Using heuristic search for optimizing maintenance plans

Teemu Mutanen
ICT, Knowledge Intensive Services
VTT Technical Research Centre of Finland
P.O.Box 1000, FI-02044 VTT, Finland
E-mail: teemu.mutanen@vtt.fi

Abstract. This work addresses the maintenance action selection process. Maintenance personnel need to evaluate maintenance actions and costs to keep the machines in working condition. Group of actions are evaluated together as maintenance plans. The maintenance plans as output provide information to the user about which actions to take if any and what future actions should be prepared for. The heuristic search method is implemented as part of general use toolbox for analysis of measurements from movable work machines. Impacts from machine’s usage restrictions and maintenance activities are analysed. The results show that once put on a temporal perspective, the prioritized order of the actions is different and provide additional information to the user.

1. Maintenance actions over time
This paper describes a Decision Support System (DSS) implementation for maintenance planning and selection. In maintenance the prioritized order of planned actions may change when actions are evaluated over time. There are several reasons which might trigger the need for a new maintenance plan, e.g. an indication of a failure, change in configurations or new or repaired components are presented. To maintain the machine in good and working condition the maintenance plan needs regular updates with re-evaluations of what maintenance actions to take when needed. Maintenance decisions are often based on the overall costs and benefits. The overall costs may include elements from several variables, for example idle hours, component costs and number of maintenance actions. This paper presents a pragmatic way to advance the measurement data and produce a selection of actions as a maintenance plan, i.e. provide decision support for maintenance personnel.

A method to support formulation of the maintenance plans was developed and implemented to a Matlab-toolbox. The solution differs from existing maintenance DSS’s in temporal aspect. Instead of ranking single maintenance actions according to multiple criteria, the implementation use search and evaluate combinations of maintenance actions’ impact over time, e.g. 100 days. The solution combines the use of expert knowledge, also known as tacit knowledge (see e.g. [1]), and failure probabilities as input. Heuristic search is used to find the most suitable maintenance plans. This paper makes the following contribution: i) presents practical research topic, ii) presents generic framework to value and compare maintenance plans based on machine’s usage and operational environment and iii) discuss future research directions.

Published under licence by IOP Publishing Ltd
After the introduction this paper has four sections, which are organized as follows: in section 2 related research is presented in more detail. In section 3 some of the implementation issues are presented along with mathematical formulations. Section 4 presents the results of our research. In section 5 limitations of the approach are discussed with possible future research topics and conclusions.

2. Background

Predictive maintenance has been studied widely before. For example in 1975 Neuman and Bonhomme evaluated maintenance policies from Markov chains [2]. Neuman and Bonhomme combined the analysis of markov-chain models with fault-tree analysis. On decision making Zadeh and Bellman has studied decision making under fuzzy environments [3]. The fuzzy set approach presented in [3] address the maintenance topic by assuming the system under control known. The uncertainty and fuzziness in the system description are found in the formulation of goals and restrictions. More often than not the persons evaluating the machine under maintenance tasks cannot quantify their knowledge in detail, and when they do the values may include large amount of uncertainty.

Recent studies of predictive maintenance include for example: fault prognosis by neural networks in [4], fuzzy transition probabilities in [5] which combines Markov process transition probability with fuzzy sets. Condition monitoring and condition-based maintenance (CBM) is wide and active research area, for example Garcia et al. developed a software solution for condition monitoring in [6] and Jardine et al. review machine diagnostic and prognostic literature in [7]. Decision support in CBM has been studied by [8].

The tacit knowledge on various decision support applications have been studied for example from the ICT point of view in [9]. Campos studied literature and artificial intelligent methods, because rarity of experts. The expert knowledge has also been studied from cognitive bias point of view by Arnott [10]. Arnott studied existing psychological research to produce information about what prevents experts from formulating clear idea about the target decision task. Nikravesh and Azvine has presented [11] a decision module for autonomous systems. Their formulation is divided in two parts: a) analysis and design; b) ranking decision alternatives. Bisc-dss [11] base their decision support for example on case-based reasoning. Their method is targeted to the environments of impression and uncertainty, mainly addressing the data integrity problem from algorithmic point of view.

Some recent decision support researches include work by Voisin et al. [12] and Yam et al. in [8], where they created a decision support module based on neural network approach and predicted the trend of equipment deterioration in a power plant. The work by Voisin et al. [12] focus on creating a generic framework for prognosis business process.

To the best of our knowledge search based methods have not been used in predictive maintenance and specifically in evaluation temporal aspects of maintenance actions. One of the closest studies is by Gerven et al. [13] in which they applied partially observable Markov decision processes (POMDP) in dynamic decision problem in clinical oncology.

3. Materials & methods

3.1. Maintenance plan

In the DSS the idea of a maintenance plan is fundamental. An example of a maintenance plan is presented in Figure 1. As can be seen from Figure 1 maintenance plan in this work’s context is read as a series of consecutive maintenance actions over time. Maintenance plan or strategy may
also be interpreted as a long time vision of machine fleet maintenance or overall maintenance support for machines (see e.g. [14]). Focus is on case specific questions such as, if there is an indication of a failure, what actions to take if any. That is to say, a plan is interpreted as an indication of what to prepare for if an action is conducted now.

Figure 1. Example of a maintenance plan produced as result. The example shows three maintenance actions placed on time scale from \( t = 1 \) to \( t = 100 \). User can rank several of these plans based on chosen criteria, for example cost.

Maintenance actions are categorized in to two groups, namely maintenance activities and operational restrictions. Maintenance activities include all part changes, repairs, scheduled maintenances and activities such as lubrication and cleaning. The operational restrictions include actions which somehow moderate the machines usage e.g. maximum usage hours, maximum loads or capacity levels. Examples of possible maintenance actions are presented in Table 1. In addition to activities and restrictions, the option to do nothing is also included in the set of available actions.

When several maintenance actions’ possibilities are included in temporal fashion and all of the options are evaluated, the result will appear similar to a tree. In Figure 2 an example of a tree of consecutive actions is presented. In this work branches of this type of tree are interpreted as maintenance plans. The question for the method, how to find a good path in a tree?

Figure 2. Consecutive maintenance actions create a tree. Branches in this tree are interpreted as paths, which are seen also as maintenance plans.

The number of possible maintenance plans can be very large, thus computing all paths is computationally infeasible. If the number of available actions is large, then the dimension of the tree grows very large when the temporal aspect is stretched out to tens or hundreds of time steps. Heuristic search is used to find the most suitable maintenance plans.
### 3.2. Relevant maintenance actions

One of the objectives of the toolbox, also of the DSS module, is the generality of use cases. As previously noted, the possible maintenance actions to be evaluated include both maintenance activities and operational restrictions. In Table 1 examples of potential action types are presented. Each of the actions is presumed to have information about variables such as cost, impact, schedule and idle hours. The input information about actions is received from users or experts. The expert’s input values can also be modified based on measurements from for example other machines in the fleet.

As the developed solution is implemented to a Matlab-toolbox, the DSS module is expected to be used together with toolbox’s other functionality. One of the related feature is a Root Cause Analysis (RCA) method: Fault Tree Analysis (FTA) [15]. The fault tree includes all the root causes for a failure state. From the fault tree we will receive an estimate for each root cause’s error probability, which help us in maintenance actions subset selection. Fault trees are a way of advancing the tacit knowledge.

### Table 1. Examples of the input information used for the maintenance plan optimization.

| ID | Component | Operation          | Schedule | Usage | Costs | Impact |
|----|-----------|--------------------|----------|-------|-------|--------|
| maintenance activities | \( a_i \) | none | none | - | \( c_0 \) | \( i_0 \) |
| | \( a_{i+1} \) | bearing | lubrication | 1 day | - | \( c_1 \) | \( i_1 \) |
| | \( a_{i+2} \) | bearing | replacement | 7 days | - | \( c_2 \) | \( i_2 \) |
| | \( a_{i+3} \) | break | inspection | 1 day | - | \( c_3 \) | \( i_3 \) |
| | \( a_{i+4} \) | break | cleaning | 2 days | - | \( c_4 \) | \( i_4 \) |
| | ... | ... | ... | ... | ... | ... | ... |

| operational restrictions | \( a_j \) | do nothing | 100% | \( j_0 \) |
| | \( a_{j+1} \) | terminate usage | 0% | \( j_1 \) |
| | \( a_{j+2} \) | monitor & terminate | 100% | \( j_2 \) |
| | \( a_{j+3} \) | adaptation, load | 75% | \( j_3 \) |
| | \( a_{j+4} \) | adaptation, hours | 80% | \( j_4 \) |
| | ... | ... | ... | ... | ... | ... |

In Table 1 is also presented a large uncertainty source to the quality of results the DSS module produces, namely reliable input data. The amount of input information required is not large. However, the heuristic search and the state representation formulation rely on the impacts and costs provided by the user. Default values can be changed during the analysis.

### 3.3. How long will machine last by current usage

Each one of the maintenance activities has decreasing impact on failure probability, thus increasing impact to the estimate of machine’s usage lifetime. The operational restrictions are analysed separately from maintenance actions. The restrictions will give information about how long machine’s condition is sufficient hence without maintenance personnel intervention. In Table 1 examples of usage restrictions of the machine are presented along with examples of maintenance activities. The usage restrictions behave as modified usage.

In Figure 3 an example of the impact of usage restriction is presented. In the example past values of lifetime estimates are received from fault tree. In the DSS implementation the future estimate is formulated based on past information. The lifetime estimates may as well be
based on direct measurements, e.g. from a fracture, but in Figure 3 the history values are not measurements but past estimates.

The operational restrictions do not cost anything. Those have an effect to the machine’s production capability. If the usage is restricted enough, in some cases not all of the production target can be met. Thus the restrictions do not have direct costs, but they may have additional effects similar to costs. In the implementation only average production values are included to the DSS, e.g. historical daily averages, and apply those to give indication of how well production targets are met.

3.4. How to find a good maintenance plan

As for the objective of toolbox’s generality, the machine’s condition simply formulated by two variables: failure probability and operational restrictions. These two variables enable the inclusion of both maintenance action types into the model and still the model’s state space is small for general use case. The A* algorithm was chosen and adapted to calculate and evaluate combinations of maintenance actions and the impacts those have. The heuristic search does not need set of action combinations as input, instead the method allows the user to formulate possible maintenance activities and the method formulates plans as computation proceeds. The search method analyses all of the most suitable directions from the start state.

The state representation for the algorithm is defined by three variables:

- failure probability
- operational restriction
- time step

To restrict the dimension of the state space and to enable comparison between two states we quantized both probabilities. Both probability ranges were divided in to 5% intervals. Although the exact values are kept in memory for calculations, two states are interpreted as equal if the exact values are within the same interval. Third variable is time. The time granularity can vary, depending of the situation, time step can be for example an hour or a day or a week. An example of the three variable state space is presented in Figure 4. The algorithm is given one start state, e.g. \([or = x, fp = y, t = 0]\), after which the following states in time are determined by available actions.
3.4.1. $A^*$ algorithm  The algorithm was first developed in 1968 ([17]) for path finding and graphical search. The $A^*$ algorithm uses heuristics for finding paths and it is widely used for its performance and accuracy. $A^*$ algorithm can be used to find the optimal solution for path finding problem, i.e. distance and cost, if you are interested about optimality of $A^*$, see e.g. [16].

The $A^*$ algorithm formulates and evaluates each possible path based on two functions, namely cost and distance. The algorithm memorizes the cost for already travelled path and calculates an estimate for remaining distance. In the algorithm, the function $f$ (Eq. 1) is assumed to be available. By sorting the values from function $f$ the algorithm determines which path (or node) is the best one to expand next. The basic idea is to begin creating paths from start node and advance from the nodes indicated by $f$. The $A^*$ algorithm is terminated when a goal node is reached. The $A^*$ algorithm is usually formulated as follows [18]:

$$f(s_i) = g(s_i) + \hat{h}(s_i)$$  \hspace{1cm} (1)

which equals to the cost of lowest-cost path to a goal - constrained to go through state $s$ in time $t$, i.e. $s_i$. $g(s_i)$ is the total cost it has taken to get from starting state ($t = 0$) to current state($t = i$) and $\hat{h}(s_i)$ is the estimate of the distance from state $s_i$ till finish state. $f(s_i)$ is the current estimated shortest path. $f(s_i)$ is the true shortest path which is discovered when $A^*$ is finished.

3.4.2. Performance  $A^*$ algorithm is widely used because of its performance. However, the algorithm is not very efficient if the distance in the graph’s path finding is large, or the time dimension grows large. The number of possible paths grows very large when the step size is small and the distance is large. This problem is addressed by providing the user an option to modify the temporal granularity, i.e. a day or a week. In addition, calculating maintenance actions’ impact on a daily basis, it is not reasonable to calculate those 6 months hence. Few months is sufficient time horizon for the toolbox’s DSS.

The results from $A^*$ algorithm depend on the heuristic written to the algorithm. From user’s perspective, this means that how the user formulates the input information determines the type of results received. The general idea is that machine usage increases the failure probability and maintenance decreases it. The operational restrictions possibly give more usage days
before possible failure. If these impacts, costs and probabilities are favourable towards some actions, then the optimal plans as result will also show similar bias. However, in the toolbox implementation, the user is provided several plans with diversity. Thus, the overall results should mimic the optimal solution, but in addition the results would include sub-optimal plans. In the machine maintenance task, good enough solutions work as well as optimal solutions, if those maintain the machine in a good working condition.

3.4.3. Machine’s condition index

In the implementation the two probabilities are combined and the share of achieved production into an index. This index gives an indication about what condition the machine is. The index enables the analysis of the suitable directions for the search algorithm. The indices is formulated in equations 2, 3 and 4.

\[
g(t) = [1 - pc(t)] + fp(t) \quad (2)
\]

\[
g(t) = [1 - pc(t)]^2 + fp(t)^2 \ast c \quad (3)
\]

\[
g(t) = fp(t)/pc(t) \quad (4)
\]

where \( g(t) \) is the index. The search id conducted to maintain the machine as good condition as possible. \( pc(t) \) is the share of achieved production if operations are restricted. \( fp(t) \) describes the machine’s failure probability on time \( t \).

4. Comparable maintenance plans

Maintenance plan presented in Figure 1 gives an idea how the results received from heuristic search algorithm are presented. The algorithm itself produces single paths from the tree, shown in Figure 2. In the implementation, the maintenance actions are plotted in to temporal perspective to form a maintenance plan. Some examples of the plans are presented in Figures 5, 6 and 7. The user is presented several, for example three different maintenance plans to support decision making.

![Figure 5](image)

**Figure 5.** The machine’s use is restricted (= \( a_2 \)) some time before maintenance, and the restriction is removed (= \( a_3 \)) as maintenance (= \( a_5 \)) is conducted.

In Figure 5 is presented a plan of a case where the machine’s usage is restricted some time before the actual maintenance takes place, after which the usage restriction is removed. The operational restriction affects the failure probability enough to enable the machine to work till the maintenance is conducted. Figure 6 presents another case where machine is kept working by conducting few minor actions before larger maintenance action. In this case no usage restrictions
The machine’s is kept working by conducting minor (\(a_3\)) actions before the larger maintenance (\(a_8\)) is conducted.

The machine’s use is restricted (\(a_4\)) same time while conducting the analysis and after larger maintenance (\(a_{12}\)) conducted the usage restriction is removed (\(a_3\)).

are needed. In figure 7 a case is presented where the machine’s usage is restricted right away and after the maintenance action the restriction is removed.

Each of the cases in Figures 5, 6 and 7 have different costs and different effects on the production capability of the machine. The operational restrictions may affect the usage in a way that some production targets cannot be met, as mentioned above the implementation only takes in to account the average production rate. The evaluation of the importance of both of these factors is left to the user.

4.1. Results as text or printout
The user is also able to see a version of the plans in printed fashion. In the following, we present an example of this print out.

- Plan A - Plan B - Plan C
Day: 1 - a: a_4 - b: a_4 - c: a_4
Day: 2 - a: a_9 - b: a_1 - c: a_2
Day: 3 - a: a_7 - b: a_2 - c: a_4
Day: 4 - a: a_1 - b: a_8 - c: a_2
Day: 5 - a: a_2 - b: a_2 - c: a_4
Day: 6 - a: a_7 - b: a_4 - c: a_1
Day: 7 - a: a_4 - b: a_2 - c: a_2
In the print out above\(^2\), three maintenance plans are presented. Along with the action on given day, the overall cost is presented.

5. Discussion

5.1. Interpretation of a maintenance plan

As in Figure 5, 6 or 7 the maintenance plans are presented in temporal fashion. Each of these plans could be interpreted as a direction, if an action is conducted right now what should be prepared for next and when. The results will give the user indication about what actions will maintain the machine good enough condition and what is needed in the future. The maintenance plans have various other elements than shown in the examples, the actions found by the algorithm after time step 50 are not provided to the user. The time frame is extended beyond 50 because the effect of larger maintenance actions is then also included, their cost would otherwise be unnecessarily large. More often than not the plans with few actions are considered better than many in maintenance. Would more or less richer plans prove to be more useful considering the objective of the user?

The current implementation of the dss module allows the user to give large, already scheduled maintenances as input. In this scenario, the user might want to have support for a decision how to maintain the machine working till the next scheduled maintenance? And given the current state, or indication of failure, how does smaller and larger actions relate to the maintenance task? These questions are addressed by the current implementation. The future research might focus on how well these implemented plans will provide information needed?

5.2. In reality group of maintenance actions are carried out at once

Machine maintenances are more often than not conducted as several actions at the same time. This has practical reasons, for example the failure probability usually increases each time the machine’s cover is opened. The actions are also conducted at the same time because of the cost of maintenance personnel’s work. This implies that in addition to the list of single maintenance actions, the user should provide information about larger maintenances. The heuristic search algorithm does not consider options such as two or more maintenance actions as a group. Of course the results could be interpreted as such, but if the user sees beneficial that larger maintenances are included to the analysis, then these should be included to the input information. The solution developed in this paper provides information about the order and impact of the selected actions.

5.3. Advanced search

The current implementation does produce maintenance plans as result. The heuristic search algorithm selected was motivated by simplicity and adaptation. In the future more research is needed for path-finding algorithms that produce alternative solutions in addition to the optimal

\(^2\) The actual values are imaginary, thus does not correspond to a case.
solution. Another research issue is the heuristic idea itself, what kind of function determines which one of the two paths being compared is better or cheaper?

The modified $A^*$ presented here is sensitive to the input values provided by the user. The amount of input data needed was decreased by incorporating fault trees and measurements from other machines in the fleet. However, there is still a lot of uncertainty produced along the input data. One relevant future research direction is to analyse the impacts uncertainty in the input values create. Another direction related to the input data’s relevance is the diversity of the results. How to produce diverse group of maintenance plans as solution, maybe with pre-defined diversity level, is outside the research scope of this work.

The problem could be formulated also as Partially Observable Markov Decision Process (POMDP) or other reinforcement learning method. In this case a path-finding algorithm need not be relied, but instead the state probabilities directly could be counted. If formulated as POMDP, then focus would be more on how sure and exact the maintenance action’s impact is, or is not? How would the POMDP idea fit to the maintenance plan comparison and how well would it be able to produce diverse options is left to future research.

6. Conclusions

The solution for maintenance DSS was implemented in the maintenance toolbox to provide support for maintenance personnel’s decision making. The current solution allows the user to provide possible already scheduled maintenance activities as input. The solution also supports time lag before the maintenance action can be conducted, i.e. the decision can be made today, but personnel and part are available a week from now, thus the question remains how to manage till then. In other words, what to do, when and for what to prepare for next? These are the questions for which the DSS will provide information.

References
[1] Polanyi M 2009 The Tacit Dimension (The University of Chicago Press)
[2] Neuman C and Bonhomme N 1975 Reliability, IEEE Transactions on R-24 37 –44 ISSN 0018-9529
[3] Bellman R E and Zadeh L A 1970 Management Science 17 B141–B164
[4] Vachtsevanos G and Wang P 2001 AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference pp 857 –870
[5] Du R and Yeung K 2004 Engineering Applications of Artificial Intelligence 17 457 – 467 ISSN 0952-1976
[6] Garcia M C, Sanz-Bobi M A and del Pico J 2006 Computers in Industry 57 552 – 568 ISSN 0166-3615
e-maintenance Special Issue
[7] Jardine A K, Lin D and Banjevic D 2006 Mechanical Systems and Signal Processing 20 1483 – 1510 ISSN 0888-3270
[8] Yam R C M, Tse P, Li L and Tu P 2001 The International Journal of Advanced Manufacturing Technology 17(5) 383–391 ISSN 0268-3768
[9] Campos J 2009 Computers in Industry 60 1 – 20 ISSN 0166-3615
[10] Arnott D 2006 Information Systems Journal 16 55–78 ISSN 1365-2575
[11] Nikravesh M 2003 Industrial Informatics, 2003. INDIN 2003. Proceedings. IEEE International Conference on pp 396 – 401
[12] Voisin A, Levrat E, Cochet E and Iung B 2010 Journal of Intelligent Manufacturing 21(2) 177–193 ISSN 0956-515
[13] van Gerven M A, De F J, Taal B G and Lucas P J 2007 Artificial Intelligence in Medicine 40 171 – 186 ISSN 0933-3657
[14] Holmberg K and Folkeson A 1992 Operational Reliability and Systematic Maintenance (Chapman and Hall/CRC) ISBN 978-1851661626
[15] Vesely W, Goldberg F, Roberts N and Haasl D 1981 Fault Tree Handbook (NUCLEAR REGULATORY COMMISSION WASHINGTON DC)
[16] Dechter R and Pearl J 1985 J. ACM 32(3) 505–536
[17] Hart P, Nilsson N and Raphael B 1968 Systems Science and Cybernetics, IEEE Transactions on 4 100 –107 ISSN 0036-1567
[18] Russell S J and Norvig P 2009 Artificial Intelligence: A Modern Approach 3rd ed (Prentice Hall)