

Exam in SF2863 Systems Engineering Monday January 13, 2014, 14.00–19.00

Examiner: Per Enqvist, tel. 790 62 98

Allowed tools: Pen/pencil, ruler and eraser. A formula sheet is handed out. No calculator is allowed!

Unless otherwise stated in the text, the problems should be solved by systematic methods, which do not become unrealistic for large problems. Explain carefully the methods you use, in particular if you use a method not taught in the course. Conclusions should always be motivated.

Note! Personal number must be written on the title page. Write only one exercise per sheet. Number the pages and write your name on each sheet!

25 points, including bonus points from the homeworks, is sufficient for passing the exam. For 23-24 points, a completion to a passing grade may be made within three weeks from the date when the results of the exam have been reported. Contact the examiner as soon as possible for such a completion.

The order of the exercises does not correspond to their difficulty level.

1. Frasse is comparing four different queueing models. The basic model is a M|M|1 queue with arrival intensity 10 customers per hour and service intensity $\mu_0 = 20$ customers per hour.

The second model is a $M|M|\infty$ queue with the same arrival intensity 10 customers per hour and service intensity μ_1 .

The third model is a M|M|1 queue with the same external arrival intensity 10 customers per hour and service intensity μ_2 , but where the customers leaving the M|M|1 queue with probability 1/2 leaves the system and with probability 1/2 goes back to the M|M|1 system.

The fourth model is a $M|M|\infty$ queue with the same external arrival intensity 10 customers per hour and service intensity μ_3 , but where the customers leaving the $M|M|\infty$ queue with probability 1/2 leaves the system and with probability 1/2 goes back to the $M|M|\infty$ system.

- 2. Frasse has over Christmas developed a strong addiction to julmust, and has no intention of breaking the addiction, on the contrary, he is busy planning his supply for the coming year and he is considering three different models.

 - (b) In the second model he plans his consumption of Julmust by the week, and he assumes that he knows exactly how much he will consume each week. He wants to make a plan for the next N weeks, and the consumption week k is d_k . The cost of purchasing the Julmust is 10 SEK per liter and there is a ordering cost of 100 SEK each time he orders a delivery, which then arrives immediately. To preserve the quality of the beverage he has developed a specialized dynamic refrigeration system that has a running cost of 7 SEK per liter of Julmust that is stored from one week to another. No shortage is tolerated!

Determine a dynamic programming recursion for the optimal cost, where you should carefully define all variables. Determine also the cost and policy for the last stage.

(c) In the third model he still plans his consumption of Julmust by the week, but he assumes that how much he will consume each week is stochastic. He wants to make a plan for the next N weeks, and the consumption week k is a stochastic variable D_k bounded by an upper limit 100 liter that Frasse has put up for himself.

The cost of purchasing the Julmust is 10 SEK per liter and there is a ordering cost of 100 SEK each time he orders a delivery, which then arrives immediately. To preserve the quality of the beverage he has developed a specialized dynamic refrigeration system that has a running cost of 7 SEK per liter of Julmust that is stored from one week to another. No shortage is tolerated!

Determine a dynamic programming recursion for the optimal cost, where you should carefully define all variables. Determine also the cost and policy for the last stage.

Solving the problem by total enumeration will not give any credits.

3. Frasse has made a new years resolution to spend more time with his companions. Unfortunately, his companions are not all getting along very well and there are four groupings, the Heathcliff family, Honest Harry and Harriet, The Nestor bunch and friendly Fritte. Frasse is willing to spend up to 7 days visiting his companions and wants to maximize the pleasure for them and himself. The pleasure for Frasse is a decreasing function 7-d, of d the total number of days he spends on his companions. According to Frasses estimations, the pleasure of the four groupings are given by the functions

 $-d_1^2+20d_1, \quad -d_2^2+19d_2, \quad -d_3^2+18d_3, \quad -d_4^2+17d_4,$

where d_i are the number of days he spends with companions *i*, and their total pleasure is the sum of the pleasures for each grouping. We assume that d_i belongs to $\{0, 1, 2, 3, 4, 5, 6, 7\}$ and that the functions are only defined for these values.

(a) Use the marginal allocation algorithm (and show that the conditions for using it are satisfied) to determine all efficient solutions for maximizing the pleasure of Frasse and his companions when the total number of days Frasse is socializing ranges in {1,...,7}.

- 4. Frasse is, already, two weeks in to the new year regretting his ambitious new years resolutions. Each year he makes either an ambitious or an easy resolution. By experience he has noted that if he manages to keep an ambitious resolution he experience a happiness of degree 10, out of ten possible, and if he keep an easy one he experiences a happiness of 5. If he fails to keep a resolution the happiness is 0. Frasse has also noted that the probability of keeping a resolution depends on if it is ambitious or easy and if he managed to keep the resolution the year before. If he kept the resolution the year before, the probability to keep an ambitious resolution is 0.5 and an easy one is 0.8 If he did not keep the resolution the year before, the probability to keep an ambitious resolution is 0.6.

 - (b) Again, consider the same problem, but now Frasse wants to add another aspect of his resolutions, the forgetting factor, he has noted that the happiness he will experience next year is only worth a factor 0.8 times the happiness he experiences this year.

Start with the policy to always make an easy resolution. Use the policy iteration algorithm to *improve on* the starting policy, if possible.