MSE2114 - Investment Science Lecturer Notes II

Fernando Dias

September 20, 2023

Contents

1 Introduction

In this course, we have focused on many definitions and calculations (among them **inflation**, **interest rates**, **present value**, **future value**, annuities, bonds, securities, forward rates, short rates and spot rates) and those define the concepts of investment science. For now, we are looking into the "should" and "should not" of this course. Using different interest rates, we can define the **weighted average cost of capital** (or **WACC**) based on the debt payments for bondholders and costs with equity.

2 Capital budgeting

In a general context, a **budget** represents any form of limitations of resources. For example, when an investor is willing to purchase a portfolio of bonds, there is a direct limitation on the amount of units due to their value and the amount the investor will pay.

Assuming a set of projects where we assume the following **three characteristics**: each project is treated as the decision, either the project is fully attempted or not attempted at all, there is no option to half it out. This scenario is not the most case for securities where a small divisible amount can be purchased; each project is independent, which means that there is no cross-correlation between benefits and costs for different projects; each project is not ready to be traded in markets.

Considering the **task** of allocating the capital C among m investment project, where b_i is the benefit of each project i, c_i is the cost of each project i.

Simultaneously, an established capital (C) for all these projects can be treated as a depleting constraint (where the capital is applied to its total amount even though the return is small or negligible). In practice, the **optimisation** is limited by several ends to fund all these projects, such as credit limitations from banks and higher interest rates associated with heavier credit burdens. At the same time, changes in the **optimisation** C, such as extension, can also happen.

Let each project be denoted by $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \{0,1\}^m$, such that:

$$x_i = \begin{cases} 1, & \text{if project } i \text{ is funded} \\ 0, & \text{otherwise} \end{cases}$$

The optimum can be solved from following the **optimisation** problem:

$$\begin{aligned} \max_{\mathbf{x}} \sum_{i=1}^{m} b_i x_i \\ \text{subject to} \sum_{i=1}^{m} c_i x_i \leq C \\ x_i \in \{0,1\}, \quad i = 1, 2, \dots, n \end{aligned}$$

This can be reduced to a simple knapsack problem (a classic problem in combinatorial/integer **optimisation**).

There are, however, some challenges to finding optimal solutions using such formulation. The **dimensionality** of the problem can generate numerical difficulties (hard-to-solve instances with $m = 50...100$), and multiple optimal solutions can be found for a single problem. Against all these odds, many instances with $m = 10\ 000$ can be solved in milliseconds with state-of-the-art optimization methods, and small and easy instances can even be solved with Excel Solver.

Alternatively, an **approximate solution** can be generated by using benefit-to-cost ratios $r_i = b_i/c_i$, where the goal is to fund projects one by one in decreasing order of ratios r_i (i.e., starting from the project whose ratio is highest) until the **budget** C has been depleted.

Some projects might have dependencies (multiple variants of the same project that allow choices between variants), and such problems can be modelled as follows.

Let x_{ij} be decision to fund variant j of project i and b_{ij} and c_{ij} be the benefit and cost of variant $j = 1, \dots, n_i$ of project i.

$$\max_{\mathbf{x}} \sum_{i=1}^{m} \sum_{j=1}^{n_i} b_{ij} x_{ij}$$

$$\text{subject to} \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \leq C$$

$$\sum_{j=1}^{n_i} x_{ij} \leq 1, \quad i = 1, 2, \ldots, m$$

$$x_{ij} \in \{0, 1\}, \quad j = 1, 2, \ldots, n_i, i = 1, 2, \ldots, m$$

Further improvements in the formulations also allow for variants such as **enable** in which if project j cannot be started unless the enabling project i is implemented, the following constraint captures such behaviour:

$$x_j \leq x_i \tag{1}$$

This can be pushed further to impose if a project or a subset of projects is selected, the following:

$$x_{i_1} + x_{i_2} + \cdots + x_{i_\ell} \begin{cases} \geq k \\ \leq k \\ = k \end{cases}$$

The further the extensions go, the more complicated and computationally challenging a problem requires better solvers. Small and easy problems can be solved with Excel Solver, while more challenging ones can be solved with CPLEX, Gurobi, Matlab, and others.

3 Portfolio Optimisation

In this type of problem, the goal is to build a portfolio of securities traded in the markets. Therefore, the price dynamics of securities have to be taken into account, and, in some occasions, this problem can be solved much in the same way as **capital budgeting problems**.

Consider the following scenario: a task where a bond portfolio is built considering that the cash flow meets or exceeds these liabilities (obligation to pay y_i euro in period $i = 1, 2, \ldots, n$) and has the smallest purchasing price.

$$\min_{\mathbf{x}} \sum_{j=1}^{m} p_j x_j$$

$$\text{subject to} \sum_{j=1}^{m} c_{ij} x_j \geq y_i, \quad i = 1, 2, \ldots, n$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, m$$

Where:

- c_{ij} = cash flow of bond j in period i,
- p_j = price of bond j.

The above formulation assumes that there is no short selling or issuing new bonds (which forces $x_j \geq 0$); otherwise, the non-negativity constraint must be eliminated. Also, no reinvestments are present if any excess cash flows are available. However, reinvestments could be modelled, e.g. by adding artificial bonds with cash flows $(0, 0, \ldots, 0, -1, 1 + r_i, 0, \ldots, 0)$ where r_i is the short rate for period i. Finally, fractional (non-integer) bond purchases can happen (which might not be the most acceptable) but are easily avoided by introducing integer constraints $x_j \in \{0, 1, 2, \ldots\}$.

4 Dynamic Programming

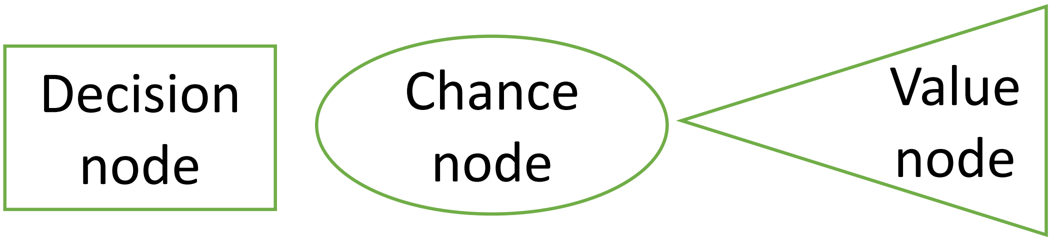
Dynamic programming is a computer programming technique where an algorithmic problem is first broken down into sub-problems, the results are saved, and then the sub-problems are optimized to find the overall solution.

In finance, dynamic problems involve decisions and the information that supports them depending on previous decisions. For example, oil well as an investment:

- Oil wells can be drilled only if the site has been acquired
- Information about profitability can be obtained through testing
- How much is the test worth?
- Based on the test results, should one buy or not?

Note: Many dynamic programming problems involve risk and uncertainties resolved over time. However, the examples in this section of Luenberger do not consider the partial resolution of uncertainties (Bayesian updating).

Generally, dynamic decisions can be structured as decision trees or lattices, which can be used to model dynamic choices and dynamic uncertainties (e.g., stock price evolution over time). For example:



Consider the following scenario. A dynamic investment problem such that:

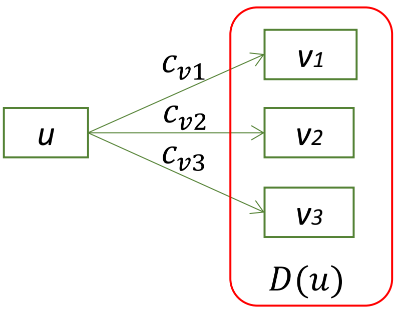
- Initial investment cost is 100 €
- The investment yields cash flows of 300 € or 0 € with the same probability 0.50
- With a cost of 10 €, an expert can be consulted to assess whether the investment is profitable (300 €) or not (0 €)
- The expert knows <u>for sure</u> the status of the well, which is thus the only uncertainty (otherwise, a larger decision tree based on Bayesian updating would be built.)

A variant of this problem is dynamic choice models, which help solve complex decision trees and simple lattices by avoiding traversing all impossible paths with large n. The solution principle is as follows (the basic principle of dynamic programming):

- What is an optimal decision at node u?
- Decision leads to node $v \in D(u)$
- Cash flow of c_v
- Optimal cash flow

$$V(u) = \max_{v \in D(u)} \{c_v + V(v)\}$$

- Solve recursively from the end to the beginning



In the calculation of the optimal cash flow, the discount can be introduced as:

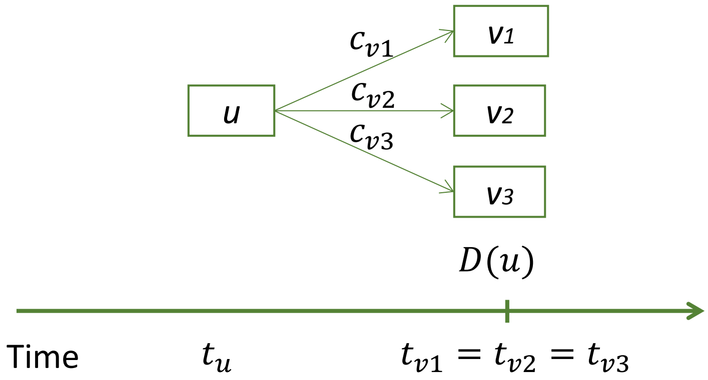
$$V(u) = \max_{v \in D(u)} \{c_v + d_{t_u,t_v} V(v)\}$$

where $d_{t_u,t_v} = 1/(1 + f_{t_u,t_v})^{t_v - t_u}$, the discount factor is based upon:

- f_{t_u,t_v} = forward rate from time t_u to time t_v

- t_u, t_v times of decisions u, v

Considering decisions in every period, $f_{k,k+1}$ can be seen as a short rate r_k.



5 Harmony theorem

The harmony theorem represents what are the strategies to make an investment attractive. Using IRR may lead to different decisions than NPV, which can be misleading.

Imagine a new patent where different options for commercialization are available. For each commercialization option i, the present value is:

$$P(i) = -c_i + \frac{1}{1+r} b_i$$

where b_i, c_i are benefit and cost of option i, respectively.
In order to maximize the present value, the best selection is the alternative i^* such that:

$$P(i^*) = \max_{i \in \{1,2,\ldots,n\}} P(i)$$

At first sight, choosing the alternative to cover the expenses leads to misleading solutions:

$$\max_{i \in \{1,2,\ldots,n\}} \left\{ \frac{b_i}{c_i} \right\}$$

However, to get the best share of benefits, the best option is to buy share α of the patent at a cost $\alpha P(i^*)$. Thus, the cost-benefit decision criterion is:

$$\max_{i \in \{1,2,\ldots,n\}} \left\{ \frac{\alpha b_i}{\alpha P(i^*) + \alpha c_i} \right\}$$

Hence, the following theorem can be established.
Theorem: The cost-benefit decision criterion is maximized for $i = i^*$, meaning that all parties' preferred options coincide.

Proof: Suppose there is an alternative i such that:

$$\frac{\alpha b_i}{\alpha P(i^*) + \alpha c_i} > \frac{\alpha b_{i^*}}{\alpha P(i^*) + \alpha c_{i^*}} = 1 + r.$$

Now, solving this for the definition of $P(i)$ we get:

$$-c_i + \frac{b_i}{1+r} > P(i^*) \Leftrightarrow P(i) > P(i^*),$$

Which contradicts the definition of i^*. 😎

Theorem 5.1. Harmony theorem

Current venture owners should want to operate the venture to maximize the present value of its cash flow stream. Potential new owners, who must pay the full value of their prospective share of the venture, will want the company to operate in the same way to maximize the return on their investment.

6 Firm valuation

In firms, their valuation is characterized by their cash flows, which consider different aspects such as dividends to stockholders, net earnings of the firm, and cash flow that could be realized by selling the firm's assets. In addition, no volatilities are included.

Considering dividend cash flows D_k in year $k = 1, 2, \ldots$, their respective present values are given as:

$$V_0 = \frac{D_1}{1+r} + \frac{D_2}{(1+r)^2} + \cdots + \frac{D_k}{(1+r)^k} + \cdots$$

In the **constant-growth dividend model** dividends grow at a constant rate g which leads to the **Gordon formula**:

$$V_0 = \frac{D_1}{1+r} + \frac{(1+g)D_1}{(1+r)^2} + \cdots + \frac{(1+g)^{k-1}D_1}{(1+r)^k} + \cdots$$
$$\Rightarrow V_0 = \frac{D_1}{r-g} = \frac{1+g}{r-g}D_0, \quad r > g,$$

which is obtained from the geometric sum $\sum_{i=0}^{\infty} at^i = \frac{a}{1-t}$ for any real a and $|t| < 1$ (here, $a = \frac{D_1}{1+r}$ and $t = \frac{1+g}{1+r}$).

The discounted growth under the same circumstance is given as:

$$V_0 = \frac{1+g}{r-g}D_0, \quad r > g,$$

Startups and very small companies pay little or no dividends in order to retain more capital for growth. In this scenario, the possible analysis is based on the **free cash flow** that the firm can pay without compromising growth.

Considering a firm that wishes to determine how large a share of its cash flow it should invest in its capital (e.g., machines) to maximize the present value of its dividends / free cash flow. Under this condition, the profit in year n is Y_n, out of which the share $u \in [0, 1]$ is invested. The annual growth rate modelled as factor $g(u)$, which leads to profit to be defined as:

$$Y_{n+1} = [1 + g(u)]\, Y_n$$
$$\Rightarrow Y_n = [1 + g(u)]^n Y_0$$

As the capital depreciates by a factor α annually but increases by the investment, it can be modelled as:

$$C_{n+1} = (1-\alpha)C_n + uY_n$$

Using recursion, the capital in year $n > 0$ is:

$$C_n = (1-\alpha)^n C_0 + uY_0 \sum_{i=1}^{n} (1-\alpha)^{n-i} \left(1 + g(u)\right)^{i-1}$$

Using mathematical identity:

Income statement	
Before-tax cash flow	Y_n
Depreciation	αC_n
Taxable income	$Y_n - \alpha C_n$
Taxes (34%)	$0.34(Y_n - \alpha C_n)$
After-tax income	$0.66(Y_n - \alpha C_n)$
After-tax income + depreciation	$0.66(Y_n - \alpha C_n) + \alpha C_n$
Sustaining investment	uY_n
Free cash flow	$0.66(Y_n - \alpha C_n) + \alpha C_n - uY_n$

$$x^n - y^n = (x-y)\sum_{i=1}^{n} x^{n-i}y^{i-1}$$

with $x = 1-\alpha$ and $y = 1+g(u)$, we get:

$$C_n = (1-\alpha)^n C_0 + uY_0 \frac{(1+g(u))^n - (1-\alpha)^n}{g(u)+\alpha}$$

Note that $x - y = 1 - \alpha - (1+g(u)) = -\alpha - g(u)) < 0$, hence the fraction $\frac{x^n - y^n}{x-y}$ is written as $\frac{y^n - x^n}{y-x}$
By combining the terms containing $(1-\alpha)^n$, this formula becomes:

$$C_n = (1-\alpha)^n \left(C_0 - \frac{uY_0}{g(u)+\alpha}\right) + uY_0 \frac{(1+g(u))^n}{g(u)+\alpha}$$

Due to depreciation $\alpha > 0$, the first term with $(1-\alpha)^n$ will tend to zero over time, and thus we get:

$$C_n \approx uY_0 \frac{(1+g(u))^n}{g(u)+\alpha} = \frac{u}{g(u)+\alpha} Y_n$$

For other combinations of parameters, the numerical solution can be found, e.g. with Excel.
Notes: From the last equation on slide 41, the sustaining investment $C_{n+1} - C_n = uY_n - \alpha C_n$ needs to be subtracted from the after-tax income to obtain the free cash flow
By using $C_n = \dfrac{u}{g(u)+\alpha} Y_n$, we obtain an analytic formula for the free cash flow as a function of period n:

$$FCF_n = 0.66(Y_n - \alpha C_n) + \alpha C_n - uY_n$$
$$\Rightarrow FCF_n = \left[0.66 + 0.34\frac{\alpha u}{g(u)+\alpha} - u\right](1+g(u))^n Y_0,$$

Because this cash flow grows by a constant factor, we can use Gordon's formula to calculate its present value:

$$PV = \left[0.66 + 0.34\frac{\alpha u}{g(u)+\alpha} - u\right] \frac{1}{r - g(u)} Y_0$$