The Ladies Nylon Stocking Problem

- A woman has a collection of n nylon stockings.
- At each moment she must select m stockings to wear (where typically $m = 2$).
- The probability that a stocking whose age is x will wear during a time of length δ is $\lambda(x)\delta + o(\delta)$.
- She wants to wear her stockings in such a way as to maximize the expected time until she has only $m-1$ good stockings left in the drawer.

Again, this is a dynamic programming problem. Let $x = (x_1, \ldots, x_n)$ be the vector of stocking ages, and let S be the set of indices of good stockings. Consider $m = 2$. The optimality equation is

$$0 = \max_{\substack{i,j\in S,\\ i\neq j}} \Bigg[1 + \lambda(x_i)F(x, S\setminus\{i\}) + \lambda(x_j)F(x, S\setminus\{j\}) - (\lambda(x_i)+\lambda(x_j))\left(\frac{\partial}{x_i} + \frac{\partial}{x_j}\right) F(x,S) \Bigg]$$

where $F(x, S) = 0$ if the number of elements in S is less than 2.

That is,

$$F(x,S) = \max_{\substack{i,j\in S,\\ i\neq j}} \Big[\delta + \lambda(x_i)\delta F(x, S\setminus\{i\}) + \lambda(x_j)\delta F(x, S\setminus\{j\}) + \big(1 - \delta\lambda(x_i) - \delta\lambda(x_j)\big) F(x + \delta e_i + \delta e_j, S) + o(\delta) \Big]$$

where e_i is the vector with a 1 in the ith component and 0 in all other components.

By subtracting $F(x, S)$ from both sides, dividing by δ and letting $\delta \to 0$, this gives

$$0 = \max_{\substack{i,j\in S,\\ i\neq j}} \Bigg[1 + \lambda(x_i)F(x, S\setminus\{i\}) + \lambda(x_j)F(x, S\setminus\{j\}) - (\lambda(x_i)+\lambda(x_j))\left(\frac{\partial}{x_i} + \frac{\partial}{x_j}\right) F(x,S) \Bigg]$$

There is an answer to this problem in the case that $\lambda(x)$ is increasing or decreasing in x. That is, either the stockings become more likely to wear out as they get older, or they become less likely to wear out. The answer is that

She should always wear the m stockings which have the smallest hazard rates, i.e., those with the smallest values of $\lambda(x_i)$.

The form of this optimal policy was conjectured in 1959, but not proved until 1980, despite considerable attempts.

As both this and the digression on search for a moving object have shown, it can sometimes be very easy to write down dynamic programming equations, but very difficult to solve them.