Automatic Generation of Optimal and Suboptimal Diagnosis Tree from a Cross-Table

Pierre-Philippe FAURE, Louise TRAVE-MASSUYES and Hervé POULARD

***Abstract*—This paper deals with an automatic diagnosis tree generation application based on the AO* algorithm. The inputs of this application are the faults which may occur on the system to diagnose with their respective occurrence probability, the tests that can be realized on it and the cross-table which affects to each (fault/test) pair the set of modalities which are expected as result of the test when the fault occurs. The main drawback of this application, is that it often spends too much computation time to obtain an optimal diagnosis tree. Then, two methods which allow to obtain a suboptimal diagnosis tree rather than an optimal one are proposed in order to reduce this too important computation time.**

I. Introduction

In the automotive manufacture domain, the use of ECUs (Electronic Control Units) to control several functions (such as engine injection, air conditioning or ABS) has been widely developed during these last years. To control the concerned function, many electronic circuits constituted by sensors and switches are plugged on these ECUs. These ECUs are equiped with an auto-diagnosis function which allows to detect with security which of these plugged on electronic circuits are failing. However, knowing the failed electronic circuit, the ECU is not able to localize the faulty replaceable component. In order to diagnose such electronic circuit, diagnosis trees are built. These trees allow the garage mechanic to find the faulty replaceable component(s) by performing a sequence of tests (measurements) which has the lowest global cost as possible. Nowadays these diagnosis trees are hand made by human experts. This task requires more and more time and becomes more and more difficult as the complexity of electric circuits and mecatronic systems increases. Consequently, errors are not unusual in the resulting diagnosis trees. As a matter of fact, it becomes urgent to reduce the human intervention in the diagnosis tree generation process at the lowest.

Especially dedicated to resistive network circuits supplied by one unique voltage source, the off-line automatic diagnosis tree generation application proposed by Faure (see [1]) may be divided into the two following steps:

1. From the structural model of the system to diagnose and the intrinsic behavioral models of each of its components, building a prediction table which crosses the set of the possible single faults that may occur in the system with the set of the possible tests that may be realized on it.
2. From the prediction table obtained from the previous step, generating the optimal diagnosis tree according to the Pattipati's method (see [2]) extended to non exclusive multi-modality tests (see [1]).

This paper deals with the second step of this application and proves the optimality of the proposed Pattipati's method extension for the diagnosis tree generation problem with multi-modality non exclusive tests.

Moreover, two methods which allow to obtain suboptimal diagnosis trees rather than optimal ones in order to reduce the application processing time are proposed.

The paper is then organized as follows.

Section 2 sets the optimal diagnosis tree generation problem with the main following concepts : fault, test, cross-table, diagnosis tree and discriminating tests subset.

Section 3 briefly presents the initial Pattipati's method to generate automatically optimal diagnosis trees with exclusive binary tests.

Section 4 details the extension of this method to non exclusive multi-modality tests.

Section 5 defines the optimal discriminating tests subset which allows to obtain the optimal diagnosis tree by the same application more quickly than with the initial tests set.

Section 6 then proposes two suboptimal discriminating tests subsets which allow to obtain suboptimal diagnosis trees by the same application more quickly than with the initial tests set.

Section 7 studies quality criterion of a discriminating tests subset which includes both the quality of the obtain diagnosis tree and the application processing time.

Manuscript received April 18, 2001.

P. P. Faure is PhD student at ACTIA 25, chemin de Pouvourville 31432 TOULOUSE CEDEX 04 (FRANCE), faure@actia.fr and at LAAS-CNRS 7, avenue du colonel Roche 31077 TOULOUSE CEDEX 04 (FRANCE), faure@laas.fr

L. Travé-Massuyès is research manager at LAAS-CNRS, louise@laas.fr

H. Poulard is research ingeneer at ACTIA, poulard@actia.fr

Finally, section 8 concludes on the contributions of this paper.

II. Diagnosis problem

A. Faults

Let E be the set of the n_E elementary components e_k with $k \in \{1, ..., n_E\}$ which constitute the system to diagnose. Let n_{e_k} be the number of abnormal behavior $AB_l(e_k)$ with $l \in \{1, ..., n_{e_k}\}$ of the elementary component e_k and $\neg AB(e_k)$ its normal behavior.

A fault is a n_E vector which associates to each of the n_E elementary component e_k one of its n_{e_k} abnormal behavior or its normal one.

For such a system, $\prod_{k=1}^{n_E}(n_{e_k}+1)$ faults may occur.

To be useful, a diagnosis application has to discriminate faults which have the highest probabilities. The aim of this subsection is to reduce the exhaustive set possible faults to the subset of faults which have the highest probability to occur. In practice, the occurrence probability is obtained from the Mean Time To Failure (MTTF) given by the designer (see [3] and [4])

For the following, let F be the set of the n_F most probable faults f_i with $i \in \{1, ..., n_F\}$ to discriminate and p_i, their respective a priori occurrence probability.

B. Tests

A test may be a physical variable (such as potential point, intensity or equivalent resistance) measurement or an observable (such as "look at the brightness of a light" or "listen to a particular noise") manifestation to realize. For each test, several results, called modalities (which may be as well exact values, value intervals, or qualitative modalities), are possible. The association of a test and one of its modalities is an observation.

Definition 1 (Exclusive test) *A test is said to be exclusive if, for any fault, one and only one modality is expected for the concerned test.*

Definition 2 (Multi-modality test) *A test is said to be multi-modality if it has a number of modalities, that is to say possible value (i.e. observations), greater or equal than two.*

Definition 3 (Binary test) *A test is said to be binary if it has a number of modalities, that is to say possible value (i.e. observations), exactly equal to two.*

Definition 4 (Test cost) *Each test has a cost which represents the difficulty (measurement tool configuration, system structure modifications required by the measurement, measurement points accessibility...) to realize it.*

Definition 5 (Unit test cost assumption) *Under unit test cost assumption, the test cost is equal to* 1 *for any considered test.*

The role of a test is to reduce the current set of faults which may be responsible of the system failure. Consequently, the diagnosis problem is equivalent to select the test which ensures the best discrimination between the faults of this current faults set.

For the following, let S be the set of the n_S tests s_j with $j \in \{1, ..., n_S\}$ to discriminate, n_M^j their respective number of possible modalities m_k^j with $k \in \{0, ..., n_M^j - 1\}$ and c_j, their respective realization cost.

C. Cross-table

Given F and S defined as previously, the corresponding cross-table has n_F lines and n_S columns. Each of its cells $C(i,j)$ contains the set of $n_M^{i,j}$ modalities among the n_M^j possible ones which are expected as result of the s_j test in occurrence of the f_i fault.

Moreover, the conditional probabilities $P(s_j = m_k^j | f_i)$ of having "s_j gives modality m_k^j" knowing that "system is in f_i fault" for any $k \in \{0, ..., n_M^j - 1\}$ are also available in the $C(i,j)$ cell. These conditional probabilities are normalized as shown on equation 1. For any modality m_k^j which does not belong to the $C(i,j)$ modalities set then $P(s_j = m_k^j | f_i) = 0$.

$$\sum_{k=1}^{n_M^j} P(s_j = m_k^j | f_i) = 1 \tag{1}$$

For the following, C represents the cross-table corresponding to the F and S sets and $C(i,j)$, the cell of C relative to the fault f_i and the test s_j.

D. Diagnosis tree

Formally, a diagnosis tree may be viewed as an AND/OR tree (see [2] and [5]). Actually, it is composed of two kinds of nodes : the OR nodes which correspond to the current set of the possible faults and the AND nodes which correspond to one test and its different resulting modalities.

Of course, the root node is an OR node composed of all the possible assumptions (faults) for the studied system. One leaf node is also one OR node represents at most one possible assumption. A not leaf OR node has one and only one AND node child corresponding to the test to apply whereas an AND node has several OR node children corresponding to the several possible modalities of the relative test. The figure 1 gives an AND/OR tree representing a diagnosis tree of 4 faults $\{f_1, ..., f_4\}$ using 5 tests $\{s_1, ..., s_5\}$.

For the following, let T be a diagnosis tree which discriminates the faults of the F set with tests of the S set according to the cross-table C. Let also n_L be the number of leaves $\{l_1, ..., l_{n_L}\}$ and $P(l_i)$ the occurrence probability of each l_i leaf such that $\sum_{i=1}^{n_L} P(l_i) = 1$. At last, let d_{ij} be a boolean variable equal to 1 if the s_j test belongs to the path from the root to the l_i leaf and 0 otherwise.

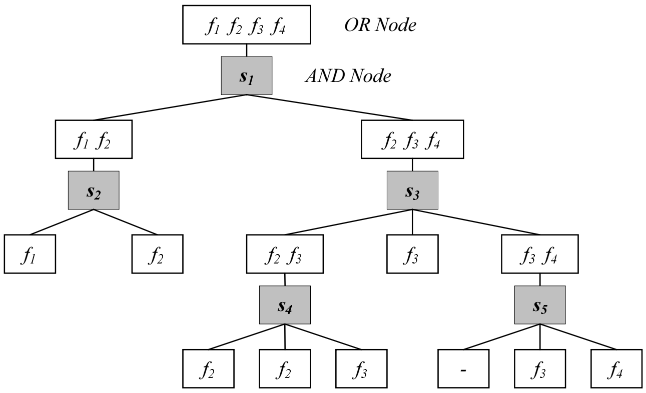

Fig. 1. Diagnosis tree

The objective function K of a diagnosis tree T, defined by equation 2, is proposed to evaluate the different possible diagnosis trees of a same system and, then, select the optimal one.

$$K(T) = \sum_{i=1}^{n_L} P(l_i) \times \left(\sum_{j=1}^{n_S} d_{ij} \times c_j \right) \qquad (2)$$

Under unit test cost assumption, the objective function $K(T)$ is equivalent to the mean depth of the tree T.

E. Discriminating tests subset

A tests subset is said to be discriminating for a set of faults if it is able to discriminate all the faults of the set.

Definition 6 (Minimal Discriminating tests subset) *A discriminating tests subset S' is said to be minimal if and only if, for any of its n'_S tests s'_j with $j \in \{1, ..., n'_S\}$, $S' - \{s'_j\}$ is not able do discriminate all the faults of the set.*

Definition 7 (Optimal Discriminating tests subset) *An optimal discriminating tests subset is a discriminating test subset composed of the tests effectively used by a given optimal diagnosis tree for the considered faults set.*

Moreover, it is important to underline that, for a same initial tests set, it exists as many optimal discriminating tests subsets as optimal diagnosis trees using different tests subsets.

While the tests set S is not a discriminating set for the faults set F, the corresponding cross-table C does not have a sufficient discrimination power to build a diagnosis tree.

It is then necessary to recursively add tests to S or remove faults from F and update the corresponding cross-table C in order to study if the updated S set is a discriminating for the updated F set.

III. Pattipati's Method

A. Presentation

From the faults set F, the tests set S and the corresponding cross-table C, the Pattipati's method allows to find the optimal diagnosis tree by using an AND/OR tree exploration algorithm (i.e. AO* algorithm).

In Pattipati's studies (see [2] and [5]), only exclusive binary tests are considered. First of all, it is important to remark that a diagnosis tree using only exclusive tests proposes one and only one leaf for each of the faults to discriminate then $n_L = n_F$.

Then, in accordance with the optimality general criterion 2, the objective function $K(T)$ may be written as shown on equation 3.

$$K(T) = \sum_{i=1}^{n_F} p_i \times \left(\sum_{j=1}^{n_S} d_{ij} \times c_j \right) \qquad (3)$$

B. AO* algorithm

An AO* algorithm is based on an AND/OR search tree. An AND/OR search tree contains AND nodes which have the property to be true if all their children are true and OR nodes which have the property to be true if at least one of their children is true. As already said, a diagnosis tree can be viewed as a succession of OR nodes and AND nodes. An OR node represent the set of the current possible faults and have one AND node child corresponding to the test to apply whereas an AND node represent one test and have several OR node children corresponding to the different modalities of this test.

The explicit AND/OR search tree represents all the possible solutions of a problem starting from the ground elements of this problem. In the optimal diagnosis tree problem, ground elements are the faults set F, the tests set S and the cross-table C. Obviously, the possible solutions are all the diagnosis tree T which allow to discriminate all the faults of F using of any subset of S.

Because of this generally too high complexity, the implicit AND/OR search tree is rarely totally expressed. Then, the idea of the AO* algorithm is to develop only parts of the implicit AND/OR search which correspond to the most interesting solutions of the problem, according to the objective function to optimize. This sub-tree of the implicit AND/OR search tree is selected according to the most pertinent as possible heuristic evaluations and called the explicit tree. The more the heuristic evaluations are pertinent, the more the explicit tree does not grow too much and keeps only the most interesting solutions. The optimal solution of the problem can be find inside the explicit AND/OR search tree. Consequently, the optimal diagnosis tree T^* is a selected sub-tree of the explicit AND/OR search tree developed during the AO* algorithm.

Given an OR node $N$of the explicit tree, $H(N)$ represents an admissible heuristic estimated value (see later for more) of $K(T_N^*)$ where T_N^* is the optimal sub-tree having N as OR node root whereas $F(N)$ represents the current estimated value of $K(T_N^*)$. If N is a leaf of the explicit tree then $F(N) = H(N)$.

1. *Initialization*
   At the beginning of the AO* algorithm, the explicit AND/OR search tree is composed of only its root, called N_r, which is an OR node representing all the faults of F. The marked current optimal diagnosis tree T^* is also composed of only this OR node. L^* is the set of the expandable OR node leaves of the current T^* tree, that is to say, leaves which do not correspond to isolated fault (i.e. real leaves of the definitive diagnosis tree).

   The $F(N_r)$ value is initialized to the heuristic value $H(N_r)$ computed for this OR node. Moreover, the probability of an OR node being the sum of the faults mentioned in this OR node, the probability of the N_r node $P(N_r)$ is the sum of the occurrence probabilities corresponding to all the faults to discriminate, so, equal to 1.

2. *Iterative treatment*
   At each step of the AO* algorithm, one of the OR node leaves of the selected current optimal diagnosis tree T^*, called N_c, is developed. This OR node is developed by creating all its n_A AND node children, called N_j^A with $j \in \{1, ..., n_A\}$, relative to all the available tests. Each of these just created AND node N_j^A is itself developed by creating their n_O^j OR node children, called $N_{j,k}^O$ with $k \in \{1, ..., n_O^j\}$, relative to the n_O^j different modalities of the corresponding test.

   For each created OR node $N_{j,k}^0$ leave, $F(N_{j,k}^0)$ is computed as $H(N_{j,k}^0)$. Then, for each created AND node N_j^A leave, $F(N_j^A)$ is computed as $Test\_Cost(N_j^A) + \sum_{k=1}^{n_O^j} F(N_{j,k}^0)$.

   At last, a recursive treatment is then applied on the successive nodes met on the path from the current OR node N_c to the root N_r. This treatment consists in updating successively the F values of these nodes and the marks of the selected AND nodes which constitute the current optimal diagnosis tree T^*.

   If the current node to update N_c is an OR node, the mark on its selected AND node child, if any, is removed. $F(N_c)$ is then computed as $\min_{j=1}^{Nb\_Child(N_c)} F(j^{th}\_Child(N_c))$ and the AND node child for which this minimum is reached is marked.

   If the current node to update N_C is an AND node, $F(N_c)$ is then computed as $Test\_Cost(N_c) + \sum_{k=1}^{Nb\_Child(N_c)} P(k^{th}\_Child(N_c)) \times F(k^{th}\_Child(N_c))$.

   While the current node to update N_c is different from the root N_r, the current node to update becomes the father of N_c.

3. *Stop condition*
   The AO* algorithm stops when the set L^* of the expandable OR node leaves of the current optimal diagnosis tree T^* is empty.

   Then the marked current optimal diagnosis tree T^* is the definitive optimal diagnosis tree and $K(T^*) = F(N_r)$.

C. Admissible Heuristic

As seen before, the AO* algorithm uses a heuristic evaluation function, more simply called heuristic, H to orient the search. For a given OR node, a heuristic is a function which gives a more or less rough evaluation of the objective function K value of the optimal diagnosis sub-tree having this OR node as root.

Then, let N be the studied OR node, and T_N^*, the optimal diagnosis sub-tree having N as root. Let also $H^*(N)$ be the exact evaluation of $K(T_N^*)$ and $H(N)$, an estimated evaluation $K(T_N^*)$. It has been proven that if, for each OR node N, $H(N) \leq H^*(N)$ (property of admissibility, see [6]), then the algorithm converges to the optimal diagnosis tree T^*.

Moreover, the convergence rate is directly related to the quality of the heuristic H. The closer $H(N)$ is from $H^*(N)$ for any OR node N, the lowest is the number of useless OR nodes (i.e. OR nodes which should not appear in the definitive optimal tree T^*) expanded during the algorithm. This can be easily illustrated by considering the two extreme cases : without heuristic and with a heuristic H such as $H(N) = H^*(N)$ for any OR node N.

- *Without heuristic*
  Without heuristic, explicit tree has to be completely explored in order to find the optimal diagnosis tree T^*.

- *With a heuristic H such as $H(N) = H^*(N)$ for any OR node N*
  With such a heuristic, the optimal sub-tree T^* is obtained immediately without expanding any useless node.

D. Huffman algorithm

Property 1 (Node Ordering Condition) *In case of exclusive tests, under unit cost test assumption, if a diagnosis tree T verifies the node ordering condition, that is to say : each node of the a^{th} level has an a priori occurrence probability greater than the one of any node appearing on the b^{th} level such that $d \geq b \geq a \geq 0$, where d represents the diagnosis tree depth, then, T is an optimal diagnosis tree according to the objective function K.*

Proof 1 (Node Ordering Condition) *As seen previously, the objective function K, with exclusive tests under unit cost assumption, is expressed as shown on equation 3.*

Let f_i with $i \in \{1, ..., n_F\}$ be one of the n_F faults to discriminate and p_i their respective occurrence probabilities. Moreover, let f_k and f_l with $k \neq l$ and $k, l \in \{1, ..., n_F\}$ be two particular distinct faults such that $p_k > p_l$. Let s_j with $j \in \{1, ..., n_S\}$ be one of the n_S binary exclusive available tests and $c_j = 1$ its unit cost, d_{ij} is a boolean variable equal to 1 if the s_j test belongs to the path from the root to the l_i leaf and 0 otherwise. Then let C be the cross-table of the fault f_i set with the test s_j set.

Now, let T_1 and T_2 be two almost identical diagnosis trees allowing to discriminate the n_F faults with a sub-set of the n_S tests according to the cross-table C. Let a and b be two different levels of the diagnosis trees T_1 and T_2 such that $d \geq b > a \geq 0$, where d represents the depth of both trees.

For the diagnosis tree T_1, f_l is placed on level a and f_k on level b and its objective function $K(T_1)$ can be written as shown on equation 4.

$$K(T_1) = (p_k \times b) + (p_l \times a) + \sum_{i=1, i \neq \{k,l\}}^{n_F} p_i \times \left(\sum_{j=1}^{n_S} d_{ij} \right) \quad (4)$$

For the diagnosis tree T_2, f_k is placed on level a and f_l on level b and its objective function $K(T_2)$ can be written as shown on equation 5.

$$K(T_2) = (p_l \times b) + (p_k \times a) + \sum_{i=1, i \neq \{k,l\}}^{n_F} p_i \times \left(\sum_{j=1}^{n_S} d_{ij} \right) \quad (5)$$

Knowing that $b > a$, expressions 4 and 5 become respectively expressions 6 and 7.

$$((p_k + p_l) \times a) + (p_k \times (b - a)) + \sum_{i=1, i \neq \{k,l\}}^{n_F} p_i \times \left(\sum_{j=1}^{n_S} d_{ij} \right) \quad (6)$$

$$((p_k + p_l) \times a) + (p_l \times (b - a)) + \sum_{i=1, i \neq \{k,l\}}^{n_F} p_i \times \left(\sum_{j=1}^{n_S} d_{ij} \right) \quad (7)$$

As $p_k > p_l$, $K(T_1) \geq K(T_2)$. So, the application of one permutation according to the node ordering condition reduces the objective function value. Moreover, once all the nodes of the current tree are sorted by level, this tree verifies the node ordering condition and no more permutation, according to the node ordering condition, may be applied. Consequently, the objective function value of this diagnosis tree can not be reduced and is minimal. Hence, this obtained diagnosis tree is the optimal one T^*.

$i \leftarrow n_F$;
Create n_F leaves corresponding to the n_F faults;
While($i > 1$)**do**
  Order the i faults by their decreasing probability;
  $p_{i-1} \leftarrow p_{i-1} + p_i$ (i.e. the two least probabilities);
  Create new node f_{i-1} father of f_i and old f_{i-1} nodes;
  Remove f_i fault and p_i probability from the faults set;
  $i \leftarrow i - 1$;
End While
Return node f_1 root of the created optimal tree;

Fig. 2. Huffman algorithm

Theorem 1 (Huffman) *In case of exclusive and binary tests, under unit cost test assumption, the Huffman algorithm creates an optimal complete binary diagnosis tree according to the objective function K.*

Proof 2 (Huffman) *From the previous property, it can be deduced that the two least occurrence probability faults of the considered faults set are always placed on the deepest level of the optimal diagnosis tree corresponding to this faults set. Actually, at each iteration of the Huffman algorithm, the two least occurrence probability faults are found, these two faults are removed from the studied faults set and a new fault is created which has, as occurrence probability, the sum of these two least occurrence probabilities. This new virtual fault is, in the complete binary tree, the father node of these two precedent least occurrence probability faults. So, at each iteration of the Huffman algorithm, a new faults set is considered and the two least occurrence probability faults of this set are affected to the two deepest leaves of the optimal diagnosis tree corresponding to this set. Consequently, by applying recursively the previous reasoning and treatment, node ordering condition is always verified and, according to the previous property, optimality is ensured. Moreover, an optimal binary diagnosis tree T^* obtained by Huffman algorithm is always complete (i.e. T^* does not contain empty leaf) by construction.*

E. Pattipati's Heuristic

The Huffman algorithm builds an optimal diagnosis tree composed of a sub-set of exclusive binary tests s_j, with their respective test cost $c_j = 1$, among the n_S available ones.

This obtained optimal diagnosis tree is not optimal anymore if the unit test cost assumption is removed. So, in this case, an AO* algorithm has to be used in order to find the optimal diagnosis tree.

The Pattipati heuristic, called H_p, is based on the diagnosis tree obtained from the Huffman algorithm and takes into account the different test costs such that it is always lower than the exact evaluation H^* in order to be admissible.

Let N be any OR node and $K^* = K(T^*)$ the optimal objective function value of the optimal diagnosis tree T^* obtained by the Huffman algorithm for the set of faults which are contained in N under unit test cost assumption. The n_S available tests are ordered by increasing costs such as $0 < c_1 \leq ... \leq c_{n_S}$. The admissible heuristic H_p proposed by Pattipati is then expressed as shown on equation 8 where $K^{'}$ is the integer part of K^*.

$$H_p(N) = \sum_{j=1}^{K^{'}} c_j + \left([K^* - K^{'}] \times c_{K^{'}+1}\right) \quad (8)$$

This heuristic is equivalent to build a Huffman tree with the tests which have the lowest costs. Consequently, this heuristic verifies the admissibility property, and hence, leads to the optimal tree by using the AO* algorithm.

F. Faults occurrence probabilities propagation

For a given OR node N_c of the AND/OR search tree, let $P_c(f_i)$ with be the occurrence probability of the n_F faults of the F set at the OR node N_c.

To compute the occurrence probability $P_{j,k}(f_i)$ of the f_i fault on the OR node $N^O_{j,k}$ associated to the m^j_k modality result of the realization of the test s_j at the node N_c, one have to use the equation 9.

$$P_{j,k}(f_i) = P(s_j = m^j_k | f_i) \times P_c(f_i) \quad (9)$$

Any $P(s_j = m^j_k|f_i)$ value with $k \in \{0, ..., n^j_M - 1\}$, $i \in \{1, ..., n_F\}$ and $j \in \{1, ..., n_S\}$ can be found as the computed probability corresponding to the m^j_k modality in the cell $C(i, j)$ of the cross-table C corresponding to the f_i fault and s_j test.

Consequently, for each created OR node during the AO* algorithm, the occurrence probabilities corresponding to the possible faults mentioned in this OR node may be computed according to the equation 9.

IV. Adapted heuristic

A. Introduction

The previous section shows that an AO* algorithm with the Pattipati's admissible heuristic H_p is able to find an optimal diagnosis tree for exclusive binary tests having different test costs.

In this work, it may happen that a same fault occurs in more than only one modality proposed by a test. This test is called non exclusive test. It may also happen that a test has more than two modalities. This test is called multi-modality test.

This section explains how these two test characteristics are taken into account in the Pattipati's admissible heuristic H_p in order to preserve its admissibility.

B. Non Exclusivity Assumption

As already seen, in case of non-exclusive tests, a same fault may appear in several leaves of a diagnosis tree.

Let n^i_L be the number of l^i_h leaves with $h \in \{1, ..., n^i_L\}$ containing the f_i fault. Let p^i_h with $h \in \{1, ..., n^i_L\}$ be their respective occurrence probabilities such that $\sum_{h=1}^{n^i_L} p^i_h = 1$. Then, the occurrence probability of the f_i fault, at the l^i_h leaf, called $P_{l^i_h}(f_i)$, may be written as shown on equation 10.

$$\begin{cases} P_{l^i_h}(f_i) = p_i \times p^i_h \\ \displaystyle\sum_{i=1}^{n_F} \sum_{h=1}^{n^i_L} P_{l^i_h}(f_i) = \sum_{i=1}^{n_F} \left(p_i \times \sum_{h=1}^{n^i_L} p^i_h \right) = 1 \end{cases} \quad (10)$$

So, the initial expression of the objective function K shown on equation 2 can then be changed into equation 11 where $d_{(i,h),j}$ is a boolean variable equal to 1 if the s_j test belongs to the path from the root to the l^i_h leaf and 0 otherwise.

$$K(T) = \sum_{i=1}^{n_F} \sum_{h=1}^{n^i_L} P_{l^i_h}(f_i) \times \left(\sum_{j=1}^{n_S} d_{(i,h),j} \times c_j \right) \quad (11)$$

Moreover, it seems obvious that the non exclusive test assumption makes the discrimination of a given faults set with the same tests set more difficult. That is, the optimal diagnosis tree obtained by the Huffman algorithm with the non exclusive test assumption is deeper than the one with the exclusive test assumption.

Consequently, assuming a priori the test exclusivity assumption in the Pattipati's admissible heuristic H_p does not affect its admissibility (see notion of asymmetrical tests in [7] and [8]).

However, it is important to underline that, if some of the tests are effectively non-exclusive, this decreases the quality of the heuristic (i.e. the distance between H_p and H^* increases).

C. Multi-Modality Assumption

At the contrary, the test multi-modality assumption makes the discrimination of a same faults set with a same tests set easier than the binary test assumption. That is, optimal diagnosis tree obtained by Huffman algorithm with the test binary assumption is deeper than the one with the multi-modality test assumption.

Consequently, considering in priority tests which have the biggest modality cardinality in the equivalent Huffman tree proposed by the admissible heuristic H_p minimizes the depth of this tree and, so, allows to keep its admissibility property of H_p.

Let T^*_b be the optimal diagnosis tree obtained by Huffman algorithm for the considered faults set and exclusive binary tests under unit test cost assumption. The multi-modality extension algorithm modifies T_b^* in order to obtain the optimal diagnosis tree T_m^* for the considered faults set and exclusive multi-modality tests under unit test cost assumption.

Order the n_S test modality cardinalities
such as $(M_1 \geq ... \geq M_{n_S} \geq 2)$;
$i \leftarrow 1$;
$Finish \leftarrow false$;
While$((Finish = false)\text{AND}(M_i > 2))$**do**
  $Finish \leftarrow true$;
  For all non-leave nodes of level $(i-1)$ **do**
    $Card \leftarrow 2$;
    While $((Card < M_i)$
    AND($\exists$ at least one non leaf child)) **do**
      break the biggest child;
      $Card \leftarrow Card + 1$;
      reorder the children of the studied node;
    End While
    If $(Card = M_i)$ **then** $Finish \leftarrow false$;
  End For
  $i \leftarrow i + 1$;
End While

Fig. 3. Multi-modality extension algorithm

Theorem 2 (Multi-Modality Extension) *From the optimal diagnosis binary tree obtained by Huffman algorithm T_h^*, the multi-modality extension algorithm builds the corresponding optimal diagnosis tree T_m^* taking into account the different modality cardinalities of the available tests.*

Proof 3 (Multi-Modality Extension) *As previously seen, the objective function K, with exclusive and binary tests under unit cost assumption, is expressed as shown on equation 3. Now, considering that some available tests may also have more than 2 modalities, this expression of objective function K does not change.*

First of all, it has to be proven that, at each step of the multi-modality algorithm, the current tree T_m^* keeps always optimal according to the objective function K for the current considered modality cardinality M_i.

To reach this optimality, the current tree T_m^* has to verify the node ordering condition (according to the property 1). Let A be the studied node during the multi-modality algorithm and T_0 the three first levels of the binary sub-tree for which A is the root. The general case is considered, where these three levels are assumed to be full (the maximum number of nodes that can be placed on the three first levels of a binary tree is 7) as shown on figure 4.

The relations between these seven nodes named $\{A, B, C, D, E, F, G\}$ whose respective probabilities are $\{p_A, p_B, p_C, p_D, p_E, p_F, p_G\}$, involved by the construction of the tree according to the Huffman algorithm, are expressed on equations system 12 and consistent with the node ordering condition (according to property 1 and theorem 1).

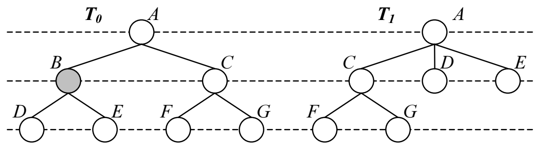

Fig. 4. Central treatment of the multi-modality extension algorithm

$$\begin{cases} p_A = p_B + p_C \\ p_B = p_D + p_E \\ p_C = p_F + p_G \\ p_B \geq p_C \\ p_D \geq p_E \\ p_F \geq p_G \\ p_C \geq p_D \\ p_E \geq p_F \end{cases} \tag{12}$$

The tree T_1 (see figure 4) is obtained by applying the central treatment (i.e. "break the biggest child;" and "reorder the children of the studied node;") on node A of the tree T_0. The relations between the nodes of the resulting tree T_1 are expressed on equations system 13 and keep also consistent with the node ordering condition.

$$\begin{cases} p_A = p_C + p_D + p_E \\ p_C = p_F + p_G \\ p_C \geq p_D \\ p_D \geq p_E \\ p_F \geq p_G \\ p_E \geq p_F \end{cases} \tag{13}$$

Consequently, the multi-modality extension algorithm applied on a binary diagnosis tree obtained by Huffman algorithm verifies this node ordering condition at each of its steps and, so, ensures optimality of the objective function K.

Then, it is important to prove that affecting the i^{th} highest modality cardinality test to the i^{th} level of the tree is the optimal way to decrease the depth of the tree and so to decrease its objective function K value.

Let the respective occurrence probabilities p_i with $i \in \{1, ..., n_F\}$ of the n_F faults of the F set be such that $1 \geq p_1 > ... > p_{n_F} \geq 0$. Let also s_j and s_k be two particular tests among the S set for which their respective modality cardinality are n_M^j and n_M^k such that $n_M^j > n_M^k$ and $n_M^j < n_F \leq (n_M^j \times n_M^k)$.

In the diagnosis tree T_j, s_j is applied on first level and s_k on the second one, then, the number of faults n_F is such that $n_F = n_M^j - a + (a \times n_M^k)$ with $1 < a \leq n_M^j$. So, the expression of the objective function K for this diagnosis tree T_j can be written as shown on equation 14.

$$K(T_j) = \left(1 \times \sum_{i=1}^{n_M^j - a} p_i\right) + \left(2 \times \sum_{i=n_M^j - a + 1}^{n_F} p_i\right) \quad (14)$$

In the diagnosis tree T_k, s_k is applied on first level and s_j on the second one, then, the number of faults n_F is such that $n_F = n_M^k - b + (b \times n_M^j)$ with $1 < b \leq n_M^j$. So, the expression of the objective function K for this diagnosis tree T_k can be written as shown on equation 15.

$$K(T_k) = \left(1 \times \sum_{i=1}^{n_M^k - b} p_i\right) + \left(2 \times \sum_{i=n_M^k - b + 1}^{n_F} p_i\right) \quad (15)$$

So, if $n_M^j - a \geq n_M^k - b$ then $K(T_j) \leq K(T_k)$ else $K(T_j) > K(T_k)$.

$$n_M^j - a \geq n_M^k - b \Leftrightarrow n_M^j - \frac{n_F - n_M^j}{n_M^k - 1} \geq n_M^k - \frac{n_F - n_M^k}{n_M^j - 1}$$

After development and simplification, this expression can be written as follows:

$$n_M^j((n_M^j \times n_M^k) - n_F) \geq n_M^k((n_M^j \times n_M^k) - n_F)$$

Since $n_F \leq (n_M^j \times n_M^k)$ and $n_M^j > n_M^k$ are true by hypothesis, $K(T_j) \leq K(T_k)$. Consequently, applying the i^{th} highest modality cardinality test on the i^{th} level of the diagnosis tree allows to reach optimality.

D. Heuristic in Case of Non Exclusive Multi-Modality Tests

As previously seen, using different test costs, an AO* algorithm has to be used in order to obtain an optimal diagnosis tree. Contrary to the case of exclusive binary tests, Pattipati heuristic is not computed from the optimal binary diagnosis tree T_b^* obtained by the Huffman algorithm, but from the optimal multi-modality diagnosis tree T_m^* obtained by the multi-modality extension algorithm applied on the T_b^* diagnosis tree initially computed.

According to the Pattipati's method, the n_S available tests are always ordered by their increasing costs such as $0 < c_1 \leq ... \leq c_{n_S}$ and the admissible heuristic H_p expressed as already shown on equation 8 where K' is the integer part of the optimal objective function value K^*.

This heuristic is equivalent to build a Huffman tree with the highest available modality cardinality tests affected by the lowest available test costs. Consequently, this heuristic reaches the admissibility property, and so, leads to the optimal tree by using the AO* algorithm.

However, if the tests which have the highest modality cardinalities do not have the lowest costs, the quality of the heuristic is decreased (i.e. the distance between H_p and H^* increases).

V. Optimal Discriminating tests subset

The only available relation between a discriminating tests subset S' with its relative optimal diagnosis tree T' and an optimal discriminating tests subset S^* with its relative optimal diagnosis tree T^* issued from a same tests set S is given by theorem 3.

Theorem 3 (Optimality)

$$S^* \subseteq S' \Rightarrow K(T') = K(T^*)$$

Proof 4 (Optimality) *If $S' = S^*$ then $T' = T^*$ (since they are obtained from the same AO* algorithm) and then $K(T') = K(T^*)$.*

If $S^* \subset S'$ then T' may be completely different from T^* but $K(T')$ is obligatory equal to $K(T^*)$. Since T^* may obtained by performing the AO algorithm from the tests set S as well as from the optimal tests subset S^*, equations 16 may be written.*

$$\begin{cases} S' \subseteq S & \Rightarrow \quad K(T') \geq K(T^*) \\ S^* \subseteq S' & \Rightarrow \quad K(T^*) \geq K(T') \end{cases} \quad (16)$$

Hence, the result $K(T') = K(T^*)$ easily follows.

However, it is impossible to have an a priori knowledge of one or several optimal tests subsets S^* without performing the AO* algorithm. The objective function K is dedicated to diagnosis trees but can not be applied to discriminating tests subsets. Actually, from a same discriminating tests subset, many diagnosis trees having distinct values according to the objective function K may be obtained.

Consequently, there is no way to reduce the initial tests set S to a discriminating tests subset S' such that $|S'| < |S|$ with the certainty that $S^* \subseteq S'$. Then, the only way to obtain the optimal diagnosis tree T^* is to perform the AO* algorithm from the initial tests set S, since $S^* \subseteq S$.

From the initial tests set S, it is impossible to define an optimal discriminating tests subset S^* or even a discriminating tests subsets S' such that $S^* \subseteq S'$ and $S' \neq S$.

Optimality is ensured only for the diagnosis trees obtained by execution of AO* algorithm from the S initial tests set. For any discriminating tests subsets S' of S, nothing can be told about the optimality of the corresponding diagnosis tree T' obtained by execution of the AO* algorithm.

VI. Suboptimal discriminating tests subset

A. Presentation

This section discusses methods which allow to generate sub-optimal discriminating tests subsets S' which are expected to give diagnosis trees T' as close as possible to the optimal one T^* obtained with the optimal discriminating tests subset S^*.

According to the objective function K, the optimality is based on both notions of cost and discrimination power of the subset of the available tests used in the diagnosis tree. By working exclusively on discriminating tests subsets, the notion of discrimination power is always reached. However, it is necessary to be able to evaluate the cost of these discriminating test subsets in order to select the one which has the lowest cost.

Obviously, the ideal evaluation of the cost of a discriminating tests subset $S^{'}$ would be proportional to the $K(T^{'})$ associated to the optimal diagnosis tree $T^{'}$ obtained by executing the AO* algorithm. However, it is very difficult to predict the $K(T^{'})$ value from the $S^{'}$ tests subset without knowing $T^{'}$ itself.

Consequently, the cost of a tests subset is then simply evaluated as the sum of the costs corresponding to the tests which constitute this subset.

B. Finding best discriminating tests subset

Finding the best discriminating tests subset S^{*} is itself a NP-complete problem. Actually, it is possible to show an polynomial reduction of this problem from the NP-complete 3-SAT problem following the NP-completeness demonstration of the knapsack problem (see [9]).

Let $G = \{g_1, ..., g_{n_G}\}$ be the set of the n_G signatures of the n_F faults of the F set among the n_S tests of the S set. . Since one fault has at least one signature, $n_G \geq n_F$. Each g_i signature has n_S components $\sigma_{i,j}$ with $j \in \{1, ..., n_S\}$ which is one of the n_M^j modalities m_k^j with $k \in \{0, ..., n_M^j - 1\}$ of the s_j test.

Let $V = (v_1, ..., v_{n_S})^T$, $\forall v_i \in \{0, 1\}$ be the test selection vector such that if s_i belongs to S^{*} then $v_i = 1$ else $v_i = 0$. If $D = Max_{j\in\{1,...,n_S\}} n_M^j$ then any signature g_i among the test subset defined by V may be evaluated according to the $E_{D,V}$ function as shown on equation 17.

$$E_{D,V}(g_i) = v_1 \times \sigma_{i,1} \times D^0 + \cdots + v_{n_S} \times \sigma_{i,n_S} \times D^{n_S - 1} \quad (17)$$

Then, a tests subset represented by the vector V discriminates all the faults of the F set if and only if the n_G relative evaluations of signatures reduced to the components corresponding to the subset are distinct two by two. This can be expressed as shown on system of $n_E = C^2_{n_G}$ equations 18 with $1 \leq k < l \leq n_G$.

$$\left\{ \begin{array}{ccccc} E_{D,V}(g_1) & - & E_{D,V}(g_2) & \neq & 0 \\ \vdots & - & \vdots & \neq & \vdots \\ E_{D,V}(g_k) & - & E_{D,V}(g_l) & \neq & 0 \\ \vdots & - & \vdots & \neq & \vdots \\ E_{D,V}(g_{n_G-1}) & - & E_{D,V}(g_{n_G}) & \neq & 0 \end{array} \right. \quad (18)$$

This optimization problem may then be expressed as : find the V vector corresponding to a discriminating tests subset such that $\sum_{j=1}^{n_S} v_j \times c_j$ is minimal.

The NP-completeness theory being restricted to decision problems (i.e. problems which have yes/no solutions), this optimization problem has to be formulated into a decision one by adding in any instance of this problem a vector $B = (b_1, ..., b_{n_E})^T$, $\forall b_i \in \mathbf{Z}^{*}$ corresponding to the gap between two distinct signatures over the selected tests.

Consequently, the optimization problem may be translated into the following decision problem : does it exist a vector $V = (v_1, ..., v_{n_S})^T$ which verifies equations system 19 with $1 \leq k < l \leq n_G$ for the given instance (S, G, B) ($S = \{s_1, ..., s_{n_S}\}$, $G = \{g_1, ..., g_{n_G}\}$ with $g_i = (\sigma_{i,1}, ..., \sigma_{i,n_S})^T$ and $B = (b_1, ..., b_{n_E})^T$).

$$\left\{ \begin{array}{ccccc} E_{D,V}(g_1) & - & E_{D,V}(g_2) & = & b_1 \\ \vdots & - & \vdots & = & \vdots \\ E_{D,V}(g_k) & - & E_{D,V}(g_l) & = & b_{l-k+\sum_{\alpha=1}^{k-1}(n_E-\alpha)} \\ \vdots & - & \vdots & = & \vdots \\ E_{D,V}(g_{n_G-1}) & - & E_{D,V}(g_{n_G}) & = & b_{n_E} \end{array} \right. \quad (19)$$

Let A be the $(n_E \times n_S)$ matrix composed of $a_{i,j}$ elements with $i \in \{1, ..., n_E\}$ and $j \in \{1, ..., n_S\}$. To each $a_{i,j}$ is associated one term $(\sigma_{k,j} - \sigma_{l,j}) \times D^{j-1}$ of the equations system 19 with $1 \leq k < l \leq n_G$ such that this equation system may be written under a matrix form $A \times V = B$ as shown on equation 20.

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n_S} \\ \vdots & & \vdots \\ a_{n_E,1} & \cdots & a_{n_E,n_S} \end{pmatrix} \times \begin{pmatrix} v_1 \\ \vdots \\ v_{n_S} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_{n_E} \end{pmatrix} \quad (20)$$

At last, the previous decision problem may be simplified into the following one : does it exist a vector $V = (v_1, ..., v_{n_S})^T$ which verifies $A \times V = B$ for the given instance (A, B) with $\{a_{1,1}, ..., a_{n_E,n_S}\} \in \mathbf{Z}$ and $\{b_1, ..., b_{n_E}\} \in \mathbf{Z}^{*}$.

Theorem 4 (NP-completeness) *The problem, defined as previously and called the best discriminating subset problem, is an NP-complete problem.*

Proof 5 (NP-completeness) *Consider a Turing machine M that on any instance (A, B) of the problem nondeterministically assigns values from $\{0, 1\}$ to $\{v_1, ..., v_{n_S}\}$, checks whether equations system 20 is verified and accepts if and only if all these equalities hold. M can be of polynomial time complexity. Therefore, the best discriminating subset problem is in NP.*

To show that the best discriminating subset problem is NP-complete, consider any instance I of the 3-satisfiability problem. Let x_i with $i \in \{1, ..., m\}$ be the variables in the Boolean expression I. I is a conjunction $C_1 \wedge ... \wedge C_k$ of clauses C_i with $i \in \{1, ..., k\}$. Each C_i is a *disjunction of some literals $c_{i,j}$ with $j \in \{1,...,3\}$. Each $c_{i,j}$ is x_l or $\neg x_l$ for $l \in \{1,...,m\}$.*

From the Boolean expression I the linear equations system 21 can be built.

$$\left\{\begin{array}{ccccccccccc} & & & & & & x_1 & + & \bar{x_1} & = & 1 \\ & & & & & & \vdots & + & \vdots & = & \vdots \\ & & & & & & x_m & + & \bar{x_m} & = & 1 \\ c_{1,1} & + & c_{1,2} & + & c_{1,3} & + & y_{1,1} & + & y_{1,2} & = & 3 \\ \vdots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & = & \vdots \\ c_{k,1} & + & c_{k,2} & + & c_{k,3} & + & y_{k,1} & + & y_{k,2} & = & 3 \end{array}\right. \tag{21}$$

The system 21 has the variables $\{x_1,...,x_m,\bar{x_1},...,\bar{x_m}, y_{1,1},...,y_{k,2}\}$. The variable x_l (resp. $\bar{x_l}$) with $l \in \{1,...,m\}$ in 21 corresponds to the literal x_i (resp. $\neg x_i$) in I. $c_{i,j}$ stands for the variable x_l (resp. $\bar{x_l}$) in 21, if x_l (resp. $\neg x_l$) is the j^{th} literal in C_i.

Each equation of the form $x_i + \bar{x_i} = 1$ has a solution over $\{0,1\}$ if and only if either $x_i = 1$ and $\bar{x_i} = 0$, or $x_i = 0$ and $\bar{x_i} = 1$. Each equation of the form $c_{i,1}+c_{i,2}+c_{i,3}+y_{i,1}+y_{i,2} = 3$ has a solution over $\{0,1\}$ if and only if at least one of the equalities $c_{i,1} = 1$, $c_{i,2} = 1$, and $c_{i,3} = 1$ holds. It follows that the system 21 has a solution over $\{0,1\}$ if and only if the Boolean expression I is satisfiable.

The system 21 can be represented in a matrix form $A \times Z = B$ as shown on equation 22 with $n_L = m + k$ and $n_C = 2m + 2k$.

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n_C} \\ \vdots & & \vdots \\ a_{n_L,1} & \cdots & a_{n_L,n_C} \end{pmatrix} \times \begin{pmatrix} z_1 \\ \vdots \\ z_{n_C} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_{n_L} \end{pmatrix} \tag{22}$$

The variables $\{z_1,...,z_{n_C}\}$ in the vector form stand for the variables $\{x_1,...,x_m,\bar{x_1},...,\bar{x_m},y_{1,1},...,y_{k,2}\}$ of 21, respectively. $a_{i,j}$ is assumed to be the coefficient of z_j in the i^{th} equation of 21. b_i is assumed to be the constant in the right-hand side of the i^{th} equation in 21.

As a result, the instance I of the 3-satisfiability problem is satisfiable if and only if the (A,B) instance of the best discriminating tests subset problem has a positive Z solution.

Moreover, a polynomially time-bounded, deterministic Turing machine can similarly construct corresponding instance of the best discriminating tests subset problem, from each instance I of the 3-satisfiability problem.

Consequently, the NP-hardness of the best discriminating subset problem follows from the NP-hardness of the 3-satisfiability problem.

C. tests subsets generation algorithm

This subsection proposes a two step discriminating tests subsets generation algorithm. The first step allow to obtain, from the S initial tests set, a discriminating tests subset, called S'_{just} such that $S'_{just} \subseteq S$. The second step generates from the previously obtained just discriminating tests subset S'_{just}, a minimal discriminating tests subset, called S'_{min} such that $S'_{min} \subseteq S'_{just} \subseteq S$.

First of all, the n_S available tests of the S set are ordered according to their decreasing heuristic function E value order. Then, let s^E_j with $j \in \{1,...,n_S\}$ be these ordered n_S tests s_j of the S tests set such that $E(s^E_1) \geq ... \geq E(s^E_{n_S}) > 0$.

Then, tests are selected one by one, according to the previous order, until a discriminating subset, called S'_{just}, of n_{just} tests such that $n_{just} \in \{1,...,n_S\}$ is obtained.

From the just discriminating tests subset S'_{just} previously obtained, a minimal discriminating tests subset, called S'_{min}, may then be generated.

First of all, a current tests subset S_c is initialized to S'_{just}. The algorithm then tries to remove one by one, from the current tests subset S_c, each of the n_{just} tests of S'_{just}, starting from the last test s^E_{just} to the first one s^E_1.

The current tried test s^E_j with $j \in \{1,...,n_{just}\}$ is removed from S_c if and only if the $S_c - \{s^E_j\}$ tests subset is still a discriminating one. The final current S_c tests subset is the definitive minimal discriminating tests subset S'_{min}.

VII. Quality evaluation

A. Presentation

The aim of this section is to propose a quality criterion, called Q, in order to compare different methods allowing to obtain suboptimal discriminating tests subsets S' from the initial tests set S. This quality criterion should be able quantify, for each of these methods, the gain in terms of AO* algorithm processing time according to the loss in terms of optimality gap.

B. Optimality gap

Let T^*, T'_{just} and T'_{min} be the optimal diagnosis trees resulting from the execution of the AO* algorithm from the initial tests set S, from the just discriminating test subset S'_{just} and from the minimal discriminating one S'_{min} , respectively.

As S'_{min} is obtained from S'_{just} itself obtained from S, $S'_{min} \subseteq S'_{just} \subseteq S$, then, $K(T^*) \leq K(T'_{just}) \leq K(T'_{min})$ according to the objective function K.

Now, let S^* be the optimal discriminating tests subset then, the optimal diagnosis tree resulting from the application of the AO* algorithm on S^* as on S is T^*. According to theorem 3, if $S^* \subseteq S'_{min}$ then $K(T^*) = K(T'_{just}) = K(T'_{min})$.

1. If $S^* \subseteq S^{'}_{min}$ then $S^* = S^{'}_{min}$. Actually, from the definition, S^* is at least a minimal discriminating tests subset. So, for any S^*, it exists at least one minimal discriminating tests S^*_{min} such that $S^*_{min} \subseteq S^*$. Consequently, if $S^* \subseteq S^{'}_{min}$ then S^* is itself the only one minimal discriminating tests subset S^*_{min} which is equal to $S^{'}_{min}$ otherwise $S^{'}_{min}$ would not be a minimal discriminating tests subset.
2. If $S^* \subseteq S^{'}_{just}$ but $S^* \nsubseteq S^{'}_{min}$ then $K(T^*) = K(T^{'}_{just}) < K(T^{'}_{min})$.
3. If $S^* \nsubseteq S^{'}_{just}$ then $K(T^*) < K(T^{'}_{just}) \leq K(T^{'}_{min})$.

These three inclusion relationships between S, $S^{'}_{just}$, $S^{'}_{min}$ and S^* are illustrated on figure 5.

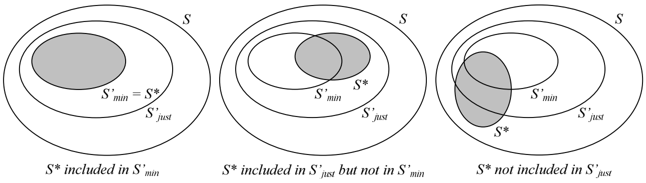

Fig. 5. Inclusion relationships between S, $S^{'}_{just}$, $S^{'}_{min}$ and S^*

C. Processing time

Let τ, $\tau^{'}_{just}$ and $\tau^{'}_{min}$ be the processing times of the AO* algorithm executed from the initial tests set S, from the just discriminating tests subset $S^{'}_{just}$ and from the minimal discriminating one $S^{'}_{min}$, respectively.

As $S^{'}_{min}$ is obtained from $S^{'}_{just}$ itself obtained from S, $S^{'}_{min} \subseteq S^{'}_{just} \subseteq S$, then $Card(S^{'}_{min}) \leq Card(S^{'}_{just}) \leq Card(S)$, hence $\tau^{'}_{min} \leq \tau^{'}_{just} \leq \tau$.

Now, let τ^* be the processing time of the AO* algorithm executed from the optimal discriminating tests subset S^*. As seen previously, no information are available about S^* and even about $Card(S^*)$ except, of course, $Card(S^*) \leq Card(S)$.

Consequently, no comparison can be done between $Card(S^*)$ and $Card(S^{'}_{min})$ or even $Card(S^{'}_{just})$. The only available result is that, for any S^*_{min} (at least one) such that $S^*_{min} \subseteq S^*$ (see the remark in the previous subsection), $Card(S^*_{min}) \leq Card(S^*)$. However, nothing is known about the order relationship between $Card(S^*_{min})$ and $Card(S^{'}_{min})$.

Hence, no comparison can also be done between τ^* and $\tau^{'}_{min}$ or even $\tau^{'}_{just}$. Then, the minimal processing AO* algorithm processing time may be either $\tau^{'}_{min}$ or τ^*.

D. quality criterion

Let S be the initial tests subset (supposed to be discriminating), τ, its relative AO* algorithm processing time and T^*, its relative optimal diagnosis tree.

Let S^* be an optimal discriminating tests subset, τ^*, its relative AO* algorithm processing time and T^*, its relative optimal diagnosis tree.

Let $S^{'}$ be either the $S^{'}_{just}$ just or the $S^{'}_{min}$ minimal discriminating tests subsets obtained from the initial tests set S, $\tau^{'}$, its relative AO* algorithm processing time and $T^{'}$, its relative optimal diagnosis tree.

$0 < K(T^*) \leq K(T^{'})$ and $0 < \tau^{'} \leq \tau$ are always verified. Moreover, if $S^{'}$ is generated according the just and minimal discriminating tests subset algorithm then, $S^{'} \in \{S^{'}_{just}, S^{'}_{min}\}$, and $0 < Min(\tau^*, \tau^{'}_{min}) \leq \tau^{'} \leq \tau$.

Consequently, the quality criterion Q, defined as shown on equation 23, illustrates the interest of a discriminating tests subset $S^{'}$ as the gain in terms of AO* algorithm processing time according to the loss in terms of optimality gap.

$$Q(S^{'}) = \frac{K(T^*)}{K(T^{'})} \times \frac{Min(\tau^*, \tau^{'}_{min})}{\tau^{'}} \tag{23}$$

This criterion may then be used, for any system to diagnose, to evaluate by a real $[0, 1]$ value the relative interests of the S initial, $S^{'}_{just}$ just, $S^{'}_{min}$ minimal or S^* optimal discriminating tests subsets.

E. Example

For 5 different systems Σ_i with $i \in \{1, ..., 5\}$ figure 6 details the computation of the $Q(S^{'})$ values with $S^{'}$ being the S initial, $S^{'}_{just}$ just, $S^{'}_{min}$ minimal or S^* optimal discriminating tests subsets.

$n_{S'}$ represents the number of tests selected in the $S^{'}$ tests set whereas $n_{T'}$ represents the number of tests effectively used in the $T^{'}$ diagnosis tree obtained by performing the AO* algorithm on the $S^{'}$ tests subset.

The processing time $\tau^{'}$ is not directly used in the quality criterion $Q(S^{'})$ computation. Actually, this information is disturbed by the fact that the AO* algorithm has sometimes to swap on the hard disk. The $\tau^{'}$ value then depends on the technical characteristics of the computer used to execute the application. Consequently, the information of the number of OR nodes which constitute the whole AND/OR search tree developed during the AO* algorithm execution is expressed for $\tau^{'}$.

Obviously, $K(T^{'})$ represents the objective function K value of the $T^{'}$ diagnosis tree obtained by performing the AO* algorithm on the $S^{'}$ tests subset.

On this figure, it is easy to see the $S^{'}_{min}$ and $S^{'}_{just}$ give acceptable suboptimal diagnosis tree according to the objective function K value for a considerable reduction of the application processing time expressed here as the total number of OR nodes expanded in the explicit AND/OR search tree.

		$n_{S'}$	$n_{T'}$	τ'	$K(T')$	$Q(S')$
Σ_1	S	7	4	46	2.5833	0.6304
	S'_{just}	4	4	29	2.5833	1.0000
	S'_{min}	4	4	29	2.5833	1.0000
	S^*	4	4	29	2.5833	1.0000
Σ_2	S	18	10	1493	3.0907	0.1594
	S'_{just}	11	9	728	3.0918	0.3268
	S'_{min}	8	8	238	3.1062	0.9950
	S^*	10	10	437	3.0907	0.5446
Σ_3	S	13	6	213	2.6823	0.3192
	S'_{just}	10	6	165	2.6823	0.4121
	S'_{min}	6	6	73	2.8448	0.8783
	S^*	6	6	68	2.6823	1.0000
Σ_3	S	12	9	287	3.7115	0.4077
	S'_{just}	11	7	251	3.7159	0.4656
	S'_{min}	7	7	117	3.7159	0.9988
	S^*	9	9	177	3.7115	0.6610
Σ_4	S	57	17	33671	2.3289	0.0543
	S'_{just}	33	17	10079	2.3289	0.1816
	S'_{min}	15	15	1830	2.8817	0.8082
	S^*	17	17	1840	2.3289	0.9946
Σ_5	S	211	21	319886	1.7346	0.0083
	S'_{just}	29	21	32308	1.7673	0.0802
	S'_{min}	17	17	2641	2.6019	0.6667
	S^*	21	21	10263	1.7346	0.2573

Fig. 6. $Q(S')$ values computation for different systems

VIII. Conclusion

This paper explains how Pattipati's method has to be extended to generate optimal diagnosis trees for non exclusive multi-modality tests.

However, the application processing time being proportional to the number n_S of tests in the initial tests set S, it seems interesting to reduce this S set in a subset S' such that $S' \subseteq S$ in oder to reduce also the application processing time. The problem is that optimality of the diagnosis tree T' obtained from a tests subset S' is not ensured anymore for the initial tests set S.

A quality criterion Q is proposed in order to quantify the gain in terms optimality measurement according to the K function, of an optimal diagnosis tree T' obtained by execution of the AO* algorithm from a discriminating tests subset S' face to the relative gain in terms of processing time of the AO* algorithm itself.

Two tests subsets S'_{just} and S'_{min} which can be obtain with polynomial algorithm from the initial tests set S are then proposed. Their respective Q values are computed for different systems. These Q values shows that is is possible to obtain acceptable suboptimal diagnosis trees and to reduce considerably the application processing time.

One can imagine many other polynomial tests subset selection algorithms, but anyway, it is always very difficult to give an idea of the optimality gap, that is to say a magnitude order of the difference between the optimal diagnosis tree T' obtained from the selected tests subset S' and the absolute optimal diagnosis tree T^*.

References

[1] P. P. Faure, L. Travé-Massuyès, and H. Poulard, "An Interval Model-Based Approach for Optimal Diagnosis Tree Generation," in *10th International Workshop on Principles of Diagnosis, DX-99*, (Loch Awe (Scotland)), pp. 78–89, 1999.

[2] K. R. Pattipati and M. G. Alexandridis, "Application of Heuristic Search and Information Theory to Sequential Fault Diagnosis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, pp. 872–887, July 1990.

[3] S. Srinivas, "On the computation of a priori failure probabilities from empirical measures of reliability," in *5th International Workshop on Principles of Diagnosis, DX-94*, (New Paltz (USA)), 1994.

[4] S. Srinivas, "Modeling failure priors and persistence in model-based diagnosis," in *11th Conference on Uncertainty in Artificial Intelligence*, (Montreal (Canada)), August 1995.

[5] K. R. Pattipati and M. Dontamsetty, "On a Generalized Test Sequencing Problem," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 392–396, Mar. 1992.

[6] A. Bagchi and A. Mahanti, "Admissible Heuristic Search in AND/OR Graphs," *Theoritical Computer Science*, vol. 24, pp. 207–219, 1983.

[7] A. Biasizzo, A. Žužek, and F. Novak, "Sequential diagnosis with asymmetrical tests," *The Computer Journal*, vol. 41, no. 3, pp. 163–170, 1998.

[8] A. Žužek, A. Biasizzo, and F. Novak, "Sequential diagnosis tool," *Microprocessor and Microsystems*, vol. 24, pp. 191–197, 2000.

[9] E. M. Gurari, *An Introduction to the Theory of Computation.* Computer Science Press, 1989.