Unit 8

Mathematical programming: modeling optimization problems with constraints

ESIGELEC

Instructor: Federico Perea

1

Contents

1. Introduction
2. Linear programming
3. Integer programming
4. Non-linear programming. An application: portfolio selection (Markowitz model)

2

1. Introduction

- Mathematical programming: objective function under constraints.
- Example, find the rectangle with the maximum area so that the short side is lower than or equal to 2m, and the total perimeter is lower than or equal to 12m.

$$\begin{array}{ll} max & xy \\ s.t.: & x \leq 2 \\ & 2x + 2y \leq 12 \\ & x, y \geq 0 \end{array}$$

3

2. LINEAR PROGRAMMING

2.1 INTRODUCTION
2.2 AN EXAMPLE
2.3 SENSITIVITY ANALYSIS

4

2.1 Introduction

- Linear objective function and linear constraints → *linear programming (LP).*
- LP is the most used technique of OR
- It was first conceived by George B. Dantzig around 1947.
- L.V. Kantorovich worked on similar problems around 1939, although his work remained unknown until 1959.
- The most used algorithm to solve LP problems is the Simplex method.

2.2 An example

- A company produces three different products, P1, P2 and P3, which can be made out of three different resources R1, R2 and R3.
- Every unit of each product needs certain units of every resource. All units are considered divisible infinitely. These quantities as well as the unitary benefit that every unit of each product can give in the market are:

	P1 needs	P2 needs	P3 needs	Availability
R1	2	9	3.5	430
R2	6	4	9	410
R3	8	9	7	570
Benefit	2.5	5	4	--

The question

- The manager wants to answer a simple question:
- What is the maximum benefit that can be obtained, and how much of each product should we produce?

THE MODEL, VARIABLES, OBJECTIVE FUNCTION AND CONSTRAINTS

VARIABLES: how can we answer the question?

- Units of product 1 to be produced: X_1
- Units of product 2 to be produced: X_2
- Units of product 3 to be produced: X_3

1^{ST} HYPOTHESIS OF LINEAR PROGRAMMING

DIVISIBILITY

All variables can take any real value

2^{ND} HYPOTHESIS OF LINEAR PROGRAMMING

NON-NEGATIVITY

All variables are non-negative

OBJECTIVE FUNCTION

Good	1	2	3
Units	X_1	X_2	X_3
Benefit per unit	2.5	5	4

- We want to MAXIMIZE the benefit:

  MAXIMIZE $Z = 2.5 \cdot X_1 + 5 \cdot X_2 + 4 \cdot X_3$ is the objective

3^{RD} HYPOTHESIS OF LINEAR PROGRAMMING

LINEARITY

All relations between variables are linear, this implies:

1. **Contributions are proportional. The contribution of each variable to the OF is proportional to its value and constant to all the range values**
2. **Contributions are additive. All the contributions added up is equal to the addition of all the individual contributions whichever the variables values**

11

Constraints 1

- We can't spend more than 430 units of resource 1.

For each unit of	P1 needs	P2 needs	P3 needs	Availability
R1	2	9	3.5	430

- Therefore, $2 \cdot X_1 + 9 \cdot X_2 + 3.5 \cdot X_3 \leq 430$

12

Constraints 2 and 3

- 410 of resource 2 and 570 of resource 3

For each unit of	P1 needs	P2 needs	P3 needs	Availability
R2	6	4	9	410
R3	8	9	7	570

- Therefore,
  $6 \cdot X_1 + 4 \cdot X_2 + 9 \cdot X_3 \leq 410$
  $8 \cdot X_1 + 9 \cdot X_2 + 7 \cdot X_3 \leq 570$

- COMPLETE FORMULATION OF THE PROBLEM
- Determine the values of the variables:

$$X_1 \geq 0 \text{ , } X_2 \geq 0 \text{ and } X_3 \geq 0$$

- In order to optimize

$$\text{MAX } Z = 2.5 \cdot X_1 + 5 \cdot X_2 + 4 \cdot X_3$$

- Subject to the constraints:

$$2 \cdot X_1 + 9 \cdot X_2 + 3.5 \cdot X_3 \leq 430 \text{ (R1)}$$

$$6 \cdot X_1 + 4 \cdot X_2 + 9 \cdot X_3 \leq 410 \text{ (R2)}$$

$$8 \cdot X_1 + 9 \cdot X_2 + 7 \cdot X_3 \leq 570 \text{ (R3)}$$

We’ll solve it with Excel (install the add-in “Solver”).

Microsoft Excel 16.0 Answer Report

Worksheet: [Class1PresentationExamples.xls]LP Example

Report Created: 23/09/2018 21:51:47
Result: Solver found a solution. All Constraints and optimality conditions are satisfied.
Solver Engine

Engine: Simplex LP

Solution Time: 0.016 Seconds.

Iterations: 2 Subproblems: 0
Solver Options

Max Time 100 sec, Iterations 100, Precision 0,000001

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 0%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
B12	Benefit Units P1	299,3283582	299,3283582

Variable Cells

Cell	Name	Original Value	Final Value	Integer
B10	Units P1	0	0	Contin
C10	Units P2	36,34328358	36,34328358	Contin
D10	Units P3	29,40298507	29,40298507	Contin

GENERAL FORMULATION OF A LINEAR PROGRAM

- Determine the values of the variables:

$$X_j, \text{ where } j=1,\ldots,n$$

- In order to optimize (maximize or minimize)

$$\text{MAX/MIN } \Sigma\, C_j \cdot X_j$$

- Satisfying the constraints:

$$\Sigma\, a_{ij} \cdot X_j \ \leq, = \text{ or } \geq b_i \quad \text{for } i=1,2,\ldots,m$$

- n variables
- m constraints
- c_j are the objective function coefficients
- a_{ij} are called technical coefficients
- b_i are called resource coefficients

4th HYPOTHESIS OF LINEAR PROGRAMMING

CERTAINTY

All model coefficients (c_j, a_{ij}, b_i) are known and deterministic

17

- When solving a linear program we can have four possible outcomes:
  - Unique Solution
  - Infinite solutions
  - No solution
  - Unbounded solution

18

2.3 SENSITIVITY ANALYSIS

OBJECTIVE FUNCTION COEFFICIENT SENSITIVITY ANALYSIS

- The slope of the objective function determines which point in the feasible region would be the optimum solution
- The slope of the objective function depends on its coefficients (c_j)
- What happens if we change those coefficients?
  MAX $Z = 2.5 \cdot X_1 + 5 \cdot X_2 + 4 \cdot X_3$
- The sensitivity report will help us answer these questions.

SENSITIVITY REPORT WITH EXCEL

Microsoft Excel 16.0 Sensitivity Report

Worksheet: [Class1PresentationExamples.xls]LP Example

Report Created: 23/09/2018 21:50:16

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B10	Units P1	0	-0,02238806	2,5	0,02238806	1E+30
C10	Units P2	36,34328358	0	5	0,5	3,222222222
D10	Units P3	29,40298507	0	4	7,25	0,032608696

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
B14	Units of R1 Units P1	430	0,432835821	430	46,88679245	270,5555556
B15	Units of R2 Units P1	410	0,276119403	410	78,88888889	218,8888889
B16	Units of R3 Units P1	532,9104478	0	570	1E+30	37,08955224

- Reduced Cost:

  The required amount that a specific objective function coefficient (c_i) has to be improved for the associated variable to be greater than cero in the optimum solution

  A symmetric definition could be *the deterioration in the objective function value per unit of a variable that we force in the solution*

  - The allowable increase/decrease is the maximum amount we can add/reduce a objective function coefficient so that the optimal values of the variables do not change (the objective function value does change because the coefficients change, though)

Example

- *What happens if the benefit per unit of P1 increases 0.01 units?* Nothing, because its reduced cost is 0.02238806, and therefore the value of its corresponding variable will remain zero.
- *What if we increase it 0.03?* We will produce some units of P1 in the optimal solution.
- *What if we force that one unit of P1, with the current unitary benefit, should be produced?* The o.f. value will deteriorate by 0.02238806 units.
- *Without solving the problem again, what is the solution if the benefit per unit of P2 was 5.3?* Same amount of each product, but the benefit changes. *If it was 6?* Then we have to solve the problem again changing this coefficient.

RESOURCE COEFFICIENT SENSITIVITY ANALYSIS

- What happens if we change a b_i?
- Example: we can buy ten extra units of R1.
- The new constraint would be

  $2 \cdot X_1 + 9 \cdot X_2 + 3.5 \cdot X_3 \leq \mathbf{440}$ (R1)
- How much extra benefit would they get for these 10 extra units?
  The answer is given by the shadow price of this constraint (see the sensitivity report)
  Extra benefit = 10·0.432835821 = 4.32835821
- How about having extra units of R2 and R3?

23

Allowable increase/decrease

- If we change the right hand sides (RHS) of constraints within the allowable ranges, the objective function changes proportionally to the shadow prices.
- Out of these ranges, we don't know what will happen!

24

- Dual Price or Shadow Price:

  Improvement in the optimum value of the objective per unit of resource coefficient increase (b_i)

  – Therefore, in a maximization problem, if we increase the resource coefficient value, the new objective function value would be:

  $$\text{New } Z = Z + \Delta\, b_i \cdot (\text{Dual Price constraint } i)$$

- In redundant (not binding) constraints the shadow price is 0
- If the slack of a constraint is strictly greater than cero (not binding constraint) the shadow price is always 0, if the slack is 0 (binding constraint) the shadow price might be non-negative.
- Binding constraints are also called *bottlenecks*.

SUMMARY

- Linear programming allows us to solve optimization problems with one linear objective function and one or more linear constraints.
- In order to do so we need to define variables, objective function and constraints
- Every combination of variable values that satisfies all constraints is called a **feasible solution**. All possible feasible solutions form the **feasible region**
- A feasible solution with best objective function value is called an **optimal solution**
- We have learned how to solve them and analyze solutions by means of a computer software

3. INTRODUCTION TO INTEGER PROGRAMMING

3.1 INTRODUCTION
3.2 A SIMPLE INTEGER PROBLEM TO DISTRUST ROUNDING UP
3.3 SOME INTEGER PROGRAMMING APPLICATIONS
3.4 FOUR CLASSIC MODELS

3.1 INTRODUCTION

- When the divisibility hypothesis is not acceptable we revert to Integer Programming
- The mathematical model in integer programming is the same as in linear programming but with the additional constraint that the values that the variables take have to be integer
- Pure Integer Programming: All variables are integer
- Mixed Integer Programming: One or more variables are integer (but not all)
- Binary Programming: All the model variables are of the type zero-one
- Integer programming is much harder than linear programming

3.2 A SIMPLE INTEGER PROBLEM TO DISTRUST ROUNDING UP

- A research center has received 250 thousand € to purchase computer equipment. Various studies indicate that there are only two different types of machines suitable for the task and that any combination of the two types would be acceptable. Tests have been conducted in order to evaluate the load capacity of the machine types in "average tasks per hour" units
- The objective of the center is to maximize the potential work capacity
- It is clear that the number of purchased equipment has to be an integer quantity

29

- X_1 = number of units of type 1 equipment that are purchased
- X_2 = number of units of type 2 equipment that are purchased

EQUIPMENT	COST (THOUSAND €)	CAPACITY (TASKS/HOUR)
1	140	28
2	60	11

- The integer linear programming model would be:

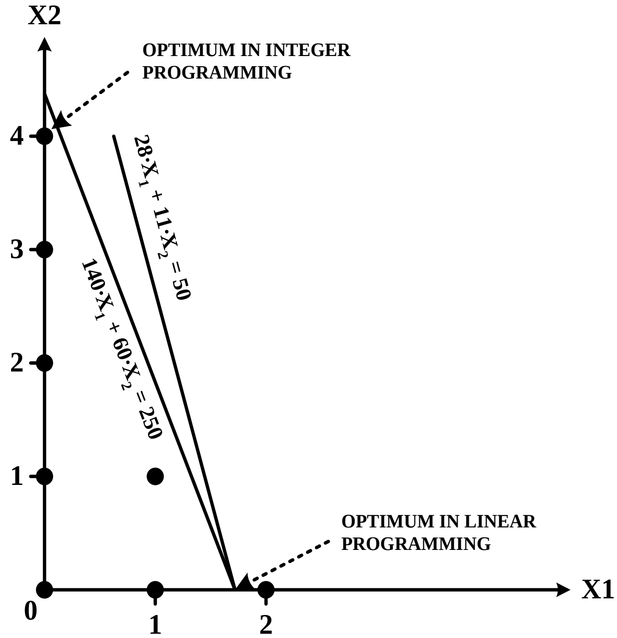
$$\text{Max } Z = 28 \cdot X_1 + 11 \cdot X_2$$

$$140 \cdot X_1 + 60 \cdot X_2 \leq 250 \text{ (monetary resources)}$$

$$X_1, X_2 \geq 0 \text{ (non negativity)}$$

$$X_1, X_2 \text{ integers}$$

30



31

- Optimal linear solution : $X_1 = 1.79$, $X_2 = 0$, $Z = 50$.
- Rounding up: $X_1 = 2$, $X_2 = 0$, $Z = 56$ → Not feasible
- Rounding down: $X_1 = 1$, $X_2 = 0$, $Z = 28$ → Not optimal
- The optimal integer solution is $X_1 = 0$, $X_2 = 4$, $Z = 44$.
- Going from the optimal linear solution to the optimal integer solution causes a decrease of a 12%, but the rounding up leads to a loss of a 44%. It is not advisable to round up when the variables take small values
- Even if we took $X_1 = 1$, and spent the rest of the budget (90 thousand €) on a type 2 machine, Z would be 28 + 11 = 39 (lower than the integer optimal solution).

32

Example

- Solve the linear production example with which we introduced linear programming, assuming the units of products are not infinitely divisible (for instance, product 1 are tables, product 2 chairs, and product 3 closets).
- Is it logical that the new optimal objective function value is lower than the continuous case? Why?
- Give a solution by rounding up the continuous solution, and explain the differences between this solution and the integer optimal one.

33

3.3 SOME INTEGER PROGRAMMING APPLICATIONS

- If the variables take high values we can round up to the closest integer and get a (nearly) optimal solution. Generally this will not be the case in problems like the previous one
- In some problems we have to make decisions of the type yes or no. Shall we invest?, Shall we build here?, Shall we start the project or not? These situations are modeled with **binary variables** (variables of the zero-one type)
- Binary variables are also used as “fictitious” variables to formulate complex problems in an easy way, i.e. we can formulate logic propositions like linear restrictions (separable programming)

34

A PROBLEM OF INVESTMENT PLANNING

- We work for a company that manufactures cars and we need to plan the optimal distribution of our budget. The available investments are:
  1. Updating our factory (increase in production of 10 thousand units)
  2. Building another factory (increase of 15 thousand units)
  3. Worker stimulation program, to enhance productivity at work (5 thousand units)
  4. Purchase of new and more technologically advanced manufacturing equipment (2 thousand units)

35

- Investments 1 and 2 are mutually exclusive and investment 4 can be done only if investment 1 is done
- The company wants to increase production capacity, but due to market demands, the increase should not be greater than 16000 units
- The problem consists of choosing from the available investments those that maximize the Net Present Value (NPV) knowing that the available budget for the next two years is 150 and 120 m.u.

Investment	NPV	Expenses	
		Year 1	Year 2
1	50	60	70
2	80	100	70
3	40	30	40
4	30	20	50

36

- Variables:

  X_j is 1 if investment j is done and 0 otherwise, for j = 1,2,3,4

- Objective Function: Maximize the NPV of the possible investments:

$$\text{Max } 50\cdot X_1 + 80\cdot X_2 + 40\cdot X_3 + 30\cdot X_4$$

- Constraints
  - Investments can not go over budget

    $60\cdot X_1 + 100\cdot X_2 + 30\cdot X_3 + 20\cdot X_4 \leq 150$ (year 1)

    $70\cdot X_1 + 70\cdot X_2 + 40\cdot X_3 + 50\cdot X_4 \leq 120$ (year 2)

  - Investments 1 and 2 are mutually exclusive

    $X_1 + X_2 \leq 1$

37

  - Investment 4 is conditioned to investment 1

    $X_4 \leq X_1$ or equivalently $X_4 - X_1 \leq 0$

  - The production increase may not be higher than the maximum sales forecast

    $10\cdot X_1 + 15\cdot X_2 + 5\cdot X_3 + 2\cdot X_4 \leq 16$

Let's solve it with Excel!

38

A PROBLEM WITH FIXED COSTS

- A company can manufacture 4 different types of products in a production line that goes through 3 three different departments. Production takes labor hours, gives benefits, and has associated fixed costs according to the following table:

Product	Men-hours per thousand product units			Gross profit (€/ud)	Fixed cost (thousand €)
	Dep 1	Dep 2	Dep 3		
P1	160	120	50	80	200
P2	150	200	50	85	200
P3	100	180	50	98	90
P4	200	175	50	100	150
Man-hour per month availability	4000	4800	1600		

- If a specific kind of product is produced the company incurs in an overhead or fixed cost of refurbishing the production line
- The company whishes to program the production so benefits are maximized
- The fixed costs can not be modeled with linear programming
- Variables

  $X_j \geq 0$ for j = 1,2,3,4 will be the size of product j

  Y_j (0,1) will be 1 when $X_j > 0$ and zero when $X_j = 0$

- Objective function:

  Max $80 \cdot X_1 + 85 \cdot X_2 + 98 \cdot X_3 + 100 \cdot X_4 - 1000(200 \cdot Y_1 + 200 \cdot Y_2 + 90 \cdot Y_3 + 150 \cdot Y_4)$

- Constraints:
  - Man-hour availability per departments:

    $(160 \cdot X_1 + 150 \cdot X_2 + 100 \cdot X_3 + 200 \cdot X_4)/1000 \leq 4000$ (dep 1)

    $(120 \cdot X_1 + 200 \cdot X_2 + 180 \cdot X_3 + 175 \cdot X_4)/1000 \leq 4800$ (dep 2)

    $(50 \cdot X_1 + 50 \cdot X_2 + 50 \cdot X_3 + 50 \cdot X_4)/1000 \leq 1600$ (dep 3)

  - Guarantee that the fixed cost is taken into account only when product is produced:

    $X_1 \leq M_1 \cdot Y_1$ $\quad$ $X_2 \leq M_2 \cdot Y_2$

    $X_3 \leq M_3 \cdot Y_3$ $\quad$ $X_4 \leq M_4 \cdot Y_4$

    M's are large enough numbers. Which are the minimum values they can take keeping the model meaningful? Have a look at: http://orinanobworld.blogspot.com/2011/07/perils-of-big-m.html

41

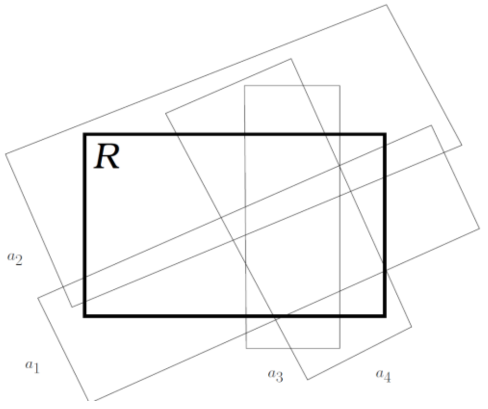
2.4 FOUR CLASSIC MODELS

- Set covering problem
- Assignment problem
- Shortest path problem
- Industry location problem

42

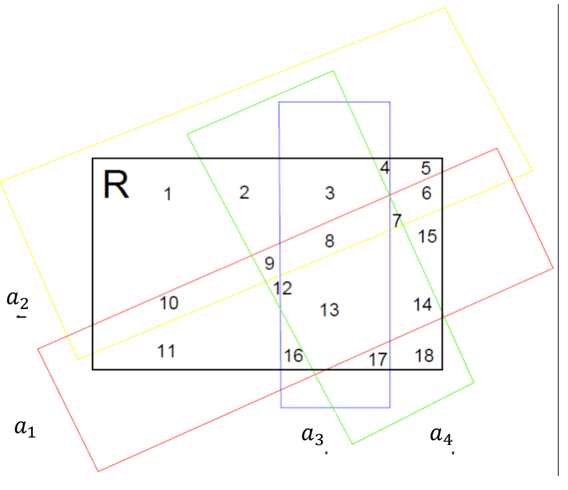
Set covering problem

- It consists of finding subsets covering a given region of interest at minimum cost.
- Given the region of interest R, and n sets C={a_1, a_2,..., a_n}, so that their union covers R, find a subset of C that covers R at minimum cost.



43

- From the sets in C, the region of interest R is divided into m different subregions, which result from intersections between the sets in C.



44

FORMULATION

- Let $q_{ij} = 1$ if set a_j covers subregion R_i , zero otherwise.
- Let c_j be the cost incurred when using set a_j.

$$\min \sum c_j x_j$$
$$s.t.: \sum_j q_{ij} x_j \geq 1, i = 1, \dots, m$$
$$x_j \in \{0,1\}$$

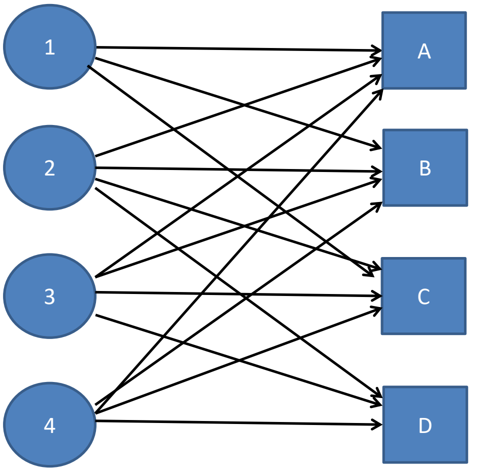
- $x_j = 1$ if set a_j is used, zero otherwise

45

Assignment problem

- Assume that *m* workers can do *m* tasks.
- All tasks must be done, and each worker may not do more than one task.
- If worker *i* does task *j*, the company obtains a benefit of c_{ij} m.u.
- The company wants to find out the assignment of workers to tasks so that the overall benefit is maximized.
- c_{ij} can be considered as a cost (for instance time). In such a case the objective function should be minimized.

46



Graphical example of an assignment problem.

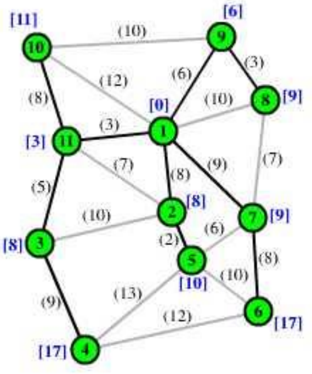
$$x_{ij} = 1 \; if \; worker \; i \; does \; task \; j$$

$$\begin{aligned} \min \quad & \sum_{i,j} c_{ij} x_{ij} \\ s.t.: \quad & \sum_j x_{ij} = 1, \qquad i = 1, \dots, m \\ & \sum_i x_{ij} = 1, \qquad j = 1, \dots, n \\ & x_{ij} \in \{0,1\} \end{aligned}$$

Shortest path problem

- Assume a set of *n* points joined by arcs forming a connected graph.
- The distance between the two points joined by arc (*i,j*) is d_{ij}
- The problem consists of finding the shortest path from point *1* (called the *source*) to point *n* (called the *sink*).
- This type of problems is solved by GPS navigators, webs where you find the shortest route from one point to another, etc

49



Instance of a shortest path problem. Source: http://www.me.utexas.edu/~jensen/ORMM/methods/unit/network/subunits/mst_spt/index.html

50

The problem can be formulated from the binary variables
$x_{ij} = 1$ if the shortest path uses arc (i,j), zero otherwise.

$$\min \quad \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} x_{ij}$$

$$s.t.: \quad \sum_{j=1}^{m} x_{1j} = 1 \qquad \text{(source node)}$$

$$\sum_{i=1}^{m} x_{im} = 1 \qquad \text{(sink node)}$$

$$\sum_{j=1}^{m} x_{kj} - \sum_{i=1}^{m} x_{ik} = 0, \ \forall\ k = 2,\ldots,m-1 \qquad \text{(intermediate nodes)}$$

$$x_{ij} \in \{0,1\}.$$

There are more efficient formulations that use arc formulations, duality techniques, etc. Besides, some algorithms, like Dijkstra's algorithm, work very well for this problem.

51

AN INDUSTRY LOCATION PROBLEM

- It is an important application of mixed integer programming in which one whishes to determine the location and optimal size of a series of factories that produce goods of great demand. It is an extension of the *Transportation Problem.*
- Product demands and location of the clients are known in advance
  - $b_1, b_2, \ldots, b_n$: Known demands of the n clients
  - $a_1, a_2, \ldots, a_m$: Production capacities of each of the m factories that can be built
  - $f_1, f_2, \ldots, f_m$: construction cost of each of the m factories
  - c_{ij} : Cost of transporting one unit of product from the i^{th} factory to the j^{th} client

52

AN INTEGER LINEAR PROGRAMMING MODEL

- Variables:

  x_{ij} : Number of units transported from factory i to client j, $i = 1, 2, \dots, m, j = 1, 2, \dots, n$

  y_i : Binary variables, take value 1 if factory i is constructed and 0 otherwise, $i = 1, 2, \dots, m$

- Objective function: $\min \sum_i (f_i y_i + \sum_j c_{ij} x_{ij})$
- Constraints
  - Clients demands: $\sum_i x_{ij} \geq b_j$, $j = 1, \dots, n$.
  - Production at factories: $\sum_j x_{ij} \leq a_i$, $i = 1, \dots, m$.

Many other constraints may arise: incompatibility between two factories, non-linear costs, limited arc capacities, …

4. NON-LINEAR PROGRAMMING

4.1 INTRODUCTION
4.2 AN EXAMPLE OF NON-LINEAR MODELS
4.3 PORTFOLIO SELECTION

4.1 INTRODUCTION

- Linear programs are most of the time approximations of reality
- For example, in production planning, when minimizing variable costs, we usually consider constant unitary costs regardless of production quantities
  - What about economies of scale?
  - What about increasing marginal costs?
- Integer programming allows modeling some non-linearities but non-linear programming might be needed

55

- Non-linear programming
  - Non-Linear objective function AND/OR Non-Linear constraints (at least one)
- No best known method for non-linear programming (no non-linear simplex or non-linear branch and bound)
- Some problems are really non-linear. Therefore, non-linear programming is necessary
- Non-linear programming is MUCH more complex than both linear continuous programming and integer programming
- Now the simplex method cannot be applied because:
  - The feasible region might not be convex
  - The optimal solution might not be located on extreme points of the feasible region, not even on its border!

Available algorithms guarantee local optimality, not global optimality as Simplex does with linear programming.

56

4.2 AN EXAMPLE OF NON-LINEAR MODELS

- A company manufactures three products
- The sales volume depends on the price
- We have the following relationships between monthly sales in thousands of units (X_j) and unitary prices (P_j) for the first two products:

X1=10-P1, X2=16-P2

- For the third product, the sales volume also depends on product 2's price (sales dependency)

X3=6-0.5P3+0.25P2

- Variable costs for each product are 6, 7 and 10 €/units, respectively

- Production is limited by available machine time and worker time
- Each month we have 1000 hours/machine and 2000 hours/man available

Resource	Product		
	P1	P2	P3
Machine hours/unit	0.4	0.2	0.1
Worker hours/unit	0.2	0.4	0.1

- Gross margin is income minus variable cost
- Product 1:
  - Income: I1= P1·X1. Since X1=10-P1 we have that P1=10-X1 and I1=(10-X1)*X1 = 10X1-X1^2
  - Variable cost of P1 is C1=6X1
  - Then, benefit for P1 is B1=I1-C1 = 10X1-X1^2-6X1= -X1^2+4X1
- Similarly, for product 2: B2=-X2^2+9X2
- For product 3
  - I3=P3·X3 -> 20X3-2X3^2-1/2X3·X2
  - B3=-2X3^2+10X3-1/2X3·X2

59

- We want to maximize the total benefit:

MAX -X1^2+4X1-X2^2+9X2-2X3^2+10X3-1/2X3·X2

- Constraints:

4X1+2X2+X3<=10

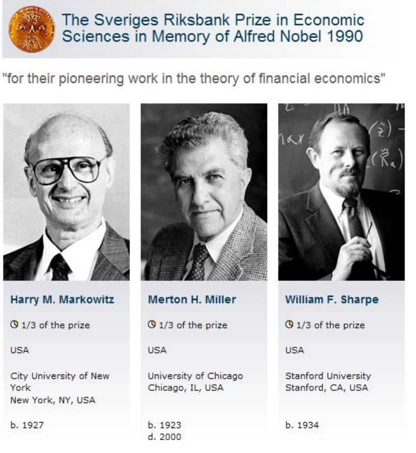
2X1+4X2+X3<=20

Let's solve it with Excel!

60

4.3 PORTFOLIO SELECTION. SOME MODELS

- Portfolio selection: One of the possible applications of non-linear programming
- Economics… financial economics. Nobel prize in 1990 (Markowitz & Sharpe)



61

- MARKOWITZ MODEL
  - MARKOWITZ, H. (1952): "Portfolio selection". Journal of Finance, vol. 7, n.º 1, march, pp. 77-91.
  - MARKOWITZ, H. (1959): Portfolio selection: Efficient diversification of investments. John Wiley & Sons. New York.
  - First approach to portfolio selection
  - Parametric model with quadratic objective function
  - It obtains the optimal portfolio selection

62

- PROBLEM DEFINITION AND OBJECTIVES
  - The portfolio is formed by a mixed set of assets
  - The variables represent the percentage of each asset in the portfolio
  - The investor wants to maximize profitability or return
  - The investor wants to minimize risk
  - Model devised to give the minimum possible risk for each given minimum return
  - The profitability of the portfolio is a random variable that can be characterized by its mean and variance

63

- Consider a portfolio with the possible assets $A_1, A_2, \ldots, A_n$
- Each variable $x_1, x_2, \ldots, x_n$ is the percentage of each asset in the portfolio (they sum up to 1)
- $r_1, r_2, \ldots, r_n$ are the expected returns of each asset
- The total expected return is then:

$$r = \sum_{i=1}^{n} r_i x_i$$

64

- How to estimate the risk:
  - Variability in the profitability or return ≈ risk
  - Diversification in the portfolio leads to less risk (do not invest in the same sector if you want a low risk)
  - If we study the variations in the profitability of the different assets (variance) we can assess the risk
  - The less related (less correlated) two possible assets are, the lower the associated risk when combining them becomes
  - Covariance as the measure of correlation between any possible pair of assets

65

- Covariance

$$Cov(A_1, A_2) = S_{1,2} = \frac{1}{n-1}\sum_j (r_{1,j} - \bar{r}_1)(r_{2,j} - \bar{r}_2)$$

r_{ij} is the return of asset i on period j.

$\bar{r}_i$ is the average return of asset i.

- If both assets' returns increase (or decrease) at the same time: positive covariance
- If one assets' return increases and the other decreases: negative covariance

66

- Furthermore, the covariance of the same asset is the variance:

$$Cov(A_1, A_1) = s_{A_1, A_1} = S_{A_1}^2$$

- Summing up: the total risk is the fluctuation (variance) of each asset and the covariance (correlated risk) of that asset against all other assets in the portfolio:

$$Risk = \sum_{i=1}^{n} \sum_{j=1}^{n} S_{i,j} \cdot x_i \cdot x_j$$

67

- The resulting model is then to minimize risk

$$MIN = \sum_{i=1}^{n} \sum_{j=1}^{n} S_{i,j} \cdot x_i \cdot x_j$$

- Subject to:

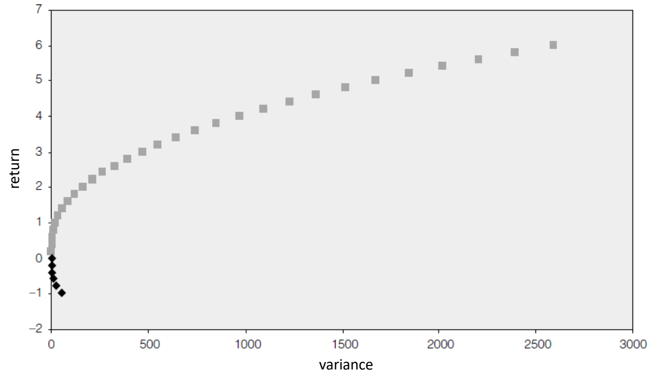
$$\sum_{i=1}^{n} r_i x_i \geq \text{Minimum return}$$

$$\sum_{i=1}^{n} x_i = 1$$

$$x_i \geq 0, \forall i = 1, \ldots, n$$

68

- For a given minimum return, the model gives the minimum risk portfolio selection



An example

Profit per Year and Asset	Asset A	Asset B	Asset C
2012	9.51	40	-3
2013	17.38	29	14
2014	19.33	20	18
2015	23.47	31	14
2016	43.43	2	12

Calculate the less risky portfolios for investors who want to ensure a profit of 18%, 20%, 22%, 24% and 26%.

SUMMARY

- Real life is non-linear in many cases
- Non-linear programming is more complex than linear programming
- There isn't a *best* method for non-linear problems
- Existing methods can only guarantee local optimum solutions
- Use only as a last option or only if the problem you have is non-linear and is solved effectively and efficiently by a well known non-linear method
- Portfolio selection is one clear example of multiobjective and non-linear programming application.

71

Heuristics and Metaheuristics

- Some problems are so difficult that exact algorithms like mathematical programming are not able to solve them
- In these cases ther are other possibilities
- Heuristics and Metaheuristics are algorithms that provide a solution 'fast', althouth such solution is not guaranteed to be optimal
- In this short introduction we will see Greedy, GRASP, and Genetic Algorithms.

72

Greedy

- An iterative process to solve optimization problems
- At each iteration, choose locally optimal steps
- The objective is to find a global optimum
- This does not always happen
- They are fast (good!)
- Sometimes they give the *worst* solution!

Example of a greedy algorithm

- Consider the assignment example we saw in class

	Task 1	Task 2	Task 3	Task 4
Worker 1	13	4	7	6
Worker 2	1	11	5	4
Worker 3	6	7	2	8
Worker 4	1.5	3	5	9

- Greedy solution: (W2,T1),(W3,T3), (W4,T2), (W1,T4). Value: 1+2+3+6=12.
- Optimal solution: (W1,T2), (W2,T4), (W3,T3), (W4, T1). Value: 11.5

GRASP

- The Greedy algorithm only provides one solution, which might be far from optimal
- The Greedy gets stuck in local optima
- One option to overcome this problem is to randomize some parts of it
- This randomized algorithm is called GRASP: “Greedy randomized adaptive search procedure“

GRASP: constructive phase

- At each step, a *restricted candidate list* (RCL) of elements is built.
- The size of the RCL is a parameter of the algorithm (typically a small number).
- Solutions are generated by adding elements to the problem's solution set from the RCL, chosen randomly.
- This procedure is repeated a number of times, in order to obtain a variety of solutions.

Example GRASP (size RCL = 2)

	Task 1	Task 2	Task 3	Task 4
Worker 1	13	4	7	6
Worker 2	1	11	5	4
Worker 3	6	7	2	8
Worker 4	1.5	3	5	9

- First iteration: RCL = {(W3,T3),(W4,T1)}. Randomly choose one: (W4,T1).
- Second iteration: {(W3,T3), (W1,T2), (W2,T4)} Randomly choose one: (W1,T2).
- Third iteration: {(W3,T3), (W2,T4)} Randomly choose one: (W2,T4)
- Fourth iteration: choose the one left: (W3,T3)

This solution coincides with the optimal one!

GRASP: local search

- The constructive phase is typically completed by a local search phase
- In this phase we look around the solution found (called neighbours), with the hope of finding a better solution there (in the neighbourhood)
- You need to define what a neighbour solution is, and check if any of these neighbours improves the current solution.

Genetic algorithm

- After repeating the GRASP a number of times, you have a population of solutions.
- A genetic algorithm tries to improve this population of solutions by means of operators that mimic the evolution of species.
  - Crossover: choose two solutions from the population, and “mix” them, obtaining tho offsprings.
  - Mutation: choose one element of the population and mutate it, by changing one of its elements.

79

Other general heuristics and metaheuristics

- Simulated annealing
- Threshold accepting
- Tabu search
- Genetic algorithm
- Ant colonies
- Local search
- ….

A very extensive list!

80

End of Unit

BIBLIOGRAPHY

- M.S. Bazaraa, J.J. Jarvis and H.D. Sherali. *Linear programming and network flows*. Wiley (1990).
- S. Luke. Essentials of metaheuristics : a set of undergraduate lecture notes. 2nd ed. Editorial Lulu, 2013.