The Satellite - A marginal allocation approach

Krister Svanberg and Per Enqvist
Optimization and Systems Theory,
KTH, Stockholm, Sweden.
{krille},{penqvist}@math.kth.se

1 Minimizing an integer-convex function of a single variable

Let $\mathcal{N}$ denote the set of natural numbers (non-negative integers), $\mathcal{N} = \{0, 1, 2, \dots, \}$, and let $I\!R$ denote the set of real numbers.

Further, let f be a given function from $\mathcal{N}$ to $I\!R$, and consider the optimization problem

$$\text{minimize} \quad f(x) \quad \text{subject to } x \in \mathcal{N}\,. \tag{1}$$

Def: The number $\hat{x} \in \mathcal{N}$ is an *optimal solution* to (1) if $f(\hat{x}) \leq f(x)$ for all $x \in \mathcal{N}$.
In this case, the *optimal value* of the problem (1) is given by $f(\hat{x})$.

For a completely general function f, (1) might be an impossible problem to solve (since there is an infinite number of numbers to compare.) But if f has certain properties, (1) could be solvable, perhaps even easily solvable. One such property will be discussed next.

For each number $x \in \mathcal{N}$, let

$$\Delta f(x) = f(x+1) - f(x). \tag{2}$$

Def: The function f from $\mathcal{N}$ to $I\!R$ is *integer-convex* if $\Delta f(x+1) \geq \Delta f(x)$ for all $x \in \mathcal{N}$.

Applying the Δ operator twice on f we have

$$\Delta^2 f(x) = \Delta f(x+1) - \Delta f(x) = f(x+2) - 2f(x+1) + f(x). \tag{3}$$

Alternative definition

Def: The function f from $\mathcal{N}$ to $I\!R$ is *integer-convex* if $\Delta^2 f(x) \geq 0$ for all $x \in \mathcal{N}$.

The following is a useful result.

Prop 1.1: If $f : I\!R^+ \to I\!R$ is a convex function on $I\!R^+$, then $f_{|\mathcal{N}} : \mathcal{N} \to I\!R$, defined by $f_{|\mathcal{N}}(n) = f(n)$ for all $n \in \mathcal{N}$, is integer-convex.

Proof: Take $x \in I\!R^+$ arbitrarily, then from the convexity of f it follows that

$$f(x) = f(\frac{1}{2}(x+1) + \frac{1}{2}(x-1)) \leq \frac{1}{2}f(x+1) + \frac{1}{2}f(x-1),$$

so

$$0 \le f(x+1) - 2f(x) + f(x-1) = (f(x+1) - f(x)) - (f(x) - f(x-1))$$

Now, if $x = n \in \mathcal{N}$, then $\Delta f_{|\mathcal{N}}(n) - \Delta f_{|\mathcal{N}}(n-1) \ge 0$, which shows that $f_{|\mathcal{N}}$ is integer-convex.

Prop 1.2: Assume that f is an integer-convex function from $\mathcal{N}$ to $I\!R$.
Then the number $\hat{x}$ is an optimal solution to problem (1) if and only if the following inequalities are satisfied:

$$\begin{aligned} \Delta f(\hat{x}-1) \le \; 0 \; &\le \Delta f(\hat{x}) \quad \text{if } \; \hat{x} > 0\,, \\ 0 \; &\le \Delta f(0) \quad \text{if } \; \hat{x} = 0\,. \end{aligned} \tag{4}$$

Proof:
If $\hat{x} > 0$ and $\Delta f(\hat{x}-1) > 0$ then $f(\hat{x}-1) < f(\hat{x})$ and $\hat{x}$ is not optimal.
If $\hat{x} \ge 0$ and $\Delta f(\hat{x}) < 0$ then $f(\hat{x}+1) < f(\hat{x})$ and $\hat{x}$ is not optimal.
If $\hat{x} = 0$ and $\Delta f(0) \ge 0$ then, since f is integer-convex, $\Delta f(x) \ge 0$ for all $x \ge 0$.
This implies that $f(0) \le f(1) \le f(2) \le \dots$, so that $\hat{x} = 0$ is optimal.
If $\hat{x} > 0$, $\Delta f(\hat{x}) \ge 0$ and $\Delta f(\hat{x}-1) \le 0$ then, since f is integer-convex,
$\Delta f(x) \ge 0$ for all $x \ge \hat{x}$ and $\Delta f(x) \le 0$ for all $x \le \hat{x} - 1$. This implies that
$f(\hat{x}) \le f(\hat{x}+1) \le f(\hat{x}+2) \le \dots$, and $f(\hat{x}) \le f(\hat{x}-1) \le \dots \le f(0)$,
so that $\hat{x}$ is optimal.

These optimality criteria can obviously be used for solving problem (1):
If $\Delta f(x) < 0$ then x is too small to be optimal, while if $\Delta f(x-1) > 0$ then x is too large.

2 Minimizing integer-convex separable functions of several variables

Let $\mathcal{N}^n$ denote the set of vectors $\mathbf{x} = (x_1, \dots, x_n)^\mathsf{T}$ with natural numbers as components.
Further, let f be a given function from $\mathcal{N}^n$ to $I\!R$, and consider the optimization problem

$$\text{minimize } \; f(\mathbf{x}) \; \text{ subject to } \mathbf{x} \in \mathcal{N}^n\,. \tag{5}$$

Def: The vector $\hat{\mathbf{x}} \in \mathcal{N}^n$ is an *optimal solution* to (5) if $f(\hat{\mathbf{x}}) \le f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}^n$.
In this case, the *optimal value* of the problem (5) is given by $f(\hat{\mathbf{x}})$.

Def: The function f from $\mathcal{N}^n$ to $I\!R$ is *separable* if f can be written

$$f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j), \tag{6}$$

where, for each $j = 1, \dots, n$, f_j is a function from $\mathcal{N}$ to $I\!R$.

For separable functions, there is a natural definition of integer-convexity:

Def: The separable function f in (6) is *integer-convex* if each f_j is integer-convex.

Prop 2.1: Assume that f is an integer-convex separable function from $\mathcal{N}^n$ to $I\!R$.
Then the vector $\hat{\mathbf{x}}$ is an optimal solution to problem (5) if and only if

the following inequalities are satisfied for each $j = 1, \ldots, n$:

$$\begin{aligned} \Delta f_j(\hat{x}_j - 1) \leq \ 0 \leq \Delta f_j(\hat{x}_j) \quad & \text{if } \ \hat{x}_j > 0\,, \\ 0 \leq \Delta f_j(0) \quad & \text{if } \ \hat{x}_j = 0\,. \end{aligned} \tag{7}$$

Proof: The statement follows from Prop 1.1, together with the observation that the sum $f(\mathbf{x}) = \sum_j f_j(x_j)$ is minimized if and only if each term $f_j(x_j)$ is minimized (since there is no coupling between the variables).

3 Efficient points for two integer-convex separable functions

Let f and g be two given integer-convex separable functions from $\mathcal{N}^n$ to $I\!R$:

$$f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j) \ \text{ and } \ g(\mathbf{x}) = \sum_{j=1}^{n} g_j(x_j). \tag{8}$$

Further assume that both $f(\mathbf{x})$ and $g(\mathbf{x})$ stands for quantities that we would like to be small, but that each $f_j(x_j)$ is strictly decreasing in x_j while each $g_j(x_j)$ is strictly increasing in x_j. This causes a conflict: Large values on the variables x_j will tend to make $f(\mathbf{x})$ small, which is desirable, but $g(\mathbf{x})$ large, which is undesirable. Small values on the variables x_j will tend to make $g(\mathbf{x})$ small, which is desirable, but $f(\mathbf{x})$ large, which is undesirable. This section deals with how to compromise between these conflicting goals.

To summarize the assumptions:

$$\begin{aligned} \Delta f_j(x_j) \leq \Delta f_j(x_j{+}1) < 0 \quad & \text{for all j and all } x_j \in \mathcal{N}, \\ 0 < \Delta g_j(x_j) \leq \Delta g_j(x_j{+}1) \quad & \text{for all j and all } x_j \in \mathcal{N}. \end{aligned} \tag{9}$$

Let $g^{\max}$ be a given upper bound on the acceptable values of $g(\mathbf{x})$: Points $\mathbf{x}$ with $g(\mathbf{x}) > g^{\max}$ are assumed to be unacceptable (e.g. too expensive). Further let

$$X = \{\, \mathbf{x} \in \mathcal{N}^n \mid g(\mathbf{x}) \leq g^{\max} \}. \tag{10}$$

Since each function g_j is strictly increasing and integer-convex, the set X contains a finite number of vectors $\mathbf{x}$, but this finite number may in practical applications be extremely large.

Def: The vector $\hat{\mathbf{x}} \in X$ is an *efficient solution* corresponding to the above setting if there are constants $\alpha > 0$ and $\beta > 0$ such that $\hat{\mathbf{x}}$ is an optimal solution to the following optimization problem in $\mathbf{x}$:

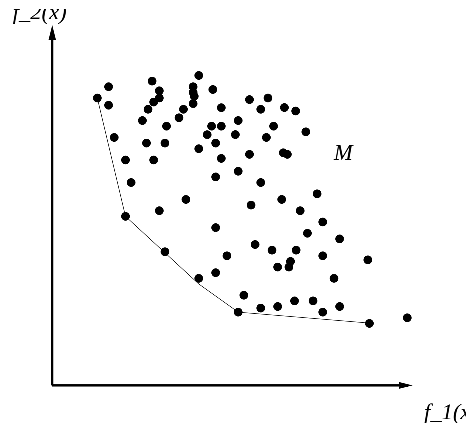
$$\text{minimize } \ \alpha\, g(\mathbf{x}) + \beta f(\mathbf{x}) \ \text{ subject to } \mathbf{x} \in X\,. \tag{11}$$

Next, we will give a natural geometric interpretation of the efficient solutions defined above.

Let

$$M = \{\, (g(\mathbf{x}), f(\mathbf{x})) \mid \mathbf{x} \in X\} \subset I\!R^2. \tag{12}$$

This set M contains a finite (but possibly extremely large) number of points in $I\!R^2$. To get a picture of M, we may imagine that the points in M are plotted in a coordinate system where the horizontal axis shows $g(\mathbf{x})$ and the vertical axis shows $f(\mathbf{x})$.



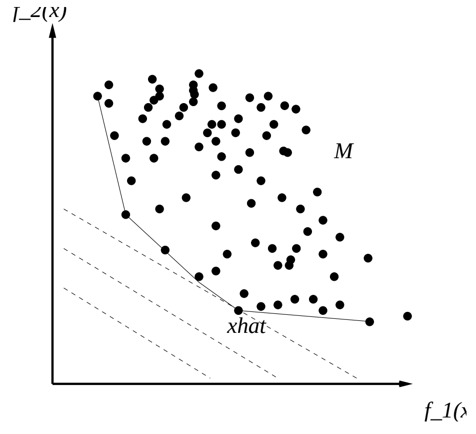
The *convex hull* of M is defined as the smallest *convex* set in $\mathbb{R}^2$ which contains M. Geometrically, the convex hull of M is what you get if you "stretch a rope" around M.

The *efficient curve* for the current setting is the piecewise linear curve that constitutes the "south-western boundary" of the convex hull of M. Points $(g(\mathbf{x}), f(\mathbf{x})) \in M$ which lie on this efficient curve are called *efficient points*, and the corresponding vectors $\mathbf{x}$ are in fact the *efficient solutions* defined above. Here is an argument to motivate this last statement:

From a two-dimensional figure where the points of M are plotted and the convex hull of M is drawn, it follows that a point $(\hat{\xi}, \hat{\eta}) = (g(\hat{\mathbf{x}}), f(\hat{\mathbf{x}})) \in M$ belongs to the "southwestern boundary" of the convex hull of M if and only if there are constants $\alpha > 0$ and $\beta > 0$ such that $(\hat{\xi}, \hat{\eta})$ is an optimal solution to the following optimization problem in ξ and η:

$$\text{minimize} \;\; \alpha\,\xi + \beta\,\eta \;\; \text{subject to } (\xi, \eta) \in M. \tag{13}$$

But this problem (13) is equivalent to the the above problem (11) in $\mathbf{x}$.



Typically, we are interested in determining the efficient curve for a given situation, with given functions f and g. It turns out that even if the number of points in M is extremely large, it is surprisingly easy to determine the efficient curve! We will describe below how this is done, but first some preparatory results.

Prop 3.1: The vector $\hat{\mathbf{x}} \in X$ minimizes $\alpha\, g(\mathbf{x}) + \beta f(\mathbf{x})$ subject to $\mathbf{x} \in X$ if and only if the following conditions are satisfied for each $j = 1, \ldots, n$:

$$\frac{-\Delta f_j(\hat{x}_j)}{\Delta g_j(\hat{x}_j)} \le \frac{\alpha}{\beta} \le \frac{-\Delta f_j(\hat{x}_j - 1)}{\Delta g_j(\hat{x}_j - 1)} \quad \text{if } \hat{x}_j > 0\,, \tag{14}$$

$$\frac{-\Delta f_j(0)}{\Delta g_j(0)} \le \frac{\alpha}{\beta} \quad \text{if } \hat{x}_j = 0\,, \tag{15}$$

Proof: Just replace $f(\mathbf{x})$ by $\alpha\, g(\mathbf{x}) + \beta f(\mathbf{x})$ in Prop 2.2

From this proposition, together with the above definition of an efficient solution, we get the following criteria for deciding whether a vector $\hat{\mathbf{x}}$ is an efficient solution or not:

Prop 3.2: $\hat{\mathbf{x}} \in X$ is an efficient solution if and only if there are constants $\alpha > 0$ and $\beta > 0$ such that the conditions (14)–(15) are satisfied for each $j = 1, \ldots, n$.

The following result shows that each efficient solution is in fact an optimal solution to two particular optimization problems.

Prop 3.3: Assume that $\hat{\mathbf{x}} \in X$ is an efficient solution, and let $\hat{g} = g(\hat{\mathbf{x}})$ and $\hat{f} = f(\hat{\mathbf{x}})$. Then $\hat{\mathbf{x}}$ is an optimal solution to both the following optimization problems:

$$\text{minimize } f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) \le \hat{g}\,,\ \mathbf{x} \in X. \tag{16}$$

$$\text{minimize } g(\mathbf{x}) \text{ subject to } f(\mathbf{x}) \le \hat{f}\,,\ \mathbf{x} \in X. \tag{17}$$

Proof: If $\hat{\mathbf{x}}$ is an efficient solution then there are constants $\alpha > 0$ and $\beta > 0$ such that

$$\alpha\, g(\hat{\mathbf{x}}) + \beta f(\hat{\mathbf{x}}) \le \alpha\, g(\mathbf{x}) + \beta f(\mathbf{x}), \quad \text{for all } \mathbf{x} \in X. \tag{18}$$

First, take an arbitrary $\mathbf{x} \in X$ such that $g(\mathbf{x}) \le \hat{g}$. Then, according to (18), it holds that

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}) \le (\alpha/\beta)(g(\mathbf{x}) - g(\hat{\mathbf{x}})) = (\alpha/\beta)(g(\mathbf{x}) - \hat{g}) \le 0, \tag{19}$$

which implies that $\hat{\mathbf{x}}$ is an optimal solution to (16).

Next, take an arbitrary $\mathbf{x} \in X$ such that $f(\mathbf{x}) \le \hat{f}$. Then, according to (18), it holds that

$$g(\hat{\mathbf{x}}) - g(\mathbf{x}) \le (\beta/\alpha)(f(\mathbf{x}) - f(\hat{\mathbf{x}})) = (\beta/\alpha)(f(\mathbf{x}) - \hat{f}) \le 0, \tag{20}$$

which implies that $\hat{\mathbf{x}}$ is an optimal solution to (17).

4 Marginal allocation algorithm for generating efficient solutions

We will now describe a surprisingly simple algorithm for determining the efficient curve described above, but first we repeat the assumptions that f is integer-convex and strictly decreasing in each variable, while g is integer-convex and strictly increasing in each variable.

The algorithm start from $\mathbf{x}^{(0)} = \mathbf{0}$ (which is an efficient solution) and generates efficient solutions $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots$ "from left to right", which means that each new generated point has a higher

value on $g(\mathbf{x})$ but a lower value on $f(\mathbf{x})$ than the previously generated point.
Throughout the algorithm $\mathbf{x}^{(k)}$ denotes the k:th generated efficient solution.
The algorithm terminates when there is no longer any efficient solution with $g(\mathbf{x}) \leq g^{\max}$.

Step 0:
Generate a table with n columns as follows. For $j = 1, \dots, n$, fill the j:th column from the top and down with the quotients $-\Delta f_j(0)/\Delta g_j(0)$, $-\Delta f_j(1)/\Delta g_j(1)$, $-\Delta f_j(2)/\Delta g_j(2)$, etc...
(A moderate number of quotients will suffice, additional quotients can be calculated as needed.)
Note that the quotients are positive and strictly decreasing in each column.
Set $k = 0$, $\mathbf{x}^{(0)} = (0, \dots, 0)^{\mathsf{T}}$, $g(\mathbf{x}^{(0)}) = g(\mathbf{0})$ and $f(\mathbf{x}^{(0)}) = f(\mathbf{0})$.
Let all the quotients in the table be *uncanceled.*

Step 1:
Select the *largest uncanceled* quotient in the table (if there are several equally large, choose one of these arbitrarily). *Cancel* this quotient and let ℓ be the number of the column from which the quotient was canceled.

Step 2:
Let $k := k + 1$. Then let $x_\ell^{(k)} = x_\ell^{(k-1)} + 1$ and $x_j^{(k)} = x_j^{(k-1)}$ for all $j \neq \ell$.
Further, calculate $f(\mathbf{x}^{(k)}) = f(\mathbf{x}^{(k-1)}) + \Delta f_\ell(x_\ell^{(k-1)})$ and $g(\mathbf{x}^{(k)}) = g(\mathbf{x}^{(k-1)}) + \Delta g_\ell(x_\ell^{(k-1)})$.
If $g(\mathbf{x}^{(k)}) \geq g^{\max}$, terminate the algorithm. Otherwise, go to Step 1.

Note that each generated solution $\mathbf{x}^{(k)}$ differs from the previously generated solution $\mathbf{x}^{(k-1)}$ in just one component. The name of the algorithm stems from the fact that

$$\frac{-\Delta f_j(x_j)}{\Delta g_j(x_j)} = \frac{\text{decrease in } f(\mathbf{x}) \text{ if } x_j \text{ is increased by 1}}{\text{increase in } g(\mathbf{x}) \text{ if } x_j \text{ is increased by 1}}.$$

Hence, in each step of the algorithm, we increase the x_j which gives marginally the largest decrease in $f(\mathbf{x})$ per increase in $g(\mathbf{x})$.

Prop 4.1: Each generated solution $\mathbf{x}^{(k)}$ is an efficient solution.

Proof: Consider a given generated solution $\mathbf{x}^{(k)}$. Choose $\alpha > 0$ and $\beta > 0$ such that $\alpha/\beta =$ the quotient that was canceled in Step 1 immediately before $\mathbf{x}^{(k)}$ was generated in Step 2. Then all canceled quotients are $\geq \alpha/\beta$, while all uncanceled quotients are $\leq \alpha/\beta$. But then $\mathbf{x}^{(k)}$ and α/β satisfy conditions (14)–(15) for each $j = 1, \dots, n$, which implies that $\mathbf{x}^{(k)}$ is an efficient solution.

Prop 4.2: Assume that all quotients $-\Delta f_j(x_j)/\Delta g_j(x_j)$ in the original table are different.
Then the algorithm generates all efficient solutions which satisfy $g(\mathbf{x}) \leq g^{\max}$.

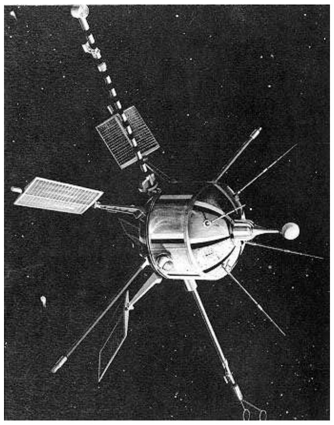
Proof: Assume that $\hat{\mathbf{x}}$ is an efficient solution. Then the conditions (14)–(15) are satisfied for some α/β. But if all quotients are different, then α/β can be perturbed such that all inequalities in (14)–(15) becomes strict inequalities, so that, for each $j = 1, \dots, n$,

$$\frac{-\Delta f_j(x_j)}{\Delta g_j(x_j)} < \frac{\alpha}{\beta} \text{ if } x_j \geq \hat{x}_j\,, \quad \text{and} \quad \frac{-\Delta f_j(x_j)}{\Delta g_j(x_j)} > \frac{\alpha}{\beta} \text{ if } x_j < \hat{x}_j\,. \tag{21}$$

These conditions determine $\hat{\mathbf{x}}$ uniquely. However, this solution will actually be generated by the algorithm in the stage where the latest canceled quotient is $> \alpha/\beta$, while the largest quotient that has not yet been canceled is $< \alpha/\beta$.

5 The satellite problem

We will consider a satellite that is going to be launched into space and we will consider the reliability of a system that will be in operation for a time period of five years. The system relies on five different subprocesses, and each subprocess needs to be operative in order for the whole system to work properly. Each subprocess is dependent on a certain component on the satellite to work. These components fail with some probability, and to increase the reliability of the satellite, the components can be duplicated. The downside with duplicating components is the extra weight this puts on the satellite.



The weight w_k for each of the components corresponding to subprocess k are given by

$$w = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 2 & 3 \end{bmatrix}. \tag{22}$$

The reliability of the process is increased by adding redundancy in terms of adding more components, and we would like to know how one optimally should select these extra components in order to maximize the reliability. In particular we will consider in which order you should add the components if you add them one by one and get a feeling for the relation between increased reliability and added weight.

We will study two different approaches where the redundancy imposed to the system is modelled in different ways. These corresponds to different assumptions on if the components fail while being active or passive.

5.1 The Active Redundance Problem

In the first case we consider active redundance. The component of type k is duplicated in a total of x_k number of components.

We assume that components of type k fail according to some distribution so that the failure times are independent and exponentially distributed with intensity λ_k.

This means that each of the components fail with some probability $p_k = 1 - e^{\lambda_k T}$, where $T = 5$ years is the mission time. In this example we assume that the probabilities are given by

$$p = \left[\begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \right] = \left[\begin{array}{ccccc} 0.2 & 0.1 & 0.3 & 0.15 & 0.05 \end{array} \right]. \tag{23}$$

The physical interpretation is that all components of the same type run the same risk of failing, no matter if it is used or not, and this failure risk would typically depend on stress caused at the launch or radiation in the space environment.

Using the independence assumption, the probability that subprocess k is not working is then the probability that none of the x_k components of type k are working, *i.e.*, $p_k^{x_k}$. The probability that subprocess k is working is then $1 - p_k^{x_k}$, and the probability that the whole process is working is

$$P_{\text{working}}(x) = \prod_{k=1}^{5} \left(1 - p_k^{x_k}\right). \tag{24}$$

We would like to maximize this function using the theory explained above, but this function is not a separable function. Taking the logarithm of $P_{\text{working}}(x)$ gives a separable function, and to get a minimization problem we change the sign and consider the function:

$$f(x) = -\log P_{\text{working}}(x) = \sum_{k=1}^{5} -\log\left(1 - p_k^{x_k}\right) = \sum_{k=1}^{5} f_k(x_k), \tag{25}$$

where $f_k(x_k) = -\log(1 - p_k^{x_k})$.

Is $f_k(x_k)$ an integer-convex function ?

We use Proposition 1.1. To obtain easier derivatives, let $p_k = e^{-a_k}$.

Then if $F(x) = -\log(1 - e^{-ax})$, it is easy to see that

$$F'(x) = -\frac{ae^{-ax}}{1 - e^{-ax}}, \qquad F''(x) = \frac{a^2 e^{-2ax}}{(1 - e^{-ax})^2}. \tag{26}$$

The second derivative $F''(x)$ is strictly positive and therefore the function $F(x)$ is convex and from Proposition 1.1 it follows that $f_k(x_k)$ is integer-convex for each k, hence f is integer-convex.

It is also easy to see that the f_k are strictly decreasing.

Define also $g(x)$

$$g(x) = \sum_{j=0}^{k} w_j x_j = \sum_{j=0}^{k} g_j(x_j), \tag{27}$$

which determines the weight of all the components that are placed on the satellite. Since $\Delta g_k(x_k) = w_k > 0$, it is trivially a strictly increasing and integer-convex seperable function.

Now we would like to solve the problem using the marginal allocation algorithm.

First we need to compute the functions $f_k(x_k)$ for a number of plausible values, this gives us the following table

k	$f_1(k)$	$f_2(k)$	$f_3(k)$	$f_4(k)$	$f_5(k)$
1	0.223143551314210	0.105360515657826	0.356674943938732	0.162518929497775	0.051293294387551
2	0.040821994520255	0.010050335853501	0.094310679471241	0.022756987122616	0.002503130218118
3	0.008032171697264	0.001000500333584	0.027371196796132	0.003380708159478	0.000125007813151
4	0.001601281366974	0.000100005000333	0.008132983230189	0.000506378187796	0.000006250019531
5	0.000320051210925	0.000010000050000	0.002432957241703	0.000075940383398	0.000000312500049
6	0.000064002048087	0.000001000000500	0.000729265849711	0.000011390689874	0.000000015625000

Then it is easy to determine the marginal costs

k	$-\frac{\Delta f_1(k)}{w_1}$	$-\frac{\Delta f_2(k)}{w_2}$	$-\frac{\Delta f_3(k)}{w_3}$	$-\frac{\Delta f_4(k)}{w_4}$	$-\frac{\Delta f_5(k)}{w_5}$
1	0.182321556793955	0.047655089902162	0.262364264467491	0.069880971187579	0.016263388056477
2	0.032789822822991	0.004524917759959	0.066939482675109	0.009688139481569	0.000792707468322
3	0.006430890330290	0.000450247666625	0.019238213565943	0.001437164985841	0.000039585931207
4	0.001281230156049	0.000045002475167	0.005700025988486	0.000215218902199	0.000001979173161
5	0.000256049162838	0.000004500024750	0.001703691391992	0.000032274846762	0.000000098958350

We can order the elements in this matrix

k	$-\frac{\Delta f_1(k)}{w_1}$	$-\frac{\Delta f_2(k)}{w_2}$	$-\frac{\Delta f_3(k)}{w_3}$	$-\frac{\Delta f_4(k)}{w_4}$	$-\frac{\Delta f_5(k)}{w_5}$
1	$\boxed{2}$	$\boxed{5}$	$\boxed{1}$	$\boxed{3}$	$\boxed{8}$
2	$\boxed{6}$	0.004524917759959	$\boxed{4}$	$\boxed{9}$	0.000792707468322
3	$\boxed{10}$	0.000450247666625	$\boxed{7}$	0.001437164985841	0.000039585931207
4	0.001281230156049	0.000045002475167	0.005700025988486	0.000215218902199	0.000001979173161
5	0.000256049162838	0.000004500024750	0.001703691391992	0.000032274846762	0.000000098958350

The marginal allocation algorithm starts with

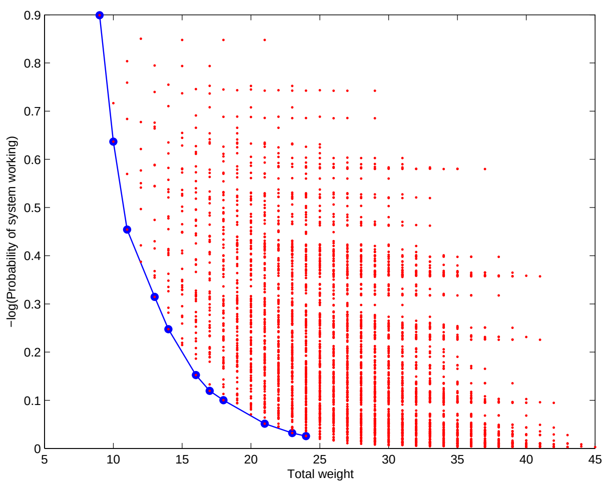
$$x^{(0)} = \left[\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array}\right] = \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \end{array}\right], \tag{28}$$

which corresponds to a weight of 9 units and a probability of the system to be operative during the whole mission time of 40.7%.

By adding the components, one by one, in the order determined by the marginal allocation algorithm, gives the following values

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$P(x^{(n)})$	$w \cdot x^{(n)}$
0	1	1	1	1	1	0.4069	9
1	1	1	2	1	1	0.5291	10
2	2	1	2	1	1	0.6349	11
3	2	1	2	2	1	0.7301	13
4	2	1	3	2	1	0.7807	14
5	2	2	3	2	1	0.8587	16
6	3	2	3	2	1	0.8874	17
7	3	2	4	2	1	0.9046	18
8	3	2	4	2	2	0.9498	21
9	3	2	4	3	2	0.9684	23
10	4	2	4	3	2	0.9747	24

These values determine the efficient curve depicted in blue in the figure below. For illustration all different values of x with components between 1 and 5 have been determined and the corresponding weights and probabilities are depicted with red dots.



5.2 The Passive Redundance Problem

In the second case we consider passive redundance. The component of type k is duplicated in a total of x_k number of components.

We assume that the active component of type k fail according to some distribution so that the failure times are independent and exponentially distributed with intensity λ_k.

This means that the components fail one by one, each with some exponential life length with intensity λ_k. The number of failures Y in a time interval of length T, when the time between failures is exponentially distrubuted with parameter λ, has a Poisson distribution with parameter λT, *i.e.* Y has the probability function

$$P(Y = k) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}, \qquad k = 0, 1, \cdots. \tag{29}$$

Here, $T = 5$ years is the mission time.

The physical interpretation is now that components of the same type run the risk of failing only if it is in use, and this failure risk would typically depend on wear and tear.

The probability that subprocess k works during the whole mission time T, given that there are x_k available components of type k on the satellite is then

$$P_{\text{subsystem k working}}(x_k) = \sum_{k=0}^{x_k - 1} \frac{(\lambda_k T)^k}{k!} e^{-\lambda_k T}. \tag{30}$$

In particular, $P_{\text{subsystem k working}}(1) = e^{-\lambda_k T} = 1 - p_k$ is the same probability as for the active redundance problem. If we have only one component of a certain type the two approaches are equal. We assume that the λ_k are such that the probabilities p_k are given as before by

$$p = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.1 & 0.3 & 0.15 & 0.05 \end{bmatrix}. \tag{31}$$

The probability that the whole process is working is

$$P_{\text{working}}(x) = \prod_{k=1}^{5} P_{\text{subsystem k working}}(x_k) \tag{32}$$

As before, this function is not a separable function. Again, taking the logarithm of $P_{\text{working}}(x)$ gives a separable function, and to get a minimization problem we change the sign and consider the function:

$$f(x) = -\log P_{\text{working}}(x) = \sum_{k=1}^{5} \lambda_k T - \log\left(1 + \lambda_k T + \cdots + \frac{(\lambda_k T)^{x_k - 1}}{(x_k - 1)!}\right) = \sum_{k=1}^{5} f_k(x_k), \tag{33}$$

where $f_k(x_k) = \lambda_k T - \log\left(1 + \lambda_k T + \cdots + \frac{(\lambda_k T)^{x_k-1}}{(x_k-1)!}\right)$.

Is $f_k(x_k)$ an integer-convex function ?

This time it is not convenient to use Proposition 1.1, so we use the definition instead.

Denote $S_k(x) = 1 + \lambda_k T + \cdots + \frac{(\lambda_k T)^{x-1}}{(x-1)!}$, then

$$\Delta f_k(x_k) = \log S_k(x_k - 1) - \log S_k(x_k) \tag{34}$$

and

$$\Delta^2 f_k(x_k) = -\log S_k(x_k - 1) + 2\log S_k(x_k) - \log S_k(x_k + 1) = \log \frac{S_k(x_k)^2}{S_k(x_k - 1) S_k(x_k + 1)}$$

which is larger than zero iff $S_k(x_k - 1) S_k(x_k + 1) \le S_k^2(x_k)$, *i.e.*

$$S_k^2(x_k) - \underbrace{\left(S_k(x_k) - \frac{(\lambda_k T)^{x_k-1}}{(x_k - 1)!}\right)}_{S_k(x_k-1)} \underbrace{\left(S_k(x_k) + \frac{(\lambda_k T)^{x_k}}{x_k!}\right)}_{S_k(x_k+1)} \ge 0$$

which reduces to

$$S_k(x_k) \left(\frac{(\lambda_k T)^{x_k-1}}{(x_k - 1)!}\right) \left(1 - \frac{\lambda_k T}{x_k}\right) + \frac{(\lambda_k T)^{2x_k - 1}}{(x_k - 1)! x_k!} \ge 0.$$

This clearly holds if $\frac{\lambda_k T}{x_k} \le 1$. For the data in our case, it holds that $\lambda_k T < 1$, and therefore $f_k(x_k)$ is integer-convex for each k, hence $f(x)$ is integer-convex. From (34) it follows that f_k is a strictly decreasing function.

Define also $g(x)$ as in (27).

Now we would like to solve the problem to minimize $\alpha f(x) + \beta g(x)$ using the marginal allocation algorithm.

First we need to compute the functions $f_k(x_k)$ for a number of plausible values, this gives us the following table

k	$f_1(k)$	$f_2(k)$	$f_3(k)$	$f_4(k)$	$f_5(k)$
1	0.223143551314210	0.105360515657826	0.356674943938732	0.162518929497775	0.051293294387551
2	0.021719325116041	0.005188975384976	0.051638131987610	0.011929787712028	0.001272179208748
3	0.001569171657521	0.000180174921625	0.005818478333388	0.000633839721685	0.000021644592734
4	0.000086478653364	0.000004720195748	0.000507948531201	0.000025532869481	0.000000276837778
5	0.000003830012198	0.000000099111735	0.000035781922905	0.000000825345840	0.000000002835107
6	0.000000141669160	0.000000001736006	0.000002108440661	0.000000022268025	0.000000000024207

Then it is easy to determine the marginal costs

k	$-\frac{\Delta f_1(k)}{w_1}$	$-\frac{\Delta f_2(k)}{w_2}$	$-\frac{\Delta f_3(k)}{w_3}$	$-\frac{\Delta f_4(k)}{w_4}$	$-\frac{\Delta f_5(k)}{w_5}$
1	0.201424226198169	0.050085770136425	0.305036811951123	0.075294570892873	0.016673705059601
2	0.020150153458520	0.002504400231675	0.045819653654222	0.005647973995172	0.000416844872005
3	0.001482693004157	0.000087727362939	0.005310529802187	0.000304153426102	0.000007122584985
4	0.000082648641166	0.000002310542006	0.000472166608296	0.000012353761820	0.000000091334224
5	0.000003688343038	0.000000048687865	0.000033673482244	0.000000401538908	0.000000000936967

We can order the elements in this matrix

k	$-\frac{\Delta f_1(k)}{w_1}$	$-\frac{\Delta f_2(k)}{w_2}$	$-\frac{\Delta f_3(k)}{w_3}$	$-\frac{\Delta f_4(k)}{w_4}$	$-\frac{\Delta f_5(k)}{w_5}$
1	$\boxed{2}$	$\boxed{4}$	$\boxed{1}$	$\boxed{3}$	$\boxed{7}$
2	$\boxed{6}$	$\boxed{10}$	$\boxed{5}$	$\boxed{8}$	0.000416844872005
3	0.001482693004157	0.000087727362939	$\boxed{9}$	0.000304153426102	0.000007122584985
4	0.000082648641166	0.000002310542006	0.000472166608296	0.000012353761820	0.000000091334224
5	0.000003688343038	0.000000048687865	0.000033673482244	0.000000401538908	0.000000000936967

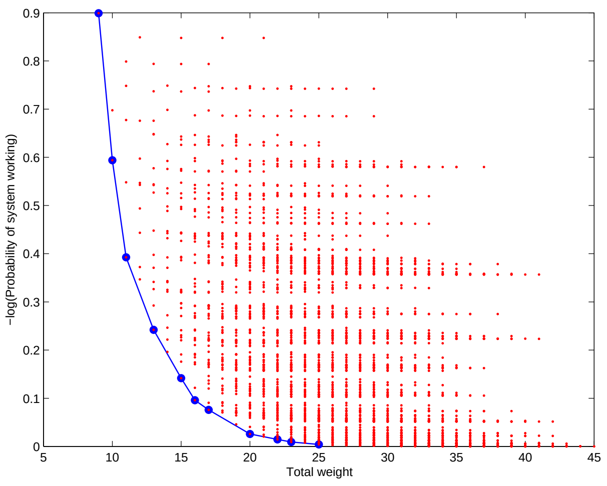
The marginal allocation algorithm starts as before with

$$x^{(0)} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{35}$$

By adding the components, one by one, in the order determined by the marginal allocation algorithm, gives the following values

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$P(x^{(n)})$	$w \cdot x^{(n)}$
0	1	1	1	1	1	0.4069	9
1	1	1	2	1	1	0.5521	10
2	2	1	2	1	1	0.6754	11
3	2	1	2	2	1	0.7851	13
4	2	2	2	2	1	0.8578	15
5	2	2	3	2	1	0.9085	16
6	3	2	3	2	1	0.9270	17
7	3	2	3	2	2	0.9746	20
8	3	2	3	3	2	0.9856	22
9	3	2	4	3	2	0.9909	23
10	3	3	4	3	2	0.9958	25

These values determine the efficient curve depicted in blue in the figure below. For illustration all different values of x with components between 1 and 5 have been determined and the corresponding weights and probabilities are depicted with red dots.



−log(Probability of system working)
Total weight
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
5
10
15
20
25
30
35
40
45