Complexity of Decision-Theoretic Troubleshooting

Marta Vomlelová
Department of Computer Science, Aalborg University, Denmark

The goal of troubleshooting is to find an optimal solution strategy consisting of actions and observations for repairing a device. We assume a probabilistic model of dependence between possible faults, actions, and observations; the goal is to minimize the expected cost of repair (ECR). We show that the task of finding an optimal solution strategy is NP hard for various troubleshooting models; therefore, approximate algorithms are necessary. © 2003 Wiley Periodicals, Inc.

"Ridge Hall, computer assistance; may I help you?"

"Yes, well I'm having trouble with WordPerfect."

"What sort of trouble?"

"Well I was just typing along and all of a sudden the words went away."

"Hmmm. So what does your screen look like now?"

"It's blank; it won't accept anything when I type."

"Are you still in WordPerfect?"

"How do I tell?"

"Can you move your cursor around your screen?"

"There isn't any cursor: I told you it won't accept anything I type."

"Does your monitor have a power indicator?"

"I don't know."

"Well then look at the back of the monitor and find where the power cord goes into it. Can you see that?"

"No."

Author to whom all correspondence should be addressed. e-mail: marta@cs.auc.dk

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 18, 267–277 (2003)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.10087

"Even if maybe you put your knee on something and lean way over?"

"Oh it's not because I don't have the right angle it's because it's dark."

"Dark?"

"Yes. The office light is off and the only light I have is coming in from the window."

"Well turn the office light on then."

"I can't."

"No? Why not?"

"Because there's a power failure."

This is an abbreviated version of a text from the web.[1]

1. INTRODUCTION

The preface illustrates the basics of a troubleshooting problem: we face a failure of a man-made device; we try out different actions and observations until we fix the problem.

The basic probabilistic troubleshooting model was introduced by Kalagnanam and Henrion.[2]

1.1. Basic Problem Definition

We have m repair actions $A_1, \ldots, A_m$ that may fix the problem. One and only one of these actions will fix the problem if it is performed. The probability that action A_i fixes the problem is p_{A_i}; we have to pay cost c_{A_i} for performing it. The cost may be interpreted as the time spent when performing the action. One action is performed after another until the device is repaired. The goal of the troubleshooting task is to find an ordering of actions that minimizes expected cost of repair (ECR) where

$$\begin{aligned}\mathrm{ECR}(A_1, \ldots, A_n) &= c_{A_1} + (1 - p_{A_1}) \cdot c_{A_2} + (1 - p_{A_1} - p_{A_2}) \cdot c_{A_3} + \cdots + p_{A_n} \cdot c_{A_n} \\ &= \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{i-1} p_{A_j} \right) \cdot c_{A_i}\end{aligned}$$

The formula may be read, we first perform action A_1, with cost c_{A_1}; then, with probability $1 - p_{A_1}$ we find that action A_1 has not fixed the problem and we perform action A_2, with cost c_{A_2}, etc.

We also use an equivalent formula with the order of summation changed:

$$\mathrm{ECR}(A_1, \ldots, A_n) = \sum_{i=1}^{n} p_{A_i} \cdot \sum_{j=1}^{i} c_{A_j}$$

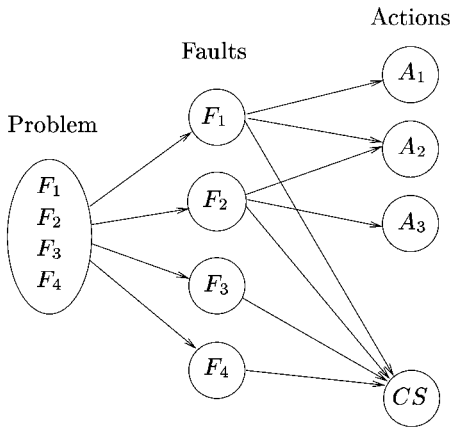

Figure 1. Bayesian network model for a troubleshooting problem with dependent actions.

This formula may be read, with the probability p_{A_i} we fix the problem in the ith step. To reach the ith step we have to perform all actions $1 \ldots (i-1)$ first, i.e., the total cost when fixing the problem in the ith step is $\sum_{j=1}^{i} c_{A_j}$.

1.2. Extended Problems

This basic model is extended in additional works.

(1) The costs may depend on previous actions.[3] The costs may be random; risk-sensitive troubleshooting[4] is modeled by costs with exponential distribution (independent of the stage of troubleshooting).
(2) There may be a distinction between causes of failure and actions repairing the faults.[5] Generally, an action may repair more than one fault and a fault may be repaired by more than one action (See Figure 1). The basic model assumes one-to-one correspondence.
(3) Most of the works accept *single fault assumption,* i.e., they assume that exactly one fault is present.[2,5,6–8] An alternative assumption is the independence of faults.[6] Generally, we assume a general relationship among faults represented by a Bayesian network.
(4) We may use observations[5,7,8] that do not fix the problem but provide additional information about possible faults in the system.

We do not allow any action or observation to be repeated; we assume the results to be persistent in time. This assumption may be relaxed;[9] e.g., a change in configuration may cause actions to produce different results if they are repeated.

1.3. Complexity of Algorithms

The simple model with *single fault assumption*, fixed costs, and no observations is known to be polynomial (a sequence ordered according to decreasing p_i/c_i ratio is an optimal one). Similar results hold for the basic model with independent faults[6]

{ordered according to $p_i/[c_i \cdot (1 - p_i)]$} and random costs with exponential distribution[4] {in brief, ordered according $p_i/[E(c_i)]$ if these ratios are distinct}.

Exact algorithms for solving troubleshooting with dependent actions and observations are based on dynamic programming[10] or the AO^* algorithm[11] and need (in general) exponential time.

An approximate algorithm introduced by Heckerman et al.[7,9] updates the $[p_i(\mathbf{e})]/c_i$ after any step where we increase our evidence $\mathbf{e}$ and introduces the expected cost of an observation (ECO) to be able to compare a sequence of actions with a strategy starting with observation O and continuing differently according to the result of O. This algorithm uses propagation in a Bayesian network that is not polynomial; the rest of the algorithm is polynomial.

Skaanning et al.[5,8] simplified the model to naive Bayes that allows polynomial propagation. A function p_{QidC} was introduced to allow sorting of observations into p_i/c_i sequence and ECO_2 was introduced to postpone an observation that is useful but may be performed later. These computations may be performed in polynomial time.

The question is whether an exact polynomial algorithm exists. In this study, we prove that any of these four extensions numbered previously leads to an NP-hard problem. This is a part of my Ph.D thesis.[12]

2. DEPENDENT COSTS

Suppose you are using a network printer and the requested page did not print out. The problem may be in your computer, in the network, or in the printer. If you are sitting by your computer, it may take couple of minutes to go and check the printer. On the other hand, if you are standing near the printer, all actions related to the printer are less expensive. We extend the basic model (Section 1.1) allowing the cost of action A_i depending on the previously performed action A_j. We assume the cost of A_i after A_j [denoted $c_{A_i}(A_j)$] to be independent of all actions before A_j.

Even this simplified version is NP hard.

THEOREM 1. *Assume a troubleshooting problem, with costs dependent on the last performed action, and the single fault assumption. Finding an optimal troubleshooting sequence is an NP-hard problem.*

Proof. We reduce the traveling repairmen problem (TRP) to the troubleshooting task with dependent costs.

Imagine a traveling repairmen who knows that in one of n cities is a broken device he should repair. The only way to find it is to visit one city after another until the device is found; then he repairs it and his job is finished. The goal is to find such ordering π of visiting cities that minimizes the expected length of travel. In the following exact definition, i is the order of the city where the broken device is found ($i = 1$ is the starting point); the inner sum calculates the distance traveled from the starting point to the broken device in city i following the ordering π.

> The TRP: Assume n points and a metric d defining the distance between any pair of points. We seek a permutation π of points minimizing

$$\min_{\pi} \sum_{i=2}^{n} \sum_{j=1}^{i-1} d_{\pi(j),\pi(j+1)}$$

where $\pi(1) = r$ is a fixed starting point.[13,14]

We transform the TRP to troubleshooting as follows.

For every nonstarting point in the TRP we define one repair action, all with the same probability of success $p_{A_i} = 1/(n-1)$, $i = 2, \ldots, n$. Symbol A_1 represents the starting point with no action performed. The cost of each action A_i, $i = 2, \ldots, n$, depends on the previously performed action A_j, $j = 1, \ldots, n$, and it is defined as $c_{A_i}(A_j) = d_{j,i}$. Now, an optimal troubleshooting sequence is equivalent to a best traveling repairman path, because for any troubleshooting sequence $A_{i_2}, \ldots, A_{i_n}$ (keeping the starting point fixed, i.e., $A_{i_1} = A_1$),

$$\mathrm{ECR}(A_{i_2}, \ldots, A_i n) = \sum_{j=2}^{n} p_{A_{i_j}} \cdot \sum_{r=1}^{j-1} c_{A_{i_{(r+1)}}}(A_{i_r}) = \frac{1}{n-1} \cdot \sum_{j=2}^{n} \sum_{r=1}^{j-1} d_{i_r, i_{(r+1)}}$$

so the ECR is equal to the $1/(n-1)$ times the traveling repairman cost function. ■

Dependent costs make the troubleshooting problem NP hard. In following sections, we will assume constant costs and relax other assumptions in the basic problem.

3. DEPENDENT ACTIONS, DEPENDENT FAULTS

Suppose a printer prints a page that is too light. You may try A_1, `remove, shake, and replace toner cartridge`, or A_2, `replace toner cartridge` (or lots of other actions). Both actions A_1 and A_2 address the same device fault F_1, `toner distribution problem`, but with a different probability of success $p(A_1 = \text{yes} | F_1 = \text{yes}) = 0.5$ and $p(A_2 = \text{yes} | F_1 = \text{yes}) = 0.99$. This may be represented by a simple[5] Bayesian network describing the dependence among faults F_i and actions A_j (See Figure 1). There are specified prior probabilities of faults $p(F_i)$; conditional probabilities of actions $p(A_i = \text{yes} | \text{pa}(A_i))$, where $\text{pa}(A_i)$ denotes parents of node A_i; and costs of actions c_{A_j}.

Because more than one action may address the same fault, we may realize an action is useless even before we tried it out. Assume the simple model in Figure 1 with $p(A_1 = \text{yes} | F_1 = \text{yes}) = 1$, $p(A_2 = \text{yes} | F_1 = \text{yes}) = 0.5$, and $p(A_3 = \text{yes} | F_2 = \text{yes}) = 0.5$. After A_1 failed we do not have to try A_2 because we are sure fault F_1 would have been fixed by action A_1. Therefore, fault F_2 must be present.

To be able to define the ECR, we introduce the call service action CS that always fixes the problem, in the worst case, by replacing the printer. The cost of CS is assumed to be so high that all other useful actions are tried first. For our purposes, $c_{\mathrm{CS}} > \{1/[\min_i p(F_i)^2]\} \cdot \max_i c_i$ fulfills this criterion.

Then, the ECR is defined only for sequences that guarantee successful repair

$$\mathrm{ECR}(A_1, \ldots, A_n) = \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{i-1} p(A_j = \text{yes} | A_1 = \text{no}, \ldots, A_{j-1} = \text{no}) \right) \cdot c_{A_i}$$

Finding an optimal troubleshooting strategy for this model is an NP-hard problem.

THEOREM 2. *Assume a troubleshooting problem with dependent actions, the single faults assumption, and a constant* $K \in \Re^+$. *The decision whether there exists a troubleshooting sequence* $\mathbf{s}$ *with* $ECR(\mathbf{s}) \leq K$ *is an NP-complete problem.*[11]

The proof[11] is based on reduction of exact cover by three sets of troubleshooting. If we assume that each action fixes exactly one fault but the faults are dependent and these dependence may be represented by a Bayesian network; then, we again get an NP-hard problem.

THEOREM 3. *Assume a troubleshooting problem with fixed costs, any action addressing only one fault, and a Bayesian network model of dependency among faults. Finding an optimal sequence is an NP-hard problem.*

Proof. We reduce the troubleshooting model from Theorem 2 (troubleshooting with dependent actions) to the troubleshooting with dependent faults. The reduction is represented in Figure 2.

We start with the Bayesian model of the troubleshooting task with dependent actions. All actions in the old model become faults in the new model so we have changed the labels from A_i to X_i. For every node X_i, we construct a new action node A_i with conditional probabilities: $p(A_i = \text{yes} | X_i = \text{yes}) = 1$ and $p(A_i = \text{yes} | X_i = \text{no}) = 0$; therefore, $p(A_i = \text{yes})$ if and only if $p(X_i = \text{yes})$. A sequence $A_{j_1}, \ldots, A_{j_m}$ minimizes ECR in the new problem if and only if the corresponding sequence $X_{j_1}, \ldots, X_{j_m}$ minimizes the original problem. ■

4. OBSERVATIONS

Often, it is useful to make observations that do not fix the problem but can bring more information about the possible causes of the problem. For example, if you print a status page on a printer, you can distinguish between a software problem (the status page is printed properly) and a printer problem (even the status page is printed badly). Troubleshooting proceeds differently, dependent on the outcome of the observation; therefore, the troubleshooting strategy is no longer a sequence but a tree, as in Figure 3.

The ECR is defined as a weighted average of total costs of reaching leaves of the strategy tree

$$\mathrm{ECR}(\mathbf{s}) = \sum_{l \in \mathcal{L}(\mathbf{s})} p(\mathbf{e}_l) \cdot t(l)$$

where $p(\mathbf{e}_l)$ denotes the probability of reaching the leaf l and $t(l)$ denotes the total cost of all actions and observations on the path from the root to the leaf l.

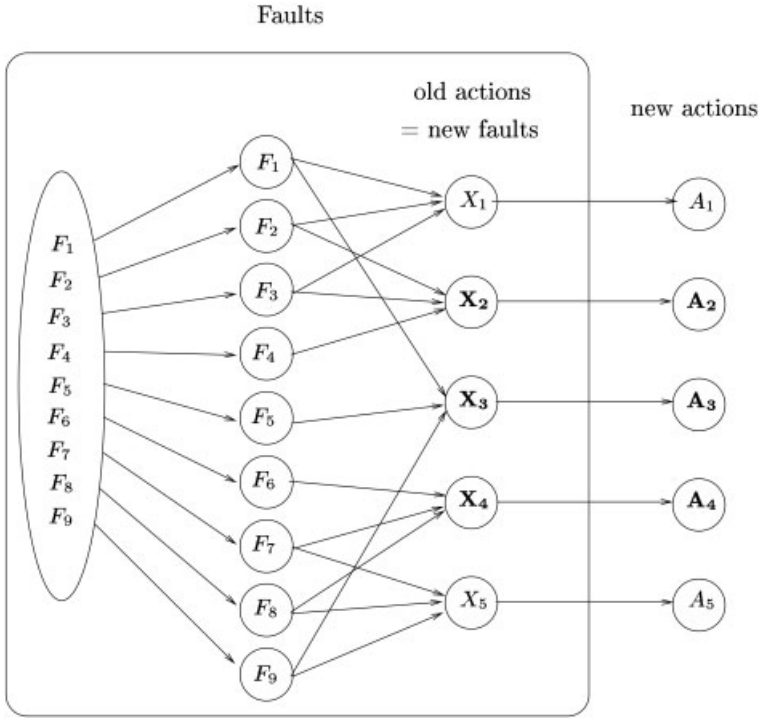

Figure 2. A reduction of troubleshooting with dependent actions to troubleshooting with dependent faults.

THEOREM 4. *Assume a troubleshooting problem with fixed costs, independent actions, observations, and the single fault assumption. Finding an optimal troubleshooting strategy is an NP-hard problem.*

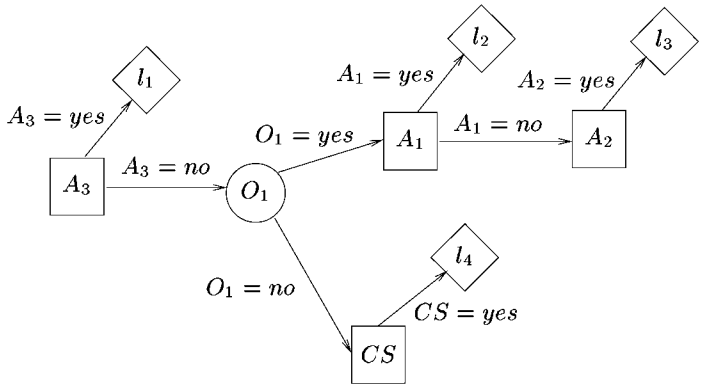

Figure 3. Strategy tree.

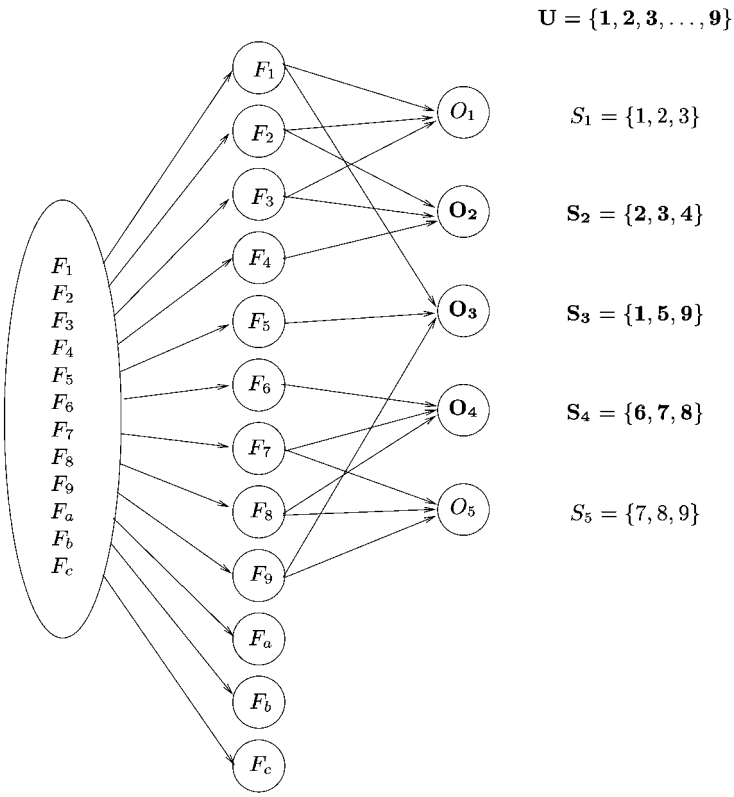

Figure 4. A troubleshooting model for an exact three-sets cover problem. Every observation corresponds to one set. The boldface sets compose the exact cover; observations corresponding to the exact cover form a best troubleshooting strategy (Figure 5).

Proof. To prove NP hardness of troubleshooting with observations, we reduce the exact cover by three-sets to this type of troubleshooting.

Assume we have n mailing lists $\{S_1, \ldots, S_n\}$ and their union is denoted U; $U = \cup_{i=1}^{n} S_i$. The goal is to send an e-mail to some mailing lists so that each person in U gets the e-mail exactly once. Even a simplified version, in which every mailing list S_i contains exactly three persons, is known to be NP complete.

> Exact cover by three-sets: We are given a family $\mathcal{S} = \{S_1, \ldots, S_n\}$ of subsets of a set U, such that $|U| = 3m$ for some integer m and $|S_i| = 3$ for all i. We are asked if there are m sets in $\mathcal{S}$ that are disjoint and have U as their union.[15]

The reduction is illustrated in Figure 4. Assume an exact cover by three-sets task with a family $\mathcal{S}$ on the universe U. In the troubleshooting model we introduce a fault for each element in U and we add three extra faults.

For each set $S_i \in \mathcal{S}$ we construct an observation O_i having answer $O_i =$ yes for the three faults corresponding to the three elements of S_i, otherwise having

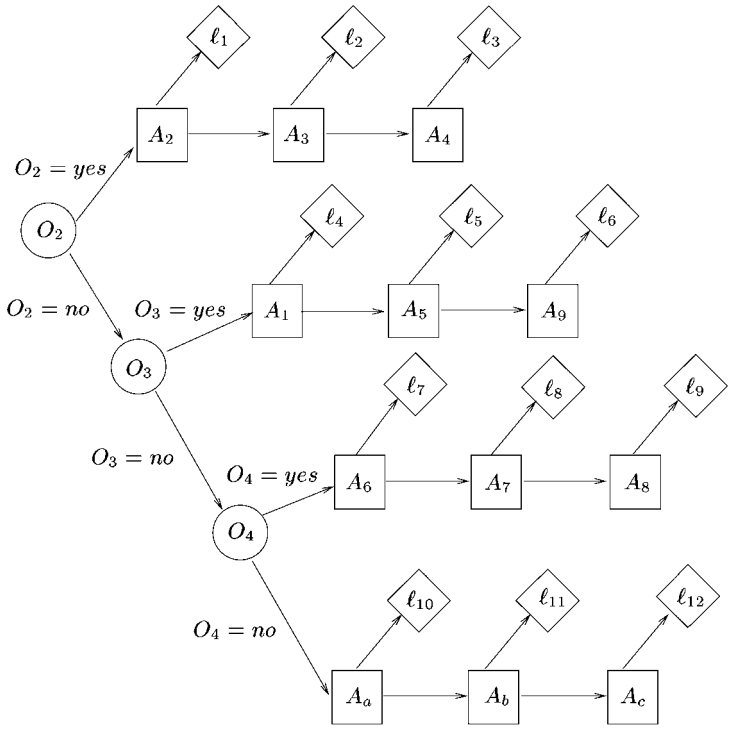

Figure 5. The optimal troubleshooting strategy of the troubleshooting problem in Figure 4 corresponding to the cover $\{S_2, S_3, S_4\}$.

answer O_i = no. For each fault we construct one action that fixes this fault with probability 1 and does not fix any other fault. All faults are equally probable; all actions and observations have cost 1.

We aim to prove that the exact cover the three sets exists iff a troubleshooting strategy s with ECR $\leq 2 + \{[m(m + 3)]/[2(m + 1)]\}$ exists. If there exists an exact cover by three sets [e.g., (S_2, S_3, S_4)] then we construct a troubleshooting strategy such as that shown in Figure 5. First, we perform observations until we get a positive answer O_i = yes. If we have a positive answer or no more observations are left, we know that one of the three faults is present. Then, we perform the three actions corresponding to those three faults (in any order).

No strategy with ECR $< 2 + \{[m(m + 3)]/[2(m + 1)]\}$ exists, and the equality holds only for strategies corresponding to an exact cover by three sets.

■

5. CONCLUSIONS

The basic troubleshooting task is polynomial.[2,6] However, if we extend it in more realistic ways, we get NP-hard problems.

The algorithms used to solve troubleshooting problems are reviewed in Section 1. Algorithms finding an optimal solution are based on dynamic programming[10] or the AO^* algorithm[11] and they appear to be prohibitively slow for realistic models. That leads to a search for approximate algorithms.

Approximate algorithms were tested in the automobile domain[7] and printing domain[5,7,8] and they performed surprisingly well. The troubleshooting methodology may be applied in many other fields where we start to act when evidence of a problem appears (e.g., medical diagnosis or software or hardware failure) or where we face a task with probabilistic results of actions (e.g., best purchase decision). If current approximate algorithms do not perform well enough in new areas, general algorithms for NP-hard problems[16] may be tried.

Acknowledgments

The author thanks F. V. Jensen for inspiring her to study the complexity of troubleshooting and the anonymous reviewer for helpful comments.

References

1. http://www.catholic-pages.com/forum/.
2. Kalagnanam J, Henrion M. A comparison of decision analysis and expert rules for sequential diagnosis. In: Proc 4th Conf on Uncertainty in Artificial Intelligence (1988). New York: Elsevier Science Publishing Company, Inc.; 1990. pp 271–281.
3. Langseth H, Jensen FV. Heuristics for two extensions of basic troubleshooting. In: 7th Scandinavian Conf on Artificial Intelligence (SCAI '01). Amsterdam: IOS Press; 2001. pp 80–89.
4. Shayman MA, Fernández-Gaucherand E. Risk-sensitive decision-theoretic diagnosis. IEEE Trans Automatic Control 2001;46(7):1166–1171.
5. Skaanning C, Jensen FV, Kjærulff U. Printer troubleshooting using Bayesian networks. In: Proc 13th Int Conf Industrial and Engineering Applications of AI and Expert Systems (IEA/AIE-2000). Holiday Inn Downtown, New Orleans, LA, June 19–22, 2000. Lecture Notes in Computer Science. Berlin: Springer-Verlag. pp. 367–379.
6. Srinivas S. A polynomial algorithm for computing the optimal repair strategy in a system with independent component failures. In: Proc 11th Conf Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann Publishers; 1995. pp 507–514.
7. Heckerman D, Breese J, Rommelse K. Decision-theoretic troubleshooting. Commun ACM 1995;38:49–57.
8. Jensen FV, Kjærulff U, Kristiansen B, Langseth H, Skaanning C, Vomlel J, Vomlelová, M. The SACSO methodology for troubleshooting complex systems. Artif Intell Eng Design, Anal Manufact 2001;15:321–333.
9. Breese JS, Heckerman D. Decision-theoretic troubleshooting: A framework for repair and experiment. In: 12th Annu Conf Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann Publishers; 1996. pp 124–132.
10. Huard J-F, Lazar AA. Fault isolation based on decision-theoretic troubleshooting. Technical Report CU/CTR TR 442-96-08, Center for Telecommunications Research, Columbia University, New York, NY, 1996.
11. Sochorová M, Vomlel J. Troubleshooting: NP-hardness and solution methods. In: Proc 5th Workshop on Uncertainty Processing, WUPES'2000. Czech Republic: Jindřichuv Hradec; 2000.

12. Vomlelová M. Decision theoretic troubleshooting. PhD thesis, University of Economics, Prague, 2001.
13. Afrati F, Cosmadakis S, Papadimitriou C, Papagerrgiou G, Papakostantinou N. The complexity of the traveling repairman problem. Inform Theor Applications 1986;20(1): 79–87.
14. Koutsoupias E, Papadimitriou C, Yannakakis M. Searching a fixed graph. Lect Notes Comput Sci 1996;1099:280–289.
15. Papadimitriou CH. Computational complexity. Reading, MA: Addison-Wesley Publishing Co.; 1994.
16. Ausiello G, Grescenzi P, Gambosi G, Kann V, Marchetti-Spaccamela A, Protasi M. Complexity and approximation. Berlin: Springer; 1999.