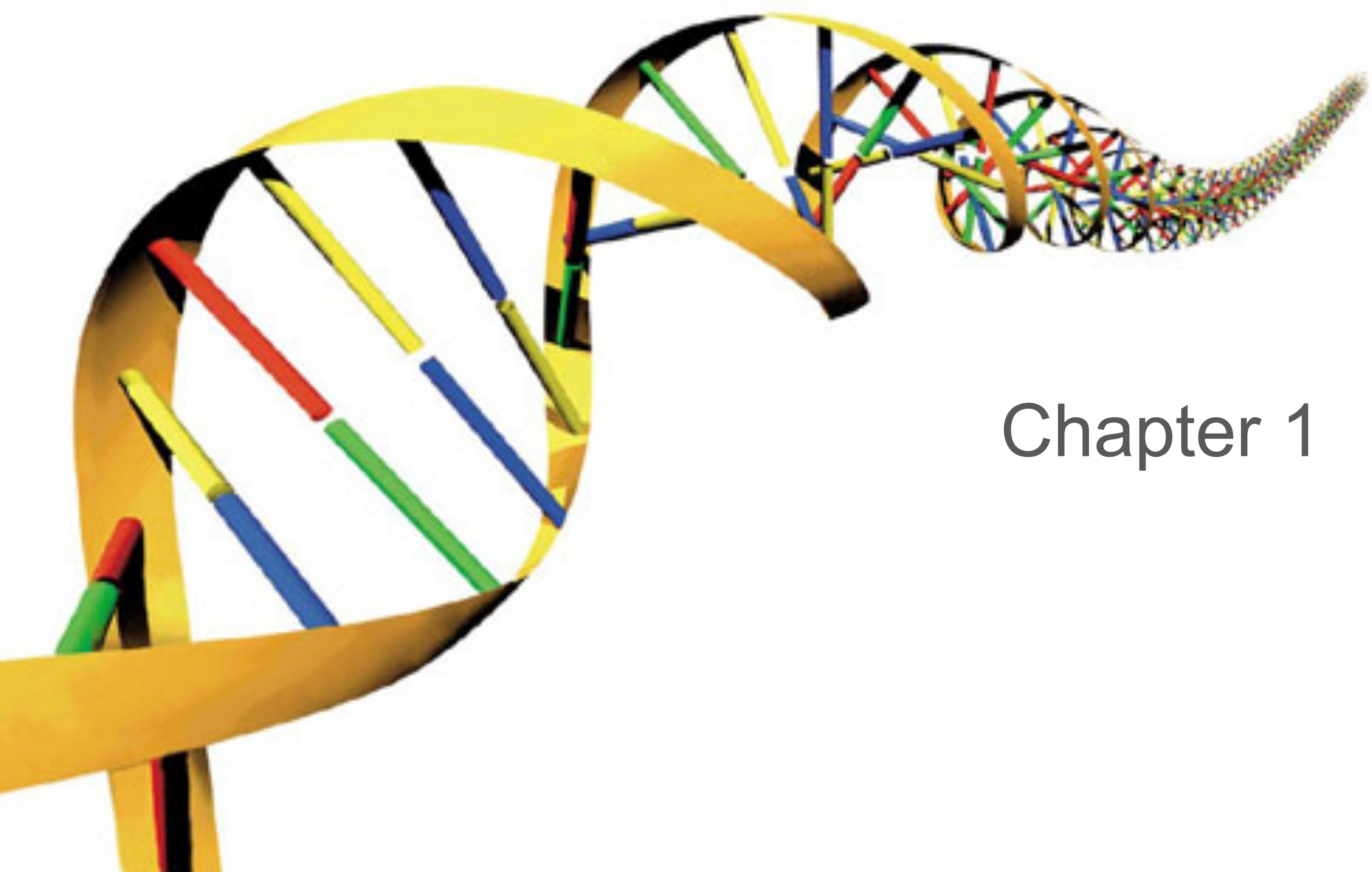
Evolutionary Computing



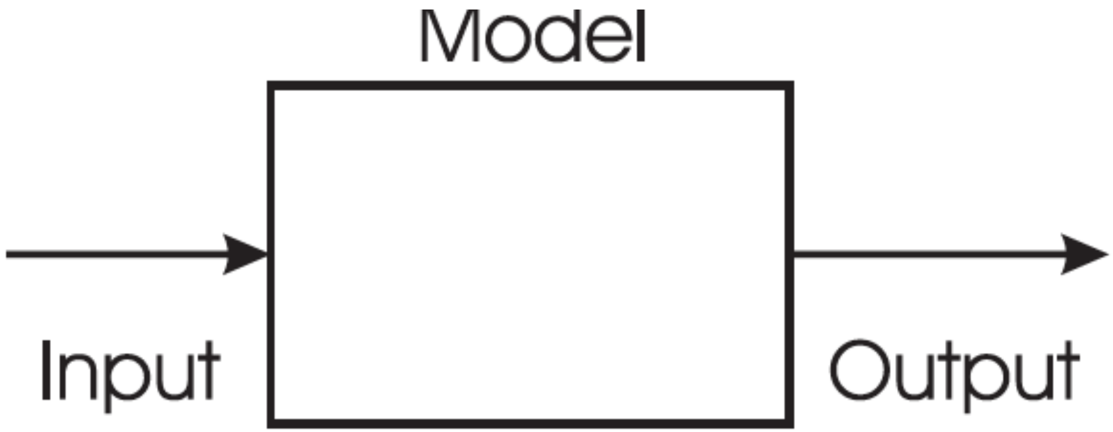
Chapter 1

Chapter 1: Problems to be solved

Problems can be classified in different ways:

- Black box model
- Search problems
- Optimization and/or constraint satisfaction
- NP problems

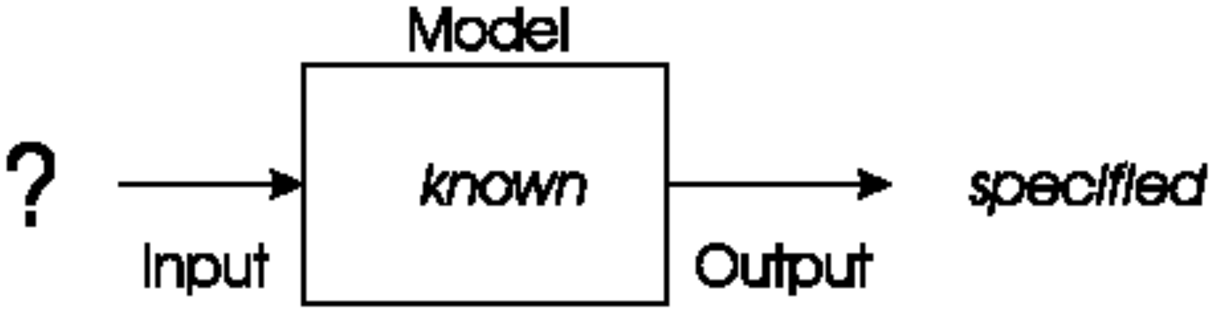
“Black box” model



- “Black box” model consists of 3 components
- When one component is unknown: new problem type

“Black box” model: Optimization

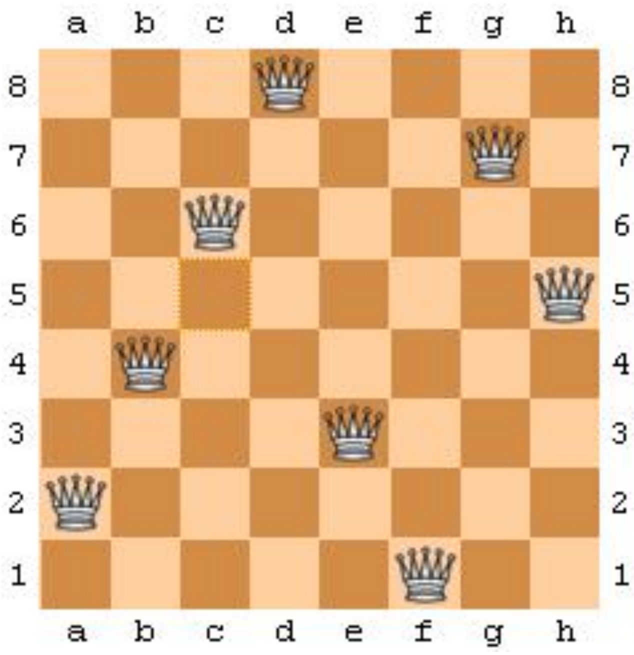
- Model and desired output is known, task is to find inputs



- Examples:
  - Time tables for university, call center, or hospital
  - Traveling salesman problem (TSP)
  - Eight-queens problem.

“Black box” model: Optimization example: 8 queens problem

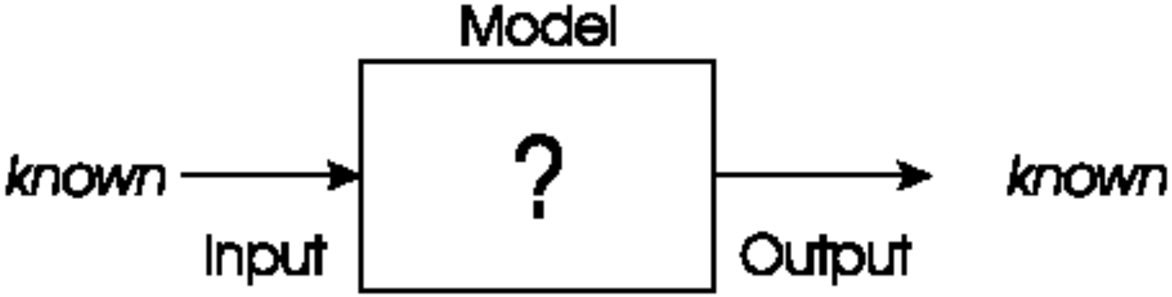
- Given an 8-by-8 chessboard and 8 queens
- Place the 8 queens on the chessboard without any conflict
- Two queens conflict if they share same row, column or diagonal
- Can be extended to an n queens problem (n>8)



- Input? Model? Output?

“Black box” model: Modelling

- We have corresponding sets of inputs & outputs and seek model that delivers correct output for every known input



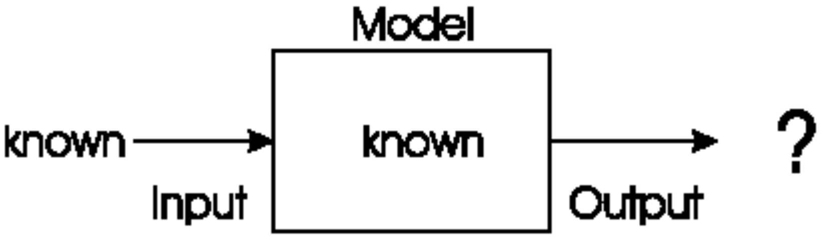
- Examples:
  - Evolutionary machine learning
  - Stock exchange
  - Voice control system for smart homes

“Black box” model:
Modelling example: load applicant creditibility

- British bank evolved creditability model to predict loan paying behavior of new applicants
- Evolving: prediction models
- Fitness: model accuracy on historical data

"Black box" model: Simulation

- We have a given model and wish to know the outputs that arise under different input conditions



- Often used to answer "what-if" questions in evolving dynamic environments
  - Evolutionary economics, Artificial Life
  - **Weather forecast system**
  - Impact analysis new tax systems

Search problems

- Simulation is different from optimization or modelling
- **Optimization or modelling problems search through huge space of possibilities**
- **Search space: collection of all objects of interest including the desired solution**
- Question: how large is the search space for different tours through *n* cities?
- The specification of the search space is the **first step** in defining a search problem
- The **second step** is the definition of a solution:
  - Explicit: e.g., a board configuration where the number of checked queens is zero
  - Implicit, e.g., a tour that is the shortest of all tours.

Optimization and constraint satisfaction (1/2)

- Objective function: a way of assigning a value to a possible solution that reflects its quality on scale
  - Number of un-checked queens (maximize)
  - Length of a tour visiting given set of cities (minimize)
- Constraint: binary evaluation telling whether a given requirement holds or not
  - Find a configuration of eight queens on a chessboard such that no two queens check each other
  - Find a tour with minimal length where city X is visited after city Y

Optimization vs. constraint satisfaction (2/2)

- When combining the two:

	Objective function	
Constraints	**Yes**	**No**
Yes	**Constrained optimization problem**	Constraint satisfaction problem
No	**Free optimization problem**	No problem

NP problems

- We only looked at classifying the problem, not discussed problem solvers
- This classification scheme needs the properties of the problem solver
- Benefit of this scheme: possible to tell how difficult the problem is
- Explain the basics of this classifier for combinatorial optimization problems (booleans or integers search space)

NP problems:
Key notions

- Problem size: dimensionality of the problem at hand and number of different values for the problem variables
- Running-time: number of operations the algorithm takes to terminate
  - Worst-case as a function of problem size
  - Polynomial, exponential
- Problem reduction: transforming current problem into another via mapping

NP-Complete problems:

- All problems can be solved in polynomial time?
  - **No**
  - Those problems are NP
  - Examples:
    - Hanoi tower: O(2n)
    - Traveling Salesman Problem: O(n!)

Possible solution: use of approximation algorithms and metaheuristics.

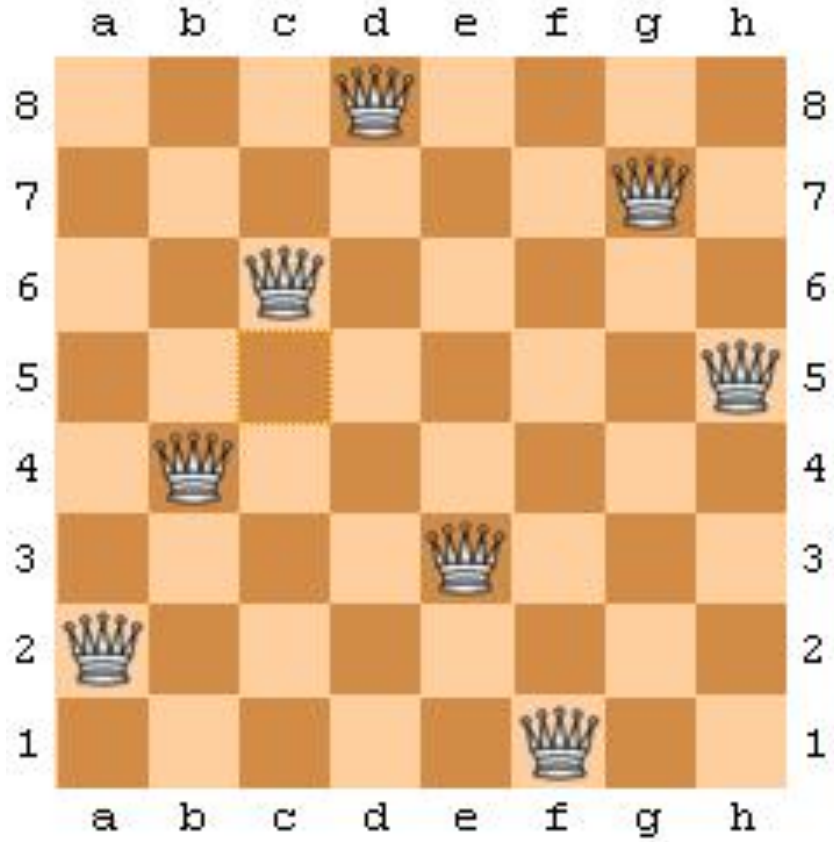
Atividade 1

- Pesquise sobre a classificação de problemas computacionais:
  - P
  - NP
  - NP-Completo
  - NP-Hard
- Para cada classe, mostre um exemplo de problema que se encaixa explicando o motivo.
- Cada dupla irá apresentar o problemas escolhidos (máximo 15 min por dupla).

- *Entregar relatório via SIGAA.*
- *Pode ser feito em dupla ou individualmente.*
- *Data: 16/11/2020*

Atividade 2

- 8 - Rainhas



Atividade 3

- Escolha um problema computacional de otimização, modelagem ou simulação.
  - Dê preferência à sua área de pesquisa/tema de dissertação.
- Apresente o problema (máximo 10 min).
- Data: 23/11/2020