Decision Theoretic Troubleshooting :

Cet article semble être une version “simplifié” de l’article “Troubleshooting under Uncertainity”, il a les mêmes auteurs et des parties égales. Donc je ne pense pas qu’il apporte beaucoup. Néanmoins il a un graphe intéressant que donne le pas-à-pas d’un programme itératif résolvant le problème avec les heuristiques présentés et qu’on peut utiliser comme base.

An Approximate Nonmyopic Computation for Value of Information :

Il traite d’une autre façon de calculer la “value of information” qu’on utilise pour décider l’ordre des réparations et observations faites. Il présente une manière de faire le calcul qui est approximative mais pas “myope”, en utilisant des propriétés des probabilités et taille de l’ensemble considéré (si j’ai bien compris ils utilisent le théorème centrale limite). Les maths semblent être lourds (j’avoue que je n’ai pas tout lu). C’est peut-être intéressant pour faire une amélioration du système qu’on va coder mais je ne sais pas si ça va bien avec l’approche voulue.

Causal Independence for Probability Assessment and Inference Using Bayesian Networks :

L’article se concentre sur comment marche la “causal independence” et comment on peut l’utiliser pour créer le réseau de Bayes, en simplifiant les probabilités et en faisant des inférences entre elles. Il utilise des idées qui sont présentes sur l’article “Elicitation of Factored Utilities” et que j’ai du mal à comprendre maintenant.

Decision-Theoretic Troubleshooting A Framework for Repair and Experiment :

Il semble parler des mêmes idées présentes sur “Troubleshooting under Uncertainty” mais avec une type d’action différente de ceux qu’on a vu jusqu’à maintenant : une action où on change la configuration du système et on observe son nouveau comportement, en utilisant cette observation pour améliorer notre connaissance du système (il faut que je le lise de façon plus attentive pour comprendre exactement ce qu’ils veulent dire avec “changer la configuration”). Ils décrivent la notion de “persistance” et montrent comment on peut l’utiliser pour trouver une solution. L’idée principale de l’article semble être que les solutions qu’on a vues jusqu’à maintenant ont un problème : quand on fait une réparation on doit “jeter” les observations qui ne sont plus à jour car elles dépendent du composant qui vient d’être modifié. La “persistance” qu’ils présentent semble être une façon de profiter des observations sans les recalculer entièrement.

Decision-Theoretic troubleshooting: Hardness of approximation :

Cet article ne parle pas de comment faire du troubleshooting mais quels sont les complexités pour les techniques existantes. Il montre que la solution exacte est NP-complet et les approximations avec des heuristiques sont NP-difficiles. L’article parle des troubleshootings où 1 des 5 hypothèses ne sont pas satisfaites (les mêmes hypothèses utilisées dans d’autres articles) mais ne parle pas de comment les résoudre, il se restreint à calculer quelle serait la meilleure approximation possible théoriquement (il dit qu’en effet pour quelques uns de ces problèmes il n’existait pas d’algorithme quand l’article a été écrit).

Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes :

Cet article semble être une présentation d’un cas spécifique de troubleshooting. Si j’ai bien compris il détaille comment un système pour diminuer les coûts des défauts dans une usine, en utilisant un réseau bayésien. Il peut être intéressant car il parle un peu plus des aspects pratiques de l'implémentation d’un algorithme qui utilise des réseaux mais sans lire plus attentivement je n’arrive pas a dire si il y a vraiment un lien avec ce qu’on se propose à faire. Le cas présenté est basé sur des variables continues et pas discrètes.

Fault Isolation based on Decision-Theoretic Troubleshooting :

Celui là m'intéresse beaucoup dans un premier temps. Il ressemble à une étude de cas (on dit comme ça en français?) présenté de façon didactique, on peut avoir des idées de comment structurer le nôtre. Le langage me semple plus “approchable” aussi et il y a une partie sur l’implémentation aussi.

Decision theoretic troubleshooting of coherent systems :

Cela est plus pratique aussi mais quelques suppositions qu’ils font sont différentes des suppositions qu’on a vu jusqu’à maintenant : ils supposent qu'un composant a peut-être été choisi pour être réparé mais, faute de connaissances, il n'est pas vraiment réparé.

Non seulement cela, mais le manque de connaissances indique également que le fait que la réparation a échoué n'est pas connu. J’ai l’impression que cela peut beaucoup changer ce qui peut se passer (car il faut alors considérer que toutes les composantes peuvent être la source du problème, même celles qui ont déjà été réparées). De tout façon, il reste intéressant.