Simulation and Optimization for Decision Support (G54SOD)

Semester 2 of Academic Session 2017-2018

Dr Dario Landa-Silva

dario.landasilva@nottingham.ac.uk

http://www.cs.nott.ac.uk/~pszjds

Lecture 6 – Introduction to Heuristics

- Introduction to Optimization

  To understand the basic concepts of optimization

- Basics of Heuristic Optimization

  To describe the key principles and mechanisms of heuristic optimization

Additional Reading

Chapters 1-2 of (Hillier and Lieberman, 2015)

Chapters 1 and 2 of the book (Talbi, 2009)

An overview of heuristic solution methods. E. A. Silver. *Journal of the Operational Research Society*, Vol. 55, pp. 936-956, 2004.

Metaheuristics in combinatorial optimization: overview and conceptual comparison. C. Blum, A. Roli. *ACM Computing Surveys*, Vol. 35, No. 3, pp. 268-308, 2003.

Guidelines for the use of meta-heuristics in combinatorial optmization. A. Hertz, M. Widmer. *European Journal of Operational Research*, Vol. 151, No. 2, pp. 247-252, 2003.

Aim of Module

Achieve an understanding of modern heuristic search techniques with emphasis in tackling search and optimisation problems.

Heuristic Search and Optimisation refers to a set of computational techniques that aim to find good quality solutions to very difficult problems in search, optimisation, design, etc. while consuming a reasonable amount of computational resources.

Heuristic methods are AI inspired approaches and are related both to computer science and operations research.

Heuristic methods have been successfully applied to many problems in different areas including: engineering, management, finance, planning and scheduling, medicine, biology, automated navigation, image processing, robotics, art design, etc.

Describing Heuristics

A heuristic search method is a technique that seeks good quality (i.e. near optimal) solutions at a reasonable computation time but that is not able to guarantee either feasibility or optimality.

There is a range of heuristic methods including: simple constructive heuristics, local search, meta-heuristics, hyper-heuristics, hybrids, evolutionary methods, etc.

Societies and Publications

- Related conferences include: CEC, GECCO, HM, MIC, PPSN, SLS
- Related journals include: Applied soft computing, Evolutionary computation, Evolutionary intelligence, IEEE Trans. on EC, Intl. journal of meta-heuristics, Journal of heuristics, Memetic computing, Swarm intelligence and others.

Introduction to Optimization

Computational Optimization

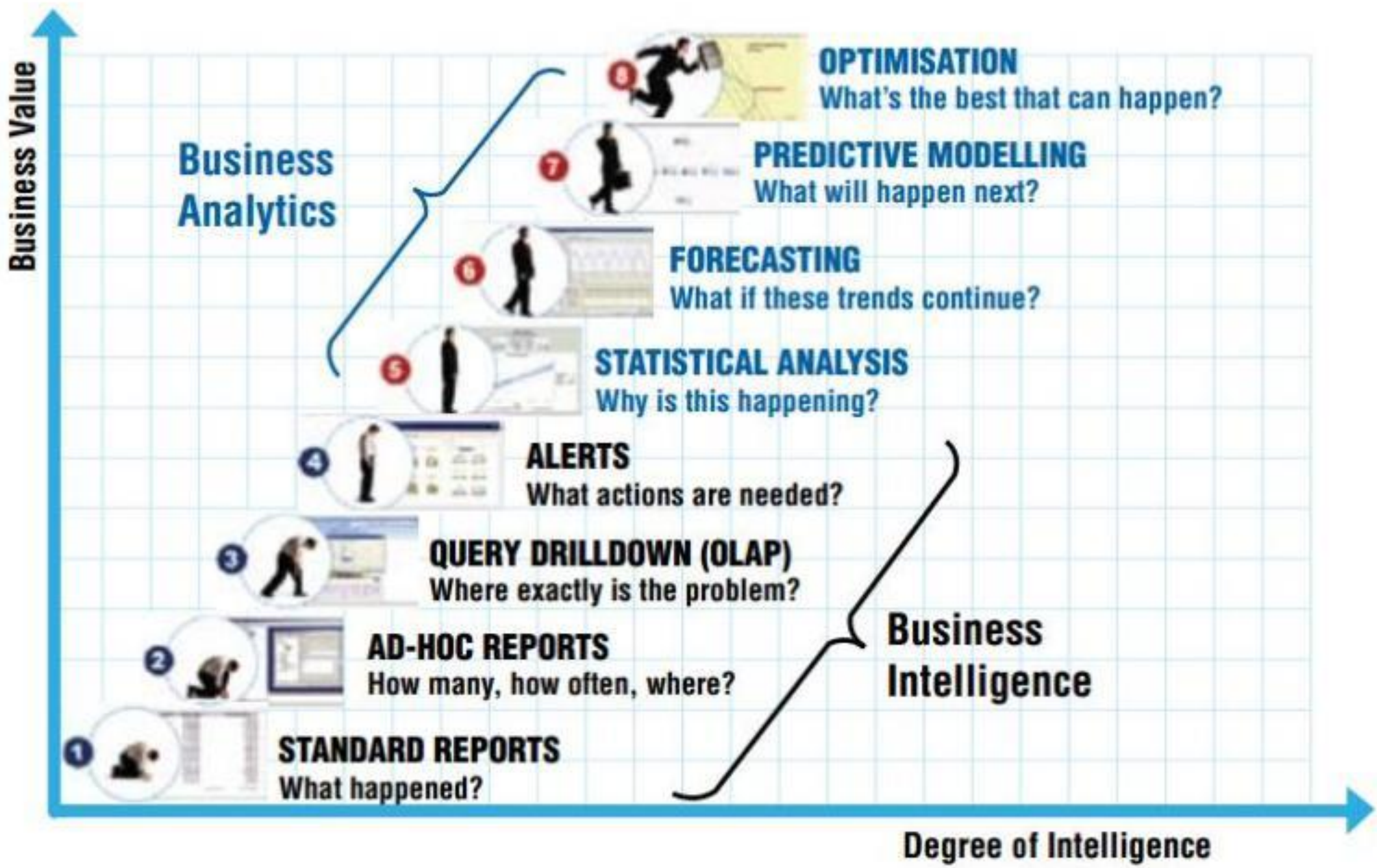
Often, optimization problems arise in industry and businesses, particularly in operational scenarios for which decision makers typically need an optimal solution.

Formal problem modelling is required in an optimization context and this provides several benefits.

Recently, analytics has been associated to operations research and management science.

According to SAS, a provider of Analytics Software and Solutions, there are 8 levels of analytics.

Optimization is the Highest Level of Analytics



Optimization Problems

Given a set of decision variables, an objective function and a set of constraints, an optimization problem is to find values for the decision variables in order to maximize or minimize the objective function.

Solutions (set of values for the decision variables) that satisfy all the constraints are called feasible solutions, otherwise they are called infeasible solutions.

Solving an optimization problem refers to finding the optimal solution(s) that gives the optimal objective function value.

A Simple Optimization Problem

The BANK ABC offers 4 types of loans to their customers at the annual interest rates shown below.

1. First mortgage at 14%
2. Second mortgage at 20%
3. Home improvement at 20%
4. Personal overdraft at 10%

The bank has a maximum lending capability of £250 million. It also has the following policies:

1. First mortgages must be at least 55% of all mortgages issued and at least 25% of all loans issued.
2. Second mortgages cannot exceed 25% of all loans issued.
3. To avoid a new windfall tax, the overall interest earned over all loans must not exceed 15% of the total amount of loans issued.

The Bank wants to maximise the interest income while satisfying all the above policies. The problem is to determine the amount that should be allocated to each type of loan in order to maximize the interest income, which is calculated simply as IR1×X1 + IR2×X2 + IR3×X3 + IR4×X4, where IR is the interest rate and X is the amount loaned.

Example. Identifying elements in the Bank ABC LP model:

Measure of solution quality

Competing activities

Limited resources

$$\text{Maximize}: \quad Z = 0.14x_1 + 0.20x_2 + 0.20x_3 + 0.10x_4 \quad (1)$$

$$\text{Subject to}: \quad x_1 + x_2 + x_3 + x_4 \leq 250 \quad (2)$$

$$0.45x_1 - 0.55x_2 \geq 0 \quad (3)$$

$$0.75x_1 - 0.25x_2 - 0.25x_3 - 0.25x_4 \geq 0 \quad (4)$$

$$-0.25x_1 + 0.75x_2 - 0.25x_3 - 0.25x_4 \leq 0 \quad (5)$$

$$-0.01x_1 + 0.05x_2 + 0.05x_3 - 0.05x_4 \leq 0 \quad (6)$$

$$x_1, x_2, x_3, x_4 \geq 0 \quad (7)$$

Optimal solution (maybe not unique):
X_1 = 62.5, X_2 = 51.13, X_3 = 48.86, X_4 = 87.5
Z= 37.5

Constraints (functional and non-negativity

Format of General LP Formulation

$$\text{Maximize}: \quad Z = c_1x_1 + c_2x_2 + \ldots c_nx_n$$

$$\text{Subject to}:$$

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n &\le b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n &\le b_2 \\
\vdots \qquad\qquad\qquad\qquad &\;\;\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n &\le b_m \\
x_1 \ge 0, x_2 \ge 0 \ldots\; x_n \ge 0 &
\end{aligned}$$

Linear objective function

Linear functional constraints

Non-negativity constraints

No uncertainty in problem parameters ($c_1 \ldots c_n$, $a_{11} \ldots a_{mn}$, $b_1 \ldots b_m$)

The size of an optimization model can be estimated by:

- The number of decision variables
- The number of constraints
- The size of the search space

Example. Estimate the size of the optimization model.

$$
\begin{array}{lll}
\text{Maximize}: & Z = 0.14x_1 + 0.20x_2 + 0.20x_3 + 0.10x_4 & (1) \\
\text{Subject to}: & x_1 + x_2 + x_3 + x_4 \le 250 & (2) \\
 & 0.45x_1 - 0.55x_2 \ge 0 & (3) \\
 & 0.75x_1 - 0.25x_2 - 0.25x_3 - 0.25x_4 \ge 0 & (4) \\
 & -0.25x_1 + 0.75x_2 - 0.25x_3 - 0.25x_4 \le 0 & (5) \\
 & -0.01x_1 + 0.05x_2 + 0.05x_3 - 0.05x_4 \le 0 & (6) \\
 & x_1, x_2, x_3, x_4 \ge 0 & (7)
\end{array}
$$

Number of decision variables:

4

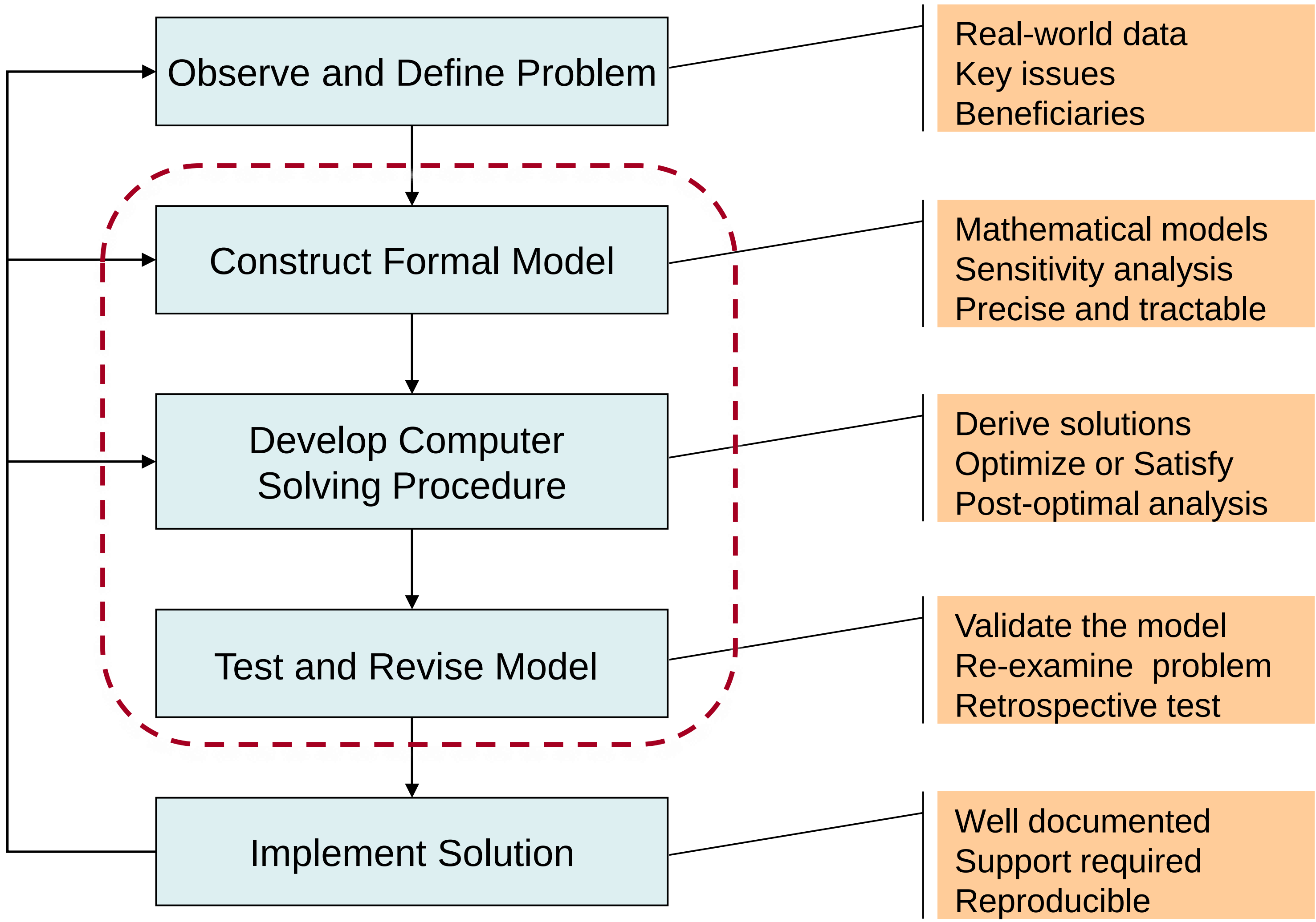
Number of Constraints:

9 including 4 non-negativity

Size of the search space:

Large number of possible values for x_1, x_2 , x_3 , x_4

Phases for Tackling an Optimization Problem



Mathematical Programming Optimization

Refers to modelling optimization problems using formulations based on mathematical expressions. Examples of these are:

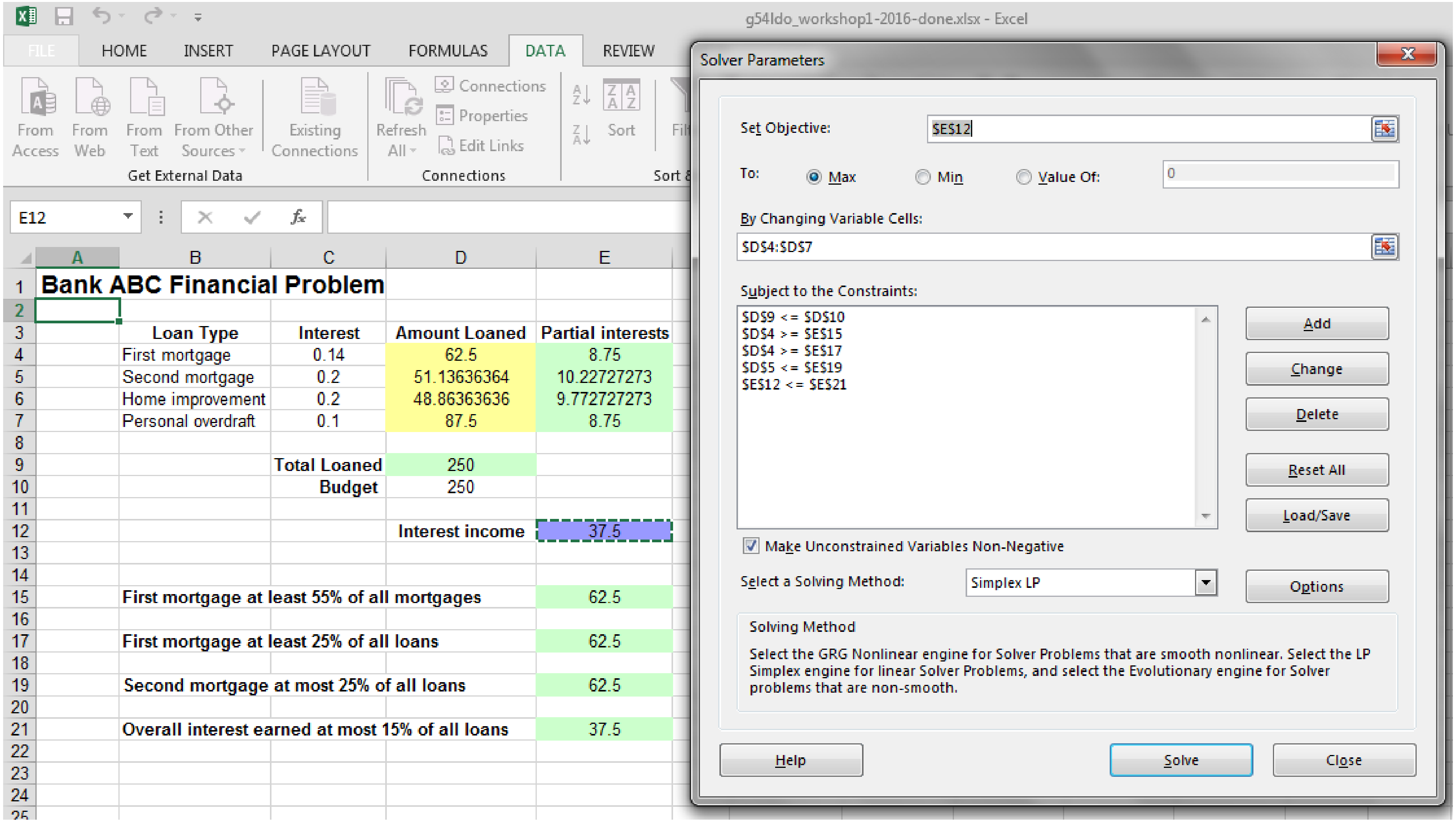
- Linear programming (LP) models
- Integer programming (IP) models
- Binary integer programming (BIP) models
- Mixed integer programming (MIP) models

Then, the models are solved with classical techniques such as:

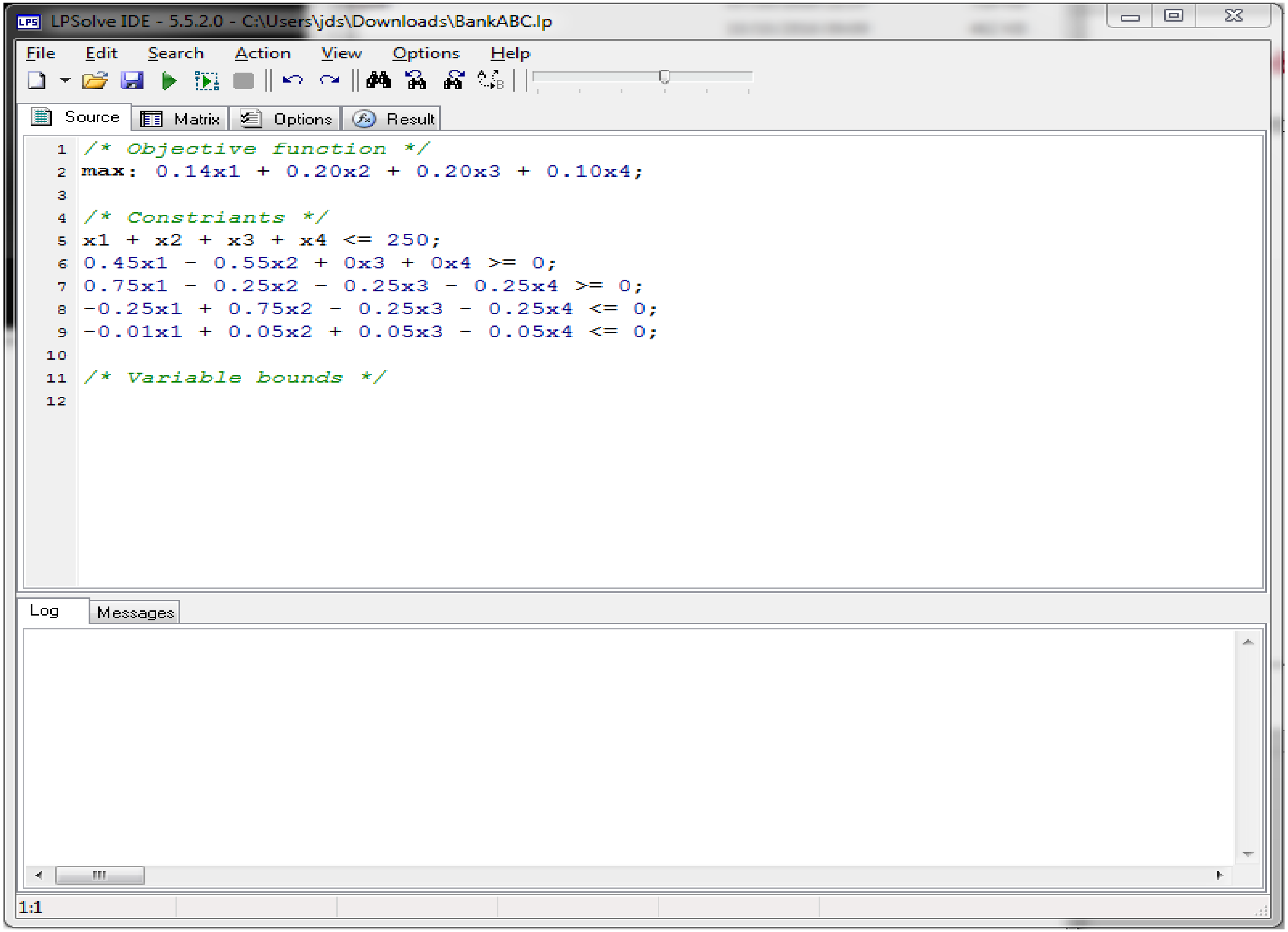
- Simplex method
- Branch and bound
- Branch and cut
- Dynamic programming

Even with improved versions of the above techniques, solving very large optimization problems might not be practical.

Spreadsheet model for the BANK ABC optimization problem.



LP-Solve model for the BANK ABC optimization problem.



Example. Formulating an LP model for a simple optimization problem.

Measure of solution quality

Competing activities

The company ATLAS produces two products A and B from two raw materials M1 and M2. The problem is to plan the production, i.e. determine the amount to produce of A and B, in order to maximise the profit. The quantity produced of B cannot larger than the quantity produced of A by more than 1 unit. Also, the demand (required production) of product B is known to be at most 2 units. The availability of raw materials and their requirement for the production of products A and B are shown in the table.

Limited resources

Functional Constraints

Non-negativity Constraints → Non-negative production

Formal LP Model for the ATLAS Optimization Problem

	product A	product B	max availability
Material M1	6	4	24
Material M2	1	2	6
Profit per unit	5	4	

$$
\begin{array}{lll}
\text{Maximize:} & Z = 5x_1 + 4x_2 & (1) \\
\text{Subject to:} & 6x_1 + 4x_2 \leq 24 & (2) \\
 & x_1 + 2x_2 \leq 6 & (3) \\
 & -x_1 + x_2 \leq 1 & (4) \\
 & x_2 \leq 2 & (5) \\
 & x_1, x_2 \geq 0 & (6)
\end{array}
$$

Example (cont.) Graphical method to solve ATLAS LP model

Find point at the intersection of (2) and (3)

$6x_1 + 4x_2 = 24$
$x_1 + 2x_2 = 6$

$6x_1 + 4x_2 = 24$
$-2x_1 - 4x_2 = -12$

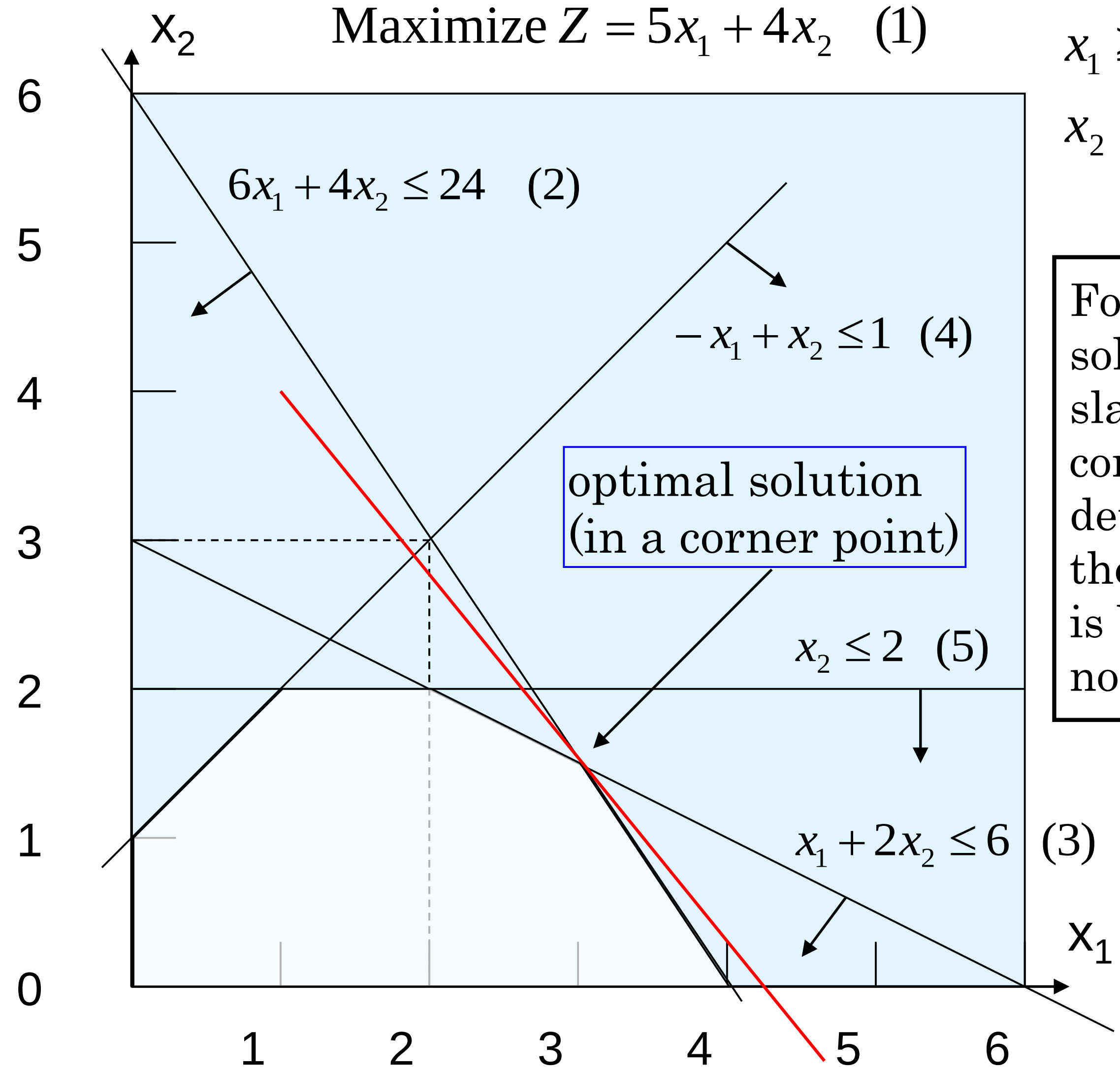
$4x_1 = 12$
$x_1 = 3$

$x_1 + 2x_2 = 6$
$3 + 2x_2 = 6$
$x_2 = 1.5$

$Z = 5x_1 + 4x_2$
$Z = 21$

Maximize $Z = 5x_1 + 4x_2$ (1)

$x_1 \geq 0$ and
$x_2 \geq 0$ (6)

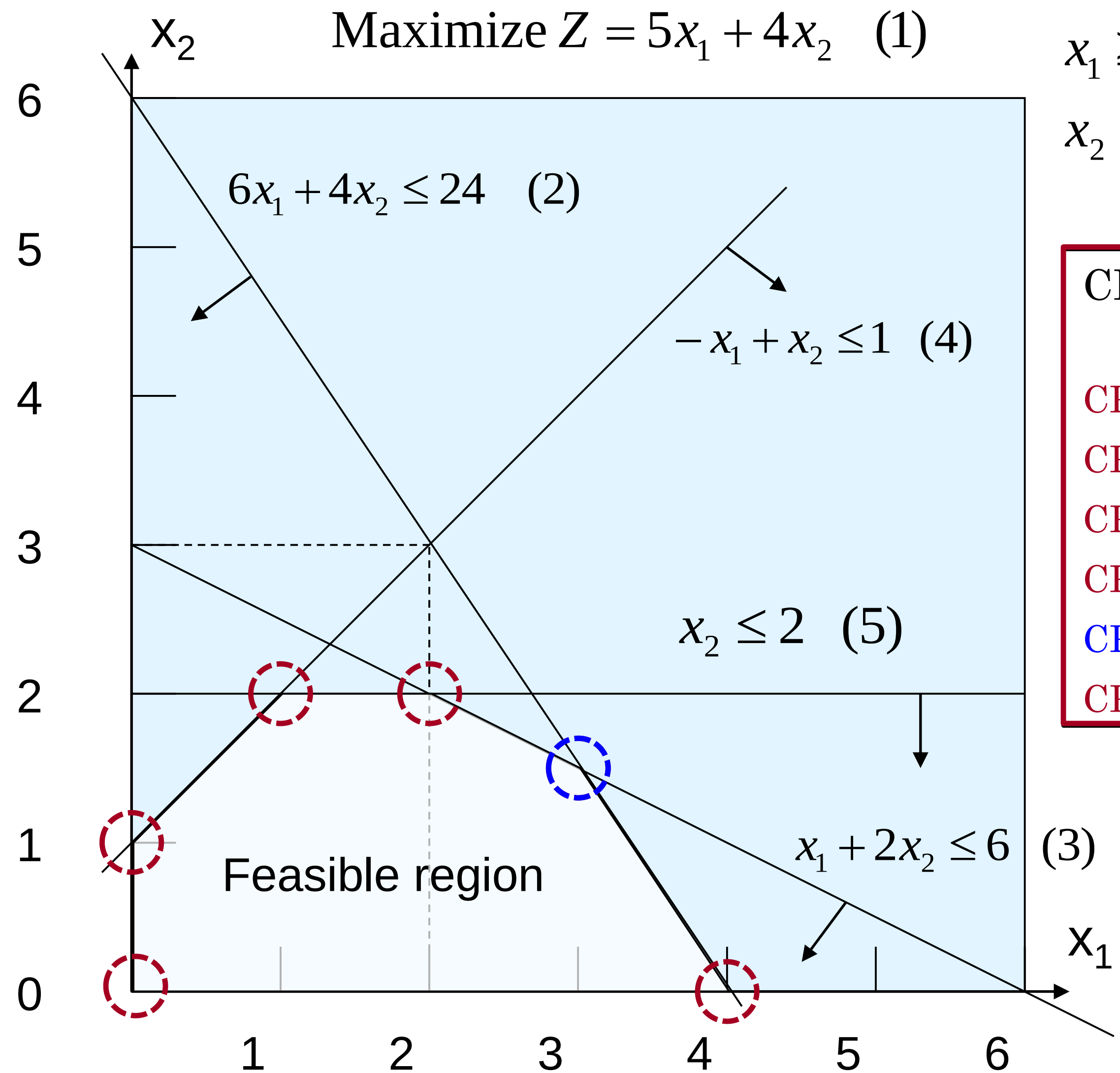


For the optimal solution, the slack in each constraint determines if the constraint is binding or non-binding.

Example (cont.) Graphical method to solve ATLAS LP model

An LP model with feasible solutions and a bounded feasible region must have CPF (corner-point feasible) solutions and at least one optimal solution.

The best CPF solution must be an optimal solution.



$x_1 \geq 0$ and

$x_2 \geq 0$ (6)

CPF solutions?

CPF (0,0) Z=0

CPF (0,1) Z=4

CPF (1,2) Z=13

CPF (2,2) Z=18

CPF (3,1.5) Z=21

CPF (4,0) Z=20

Example. The search space for the IP version of the ATLAS problem can be represented as a search tree.

$$\begin{aligned}
\text{Maximize:}\quad & Z = 5x_1 + 4x_2 & \\
\text{Subject to:}\quad & 6x_1 + 4x_2 \le 24 & (1)\\
& x_1 + 2x_2 \le 6 & (2)\\
& -x_1 + x_2 \le 1 & (3)\\
& x_2 \le 2 & (4)\\
& x_1, x_2 \ge 0 \text{ are integer} & (5)
\end{aligned}$$

Possible values:

$0 \le x_1 \le 4$

$0 \le x_2 \le 2$

Optimal values:

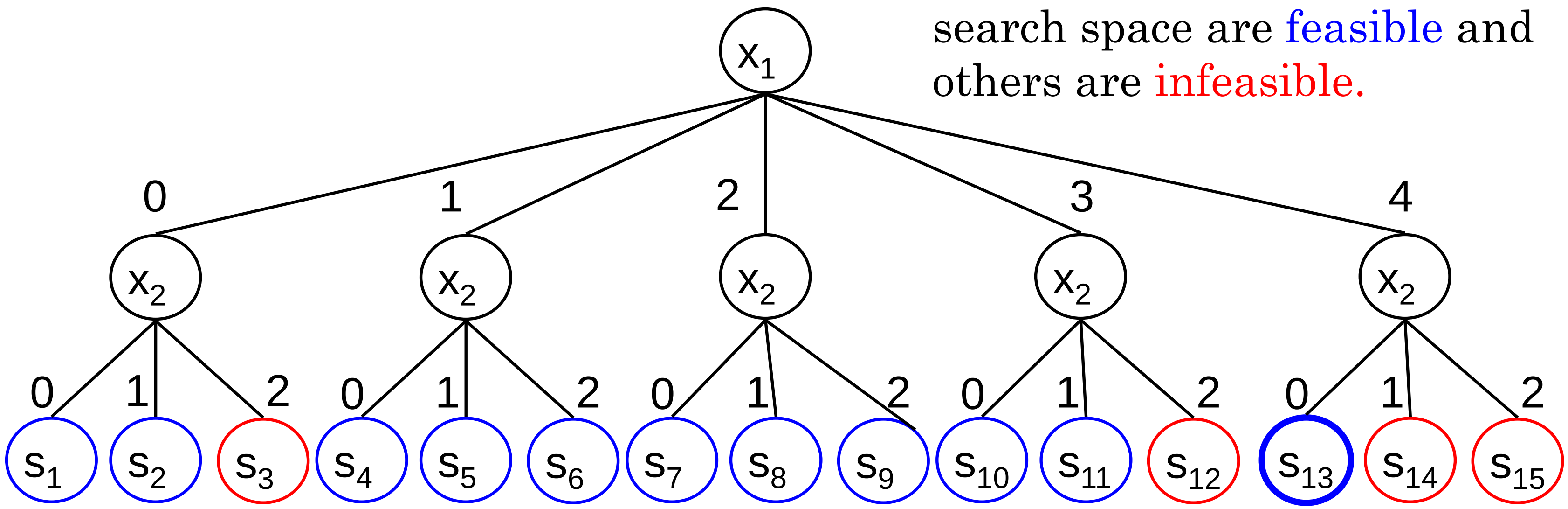
$x_1 = 4$

$x_2 = 0$

Considering all constraints, we have $0 \le x_2 \le 2$

Then, considering all constraints and $0 \le x_2 \le 2$ then we have $0 \le x_1 \le 4$

Some of the solutions in the search space are feasible and others are infeasible.

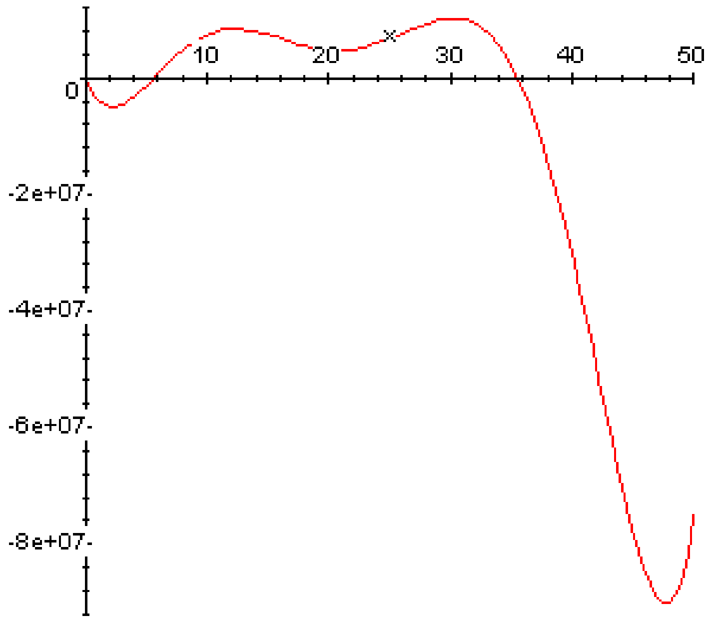


Non-linear Optimization Problems

There are many optimization problems for which it is not possible to develop a linear mathematical programming model.

$$\text{Minimize}: \quad Z = x^6 - 136x^5 + 6800x^4 - 155000x^3 + 1570000x^2 - 5000000x$$

$$\text{Subject to}: \quad 0 < x < 50$$



Examples of non-linear continuous optimization problems.

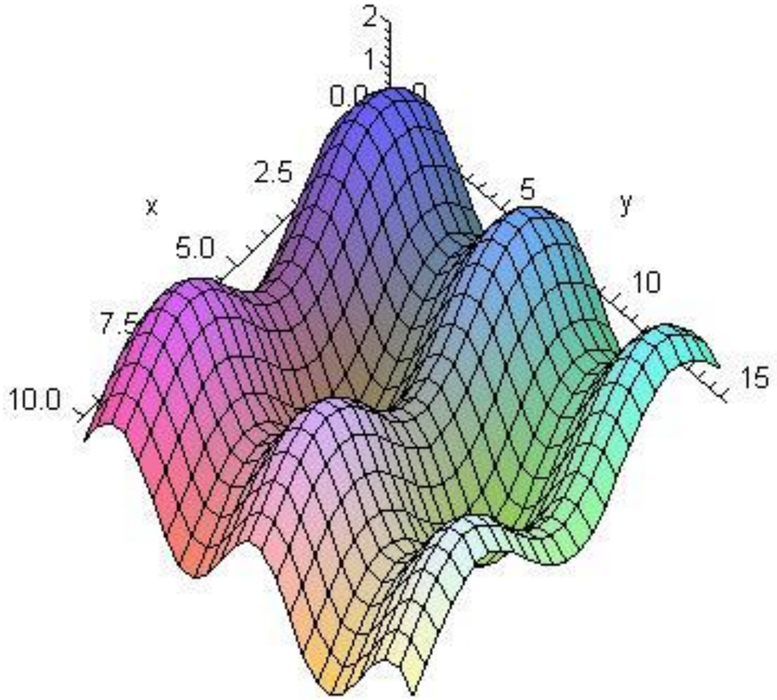
Maximize: $f(x,y) = x \cdot exp^{\left(-(x^2+y^2)+\frac{x^2+y^2}{20}\right)}$

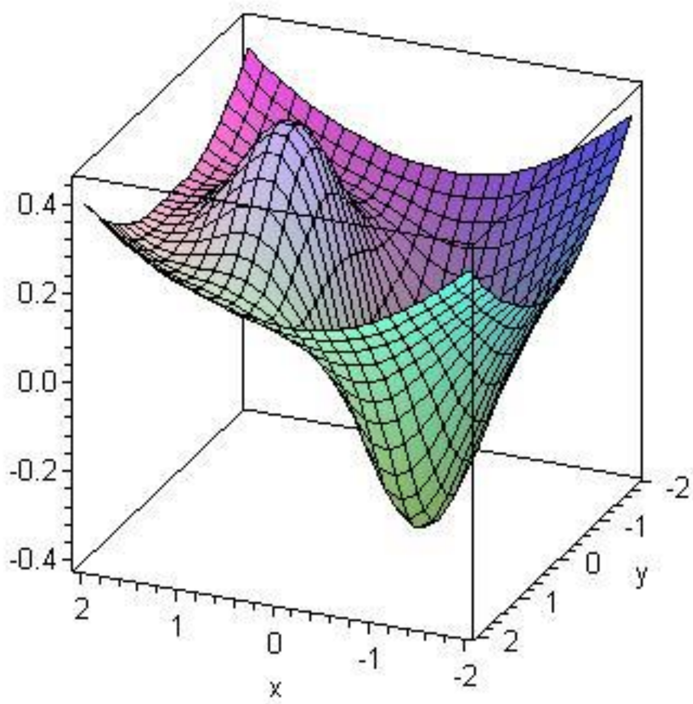
Subject to: $0 < x < 10$

$0 < y < 15$

Minimize: $f(x,y) = Sin(x) + Sin(y)$

Subject to: $-2 \le x, y \le 2$





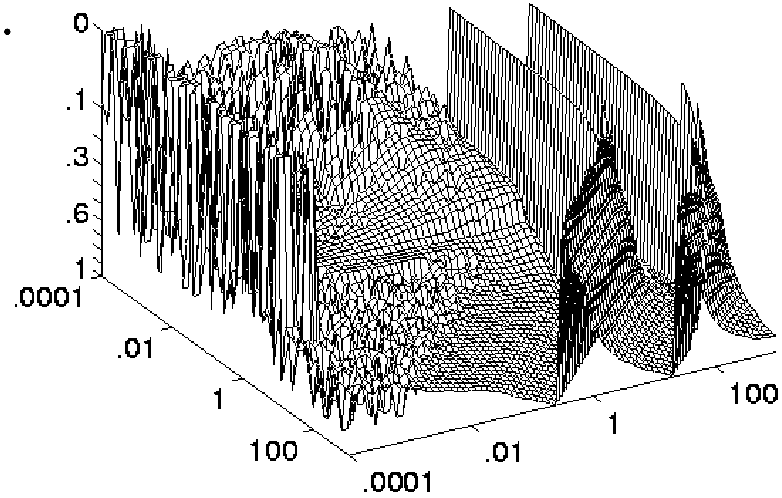
Difficulty of Optimization Problems

One or more of the following aspects can make an optimization problem very difficult to solve:

- The size of the search space in terms of the number of decision variables and the number of constraints (linear, non-linear, equality, inequality, etc.)
- A fitness landscape that is difficult to explore or search.
- The distribution of feasible and infeasible solutions in the search space.
- The number of local and global optima.

Example of roughed fitness landscape for some optimization problem.

The x, y axes are decision variables and the z axis is the objective function.



Combinatorial Optimization Problems

Discrete optimization refers in general to those optimization problems (linear or non-linear) involving discrete decision variables.

In combinatorial optimization problems (COP) the goal is to find the optimal setting of a set of discrete entities such that given requirements and perhaps constraints are satisfied. The optimal setting can be an arrangement, ordering, grouping, selection, distribution, etc.

For some COPs it might not be possible to develop a linear optimization model, so they have to be tackled as non-linear problems. Many COPs are very difficult to solve due to the huge size of their search space, hence finding a near-optimal solution might be sufficient.

The Symmetric TSP (Travelling Salesman Problem)

Given:

A undirected graph G with a set of nodes V and set of edges E

N = |V| is the number of vertices or nodes

Each node represents a city

Edges are undirected

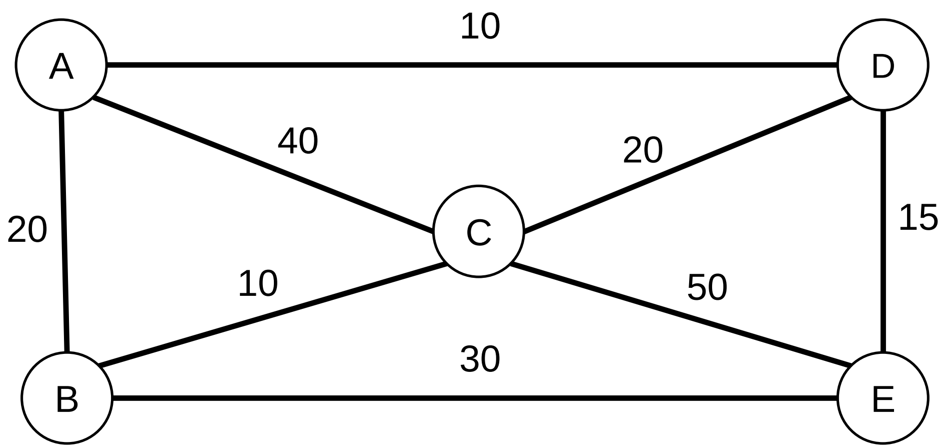
The distance between cities i and j is $D_{ij} = D_{ji}$

The problem is to find a tour to visit all the cities so that the total travelled distance is minimized.

Two sets of constraints:

- Ensure that each node is visited once
- Ensure there are no sub-tours

Example. Solving the Symmetric TSP, find shortest length tour.



$$\text{Minimize}: \quad Z=\frac{1}{2}\left(\begin{array}{l}(20x_{AB}+40x_{AC}+10x_{AD})+(20x_{AB}+10x_{BC}+30x_{BE})+\\(40x_{AC}+10x_{BC}+20x_{CD}+50x_{CE})+(10x_{AD}+20x_{CD}+15x_{DE})+(30x_{BE}+50x_{CE}+15x_{DE})\end{array}\right)$$

$$\begin{array}{lll}\text{Subject to}: & x_{AB}+x_{AC}+x_{AD}=2 & (1)\\ & x_{AB}+x_{BC}+x_{BE}=2 & (2)\\ & x_{AC}+x_{BC}+x_{CD}+x_{CE}=2 & (3)\\ & x_{AD}+x_{CD}+x_{DE}=2 & (4)\\ & x_{BE}+x_{CE}+x_{DE}=2 & (5)\\ & \text{All } x_{ij}\in\{0,1\} & \end{array}$$

Minimum Length: 105

Linear Optimization Model for the Symmetric TSP

$$\text{Minimize:} \quad Z = \frac{1}{2}\sum_{i=1}^{N}\sum_{e \in E_i} D_e X_e$$

$$\text{Subject to:} \quad \sum_{e \in E_i} X_e = 2 \quad \text{for each node/city } i = 1 \ldots N \qquad (1)$$

$$\sum_{e \in (S,S)} X_e \leq |S| - 1 \quad \forall S \in V \qquad (2)$$

$$X_{ij} \in \{0,1\} \quad \text{for each edge between nodes } i \text{ and } j \qquad (3)$$

Adding all the sub-tour elimination constraints in the Symmetric TSP can make the model more difficult because potentially there is an exponential number of constraints required. So, in practice is often desirable to add these constraints as sub-tours arise during the optimization process.

The Multiple Knapsack Problem

Given:

A set of M knapsacks each of given capacity B_j

A set of N items

Each item has a given size S_i

Each item generates profit P_{ij} if it is included in knapsack j.

The problem is to select which items to pack in each of the knapsacks so that the total profit is maximized without exceeding the capacity of any knapsack.

Example. Given a limited budget in each of 5 years, select the subset of investments to make in each year (each investment can only be made in at most one year) that maximizes the overall return.

Example. Sketch of the multiple knapsack problem.

$$\begin{bmatrix} P_{11} & P_{12} & \Lambda & P_{1M} \\ P_{21} & P_{22} & \Lambda & P_{2M} \\ P_{31} & P_{32} & \Lambda & P_{3M} \\ M & M & M & M \\ P_{N1} & P_{N2} & \Lambda & P_{NM} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & \Lambda & X_{1M} \\ X_{21} & X_{22} & \Lambda & X_{2M} \\ X_{31} & X_{32} & \Lambda & X_{3M} \\ M & M & M & M \\ X_{N1} & X_{N2} & \Lambda & X_{NM} \end{bmatrix} = Z$$

Objective function is the sum-product of decision variables and profits.

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ M \\ S_N \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ X_{31} \\ M \\ X_{N1} \end{bmatrix} \leq B_1 \quad \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ M \\ S_N \end{bmatrix} \begin{bmatrix} X_{12} \\ X_{22} \\ X_{32} \\ M \\ X_{N2} \end{bmatrix} \leq B_2 \;\; \Lambda \quad \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ M \\ S_N \end{bmatrix} \begin{bmatrix} X_{1M} \\ X_{2M} \\ X_{3M} \\ M \\ X_{NM} \end{bmatrix} \leq B_M$$

The capacity constraint for each knapsack is the sum-product of the knapsack's decision variables and the item sizes.

$$\begin{aligned} &[X_{11} + X_{12} + X_{13} + \Lambda + X_{1M}] \leq 1 \\ &[X_{21} + X_{22} + X_{23} + \Lambda + X_{2M}] \leq 1 \\ &[X_{31} + X_{32} + X_{33} + \Lambda + X_{3M}] \leq 1 \\ &M \qquad\qquad\qquad\qquad\qquad M \\ &[X_{N1} + X_{N2} + X_{N3} + \Lambda + X_{NM}] \leq 1 \end{aligned}$$

The constraint for each item being packed exactly in one knapsack is the sum of the item's decision variables on all knapsacks.

Linear Optimization Model for the Multiple Knapsack Problem

$$\text{Maximize}: \quad Z = \sum_{i=1}^{N} \sum_{j=1}^{M} P_{ij} X_{ij}$$

$$\text{Subject to}: \quad \sum_{i=1}^{N} S_i X_{ij} \leq B_j \quad \text{for } j = 1,2,\ldots M \qquad (1)$$

$$\sum_{j=1}^{M} X_{ij} \leq 1 \qquad \text{for } i = 1,2,\ldots N \qquad (2)$$

$$X_{ij} = 1 \text{ if item } i \text{ is packed into knapsack } j, \text{ 0 otherwise} \qquad (3)$$

BIP model for problem instance with $N = 3$ and $M = 2$

$$\text{Maximize}: \quad Z = P_{11}X_{11} + P_{12}X_{12} + P_{21}X_{21} + P_{22}X_{22} + P_{31}X_{31} + P_{32}X_{32}$$

$$\text{Subject to}: \quad S_1 X_{11} + S_2 X_{21} + S_3 X_{31} \leq B_1 \qquad (1)$$

$$S_1 X_{12} + S_2 X_{22} + S_3 X_{33} \leq B_2 \qquad (2)$$

$$X_{11} + X_{12} + X_{13} \leq 1 \qquad (3)$$

$$X_{21} + X_{22} + X_{23} \leq 1 \qquad (4)$$

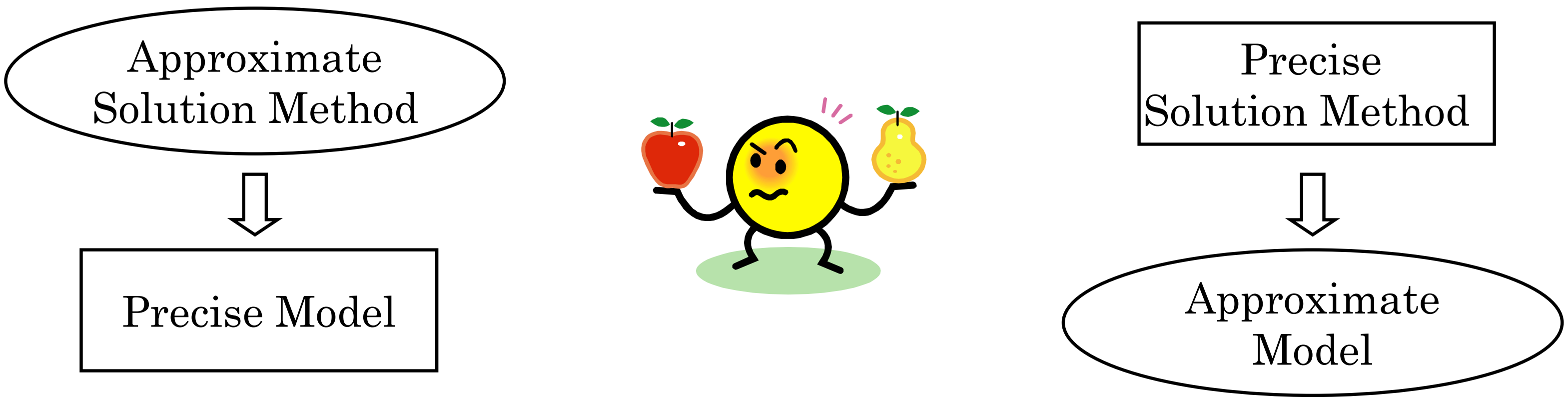
$$X_{31} + X_{32} + X_{33} \leq 1 \qquad (5)$$

$$X_{ij} \in \{0,1\} \quad \text{for } i = 1\ldots 8 \quad \text{and} \quad j = 1\ldots 2 \qquad (6)$$

Problem vs. Model in Optimization

A solution is only a solution in terms of the model used to represent the problem.

Most times, assumptions have to be made to simplify the complexity of real-world problems.



Hard constraints must be satisfied for a solution to be feasible.

Soft constraints are not mandatory but desirable conditions in good quality solutions.

Basics of Heuristic Optimization

A heuristic search method is a technique that aims to find good-quality (i.e. near-optimal) solutions at a reasonable computation time but that is not able to guarantee either feasibility or optimality.

Heuristic methods have been successfully applied to many problems in different areas including: engineering, management, finance, planning and scheduling, medicine, biology, automated navigation, image processing, robotics, art design, etc.

There is a range of heuristic methods including: constructive heuristics, local search, meta-heuristics, evolutionary algorithms, hyper-heuristics, hybrid heuristics, etc.

Motivation for Heuristic Optimization

Many real-world optimization problems (continuous, discrete or combinatorial) are difficult to solve for several reasons:

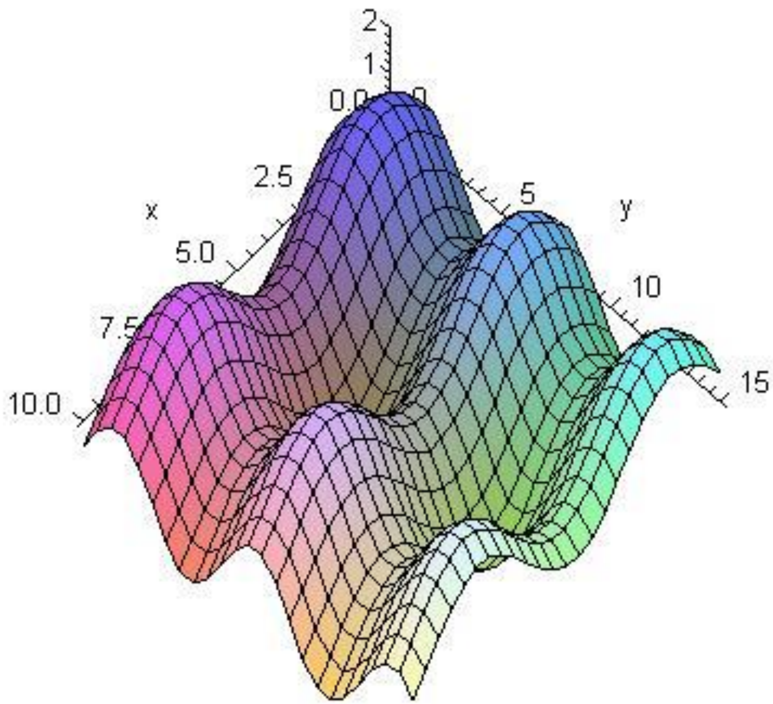
- Even if a formal optimization model exists, the search space is too large and difficult to explore.
- Difficult to develop a formal optimization model:
  - Difficult to formulate the many types of real-world hard and soft constraints.
  - Difficult to design an accurate objective function.
  - Therefore, ad-hoc computer models are used to represent the problem and solutions.
- No guarantee of optimality, finding near-optimal solutions in practical computation time is sufficient.
- Heuristic methods can be tailored and tuned for the specific problem and ad-hoc model in hand.

Size of the Search Space | S |

In both continuous and combinatorial problems, | S | depends on the number of variables and their number of permissible values.

$$f(x,y) = x \cdot exp^{\left(-(x^2+y^2)+\frac{x^2+y^2}{20}\right)}$$

$$\text{s.t.} \quad 0 < x < 10 \text{ and } 0 < y < 15$$

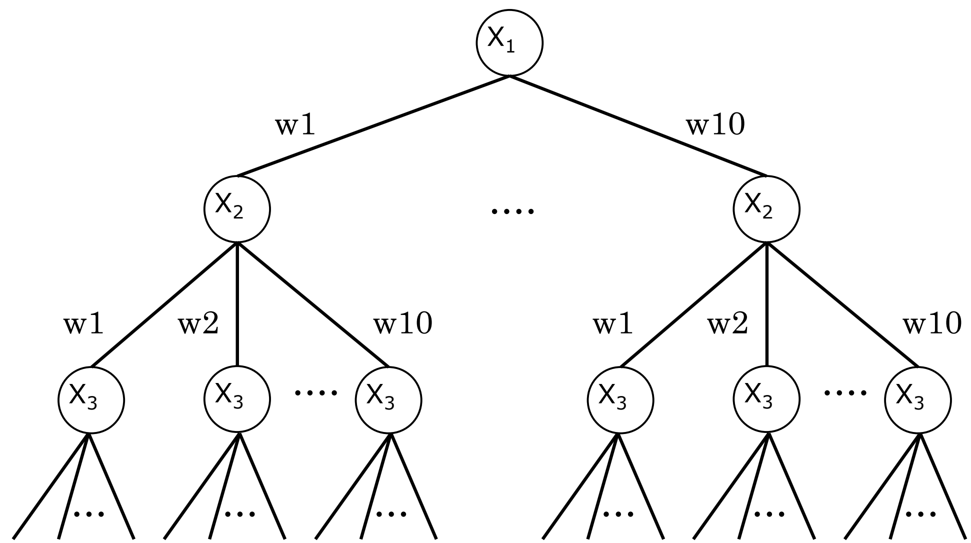


$$\text{Minimise} \quad Z = \sum_{j=1}^{10} \sum_{i=1}^{50} c_{ij} x_{ij}$$

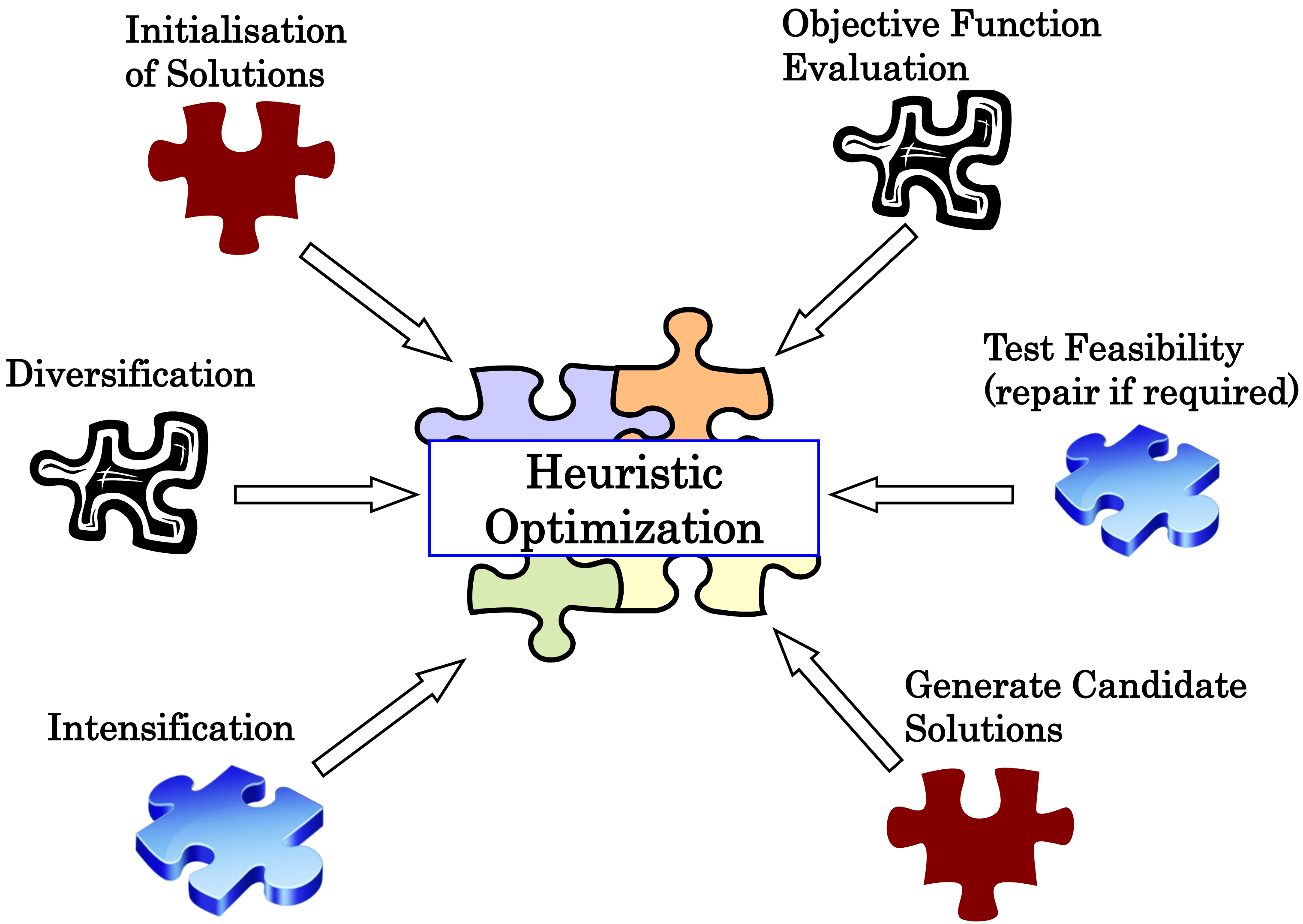
$$\text{subject to} \quad \sum_{j=1}^{10} x_{ij} = 1 \qquad \text{for } i = 1 \ldots 50 \quad (1)$$

$$\sum_{i=1}^{50} t_{ij} x_{ij} \le T_j \qquad \text{for } j = 1 \ldots 10 \quad (2)$$

$$x_{ij} = 1 \text{ if task } i \text{ assigned to worker } j, 0 \text{ otherwise}$$



Key Strategies Needed in Heuristic Optimization



Some Types of Heuristic Methods

Constructive Heuristics
Work on partial solutions constructing a complete solution step by step adding one element in each step.

Greedy Constructive Heuristics
Construct a solution step by step and making the best possible move in each step based in some criteria.

Local Search Heuristics
Work on complete solutions and explore alternative solutions by making moves to generate solutions in the 'neighbourhood' of the current one.

Evolutionary Heuristics
Mimic evolutionary processes in nature in order to explore alternative solutions usually by 'evolving' a population of solutions.

Basic Implementation Decisions for Heuristics

Solution Representation

- Different representations (encodings) for the same problem.
- Defines the size of the search space.
- An appropriate encoding (creativity?) is crucially important.

Objective

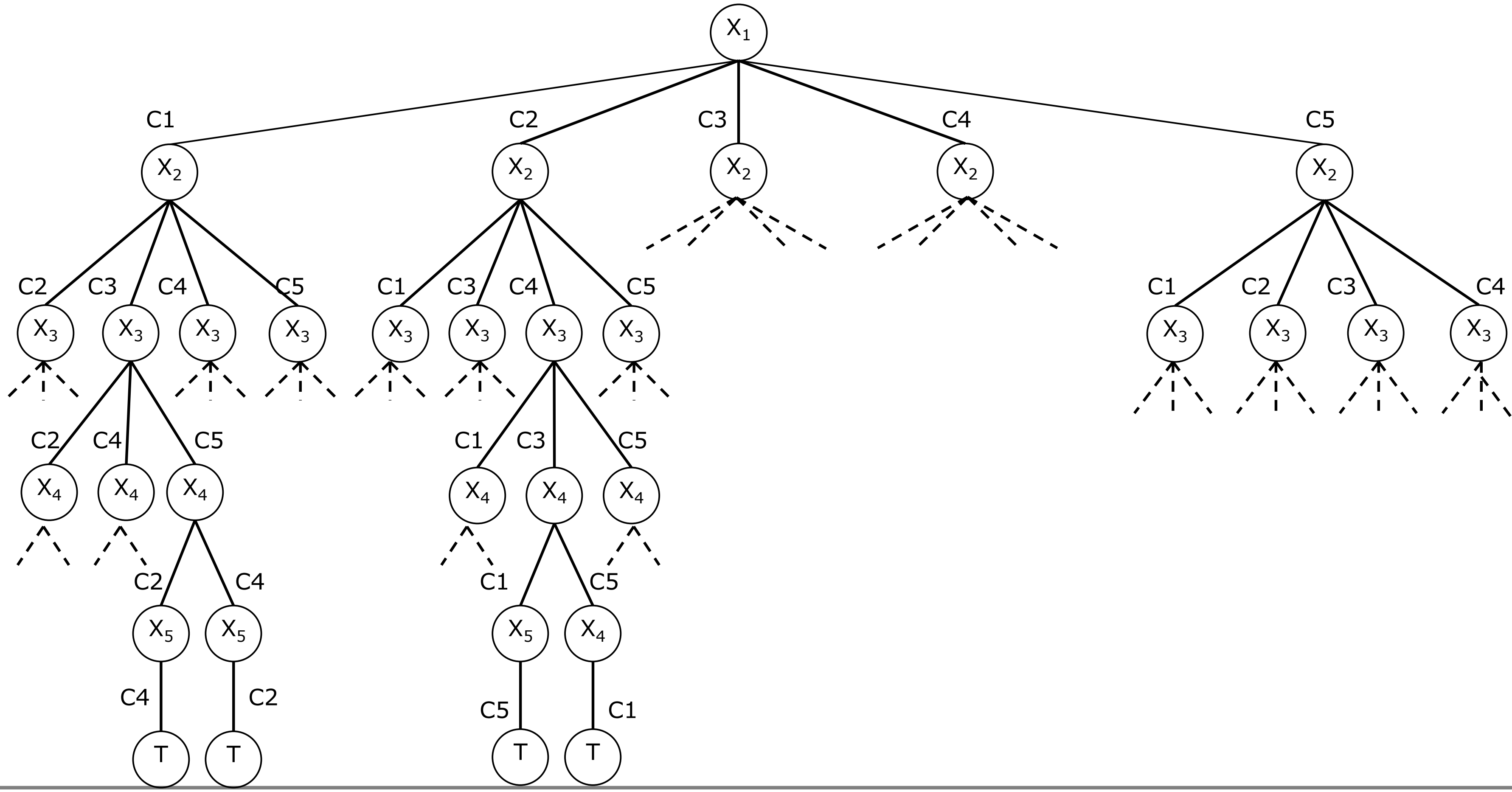
- Minimize or maximize or find one solution?

Evaluation Function (fitness function)

- Maps the representation of a solution to a numeric value that indicates the quality of the solution.
- Exact or approximate evaluation?
- Differentiate between feasible and infeasible solutions.

Example. Sketch of the search space for a 5-city TSP instance.

- There are 5 decision variables x_1, x_2, x_3, x_4, x_5
- A solution (tour) is a permutation of length equal to 5
- There are 5! = 120 different permutations (not all different tours)



Example. Consider a symmetric n-city TSP over a complete graph.

Representation

permutation of integers 1 to n e.g. 13546278

total number of permutations : $n!$

ignoring symmetric tours : $\frac{n!}{2}$ e.g. $13546278 = 87264531$

ignoring shifted identical tours : $\frac{(n-1)!}{2}$ e.g. $13546278 = 54627813$

Objective Function

S is set of solutions and a solution $s = c_1 c_2 c_3 \Lambda\ c_n$ then

$$f(s) = d(c_1, c_2) + d(c_2, c_3) + \Lambda\ d(c_{n-1}, c_n) + d(c_n, c_1)$$

Optimization Problem

find $s \in S$ such that $f(s) \leq f(s') \quad \forall s' \in S$

i.e. find a tour with the minimum length

Neighbour Solutions

2-edge-exchange move : interchanges 2 non-adjacent edges

current : 13546278 neighbours : $13\bar{2}46\bar{5}78$, $13\bar{7}462\bar{5}8$, $135\bar{8}627\bar{4}$, etc.

2-right-insert move : moves a city 2 positions to the right

current : 13546278 neighbours : $154\bar{3}6278$, $13542\bar{7}68$, etc.

Example. These are two different greedy constructive heuristics for the symmetric TSP.

Greedy Heuristic 1

1. Select random starting city
2. Proceed to nearest unvisited city
3. Repeat step 2 until all cities are visited
4. Return to starting city

Greedy Heuristic 2

1. Find shortest edge (c_i, c_j) and add cities c_i, c_j to the tour
2. Select next chapest edge (c_r, c_t)
   (making sure no city is visited more than once)
3. Add cities c_r, c_t to the tour
4. Repeat steps 2 - 3 until a tour is formed

Neighbourhood Search

In COPs, the neighbourhood of a solution x in the search space S can be defined as:

$$N(x) = \{ y \in S \mid y = \Phi(x) \}$$

where Φ is a move or sequence of moves

A feasible solution x is a local optimum with respect to N(x) if:

$$f(x) \le f(y) \quad \forall y \in N(x) \quad \text{(assuming minimization)}$$

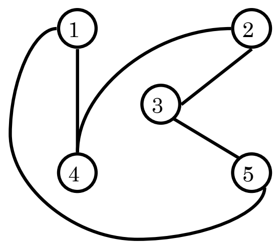
If the inequality above is strict then x is a unique local optimum, otherwise x is one of the more than one local optima.

A plateau is a region of the search space in which there are more than one local optimum, i.e. a plateau is a region where no changes in the objective function occur between neighbour solutions.

Example. Considering the 2-edge-exchange neighbourhood structure, illustrate neighbour solutions for a symmetric TSP with 5 cities. Start with the following tour: 14235

edges
14
42
23
35
51

tour: 14235

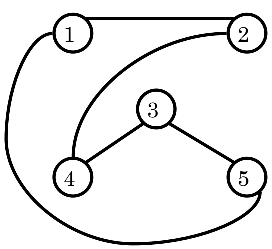


possible exchanges
14,23 to 12,43
14,35 to 13,45
42,35 to 43,25
42,51 to 45,21
23,51 to 25,31

Each of the possible 2-edge-exchanges generates a neighbour solution

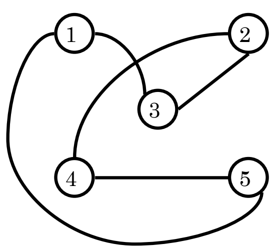
14,23 to 12,43

tour: 12435



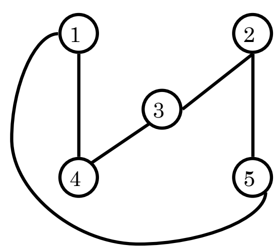
14,35 to 13,45

tour: 13245



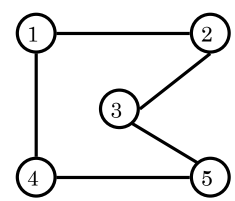
42,35 to 43,25

tour: 14325



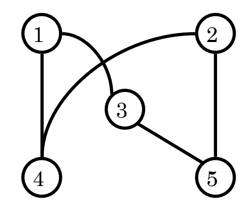
42,51 to 45,21

tour: 12354



23,51 to 25,31

tour: 13524



Based on the 2-edge-exchange move, each tour has 5 neighbour solutions, i.e. $|N(x)| = 5$