Prof. Friedrich Eisenbrand
Location: MA A3 31

Question session: 25.11.09
Discussion: 2.12.09

Exercises

Optimization Methods in Finance

Fall 2009

Sheet 5

Note: This is just <u>one</u> way, a solution could look like. We do not guarantee correctness. It is your task to find and report mistakes.

Exercise 5.1
Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a polyhedron with $P \neq \emptyset$ and let $\varepsilon > 0$. Show that

$$P_\varepsilon = \{x \in \mathbb{R}^n \mid Ax \leq b + \varepsilon \cdot \mathbf{1}\}$$

is fulldimensional (here $\mathbf{1} = (1, \ldots, 1)$).
Hint: Recall that P_ε is fulldimensional if there exists a $y \in P_\varepsilon$ and a $\delta > 0$, such that the ball of radius δ around y is contained in P_ε, i.e. $\{x \in \mathbb{R}^n \mid \|x - y\| \leq \delta\} \subseteq P_\varepsilon$.

Solution:

Abbreviate $A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ and $a_{\max} := \max_{i,j} |a_{ij}|$. Choose a $y \in P$ (recall that $P \neq \emptyset$) and choose $\delta := \frac{\varepsilon}{n \cdot a_{\max}}$. Let x be any point in $\{x \in \mathbb{R}^n \mid \|x - y\| \leq \delta\}$. Then we can write $x := y + z$ with $\|z\| \leq \delta$ and

$$a_i^T z \leq \sum_{j=1}^{n} \underbrace{|a_{ij}|}_{\leq a_{\max}} \cdot \underbrace{|z_j|}_{\leq \varepsilon/(a_{\max} \cdot n)} \leq \sum_{j=1}^{n} a_{\max} \cdot \frac{\varepsilon}{a_{\max} n} \leq \varepsilon$$

(a_i gives the ith row of A). Consequently

$$Ax = A(y + z) = \underbrace{Ay}_{\leq b} + \underbrace{Az}_{\leq \varepsilon \cdot \mathbf{1}} \leq b + \varepsilon \cdot \mathbf{1}$$

and $x \in P_\varepsilon$.

Exercise 5.2
Suppose we have a fixed budget of $B \in \mathbb{N}$ and we want to invest this money into bonds $i = 1, \ldots, n$. Buying once bond i costs $w_i \in \mathbb{N}$ today and gives us a profit of $p_i \in \mathbb{Q}_+$ next year. Suppose that we can buy an arbitrary (non-negative) <u>integer</u> amount of every bond. We want to make a decision that maximizes the cumulated profit next year (not invested money does not give a profit) using dynamic programming.

i) Define suitable states/table entries and a Bellman equation that will give you an optimum solution.

ii) Compute the concrete table entries for $B = 7$ and the following setting

bond	w_i	p_i
$i = 1$	5	2.6
$i = 2$	2	1.0
$i = 3$	3	1.9

iii) What is (roughly) the running time of your dynamic programming algorithm (for general n)? What would be the running time of a naive algorithm that tries out all combinations? Compare them for say $n = B = 100$ if your computer can process 1 billion operations per second.

Solution:

i) Let $A(i, B')$ be the maximum profit that we can obtain by investing an amount of at most B' into the bonds $1, \ldots, i$. The base entries are $A(1, B') = \lfloor B'/w_1 \rfloor \cdot p_1$. Furthermore

$$A(i, B') = \max_{k=0,\ldots,\lfloor B'/w_i \rfloor} \{A(i-1, B' - k \cdot w_i) + k \cdot p_i\}$$

for $B' = 0, \ldots, B$ and $i = 2, \ldots, n$. Then $A(n, B)$ gives the optimum profit.

ii) The table entries are

$A(i, B')$	$B' = 0$	$B' = 1$	$B' = 2$	$B' = 3$	$B' = 4$	$B' = 5$	$B' = 6$	$B' = 7$
$i = 1$	0	0	0	0	0	2.6	2.6	2.6
$i = 2$	0	0	1	1	2	2.6	3	3.6
$i = 3$	0	0	1	1.9	2	2.9	3.8	3.9

Hence $A(3, 7) = 3.9$ is the optimum profit, which is achieved by taking 2 times bond 2 and once bond 3.

iii) The dynamic program has $O(n \cdot B)$ states. The max runs over at most $B + 1$ values, hence to compute a single state, we need running time $O(B)$. Hence the dynamic program needs in total a running time of $O(nB^2)$. A (rough) upper bound on the number of combinations would be $\prod \frac{B}{p_i} \leq B^n$. For $n = B = 100$ we get roughly $nB^2 = 1.000.000$ operations, which can be processed in 0.001 seconds, where B^n results in $3 \cdot 10^{183}$ years.

Exercise 5.3
Suppose we are given a map of canton Vaud with n many villages. For villages i and j, let $c_{ij} \geq 0$ be the length of the road from i to j (or $c_{ij} = \infty$ if no direct road exists). Consider table entries

$A(i, j, k) =$ length of the shortest path from i to j, where we use at most k many intermediate stations

Give Bellman equations to compute $A(i, j, k)$. How is the base case $A(i, j, 0)$ defined? How can we read the length of the shortest route (using arbitrarily many interstations) from i to j from the table?

Solution:

The bases case is $A(i,j,0) = c_{ij}$ (might be ∞). Then

$$A(i,j,k) = \min_{\ell=1,\ldots,n} \{c_{i\ell} + A(\ell, j, k-1)\}$$

(we assume that $c_{ii} = 0$ for all i). The shortest route from i to j will never visit a village twice, since $c_{ij} \geq 0$. Hence $A(i,j,n)$ gives the length of the best tour (in fact, even $A(i,j,n-2)$ would work).

Exercise 5.4

Compute the value of an American call option on a stock with current price equal to 100 CHF, strike price equal to 102 CHF and expiration date four weeks from now. The yearly volatility of the logarithm of the stock return is $\sigma = 0.30$. The risk-free interest rate is 4%. Use a binomial lattice with $N = 4$.

Solution:

Using the formulas from the lecture (or the book of Cornuejols et al) one has

$$\Delta = \frac{1}{52}, \quad u = e^{\sigma\sqrt{\Delta}} = 1.0425, \quad d = e^{-\sigma\sqrt{\Delta}} = 0.9592,$$
$$R = 1 + \frac{0.04}{52} = 1.000769, \quad p_u = \frac{R-d}{u-d} = 0.4988, \quad p_d = \frac{u-R}{u-d} = 0.5012$$

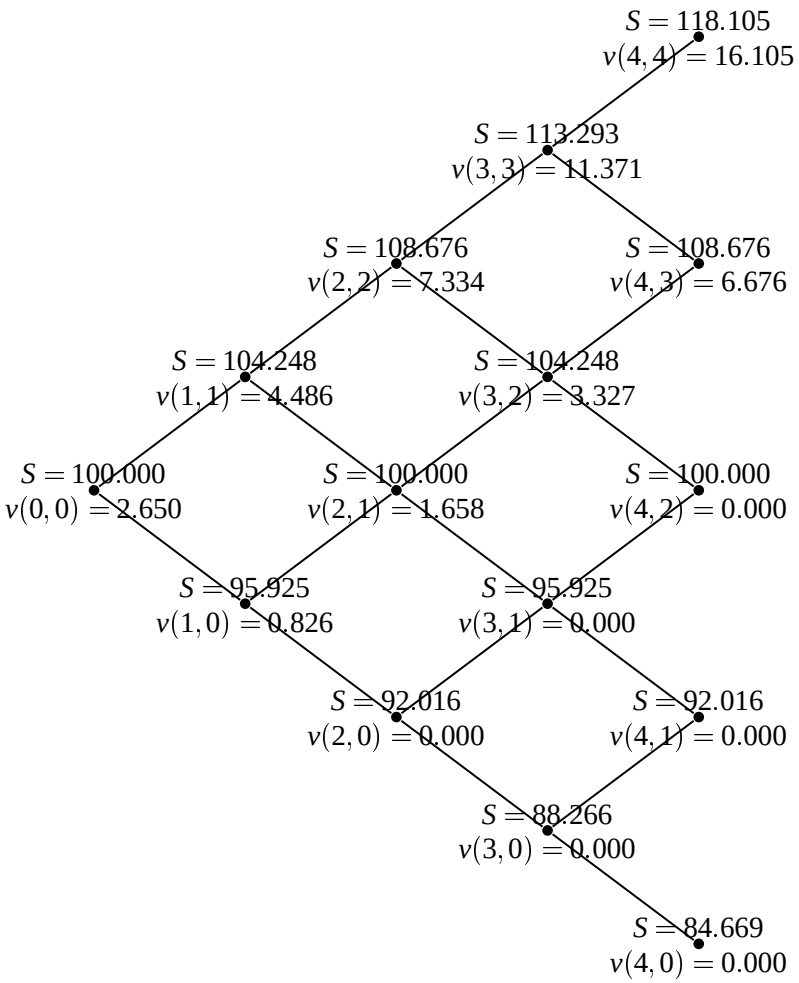
(strictly spoken some of this formulas are just approximating the true values). We compute $v(k,j)$ as the value of the option at time k if the underlying stock increased exactly j times. The formulas used in the dynamic program are then

$$v(n,j) = \max\{u^j d^{n-j} S_0 - c, 0\}$$

(since we are considering a call option) and

$$v(k,j) = \max\left\{\frac{1}{R}(p_u v(k+1, j+1) + p_d v(k+1, j)), u^j d^{k-j} S_0 - c\right\}$$

(since it is an American option) for $k = 0, \ldots, n-1$. The binomial lattice looks as follows



(S gives the price of the underlying asset w.r.t. time k and number of increases j). Thus the value of the option (at time 0) is $v(0,0) = 2.65$.

A side remark: Repeating the procedure with a larger binomial lattice of size $n = 1000$ gives a more precise price of 2.57.
