PK8207 - Lecture memo

Jørn Vatn
Email: jorn.vatn@ntnu.no

Updated 2022-04-27

Spare part optimization

Introduction

When optimizing models for individual components in relation to the interval τ, it may be appropriate to consider whether it pays to have a spare part in stock by allowing us to reduce downtime. In the analysis, we can then compare the situation with and without spare part in stock, and find out if the cost of inventory can be justified. In many situations, there are several components that "fight" for the same spare part, and it becomes a question of how many spare parts we need. We can compare this with the situation at home where the question is how many light bulbs (of a given type) we will normally have in stock to avoid not running out of light bulbs. In this lesson, we'll look at two different ways to model this:

- An analytical model where we can set up equations to calculate the expected share of the time we lack one, two or more spare parts
- A Markov model where we can find the same answer, but where we have more flexibility to give in different assumptions

An analytical model

- Constant failure rate (i.e., the total failure rate for many components that need a new spare part in the event of failure) = λ
- Number of spare parts = s
- The spare parts are stored in a stock and are retrieved from there if necessary
- Failed components are repaired in a workshop
- The number of components under repair in the workshop = X
- Repair rate for each components repaired = μ

- We have endless number of repair men, i.e., a repairman can always start repairing a component that comes to the workshop

Note that we have assumed that components that fail will be repaired. If we instead have to buy new components in the event of a failure, the model will be identical if we allow the expected time it takes to obtain a new component $= 1/\mu$.

Mathematical model

- According to Palm's theorem, $X \sim Po(\lambda/\mu)$
- From the Poisson distribution it follows by introducing $p(k) = \Pr(X = k) = \frac{(\lambda/\mu)^k}{k!} e^{-\lambda/\mu}$:
  - $p(0) = e^{-\lambda/\mu}$
  - $p(s+1) = \frac{\lambda/\mu}{s+1} p(s)$
- The probability of missing spare parts is: $R(s) = \Pr(X > s) = \sum_{k=s+1}^{\infty} p(k)$, which gives:
  - $R(0) = 1 - p(0)$
  - $R(s+1) = \sum_{k=s+2}^{\infty} p(k) = \sum_{k=s+1}^{\infty} p(k) - p(s+1) = R(s) - p(s+1)$
- The number of units we may lack is referred to as BO (Backorders):
  - $\text{EBO}(s) = \text{E(BO)} = \text{E}(max(0, X - s)) = \sum_{k=s+1}^{\infty} (k-s) p(k)$
  - $\text{EBO}(s+1) = \sum_{k=s+2}^{\infty} (k-s-1) p(k) = \sum_{k=s+1}^{\infty} (k-s-1) p(k)$
  - $\text{EBO}(s+1) = \text{EBO}(s) + \sum_{k=s+1}^{\infty} (-1) p(k) = \text{EBO}(s) - R(s)$
- The following recursive regime can then be used
  - $p(0) = e^{-\lambda/\mu}$
  - $R(0) = 1 - p(0)$
  - $\text{EBO}(0) = \text{E}(X) = \lambda/\mu$
  - $p(s+1) = \frac{\lambda/\mu}{s+1} p(s)$
  - $R(s+1) = R(s) - p(s+1)$
  - $\text{EBO}(s+1) = \text{EBO}(s) - R(s)$

Simple cost model

- Cost elements
  - C_U = Unavailability cost per unit of time
  - C_S = Capital cost per unit of time to keep a unit in stock

- Cost equation, i.e., the objective function:
  - $C(s) = C_S s + C_U \text{EBO}(s)$

To minimize the cost equation, $C(s)$ is calculated for different values of s. We must then use the recursive formulas to find EBO(s)

Exercise 1

Consider the following situation:

- Constant failure rate (i.e., the total demand rate of spare parts) = λ = 0.01
- Number of spare parts = s = decision variable
- The spare parts are stored in a stock and retrieved from there upon a demand
- Failed components are repaired in the workshop
- Repair rate for each components repaired = $\mu = 0.1$
- We have endless number of repair men, i.e., a repairman can always start repairing a component that comes to the workshop
- C_U = 10 000 = Unavailability cost per unit of time
- C_S = 2 = Capital cost per unit of time to keep a unit in stock

Find the optimal value of s.

Markov modelling

Markov modelling is a special way to model transitions between system states. Here we will investigate Markov models where we have a limited number of states. Each state is given a number (identifier). It turns out appropriate to let the identifier of a state be the number of spare parts in stock. If the stock is empty and no one is requesting a component, we give the state number 0, while negative state numbers correspond to the number of spare parts we have shortages, i.e., a stock-out situation.

Markov models can in some cases be solved analytically, but we usually need a computer program to calculate the Markov models.

The following assumptions and limitations apply:

- Failures and repair times are exponentially distributed
- We can introduce different strategies, e.g., vary how many repair men we want

- For non-exponential repair times, we can use so-called phase type distributions. This is a little more to elaborate, but can provide reasonably good solutions with not too much extra modelling work
- Disadvantages
  - In principle, we may have infinite number of backorders, while in the model we must limit the number of states in the transition matrix, limiting the number of backorders the model can hold
  - We must manually specify the transition matrix, which can be tedious when testing different strategies, with programming this is not that difficult
  - For very large systems, there may be challenges with computational speed

Model specification

- Constant failure rate = λ, i.e., demand rate of spare parts
- Number of spare parts = s
- The spare parts are stored in a stock and are retrieved from there if necessary
- Failed components are repaired in a workshop
- The number of components under repair in the workshop = X
- Repair rate for each component being repaired = μ
- We have a limited number of repair men, and the number = m

Graphical representation

The following are transitions between states. We assume in the first place that we have a large number of repairmen.

From the Markov model we find the steady state solution, and unavailability (or expected backorder) is given by

$$U(s) = \text{EBO}(s) = P_{-1} + 2P_{-2} + 3P_{-3} + \ldots \tag{1}$$

where P_{-1} is the element in the solution vector representing shortage of exactly one item, P_{-2} shortage of exactly two items etc.

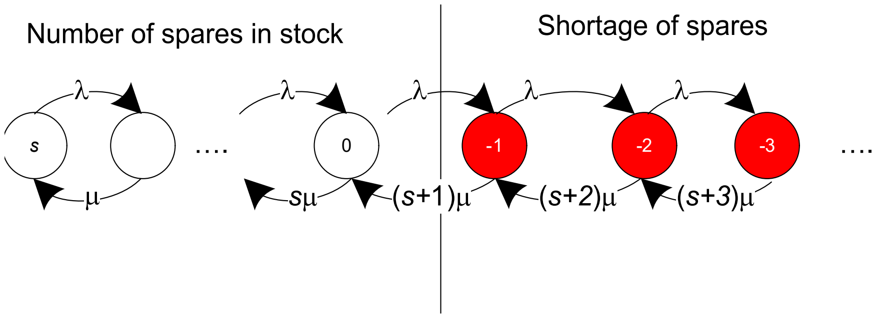

Figure 1: Markov transition diagram

m-Rapairmen

In the model so far, we have assumed that we have an infinite number of repair men. That means that the more devices that are for repair, the greater the repair rate will be. In general, we have assumed that the repair rate is $X\mu$, where X is the number of units for repair in the workshop, while μ is the repair rate (completion rate) each repairman has.

If we only have m repairmen, the transition rate is $min(X,m)\mu$, where X is most easily determined by assessing how many units are for repair for the current state. For example, if $s = 5$, and we consider state -1, X will be $5+1 = 6$, and if we have only $m = 4$ repairers, the rate from state -1 to state 0 will be equal to 4μ.

The cost equation is given by:

$$C(s,m) = C_S s + C_U \text{EBO}(s) + C_M m \tag{2}$$

where C_M is the cost per unit time of having one repairman available. Note that a repair man is doing other tasks, so C_M is not necessarily very large.

Exercise 2

Consider the situation in Exercise 1. Solve the problem by Markov theory. Hint: Set $m = \infty$.

Exercise 3

Consider the situation in Exercise 1. We now also introduce $C_M = 0.25$ equal the cost per unit time per repairman available. Use Markov theory to find the optimal value of s and m

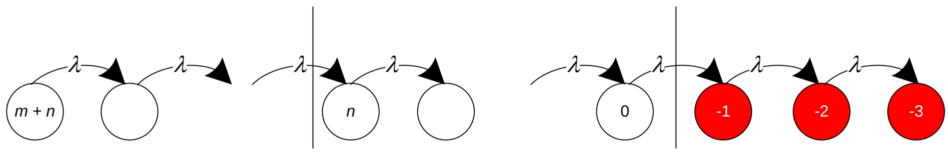

Figure 2: Step 1

A reorder policy model

The following model is similar to the lot size, reorder point policy, (r,Q), used in inventory management. The model assumptions are:

- Constant rate of failure λ
- Mean lead time when ordering new spares = MLT
- Lead times are Gamma (Erlang) distributed with parameters $\alpha = 4$, and $\mu = \alpha/\text{MLT}$
- Totally m new spares are ordered when stock level equals n

Note that $\alpha = 4$ may be changed to account for general value of SD(LT) = $\alpha^{1/2}/\mu$

Figure 2 illustrates the situation for taking components out of the stock. Initially we have $m+n$ components in stock. Then when the level reaches n, i.e., the re-order point, an order is placed for replenishment of the stock.

To model the lead time, we introduce intermediate states representing the gamma distribution, i.e., a transition from state n to state n_1 to state n_2 to state n_3 and finally to state $m+n$. Figure 3 illustrates this.

During the lead time, there might be a new demand for a spare (i.e., a failure). This means that stock level is reduced by one. Figure 4 illustrates this by the transitions from n_1 to $(n-1)_1$ and so on.

Figure 5 illustrates the complete picture.

Optimization

To optimize the model, we need to specify

- c_F = Fixed cost per order
- c_H = Holding cost per item per unit time

From the Markov calculations we can obtain both expected holding cost, and expected number of orders per unit time, together with the expected number of backorders per unit time.

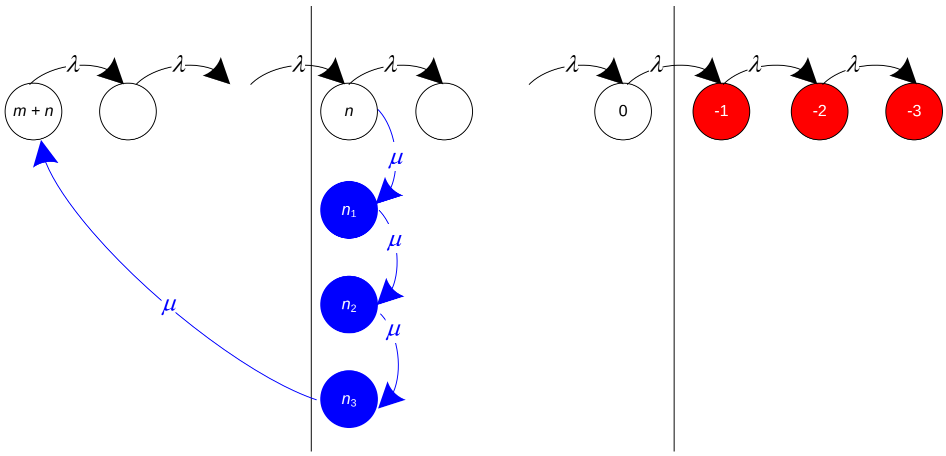

Figure 3: Step 2

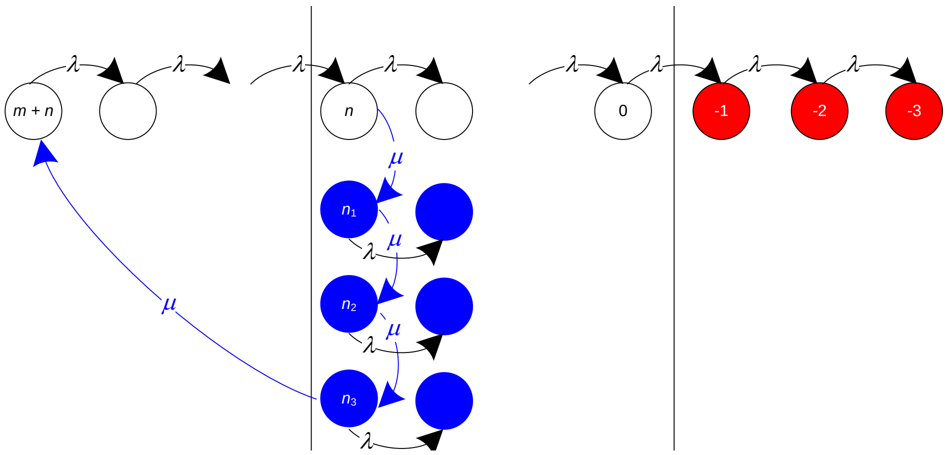

Figure 4: Step 3

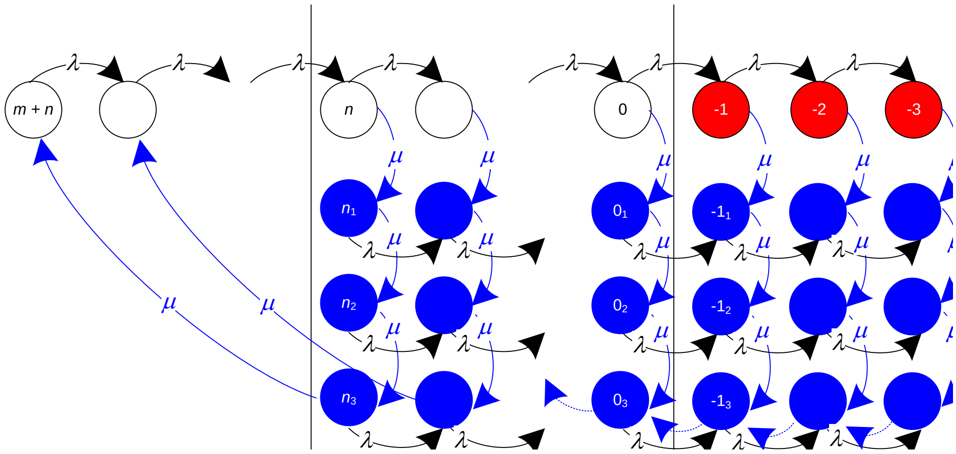

Figure 5: Step 4

APPENDIX - Solving the Markov differential equations

Let $\cdot\mathbf{A}$ be the transition matrix obtained from the Markov transition diagram, where each row correspond to departure from the corresponding sate, i.e., the row number, and each column correspond to arrival into the corresponding state, i.e., the column number. Further $\mathbf{P}(t)$ is the vector of probabilities for each state.

From Markov theory we have:

$$\mathbf{P}(t)\cdot\mathbf{A} = \dot{\mathbf{P}}(t) \tag{3}$$

Time dependent solution for the Markov process

To solve Equation (3) as a function of time we may use an analogy to ordinary differential equations in one dimension and we get:

$$\mathbf{P}(t) = \mathbf{P}(0)e^{t\mathbf{A}}$$

Although this is a very elegant solution, it is not very attractive since taking the exponential of a matrix is not that easy. Computer codes such as Matlab is required. We may, however, rewrite Equation (3) as:

$$\dot{\mathbf{P}}(t) = \lim_{\Delta t\to 0}\frac{\mathbf{P}(t+\Delta t)-\mathbf{P}(t)}{\Delta t} = \mathbf{P}(t)\cdot\mathbf{A}$$

yielding

$$\mathbf{P}(t+\Delta t) \approx \mathbf{P}(t)[\mathbf{A}\Delta t + \mathbf{I}] \tag{4}$$

where $\mathbf{I}$ is the identity matrix. This equation may now be used iteratively with a sufficient small time interval Δt and starting point $\mathbf{P}(0)$ to find the time dependent solution. Only simple matrix multiplication is required. Implementing a solution in for example VBA some considerations are required regarding the step length Δt. Choosing a too low value gives numerical problems and will also require longer computational time. Choosing a too high step length will cause the approximation in Equation (4) to be inaccurate. A rule of thumb will be to use a value of one tenth of the inverse value of the highest transition rate.

Note that in Markov analysis we usually only require the time-dependent solution for a limiting time period, and typically we would like to calculate $\mathbf{P}(t)$ at values $t = 0, \Delta t, 2\Delta t, \ldots$. Using Equation (4) is therefore attractive. To improve the approximation in Equation (4) we could use one "intermediate" point, i.e., we could use:

$$\mathbf{P}(t + \Delta t) \approx \mathbf{P}(t)[\mathbf{A}\Delta t/2 + \mathbf{I}][\mathbf{A}\Delta t/2 + \mathbf{I}] \tag{5}$$

and even improve by splitting into 2^n sub-intervals, yielding:

$$\mathbf{P}(t + \Delta t) \approx \mathbf{P}(t)\left[\mathbf{A}\Delta t/2^n + \mathbf{I}\right]^{2^n} \tag{6}$$

Note the similarity between Equation (6) and Equation (11.106) in the textbook. The advantage of Equation (6) is the calculation efficiency, i.e., we only need n matrix multiplications to reduce the step-length by a factor 2^n. Note that we only calculate $[\mathbf{A}\Delta t/2^n + \mathbf{I}]^{2^n}$ once in Equation (6), so we could afford double precision in that part of the calculations to increase the precision. It should be noted that there is still a trade-off between round-off errors and accuracy in the approximation in Equation (6), and a good choice of n would be in the range 4-6.

Steady state solution for the Markov process

In the long run we will have that $\dot{\mathbf{P}}(t) \rightarrow \mathbf{0}$ when $t \rightarrow \infty$, hence $\mathbf{P}(t) \cdot \mathbf{A} = \mathbf{0}$. We define the steady state probabilities by the vector $\mathbf{P} = [P_1, P_2, \ldots, P_r]$, where we have omitted the time dependency (t) to reflect that in the long run the state probabilities are not changing any more.

To solve the steady state equations we realize that the matrix $\mathbf{A}$ has not full rank due to the way have have established the diagonal elements. To overcome this problem we remove one (arbitrary) of the equations in the following set of equations:

$$[P_1, P_2, \ldots, P_r] \cdot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1r} \\ a_{21} & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rr} \end{bmatrix} = [0, 0, \ldots, 0]$$

and replace it by the following equation:

$$\sum_{j=1}^{r} P_j = 1$$

For example replacing the first equation gives:

$$[P_1, P_2, \ldots, P_r] \cdot \begin{bmatrix} 1 & a_{12} & \cdots & a_{1r} \\ 1 & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & a_{r2} & \cdots & a_{rr} \end{bmatrix} = [1, 0, \ldots, 0]$$

In matrix form we write:

$$\mathbf{P} \cdot \mathbf{A}_1 = \mathbf{b} \tag{7}$$

where **b** is a row vector of zeros except for the first element which equals one. Note that Equation (7) is not on standard form $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$. Transposing each side on the equal symbol in Equation (7) gives $\mathbf{A}_1^{\mathrm{T}} \cdot \mathbf{P}^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}}$ which could be solved by standard Gauss-Jordan elimination.

Note that in this section we have numbered the transition matrix from 1 to r, whereas in the spare part situation we typically have negative state numbers, and we need to introduce an offset number in the matrix manipulation.