Developing Expert System for Managing Maintenance Projects

A Business Intelligence Solution

Zsolt T. Kosztyán, Csaba Hegedűs, and Judit Kiss

University of Pannonia, Department of Management
Egyetem u. 10, H-8200 Veszprém, Hungary
{kzst, hegeduscs, kissjudit}@gtk.uni-pannon.hu

Abstract. Handling specialities of maintenance projects is a highly challenging task. On the one hand the operations of the maintenance task are fixed, and can be described with a network or process planning methods. On the other hand the sequence of realizing maintenance tasks depend on the risks and reliabilities. Therefore, traditional project and process planning methods are not the most appropriate ones at managing the sequence of maintenance tasks. In this paper a new planning method is introduced, where project constraints can be considered in determining the optimal maintenance project. Tasks which have to be realized can be ranked with our method based on their reliabilities or risks. Estimating reliabilities, theory of stochastic process and expressing of measurement uncertainty are also applied and improved in order to handle decision errors and their consequences. To determine feasible maintenance projects an expert system has been developed to determine which application can be used for diagnostic processes and forecasting the failures.

Keywords: Project Expert Matrix, managing maintenance projects, project expert system

1 Reliability and Risk Based Ranking of Maintenance Tasks

In case of high number of devices the planning and scheduling of maintenance tasks is very difficult. The reliability of the devices or the risk of its failure must be considered as well as the time, cost and resource constraints.

In this paper an expert system is introduced that use the measurement results of the observed characteristics of the processes or devices as input and determine the priority of maintenance tasks. It specifies which operations should be realized in different kind of maintenance tasks, what maintenance task should be realized, how to realize these tasks, how long does it take and how many resources these tasks require.

According to the RCM (Reliability-Centered Maintenance) [1] the maintenance tasks are ranked by the probability of the device failure in the time interval of the maintenance planning. Those devices that have higher failure rate and therefore have lower operating time between failures are preferred and their maintenance tasks are completed first. The order of priority can be more exact if the costs and required time of the inspection and preventive maintenance are compared to the costs and required time of the corrective maintenance and both the material and immaterial loss of the failures. The severity of the failures consequence is weighted by the probability of the failure so the classification and decisions are carried out on the basis of risks (RBM – Risk-Based Maintenance) [2].

However, in the measurement of the observed characteristic of a device measurement error must be taken into consideration. Our method regarding the consideration of these measurement uncertainties are shown in the following chapter.

2 The Consideration of Measurement Uncertainty when Dealing with Reliability and Risk

A number of our previous papers [3-5] deal with the consideration of uncertainty resulting from the measurement and sampling. The consideration of measurement uncertainty is necessary to make a proper decision. The decision errors result from sampling and measurement uncertainty in the case of the inspection of the degradation of a device.

The process of the observed characteristic can be treated as time series and after the decomposition the next values of the characteristic can be predicted with linear stochastic models. This model predicts the next value on the basis of actual and previous values of real process and prediction error. However, this prediction can also contain uncertainty so the lower and upper bound of confidence interval appears below and above the time series. After the validation of the stochastic model it can be used for forecasting. The further we try to forecast the higher the uncertainty will be.

The costs of decision errors must also be determined. If maintenance is performed before it is required some parts of the productivity will not get utilized. The extent and therefore the cost of the unutilized productivity decrease with time. On the other hand if the maintenance is not performed and the device failure occurs beside the breakdown additional costs (health, environmental, economic) might appear. These additional costs are independent from the time of the failure. Therefore there are time dependent and independent costs. The control limit belonging to the minimal total risk can be determined from these costs and probabilities for every point of the time. For the specification of the curve of minimal total risk simulations or analytic calculations can be used.

With a given time interval of the maintenance planning we can calculate (or determine with simulation) the confidence level for each device with its observed value which just reaches the curve of minimal total risk. With our approach probabilities can be assigned to the devices. These probabilities can be the probabilities of the operation in the time interval of the maintenance planning, or just the opposite the probability of the failure within this interval. With the consideration of these probabilities the maintenance plan can be optimized.

3 Planning and Organizing Maintenance Projects

Realization of maintenance tasks can be regarded as a special maintenance project. However, traditional network planning methods have several deficiencies and throw difficulties in the way of using project planning methods in maintenance. The first shortcoming of network planning methods is circles handling. Frequently occurring problem is to realize a task more than once in a project (i.e. diagnose and revise a part of equipment until it works). In spite of the fact that GERT method [6] can handle circles for detecting and managing circles the matrix-based methods give better alternatives.

The other problem is to determine the sequences of the maintenance task. Traditional logic planning methods are hard to use for these projects, because on the hand the sequence of the operation of a maintenance task can be described with a deterministic logic plan (network plan, Gantt chart etc.). On the other hand most sequences of maintenance tasks (i.e. repairing different kind of equipments) are independent from each other. This means the sequence of maintenance tasks can be reversed, or can be ranked by their values of reliability or risk. Since network planning methods cannot be used for ranking the maintenance tasks, hereinafter matrix-based methods are introduced. These methods can handle circles and also can be used for ranking maintenance task sequence.

3.1 Matrix-Based Project Planning Methods

Mainly in case of scheduling production development projects matrix based methods are also used for planning and scheduling. These matrix methods are based on DSM (Design Structure Matrix/Dependency Structure Matrix) methods published by Steward [7]. Tasks of the project are represented in the rows and columns of a matrix. This method handles the iterations between two tasks. Iterative relation between task *A* and *B* means that the sequence of task *A* and task *B* has to be realized more than once. This relation is the simplest cyclic dependency. The cyclic dependencies are called “Circuits” or “Cycles”. Cycles can contain more than two tasks. Detecting cycles is very important in project management, because the iteration can cause the increase of project duration. When using matrix based methods for project planning one of the most important functions is to determine the sequence of the tasks. If the project plan does not contain cycles the matrix of project plan can be reordered into an upper triangular matrix and an activity-on-node graph of the project plan can be topologically ordered. This method is called sequencing [8, 9].

Formerly introduced methods show how to use matrix based methods for project planning. However, DSM matrices can also be used for scheduling [10] and resource allocation [11] or in addition this method can be used for reorganizing the projects. [12, 13]. In this way duration, cost or resource demands of tasks can be represented in the diagonal or an additional column. Numbers that are out of the diagonal can show the lags of successors/predecessors. [14]

Handling Uncertainty Relations. Yassine [15] and Tang [16] showed that in case of project planning there could be uncertain relations between two tasks. They introduced a new method called numerical DSM (NDSM), which handle the strength of the relations between two tasks. When using NDSM matrices the level of the dependency of relations between two tasks can be represented. Numbers instead of “X” in NDSM can represent [17] i.e.:

- Dependency Strength: This can be a measure between 0 and 1, where 1 represents an extremely strong dependency. The matrix can, now, be partitioned by minimizing the sum of the dependency strengths below the diagonal.

- Probability of Repetition: This number reflects the probability of one activity causing rework in another. Feedback relationships represent the probability of having to loop back (i.e. iteration) to earlier (upstream) activities after a downstream activity was performed, while feed-forward relationships can represent the probability of a second-order rework following an iteration.

Despite the fact that these methods handle the uncertain relations they cannot handle the realization priority of the tasks.

Authors have formerly published a Stochastic Network Planning Method (SNPM) [18] for generating all possible project net. Acronym of SNPM alludes to uncertain project net. Enhanced method of SNPM called Project Expert Matrix (PEM) can handle the uncertain realization of the tasks [19, 20]. Similarly to uncertain relations the uncertain realizations of the tasks can mean (1) the probability of task realizations; (2) or the otherwise relative importance of task realizations. The uncertainty of the task realizations can be notated in the diagonal of the PEM matrix (see Table 1). The certain task realizations denoted as 1 or "X" in the PEM matrix.

Table 1. Evaluating the Project Expert Matrix.

<table>
<tr><th>PEM</th><th>NDSM/SNPM</th><th>DSM</th><th>AoN Net</th></tr>
<tr>
<td rowspan="3"><table><tr><td>↥</td><td>A</td><td>B</td></tr><tr><td>A</td><td>X</td><td>?</td></tr><tr><td>B</td><td></td><td>?</td></tr></table></td>
<td rowspan="2"><table><tr><td>↥</td><td>A</td><td>B</td></tr><tr><td>A</td><td></td><td>?</td></tr><tr><td>B</td><td></td><td></td></tr></table></td>
<td><table><tr><td>↥</td><td>A</td><td>B</td></tr><tr><td>A</td><td></td><td>X</td></tr><tr><td>B</td><td></td><td></td></tr></table></td>
<td>A → B</td>
</tr>
<tr>
<td><table><tr><td>↥</td><td>A</td><td>B</td></tr><tr><td>A</td><td></td><td></td></tr><tr><td>B</td><td></td><td></td></tr></table></td>
<td>A
B</td>
</tr>
<tr>
<td colspan="2"><table><tr><td>↥</td><td>A</td></tr><tr><td>A</td><td></td></tr></table></td>
<td>A</td>
</tr>
</table>

3.2 Applying Matrix-Based Project Planning Methods for Organizing Maintenance Projects

In this chapter we show how to apply previously introduced matrix-based methods for planning and organizing maintenance projects. Operations of maintenance (i.e. servicing an equipment, evaluating results of diagnostics etc.) usually can be described as a deterministic logic plan (using a task list, Gant chart etc.). In this case the operations have to be realized in a sequence following the task list. These operations can be represented by a DSM matrix or can be contracted into a maintenance task. However, realization order of maintenance tasks can be determined considering the values of reliabilities and risks. Duration, costs, resources can be assigned to the maintenance tasks.

We developed a method called Maintenance Project Planning and Scheduling (MPPS): Values of risks or reliabilities of equipments can be represented in the diagonal of Project Expert Matrix. These values can be determined as introduced in chapter 1-2. Realized maintenance tasks categorized by level of risk or reliability (see chapter 1). Maintenance tasks with low level risk or low level reliability can be realized with a succeeding project. If the level of risk of a maintenance task is high the probability of realization of this task is also high. This probability is marked as *p*. (In this case 1-*p* is the probability of leaving out this task from the project. If *p* is the importance of realization of the task, than 1-*p* is the value of ignorance). The probable project scenarios can be ranked by its probabilities. Probability of a project scenario is the geometric average of the probability of realizations (of realized tasks) and probability of ignorance (of non realized tasks). Using a PEM matrix maintenance tasks will be ranked by their realization probabilities.

Similarly the probability of a project structure can be defined. If two maintenance tasks can be realized both parallel and in a sequence and the sequential and parallel realization is indifferent, the probability of relation between two tasks can be notated as 0.5. If sequential realization is preferred, probability of relation between two tasks are higher than 0.5 otherwise lower than 0.5. The target function is to select the most probable project scenario according to the project budget, and the most probable project structure according to the time and resource constraints.

When constructing an aggregate PEM matrix (aPEM) the plan of operations of a maintenance task is an input data and can be described with a DSM matrix or a network plan. These operations can be contracted into different kind of maintenance tasks. The probability of task realizations can be determined from the values of reliabilities or level of risks and can be represented into the diagonal of the aPEM matrix. Maintenance tasks should be ordered by their probability of realizations. Preferred sequential and parallel realization can be expressed by the strength of relation between two tasks. Accordingly the budget feasible project scenarios can be specified and can be ranked by the probability of project scenarios. Project scenarios can be represented by an SNPM or an NDSM matrix. In this phase we can answer *WHAT* maintenance tasks should be realized in agreement with the project budget. According to the time and resource constraints feasible project structures can be specified and can be ranked depending on the probabilities. In this phase we can answer *HOW* to realize the maintenance tasks. The logic plan of project structures can be represented by a DSM matrix. Duration and resource demands can be represented by a resource sheet. At the end we can also answer *HOW LONG* maintenance tasks take, *HOW MUCH* are the cost and resource demands.

A project plan is *infeasible* if (time, cost, resource) demands are higher than the project constraints. The optimal project plan is the feasible project plan that possesses the highest probability.

For determining feasible project scenarios and feasible project structures a genetic algorithm can be used in order to decrease the need of computation resources. Genetic algorithms can be used for NP complete or NP hard problems [21] and also can be used if optimal solution should be determined from the large scale probable/feasible solutions. When using genetic algorithms for project scheduling, the initial population will be the set of probable project scenarios and project structures of a project scenario. Evolution operators (selection, mutation, recombination, etc.) are fulfilled in the entities of the population. Every entity (project structure of a project scenario, which can be represented by a DSM matrix) has a fitness value. To increase effectiveness, and decrease computation time the fitness value is the combination of probability/ importance of project scenario, the resource and time constraints. If project scenario or project structure is infeasible, the fitness value is 0. Effectiveness of genetic algorithms can be improved if we use distributive architectures (CPUs or computers). For handling the numerous probable solutions and computations we used a promising distributive technology called Compute Unified Device Architecture (CUDA) [22], which distributes computation tasks amongst the Graphical Process Units (GPUs).

However, genetic algorithms do not surely find the optimal solution with the help of appropriate fitness function and evolution operators the optimal solution can be approximated. In order to comparing various genetic algorithms all possible project scenarios and all possible project structures are ranked, and most important/most probable feasible project structures are selected for the test projects. Both applied genetic algorithms solved the introduced information project shorter than 1 ms. Henceforth, the formerly introduced MPPS method called naive MPPS method. Using genetic algorithm for this problem called as genetic MPPS method, and using distributed genetic algorithm (applying CUDA) for this problem called as distributive genetic algorithm for Maintenance Project Planning and Scheduling.

At the developing we had to write some interface program between maintenance diagnostic systems and MPPS as well as between MS Project (where the operations of maintenance tasks are) and MPPS. After the genetic algorithm framework finds and ranks the possible project scenarios and structures the interface program export the optimal project structure to MS Project.

4 Results

For comparing different kind of methods, different sizes of random PEM matrices are generated, where the number of uncertain tasks and number of uncertain relations can be defined as a percentage of all tasks and possible relations. Duration of the tasks, project budget and 3 different kinds of resource demands of the tasks are generated randomly. In the comparison test the time constraint was specified. This time con¬straint was the half of the project duration, if all uncertain relation specified to 1. In this case all uncertain relations between tasks were considered as a certain dependency. The resource constraints were the half of maximal resource demands, when all un¬certain relations are considered as a certain independency and the tasks scheduled for earliest start time. Percentage of uncertain task realizations was 80% of all probable tasks. The percentage of uncertain relations between two tasks was 20% of all possible relations. And the time budget was the 15% lower than if all tasks are realized.

For the comparison of the run time of different kinds of methods all methods are run in the same computer architecture, which was a Pentium PC, with Intel Core i7-720QM CPU, 6GB RAM, 320 GB HDD. The GPU was the 1GB NVidia GeForce GT 230M. For the implementing distributive genetic algorithms, the CUDA technology was applied. Table 2 shows specifications and the efficiencies of different kind of MPPS methods.

Table 2. Comparing various Maintenance Project Planning and Scheduling methods.

Size of PEM matrices	Runtime of full evaluating algorithm	Runtime of genetic algorithm	Runtime of distributive genetic algorithm
10x10	62 ms	42 ms	38 ms
20x20	2,1 hours	493 ms	451 ms
50x50	12,2 hours	48 sec	37 sec

Using genetic algorithms the computation time can be extensively decreased. When the number of tasks is more than 300 naive MPPS cannot be used because the computation time is more than a week. Nevertheless, using genetic algorithms for Maintenance Project Planning and Scheduling, the computation time is lower than an hour for 300 tasks (the formerly specified constraints are taken into account).

5 Summary

The introduced methods are the results of a four-year research. Taking measurement uncertainty into consideration in maintenance related decision and forecasting is one of the basic concepts of our model. In this way maintenance can be planned. With our forecasting method the degradation can be prognosed more accurately considering incorrect decision consequences. Using this model the probability of maintenance task realization can be specified and can be used for project planning. The other main concept is using PEM matrix for project planning and scheduling. Since finding optimal project plan (considering the project constraints) is a combinatorial problem, using genetic algorithm can be applied for decreasing computation time extensively. The introduced expert system can be interacted with project planning applications (like MS Project) and diagnostic applications.

References

1. Rausand, M.: Reliabilty centered maintenance. Reliability Engineering System Safety, 60, 121-132 (1998)
2. Khan, F.I., Haddara, M.M.: Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning. Journal of Loss Prevention in the Process Industries vol. 16, iss. 6, 561-573 (2003)
3. Kosztyán, Zs. T., Csizmadia T., Hegedűs Cs., Kovács Z. (2009): A new approach to forecasting conformity treating measurement uncertainty in SPC, AVA (Int. Cong. on the Aspects and Vision of Applied Economics and Informatics), (2009)
4. Kosztyán, Zs.T., Csizmadia, T., Hegedűs, Cs. Kovács, Z.: Treating measurement uncertainty in complete conformity control system. Innovations and Advances in Computer Sciences and Engineering, 79-84 Springer Netherlands, (2010)
5. Hegedűs, Cs., Kosztyán, Zs.T.: Treating measurement uncertainty in maintenance related decisions, 38th ESReDA Seminar on Advanced Maintenance Modelling, Pécs (2010)
6. Pritsker A.A.: GERT: Grafical Evaluation and Review Technique, MEMORANDUM, RM-4973-NASA (1966)

7. Steward, D.V.: System Analysis and Management: Structure, Strategy and Design. New York: Petrocelli Books (1981)
8. Danilovic, M., Browning, T.R.: Managing complex product development projects with design structure matrices and domain mapping matrices. Int. Journal of Project Management 25, 300-314. (2007)
9. Eppinger, S.D. - Whitney, D.E. - Smith, R.P. - Gebala, D.A.: A model-based method for organizing tasks in product development, Research in Engineering Design 6, 1-13. (1994)
10. Huang, E., Chen, S-J.(G.): Estimation of Project Completion Time and Factors Analysis for Concurrent Engineering Project Management: A Simulation Approach. Concurrent Engineering 14, 329-341. (2006)
11. Yan, H-S., Wang, Z., Jiang, M. A.: Quantitative Approach to the Process Modeling and Planning in Concurrent Engineering. Concurrent Engineering 10, 97-111. (2002)
12. Khoo, L.P., Chen, C-H., Jiao, L.: A Dynamic Fuzzy Decision Support Scheme for Concurrent Design Planning. Concurrent Engineering 11, 279-288. (2003)
13. Rick, T., Horváth, M., Bercsey, T.: Design tasks scheduling using genetic algorithms. Periodica Politechnica Ser. Mech. Eng., 50(1), 37-51. (2006)
14. Chen, C-H., Ling, S.F., Chen, W.: Project scheduling for collaborative product development using DSM. Int. Journal of Project Management 21, 291-299. (2003)
15. Yassine, A.A., Falkenburg, D., Chelst, K.: Engineering design management: An information structure approach, Int. Journal of Production Research 37 (1999)
16. Tang, D., Zhu, R., Tang, J., Xu, R., He, R.: Product design knowledge management based on design structure matrix, Advanced Engineering Informatics (2009)
17. Browning, T.R., Eppinger, S.D.: Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development, IEEE Transactions on Engineering Management, 49(4), 428-442. (2002)
18. Kosztyán, Zs.T., Kiss, J.: Stochastic Network Planning Method. Advanced Techniques in Computing Sciences and Software Engineering, 263-268, Springer Netherlands, (2010)
19. Kiss, J., Kosztyán, Zs.T.: Handling the Specialties of Agile IT Projects with a New Planning Method, CONFENIS (The Enterprise Information Systems International Conference on Research and Practical Issues of EIS), Győr (2009)
20. Kiss, J., Kosztyán, Zs.T.: The importance of logic planning in case of IT and innovation projects, AVA (Int. Congr. on the Aspects and Vision of Applied Economics and Informatics), Debrecen (2009)
21. Hartmann, S.: A competitive genetic algorithm for resource-constrained project scheduling. Naval Research Logistics, 45(7), 733-750. (1998)
22. CUDA (Compute Unified Device Architecture) Programming Guide 3.0 (2007), http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf