Simulation - Assignment 2

Anton Roth

May 4, 2017

Introduction

All tasks have been solely implemented in *Matlab*. Files *task1.m* and *task2.m* contain the implementation where each sub-task are clearly separated with a lot of #.

Task 1

1.a - Problem formulation

$$\begin{aligned}
\text{Number of packages (in 1000s) of each type: } & x_i,\ i = 1, 2, 3, 4, 5 \\
\text{Maximise: } & Z = 4x_1 + 3x_2 + 2x_3 + 2x_4 + x_5 \\
\text{Demand limitations: } & x_1 \leq 16 \\
& x_3 \leq 2 \\
& x_1 + x_2 + x_3 \leq 34 \\
& x_4 + x_5 \leq 28 \\
\text{Capacity limitations: } & 2x_1 \leq 36 \\
& 2x_2 + 2x_3 + 2x_4 + x_5 \leq 216 \\
& 0.1x_1 + x_2 + 0.5x_4 \leq 18 \\
\text{Lower bound: } & x_i \geq 0
\end{aligned}$$

1.b - Problem solution

The problem formulation was translated into the matrix form used by the function *linprog*. The *dual-simplex* algorithm was chosen. The solution is as follows:

$$\begin{aligned}
& x^T = [16, 14.8, 2, 0, 28] \\
& Z = 140.4
\end{aligned}$$

I.e. UniCam's profit is maximised when 16000 packages of type **I**, 14800 packages of type **II**, 2000 packages of type **III**, 0 packages of type **IV** and 28000 packages of type **V** is sold. The total profit in this case is $1,404,000,000$ SEK.

1.c - Shadow prices

The shadow prices for all constraints, i.e. the profit that can be made with an incrementation of the constraint by one unit, can easily be determined with the Lagrange multipliers given by Matlabs *linprog* function. All shadow prices, denoted S, are presented below in the order as the constraints are introduced in section :

$$S = [3.4, 2, 0, 1, 0, 0, 3]$$

A conclusion that can be made from this is that the by adjusting the first constraint, i.e. $x_1 \leq 16$ (demand of package **I** is at maximum 16 000), the largest possible increase in the optimised profit ($34,000,000$ per thousand units of package **I**) can be made.

1.d - "Allowable increase"

The dependence of the maximum demand of package **III** and the optimised maximal profit is presented in Figure 1.

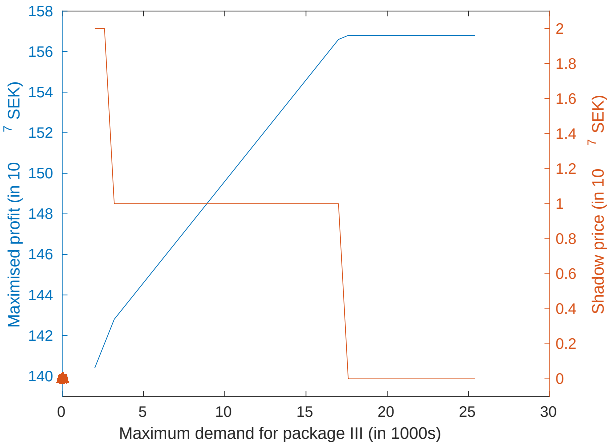

Figure 1: The maximum demand of package **III** is plotted against the optimised maximal profit (left y-axis) in blue. The shadow price for the increased maximum demand of package **III** is presented on the right y-axis in red.

The optimised maximal profit flattens after 26 iterations, i.e. when the maximum demand of the package is 17600. From Figure 1 it is clear that the shadow price changes twice, first from 2 to 1 and then from 1 to 0. Of course, if the shadow price is 0, no increase in the total profit is observed. A conclusion one can draw is that for a demand of package **III** above 17600 the total profit for UniCAM cannot be increased (for the current situation).

1.e - Price variations

The code was implemented after the hint and the results were the following:

$$0.6 \leq \text{Price of package } \mathbf{I} \leq \infty$$

It was concluded that the upper limit must be ∞ since the optimal solution did not change even though the price was huge.

The optimal solution is very robust to changes in the price of package **I** and this is reasonable I think since the profit from this package is the largest with some margin (cf. $4 * 16 = 64$ for package **I** and $1 * 28 = 28$ for package V which has the second largest profit). The reason why the upper limit is ∞ is because the coefficient in the objective function is the largest for parameter 1, which corresponds to the price of package **I**, cause you will always gain the most profit by increasing the price.

Task 2

2.a - Problem solution

The integer problem was solved with the function *intlinprog* and the following was the result:

$$x_1 = 3,\ x_2 = 2 \ \&\ z = 13 \tag{1}$$

2.b - LP and rounding

The relaxed integer problem was solved with *linprog* with the following result:

$$x_1 = 1.6,\ x_2 = 2.6 \ \&\ z = 14.6 \tag{2}$$

The obtained parameter of the relaxed problem was rounded up/down and the paired together in all possible four combinations. The value of the objective function was calculated for each combination and all constraints were checked. The result is presented in Table 1.

Table 1: Value of objective function for each combination of rounded parameter solutions. In the last column it is also noted of which constraint that is violated, if that is the case.

x_1	x_2	z	**Constraint violated**
1	2	11	-
2	2	12	-
1	3	16	first
2	3	17	third

Clear from Table 1 is that the optimal integer solution is not found with rounding the relaxed problem.

2.c - Branch and Bound Method

The result of the branch and bound method is presented with Figure 2.

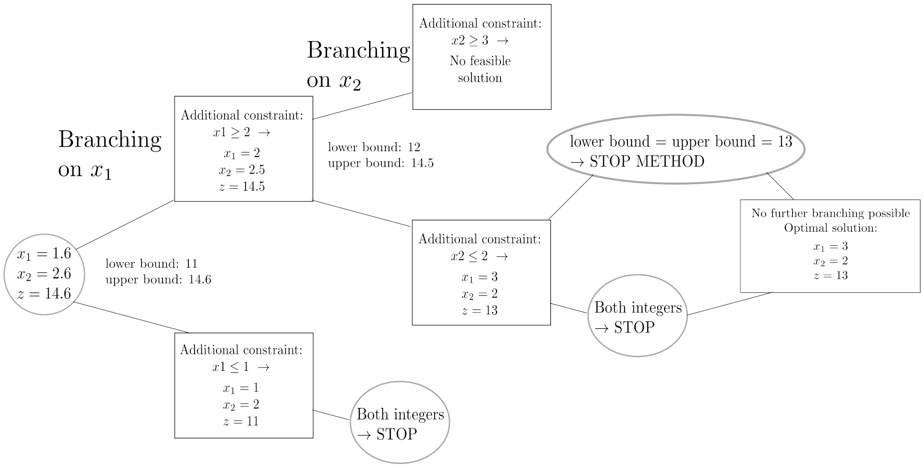

Figure 2: Branch and bound method tree for the relaxed integer problem.

A branch is cut either if there is no feasible solution or both solution parameters are integers. The process is stopped when the upper bound is less than or equal to the upper bound and/or when no further branching is possible (this ocurred simultaneously for the above example).