Probabilistic Intelligent Systems for Thermal Power Plants
Sistemas Inteligentes Probabilistas para Plantas Termoeléctricas

Pablo Héctor Ibargüengoytia, Alberto Reyes and Zenón Flores

Instituto de Investigaciones Eléctricas, Av. Reforma 113, Palmira, Cuernavaca, Mor., 62490, México
`{pibar,areyes,zfl}@iie.org.mx`

Abstract

Artificial Intelligence applications in large-scale industry, such as thermal power plants, require the ability to manage uncertainty because current applications are large, complex and influenced by unexpected events and their evolution in time. This paper shows some of the efforts developed at the Instituto de Investigaciones Eléctricas (IIE) to assist operators of thermal power plants in the diagnosis and planning tasks using probabilistic intelligent systems. A diagnosis system, a planning system and a decision support system are presented. The diagnosis system is based on qualitative probabilistic networks, and the decision support system uses influence diagrams. The planning system is based on the Markov Decision Processes formalism. These approaches were validated in different power plant applications. Current results have shown that the use of probabilistic techniques can play an important role in the design of intelligent support systems for thermal power plants.

Keywords: power plants, diagnosis, probabilistic reasoning, Bayesian networks, influence diagrams, Markov decision processes

Resumen

Las aplicaciones de Inteligencia Artificial (IA) en industrias de gran escala, como las centrales generadoras termoeléctricas, requieren de la habilidad de manejar incertidumbre ya que estas aplicaciones son complejas e influenciadas por eventos inesperados que evolucionan en el tiempo. Este artículo muestra algunos de los esfuerzos desarrollados en el Instituto de Investigaciones Eléctricas (IIE) para apoyar a los operadores de plantas termoeléctricas en sus tareas de planeación y diagnóstico, usando sistemas inteligentes probabilistas. Se presentan en este artículo un sistema de diagnóstico, un sistema de planificación y un sistema de soporte a las decisiones. El sistema de diagnóstico está basado en redes probabilistas cualitativas y el sistema de diagnóstico en diagramas de influencia. El sistema de planificación está basado en el formalismo de los procesos de decisión de Markov. Estos tres sistemas fueron validados en diferentes aplicaciones dentro de la operación de la planta termoeléctrica. Los resultados obtenidos muestran que las técnicas probabilistas pueden jugar un importante papel en el diseño de sistemas de ayuda en la operación de plantas termoeléctricas.

Palabras clave: Plantas termoeléctricas, diagnóstico, razonamiento probabilista, redes Bayesianas, diagramas de influencia, procesos de decisión de Markov.

1 Introduction

A thermal power plant can be described by a great variety of processes with multiples state variables, events and disturbances. The state variables change over time in response to both events and disturbances, as well as the transition of time itself. In this process, a variable (or signal) exceeding its specified range of normal operation is considered an event, and a sequence of events that have the same underlying cause are considered as a disturbance. During disturbances or faults, the operator must determine the best recovery action according to the type and sequence of the signals received.

Besides the multimodal dynamics of a power plant there are other factors and conditions that make more complex its operation. Faced with vast amount of raw process data, human operators will require the time consuming task of analyzing the incoming data in order to understand the plant condition. On the other hand, current control systems do not provide the means for intelligent interpretation of sensor data, diagnostic problems, copying with large process disturbances or predicting the consequences of control action. Furthermore, the age of a plant, for instance, increases the probability of failures in process equipment, which in the future will affect its reliability and performance. At the same time, power plants require to keep higher production profits, safer operation and stringent

environment regulation. In general, these situations not only increase the number of diagnoses and control decisions that a human operator must make, but they limit operators to contribute timely with effective solutions.

The computer and information technology have been extensively used in thermal plant process operation. Distributed control systems (DCS) and management information systems (MIS) have been playing an important role to show the plant status. However, in non-routine operations such as equipment failures and extreme operation (start up phase, changes in the load, etc.), human operators have to rely on their own experience. Process operations are knowledge-intensive work tasks because thermal plants are large, complex and influenced by unexpected disturbances and event over the time. The process industry demands new computer integrated technologies that reduce operator´s working burden by providing operation support systems. Artificial Intelligence (AI) has been considered promising to deal with problems that require both, human expertise and heuristic combined.

The Electrical Research Institute of Mexico (IIE) has been working in various fields related to operational support using AI techniques. Special interest has been developed in uncertainty management, given the complexity of the application. This paper describes some of the research in the context of probabilistic intelligent system applications for the operation of thermal power plants. The first system uses Bayesian networks with quantified states for diagnosis of gas turbines. The second system demonstrates the utilization of influence diagrams in the diagnosis, and the third system applies Markov decision processes to find the optimal recommendation to the operator in case of abnormal events or optimization ends.

This paper is organized as follows. First, section 2 presents a brief background about probabilistic graphical models and introduces the related work in this area. Section 3 describes the basis of the diagnosis systems, namely the qualitative and probabilistic diagnosis system in section 3.1, and the decision support system utilizing influence diagrams in section 3.2. Next, section 4 explains the planning system based on Markov decision processes and how a power plant application can be specified in this framework for making optimal decisions on line. In each of these projects, the experiments carried out are described discussing the results obtained. Finally, section 5 concludes the paper and establishes the future direction in this work.

2 Technical Background

This section briefly introduces the basis of probabilistic reasoning techniques utilized in this paper, namely Bayesian networks, influence diagrams and Markov decision processes (MDP). The following sections will describe the probabilistic intelligent systems developed under these formalisms.

2.1. Bayesian Networks
By definition, a Bayesian network is a directed acyclic graph (DAG) representing the joint probability distribution of all variables in the domain [10]. The nodes represent the variables and the arcs represent the probabilistic relation between the variables. The topology of the network gives direct information about the dependency relationship between the variables involved. In particular, it represents what variables are conditionally independent given another variables, and what variables are dependent of others. The variable destine of the arc is probabilistically dependent of the variable at the source of the arc. This follows the Bayes theorem that allows calculating the probability of a hypothesis given certain evidence.

As an example, if we want to calculate the probability of a faulty turbine hypothesis $P(H \mid E)$ given that we observe high vibration as evidence, we could easily calculate by counting the times that we observe high vibration given that we knew that the turbine failed, i.e., $P(E \mid H)$.

Given a knowledge base represented as a Bayesian network, it can be used to reason about the consequences of specific input data, by what is called probabilistic reasoning. This consists in assigning a value to the input variables, and propagating their effect through the network to update the probability of the hypothesis variables. The updating of the certainty measures is consistent with probability theory, based on the application of Bayesian calculus and the dependencies represented in the network.

2.2. Influence Diagrams

An influence diagram (ID) is a Bayesian network augmented with decision variables and utility functions. Thus, the nodes in the ID can be partitioned into three disjoint subsets: decision nodes $\mathbf{V}_D$, chance nodes $\mathbf{V}_C$, and utility nodes $\mathbf{V}_U$. Chance nodes represent variables or events that are not under the direct control of the decision maker. Decision nodes correspond to actions under the direct control of the decision maker. We suppose that there is a total ordering among the decisions, which indicates the order in which the decisions are made. A stochastic policy for a decision D is a probability distribution defined over D and conditioned on the set of its informational predecessors *iPred*, $P_D(d \mid \text{iPred}(D))$. If P_D is degenerate (consisting of ones and zeros only) then we say that the policy is deterministic. For a summary of IDs see [8].

2.3. Factored MDPs

The formalism of Markov Decision Processes (MDPs) [11] provides a powerful framework for solving sequential decision problems under uncertainty. A Markov decision process (MDP) models a sequential decision problem, in which a system evolves in time and is controlled by an agent. The system dynamics is governed by a probabilistic transition function Φ that maps states $\mathbf{S}$ and actions $\mathbf{A}$ to new states $\mathbf{S}'$. At each time, an agent receives a reward R that depends on the current state s and the applied action a. Thus, the main problem is to find a control strategy or policy π that maximizes the expected reward V over time.

Formally, an MDP is a tuple $M = <S, A, \Phi, R>$, where S is a finite set of states $\{s_1, ..., s_n\}$. A is a finite set of actions for all states. $\Phi : A \times S \times S$ is the state transition function specified as a probability distribution. The probability of reaching state s' by performing action a in state s is written as $\Phi(a, s, s')$. $R : S \times A \rightarrow \Re$ is the reward function. $R(s, a)$ is the reward that the agent receives if it takes action a in state s.

For the discounted infinite-horizon case with any given discount factor γ, there is a policy π^* that is a mapping $S \rightarrow A$ that selects an action for each state, it is optimal regardless of the starting state, and that satisfies the Bellman equation [4]:

$$V^{\pi}(s) = max_a\{R(s,a) + \gamma \sum_{s \in \mathbf{S}} \Phi(a, s, s')V^{\pi}(s')\} \tag{1}$$

Two methods for solving this equation and finding an optimal policy for an MDP are: (a) dynamic programming and (b) linear programming [11].

In a factored MDP, the set of states is described via a set of random variables S = { $X_1, ..., X_n$ }, where each X_i takes on values in some finite domain $Dom(X_i)$. A state x defines a value $x_i \in Dom(X_i)$ for each variable X_i. Thus, when the set of states S = $Dom(X_i)$ is exponentially large, it results impractical to represent the transition model explicitly as matrices. Fortunately, the framework of dynamic Bayesian networks (DBN) gives us the tools to describe the transition model concisely. In these representations, the post-action nodes (at the time t+1) contain smaller matrices with the probabilities of their values given their parents' values under the effects of an action. For a more detailed description of factored MDPs see [1].

2.4. Related Work

In the context of our application domain, some of the intelligent systems found in the literature are: ASTRAL [2], which is a simulator-based assistant for power operator's training. Its main functions are the recognition of the actions executed by an operator in a plant simulator, and the classification of detected errors with respect to an expected behavior. The main effort in this work is oriented to the development of explanation systems that support

the operator's understanding about the plant state. SOCRATES [14] is a real time assistant for control center operators in alarm processing and energy restoration. The core of the system is SPARSE, an expert system initially developed to use it in power transmission and distribution control centers. SOCRATES also provides an intelligent tutor, SPARSE-IT, which fulfill two purposes: i) to show the users how a trained operator solves problems and ii) to train the user to deal with specific situations and evaluate its performance. SART is a traffic control support system for the French subway (SART is the acronym for the system in French). The SART project implements an intelligent tutor with several functions such as: knowledge acquisition from operators, traffic simulation, model management of the network to test with alternative cases, enumeration of alternative solutions to incidents, and training of new operators. SART uses a multi-agent approach to have an evolutionary architecture.

3 Diagnosis systems

The objective of a diagnostics system is to determinate the main cause of a fault or disturbance. The analysis starts when an event is detected by the supervisory system. The analysis is performed separately from other modules, so that data acquisition can continue while the diagnosis system is under way. The IIE has been working in different approaches for the diagnosis of faults and disturbance based on the following requirements: early detection and diagnosis, isolability, robustness, multiple faults, explanation facility, adaptability, reasonable storage and computational requirements. In this section two of these approaches will be presented, one using a Bayesian and qualitative model, and other using influence diagrams.

3.1. Probabilistic and qualitative diagnostic model
This section presents a qualitative and probabilistic diagnosis system for gas turbines [6]. Three special challenges are solved in this project. The first is the definition of the most common faults in the diverse stages of the operation of a gas turbine, namely starting up, normal generation of power, and special maneuvers. The second is the construction of probabilistic models of the gas turbine, given real data from the plant and special advice from the experts. Third, since most of the signals utilized are continuous value variables, then a practical treatment is required. Thus, the main contribution of this work can be established as follows: It presents a complete methodology for the construction of real applications of Bayesian networks for diagnosis. This methodology consists in the discretization of variables to their qualitative tendencies, the model induction, and their integration in an architecture for on-line diagnosis.

Figure 1 shows the architecture of the proposed system. Data from the gas turbine simulator are utilized to create data files that are used by the learning process. In this research, the K2 induction algorithm for Bayesian network is used. K2 [3] uses a Bayesian score, $P(Bs, D)$, to rank different structures and it uses a greedy searching algorithm to maximize $P(Bs, D)$. Inputs to K2 include: a set of nodes, an ordering on the nodes, an upper bound on the number of parents a node may have, and a database D containing m cases of training data. For each node, K2 returns its most probable parents given the training data D. This is carried out off-line.

The learning module generates different models for different faults. This schema assumes that all the faults are independent and assumes that multiple simultaneous faults can be detected independently, i.e., they do not cancel one another. During simulator operation, data are collected every 250 ms. with the value of the variables before the simulated fault, and some samples of data after the simulated fault. This data is given to the automatic learning module and different models for different faults are obtained.

The initial experiments carried out in this project include the following faults:

- Low fuel supply pressure: The fuel provider presents low pressure in the supply. The control calculates the aperture of the gas valve based on a certain supply pressure, so incorrect aperture is commanded and the fault occurs.
- Fault in the compressor bleed valves: theses valves stabilize the pressure of the turbine in the start up phase. The pressure may increase or decrease to dangerous levels if the valve is stuck at incorrect values.
- Permanent stuck of the fuel valve: the fuel valve gets stuck and has no response to the control commands.
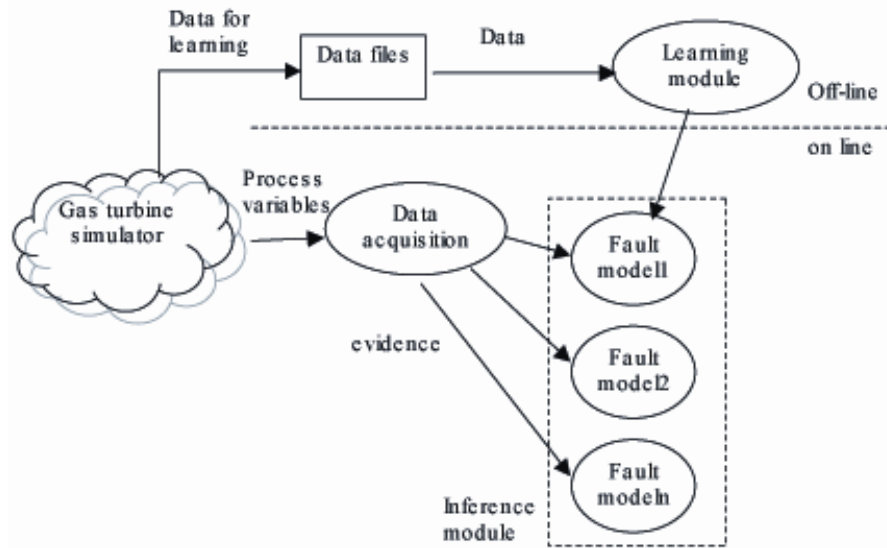
Fig. 1. Software architecture for multiple faults diagnosis of gas turbines

The fuel valve malfunction has three main sources. First, excess of friction in the actuator of the valve causes an increment in the difference between the real position and the commanded position. Second, the position of the valve remains stuck in some value. Third, the valve may unstuck unexpectedly and cause a large amount of fuel flow. This may cause an increment of temperature that the control algorithm will try to compensate, with the final result in a dangerous unbalance of the process and an oscillation of temperature.

The inference module in Figure 1 is formed by independent threads that receive evidence from the data acquisition module. Every one of these threads is a model of one fault considered. Data are refreshed every 250 ms, so the calculation of the probability of the different faults is calculated on–line and reported to the operator.

3.1.1. Experiments

The model was applied for fault diagnosis in a gas turbine of thermal power plant. Data for the experiments were obtained utilizing a gas turbine simulator at the IIE laboratory. Table 1 shows the set of variables utilized and their identifier

Table 1. Variables involved in the experiments

IDentifier	Description
MW	generated power
RMW	demand of power generation
XVc	valve of gas position
RXVc	reference of valve of gas position
EXVc	error in the valve of gas position
Pcc	combustion chamber pressure
Piqg	gas pressure in the burners

One experiment corresponds to the start up phase of the turbine, in which the speed goes from 0 to 5100 rpm without load. Other experiment corresponds to the power generation phase. The simulation includes generating from 2 MW to the maximum load of 24 MW, and then returning to 2 MW.

Practically all the signals in table 1 are continuous variables. Using a representation based on traditional Bayesian networks, a discretization process is required. However, if a large number of intervals is chosen, then a great amount of memory and computer time is required to obtain the posterior probabilities. An additional problem when using a large number of intervals is that many configurations are not represented in the database, and so the corresponding probabilities cannot be estimated, thus reducing the accuracy of the model. If a short number of intervals is chosen, then lack of expressivity is obtained. In this work, a qualification of the variables is proposed, inspired in the work by Suc and Bratko [13] that express the state of a system as qualitative changes. This is similar to the human reasoning performed in the control rooms where operators observe if a signal increases, decreases or remains unchanged. For example, the operator knows that at the start up phase, if the temperature increases, then the speed must also increase. Thus, the nodes of the network can have one of three states, according to the relation shown in table 2.

Table 2. Description of the qualitative changes of variable v

Condition	Variable state
if $v_t - v_{t-1} > \delta$	S_{up}
if $v_t - v_{t-1} < \delta$	S_{down}
if $\lvert v_t - v_{t-1} \rvert \leq \delta$	S_{rem}

where v_t represents the value of a variable at current time, v_{t-1} represents the previous value, and δ represents a small threshold dependant on a percentage according to the noise in the signal. S_{up}, S_{down}, S_{rem} are the state values that a variable can take.

Using this qualification process, data obtained in the simulator are transformed and applied to the automatic learning process.

Similar models were obtained for different faults as described in Figure 1. Table 2 shows the initial results. The rows indicate the case study applied to the prototype. The columns describe the results obtained for the different fault models given the same evidence.

Table 3. Preliminary results of the experiments

Case	Fault 1	Fault 2	Fault 3
start up, no fault	Not detected	15 %	Not detected
generating, no fault	Not detected	Not detected	Not detected
simulating fault 1	100 %	Not detected	Not detected
simulating fault 2	Not detected	98 %	Not detected
simulating fault 3	Not detected	Not detected	95 %

Experiments consisted in the operation of the gas turbine under different operating conditions. One specific condition was the start-up phase that was executed several times. Other conditions consisted in the steady power generation with changes in the load. Notice that the simulation is governed by a distributed control system running in parallel with the diagnosis system. In some of these experiments, and in certain point of time, a fault was inserted. For example, in the first row of Table 2, the experiments in all the instances of this case (start up phase without simulated fault), produced only a 15% probability of fault in the compressor bleed valve (Fault 2). This percentage is considered very low, and is considered as no fault detected. The propagation of probabilities of Fault 1 and Fault 3 resulted in 0 %, or not detected. In the last 4 rows, a specific fault was simulated and reported. Additional tests were developed in the laboratory, including variations of operational conditions and for other faults.

3.2. Decision support system using influence diagrams

This section presents a gas turbine decision support system that manages the natural uncertainty found in thermodynamic conditions of the gas turbine for power generation. This project tackles the problem of modeling the state of a process and calculates the optimal decision that the operators may take, according to the state detected. The final goal is the early detection of small deviations of the normal behavior in order to recommend actions for maintaining the generation of electricity in optimal condition [9]. In other words, when the turbine manifests changes in its behavior, the operator may decide to execute a revision of the turbine in order to observe defects in the turbine components. This is similar to our decision to take our car to the service station for revision when feeling vibrations or before vacations.

This project utilized a gas turbine simulator at the IIE laboratory. The simulation executed for the experiments consists in the generation of electricity, increasing load from 2 MW to 23 MW of power. Six analog signals were sampled every half second, so a number of 2111 records were obtained during a time close to 18 minutes. Table 4 explains the variables considered, their identifiers, their description, and the number of intervals in which every variable was discretized. The number of intervals was chosen according to the value range and the requirements of granularity in the experiments.

Table 4. Variables participating in the experiments

ID	Description	Num. states
MWdem	Demanded power	8
Vgas	Valve of gas position	10
MW	Generated power	8
TTXD1	Exhaust gas temperature 1	10
TTXD2	Exhaust gas temperature 2	10
TTXD3	Exhaust gas temperature 3	10
Decision	decision	3
Utility	Utility	24

The learning module of Elvira [5] with the K2 algorithm were executed utilizing the data table acquired in the simulation. The K2 algorithm was chosen since it permits to suggest the structure of the network through the ordering of the variables in the data table. This ordering was obtained with expert advice.

The *MWdem* variable represents the power demanded by the operator. In an automatic control system, this is the only set-up that the operator manipulates. The rest of the variables will move according to this set–up. The position of gas control valve *Vgas*, is the control variable that represents the aperture of the gas valve. Thus, if more power is demanded, wider aperture will be read in this variable. Variable *MW* is the measure of the mega watts generated. If the gas valve is opened, then the mega watts will increase. Finally, the *TTXDi* variables represent the exhaust gas temperature in the turbine, at different parts of the circumference.

Figure 2 shows the resulting influence diagram involving the six variables. In fact, the learning module defined only the structure concerning the six chance nodes (ovals in Fig. 2). Experts consider that the decision for turbine revision can be taken knowing the demand of power and the aperture of the gas valve. This means that if certain demand of power requires more gas than normal situations, then this is considered as a deviation. Thus, power plant operators can revise and clean the turbine while working (with low load of course), or stop the turbine for an off-line revision, or maintain the turbine working. The Decision node can consequently take three values: in-line revision, off-line revision, and do nothing.
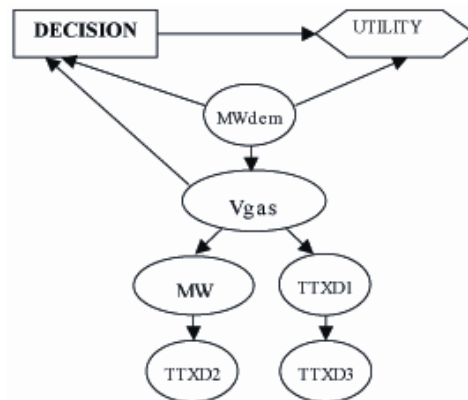
Fig. 2. Influence diagram obtained with automatic learning and experts advice

The utility of the decision depends only on the power demand and the decision itself. Thus, the Utility node has 24 values corresponding to all the combinations between 8 values of *MWdem* and 3 values of Decision. The utility function considers the cost of taking a decision and the gain obtained if this decision is made in the current situation. For example, if more power is demanded, the associated cost of stopping the turbine for an off-line revision is much higher and the benefit may not be high since full power was demanded. On the contrary, if there was a fault detected and the decision is to make a major revision, then the cost is high but the benefit is also very high. In this case, do nothing can be cheap but the benefits can be negative, e.g., a major malfunctioning of the turbine.

Therefore, the utility function defined in these experiments consists on an equation depending on the parents of the utility node, i.e., the decision made and the demand of power. The equation relates the cost of the decision made and the gain obtained. Experiments are being carried out to find the best equation that produces the optimal results according to controlled scenarios.

3.2.1. Experiments

Several tests were made with different discretization values in order to see the best result in the learning process. Table 5 shows the results obtained when the program carries out ten different cycles of execution (at random).

Table 5. Results obtained in ten executions of the prototype

MWdem	Vgas	MW	TTXD1	TTXD2	TTXD3	Decision		
						Off − line	In − line	nothing
16 (5)	38 (2)	22 (7)	353 (5)	331 (5)	503 (8)	10.0	11.0	5.0
14 (4)	41 (2)	12 (3)	370 (6)	224 (2)	454 (7)	8.0	9.0	4.0
15 (4)	59 (5)	13 (4)	217 (2)	156 (1)	433 (7)	8.0	9.0	4.0
17 (5)	77 (9)	12 (3)	183 (1)	401 (6)	435 (7)	10.0	11.0	5.0
9 (2)	47 (3)	4 (1)	302 (4)	525 (9)	234 (3)	4.0	5.0	2.0
15 (4)	65 (7)	12 (3)	424 (7)	107 (0)	510 (9)	8.0	9.0	4.0
2 (0)	48 (4)	19 (6)	141 (1)	392 (6)	180 (1)	0.0	1.0	0.0
2 (0)	47 (3)	18 (5)	214 (2)	268 (3)	325 (5)	0.0	1.0	0.0
11 (3)	60 (6)	8 (2)	384 (6)	453 (7)	417 (7)	6.0	7.0	3.0
23 (7)	69 (7)	16 (5)	284 (4)	376 (6)	417 (7)	0.0	15.0	0.0

The first six columns describe the current value of the corresponding variables and their assigned interval. The value at the left is the real value read by the system, and the value in parenthesis represents the number of discretization interval corresponding to the value. For example, in the first cell, variable *MWdem* has a value of 16

MW that corresponds to the interval 5 in the discretization. The last three columns correspond to the expected utility values for each possible decision. For example, in the first row, the best decision to recommend is the execution of an in-line revision (value of 11), and the worst decision is to do nothing (value of 5). The rest of the rows also decide the in-line revision but with different levels of intensity. For example, the last row suggests in-line revision with 15 to 0, in contrast to 11 to 10 units in the first row. In the last row case, in-line revision is mandatory.

These results of course depend on the knowledge captured by the experts in two aspects. First, the structure of the influence diagram, and second the numerical parameters required in the reasoning. They are the a-priori and conditional probabilities of the chance nodes, and the utility values stored in the utility node. These ten runs of the prototype demonstrated the advantage of performing a state detection of the system and the in-line generation of the optimal action recommendation in the current situation. All the decisions suggested by the system were discussed with experts who analyzed the current situation and the suggestion. All they agree in the accuracy and opportunity of the system recommendation.

4 Planning system

A planning system automatically chooses and organizes actions, by anticipating their expected outcomes, to achieve some pre-established goals. In the case of the Markov Decision Process formalism, the planning system assumes that the outcome of an action is probabilistic and the goals are specified as utility/cost functions. Our planning system, in particular, provides to power plant operators with a set of suggested actions or operational recommendations to optimize safety and power generation. Every time a recommendation is executed, the plant state process changes into a new state which directly maps to another recommendation. The sequential concatenation of these recommendations produces an optimal action plan that manages the process behavior to achieve a desired goal state. In this section, we will set an example of a planning problem specification with the MDP methodology, and we will show the results obtained in a thermal power plant simulator using a planning tool for learning and solving factored MDPs.

4.1. Problem specification

The set of states in a power plant domain are directly obtained from the relevant operation state variables, and discretized according to the abstraction and refinement method detailed in [12]. The set of actions corresponds to the commands that the operator can decide to execute, e.g. open/close a valve or start/stop a pump. The reward function for the MDP is based on the optimal operation response, for example, one could optimize safety, utility or risk. The transition function specifies a probability distribution over the effects on the process after a command execution. The planning problem is to find a policy function that maps states to recommended actions to optimize an expected utility function.

The planning system is based on SPI ("Planning under Uncertainty System" in Spanish) a computational tool that integrates capabilities for model approximation from data, e.g. transition and reward functions, and dynamic programming methods for solving MDPS such as value iteration or policy iteration algorithms. In its general form the SPI's learning module builds a decision model from simulation data to produce a problem specification. The problem specification is then entered to the SPI's dynamic programming module for finding an optimal policy that can later be queried. For complex problem specifications (more than 10,000 states), we use the SPUDD system [7], which includes a more powerful version of the value iteration algorithm. The SPUDD's power relies on its way to represent transition, value and policy functions using algebraic decision diagrams. Since transition functions are represented more compactly as two-stage Bayesian networks, SPI uses Elvira [5] to represent, infer and learn transition functions. Similarly, since reward functions can be represented in its factored form as decision trees, SPI uses Weka [15] to approximate a reward model from data in the case of a non-structured reward function.

4.2. Experiments

The planning system was tested in the steam generation subsystem of a thermal power plant simulator. The idea is to obtain a control strategy that considers stochastic commands on the valves and, according to an experience-based preference function, maximizes the security in the drum, and/or the power generation.

For a first set of experiments, we specified a 5-action problem with 5 hybrid variables (F_{ms}, F_{fw}, P_d, g, d). We also defined a simple binary preference function that rewarded safety parameters in the drum, P_d and F_{ms}. The relationship between their values and the reward received can be seen in figure 3 (left). Central black squares denote safe states (desired operation regions), and white zones represent non-rewarded zones (indifferent regions). To learn the model and the initial abstraction, samples of the system dynamics were gathered using simulation. Black dots in figure 3 (right) represent sampled states with positive reward, red dots have no reward, and white zones were simply not explored. Figure 3 (left) shows the state partition and policy found (green arrows) by the learning system. For this simple example, although the resulting policy is not very detailed (states are quite large), it follows the idea of going to the lower black regions. When analyzed by an expert operator, this control strategy is near optimal in most of the abstract states. We solved the same problem but adding two extra variables, the position for valves *msv* and *fwv*, and using 9 actions (all the combinations of open-close valves *msv* and *fwv*). We also redefined the reward function to maximize power generation, g, under safe conditions in the drum. Although the problem increased significantly in complexity, the policy obtained is "smoother" than the 5-action simple version presented above. To give an idea about the computational saving, for a fine discretization (15,200 discrete states) this problem was solved in 859.2350 seconds, while our abstract representation (40 qstates) took only 14.2970 seconds. In both cases, the approximated models were found using the SPUDD system [7].
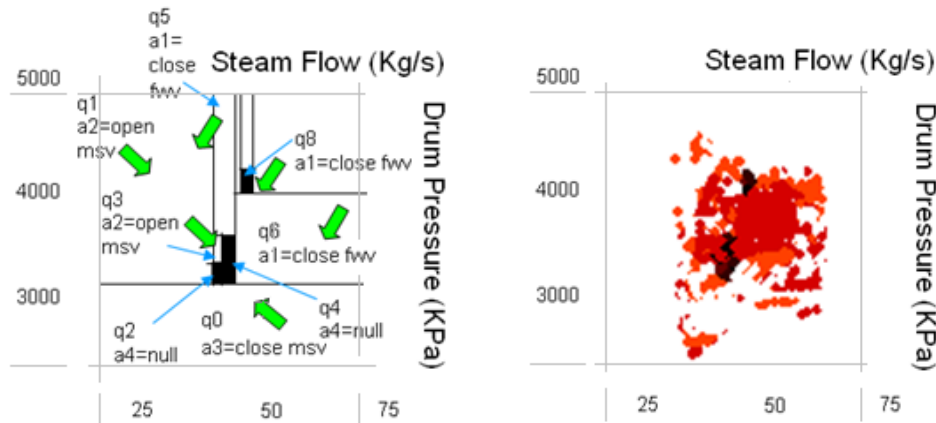

Fig. 3. Process control problem. Left: reward distribution, state partition and policy found. Right: exploration trace

5 Conclusions

In this paper, we showed three different intelligent systems using probabilistic techniques for diagnosis, decision support and planning tasks in a thermal power plant.

The first system is based on a probabilistic model over qualitative changes of the variables in the application. The system demonstrates how a probabilistic diagnosis can be carried out utilizing common expert criteria over signals increasing their value, decreasing or staying. The model was evaluated in a simulated environment for gas turbines of a 350 MW thermal power plant. The second system utilizes influence diagrams that use probabilistic reasoning and decision theory techniques. The system demonstrates that the use of decision theory together with probabilistic modeling provide more robust diagnosis capabilities. Experiments on a gas turbine simulator have shown the feasibility for the use of this technique in real applications. Finally, the planning system uses the formalism of Markov Decision Processes (MDPs) which provide a powerful framework for solving sequential decision problems under uncertainty. Given a problem specification, the objective of the MDP is to obtain the optimal policy for getting the plant to a state under optimal operation. The experiments, developed in a combined cycle power plant simulator, demonstrated that probabilistic automated planning can solve complex problems successfully.

We consider that the contributions of this work can be both scientific and technological. The scientific contribution consists in that our work relies on the use of probabilistic techniques applied to real-world domains. Technologically, we contribute to the safe and economic operation of power plants.

In the future, we plan to integrate these diagnoses and planning systems in the form of an intelligent support suite (ISS) to aid human operators of real thermal power plants. This ISS will be used to assist in the early diagnosis, prediction and management of faults and disturbances that could lead the plant to a shutdown condition.

Acknowledgments

References

1. **C. Boutilier, T. Dean, and S. Hanks**. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
2. **M. Caimi, C. Lanza, and B. Ruiz-Ruiz**. An assistant for simulator-based training of plant operator. *Marie Curie Fellowships Annals*, Vol. 1, 1999.
3. **G.F. Cooper and E. Herskovitz**. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–348, 1992.
4. **R. E. Bellman.** *Dynamic Programming*. Princeton University Press, Princeton, N.J., U.S.A., 1957.
5. **The Elvira Consortium**. Elvira: An environment for creating and using probabilistic graphical models. In *Proceedings of the First European Workshop on Probabilistic graphical models (PGM'02)"*, pages 1–11, Cuenca, Spain, 2002.
6. **Z. Flores, P.H. Ibargüengoytia, and E. Morales**. On-line diagnosis of gas turbines using probabilistic and qualitative reasoning. In *th Intl. Conf.on Intelligent Systems Application to Power Systems, ISAP-2005*, Washington, D.C., U.S.A., 2005.
7. **J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. Spudd**: Stochastic planning using decision diagrams. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, UAI-99*, pages 279–288, 1999.
8. **C. Lacave, M. Luque, and F. J. Díez**. Explanation of Bayesian networks and influence diagrams in Elvira. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, 37:952–965, 2007. In press.
9. **B. Morales and P.H. Ibargüengoytia**. On-line diagnosis using influence diagrams. In L.E. Sucar R. Monroy, G. Arroyo and H. Sossa, editors, *Advances in Artificial Intelligence - MICAI 2004, LNAI 2313*, pages 546–554, Berlin Heidelberg, 2004. Springer-Verlag.
10. **J. Pearl**. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco, CA., 1988.
11. **M.L. Puterman**. *Markov Decision Processes: Discrete stochastic dynamic programming*. Wiley, New York, N.Y., U.S.A., 1995.
12. **A. Reyes, L.E. Sucar, and P.H. Ibargüengoytia**. Abstraction and refinement for solving markov decision processes. In *Proceedings of the European Workshop on Probabilistic Graphical Models (PGM'06)"*, pages 263–270, Chezch Republic, 2006.
13. **D. Suc and I. Bratko**. Qualitative reverse engineering. In *Proceedings of the 19th International Conference on Machine Learning*, 2000.
14. **Z.A. Vale, C. Ramos, A. Silva, L. Faria, J. Santos, F. Fernandez, C. Rosado, and A. Marques.** Socrates an integrated intelligent system for power system control center operator assistance and training. In *IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 27–30, Cancun, México, 1998.
15. **I.H. Witten and E. Frank.** *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, CA, U.S.A., 2nd edition edition, 2005.

Pablo H. Ibargüengoytia *is a researcher at the Electrical Research Institute in México (IIE) and part-time professor at Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) campus Cuernavaca. His research interests include probabilistic reasoning, machine learning, and their applications in the electric industry. He received a PhD in Computer Science from University of Salford in U.K., and M Sc. Degree from University of Minnesota in U.S.A. and a B.Sc. degree in Electronic Engineering from Univeridad Autonoma Metropolitana, in Mexico City He is a member of the National Research System in Mexico.*

Alberto Reyes *is a researcher at the Electrical Research Institute in México (IIE) and part-time professor at Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) campus México City. His research interests include decision-theoretic planning, machine learning, and their applications in robotics and intelligent assistants for industry. He received a PhD in Computer Science from ITESM campus Cuernavaca.*

Zenón Flores Loredo. *B. Sc. in Electronics and Communications Engineering by the University of Veracruz campus Poza Rica-Tuxpan (1992) and M. Sc. in Computer Science (Artificial Intelligence) by the Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Cuernavaca (2004). From 1992 to date he has been a Research Engineer of the Instituto de Investigaciones Eléctricas where he has been participating at design, development and commissioning of acquisition and control systems for power plants of CFE.*