Máster Universitario Oficial en Ciencia de Datos e Ingeniería de Computadores

Técnicas de Soft Computing para Aprendizaje y Optimización

Redes Neuronales y Metaheurísticas, Programación Evolutiva y Bioinspirada

David A. Pelta Mochcovsky

Grupo de Trabajo en Modelos de Decisión y Optimización

Datos de Contacto

- E-mail: *dpelta@ugr.es*
- Página web: *https://wpd.ugr.es/~dpelta/wordpress/*
- Despacho Nº 21, 4ta planta ETSI Informática

Contenidos

1. Conceptos Básicos de Optimización y Búsqueda
2. Heurísticas y Metaheurísticas basadas en trayectorias

Contexto y Definiciones Preliminares

Veamos algunos conceptos básicos.

Soft Computing

"… es una mezcla de distintos métodos que de una forma u otra cooperan desde sus fundamentos. En este sentido, el principal objetivo de la Soft Computing es aprovechar la tolerancia que conllevan la imprecisión y la incertidumbre, para conseguir manejabilidad, robustez y soluciones de bajo costo..."

"…Se trata de considerarla como antítesis de lo que podríamos denominar "Hard Computing", de manera que podría verse la Soft Computing como un conjunto de técnicas y métodos que permitan tratar las situaciones practicas reales de la misma forma que suelen hacerlo los seres humanos, es decir, en base a inteligencia, sentido común, consideración de analogías, aproximaciones, etc."

Soft Computing and Fuzzy Logic

LOTFI A. ZADEH, University of California at Berkeley

Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost. Its principal constituents are fuzzy logic, neurocomputing, and probabilistic reasoning. Soft computing is likely to play an increasingly important role in many application areas, including software engineering. The role model for soft computing is the human mind.

One of the deepest traditions in science is that of according respectability to what is quantitative, precise, rigorous, and categorically true. It is a fact, however, that we live in a world that is pervasively imprecise, uncertain, and hard to be categorical about. It is also a fact that precision and certainty carry a cost. Driven by our quest for respectability, we tend to close our eyes to these facts and thereby lose sight of the steep price we must pay for high precision and low uncertainty. Another visible concomitant of the quest for respectability is that in much of the scientific literature elegance takes precedence over relevance.

A case in point is the traveling salesman problem, which is frequently used as a testbed for assessing the effectiveness of various methods of solution. What is striking about this problem is the steep rise in computing time as a function of precision of solution. As the data in Table 1 show, lowering the accuracy to 1.50 percent reduces the computing time by an order of magnitude for a ten-fold increase in the number of cities.

A more familiar example that illustrates the point is the problem of parking a car. We find it relatively easy to park a car because the final position of the car is not specified precisely. If it were, the difficulty of parking would increase geometrically with the increase in precision, and eventually parking would become impossible.

Guiding principle. These and many similar examples lead to the basic premises and the guiding principle of soft computing.

The basic premises of soft computing are

- The real world is pervasively imprecise and uncertain.

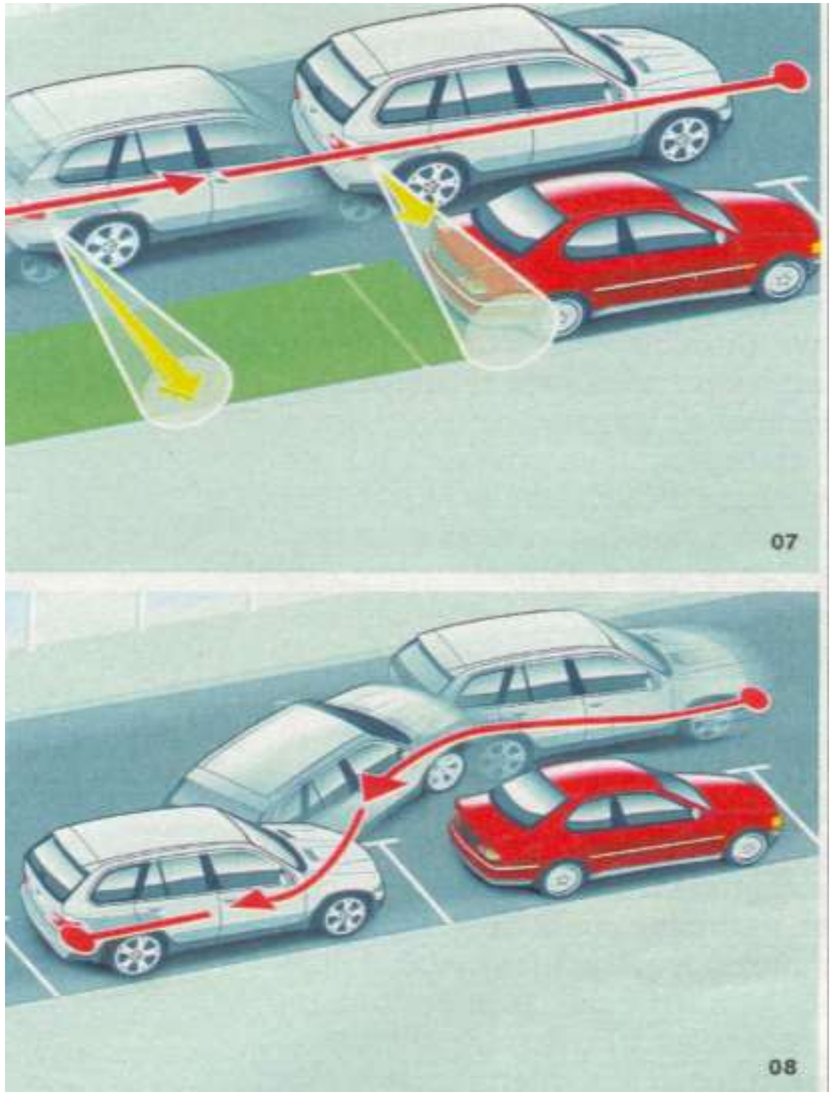
48 NOVEMBER 1994

Zadeh, l. IEEE Software 11, 6, pags 48-56. 1994

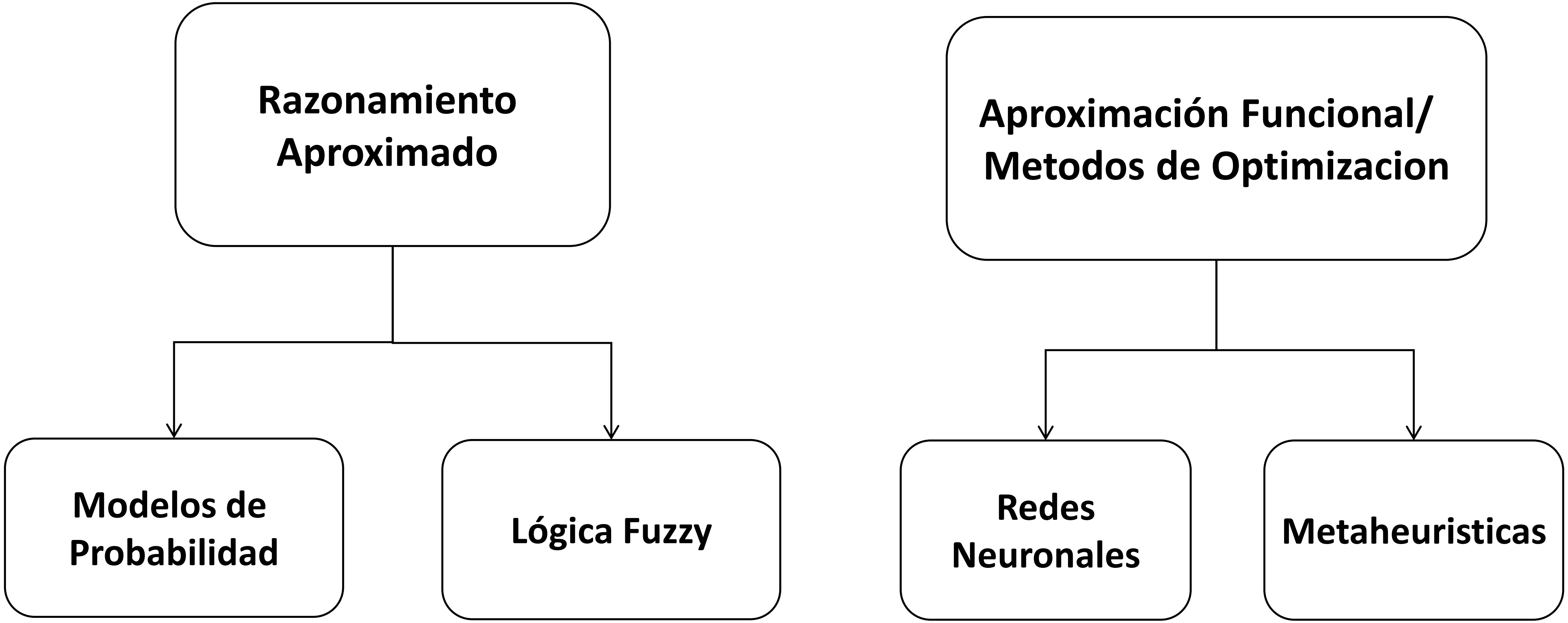
El tren debe pasar por el centro de la ciudad

Aparcar un Coche

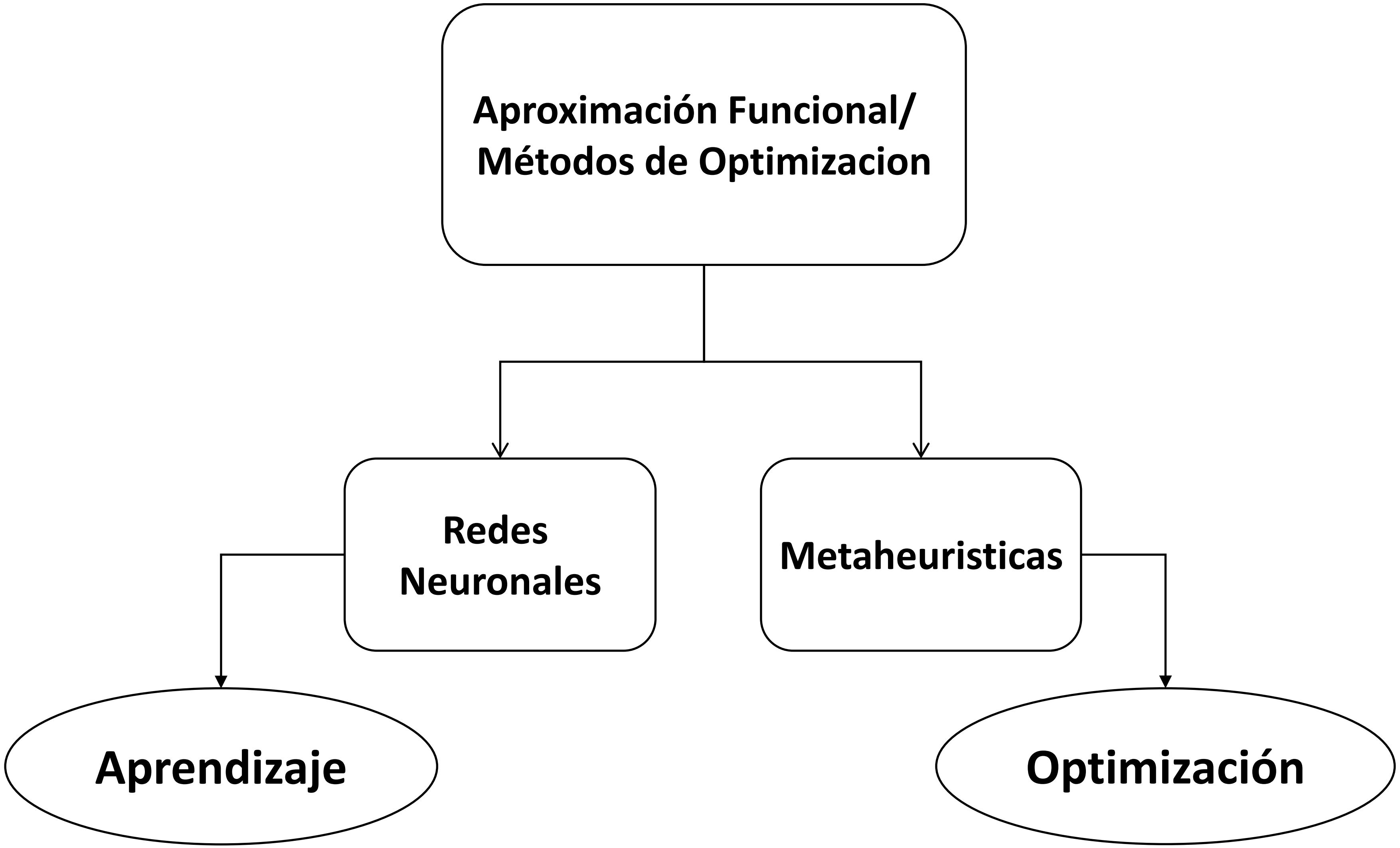
- Aparcar un coche es sencillo puesto que la posición final que ha de ocupar el coche, no está fijada de antemano con exactitud.
- Aparcar con exactitud, requeriría movimientos en función de unidades de fracciones de milímetros, la tarea nos llevaría horas o días enteros, con un número de maniobras difícil de calcular.
- Se ve que una alta precisión conlleva un alto costo, incluso para tareas que entendemos como sencillas.
- El desafío supone aprovechar la tolerancia de la imprecisión proponiendo métodos de cálculo que faciliten soluciones aceptables a bajo costo (Principio básico de la Soft Computing).



Componentes de la Soft Computing



Componentes de la Soft Computing



optimizar Conjugar

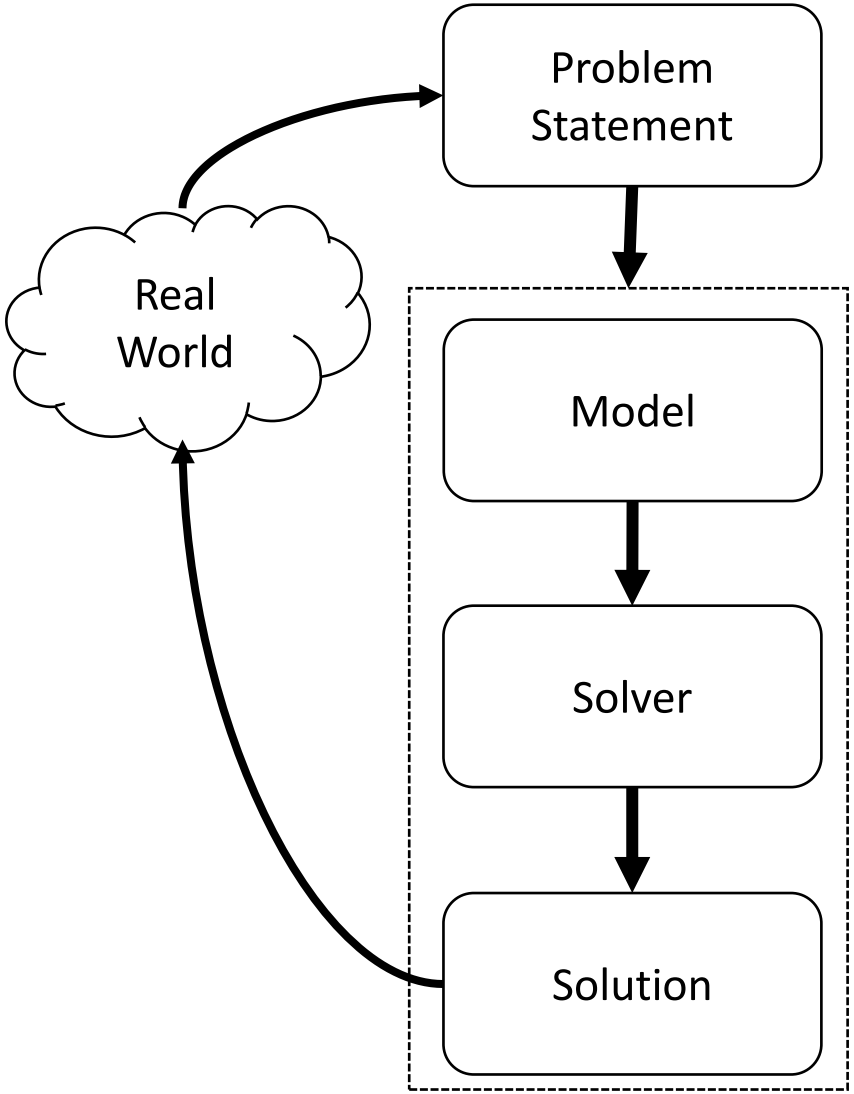
De *óptimo* e *-izar*.

1. tr. Buscar la mejor manera de realizar una actividad.

Real Academia Española ©

- Camino más corto entre dos puntos.
- Asignación de trabajadores a tareas.
- Diseño de redes de comunicación.
- Asignación de tareas a procesadores.
- Localización de antenas de telefonía móvil.
- Determinar ubicación de paradas de taxis.
- Corte de patrones en telas/chapas metálicas reduciendo el desperdicio.
- Encontrar los parámetros para que un modelo proporcione el mejor ajuste.

A Solving Approach

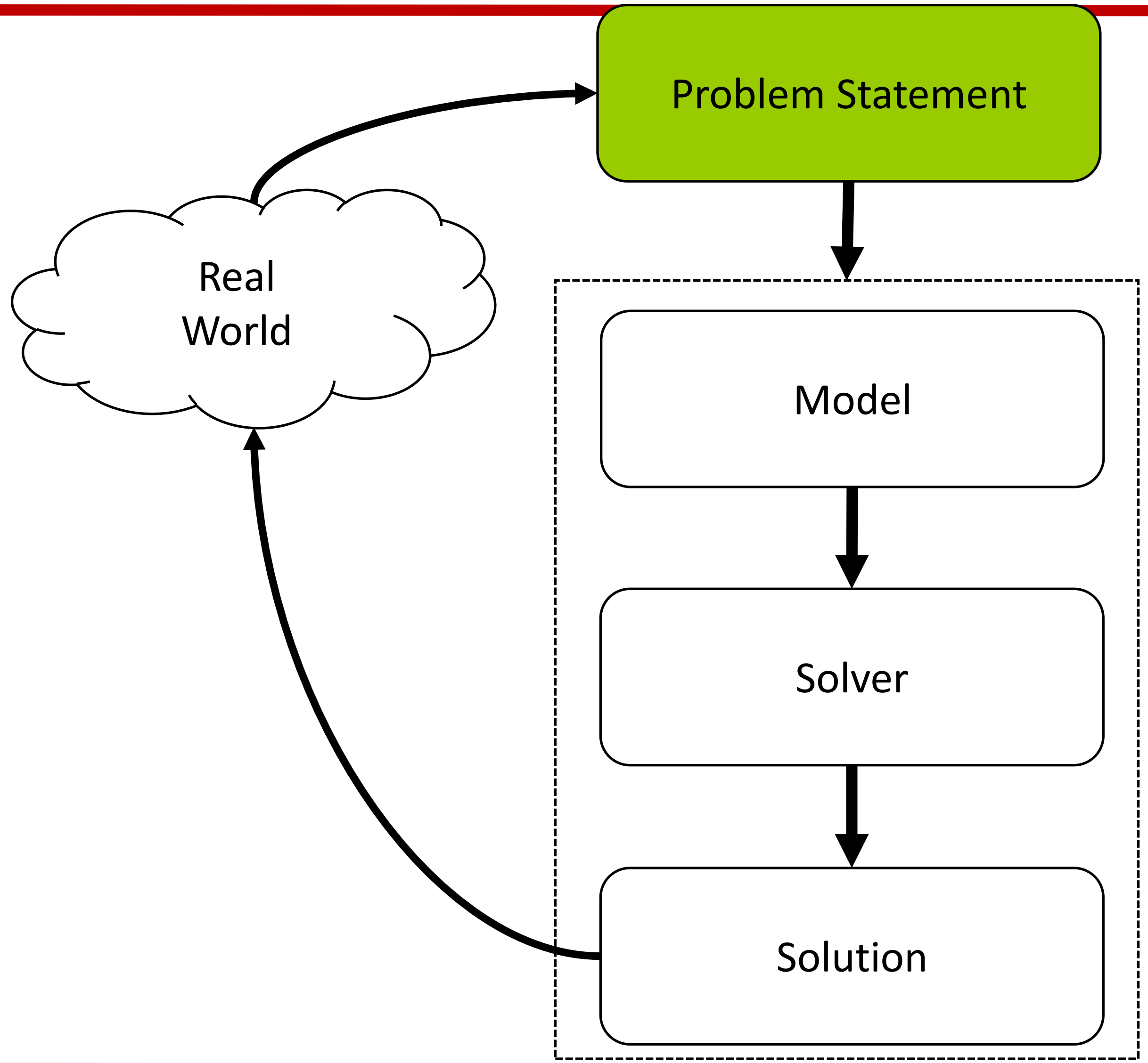


- A problem can be modelled in several ways.
- As every model has certain features, then an appropiated solver should be chosen.
- Sometimes the same solver can be used to solve different models.

We want the “best” solution!

How we can find such best? (the optimum)

Let’s see first some problem statements


Problem Statement
Real
World
Model
Solver
Solution

Example 1: Statement

A farmer needs to transport his/her 400 hives.

A transport company has 8 trucks with 40 hives' capacity and 10 with 50 hives' capacity. The Company has 9 drivers available.

Renting the small truck cost 600 €, while the bigger one costs 800 €.

The farmer wants to know the number of trucks of each type she/he should rent to make the transport as cheapest as posible.

Example 2: Statement

A small brewery produces ale and pilsen.

Production limited by scarce resources: corn, hop, barley malt.

Recipes for ale and pilsen require different proportions of resources.

Type	Corn	Hop	Malt	Profit
Ale	5	4	35	13
Pilsen	15	4	20	23
Resources	480	160	1190	

How can the brewer maximize the profit?

Example 3: Statement

A Cooperative of Farmers (CoFar) wants to sold out 200 baskets of fruit and 100 baskets of vegetables. It is ready to promote two offers A and B.

Offer A is a pack of 1 fruits' basket and 1 vegetables' basket, with a price of 30€. Offer B is a pack of of 3 fruits' basket and 1 vegetables' basket, with a price of 50 €.

CoFar wants to have at least 20 packs for offer A and 10 pack for offer B.

CoFar wants to know how many packs of each type should be made to maximize the benefits

Example 4: Statement

A farmer has 10 ha available for growing potatoes and/or tomatoes in the combination that yields the highest profit.

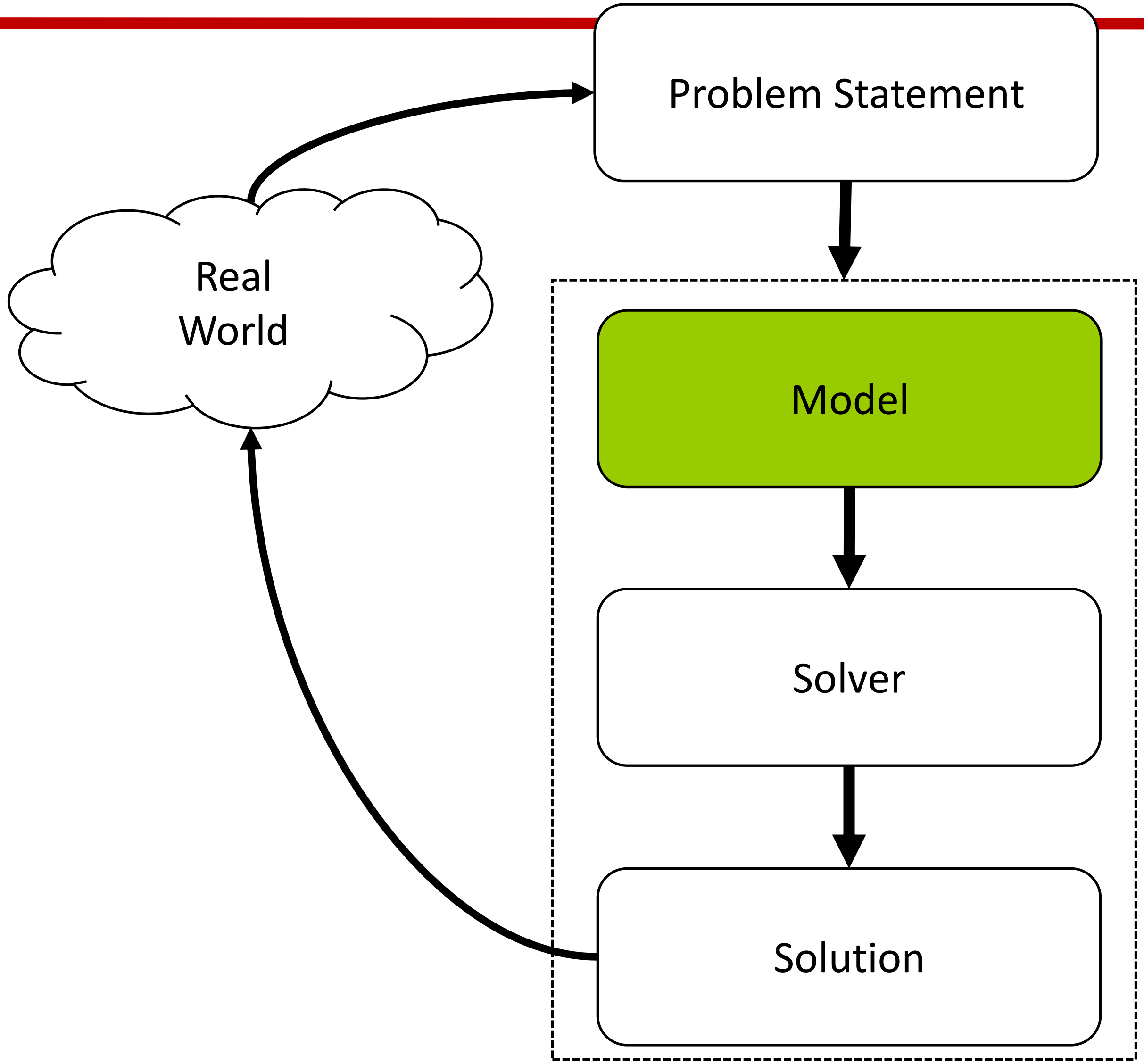
A special contract with a tomato ketchup factory regarding the supply of tomatoes requires production of at least 2 ha of tomatoes for the factory.

The farmer has the possibility of using 12 h per week for the cultivation of the 10 ha. Each hectare of potatoes requires 2 h per week, while the tomatoes require 0.5 h per week and hectare.

Potatoes provide total revenue of MU 4,000 per hectare, while tomatoes provide MU 3,000 per hectare.

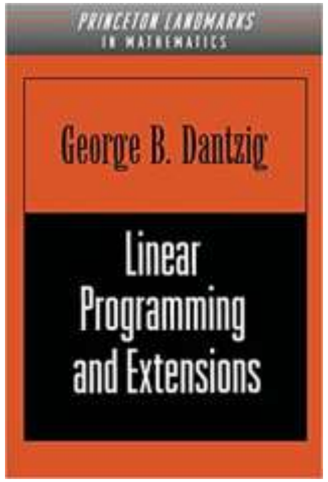
Fertiliser costs MU 1,000 per ton. and should be applied in the following amounts: 1 ton. per hectare for potatoes, and 0.5 ton. per hectare for tomatoes.

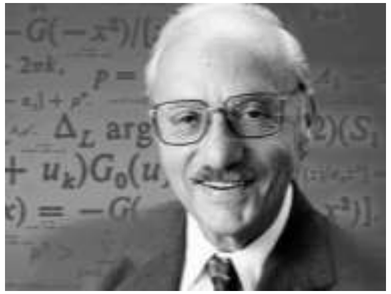
All these problems can be represented by a single model


Problem Statement
Real
World
Model
Solver
Solution

Linear Programming

- Developed by Dantzig in the late 1940’s
- A mathematical method of allocating scarce resources to achieve a single objective
- The objective may be profit, cost, return on investment, sales, market share, space, time…
- A wide range of applications.





Linear Programming: Elements

Decision Variables

Symbols used to represent an item that can take a value.

x_1 = number of hours

x_2 = number of workers

Parameters

Known values: price per unit, truck capacity, and so on.

Decision variables and parameters are problem dependant.

Linear Programming: Elements

The objective function

Either Maximize o Minimize

Maximize total production,
Maximize revenue/income,
Minimize cost,
Minimize labour force use,
Minimize fertilizer usage,
Minimize water usage,
Minimize total travelling distance of truck....

It is a **linear function** on the decision variables

- Maximize Benefit $Z = 3x_1 + 5x_2$
- Minimize Cost $Z = 6x_1 - 15x_2$

$f(x) = 5x + 1$
$g(x_1, x_2) = x_1 + x_2$

$f(x) = 5x^2 + 1$
$g(x_1, x_2) = x_1 x_2 + x_2$

Linear Programming: Elements

Constraints

- Total number of working hours should be less than 50
- Fat intake should be at least 120 units
- The available funding is 10000 euros.

Restrictions are defined using linear relations among the decision variables and/or between them and the parameters.

Linear Programming: Elements

Constraints

- There are 100 parcels. The number of parcels for wheat (x_1) should be higher than the one for soy (x_2).
  - $x_1 + x_2 \leq 100$
  - $x_1 > x_2$

- Fat intake should be at least 120 units. We have three foods x_1, x_2, x_3

$$\underbrace{0.25\, x_1 + 1.1\, x_2 + 0.67 x_3}_{\text{Left hand side (LHS)}} \geq \underbrace{120}_{\text{Right hand side (RHS)}}$$

Linear Programming: Elements

Non-negativity assumption

Decision variables should take positive values.

Such assumption is taken as a constraint.

- $x_1, x_2 \geq 0$
- $x_i \geq 0 \ \forall i = 1, \ldots, n$

LP: Formulation

Identify and define the decision variables

Identify the parameters

Determine the objective function

Define the constraints

Structure of a LP Problem

Maximize $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

Subject to $a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}\, x_n \leq b_1$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m$$
$$x_1, x_2, \ldots, x_n \geq 0$$

Maximize $c^T\boldsymbol{x}$

subject to $A\boldsymbol{x} \leq b$, $\boldsymbol{x} \geq 0$

where c, b are vectors of known coefficients
A is a matrix of known coefficients

Maximize $\sum_{j=1}^{n} c_j x_j$

Subject to $\sum_{j=1}^{n} a_{ij}x_j \leq b_i \,, 1 \leq i \leq m$

$$x_j \geq 0 \,, 1 \leq j \leq n$$

We have to determine the values of x_i

Let's see the modelization stage of the given examples.

Example 1: Statement

A farmer needs to transport his/her 400 hives.

A transport company has 8 trucks with 40 hives' capacity and 10 with 50 hives' capacity. The Company has 9 drivers available.

Renting the big small truck cost 600 €, while the bigger one costs 800 €.

The farmer wants to know the number of trucks of each type she/he should rent to make the transport as cheapest as posible.

Example 1: Formulation

Decision variables

$$x \ = number\ of\ small\ trucks$$
$$y \ = number\ of\ big\ trucks$$

Objective function

$$f(x, y) \ = \ 600x \ + \ 800y$$

Constraints

All the hives should go

$$40x \ + \ 50y \ \geq \ 400$$

No more trucks than existing ones. And no more than drivers available

$$x \leq 8$$
$$y \leq 10;$$
$$x \ + \ y \ \leq \ 9$$

Number of trucks should be positive

$$x, y \ \geq \ 0$$

Example 2: Statement

A small brewery produces ale and pilsen.

Production limited by scarce resources: corn, hops, barley malt.

Recipes for ale and beer require different proportions of resources.

Type	Corn	Hop	Malt	Profit
Ale	5	4	35	13
Pilsen	15	4	20	23
Resources	480	160	1190	

How can the brewer maximize profits?

Example 2: Formulation

Type	Corn	Hop	Malt	Profit
Ale	5	4	35	13
Pilsen	15	4	20	23
Resources	480	160	1190	

A = barrels of ale
B = barrels of pilsen

Corn
Hop
Malt

$$
\begin{array}{lrcrcr}
\max & 13A & + & 23B & & \\
\text{s.t.} & 5A & + & 15B & \le & 480 \\
 & 4A & + & 4B & \le & 160 \\
 & 35A & + & 20B & \le & 1190 \\
 & A & , & B & \ge & 0
\end{array}
$$

Devote all resources to beer: 32 barrels of pilsen $\Rightarrow$ \$736.
Devote all resources to ale: 34 barrels of ale $\Rightarrow$ \$442.
12 barrels of ale, 28 barrels of pilsen $\Rightarrow$ \$800.
7½ barrels of ale, 29½ barrels of pilsen $\Rightarrow$ \$776.

Example 3: Statement

A Cooperative of Farmers (CoFar) wants to sold out 200 baskets of fruit and 100 baskets of vegetables. It is ready to promote two offers A and B.

Offer A is a pack of 1 fruits' basket and 1 vegetables' basket, with a price of 30€. Offer B is a pack of of 3 fruits' basket and 1 vegetables' basket, with a price of 50 €.

CoFar wants to have at least 20 packs for offer A and 10 pack for offer B.

CoFar wants to know how many packs of each type should be made to maximize the benefits

Example 3: Formulation

Decision variables

$$x \;= number\ of\ packs\ A$$
$$y \;= number\ of\ packs\ B$$

Objective function

$$f(x,y) \;=\; 30x \;+\; 50y$$

Example 3: Formulation

Constraints

We can organize the information in a table like this:

	Pack A	Pack B	Stock
Fruits	1	3	200
Vegetables	1	1	100

$x + 3y \leq 200$ number of fruits' basket

$x + y \leq 100$ number of vegetables' basket

$x \geq 20$ at least 20 packs for offer A

$y \geq 10$ at least 10 packs for offer B

Example 4: Statement

A farmer has 10 ha available for growing potatoes and/or tomatoes in the combination that yields the highest profit.

A special contract with a tomato ketchup factory regarding the supply of tomatoes requires production of at least 2 ha of tomatoes for the factory.

The farmer has the possibility of using 12 h per week for the cultivation of the 10 ha. Each hectare of potatoes requires 2 h per week, while the tomatoes require 0.5 h per week and hectare.

Potatoes provide total revenue of MU 4,000 per hectare, while tomatoes provide MU 3,000 per hectare.

Fertiliser costs MU 1,000 per ton. and should be applied in the following amounts: 1 ton. per hectare for potatoes, and 0.5 ton. per hectare for tomatoes.

Example 4: Formulation

Decision variables

$$T \ = number\ of\ ha\ for\ tomatoes$$
$$P = number\ of\ ha\ for\ potatoes$$

Objective function

- The profit (the gross margin) is calculated for each of the crops by deducting costs for fertiliser from the total revenue
- 1ha of tomatoes provides a profit of 3000 – cost of the fertilizer (0.5 x 1000)
- 1ha of potatoes provides a profit of 4000 – cost of the fertilizer (1 x 1000)

$$f(T,P) = 2500\,T \ + 3000\,P$$

Example 4: Formulation

Constraints

10 ha available

$$T + P \leq 10$$

Production of at least 2 ha of tomatoes for the ketchup factory

$$T \geq 2$$

12 h per week available for the cultivation of the 10 ha. Each hectare of potatoes requires 2 h per week. Tomatoes require 0.5 h per week and hectare.

$$0.5\,T + 2\,P \leq 12$$

Example 4: Full Model

$$f(T,P) = 2500\,T \; + 3000\,P$$

$$T + P \leq 10$$

Land

$$T \geq 2$$

Contract

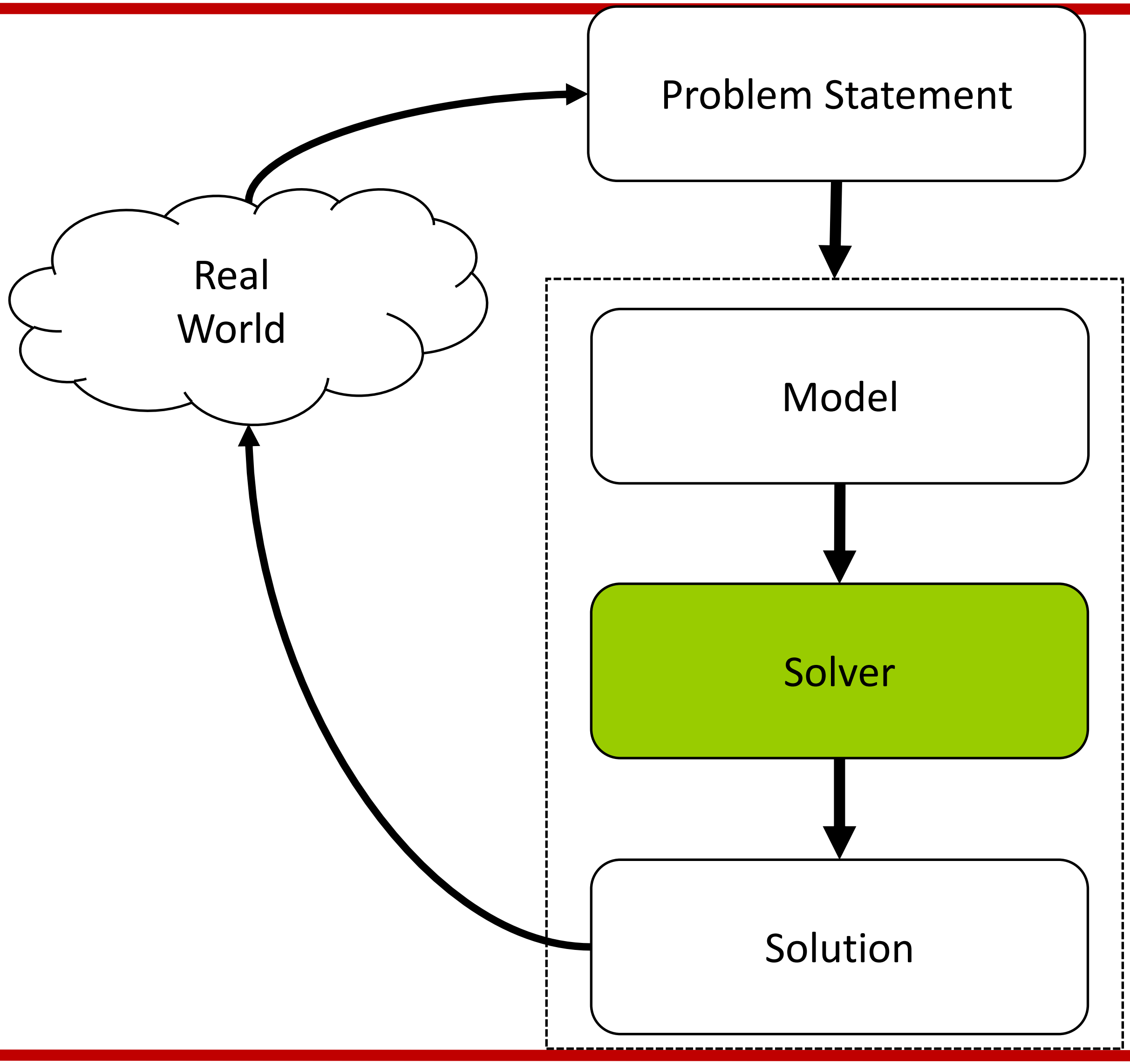
$$0.5\,T + 2\,P \leq 12$$

Labour

$$P, T \geq 0$$

Ok, I more or less understood formulation.

¡But now I want to solve the problems!

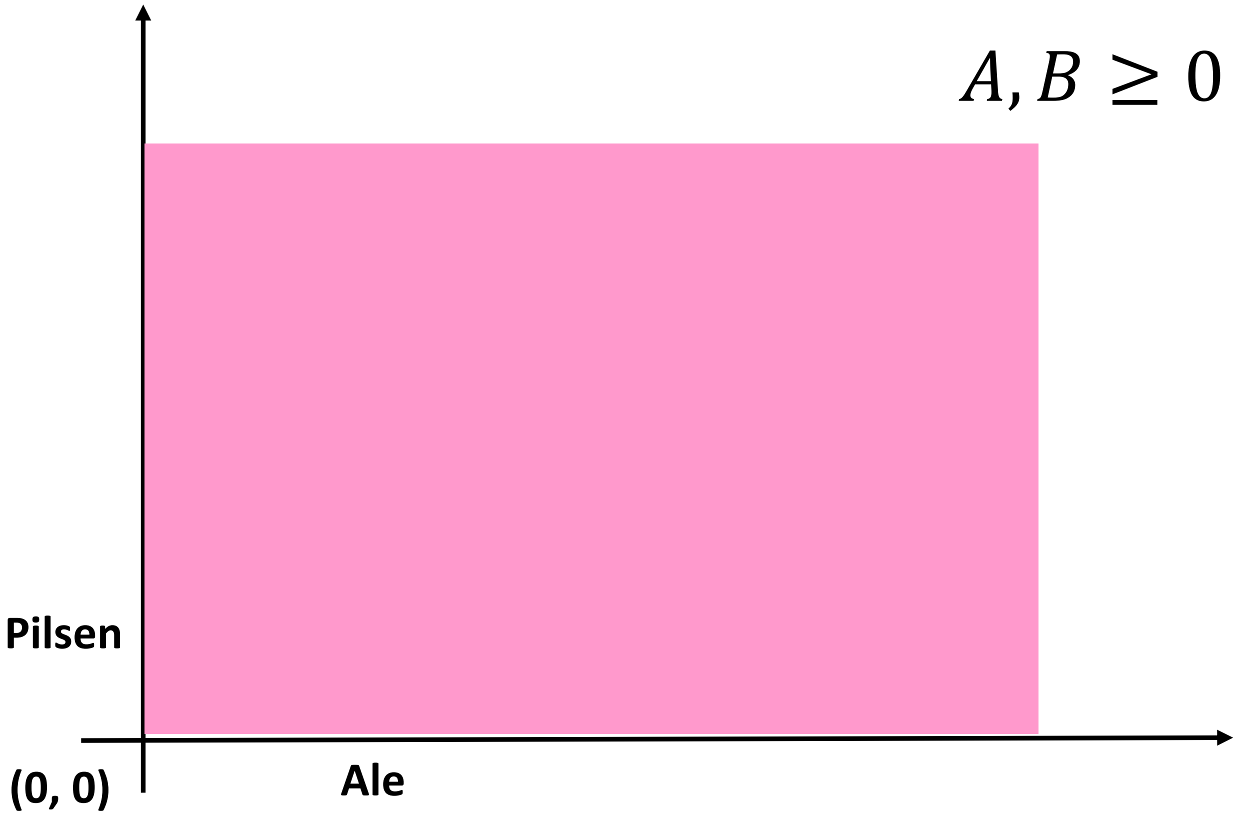

Problem Statement
Real
World
Model
Solver
Solution

The Graphical Method

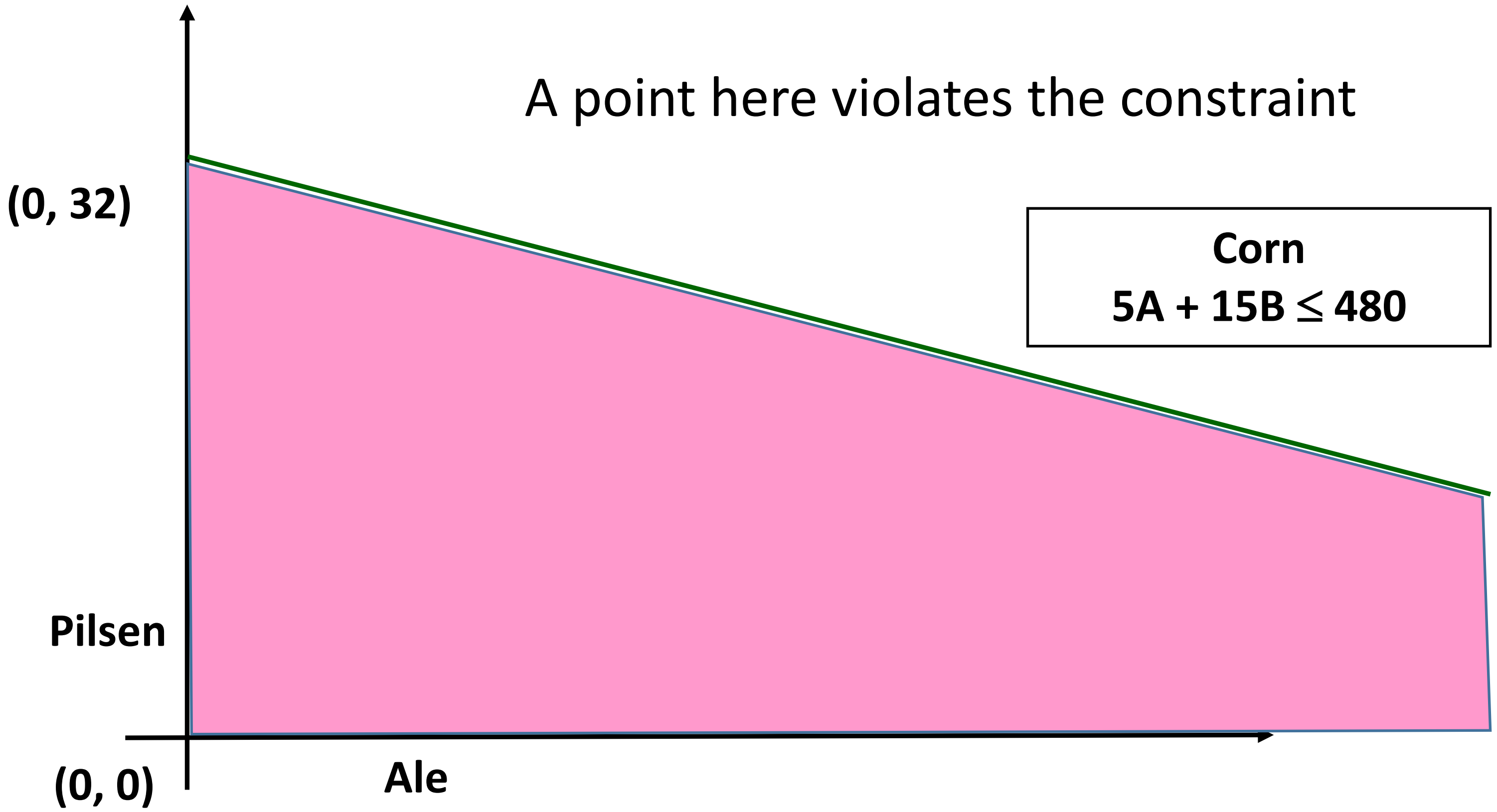
Brewery Problem

$$
\begin{array}{rrcrcr}
\max & 13A & + & 23B & & \\
\text{s.t.} & 5A & + & 15B & \leq & 480 \\
 & 4A & + & 4B & \leq & 160 \\
 & 35A & + & 20B & \leq & 1190 \\
 & A & , & B & \geq & 0
\end{array}
$$

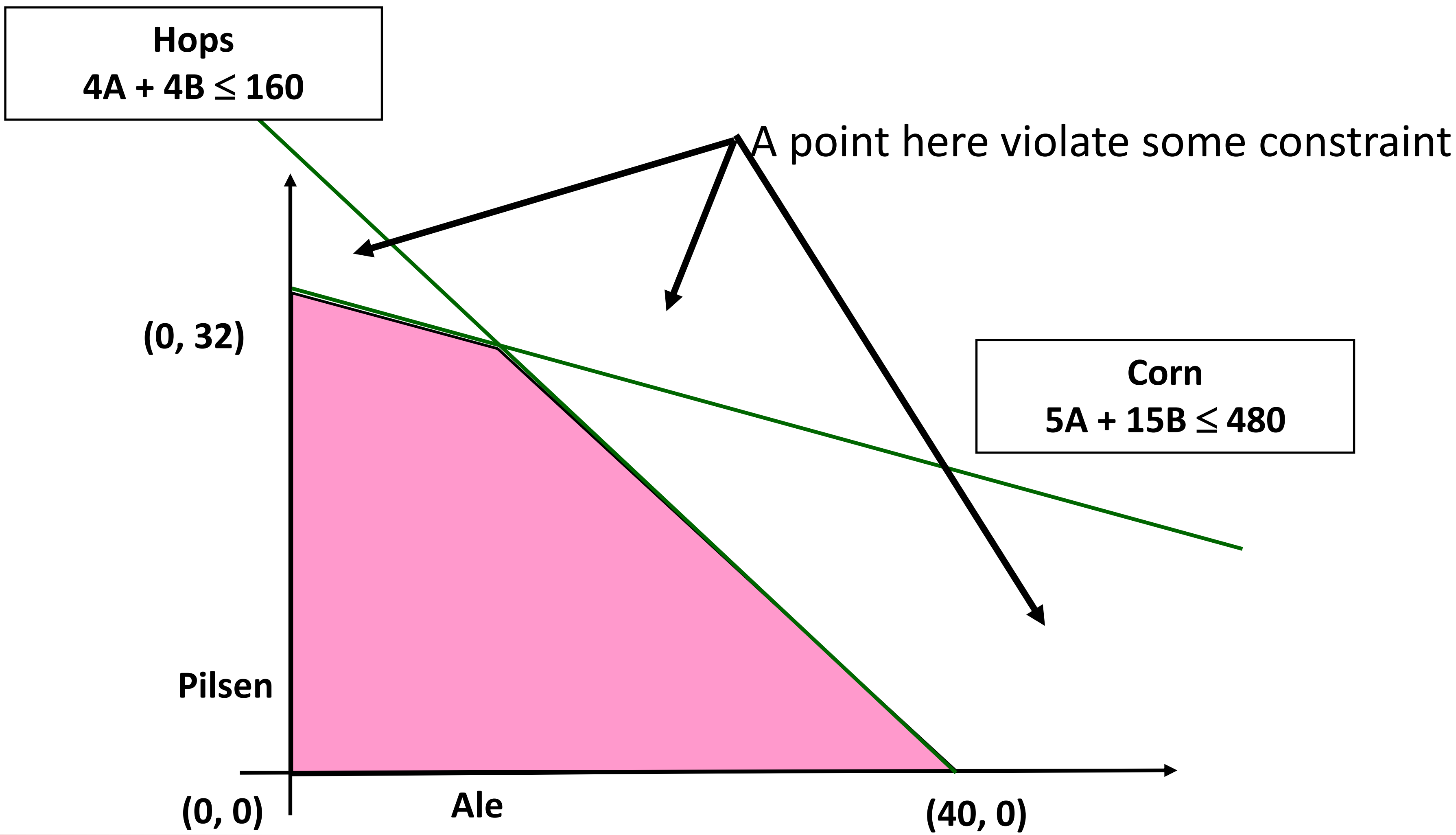
Brewery Problem: Feasible Region



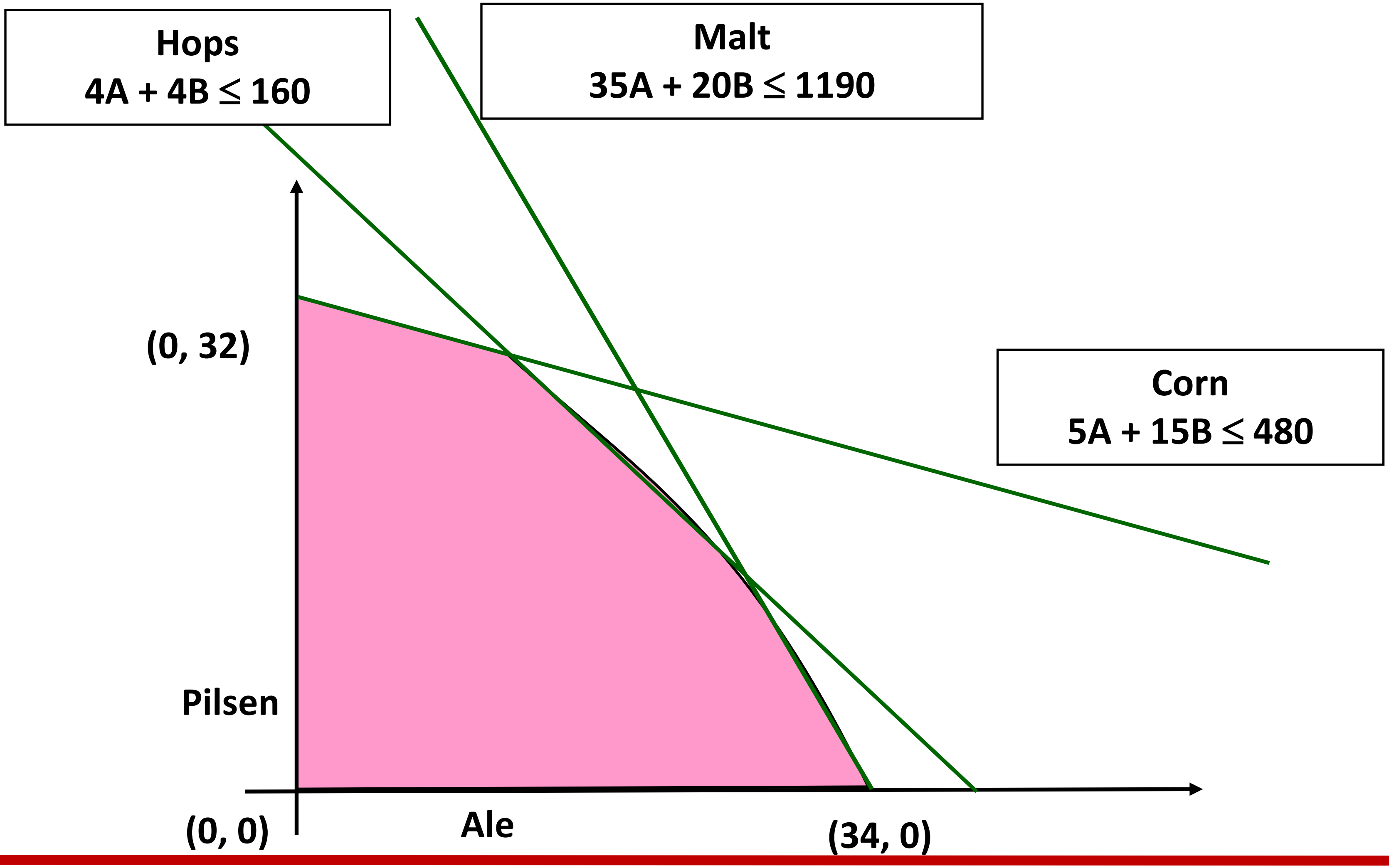
Brewery Problem: Feasible Region



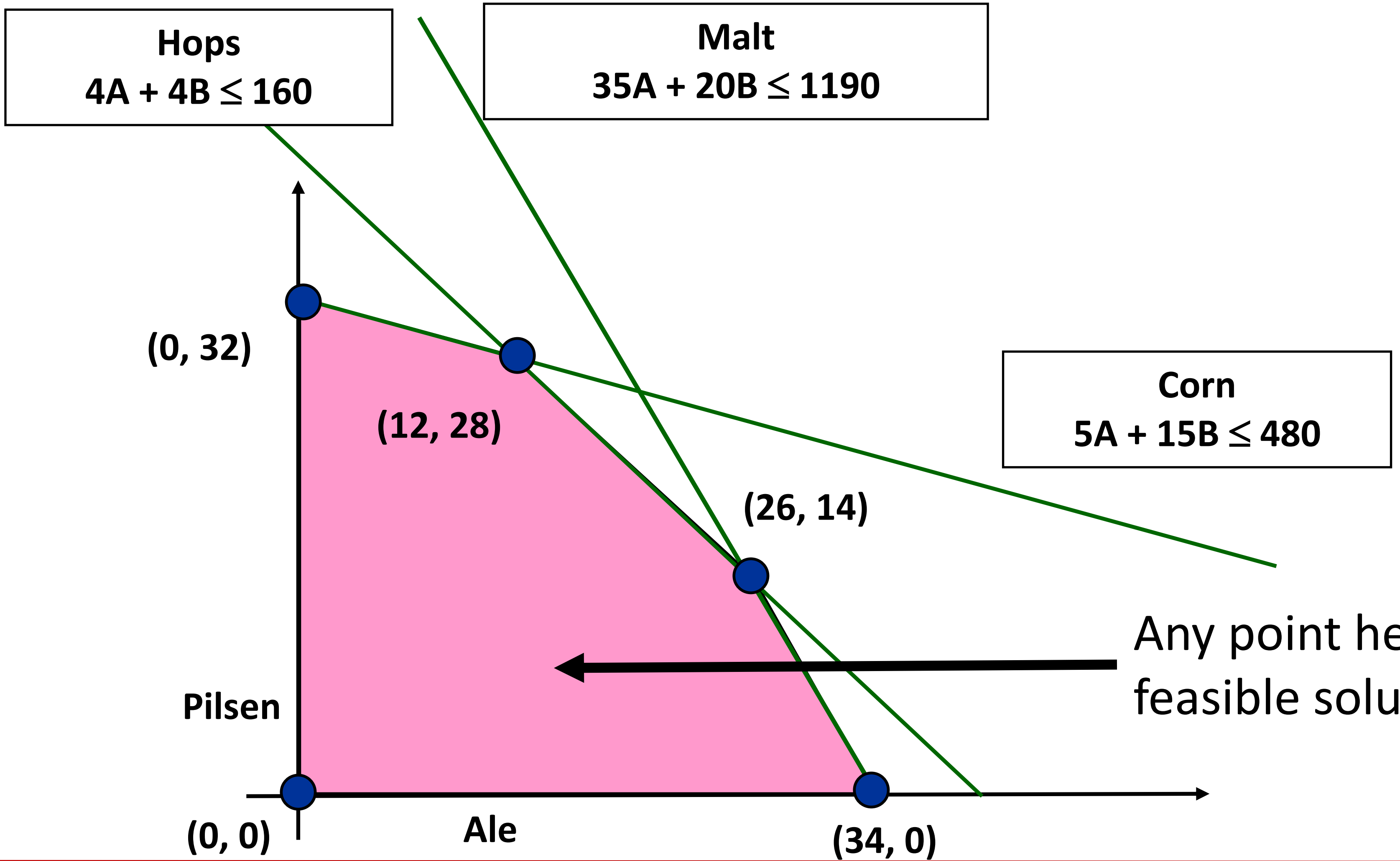
Brewery Problem: Feasible Region



Brewery Problem: Feasible Region

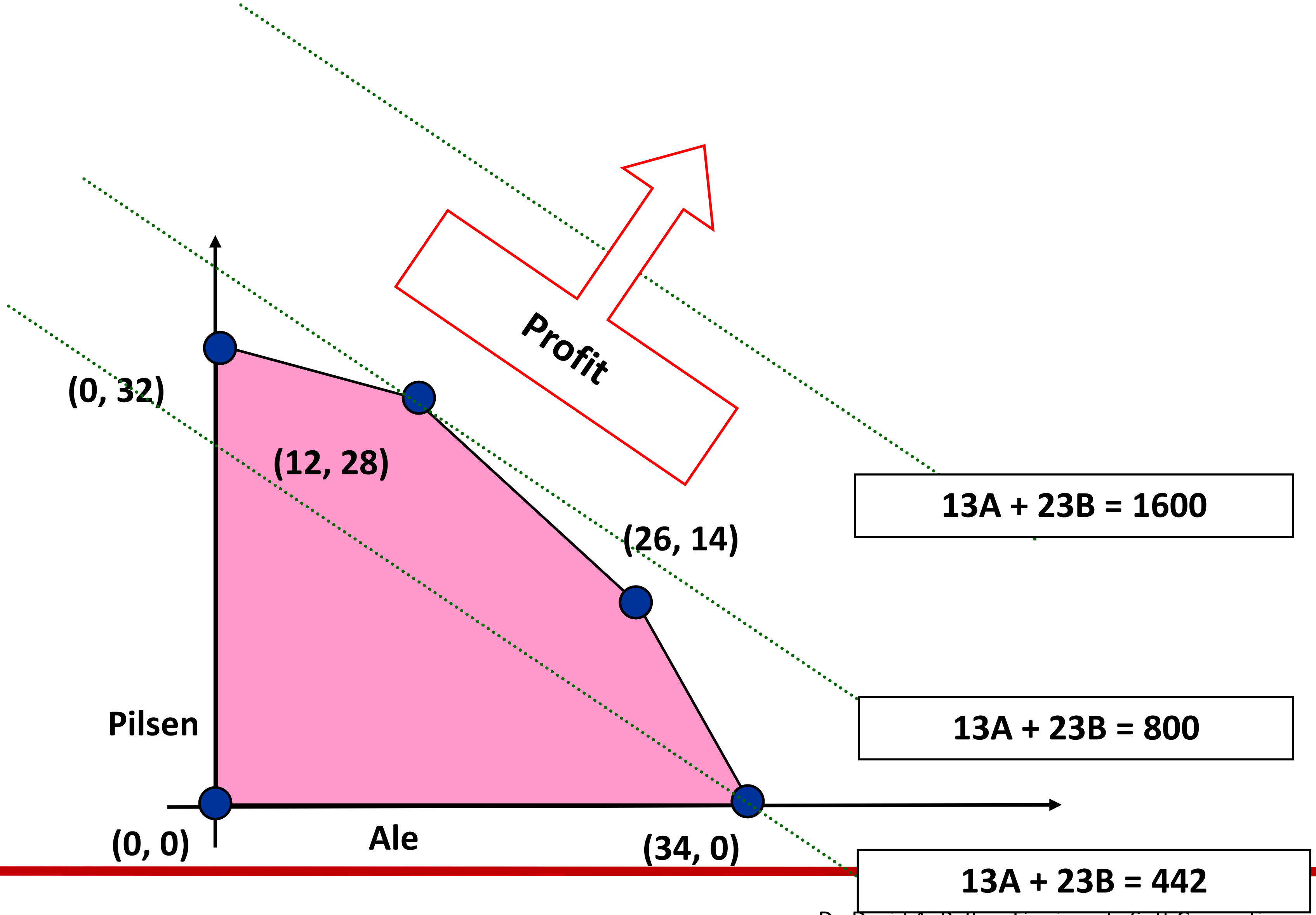


Brewery Problem: Feasible Region



Any point here is a feasible solution.

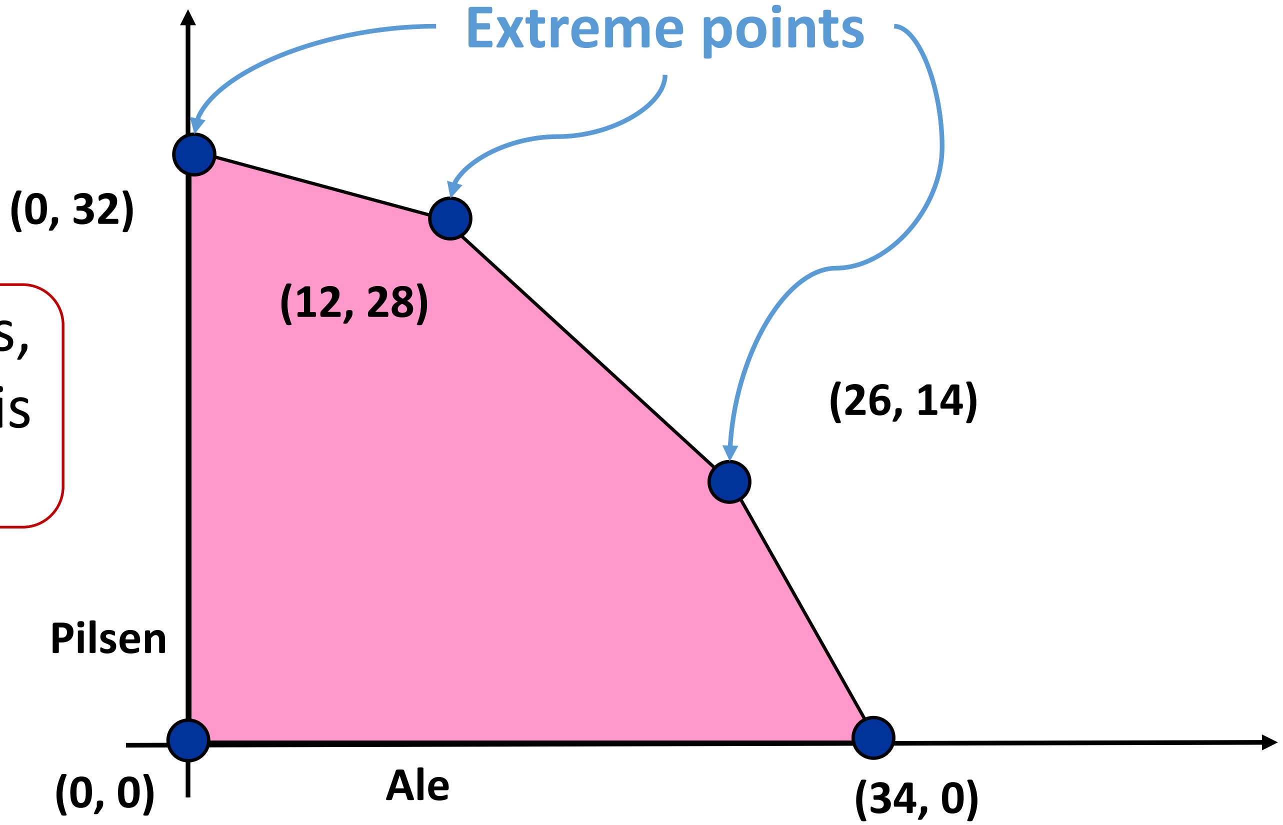
Objective Function



Brewery Problem: Geometry

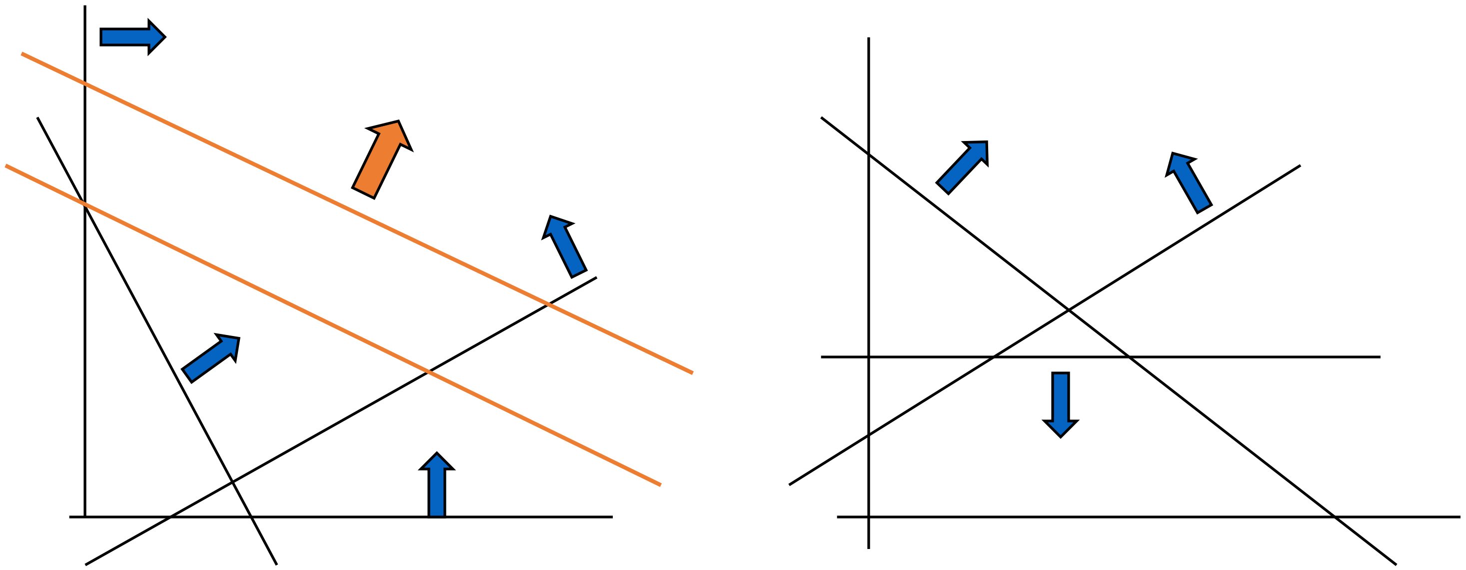
Regardless of the coefficients of linear objective function, there exists an optimal solution that is an extreme point.

In other words, the best solution is an extreme point.



Unbounded or Infeasible Case

- On the left, the objective function is unbounded
- On the right, the feasible set is empty



Graphical Solution Seeking

- Plot the feasible region.
- If the region is empty, stop: the problem is infeasible; there must be conflicting constraints in the model.
- Plot the objective function contour and choose the optimizing direction.
- Determine whether the objective value is bounded or not. If not, stop: the problem is unbounded; there must be mistakes in model formulation.
- Determine an optimal corner point.
- Identify active constraints at this corner.
- Solve simultaneous linear equations for the optimal solution.
- Evaluate the objective function at the optimal solution to obtain the optimal value of the problem.

mmm....

But if I have more than two variables? What??

The Simplex Method

- The simplex method is an algorithm for solving problems in linear programming.
- Invented by George Dantzig in 1947. Tests adjacent vertices of the feasible set in sequence so that at each new vertex the objective function improves or is unchanged.
- Very efficient in practice, generally taking $2m$ to $3m$ iterations at most (where m is the number of equality constraints), and converging in expected polynomial time for certain distributions of random inputs (Nocedal and Wright 1999, Forsgren 2002).
- However, its worst-case complexity is exponential, as can be demonstrated with carefully constructed examples (Klee and Minty 1972).

Other techniques

- Interior point methods.
- The complexity is polynomial for both average and worst case.
- These methods construct a sequence of strictly feasible points (i.e., lying in the interior of the polytope but never on its boundary) that converges to the solution.
- Research on interior point methods was spurred by a paper from Karmarkar (1984).
- In practice, one of the best interior-point methods is the predictor-corrector method of Mehrotra (1992), which is competitive with the simplex method, particularly for large-scale problems.

Basic Implementations

Online versions

http://www.phpsimplex.com

http://www.zweigmedia.com/RealWorld/simplex.html

Solving Problems using Excel

FrontlineSolvers®
The Leader in Analytics for Spreadsheets and the Web

Products Examples Support Order Free Trial

ANALYTIC SOLVER BASIC

Home > Products > Analyst Pro Tools

Start Your Journey from *Business Analyst Pro* to *Analytics Ninja:* Learn with Analytic Solver Basic in Excel and AnalyticSolver.com.

Start with a free trial of **Analytic Solver Basic** - get access to *every feature* for Optimization, Simulation and Data Mining.
Then **keep** access to every feature for as long as you want, with size limits suitable for learning, at a low subscription price.
Use *both* desktop and cloud. When you're ready, **upgrade** (and pay for) greater size and speed, only in the area(s) you need.

Model/Data Element	Analytic Solver Basic	Analytic Solver Optimization	Analytic Solver Simulation	Analytic Solver Data Mining
Linear Variables x Constraints	200 x 100	**8000 x 8000** ***Unlimited*** *with Solver Engines*	*Basic 200 x 100*	*Basic 200 x 100*
Nonlinear Variables x Constraints	200 x 100	**1000 x 1000** ***Unlimited*** *with Solver Engines*	*Basic 200 x 100*	*Basic 200 x 100*
Uncertain Variables x Functions	50 x 25	*Basic 50 x 25*	**Unlimited x Unlimited**	*Basic 50 x 25*
Simulations x Monte Carlo Trials	10 x 1000	*Basic 10 x 1000*	**Unlimited x Unlimited**	*Basic 10 x 1000*
Data Partition Columns x Rows	50 x 65,000	*Basic 50 x 65,000*	*Basic 50 x 65,000*	**Unlimited x Unlimited**
Training Set / Database Query Rows	10,000 x 1,000,000	*Basic 10,000 x 1,000,000*	*Basic 10,000 x 1,000,000*	**Unlimited x Unlimited**
Processor Cores Used in Parallel	One	**Many** for Optimization	**Many** for Simulation	**Many** for Data Mining

OpenSolver (opensolver.org)

Welcome to OpenSolver, the Open Source linear, integer and non-linear optimizer for Microsoft Excel.

The latest stable version, **OpenSolver 2.8.6** (6 Mar 2017) is available for download. Refer to the release blog for the new 2.7, 2.8, and 2.8.3 improvements. View all releases.

OpenSolver uses the COIN-OR CBC optimization engine

OpenSolver is also available for Google Sheets

OpenSolver for Google Sheets; see our dedicated OpenSolver for Google Sheets page for more info on the Google Sheets versions of OpenSolver.

COIN-OR Cup Winner: We are pleased to announce that OpenSolver is the winner of the 2011 INFORMS COIN-OR Cup sponsored by IBM. Thanks, COIN-OR, for this honour.

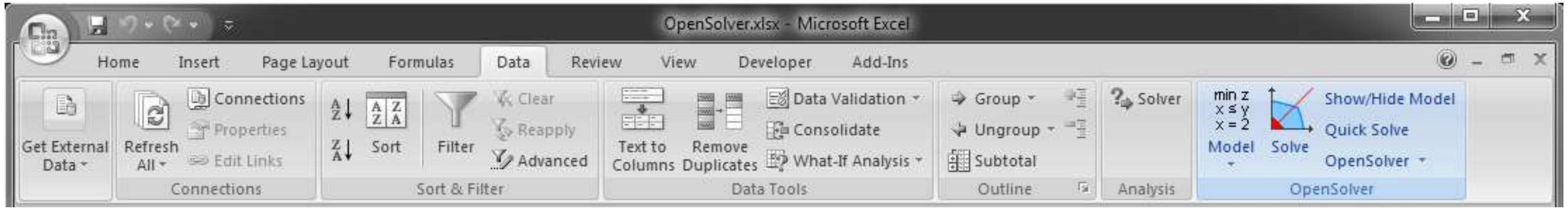
OpenSolver is an Excel VBA add-in that extends Excel's built-in Solver with more powerful solvers. It is developed and maintained by Andrew Mason and students at the Engineering Science department, University of Auckland, NZ. Recent developments are courtesy of Jack Dunn at MIT.

OpenSolver

- OpenSolver offers a range of solvers for use in Excel, including the excellent, Open Source, COIN-OR CBC optimization engine which can quickly solve large Linear and Integer problems.
- Compatible with your existing Solver models, so there is no need to change your spreadsheets
- No artificial limits on the size of problem you can solve
- OpenSolver is free, open source software.

> OpenSolver has been used to successfully solve problems which have as many as 70,000 variables and 76,000 constraints [6]. Further, users report that they appreciate being able to view the algebraic form of the Excel model given in the .lp file [4].

OpenSolver

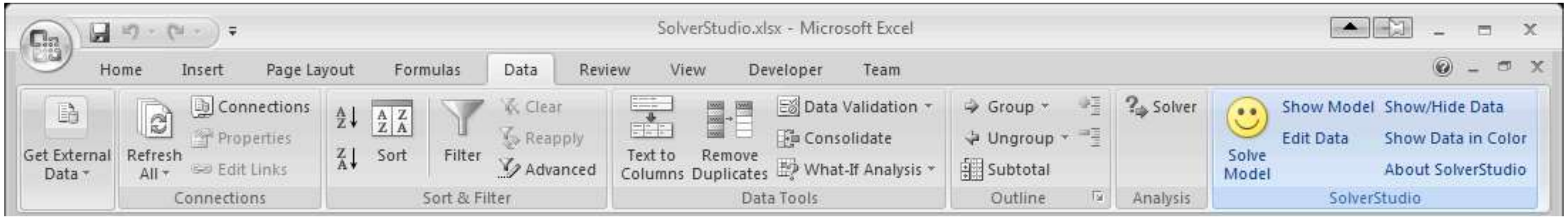


Mason, A.J., *"OpenSolver – An Open Source Add-in to Solve Linear and Integer Programmes in Excel"*, Operations Research Proceedings 2011, eds. Klatte, Diethard, Lüthi, Hans-Jakob, Schmedders, Karl, Springer Berlin Heidelberg, pp 401-406, 2012,

http://dx.doi.org/10.1007/978-3-642-29210-1_64
http://opensolver.org

SolverStudio (solverstudio.org)

- SolverStudio is an add-in for Excel 2007 and later on Windows that allows you to build and solve optimisation models in Excel using several optimisation modelling languages.
- Mason AJ (2013) SolverStudio: *A new tool for better optimisation and simulation modelling in Excel*. INFORMS Trans. Ed. 14(1):45–52.



¡A resolver!

Utilizando la herramienta Solver de Excel (o equivalente en LibreOffice):

- Resolver el problema de las colmenas
- Resolver el problema de las cestas de frutas

Producción de hormigón

- Una empresa produce hormigón usando los ingredientes A y B.
- Cada kilo de ingrediente A cuesta 60$ y contiene 4 unidades de arena fina, 3 unidades de arena gruesa y 5 unidades de grava.
- Cada kilo de ingrediente B cuesta 100$ y contiene 3 unidades de arena fina, 6 unidades de arena gruesa y 2 unidades de grava.
- Cada amasada debe contener, por lo menos, 12 unidades de arena fina, 12 unidades de arena gruesa y 10 unidades de grava.

Commercial Solvers

- Linear programming was revolutionized when CPLEX software was created over 20 years ago
- It was the first commercial linear optimizer on the market written in the C language, and it gave operations researchers unprecedented flexibility, reliability and performance to create novel optimization algorithms, models, and applications.
- CPLEX can solve integer programming, mixed-integer programming and quadratic programming problems, too.

IBM ILOG CPLEX Optimization Studio

IBM ILOG CPLEX Optimization Studio provides a model development toolkit for mathematical and constraint programming to optimize business decisions.

https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex

www.gurobi.com

GUROBI OPTIMIZATION

http://www.gurobi.com/

Support for all common problem types

- Linear Programming (LP)
- Mixed-Integer Linear Programming (MILP)
- Quadratic Programming (QP)
- Mixed-Integer Quadratic Programming (MIQP)
- Quadratically Constrained Programming (QCP)
- Mixed-Integer Quadratically Constrained Programming (MIQCP)

For University Users

If you are a student, faculty, or staff member at a recognized degree-granting institution, you may be eligible for a free, full-featured, no-size limit, academic version of Gurobi.

www.mosek.com

Home | Downloads | Documentation | Resources | Products | Sales | Support | About us

Mosek

Features

The MOSEK optimization software is designed to solve large-scale mathematical optimization problems. MOSEK main features are listed below. For additional questions, contact our support or browse the online documentation.

Problem types MOSEK can solve

- Linear.
- Conic quadratic.
- Semi-definite (Positive semi-definite matrix variables).
- Quadratic and quadratically constrained.
- General convex nonlinear.
- Mixed integer linear, conic and quadratic.

Technical highlights

- Problem size limited only by the available memory.
- Primal and dual simplex optimizers for linear programming.
- Highly efficient pre-solver for reducing problem size before optimization.
- Branch&bound&cut algorithm for mixed integer problems.

Strengths and features of MOSEK

- The strongest point of MOSEK is its state-of-the-art interior-point optimizer for continous linear, quadratic and conic problems.
- The optimizer is parallelized and capable of exploiting multiple CPUs/cores.
- The optimizer is run-to-run deterministic.
- Reads and writes industry standard formats such as the MPS, CBF and LP formats.
- Includes tools for infeasibility diagnosis, repair and sensitivity analysis for linear problems.
- Ships with an optimization server for remote optimization.

Other approaches

- Matlab
- R
- Python

In general, these solvers requires

- some computer programming knowledge.
- the use of an algebraic modeling language (like GAMS, AMPL, etc.) to describe the model of the problem.

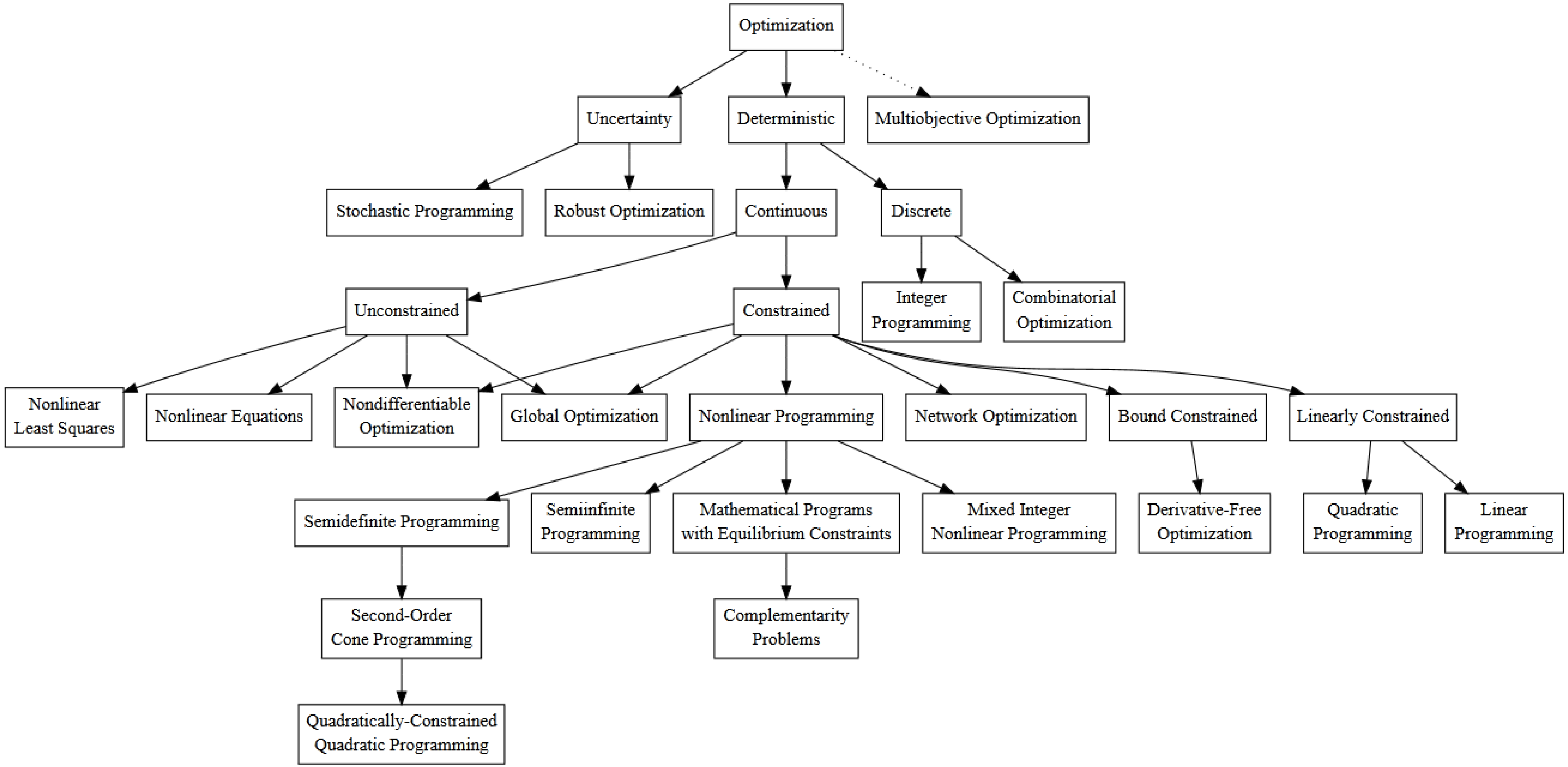
```
Sets
       i   canning plants    / Seattle, San-Diego /
       j   markets           / New-York, Chicago, Topeka / ;
Parameters
       a(i)  capacity of plant i in cases
         /    Seattle     350
              San-Diego   600  /
       b(j)  demand at market j in cases
         /    New-York    325
              Chicago     300
              Topeka      275  / ;
Table  d(i,j)  distance in thousands of miles
                  New-York       Chicago      Topeka
    Seattle          2.5           1.7          1.8
    San-Diego        2.5           1.8          1.4  ;
Scalar f  freight in dollars per case per thousand miles  /90/ ;
Parameter
       c(i,j)  transport cost in thousands of dollars per case ;
c(i,j) = f * d(i,j) / 1000 ;
Variables
     x(i,j)  shipment quantities in cases
     z       total transportation costs in thousands of dollars ;
Positive variables x ;
Equations
     cost        define objective function
     supply(i)   observe supply limit at plant i
     demand(j)   satisfy demand at market j ;
cost ..        z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..   sum(i, x(i,j))  =g=  b(j) ;
Model transport /all/ ;
Solve transport using LP minimizing z ;
```

A Word of Caution

In the basic approach:

- We are assuming that everything is exactly known.
- We are not considering uncertainty, imprecision, ignorance…
- In principle, the variables take real values. But other options are posible: binary, integer, mixed.

The “world” of optimization problems

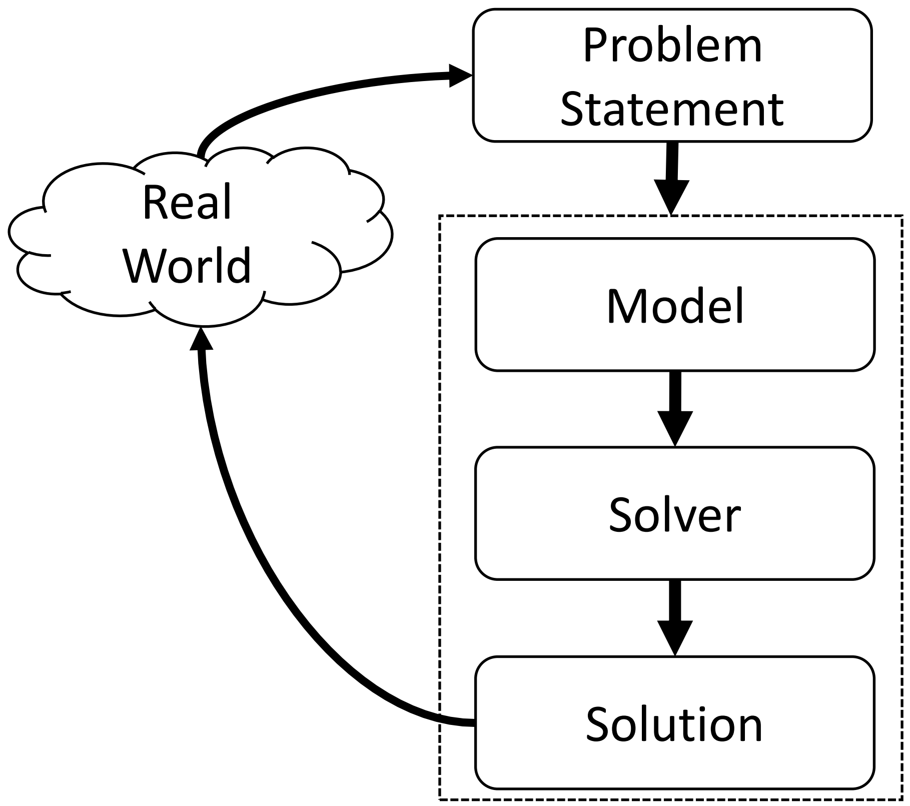


https://neos-guide.org/Optimization-Guide

Final Comments

We have just saw a quick overview on a specific type of optimization problem: linear programming problems.

Enough to have an insight of what can be done.



But remember:

- Someone decides which features are included /excluded from the model,
- What do you want to optimize?,
- We should not linearize the problem to make it fit under the linear programming framework.
- The better the model, the better the solution will be for the real world.
- Where the information come from??
- Errors, imprecision, uncertainty, etc. should be properly managed