Discrete Optimisation
Exercise Session 9: Dynamic programming

27th November 2015

Exercise 1 (knapsack)**.** Solve the binary knapsack problem for five objects: the values are $(3, 4, 6, 5, 8)$ and the weights $(412, 507, 714, 671, 920)$, with a total capacity of 1794.

1. Propose a MILP formulation.
2. Propose an algorithm to solve it by dynamic programming.
3. Dualise this problem and solve it by dynamic programming. To this end, with the following definitions ($X \subseteq \mathbb{R}^n$)

$$f(\lambda) = \begin{array}{ll} \max & \mathbf{c}^\mathbf{T}\,\mathbf{x} \\ \text{s.t.} & \mathbf{a}^\mathbf{T}\,\mathbf{x} \leq \lambda, \\ & x \in X \end{array} \qquad h(t) = \begin{array}{ll} \min & \mathbf{a}^\mathbf{T}\,\mathbf{x} \\ \text{s.t.} & \mathbf{c}^\mathbf{T}\,\mathbf{x} \geq t \\ & x \in X \end{array}$$

   prove the two following statements:

   (a) $f(\lambda) \geq t$ if and only if $h(t) \leq \lambda$.

   (b) $f(\lambda) = \max \left\{ t \,\middle|\, h(t) \leq \lambda \right\}$.

Exercise 2 (lot-sizing)**.** A steel factory manufactures spools; the director wants to plan the production for the next four weeks. The demand for spools is exactly known for each week. The cost for using the furnace can be divided into two parts: a fixed cost to get the furnace turned on and a variable cost proportional to the number of spools produced. Storing spools from a time period to another is costly, and is proportional to the number of spools to store.

Week	t	1	2	3	4
Demand [10 spools]	d_t	8	5	13	4
Variable cost [1000€ per 10 spools]	p_t	1	1	1	2
Fixed cost [1000€]	f_t	20	10	45	15
Storage cost [1000€ per 10 spools]	h_t	1	1	1	1

Table 1: Spool factory planning requirements.

1. Propose a MILP formulation.
2. Propose an algorithm to solve it by dynamic programming.

Exercise 3 (time-constrained shortest path)**.** A traveller wants to go from one place to another. However, they must go through customs, whose office has strict opening hours: with the considered time window, only the time at which they start working is important. As such, they want to determine their minimum travel time.

This travel must then be made in a graph $G = (V, E)$, which is annotated with some data: c_{ij} the time to cross edge $e = (i, j) \in E$; r_i, the earliest time to travel through node $i \in V$ (if the traveller gets at i earlier than r_i, they will have to wait until r_i).

1. Propose a MILP formulation.

2. Propose an algorithm to solve it by dynamic programming.

3.