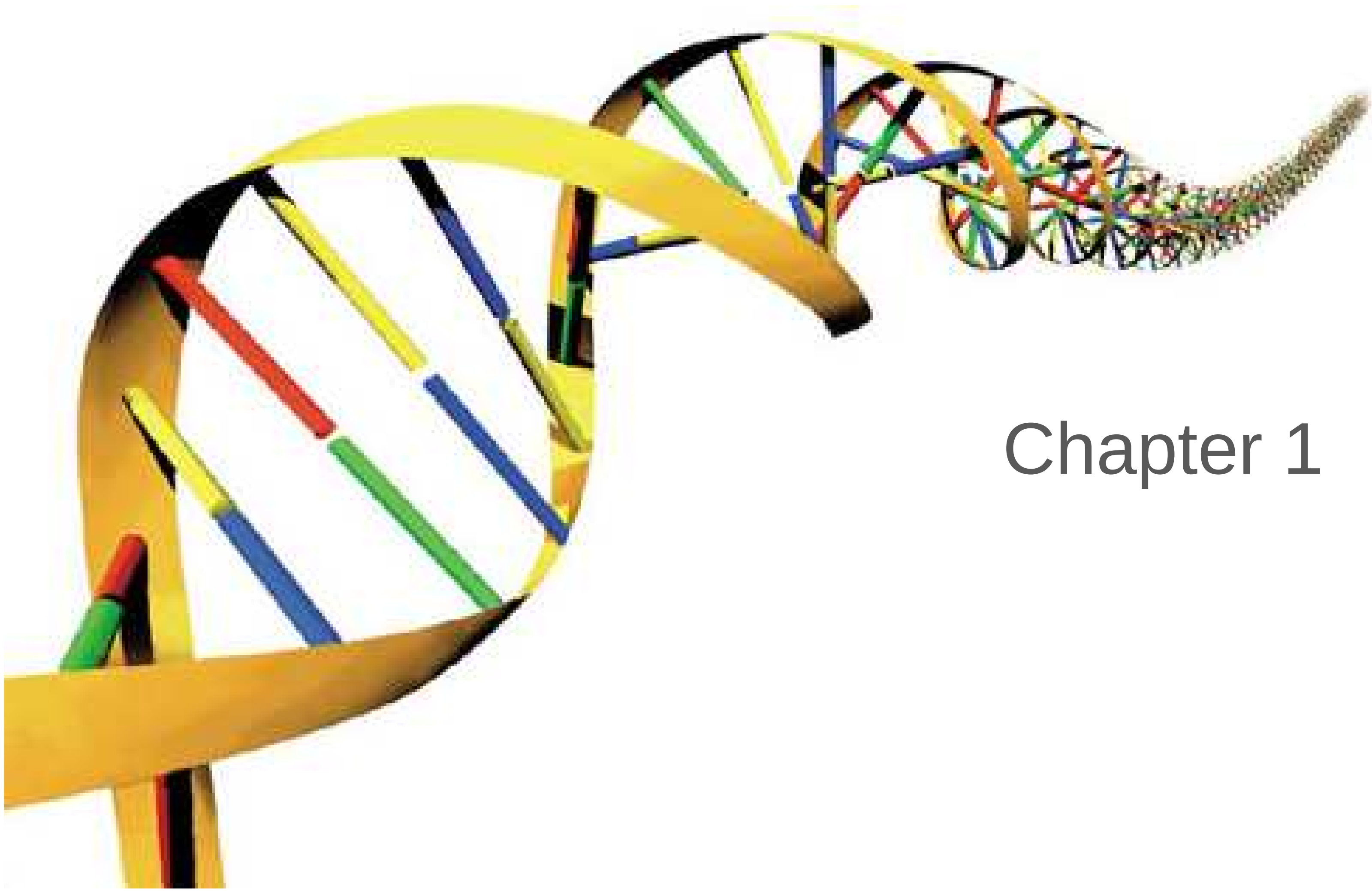
Evolutionary Computing



Chapter 1

Chapter 1: Problems to be solved

Problems can be classified in different ways:

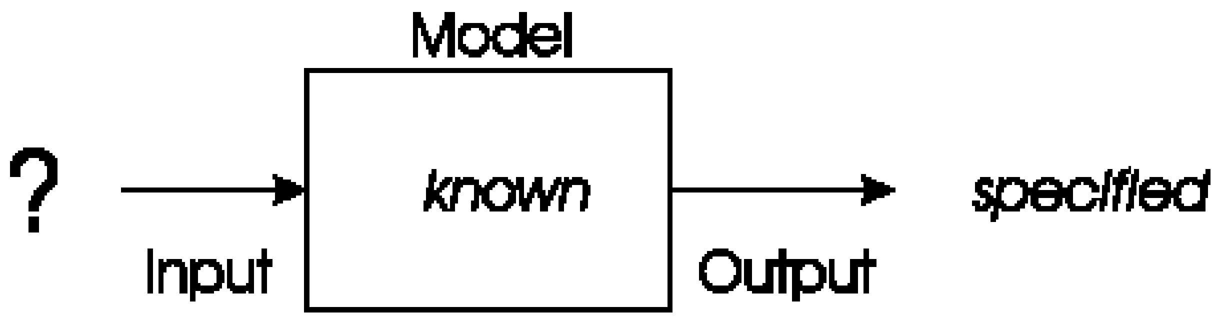
- Black box model
- Search problems
- Optimisation vs constraint satisfaction
- NP problems

“Black box” model

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

"Black box" model: Optimisation

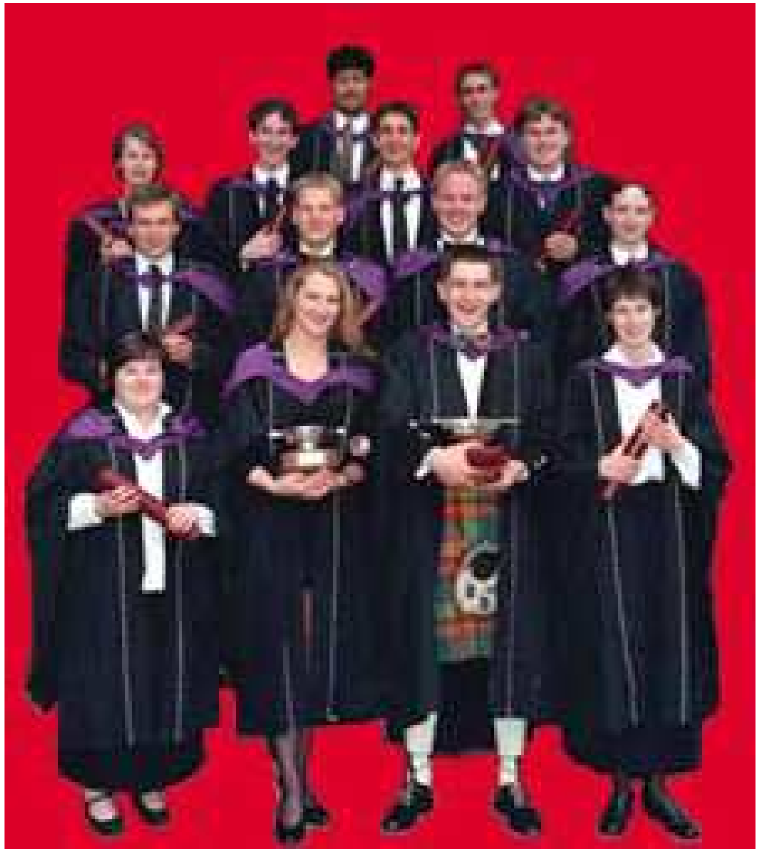
- Model and desired output is known, task is to find inputs

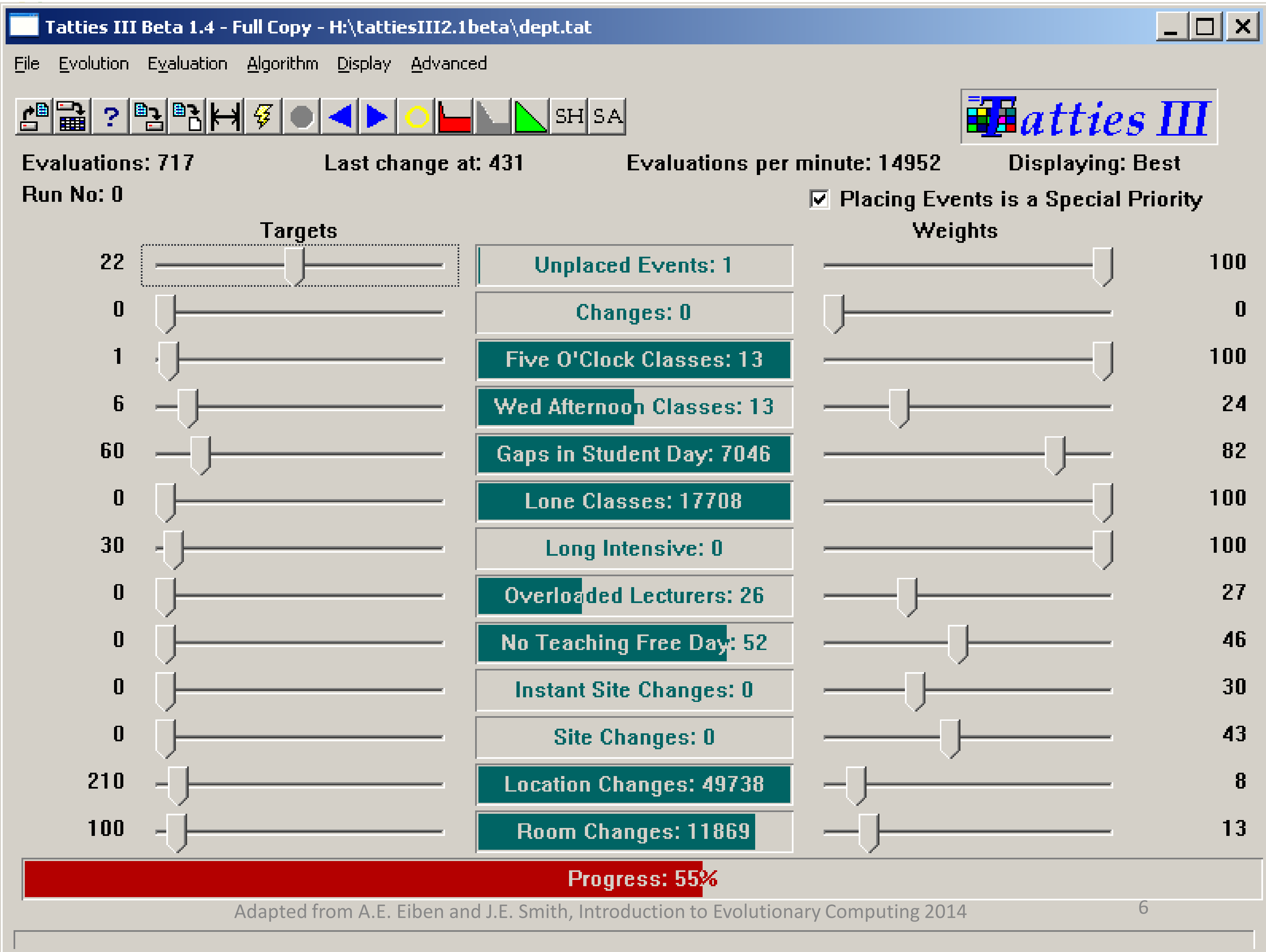


- Examples:
  - Time tables for university, call center, or hospital
  - Design specifications
  - Traveling salesman problem (TSP)
  - Eight-queens problem, etc.

"Black box" model: Optimisation example 1: university timetabling

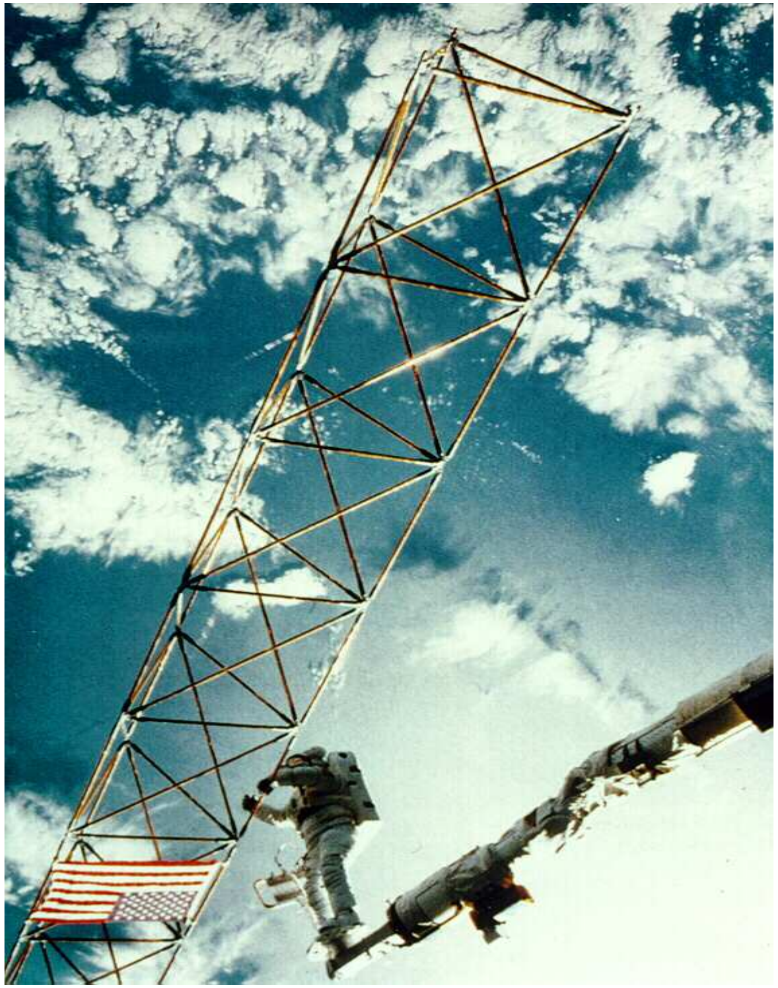
- Enormously big search space
- Timetables must be *good*
- "Good" is defined by a number of competing criteria
- Timetables must be feasible
- Vast majority of search space is infeasible




Tatties III Beta 1.4 - Full Copy - H:\tattiesIII2.1beta\dept.tat
File Evolution Evaluation Algorithm Display Advanced
SH SA
Tatties III
Evaluations: 717
Last change at: 431
Evaluations per minute: 14952
Displaying: Best
Run No: 0
Placing Events is a Special Priority
Targets
Weights
22 Unplaced Events: 1 100
0 Changes: 0 0
1 Five O'Clock Classes: 13 100
6 Wed Afternoon Classes: 13 24
60 Gaps in Student Day: 7046 82
0 Lone Classes: 17708 100
30 Long Intensive: 0 100
0 Overloaded Lecturers: 26 27
0 No Teaching Free Day: 52 46
0 Instant Site Changes: 0 30
0 Site Changes: 0 43
210 Location Changes: 49738 8
100 Room Changes: 11869 13
Progress: 55%

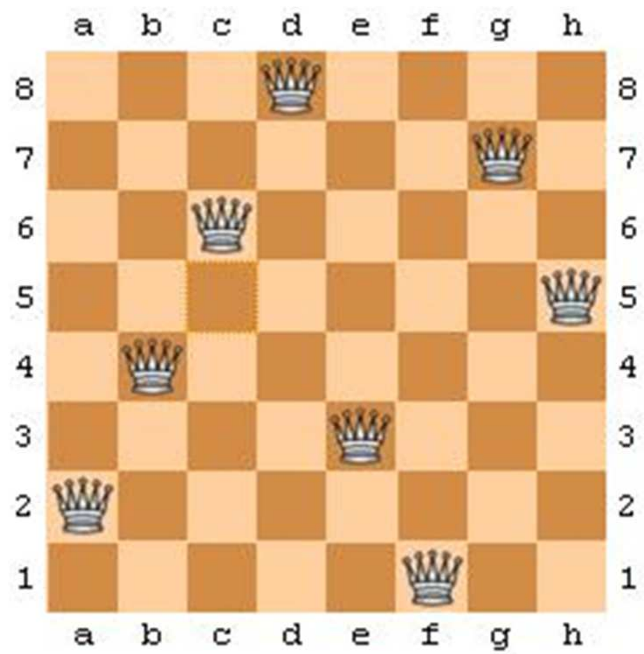
"Black box" model:
Optimisation example 2: satellite structure

- Optimised satellite designs for NASA to maximize vibration isolation
- Evolving: design structures
- Fitness: vibration resistance
- Evolutionary "creativity"



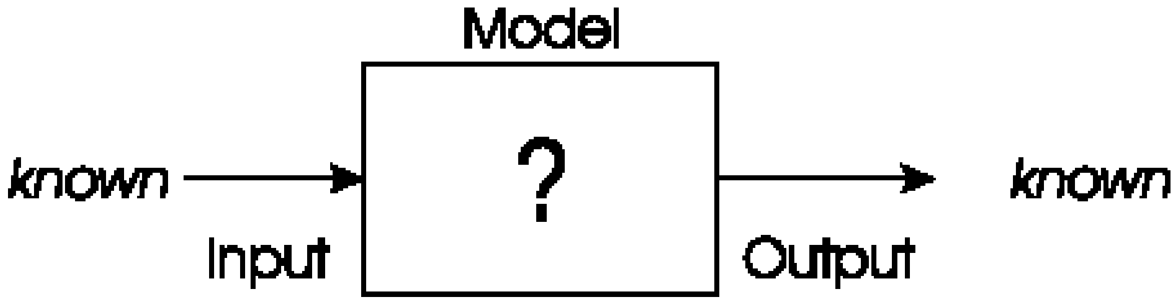
“Black box” model:
Optimisation example 3: 8 queens problem

- 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)



"Black box" model: Modelling

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



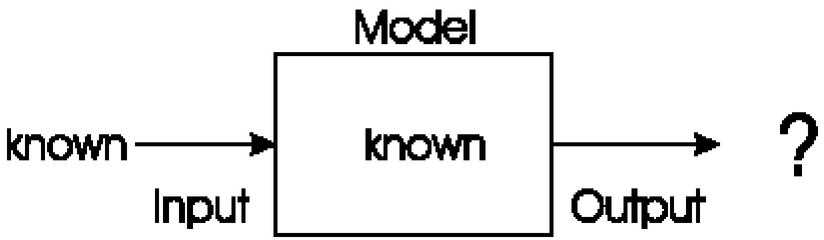
- Note: modelling problems can be transformed into optimisation problems
  - Evolutionary machine learning
  - Predicting stock exchange
  - Voice control system for smart homes

“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

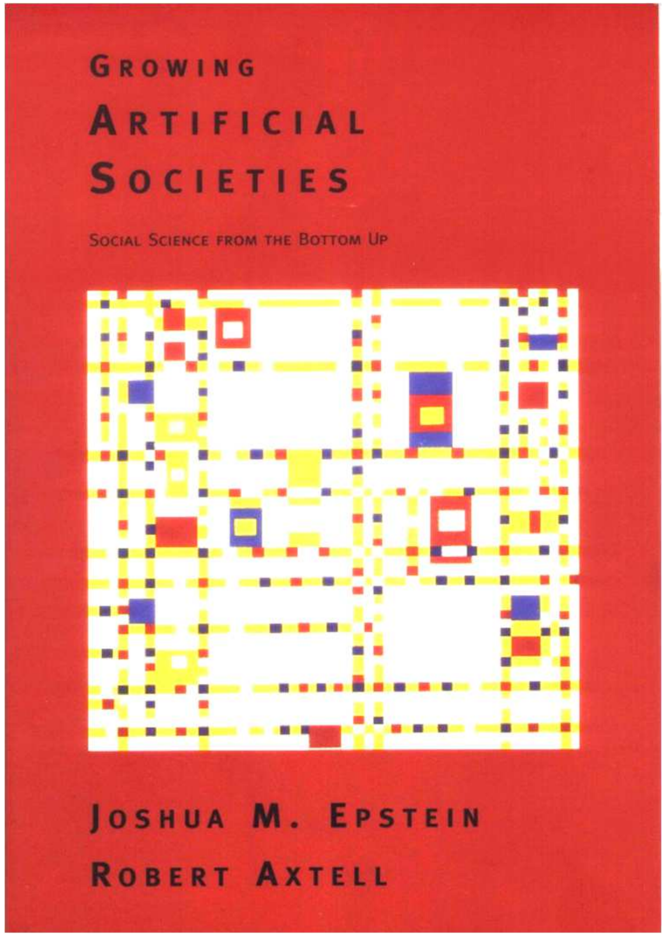
“Black box” model: Simulation

- 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

"Black box" model: Simulation example: evolving artificial societies



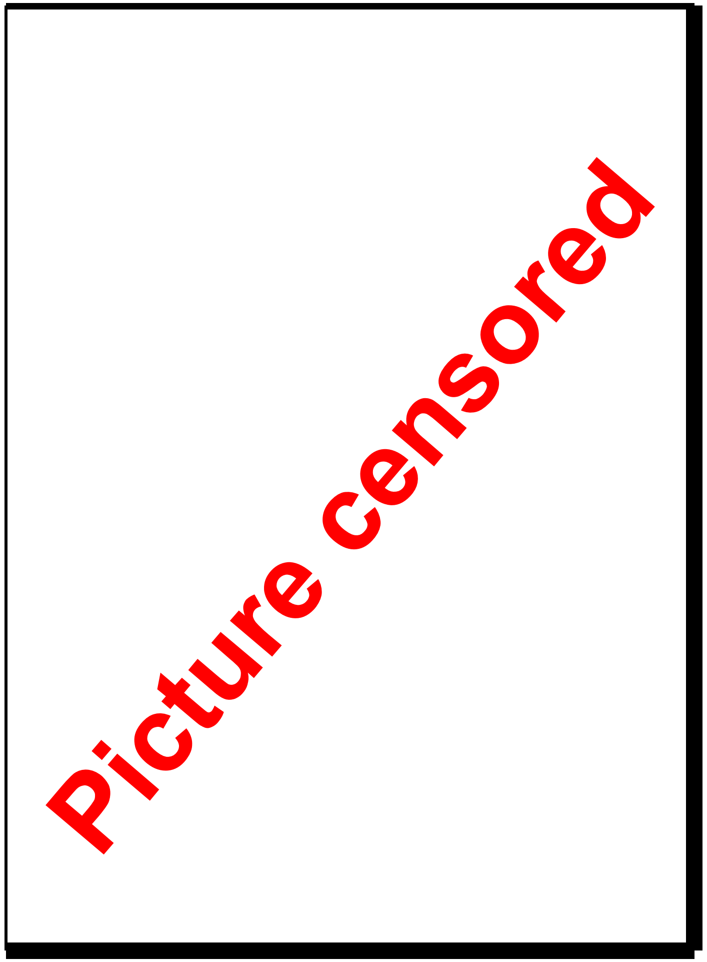
Simulating trade, economic competition, etc. to calibrate models

Use models to optimise strategies and policies

Evolutionary economy

Survival of the fittest is universal (big/small fish)

"Black box" model:
Simulation example 2: biological interpretations



Incest prevention keeps evolution from rapid degeneration

(we knew this)

Multi-parent reproduction, makes evolution more efficient

(this does not exist on Earth in carbon)

2nd sample of Life

Search problems

- Simulation is different from optimisation/modelling
- Optimisation/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?

Benefit of classifying these problems: distinction between
- search problems, which define search spaces, and
- problem-solvers, which tell how to move through search spaces.

Optimisation vs. 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

Optimisation vs. constraint satisfaction (2/2)

- When combining the two:

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

- Where do the examples fit?
- Note: constraint problems can be transformed into optimisation problems
- Question: how can we formulate the 8-queens problem in to a FOP/CSP/COP?

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 optimisation problems (booleans or integers search space)

NP problems: Key notions

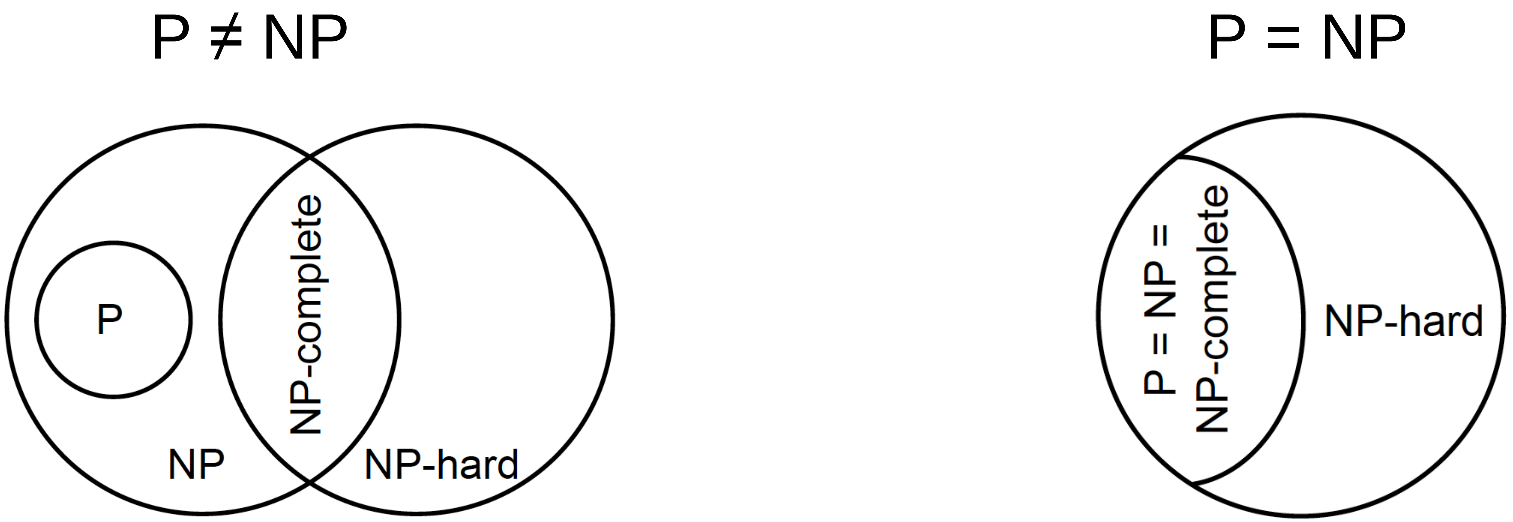
- Problem size: dimensionality of the problem at hand and number of different values for the problem variables
- Running-time: number of operations the algorithm takes to terminate
  - Worst-case as a function of problem size
  - Polynomial, super-polynomial, exponential
- Problem reduction: transforming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

NP problems: Class

- The difficultness of a problem can now be classified:
  - Class P: algorithm can solve the problem in polynomial time (worst-case running-time for problem size n is less than F(n) for some polynomial formula F)
  - Class NP: problem can be solved and any solution can be verified within polynomial time by some other algorithm (P subset of NP)
  - Class NP-complete: problem belongs to class NP and any other problem in NP can be reduced to this problem by al algorithm running in polynomial time
  - Class NP-hard: problem is at least as hard as any other problem in NP-complete but solution cannot necessarily be verified within polynomial time

NP problems: Difference between classes

- P is different from NP-hard
- Not known whether P is different from NP



- For now: use of approximation algorithms and metaheuristics