Decision-Theoretic Troubleshooting

Hardness of Approximation

Václav Lín

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

Faculty of Management
University of Economics, Prague

PGM, 2012

Outline

Early Formulations of the Problem

. . . military applications in 1950's:

Early Formulations of the Problem

. . . military applications in 1950's:

S.M. Johnson, *Optimal Sequential Testing*, RAND Corporation (1956).

> *A problem of importance to the Air Force is that of troubleshooting to find a malfunctioning part of a complex piece of electronic equipment. ... A complicated machine may break down because of the failure of some of its components. In what sequence should its components be tested and repaired in order to minimize the expected delay time?*

Early Formulations of the Problem

. . . military applications in 1950's:

S.M. Johnson, *Optimal Sequential Testing*, RAND Corporation (1956).

> *A problem of importance to the Air Force is that of troubleshooting to find a malfunctioning part of a complex piece of electronic equipment. ... A complicated machine may break down because of the failure of some of its components. In what sequence should its components be tested and repaired in order to minimize the expected delay time?*

R. Bellman, *Dynamic Programming*, 1957.
B. Gluss, *An Optimum Policy for Detecting a Fault in a Complex System*, Operations Research, 1959.

Early Solutions – Dynamic Programming

R. Bellman - *Dynamic Programming* (1957):

> *A ball is in one of n boxes, with the a-priory probability p_k that it is in the k^{th} box. The time of examining the k^{th} box is c_k. What is the optimal policy to obtain the ball with minimum expected time?*

Early Solutions – Dynamic Programming

R. Bellman - *Dynamic Programming* (1957):

> *A ball is in one of n boxes, with the a-priory probability p_k that it is in the k^{th} box. The time of examining the k^{th} box is c_k. What is the optimal policy to obtain the ball with minimum expected time?*

The minimum expected time is

$$E(p_1, \ldots, p_n) = min_k\{c_k + (1 - p_k) \cdot E(p'_1, \ldots, p'_n)\},$$

where

$$\begin{aligned} p'_k &= 0, \text{ and} \\ p'_i &= p_i/(1 - p_k) \text{ for } i \neq k \ . \end{aligned}$$

Early Solutions – Dynamic Programming

R. Bellman - *Dynamic Programming* (1957):

> *A ball is in one of n boxes, with the a-priory probability p_k that it is in the k^{th} box. The time of examining the k^{th} box is c_k. What is the optimal policy to obtain the ball with minimum expected time?*

The minimum expected time is

$$E(p_1, \ldots, p_n) = min_k\{c_k + (1 - p_k) \cdot E(p'_1, \ldots, p'_n)\},$$

where

$$\begin{aligned} p'_k &= 0, \text{ and} \\ p'_i &= p_i/(1 - p_k) \text{ for } i \neq k\ . \end{aligned}$$

Solution – open the boxes according to the decreasing ratio p_k/c_k.

Troubleshooting with Bayesian Networks

D. Heckerman, J. S. Breese, K. Rommelse, *Decision Theoretic Troubleshooting*, CACM, 1995.

- printer troubleshooting (Microsoft research),
- usage of Bayesian networks,
- generation of diagnostic / repair plans.

Troubleshooting with Bayesian Networks

D. Heckerman, J. S. Breese, K. Rommelse, *Decision Theoretic Troubleshooting*, CACM, 1995.

- printer troubleshooting (Microsoft research),
- usage of Bayesian networks,
- generation of diagnostic / repair plans.

Bayesian networks:

- faults,
- repair actions,
- diagnostic actions,
- intermediate variables.

Troubleshooting with Bayesian Networks

D. Heckerman, J. S. Breese, K. Rommelse, *Decision Theoretic Troubleshooting*, CACM, 1995.

- printer troubleshooting (Microsoft research),
- usage of Bayesian networks,
- generation of diagnostic / repair plans.

Bayesian networks:

- faults,
- repair actions,
- diagnostic actions,
- intermediate variables.

Recent dissertations:

- Håkan Warnquist (2011).
- Thorsten J. Ottosen (2012).

Basic Troubleshooting – an Example.

You are printing a report but the colors come out very light ...

Basic Troubleshooting – an Example.

You are printing a report but the colors come out very light ...

Possible actions:

- restart the printer,
- change the print settings,
- re-seat the toner cartridge
- get a new cartridge altogether

These actions differ both in their difficulty and in the likelihood of fixing the print problem.

Basic Troubleshooting – an Example.

You are printing a report but the colors come out very light ...

Possible actions:

- restart the printer,
- change the print settings,
- re-seat the toner cartridge
- get a new cartridge altogether

These actions differ both in their difficulty and in the likelihood of fixing the print problem.

Our goal: sequence the available repair actions so that the expected cost of the repair is as low as possible.

Troubleshooting is hard

Already very simple scenarios are *NP*-hard.

- Marta Vomlelová, *Complexity of decision-theoretic troubleshooting*, International Journal of Intelligent Systems, 2003.

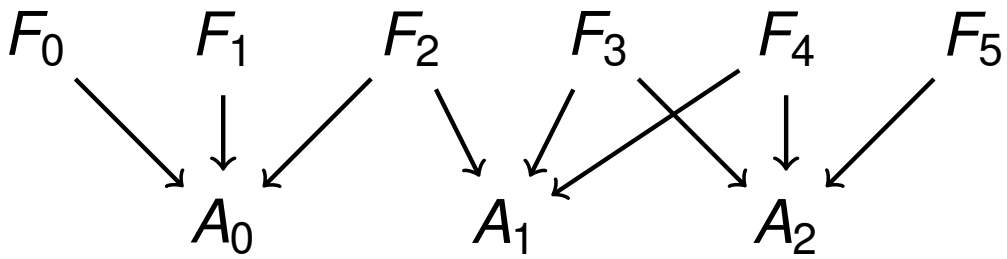
Troubleshooting is hard

Already very simple scenarios are *NP*-hard.

- Marta Vomlelová, *Complexity of decision-theoretic troubleshooting*, International Journal of Intelligent Systems, 2003.

The following is an instance of the Set Cover problem:

- Let $F_0, \ldots, F_5$ be faults, all equally likely,
- let A_0, A_1, A_2 be actions, all with the same cost.



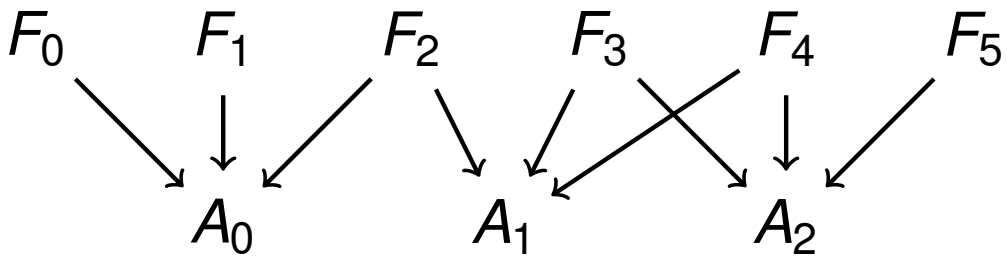
Troubleshooting is hard

Already very simple scenarios are *NP*-hard.

- Marta Vomlelová, *Complexity of decision-theoretic troubleshooting*, International Journal of Intelligent Systems, 2003.

The following is an instance of the Set Cover problem:

- Let $F_0, \ldots, F_5$ be faults, all equally likely,
- let A_0, A_1, A_2 be actions, all with the same cost.



Find the smallest subset of actions solving all the faults.

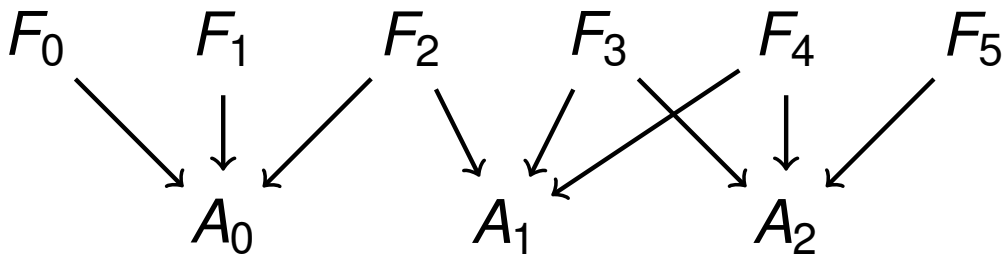
Troubleshooting is hard

Already very simple scenarios are *NP*-hard.

- Marta Vomlelová, *Complexity of decision-theoretic troubleshooting*, International Journal of Intelligent Systems, 2003.

The following is an instance of the Set Cover problem:

- Let $F_0, \ldots, F_5$ be faults, all equally likely,
- let A_0, A_1, A_2 be actions, all with the same cost.



Find the smallest subset of actions solving all the faults. $\longrightarrow \{A_0, A_2\}$

Hardness of Approximation

In the paper, we show that some troubleshooting problems are not only *NP*-hard, but also hard to approximate.

Hardness of Approximation

In the paper, we show that some troubleshooting problems are not only *NP*-hard, but also hard to approximate.

Algorithm A is a ρ-approximation algorithm ($\rho > 1$) for a minimization problem L if for all instances x of L we have

$$A(x) \leq \rho \cdot opt(x) \ ,$$

where

- $A(x)$ – the value returned by A when applied to instance x,
- $opt(x)$ – the optimum of x.

Hardness of Approximation

In the paper, we show that some troubleshooting problems are not only *NP*-hard, but also hard to approximate.

Algorithm A is a ρ-approximation algorithm ($\rho > 1$) for a minimization problem L if for all instances x of L we have

$$A(x) \leq \rho \cdot opt(x) \ ,$$

where

- $A(x)$ – the value returned by A when applied to instance x,
- $opt(x)$ – the optimum of x.

Hardness of approximation result shows that for certain ρ, there is no polynomial-time ρ-approximation algorithm for L unless P=NP.

Troubleshooting with Dependent Actions

Input:

- Random variable F with values (faults) $f_1, \ldots, f_n$.

Troubleshooting with Dependent Actions

Input:

- Random variable F with values (faults) $f_1, \ldots, f_n$.
- Set of actions $\{A_i\}$.

Each action fixes a subset of faults $F(A_i) \subseteq \{f_j\}$ with certainty and no other faults. Each action bears a constant cost $c(A_i)$.

Troubleshooting with Dependent Actions

Input:

- Random variable F with values (faults) $f_1, \ldots, f_n$.
- Set of actions $\{A_i\}$.

Each action fixes a subset of faults $F(A_i) \subseteq \{f_j\}$ with certainty and no other faults. Each action bears a constant cost $c(A_i)$.

Objective: Find a linear ordering π of actions minimizing the expected cost of repair:

$$ECR(A_{\pi(1)}, A_{\pi(2)}, \cdots) = \sum_i c(A_{\pi(i)}) \cdot \underbrace{P\Big(\bigwedge_{j<i} \{A_{\pi(j)} = \text{no}\}\Big)}_{\text{probability of reaching } A_{\pi(i)}}$$

Troubleshooting with Dependent Actions: Hardness of Approximation

Theorem

Unless P=NP, *troubleshooting with Dependent Actions has no polynomial-time* $(4-\epsilon)$*-approximation algorithm for any* $\epsilon > 0$.

Troubleshooting with Dependent Actions: Hardness of Approximation

Theorem

Unless P=NP*, troubleshooting with Dependent Actions has no polynomial-time* $(4-\epsilon)$*-approximation algorithm for any* $\epsilon > 0$.

Proof by reduction from Min-sum Set Cover.

- Feige, Lovász, Tetali, *Approximating Min-sum Set Cover*, Algorithmica, 2004.

Other results

Come see the poster!

Hardness of approximation for . . .

Other results

Come see the poster!

Hardness of approximation for . . .

- troubleshooting with multiple dependent faults,

Other results

Come see the poster!

Hardness of approximation for ...

- troubleshooting with multiple dependent faults,
- troubleshooting with cost clusters forming a DAG.

Summary

In the paper, we prove three new hardness of approximation results.

Summary

In the paper, we prove three new hardness of approximation results.

Outlook:

- Greedy algorithms
  - empirical evaluation,
  - performance guarantees for special cases,
  - worst case conditions – when do the algorithms perform badly?

Thank you

Q & A