1992

# Casual [i.e., causal] nets for diagnosis

Chung-Lung Chen
*Carnegie Mellon University*

Sarosh Talukdar

Carnegie Mellon University.Engineering Design Research Center.

**Casual Nets for Diagnosis**

Chung-Lung Chen, Sarosh Talukdar

EDRC 18-37-92

# CAUSAL NETS FOR DIAGNOSIS

Chung-Lung Chen Sarosh Talukdar

Engineering Design Research Center
Hammerschlag Hall 2221
Carnegie Mellon University
Pittsburgh, PA 15213
Ph: (412) 268-8778
Fax: (412) 268-5229

Fourth International Symposium on Expert Systems Application to Power Systems, La Trobe University, Melbourne, Australia, January 4-8, 1993.

# CAUSAL NETS FOR DIAGNOSIS

Chung-Lung Chen Sarosh N. Talukdar
Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213, USA

**Abstract:** Cause-effect relations are widely used for fault diagnosis, but a perception gap seems to exist between the causal reasoning technology and the real world expectation. This is because the causal modeling of a power system falls into a cyclic loop if not treated properly. This paper shows how a Bayesian net model can be formulated without loop. Based on this model, a qualitative way is proposed to evaluate the net and the diagnosis problem is transformed into an optimization problem. A special organization called A-Teams is used to solve this transformed problem. Real examples are given to illustrate the process and demonstrate the results.

**Keywords:** Heuristic Approaches, Diagnosis, Bayesian Networks.

## I. INTRODUCTION

A fault in a power system precipitates a train of operations: relays operate to identify the fault and breakers operate to isolate it. Each of these operations generates a signal, called an alarm, which is sent to an EMS (energy management system). The information content of these signals varies considerably with the devices in which they originate and the bandwidth of the communication system. For instance, old, electromechanical relays signal only that they have operated. In contrast, newer, digital relays can provide more information than most communication systems can handle.

Of course, relays and breakers sometimes fail to operate correctly. Moreover, communication systems can make errors in delivering signals. Therefore, the alarms received by an EMS do not always present a true or complete picture of relay and breaker operations. To further complicate the picture, these alarms may be the result of several faults that have occurred almost simultaneously in the power network.

The diagnosis problem is to take these uncertainties into account in determining what caused a given set of alarms. More precisely, the problem can be stated as follows:

Given: $C_0$, the pre-fault configuration of a power network; and A, a set of alarms;

Find: $H_1$, $H_2$,...., $H_N$, hypotheses that provide plausible explanations for A. Each of these hypotheses is a set whose elements are faults, protective system operations (such as breaker openings), misoperations (such as breakers that should have opened but did not) and communication errors (such as alarms that were sent but not received).

This paper is organized as follows: Section 2 discusses models of diagnosis. Section 3 introduces conventional Bayesian network as a tool to build a model. Section 4 suggests a new model for power systems based on Bayesian net. Section 5 proposes a qualitative probability approach which efficiently evaluates the qualitative fers the formal defini-
Bayesian net model. Section 6 of
tions of simple/multiple faults diagnosis problem for power systems. Section 7 introduces A-Teams as the organization to solve fault diagnosis problem. Section 8 explains different agents of A-Teams. Section 9 gives the A-Team structure. Section 10 shows the results. Section 11 gives the conclusion.

## II. MODELS FOR DIAGNOSIS

Variations of power system diagnosis problem have been the subject of research for a long time. Basically they can be divided into two categories: Monitoring

Information-Based Approach and Model-based Approach [1]. The first approach consists of organizing monitoring information from operating relays and tripped circuit breakers during a fault into a tree structure or in tabular form. The second approach models the structure and functions of the protective relaying system, simulates the fault conditions and compares the simulation results with the monitoring information to obtain the diagnoses. Generally speaking, the latter is more powerful if the model representation is accurate enough. We see only one model [2] covers important cause and effect relationships governing the behavior of protective systems. Unfortunately, with their model, one still can not visualize (1) the communication errors and (2) the changes in relay protection zone due to circuit breaker status. Since (1) and (2) are necessary for correct diagnosis while alarms received do not present a true or complete picture of device operation and simultaneous multiple faults which often occur in the lightning strokes, an improved model is needed. Our improved model bases on Bayesian Networks.

## III. BAYESIAN NETWORKS

Bayesian networks (also called belief networks and causal networks) are being used with increasing frequency to deal with uncertainties in artificial intelligence work [3]. In form, a Bayesian network is a directed, acyclic graph whose nodes represent uncertain events and whose arcs represent the dependencies among these events. Specifically, it can be defined as follows.

**Definition 1:** A Bayesian net is a tuple < N, E, S> where:

- $N = \{X_1, X_2, X_3, \ldots, X_n\}$ is a set of nodes and $x_i \in \{0, 1, \ldots\}$ is one of the discrete states that $X_i$ can be;
- $E = \{Y_1, Y_2, Y_3, \ldots, Y_k\}$ is a set of edges which connect one node to another;
- $S = \{Z_1, Z_2, Z_3, \ldots, Z_m\}$ is a set of all conditional probabilities $P(x_i \mid \Pi X_i)$ and $\Pi X_i$ is the set of adjacent predecessors of $X_i$ which have been assigned states.

The best way to understand Bayesian networks is to go through an example. Suppose one night I went home on a highway. I saw a car driving slowly in front of me. I started to think if the car had problem. I knew that a driver would slow down if he/she realized an engine problem. However, an inexperienced driver might feel uncomfortable to drive fast. With this only evidence, driving-slowly, I could not decide what had happened. When came near, I heard big noise from the engine. This gave me another clue of the event. With this evidence added, I evaluated the whole possibility again and now my belief of the car problem increased almost to one. This example is illustrated in Fig. 1.

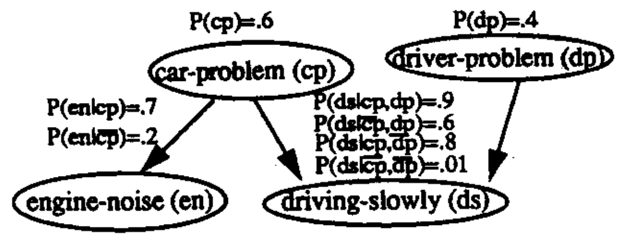



Fig. 1. A Bayesian network for the car-problem
P(*): Probability of (*)
$\bar{A}$: Not A.

The important thing to note is that there are two sets of probabilities associated with each node. The elements of the first set are called "prior" probabilities. They represent raw or historical information about the event associated with the nodes. Use the network, the "prior" probabilities and the evidence, one can find the second set of probabilities. They are called "evaluated", or "posterior", probabilities. They are modifications of the prior probabilities to take into account the latest evidence. In this example, the evaluated probability P(car-problem | engine-noise, driving-slowly)= 0.95 is much higher than prior probability P(car-problem)=0.6. The evaluated probability is sometimes called "*belief*".

Given a Bayesian network, its prior probabilities and some evidence, the "*beliefs*" identify the events likely to have caused the evidence.

## IV. BAYESIAN NETS FOR POWER SYSTEMS

In the case of power systems, we propose a Bayesian network model which has five levels of nodes. They are

Level 1: faults

Level 2: state variables out of "range" (e.g. high fault current or voltage)

Level 3: relay operations

Level 4: breaker operations

Level 5: alarms at EMS

Figure 2 is a very simple power system with two transmission lines and three buses. The corresponding Bayesian net, probabilities not specified, is shown in Figure 3.

In this figure, a node represents a random binary variable (1: on and 0: off); a solid arc from a parent node causes the child event to happen and a broken arc from a

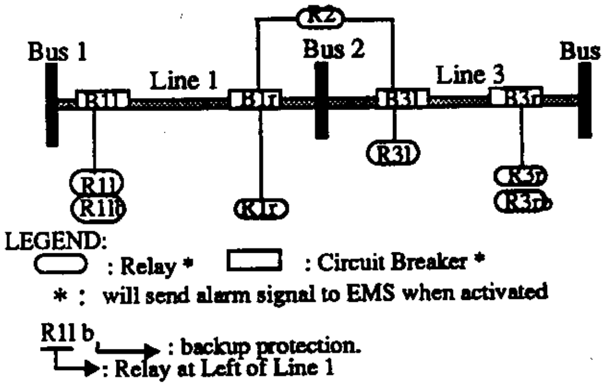



Figure 2: A sample electric power system

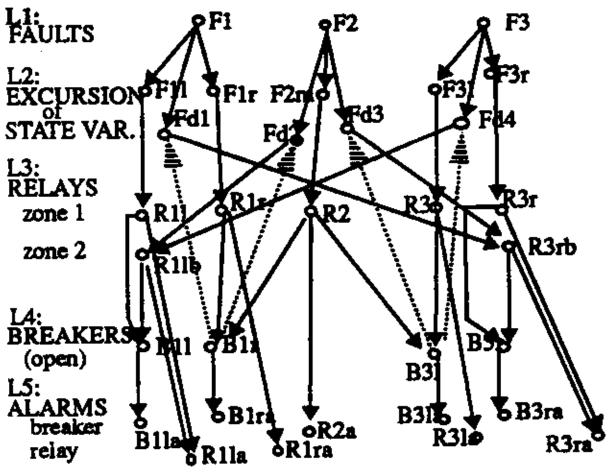



Figure 3: A Bayesian network for the power system of Fig. 2

parent node inhibits the child event to happen. For example, fault at Line1 (F1) will cause an excursion of fault current (F1r) in the right side of Line 1. This current will cause relay R1r to operate. The operation of R1r will cause breaker B1r to trip and also cause the alarm R1ra to turn on. The trip of B1r will cause the alarm B1ra to turn on and inhibit fault current Fd1, the current appeared near R3r, to continue. There are five important features of this net.

1. It introduces level 2 variables to make the cyclic net acyclic. Without level 2 variables, following cycle will happen: if F1 is "on" first, the relay R1r will operate and trip breaker B1r; the trip of B1r will go back to inhibit F1 to be "off". this cycle makes representation of Bayesian net impossible. In other words, Bayesian net can not treat cyclic events.

2. It shows communication errors. For example, if node R1ra is "off" and node R1r is "on", the alarm R1ra is missing. This is because the operated relay R1r("on") caused the alarm R1ra to turn "on" but it didn't respond correctly. On the other hand, if node R1ra is "on" and node R1r is "off", the alarm R1ra is a false alarm.

3. It shows topology changes. For example, if for some reason breaker B1r opened, level 2 nodes Fd1, F1r, Fd2 and all arcs connected will be removed from the network. This change changes relay protection zone and is easily visualized.

4. This net, though very wide horizontally (in the order of 1000), is only five layers deep. This strongly vertical configuration means alarms can be treated locally by looking a small part of the net.

5. We adopt the most common situations that one relay send only one alarm to EMS. In this net, the relay "backup" alarm is indistinguishable from its "primary" alarm.

In a real world, when faults happen, the set of alarms *A* initializes the level 5 nodes of the net *N* into 1 or 0. The diagnostic system should then find *marking M(N)*'s which set the other nodes of net *N* to 1 or 0 to explain the situation. Each explanation has a *belief* P(*M(N)*/*A*). Our task is to find all the plausible markings which have beliefs better than a given value.

One typical approach to this problem is "Belief Revision method" suggested by Judea Pearl [4]. The thesis of this method is:

> Let BEL(x*) be the highest probability of X=x given the best complementary marking of X's neighboring nodes. Upon receiving evidence (alarms in our case), start calculate BEL($x_i$*) for each node $X_i$ connected to the evidence. Propagate this calculation through all the node $X_i$ whenever its neighboring nodes' marking changed. The propagation iteratively executes until there is no more marking changes.

Unfortunately, there are several disadvantages of this conventional Bayesian nets approach:

1. It focuses on finding the best belief marking, or Most Probable Explanation (MPE), and can be extended to second best explanation only. That is not adequate for us.
2. Special handlings (clustering or conditioning) are needed to treat multiple connected nets such as our model for power systems.
3. Most of the time the probabilities associated with the net for belief evaluation are not available.
4. The computational effort required for this evaluation grows very rapidly with the size of the net.

How can one handle this situation?

## V. QUALITATIVE PROBABILITY

The ability to operate with qualitative information inspires the field of qualitative reasoning. Many works have been done for physical systems under the name of qualitative simulation. Nothing similar has been done for evaluation of belief of a Bayesian network as we know. We will define qualitative probability inside the model of qualitative Bayesian net and develope an algebra to operate on it.

**Definition 2:** Given

(1). a Bayesian net <N, E, S>

(2). every $P(x_i \mid \Pi X_i)$ is either 0, low, high or 1

Define

*Qualitative Bayesian Net* as a Bayesian net with every probability p(.) being discretized into $\wp(.) \in D = \{0, \varepsilon^n, \varepsilon^{n-1}, \ldots, \varepsilon, 1\text{-}\varepsilon, 1\}$, where $0 < \varepsilon << 1$.

For example, the *qualitative conditional probability* of every prior probability $P(x_i \mid \Pi X_i)$ is discretized to $\wp(x_i \mid \Pi X_i)$ and $\wp(x_i \mid \Pi X_i) \in \{0, \varepsilon, 1\text{-}\varepsilon, 1\}$.

Notice that conventional Bayesian networks have probability $P(.) \in [0, 1]$, a continuous interval. Here, we define $\wp(.) \in D = \{0, \varepsilon^n, \varepsilon^{n-1}, \ldots, \varepsilon, 1\text{-}\varepsilon, 1\}$, a discrete set. The main reason is that we can represent an interval by a discrete number. Also the discretization of this interval makes computation easier (we will see this soon). The set is illustrated in Fig. 4.

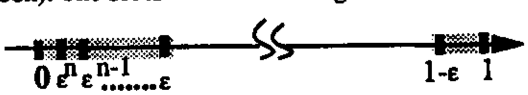



Fig. 4: The discrete set in qualitative probability

For X, Y $\in$ D, let us define operator $\otimes$ as follows.

**Definition 3:**

$X \otimes Y = Y \otimes X$

$X \otimes Y = 0$ if Y=0

$X \otimes Y = X$ if Y= 1- ε

$X \otimes Y = X$ if Y=1

$X \otimes Y = \varepsilon^{a+b}$ if $X=\varepsilon^a$, $Y=\varepsilon^b$ where $a,b \in \{1, .., n\}$.

**Theorem 1:** The *qualitative joint probability*

$\wp(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) = \wp(x_1 \mid \Pi X1) \otimes \wp(x_2 \mid \Pi X2) \otimes \ldots\ldots\ldots \wp(x_n \mid \Pi X_n)$.

Proof: Through recursively using of chain rule formula, we get

$P(x_1, x_2, \ldots x_i, x_{i+1}, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1)P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \ldots P(x_3 \mid x_2, x_1)P(x_2 \mid x_1)P(x_1)$.

Since $\wp(x_i \mid \Pi X_i) \in \{0, \varepsilon, 1\text{-}\varepsilon, 1\}$, the discretized $\wp(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ is different from $\wp(x_1 \mid \Pi X1) \otimes \wp(x_2 \mid \Pi X2) \otimes \ldots\ldots\ldots \wp(x_n \mid \Pi X_n)$ in the order of $\varepsilon$ every time Y= 1- ε appeared. That is, given X, Y (=1- ε) as qualitative conditional probabilities,

$$X \times Y = X \times (1\text{-}\varepsilon) = X + O(\varepsilon)X$$

$$X \otimes Y = X$$

Notice that the $O(\varepsilon)X$ has been omitted in the second equation where we use operator $\otimes$ in stead of multiplication. This is OK since $O(\varepsilon)$ is very small and the omission will not affect the discretization accuracy. This completes the proof.

**Theorem 2:** Let f be a Qualitative Bayesian Net defined on a set $N=\{X_1, X_2, X_3, \ldots, X_n\}$ of nodes. Given an initialized subset E (evidence) = $\{x_i, x_{i+1}, \ldots, x_n\}$ and a set of markings M(f) = $\{x_1, x_2, \ldots, x_{i-1}\}$, the linear ordering in terms of qualitative belief $\wp(M(f) \mid E)$ is the same as the linear ordering in terms of $\wp(M(f), E)$

Proof: The belief

$P(x_1, x_2, \ldots, x_{i-1} \mid x_i, x_{i+1}, \ldots, x_n) = P(x_1, x_2, \ldots, x_i, x_{i+1}, \ldots, x_n)/P(x_i, x_{i+1}, \ldots, x_n)$ -- (eq. 1)

Since $P(x_i, x_{i+1}, \ldots, x_n)$ is a constant $\alpha$ for the given set $(x_i, x_{i+1}, \ldots, x_n)$, we write (eq. 1) as

$P(x_1, x_2, .., x_{i-1} \mid x_i, x_{i+1}, .., x_n) = \alpha^{-1} P(x_1, .., x_i, x_{i+1}, .., x_n)$.

Therefore, after discretization the linear ordering of $\wp(x_1, x_2, \ldots, x_{i-1} \mid x_i, x_{i+1}, \ldots, x_n)$ has no difference with $\wp(x_1, x_2, \ldots, x_i, x_{i+1}, \ldots, x_n)$. This completes the proof.

Using Theorem 1 and 2, for each nonzero $\wp(M(f) \mid E)$, its ordering is inversely proportional to the number of $\wp(x_n \mid \Pi X_n)=\varepsilon$ it has. As a result, the evaluation for the belief of a given marking becomes a simple counting problem -- traverse over the marked net and count the number of node which has a $\wp(x_n \mid \Pi X_n)=\varepsilon$.

## VI. DIAGNOSIS PROBLEM IN TERMS OF QUALITATIVE BAYESIAN NETWORKS

However in the field of diagnosis, it is more familiar to talk about "abnormal element" in stead of $\wp(x_n \mid \Pi X_n)$, we will start by defining discrepancy.

**Definition 3:** Let Boolean variable $AB(X_n)$, an abnormal behavior of node $X_n$, be defined as

If $\wp(x_n \mid \Pi X_n) = \varepsilon$, then $AB(X_n) = 1$.

The *discrepancy* $\Theta(M(f) \mid E)$ of a marking M over a net *f* given evidence E is the sum of all $AB(X_n)$'s over $M(f)$.

**Theorem 3:** The problem of finding the linear ordering of a set of marking $M(f)$'s given evidence E in terms of *qualitative belief* $\wp(M(f) \mid E)$ is equivalent to finding their linear ordering in terms of *discrepancy* $\Theta(M(f) \mid E)$.

*Proof:* It comes from Theorem 2 and definition 3.

We now define a simple fault diagnosis problem in the terminology of qualitative Bayesian networks.

**Definition 4:** Let $f_0$ be a qualitative Bayesian net for power system defined as a tuple$<N_0, E_0, S_0>$; for each node $X_k$, $x_k \in \{1,0\}$; $\wp(x_k \mid \Pi X k)$ has been completely specified; $M(f_0)$ is a marking of $f_0$; A is a set of alarms and $\wp(M(f_0) \mid A)$ be the qualitative belief. A *simple fault diagnosis problem* is

Given: A, $f_0$, $\alpha$

Find: $\Theta(M(f_0) \mid A) \geq \alpha$

such that: $M(f_0) \Rightarrow f_1$

Where: $\alpha$ is a parameter, in the unit of discrepancy

M is the marking on the net $f_0$; Only one fault on this marking

$f_1$ is the qualitative Bayesian net of the new steady state after M happened and everything has settled down.

For multiple faults, without lose of generality, we assume that one fault happens after another and there is enough time for things to settle down before next fault occurs.This assumption makes it possible to visualize the multiple faults in several time stages and within each time stage there is only one fault. The multiple faults diagnosis problem is defined accordingly.

**Definition 5:** A *multiple faults diagnosis problem* is

Given: A, $f_0$, $\alpha$

Find: $\Sigma\Theta(M_k(f_{k-1}) \mid A_k) \geq \alpha$

such that: $M_k(f_{k-1}) \Rightarrow f_k$

$$Ak \subset Ak-1 \ldots \subset A0 = A$$

Where: $\alpha$ is a parameter, in the unit of discrepancies

$f_0, f_1, \ldots, f_k$ are a sequence of nets

$M_1, M_2, \ldots, M_k$ is a sequence of markings on those nets. $A_1, A_2, \ldots, A_k$ are sets of alarms remaining to be explained in each net $f_1, \ldots, f_k$.

Notice that this is a multi-stage optimization problem consists of k time stages. Can the coupling of each time stage be made more loose so their parallel solution becomes more easy? The answer is yes. Use the same philosophy of dynamic programming, one can decompose the problem into k subproblems and calculate the solution to the subproblem. The computation proceeds from the small subproblems to the larger subproblems, stores the answers somewhere else and assembles them later. Since at each time stage, there is only one simple fault diagnosis problem, any updating at this time stage needs no recalculation of its marking at last time stages. The decomposed problem is more general and easily be parallelized.

**Definition 6:** An *over all fault(s) diagnosis problem* is

Given: A, $f_0$, $\alpha$

Find: $\Theta(M_k(f_{k-1}) \mid A_{k-1}) \geq \alpha$

such that: $M_k(f_{k-1}) \Rightarrow f_k$

$$Ak \subset Ak-1 \ldots \subset A0 = A$$

## VII. A-TEAMS

The preceding sections have decomposed the multiple faults diagnosis problem into a set of smaller problems, each of the form and size of a simple fault diagnosis problem. The smaller problems are very loosely coupled and can be solved by a team of agents working in parallel, provided the team is properly organized.

An organization can be characterized by a quadruple: (C, D, A, I); where: C is a graph, called a control flow, that shows who supervises whom; D is a graph, called a data flow, that indicates who does what and who may exchange data with whom; A is a set of criteria, called activity constraints, that prescribe how agents are to operate in time; and I is a set of criteria, called insertion constraints, that specify what must be done to add or delete an agent from the organization [5]. The space of all organizations contains a set whose members, called A-Teams, have two very desirable properties. First, they are exceedingly open (new agents can be added to an A-Team almost effortlessly). Second, they are easily distributed (an A-Team fits naturally into a network of computers, its agents use only locally available information and it is less sensitive to communication delays than other organizations).

An A-Team is defined as follows [5]:

. C, its control flow, is null, meaning that it contains no supervisors; all its agents are autonomous.

. D, its data flow is cyclic so its agents can use feedback and iteration in developing solutions.

. A, its set of activity constraints, is empty, meaning that its agents are free to act when they wish. In particular, there is no predetermined schedule for exchanges of information; rather, exchanges occur asynchronously (spontaneously). Moreover, all the agents can work in parallel all the time.

. I, its set of insertion constraints, is unspecified but tends to be "half empty." (Because the agents are autonomous, there is no managerial superstructure to modify when an agent is added or deleted; the only changes that need to be made are to the agent itself.)

Clearly, the structure of an A-Team allows for anarchic behavior. Autonomous agents, each deciding for itself what it is going to do and when, if ever, it will communicate its results, can act at cross purposes. Surprisingly, there are simple strategies, not only to prevent this from happening, but to make A-Teams high in performance (fast at finding good solutions to difficult problems). Two categories of these strategies are [5]: mixing and socialization. "Mixing" means choosing agents so there is a balance between those that create solutions and those that destroy them. The balance must be such that a population of solutions is maintained and herded along paths that lead to profitable conclusions. "Socialization" means the insertion of a few instincts (rules) in each agent that cause it to seek a local consensus (align its actions with those of its immediate neighbors).

Often, a well selected mix of agents and a few simple instincts are sufficient to make an A-Team effective. This seems to be the case in the power system fault diagnosis problem.

## VIII. PARALLEL PROCESSING WITH A-TEAMS

There are many algorithms for the variations of the power system diagnosis such as: rule based approaches for eliminating redundant alarms [6], distributed approaches for synthesizing hypotheses [7], set-theoretic approaches [8], model-based approaches [9], and search techniques [10]. Since their algorithms base on different assumptions from this model, we can not use them without modification of our model. However, we do build a modified "patchwork synthesis" algorithm for reference purpose. This algorithm is:

Step 1. Given an unmarked causal net with alarms, initialize one fault as "on". For each "on" alarm node *c*, find the pair of its parent *b* and grandparent *a*. If *a* is "on", mark *b* "on". This is because *a* "on" gives an explanation support and *c* "on" gives a forecast support to *b* "on". Repeat this procedure until all the "on" alarms are checked. Put the marked net into the store (a data object). Do this for all the single faults.

Step 2. Take one copy of a causal net from store (the original one in the store is not changed), randomly select one relay or breaker node and mark it opposite to its original state. Evaluate it with the evaluation function. If its discrepancy is decreased, put it into the store for further improvement, otherwise delete it.

Step 3. Repeat this procedure until no further improvement.

We build 12 agents for the A-Team. Two of them use domain knowledge such as loops and isolation. A loop is a set of linked nodes including excursion of state variable, relay alarm, breaker alarm, relay and breaker. A primary loop is a loop without an inhibitive arc to the node of excursion. A backup loop is a loop with an inhibitive arc to the node of excursion. A fault is called isolated when that transmission element has been de-energized by the opening of the related breakers. These two can generate complete solutions by themselves. They are:

. Construction Agent 1 (CA1): Given alarms and the fault, find the primary loops they belong to. Mark the relay and breaker nodes "on". Repeat this procedure until the fault is isolated within the primary loop.

. Construction Agent 2 (CA2): Given alarms and the fault, find the primary loops they belong to. If any of other alarms indicates the backup loop alarms of the primary loop, mark the breaker node of the primary loop "off" and the other nodes "on". Repeat this procedure until the fault is isolated within the backup loop.

Other agents use pieces of causal net knowledge and are not so powerful. They are:

. Construction Agent 3 (CA3): Given alarms, find each alarm the pair of its parent and grandparent. If the grandparent is "on", mark its parent "on". Repeat this procedure until all the alarms are checked.

. Randomizing Agent (RA): Given a marking, randomly find one relay or breaker node and mark it opposite to its original state. Evaluate it with the evaluation function. If its discrepancy is decreased, put it into the store for further improvement, otherwise delete it.

. Stage Change Agent (SCA): Go through the input store to find a causal net which has reached steady state of the time stage. Change this causal net into a new net of next time stage.

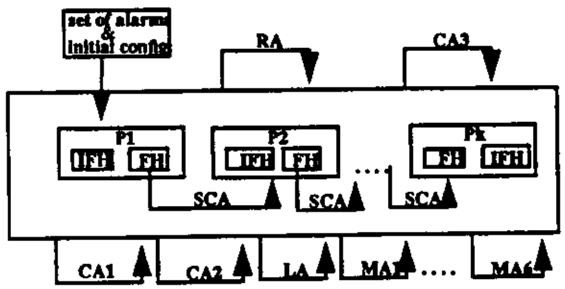
. Learning Agent (LA): Given two causal nets with same fault and time stage, extract the nodes which have same states of the two and mark other nodes "off" to produce a new causal net.

. Modification Agent 1 (MA1): Given a causal net with alarms and the fault, check to find a loop. If the first 3 nodes of the loop match the pattern "on", "on" and "on", then mark the last two nodes as "on" and "on". Do this once if there is any match. Else do it for another pattern: "on", "on" and "off" and mark "on" and "on".

. Modification Agent 2 to 6 (MA2 to MA6): Same as above except different patterns used.

## IX. STRUCTURE OF A-TEAM

There are different ways to put these agents together to generate a data-flow for an A-Team. Fig.5 is one possible configuration. The coordination policy is very sim-




LEGEND: FH = feasible hypotheses IFH = infeasible hypotheses
CA1 – Construction agent 1 RA – Randomizing agent
CA2 – Construction agent 2 SCA – Stage Change agent
CA3 – Construction agent 3 LA – Learning agent
MA1 – MA6 – Modification agents

($P_i$): Given alarms $A_i$ and causal net $f_{i-1}$
Find marking $M_i(f_{i-1})$
Such that $\Theta(M_i(f_{i-1}) \mid Ai) > \alpha i$

Figure 5: The A-Team data flow.

ple. The observed alarms and the initial configuration are used to initialize causal nets with different simple fault in bin $P_1$ (a small store) to start with. Bin $P_1$ is the store which keeps all the hypotheses, some are feasible and some are infeasible, of problem ($P_1$), the simple fault diagnosis problem in time stage 1. A feasible hypothesis is one which represents a steady state of current time stage. All agents select their inputs randomly and without repetition from their input stores. The generated causal nets with discrepancy measure are sent back to the store. There are several bins which keep hypotheses of different problem ($P_i$). SCA looks at each bin to find the feasible hypothesis. If there exists one in bin $P_i$, SCA will update it to the next bin $P_{i+1}$ by generating the corresponding causal net, reintroducing the alarms not yet explained and initializing it into different simple fault nets. This update means there may have multiple faults. Every hypothesis in the bin has its complete record of last stages' markings. The store keeps all the causal nets in order of each accumulated discrepancies. The better this measure one causal net has, the more chance it will be selected by the agents. There are only a certain amount of slots (e.g. 200) in the store. All the nets being ranked 201 and above are destroyed automatically. The process terminates when no further good causal nets can be generated.

## X. RESULTS

We ran this A-Team with four cases using a model system shown below (Fig. 6). The right hand side of each block represents the alarm; the black mark represents "on"state; the small block at each bus/line element represents the state of fault. The protection schema is designed as this:

The primary protections for each element are: R1r and R1l to protect line 1; R2 to protect bus 2; R3r and R3l to protect line 3; R4r and R4l to protect line 4.

R1l is to backup: R2 when B1r failed to trip; R3l when B3l failed to trip and R4l when B4l failed to trip.

R3r is to backup: R2 when B3l failed to trip; R1r when B1r failed to trip and R4l when B4l failed to trip.

R4r is to backup: R2 when B4l failed to trip; R1r when B1r failed to trip and R3l when B3l failed to trip.

For example, in case 1 there are three relay alarms: R1la, R1ra, R3ra and four breaker alarms: B1la, B3la, B3ra, B4la

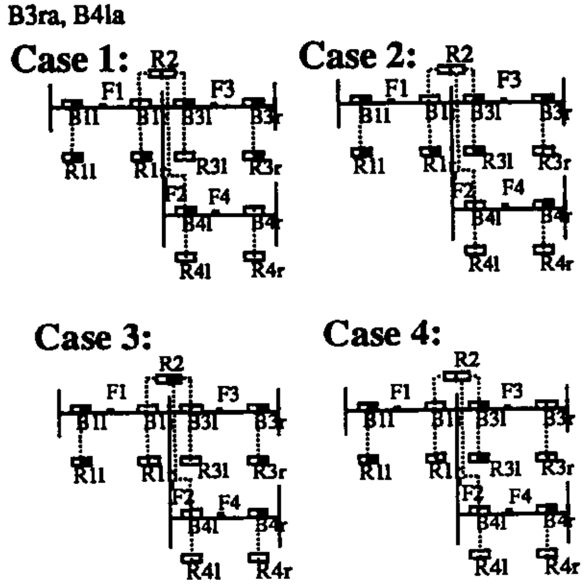



Figure 6: Test cases for power system fault diagnosis

For case 1 to 3, we took diagnoses with discrepancy less than the number of alarms as good diagnoses. For case 4, we took diagnoses with discrepancy less or equal to the number of alarms as good diagnoses. Table 1 shows the result for simple fault assumption.

| Problem | number of Alarms | Number of good diagnoses | | | |
|---|---|---|---|---|---|
| | | A-Team | P-S | CA1 | CA2 |
| case 1 | 7 | 29 | 19 | 2 | 0 |
| case 2 | 7 | 25 | 4 | 2 | 1 |
| case 3 | 6 | 14 | 10 | 0 | 3 |
| case 4 | 5 | 9 | 0 | 0 | 0 |

TABLE 1: Simple fault diagnosis

In table 1, column two represents the number of alarms for each case. Column three shows the number of good diagnoses found by A-Team. Column four to column seven show the results found by Patchwork Synthesis, Construction Agent 1 and Construction Agent 2 separately. Patchwork Synthesis sometimes (e.g. in case 3) works as good as A-Team. However, A-Team is more robust and always generates more good diagnoses. All the runs are performed in a DEC 5000 workstation. The agents are built with CLIPS (an expert system shell from NASA) embedded in C language. The store is built with RPC2 (a paradigm of communication between a server and several clients from CMU) which makes the asynchronous execution of each agent possible.

Use same set of cases, we take out the simple fault assumption and let A-Team go to find multiple faults, if they are there. Table 2 shows the results under normal operation of A-Team.

| Problem | number of Alarms | number of good diagnoses by A-Team |
|---|---|---|
| case 1 | 7 | 33 |
| case 2 | 7 | 31 |
| case 3 | 6 | 14* |
| case 4 | 5 | 9* |

*: No multiple faults.

TABLE 2: Multiple faults diagnosis

## XI. CONCLUSION

In this paper, we show that how a power system can be formulated into a Bayesian network model. Base on this model and a suggested qualitative way of belief evaluation, we show that how we transform the power system diagnosis problem into multi-stage optimization problem. We then use a special organization called A-Teams to solve this transformed problem. The result shows that A-Team effectively integrates different heuristics to give very good performance. The easiness of adding new heuristic without modification of the existing system is a big plus comparing to other diagnostic programs written in expert system shell.

## XII. REFERENCE


[1] Sekine Y. et. al., Fault Diagnosis of Power Systems (Proceedings of the IEEE, vol. 80, No. 5, May 1992.)

[2] Sekine Y. et. al., Fault section estimation using cause-effect network (Proc. Second Symposium on Expert Systems application to Power Systems, 1989. Seattle, WA.)

[3] Charniak Eugene, Bayesian Networks without tears (AI Magazine, vol. 12, no. 4, 1991.)

[4] Pearl Judea, Probabilistic Reasoning in Intelligent Systems : Networks of Plausable Inference (Morgan Kaufmann Publishers, Inc. 1988.)

[5] Talukdar S.N., Asynchronous Teams (Fourth Symposium on expert systems application to power sytems, January, 1993, Australia.)

[6] Wollenberg Bruce F., Feasibility study for an energy management system intelligent alarm processor (IEEE transactions on Power Systems, Vol. PWRS-1, No. 2, May 1986.)

[7] Cardozo E. and Talukdar S.N., D2: a Distributed Expert System for Fault Diagnosis (Proc. IEEE Power Industry Computer Application Conference, Montreal, Canada,May,1987.)

[8] Dijk Harold E., AI-Based Techniques for Alarm Handling (Proc. Third Symposium on Expert Systems application to Power Systems, April,1991. Japan.)

[9] Yang Chunling et. al., Expert System for Fault Section Estimation of Power System using Time Sequence Information (Proc. Third Symposium on Expert Systems application to Power Systems, April,1991.Japan.)

[10] Minakawa T. et. al., Requirement of on-line Data for Configuration of an Advanced Integrated Fault Detection Expert system (Proc. Third Symposium on Expert Systems application to Power Systems, April,1991. Japan.)