANS greedily selects such a truck with the least estimated total cost. This, however, assumes full observability of the health state of the trucks which in general might be too costly or in some cases infeasible.

Model based diagnosis is used to infer the set of faulty component(s) in a system from observations and background knowledge (Reiter, 1987). It exploits a descriptive behavioral model of components together with a structural model of how the components are connected to compute the implications of observations. The idea of MBD can be further extended to diagnose faulty actions in a plan where the plan can be seen as a system and the action can understood as a component MINI-MAX (Roos and Witteveen, 2009). This view enables the application of well known diagnosis techniques to plan descriptions. For instance, knowing that a road must be clear for a truck to pass, and observing that a truck has arrived from a distant city allows the system to infer that the road from that distant city is clear even though this cannot be directly observed. Methodology such as the pervasive diagnosis framework (Kuhn et al., 2010), has demonstrated how diagnosis can be performed on systems controlled by plans, but makes the simplifying assumption that the planning goal is a single output for the system and that any failed action has a direct observable effect on the output. It is therefore unsuitable for domains such as the logistics domain where many goals must be achieved simultaneously and action failures have local effects that are only indirectly related to the goals.

While powerful, model-based techniques require accurate fault models which are expensive to develop and in some cases the required data cannot be obtained at all. For example, it may be difficult to model all the ways in which a truck can fail to deliver a package to a destination. The proposed Spectrum Based Diagnosis (SBD) approach makes use of abstract frequency statistics to reveal possible causes of a problem without a fault model of the system. SBD has been successfully applied for software fault localization (Abreu et al., 2009) and hardware diagnosis (Arjan Van Gemund and Abreu, 2011). In our approach we use SBD to determine the health state of a plan step which infers the health state of corresponding action. In the planning domain, it is common for a single plan operator to be instantiated many times for different plan steps. For instance, a transport operation might be instantiated with the same truck to carry packages on several different routes in a plan. All plan steps that are instantiated from the same operator will fail if there is something wrong with the plan operator. For instance, every attempt to schedule a shipment on a blocked road will fail. In the online

replanning context, we are given the plan ahead of time, so we can exploit knowledge about the operator dependencies of actions within a plan. We propose Extended Spectrum Based Diagnosis which is able to exploit available information about such dependencies in the plan by elegantly extending the spectrum matrix. Finally, in domains, such as the logistics domain, we often have the ability to take information gathering actions. Perhaps we could get the dispatcher to call drivers and ask for a report on road conditions along a particular segment. However, each of these actions has costs involved. We address this, by combining our extended spectrum based diagnosis with an optimal probing strategy, which uses a mutual information criteria. Given the resulting information, standard replanting techniques are used to repair the plan. The result is a practical approach to planning for online systems with dynamic failures that works with incompletely described systems but exploits the known information to efficiently repair plans with the lowest cost. In the following sections we develop the mathematical framework for extended spectrum based diagnosis and demonstrate it on a notional multimodal transportation problem.

Preliminaries

Our planning formalism is modeled after the STRIPS planner (Fikes and Nilsson, 1971). Our specific notation is covered in following subsections.

State The world can be described by a finite set $Var = \{v_1, v_2, \ldots, v_n\}$ of variables and their respective *value domains* D_i. A particular state is denoted by an n-tuple $\sigma = (\sigma(v_1), \ldots, \sigma(v_n)) \in D_1 \times D_2 \times \cdots \times D_n$. In multimodal transportation system, the variables would represent the locations of individual items such as trucks and goods to be shipped.

Actions, plan operators and plan steps An *action* refers to an activity that results in some change of the (current) state of the world. A *plan operator* refers to a description of such an action in a plan. More exactly, a plan operator o is a function mapping state (σ_0) to another state (σ_1).

An instantiation of an operator o with specific arguments is called a plan step. It maps a specific state into another specific state. Therefore, given a set O of plan operators, we consider a set $S = inst(O)$ of instances of plan operators in O, called the set of plan steps. A plan step will be denoted by a small roman letter s_i. For example, a plan operator can be understood as a shipping action by a specific mode of transportation, i.e., a truck, a train or a ship. Such a shipping action can be used at several places in the plan using the same truck. Each specific occurrence of such a truck transportation is a plan step.

If plan step s is an instantiation of operator o, we say that $o(s) = o$. If for two plan steps s and s' it holds that $o(s) = o(s')$ they are said to be *related* to each other. In other words, s and s' are sharing same resource therefore there resource dependency between these two plan steps. For example, if the same truck (plan operator) is used to execute two different transportations (plan steps), these plan steps are related. Note that here plans differ from systems where normally components operate quite independently from each other. In plans, it seems rational to assume that a structural fault in the truck might affect at least a subset of its instantiations (plan steps).

If two plan steps are instantiated from the same operator, we say that they are related. Let $o(s)$ be the operator that step s is instantiated from. Given two plan steps s and s', if $o(s) = o(s')$ then they are related. For example, if the same truck (plan operator) is used to execute two different transportations (plan steps), these plan steps are related. Note that unlike typical physical systems in which components that make up the system fail independently, plans which contain related plan steps need to model the dependence in the failures between the related plan steps.

Plan and plan execution We represent our plans as a partially ordered set of steps. Formally, a plan is a tuple $P = \langle \mathcal{O}, S, < \rangle$ where $S \subseteq inst(\mathcal{O})$ is a set of plan steps occurring in $\mathcal{O}$ and $(S, <)$ is a partial order (Cox, Durfee, and Bartold, 2005). If step $s' < s$ then s' must be executed before s. Same $<$ relation can be used to denote the relative order between states. Figure 1 gives an illustration of a partially ordered plan.

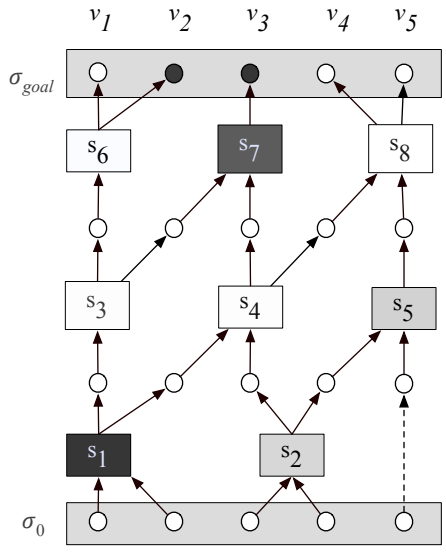

Figure 1: A partially ordered plan graph in which an initial state σ_0 is transformed by plan steps (s_i for $i = 1, 2, \ldots, 8$) into a goal state σ_{goal}. Each state characterizes the values of five variables v_1 , v_2 , v_3 , v_4 and v_5 . Plan steps having the same color (e.g. s_1 and s_7, and s_2 and s_5 are instantiations of the same plan operator.

Observations Our framework enables us to observe a set of values of the variables making up a state of the world. We denote an observation of variable v in state σ by σ_v. We assume that a cost is associated with any observation, except for observing the initial state (σ_0) and the goal state σ_{goal}. For ease of exposition, we assume all probes have equal cost.

Plan Diagnosis

We use conventional MBD notation to represent the plan we are diagnosing.

Definition 1 *A system is a pair (P, OBS) where P is a plan tuple $\langle \mathcal{O}, S, < \rangle$ and OBS is the set of values of the observed variables at the initial state σ_0 and the goal state σ_{goal}.*

Plan execution is validated by continuously monitoring the goal state. The difference in the observed value $\sigma_{goal(v)'}$ of any variable v in the goal state from the expected value $\sigma_{goal(v)}$ implies the plan execution failure, i.e., some plan steps are not executed in a correct way. For example, consider the plan shown in Figure 1. Suppose this plan represents a multimodal transportation plan where five goods $(v_1 \dots v_5)$ need to be delivered from initial location to goal location using different transportation modes $(s_1 \dots s_8)$. In the final destination, it is observed that two goods (v_2 and v_3) have not arrived which implies one or more plan steps are faulty.

Let $h_j \in \{ok, ab\}$ be the health state of plan step s_j where ok represents normal behavior and ab abnormal behavior. In establishing which part of the plan fails, we are only interested in those plan steps qualified as abnormal. Therefore, a plan diagnosis can be defined as following:

In establishing which part of the plan fails, we are only interested in those plan steps qualified as abnormal. Therefore, a plan diagnosis can be defined as following:

Definition 2 (Diagnosis) *A diagnosis P_D of a plan $P = \langle \mathcal{O}, S, < \rangle$ is a tuple $P_D = \langle \mathcal{O}, S, <, D \rangle$, where $D \subseteq \mathcal{S}$ is the subset of plan steps qualified as abnormal (and therefore, $S - D$ is the subset of plan steps qualified as ok).*

Spectrum Based Diagnosis

In absence of a detailed fault model of plan operators and plan steps, SBD is a suitable diagnosis methodology for the problem in hand. The basic principle of SBD can be described as follows: if the value of a variable in the goal state is incorrect, then one or more plan steps involved in generation of that variable are abnormal.

Obtaining the Spectrum Matrix The spectrum matrix shows for every variable in σ_{goal} which plan steps are involved from the state σ_0 to σ_{goal}. It records, in the goal state, whether a particular variable v_i has the expected value or not. Together with the information about involvement of plan steps, the resulting spectrum gives debuggers hints about the plan steps which are more likely related to failure, and hence have higher possibility to contain the faults.

The spectrum matrix (A, e), where $A = [a_{ij}]$ is the plan spectrum and e is the error vector can be constructed as follows: The plan spectrum A has N rows (one for each variable) and M columns (one for each plan step). We have $a_{ij} = 1$ if a plan step s_j is involved in the generation of variable v_i in σ_{goal}, else $a_{ij} = 0$. The vector e stores whether the outcome for variable v_i has the expected value $(e_i = +)$ or not $(e_i = -)$.

For example, suppose that in the plan presented in Figure 1, the value of variable v_2 and v_3 is not what we would expect in the goal state. Therefore $e_i = -$ for $i = 2$ and $i = 3$ and the following spectrum matrix can be obtained:

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	e
v_1	1	0	1	0	0	1	0	0	+
v_2	1	0	1	0	0	1	0	0	-
v_3	1	1	1	1	0	0	1	0	-
v_4	1	1	0	1	1	0	0	1	+
v_5	1	1	0	1	1	0	0	1	+

In any row with an unexpected outcome, at least on of the components used must be faulty. A minimal hitting set algorithm, STACCATO (Abreu et al., 2009), can be applied to the set of rows with unexpected outcomes to generate the set of diagnoses candidates (c_k) $\{c_1 =< s_1 >, c_2 =< s_3 >, c_3 =< s_2, s_6 >, c_4 =< s_4, s_6 >, c_5 =< s_7, s_6 >\}$.

The Spectrum Matrix for Plan Steps with Shared Resources The candidate $< s_2, s_6 >$ implies that the operator associated with s_2 may be faulty but it could be expensive or difficult to probe the output of s_2. From our knowledge of the plan, we know that s_2 is instantiated from the same operator as s_5. Therefore s_5 is also likely to fail, if s_2 fails. In this case, the failure of s_5 may have been intermittent or the failure may not have been relevant to the preconditions of the subsequent step s_8 so it did not have an effect on the final goal state σ_{goal}. This is called a masked fault and it is not picked up by standard SBD methods. This insight is important, because probing at s_5 may be easier and cheaper than probing at s_2. Imagine a scenario in which steps s_2 and s_5 use the same truck. Suppose in s_2, the truck is used at a distant location where it is difficult to inspect. If it is later used in a plan step s_5 at a location with inspection facilities it will be much easier to measure the health of this resource. There is one small complication. If an operator is used more than once in a plan, it could be heathy earlier in the plan and then fail at some later point.

To take these related plan steps into account, we modify the spectrum matrix in such a way that these relations are encoded in the matrix A itself. Suppose that the plan steps s and s' are related. If s is detected as faulty and $s < s'$, it seems reasonable to consider s' as faulty as well. Formally, we calculate the extended spectrum matrix $A' = [a'_{ij}]$ from A as follows:

$$a'_{ij} = \bigvee_{j' < j, o(j') = o(j)} a'_{ij'} \vee a_{ij} \quad (1)$$

In the plan depicted in Figure 1, plan steps with the same background are related. So s_1 and s_7 are related and s_2 and s_5 are related. The extended spectrum matrix would be (new entries appear in bold face):

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	e
v_1	1	0	1	0	0	1	**1**	0	+
v_2	1	0	1	0	0	1	**1**	0	-
v_3	1	1	1	1	**1**	0	1	0	-
v_4	1	1	0	1	1	0	**1**	1	+
v_5	1	1	0	1	1	0	**1**	1	+

Similar to other MBD engine our diagnosis engine assumes that plan steps are failing independently while computing posterior probability for every diagnosis. Therefore,

if we have a diagnosis in plan steps are related to each other our engine will compute incorrect posteriors. Hence diagnosis must not contain related plan steps. The extended matrix ensures that application of MHS algorithm on that matrix will produce diagnosis comprises of independent plan steps.

Application of minimal hitting set algorithm on extended matrix A' will generate diagnoses candidates (c_k) $\{c_1 =< s_1 >, c_2 =< s_3 >, c_3 =< s_7 >, c_4 =< s_2, s_6 >, c_4 =< s_4, s_6 >, c_5 =< s_7, s_6 >, c_6 =< s_5, s_6 >\}$.

Theorem 1 *Introducing related plan steps into the extended matrix ensures that the MHS algorithm will never return a diagnosis that includes two related plan steps.*

Proof Two plan steps will only appear together in a diagnosis if they individually explain distinct error observations. When we insert a pseudo observation for one of the steps into the matrix, the second step becomes an explanation for both error outputs and becomes a singleton diagnosis breaking up the joint diagnosis. Schematically,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Keeping related actions from appearing in the same diagnosis prevents us from multiplying these correlated failures together as if they were independent failures. This preserves the accuracy of the diagnosis. The diagnosis set for the extended matrix is $c_k = \{c_1 =< s_1 >, c_2 =< s_3 >, c_3 =< s_7 >, c_4 =< s_2, s_6 >, c_4 =< s_4, s_6 >, c_5 =< s_7, s_6 >, c_6 =< s_5, s_6 >\}$.

Probability Calculation Having corrected the spectrum matrix, we can use the BARINEL (Abreu et al., 2009) diagnostic engine to compute a fault probability for every diagnosis candidate using Bayes rule. For each variable observation $\sigma_{goal(v_i)}$, the posteriors are update according to the following rule for every candidate c.

$$Pr(c_k|\sigma_{goal}(v_i)) = \frac{Pr(\sigma_{goal}(v_i)|c_k) \cdot Pr(c_k|\sigma_{goal}(v_{i-1}))}{Pr(\sigma_{goal}(v_i))} \quad (2)$$

.

The recursion bottoms out with the prior for the candidate, $Pr(c_k)$, which is computed from the individual step priors assuming independent failures. Note that the candidate $< s_i >$ implies health variable $h_i = ab$. Generally:

$$Pr(c_k) = \prod_i \begin{cases} p_i & \text{if } h_i = 1 \\ 1 - p_i & \text{otherwise} \end{cases} \quad (3)$$

where p_i is the prior probability that plan step s_i is faulty.[1]

The BARINEL engine propagates failure probabilities along the plan step dependencies to calculate the probability $Pr(\sigma_1(v_i)|c_k)$ for each output variable i using maximum likelihood estimation (Abreu, 2009). The final posterior probability is computed by combining Equations 2, 3 and $Pr(\sigma_1(v_i)|c_k)$, and fault probabilities are assigned to plan step as shown in Table 1.

[1]In our case, the prior probability of every plan step is assumed to be to be 0.1.

s_i	$Pr(s_i)$	$Pr'(s_i)$	$I(X;Y)$	$I'(X;Y)$
s_1	0.200	0.160	0.512884	0.512884
s_2	0.002	0.002	0.008762	0.008762
s_3	0.800	0.762	0.016707	0.016707
s_4	0.002	0.002	0.264348	0.264348
s_5	0.000	0.002	**0.004198**	**0.011041**
s_6	0.007	0.008	0.000000	0.000000
s_7	**0.003**	**0.160**	0.000000	0.000000
s_8	0.000	0.000	0.074128	0.074128

Table 1: $Pr(s_i)$ and $I(X;Y)$ are derived for original matrix A. $Pr'(s_i)$ and $I'(X;Y)$ are derived for extended matrix A'

Probing Strategy

A major challenge for a diagnostician is to identify a suitable location for a new probe. In conventional MBD, mutual information criterion can be used to evaluate and compare measurement choice based on their information contribution (de Kleer and Williams, 1987), we have adapted this criterion to probing plan based systems with related steps. To illustrate the formulation, assume X is a diagnostic state of a plan and Y is the measure value of a variable at a probing location where X and Y are both random variables. The mutual information between X and Y is defined as:

$$I(X;Y) = \sum_{x,y} \left[p(x,y) \cdot \log \frac{p(x,y)}{p(x)p(y)} \right] \quad (4)$$

For example, suppose we derive mutual information about the value of location $l1$ and $l2$ as $I(X;Y_{l1})$ and $I(X;Y_{l2})$, respectively. In choosing between $l1$ and $l2$, we will choose $l1$ to probe if $I(X;Y_{l1}) > I(X;Y_{l2})$. As described in (Juan Liu and Zhou, 2008), the above expression can be estimated using entropy calculation, which is given as $I(X;Y) = H(Y) - H(Y|X)$, where $H(Y) = \sum_y \left[p(x) \cdot \log \frac{1}{p(x)} \right]$ is the entropy of Y and $H(Y|X) = \sum_{x,y} \left[p(y|x) \cdot \log \frac{1}{p(y|x)} \right]$ is the conditional entropy. For the plan example shown in Figure 1, observations are already given and fault probability has been computed from SBD, shown in Table 1. Estimated fault probabilities and observations in the goal state are used to compute $H(Y)$ and $H(Y|X)$ as described in (Juan Liu and Zhou, 2008). Mutual information for different probing location in our example (Figure 1) is summarized in Table 1.

Exploiting Related Plan Steps in Diagnosis

In the plan described in Figure 1, s_3 has the strongest participation in the unexpected goal state outcomes for variables, v_2 and v_3. In the first column of Table 1, $Pr(s_i)$, we see that the diagnoser assigns s_3 the highest probability of failure. The standard spectrum A assigns different probabilities to plan steps s_1 and s_7. The extended spectrum, which recognizes that s_1 and s_7 are related, increased the fault probability of s_7 and now s_7 and s_1 have equal probability. Similar conclusions can be made for other related plan steps s_2 and s_5.

The mutual information results shown in Table 1 provides us some interesting conclusions. Without any ambiguity both the spectrum matrices suggest that s_1 is the most informative location to probe and that s_7 is the least. Therefore, probing at the output of s_1 is going to improve the diagnosis by the maximum amount. Since s_7 is in the goal state (no cost) of the plan therefore no extra information can be gained which matches our mutual information computation. At the same time, extending the matrix reveals the information content at the output of plan step s_5 to the diagnoser. In this case, s_5 is closer to the middle of the plan than s_2 which means that it better splits the hypothesis space about possible causes of failure and therefore is more informative. In some cases, s_5 may not be more informative, but may be cheaper or easier to measure. In any case, the extended spectrum matrix opens up new options to increase the accuracy and decrease the cost of diagnosis in plans with related plan steps.

Conclusion

Continuous planning in online dynamic real world environments requires accurate diagnosis to pinpoint which plan steps need to be repaired. Spectrum based diagnosis approaches are a natural approach as they do not require explicit fault models to provide useful diagnostic information. We have seen that extended spectrum based diagnosis extends the advantage of traditional spectrum based diagnosis to systems controlled by a plan which can have related plan steps. The extended spectrum matrix also increases the options for probing potentially leading to more accurate and cheaper diagnosis. The technique can be easily extended in many ways such as computing explicit expected probe costs and considering other ways in which operators can be related. Extended spectrum based diagnosis therefore represents an important technology option for robust, practical and efficient plan based control of real world systems.

References

Abreu, R.; Zoeteweij, P.; Golsteijn, R.; and van Gemund, A. 2009. A practical evaluation of spectrum-based fault localization. *Journal of Systems and Software*.

Arjan Van Gemund, S. G., and Abreu, R. 2011. The antares approach to automatic system diagnosis. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-2011)*, 5–12.

Cox, J. S.; Durfee, E. H.; and Bartold, T. 2005. A distributed framework for solving the multiagent plan coordination problem. In *In AAMAS*, 821–827. ACM Press.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artif. Intell.* 32(1):97–130.

Fikes, R., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. In *IJCAI*, 608–620.

Flórez, J. E.; de Reyna, Á. T. A.; García, J.; López, C. L.; Olaya, A. G.; and Borrajo, D. 2011. Planning multi-modal transportation problems. In *ICAPS*.

Juan Liu, Johan de Kleer, L. K., and Zhou, R. 2008. A unified information criterion for evaluating probe and test selection. In *PHM-2008*.

Kuhn, L.; Price, B.; Do, M. B.; Liu, J.; Zhou, R.; Schmidt, T.; and de Kleer, J. 2010. Pervasive diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 40(5):932–944.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.

Roos, N., and Witteveen, C. 2009. Models and methods for plan diagnosis. *Journal of Autonomous Agents and Multi-Agent Systems* 19(1):30–52.

van der Krogt, R., and de Weerdt, M. 2005. Plan repair as an extension of planning. In *ICAPS*, 161–170.