Assignment 2: Maintenance Planning

Simon Sigurdhsson

26 april 2010

1 Part one

1.1 The `uh-stor.mod` implementation

1.2 The `uh-small.mod` implementation

$$\text{Both binary} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{None binary} = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 0 \\ 0.5 & 0 \\ 0.5 & 0.5 \end{bmatrix} \quad \text{With cgcut} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2 Part two

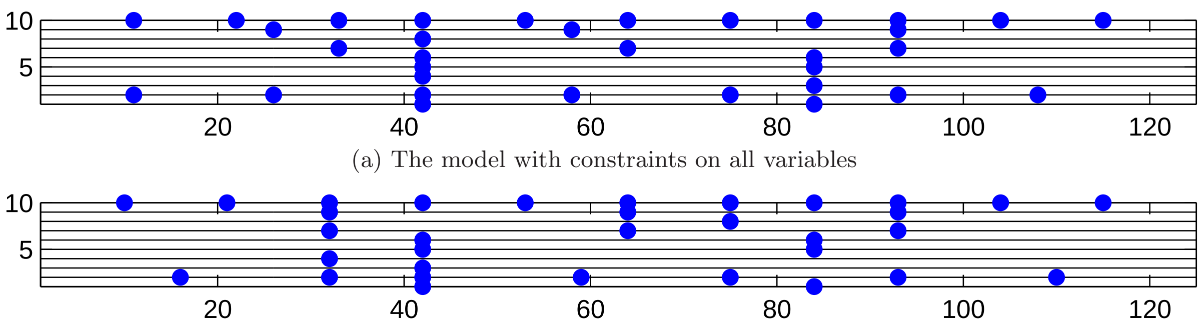
2.1 Varying fixed maintenance cost

2.2 Increasing component usage time

By simply weighing the objective function with the factor $(T - t + 1)$, as seen in (1), we shift maintenances to occur later in time, thus utilizing the components for as long as possible before replacing them. One quite relevant drawback here is that we have to manually recalculate the real cost of the maintenance schedule after optimization, but this is an easy task when all the output is known.

$$\sum_{t=1}^{T} \left(\sum_{i \in \mathcal{N}} (T - t + 1) c_{it} x_{it} + d_t z_t \right) \tag{1}$$

(analysis of output and comments regarding CPU time de/increase and other potential problems here)



(a) The model with constraints on all variables

(b) The model with constraints only on the z_t variables

Figur 1: The model changes significantly when constraints are relaxed

3 Part three

3.1 Time spent in the presolver

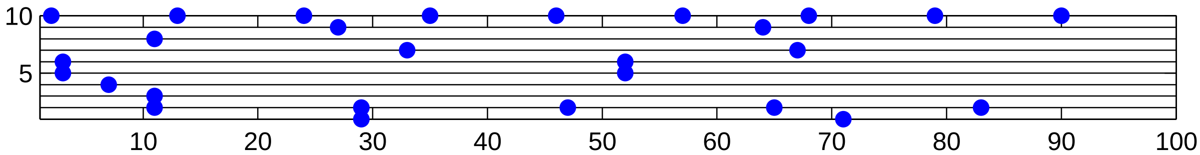
4 Part five

Modifying the model to include a minimum remaining life at time T is actually quite trivial; we add a constraint (2). This makes sure that there is at least one replacement of each component in the last $T_i - r$ time steps, thus making sure that every component has a life time of at least r at time T.

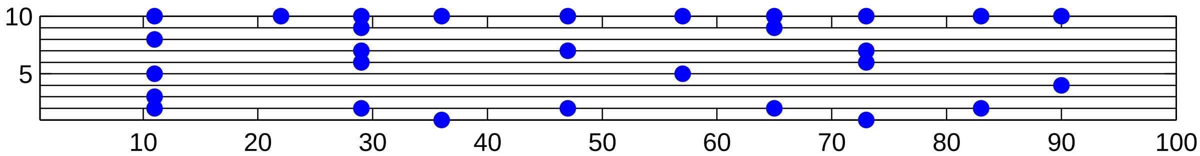
$$\sum_{t=T-(T_i-r)}^{T} x_{it} \geq 1, \quad i \in \mathcal{N} \tag{2}$$

(...)

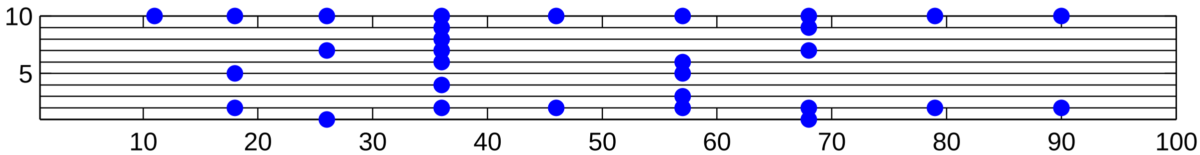
In this model, the relevant values of r are integer values such that $0 \leq r < 11$, since the component with the shortest life time has a life time of 11 time steps. Increasing r beyond this value yields an unsolvable (even undefined) model. Negative values of r are of course never relevant, and integer constraints on other variables imply that r also has to be an integer.



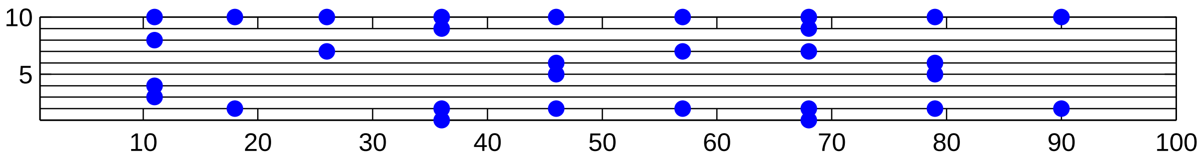
(a) $d_t = 0$, solved in 0.02 seconds



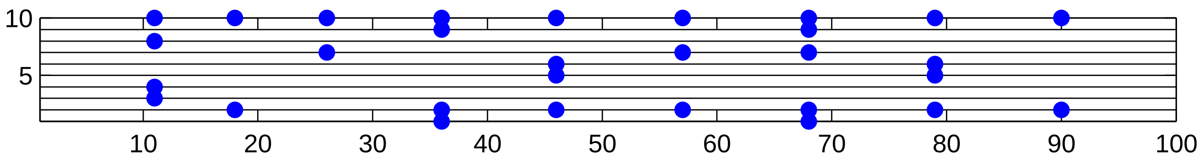
(b) $d_t = 25$, solved in 4.4 seconds



(c) $d_t = 50$, solved in 16.9 seconds

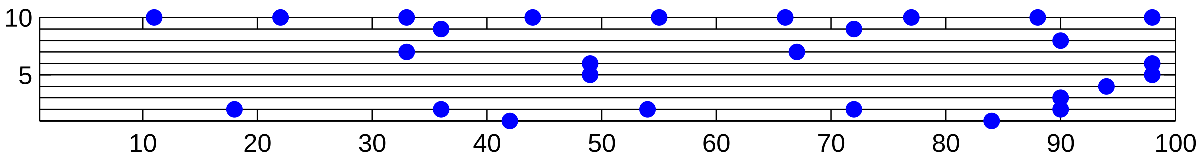


(d) $d_t = 75$, solved in 18.1 seconds

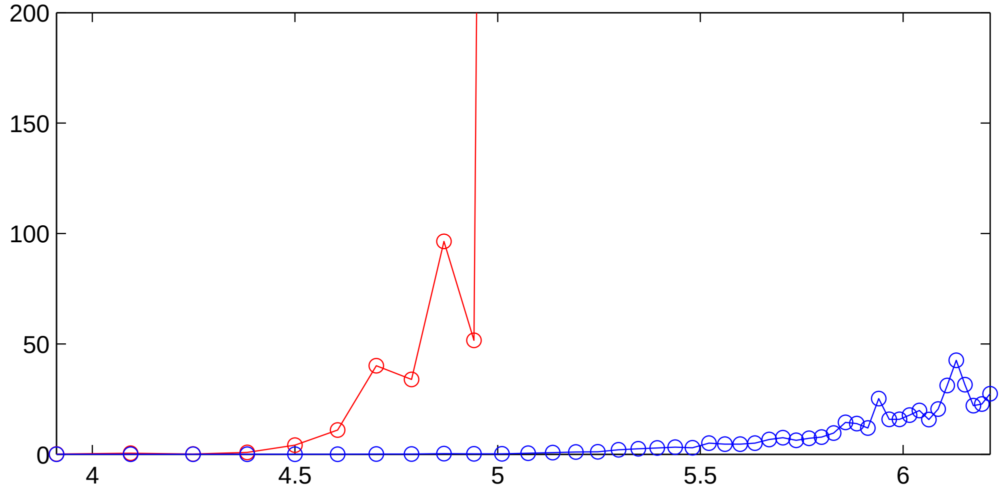


(e) $d_t = 100$, solved in 8.1 seconds

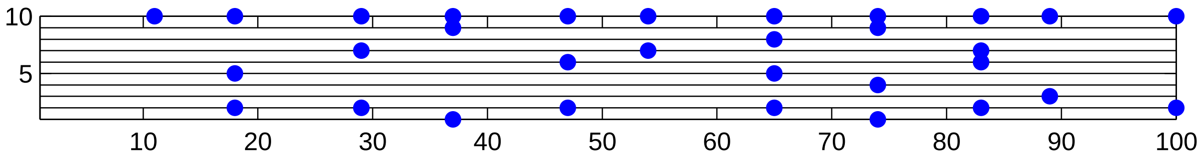
Figur 2: Solutions for different values of d_t



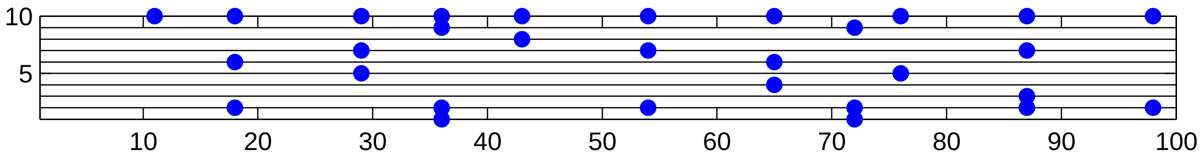
Figur 3: The solution with a modified objective function and $d_t = 25$. Solved in 0.04 seconds.



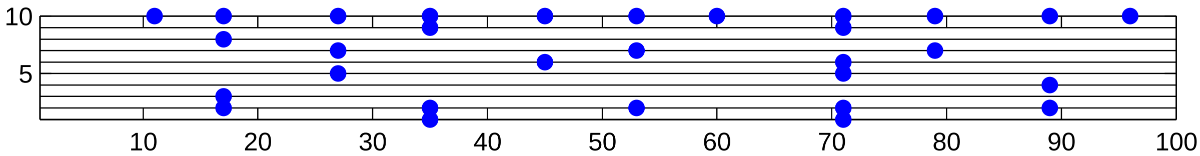
Figur 4: CPU time required to solve the model, with $\ln T$ on the x axis. Red is with integrality requirements, blue is without.



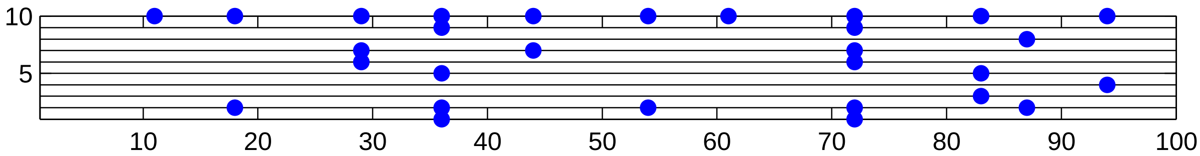
(a) $r = 11$, solved in 28.3 seconds with final cost 670



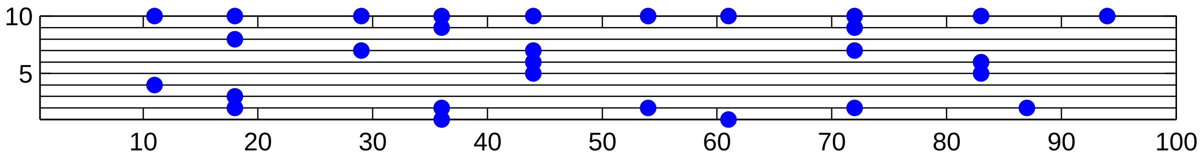
(b) $r = 9$, solved in 23.8 seconds with final cost 660



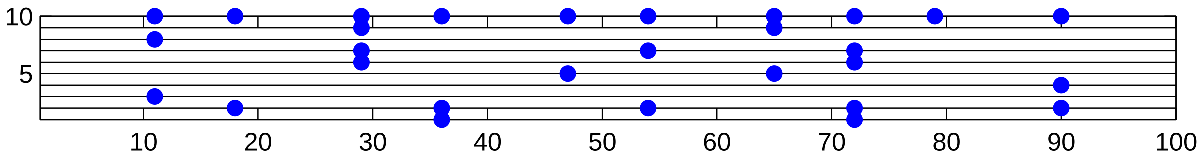
(c) $r = 7$, solved in 13.6 seconds with final cost 645



(d) $r = 5$, solved in 2.6 seconds with final cost 635



(e) $r = 3$, solved in 15.7 seconds with final cost 635



(f) For comparison: $r = 0$, solved in 12.7 seconds with final cost 615

Figur 5: Solutions to the new model with some different values of r