Aalto University
School of Science

Application of policy iteration to strategic maintenance scheduling

Kalle Alaluusua
Presentation
6.11.2020

MS-E2191 Graduate Seminar on Operations Research
Fall 2020

Aalto University
School of Science

Problem description

Maintenance scheduling

Consider a system that consists of multiple critical components.

How to schedule components' maintenance in the long run when the maintenance decisions influence

- the state of the system
- future wear-off

Keep costs low and reliability high

Maintenance scheduling

Consider a system that consists of multiple critical components.

How to schedule components' maintenance in the long run when the maintenance decisions influence

- the state of the system
- future wear-off

Keep costs low and reliability high

Reliability = ability to perform the required function under prevailing operational conditions for a stated time period

Maintenance scheduling

Naïve approach

- Minimize expected maintenance costs of single components

⇨ Take reliability measures into account poorly

Maintenance scheduling

Naïve approach

- Minimize expected maintenance costs of single components

⇨ Take reliability measures into account poorly

Improvement

Extend the naïve approach and use dynamic programming to

- Group maintenance operations of multiple components effectively
- Introduce a reliability threshold to keep reliability high enough

Aalto University
School of Science

Problem formulation

System state

- Fixed maintenance interval $\Delta t > 0$
- For the system to operate, every component must operate
- Components fail according to some known probability distributions

A discrete time Markov decision process with state variables

- Age $(a_k)_i$
- Failure state $(f_k)_i \in \{0,1\}$

of component i at maintenance instance t_k

Reliability = probability that a system is operational in t_{k+1} given a_k

Costs and dependencies

Costs

- Set-up
- Component specific
- Shutdown
- Downtime

Dependencies

- Economic
- Structural
- Stochastic

Costs and dependencies

Costs

- Set-up
- Component specific
- Shutdown
- Downtime

Dependencies

- Economic
- Structural
- Stochastic

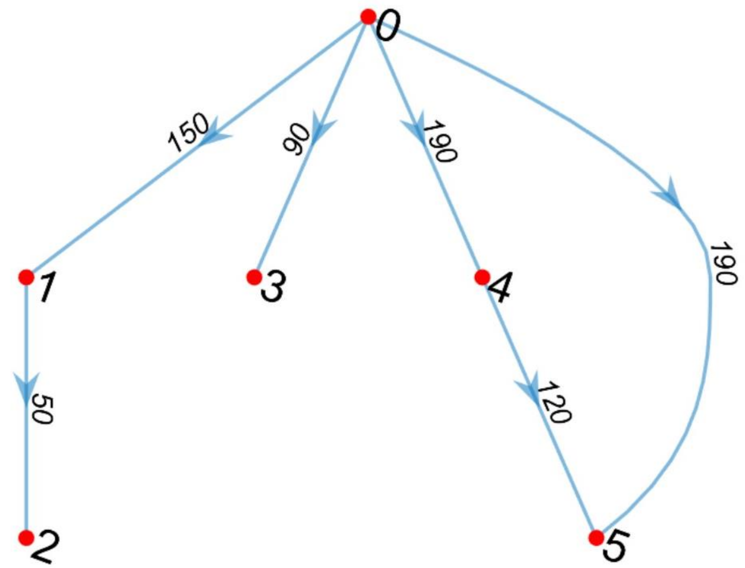


Figure 3.1: Example of a system of five components and corresponding costs of c_{ij}

Leppinen, J. (2020)

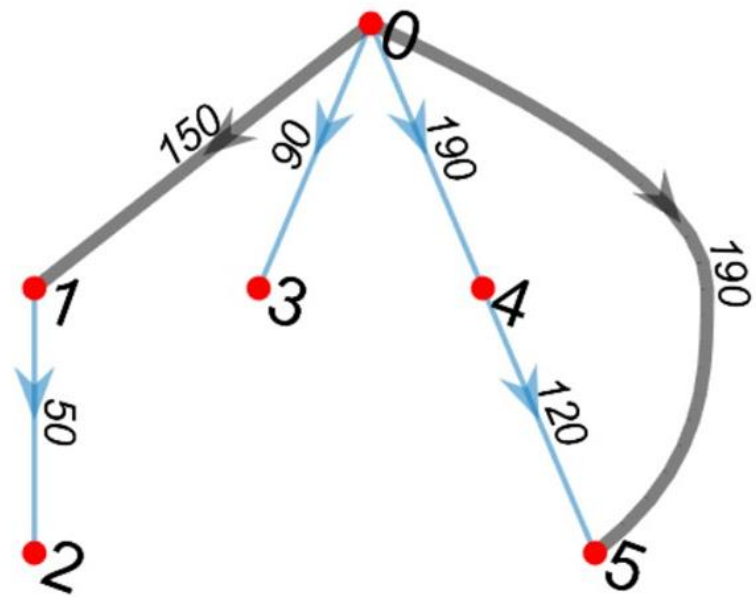
Costs and dependencies

Costs

- Set-up
- Component specific
- Shutdown
- Downtime

Dependencies

- Economic
- Structural
- Stochastic



(a) $x_k = \{1, 5\}$

Leppinen, J. (2020)

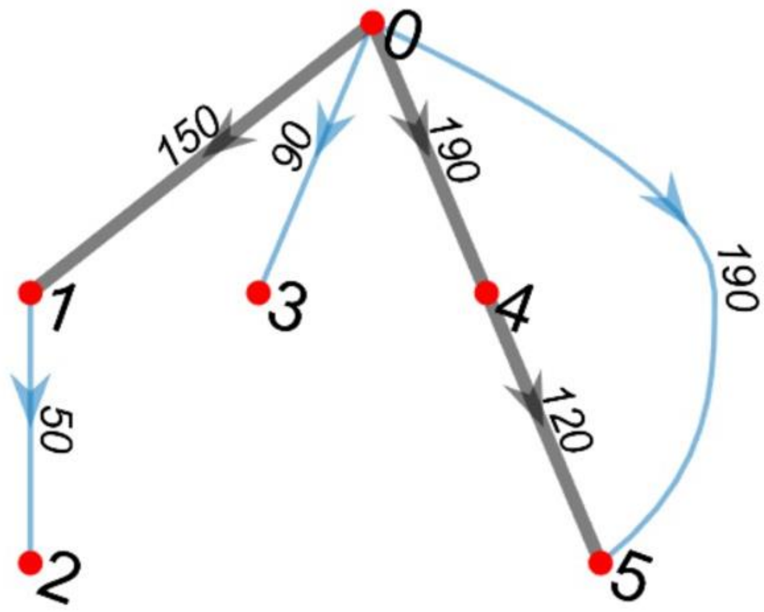
Costs and dependencies

Costs

- Set-up
- Component specific
- Shutdown
- Downtime

Dependencies

- Economic
- Structural
- Stochastic



(b) $x_k = \{1, 4, 5\}$

Leppinen, J. (2020)

Assumptions

- At most single failure per maintenance period
- Components can only be replaced into new ones
- No downtime cost
- No stochastic dependencies

Policy

Portfolio

$x \in \{0,1\}^N$

has $x_i = 1$ when component i is replaced

Feasible portfolios

- Fulfil the reliability threshold
- Replace failed components
- Satisfy structural dependencies

Objective

Find a stationary policy which

- Is feasible
- Minimizes the long run average cost per time unit

Policy iteration algorithm

1. *Step: Initialization* Choose a stationary policy U.

2. *Step: Value-determination step* For the current policy U, compute the unique solution $\{g(U), v(U)\}$ to the following system of linear equations:

$$v_{\sigma_i} = c_{\sigma_i}(U_{\sigma_i}) - g + \sum_{\sigma_j \in S} p_{\sigma_i\sigma_j}(U_{\sigma_i})v_{\sigma_j}, \quad \sigma_i \in S \tag{3.20}$$

$$v_{\sigma_s} = 0,$$

   where σ_s is an arbitrarily chosen state.

3. *Step: Policy-improvement step* For each state $\sigma_i \in S$, determine a portfolio x_k yielding the minimum in

$$\min_{x_k \in X_{\sigma_i}} \left\{ c_{\sigma_i}(x_k) - g(U) + \sum_{\sigma_j \in S} p_{\sigma_i\sigma_j}(x_k)v_{\sigma_j}(U) \right\} \tag{3.21}$$

   The new stationary policy U' is obtained by setting $U'_{\sigma_i} = x_k$ for all $\sigma_i \in S$.

4. *Step: Convergence test* If the new policy U' equals U the algorithm is stopped with policy U. Otherwise, set $U = U'$ and go to step 2.

Leppinen, J. (2020)

Aalto University
School of Science

Case study: A ground transportation equipment system

Components and dependencies

Components

- Engine 1 (E1), engine 2 (E2), chassis (C) and wheels (W)
- Deteriorate over time and have structural dependencies

Structural dependencies

- An engine must be dismantled before it can be replaced
- To replace the chassis, the chassis and both engines must be dismantled
- To replace the wheels, the chassis and the engines must be dismantled

Costs

A fixed set-up cost c0 = 388 for every operation and component specific costs:

Table 5.1: Maintenance costs of different components

component	symbols	dismantle	replacement	corrective surplus
		component specific costs		
engine 1	1, E1	23	393	300
engine 2	2, E2	28	403	300
chassis	3, C	167	413	160
wheels	4, W	0	1000	613

Leppinen, J. (2020)

Directed graph of cost structure

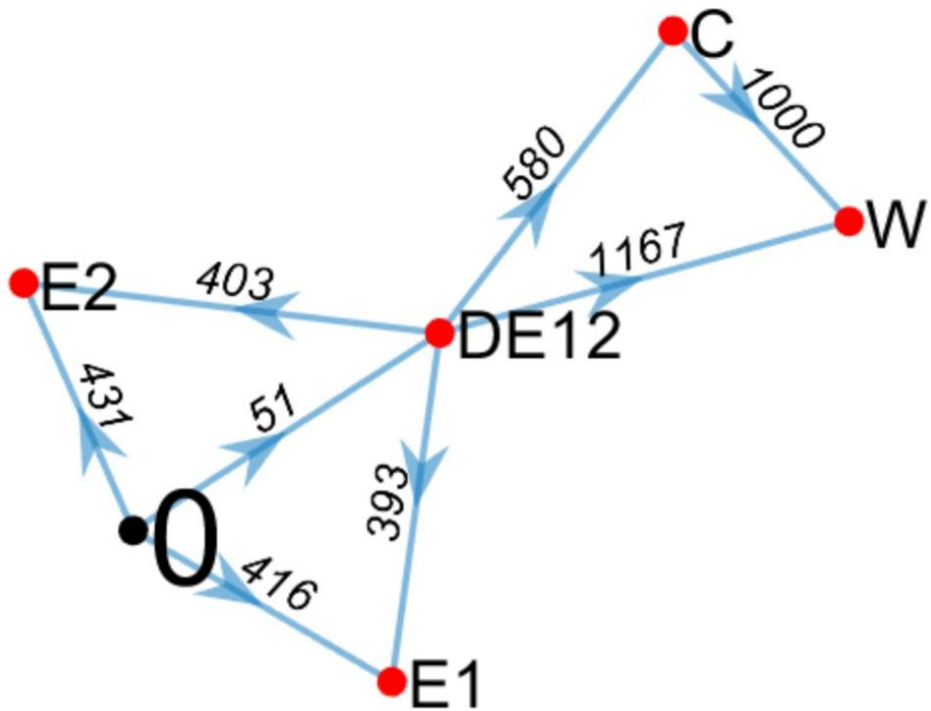
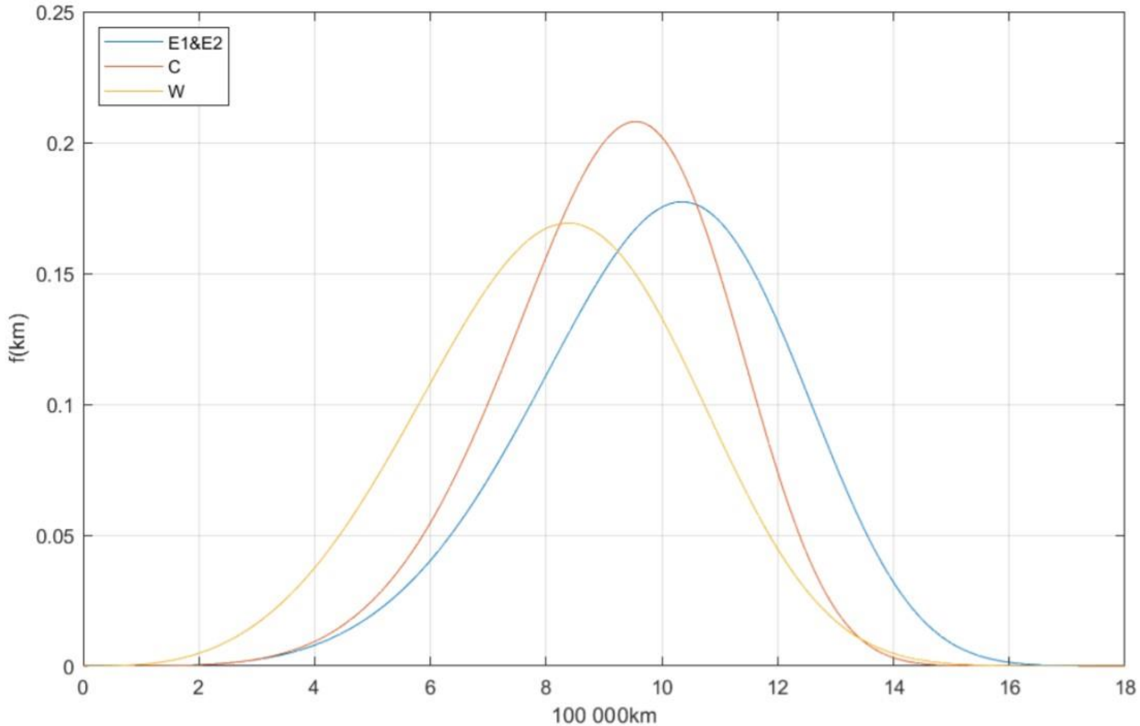


Figure 5.1: Cost structure of the system where the root node is on the left

Leppinen, J. (2020)

Weibull distributed failure probabilites



component	shape k	scale λ
engine 1	5.1	10.8
engine 2	5.1	10.8
chassis	5.5	9.9
wheels	4.0	9.0

Figure 5.2: Failure probability density as a function of distance driven from last replacement

Leppinen, J. (2020)

The results

Portfolio

$x_{E1}x_{E2}x_Cx_W$, $x_i \in \{0,1\}$ has $x_i = 1$ when component i is replaced

Table 5.4: Comparing replacement portfolios when changing reliability threshold as a function of $(a_k)_{E2}$ and $(a_k)_W$, when $(a_k)_{E1} = (a_k)_C = 75$ and $f_k = 0$

$(a_k)_2$, engine 2	$(a_k)_4$, wheels 75	150	225	300	375	450	525	600
				$\rho = 0.90$				
75	0	0	0	0	0	0	0001	0001
150	0	0	0	0	0	0	0001	0001
225	0	0	0	0	0	0	0001	0001
300	0	0	0	0	0	0	0	0101
375	0	0	0	0	0	0	0	0101
450	0	0	0	0	0	0	0	0101
525	0	0	0	0	0	0	0	0101
600	0100	0	0	0	0	0	0101	0101
675	0100	0100	0100	0	0	0101	0101	
750	0100	0100	0100	0100	0101	0101		
825	0100	0100	0100	0100				
				$\rho = 0.95$				
75	0	0	0	0	0	0001		
150	0	0	0	0	0	0001		
225	0	0	0	0	0	0001		
300	0	0	0	0	0	0001		
375	0	0	0	0	0	0101		
450	0	0	0	0	0	0101		
525	0	0	0	0	0101	0101		
600	0	0	0	0100	0101			
675	0100	0100	0100	0100				

Leppinen, J. (2020)

Aalto University
School of Science

Conclusions

Summary

Maintenance scheduling problem

- A discrete time Markov decision process where the state depends on the components ages and the failure state
- Apply policy-iteration to find a stationary policy
- Optimal in terms of average cost over a very long time period

Summary

Design decisions

- Component level: distributions of failure probabilities
- System level: structure as a directed graph
- Environmental level: discretization period

Source

Leppinen, J. (2020). A Dynamic Optimization Model for Maintenance Scheduling of a Multi-Component System (Master’s thesis, Aalto University).

Homework

Consider the case example (Chapter 5) in Leppinen, J. (2020) available in course material.

Briefly explain why the policy-iteration algorithm outperforms the simple and heuristic opportunistic policy. Why, in some cases should you still consider the simple policy over the presented policy-iteration algorithm?

Return your solution to kalle.alaluusua@aalto.fi by 13.11. 09:15.