Math Models of OR:
More on Equipment Replacement

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

December 2018
Outline

1. Equipment replacement problems requiring linear optimization

2. Backward recursion equations
Complicated cost functions

In some situations, the cost of using older equipment is not a simple function of the state.

We consider an example where it is necessary to solve a linear optimization problem in order to determine the cost of using older equipment.
A production problem

Say we have a production problem with $n$ products, each producing revenue $r(j)$, $j = 1, \ldots, n$, per unit sold.

Producing one unit of $j$ requires time $g(j)$ on a machine that decays; as the machine decays, it is shut for repairs more frequently.

The production amounts also have to satisfy the linear constraints $Ax \leq b$, and there are limits on demands for the items, so $0 \leq x_j \leq d_j$, $j = 1, \ldots, n$.

The time the machine is available is $h \ast q$, where $h$ is the total number of hours the machine would be up if it required no repairs, and $q$ is the proportion of the time that the machine is up.

The parameter $q$ is a function of the state $s$. 
Determining revenue at each stage

We use the index $k$ to refer to the stage, with each stage being one month.

We allow $r$, $b$, $d$, $g$, $h$, and $A$ to vary with the stage $k$.

Given the state $s$, the problem of maximizing revenue at stage $k$ can be calculated as

$$c_k(s) = \max_{x \in \mathbb{R}^n} \quad r_k^T x$$

subject to

$$A_k x \leq b_k$$

$$g_k^T x \leq h_k q(s)$$

$$0 \leq x \leq d_k$$
Outline

1. Equipment replacement problems requiring linear optimization

2. Backward recursion equations
Backward recursive equations

We still have a purchase price $p$ for a new machine, and we have trade-in values $r(s)$ for a used machine that is $s$ months old.

The possible decisions at stage $k$ are $z_k = \text{BUY}$ or $z_k = \text{NO-BUY}$.

We are now looking to maximize net revenue. We have

$$f_k(s) = \text{net revenue of optimal policy from the start of month } k \text{ to the end of the time horizon, given that the machine is } s \text{ months old at the start of month } k$$

$$f_k(s, z_k) = \text{net revenue of optimal policy from the start of month } k \text{ to the end of the time horizon, given that the machine is } s \text{ months old at the start of month } k, \text{ with the decision } z_k$$

Then

$$f_k(s) = \max \{ f_k(s, \text{BUY}), f_k(s, \text{NO-BUY}) \}$$
The recursion

If we make the decision \( z_k = \text{NO-BUY} \) then the immediate revenue in the upcoming month is \( c_k(s) \) calculated by solving the linear optimization problem; future net revenues are \( f_{k+1}(s + 1) \).

If we make the decision \( z_k = \text{BUY} \) then the net revenue in the upcoming month is \( c_k(0) - p + r(s) \), where \( c_k(0) \) is calculated by solving the linear optimization problem. Future net revenues are \( f_{k+1}(1) \) since the machine will be one month old at the start of the next month. The recursion for \( k \) before the time horizon is then

\[
\begin{align*}
  f_k(s, \text{NO-BUY}) & = c_k(s) + f_{k+1}(s + 1) \\
  f_k(s, \text{BUY}) & = c_k(0) - p + r(s) + f_{k+1}(1)
\end{align*}
\]

In the end, we want to find \( f_1(s_0) \), where \( s_0 \) is the initial age of the machine.
Using the recursion

We initialize with \( f_T(s) = r(s) \), where \( T \) is the time horizon, and \( r(s) \) is the trade-in value of the final machine. We then find \( f_{T-1}(s) \), \( f_{T-2}(s) \), \ldots, \( f_2(s) \), \( f_1(s_0) \) using backwards recursion, for all valid states \( s \).

The determination of each \( f_k(s) \) requires the solution of 2 linear optimization problems, one for the BUY decision and one for the NO-BUY decision.