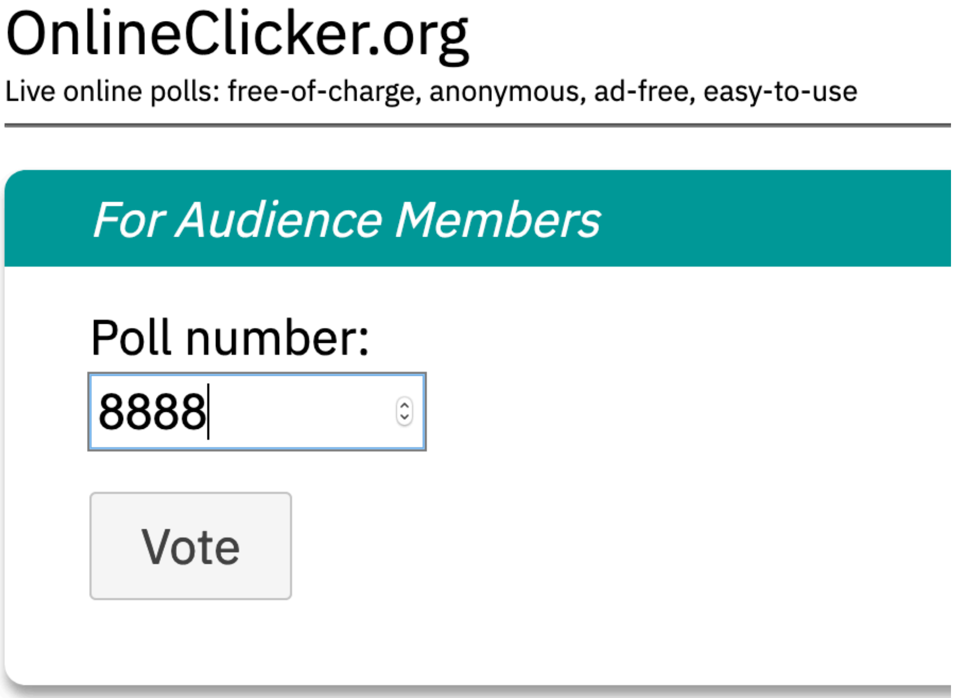
Evolutionary Computation

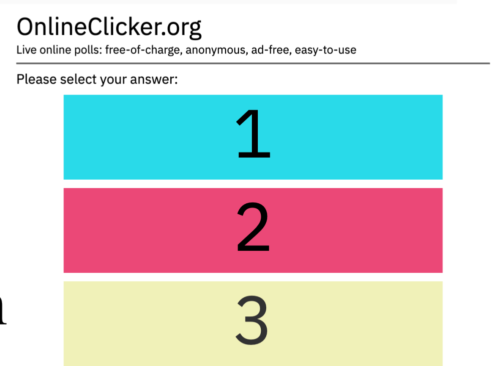
December 2020

Ice Breaker

- Go to



➔ Vote for a superpower:
1. flight
2. invisibility
3. evolutionary computation



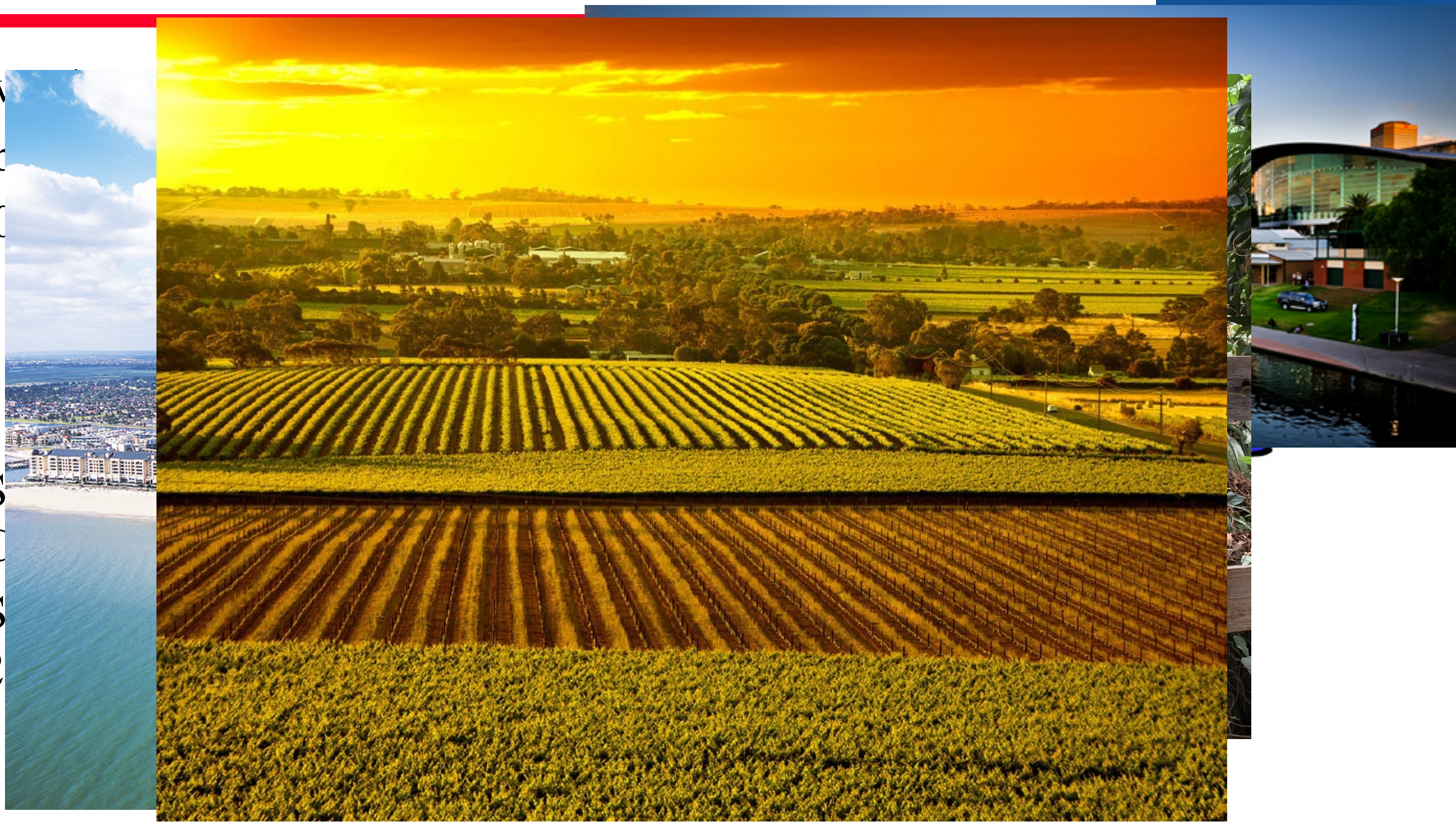
Evolutionary Computation

But first a few words on the lecturer
Dr Markus Wagner and on his group…

Univ

Harb

910(



- S
  C
- S
  2

Optimisation and Logistics Group
University of Adelaide, Australia

Prof. Frank Neumann

E/Prof. Zbigniew Michalewicz

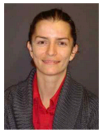
SolveIT Software Excellence Award (STEM Professional)] COMPLEXICA

10-15 PhD students.

http://cs.adelaide.edu.au/~optlog

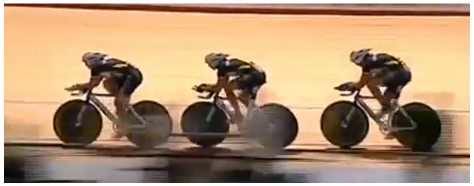
Our research agenda:
- Develop algorithms for problems of high significance
- Build up a theory that explains how heuristic methods work

Academic Staff

	Dr. Bradley Alexander Senior Lecturer	4.43	34484	brad@cs.adelaide.edu.au Search Based Geoscience Modelling, Genetic Programming for Agent Control, Search Based Software Engineering, Optimisation of Pipe Networks, Computer Science Education
	Dr Wanru (Kelly) Gao Lecturer	4.53	31868	wanru.gao@adelaide.edu.au
	Dr. Mingyu Guo Lecturer	4.19		Computational Microeconomics / Algorithmic Game Theory, Multiagent Systems, Electronic Commerce, Mechanism Design, Prediction Markets.
	Prof. Frank Neumann Group Leader	4.55	34477	frank.neumann@adelaide.edu.au Bio-Inspired Computing, Evolutionary Algorithms, Combinatorial Optimisation, Multi-Objective Optimisation, Renewable Energy, Optimisation Problems in Sports
	Dr Sergey Polyakovskiy Lecturer	4.56	34486	sergey.polyakovskiy@adelaide.edu.au Combinatorial Optimisation, Mathematical & Constrained Programming, Hybrid Optimization Approaches, Industrial Applications
	Dr. Claudia Szabo Senior Lecturer	4.17	36744	claudia.szabo@adelaide.edu.au Model-Driven Engineering, Cloud Computing, Software Engineering, Simulation
	Dr. Markus Wagner Senior Lecturer DECRA Fellow	4.45	35405	markus.wagner@adelaide.edu.au Bio-inspired algorithms, multi-objective optimisation, computational complexity analysis, real-world problems
	Dr. Amali Weerasinghe Lecturer	4.48	34626	amali.weerasinghe@adelaide.edu.au Artificial intelligence, education
	Dr. Maria-Yaneli Ameca-Alducin Postdoctoral Researcher	4.52	36196	maria-yaneli.ameca-alducin@adelaide.edu.au
	Dr. Mojgan Pourhassan Postdoctoral Researcher	4.52	36196	mojgan.pourhassan@adelaide.edu.au

Real-World Optimisation: some of Dr Markus' work

THE UNIVERSITY OF ADELAIDE

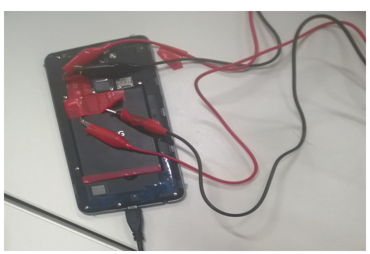


AUSTRALIAN INSTITUTE OF SPORT

Carnegie
CLEAN ENERGY

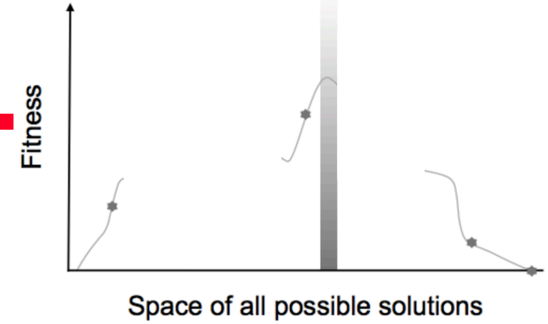
Premier's Research and Industry Fund: $14.6m Research Consortium "Unlocking Complex Resources through Lean Processing" 2017 [plus another $12.5m granted in 2019]

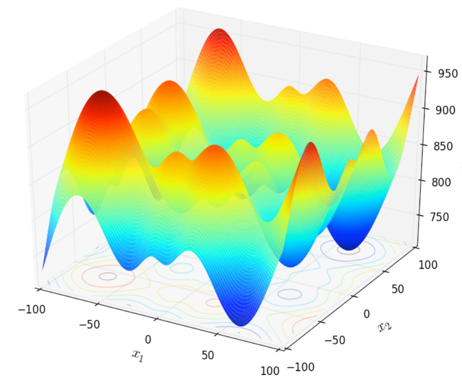
Partners: BHP, OZ Minerals, AMIRA International, Australian Information Industries Association (AIIA) IoT Cluster for Mining and Energy Resources, Australian Semi-Conductor Technology Company, Boart Longyear, Consilium Technology, CRC Optimising Resource Extraction, Datanet, Data to Decisions CRC, Eka, Innovyz, Magotteaux, Manta Controls, Maptek, METS Ignited Industry Growth Centre, Mine Vision Systems, Rockwell Automation, SACOME, SAGE Automation, Sandvik, Scantech, South Australian Mining Industry Participation Office (SA MIPO), SRA IT and Thermo Fisher Scientific Australia (Processing Instruments & Equipment), with the University of South Australia as a key research partner.

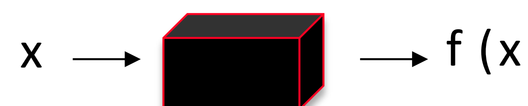


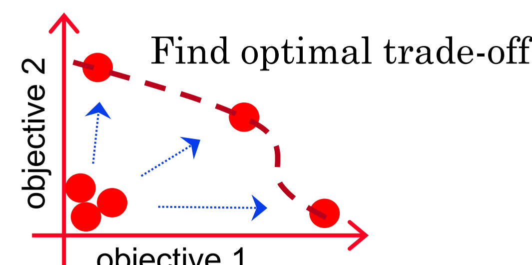
Australian Government
Australian Research Council

My DE16: Dynamic adaptive software configurations









➔ Everybody is after decision support!

(to be faster, decrease wear, deliver on time, …)

Fundamental Research on Optimisation

Supply Chain Management (ARC funded)

- Large scale industrial optimisation problems with many interacting components.

Dynamic Constraints (ARC funded)

- Algorithms for problems with dynamically changing constraints.

Dynamic Adaptive Software Configurations (ARC funded)

- Self-adapt system configurations to changing conditions.

And other grants, and…

- Other knowledge: system modelling + speed-up of simulations (algorithmically or using machine learning), mathematical modelling + optimisation, cybersecurity, …
- High international visibility, about 20-40 papers each year

Introductory Example

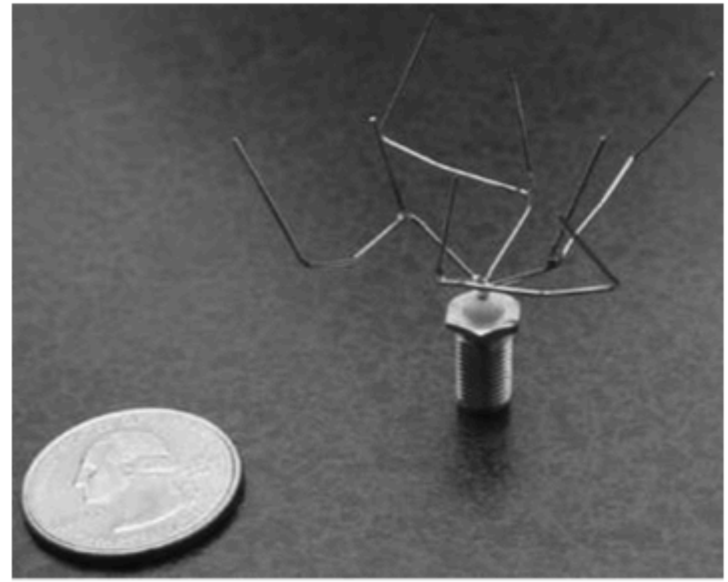
Computer-Automated Evolution of an X-Band Antenna for NASA's Space Technology 5 Mission

This evolved antenna design is the first computer-evolved antenna to be deployed for any application and is the first computer-evolved hardware in space.

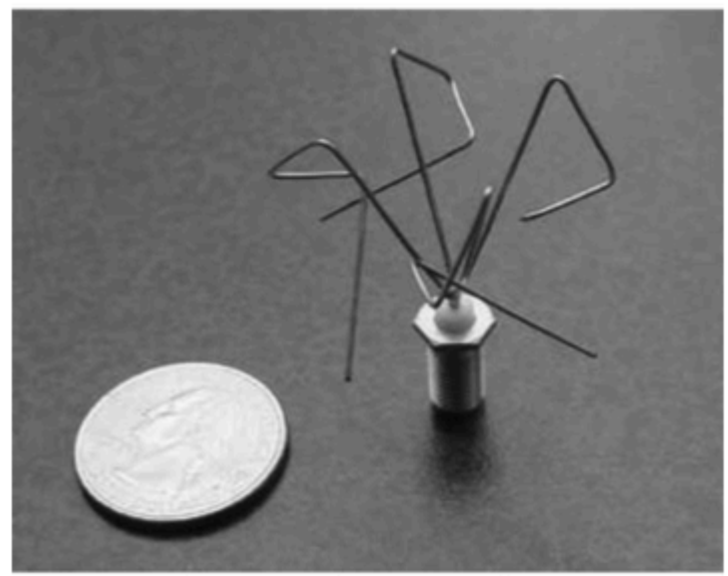
PDF: http://www.mitpressjournals.org/doi/pdf/10.1162/EVCO_a_00005

Youtube video:

https://www.youtube.com/watch?v=HAjozNpBiL4&t=1261s



(a)



(b)

Figure 5: Photographs of prototype evolved antennas: (a) ST5-3-10; (b) ST5-4W-03.

Evolutionary Computation

Teaching Arrangements

- Lecturer
  Dr Markus Wagner (course coordinator)
  http://cs.adelaide.edu.au/~markus
  >150 papers, >150 co-authors, >$9m in research funding

- Consulting and practical help will be available
  Hangyan & Zhibo

- Ask questions, e.g. in the Zoom chat window or on the course page or in the QQ group, or ...
  Your English is better than Markus' Chinese!

Teaching Arrangements

Course Website:

http://linoit.com/users/markuswagnerlinoit/canvases/HIT%20Evolutionary%20Comp.%20Dec%202020

Slides, additional details, questions & answers, homework, …

Quizzes: we are sorting out technical issues (yuketang?)

Text Books

- A. E. Eiben, J. E. Smith: Introduction to Evolutionary Computing, Springer, 2003. 1st Edition. [strongly recommended]
- Z. Michalewicz, D. B. Fogel: How to Solve It: Modern Heuristics, Springer, 2004.
- F. Neumann, C. Witt: Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity, Springer, 2010.
- F. Rothlauf: Design of Modern Heuristics - Principles and Applications, Springer, 2011. [strongly recommended] (slides online: https://www.blogs.uni-mainz.de/fb03-wi-isym/files/2020/01/design-of-modern-heuristics-slides.pdf)

Assessment

There will be 2 assignments:

- 1 group project: implementation + video report [due 18 December]
- Several online quizzes (these are <u>not</u> online yet ➔ if we use yuketang, please use your real name and your student ID)
- Details and deadlines are published on the course's web site

Academic Honesty

- We do not give you assignment work just to keep you busy, we do it to develop your understanding and ability to apply important techniques.
- In general: if you do not do the work yourself, you will not be able to do it in the work force.
- The best way to avoid the need to plagiarise is to start preparing your assignments early. Do not let your work get on top of you!
- You must not copy code from another student or give another student your code to copy from unless specifically authorised to do so by a staff member.
- You may not copy code from anywhere else, without permission.
- If caught, you may receive zero for the assignment, zero for the course or be expelled.
- If you get stuck, seek help from us rather than copying from someone else.

Lecture Times

ONLINE

1. December 1, 13:45-15:30: Introduction
2. December 2, 13:45-15:30: Genetic Algorithms
3. December 3, 13:45-15:30: Evolution Strategies and Genetic Programming
4. December 4, 13:45-15:30: Niching and Multi-Objective Optimisation

5. December 8, 13:45-15:30: Parameter Control and Working with EAs
6. December 9, 13:45-15:30: Ant Colony Optimisation and Particle Swarm Optimisation
7. December 10, 13:45-15:30: Search-Based Software Engineering and Genetic Improvement of Software
8. December 11, 13:45-15:30: More on Genetic Improvement of Software

http://linoit.com/users/markuswagnerlinoit/canvases/HIT%20Evolutionary%20Comp.%20Dec%202020

THE UNIVERSITY OF ADELAIDE AUSTRALIA

Welcome to Evolutionary Computation@HIT, December 2020

This is the lecturer: Dr Markus Wagner https://cs.adelaide.edu.au/~markus/

Assessment

(1) online quizzes:
If possible using the HIT system yuketang (for individual marks).
If not possible: using onlineclicker.org (the entire class then gets the same mark)
==> Deadline: the NEXT day at midnight (11:55pm)

(2) group assignment "implementation + video report":
i) the TAs will mark the code
ii) Markus will mark the video report
==> Deadline: 18 December (Friday, 11:55pm) [this is 1 week after the last lecture]

Final mark:
- group work (implementation + video): 80% (68% for implementation + 32% video)
- quizzes: 20%

Schedule

ONLINE (VooV or Zoom, all times are in 'Harbin time')

NOTE: SLIDES ARE SUBJECT TO SMALL CHANGES --> download them on the next day to ensure you have the most current version

1. December 1, 13:45-15:30: Introduction
https://cs.adelaide.edu.au/~markus/teaching/20hitec/01-Intro.pdf
2. December 2, 13:45-15:30: Genetic Algorithms
https://cs.adelaide.edu.au/~markus/teaching/20hitec/02-Genetic-Algorithms.pdf
3. December 3, 13:45-15:30: Evolution Strategies and Genetic Programming
https://cs.adelaide.edu.au/~markus/teaching/20hitec/03-Evolution-Strategies-Genetic-Programming.pdf
4. December 4, 13:45-15:30: Niching and Multi-Objective Optimisation
https://cs.adelaide.edu.au/~markus/teaching/20hitec/04-EMO.pdf

5. December 8, 13:45-15:30: Parameter Control and Working with EAs
https://cs.adelaide.edu.au/~markus/teaching/20hitec/06-workingWithAlgorithms.pdf
6. December 9, 13:45-15:30: Ant Colony Optimisation and Particle Swarm Optimisation
https://cs.adelaide.edu.au/~markus/teaching/20hitec/07-ACO.pdf
https://cs.adelaide.edu.au/~markus/teaching/20hitec/07-PSO.pdf
7. December 10, 13:45-15:30: Search-Based Software Engineering and Genetic Improvement of Software
https://cs.adelaide.edu.au/~markus/teaching/20hitec/08-01-sbse.pdf
https://cs.adelaide.edu.au/~markus/teaching/20hitec/08-02-gi.pdf
8. December 11, 13:45-15:30: More on Genetic Improvement of Software
https://cs.adelaide.edu.au/~markus/teaching/20hitec/08-03-gin.pdf

voluntary reading material:
Theory of Evolutionary Algorithms: https://cs.adelaide.edu.au/~markus/teaching/20hitec/05-Complexity.pdf

POST QUESTIONS BELOW

this is a test

Study in Adelaide, Australia?

Chinese Scholarship Council (CSC), UoA brochure: https://cs.adelaide.edu.au/~markus/teaching/2019CSCbrochure.pdf (unsure if the CSC still supports this)

CSC in Wind/Wave Energy (joint degree with Shanghai Jiao Tong University): https://cs.adelaide.edu.au/~markus/teaching/2019CSCscholarshipforPhDstudywithintheAustralia-ChinaJointResearchCentreofOffshoreWindandWaveEnergyHarnessing.pdf (while closed, we sometimes have money for scholarships)

University of Adelaide videos:
https://cs.adelaide.edu.au/~markus/teaching/ECMSpromotionalvideos.xlsx

Adelaide is regularly one of the world's most liveable cities: https://www.australiasbestcity.com.au/adelaide-is-australias-most-liveable-city/

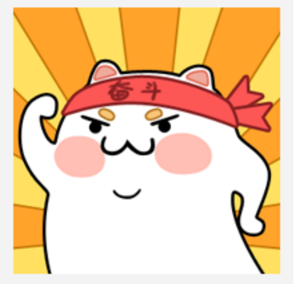
Uni Adelaide rankings:
- QS World University Rankings: 106
- Times Higher Education World University Rankings: 118
- Shanghai ranking: School of Computer Science ranked 40 in the world (2nd in Australia)

More information:
Optimisation and Logistics Group: https://cs.adelaide.edu.au/~optlog/
School of Computer Science: https://ecms.adelaide.edu.au/computer-science/
University of Adelaide: https://www.adelaide.edu.au/

HIT Evolutionary Comp. D

Highlight New Show Private


好好学习
天天向上



Today’s Contents

- Positioning of EC and the basic EC metaphor
- Historical perspective
- Biological inspiration:
  - Darwinian evolution theory (simplified!)
- Motivation for EC
- Example of an evolutionary algorithm

Positioning of EC

- EC is part of computer science
- EC is not part of life sciences/biology
- Biology delivered inspiration and terminology
- EC can be applied in biological research

The Main Evolutionary Computing Metaphor

EVOLUTION | **PROBLEM SOLVING**

EVOLUTION		PROBLEM SOLVING
Environment	⟷	Problem
Individual	⟷	Candidate Solution
Fitness	⟷	Quality

Fitness → chance for survival and reproduction

Quality → chance for seeding new solutions

Brief History 1: the ancestors

- 1948, Turing:
  proposes "genetical or evolutionary search"
- 1962, Bremermann
  optimization through evolution and recombination
- 1964, Rechenberg
  introduces evolution strategies
- 1965, L. Fogel, Owens and Walsh
  introduce evolutionary programming
- 1975, Holland
  introduces genetic algorithms
- 1992, Koza
  introduces genetic programming
- 1992, Dorigo
  introduces ant-colony optimisation
- 1995, Kennedy, Eberhardt and Shi
  introduce particle swarm optimisation

Brief History 2: The rise of EC

1985: first international conference (ICGA)

1990: first international conference in Europe (PPSN)

1993: first scientific EC journal (MIT Press)

EC in the early 21st Century

- 3 major EC conferences (GECCO, PPSN, CEC), about 10 small related ones
- 2 scientific core EC journals (MIT Evolutionary Computation, IEEE Transactions on Evolutionary Computation) and many others
- uncountable (meaning: many) applications
- uncountable (meaning: ?) consultancy and R&D firms

Em/Prof Zbigniew Michalewicz:

NuTech (now part of IBM)

SolveIT (now part of Schneider Electric)

Darwinian Evolution 1: Survival of the fittest

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- All environments have finite resources
  (i.e., can only support a limited number of individuals)
- Life forms have basic instincts/lifecycles geared towards reproduction
- Therefore some kind of selection is inevitable
- Those individuals that compete for the resources most effectively have an increased chance of reproduction
- Note: fitness in natural evolution is a derived, secondary measure, i.e., we (humans) assign a high fitness to individuals with many offspring (?)

Darwinian Evolution 2: Diversity drives change

- Phenotypic traits:
  - Behaviour / physical differences that affect response to environment
  - Partly determined by inheritance, partly by factors during development
  - Unique to each individual, partly as a result of random changes
- “Suitable” phenotypic traits:
  - Lead to higher chances of reproduction
  - Can be inherited

  then they will tend to increase in subsequent generations,
- leading to new combinations of traits …

Darwinian Evolution: Summary

- Population consists of a diverse set of individuals
- Combinations of traits that are better adapted tend to increase representation in population

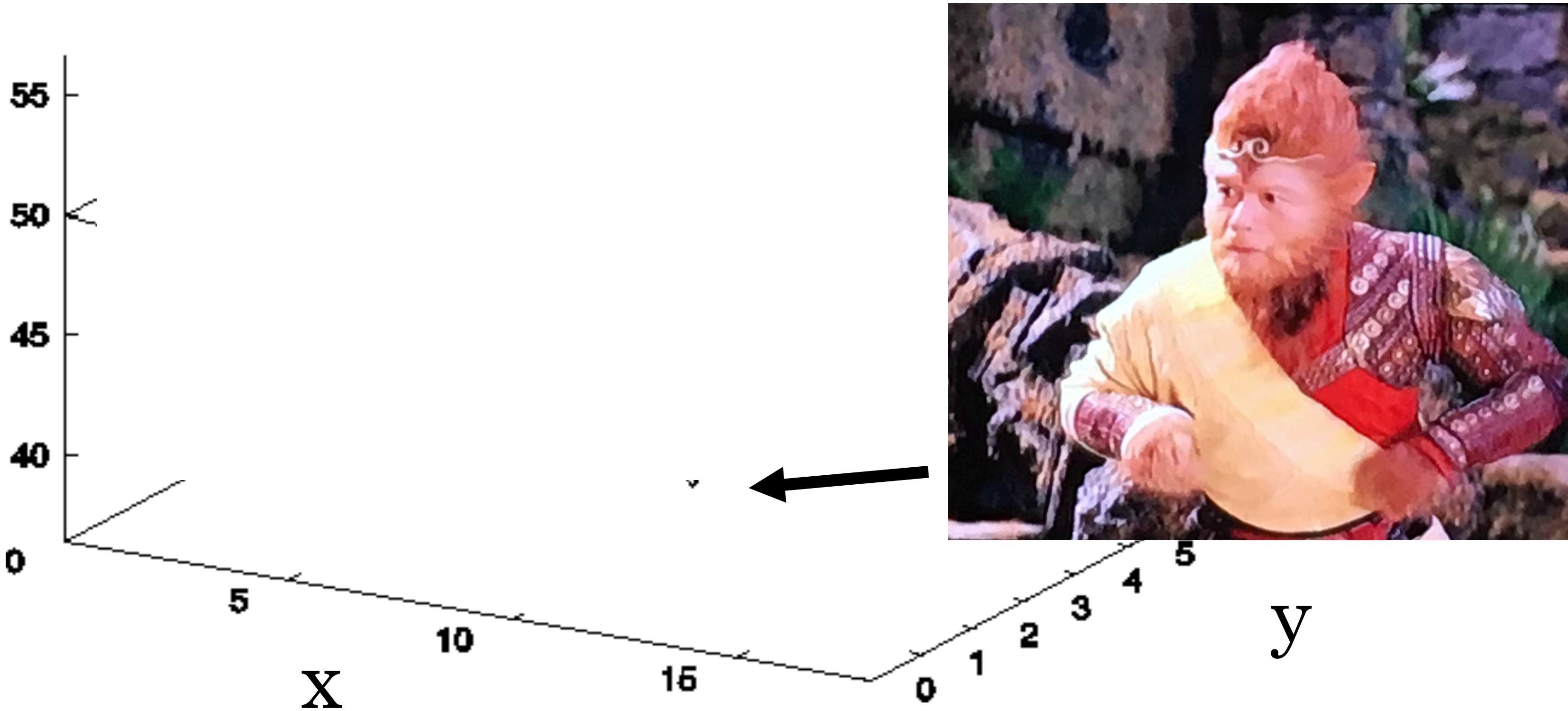
Individuals are "units of selection"

- Variations occur through random changes yielding constant source of diversity, coupled with selection means that:

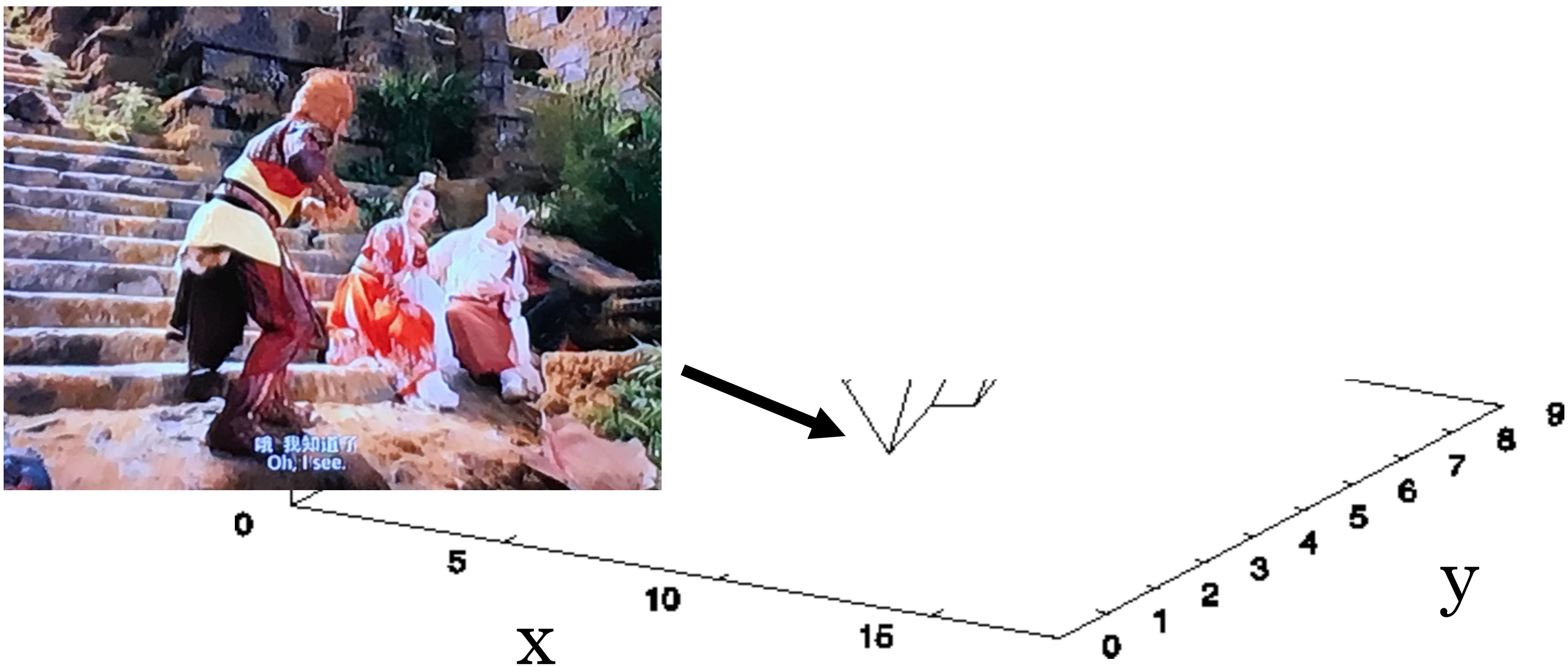
Population is the "unit of evolution"

- Note the absence of a "guiding force"

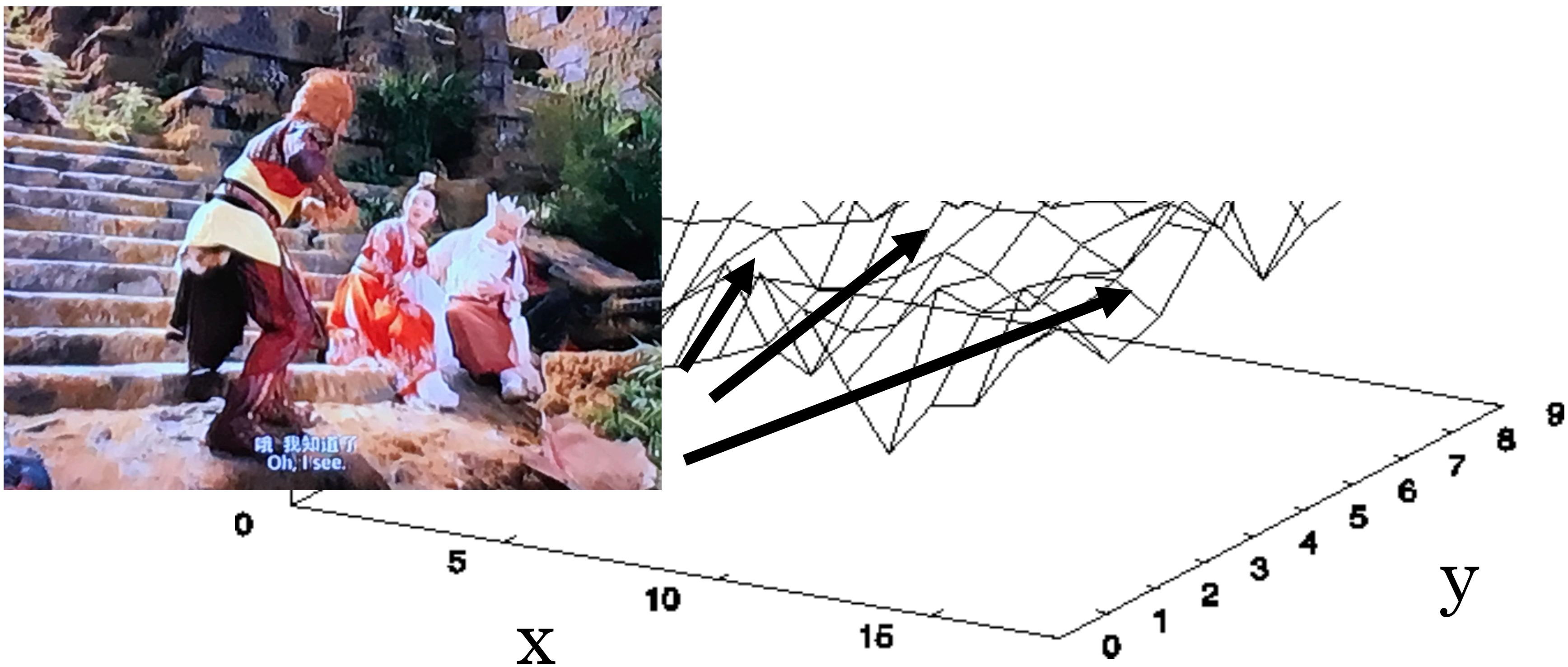
Monkey King: find the treasure on the highest mountain



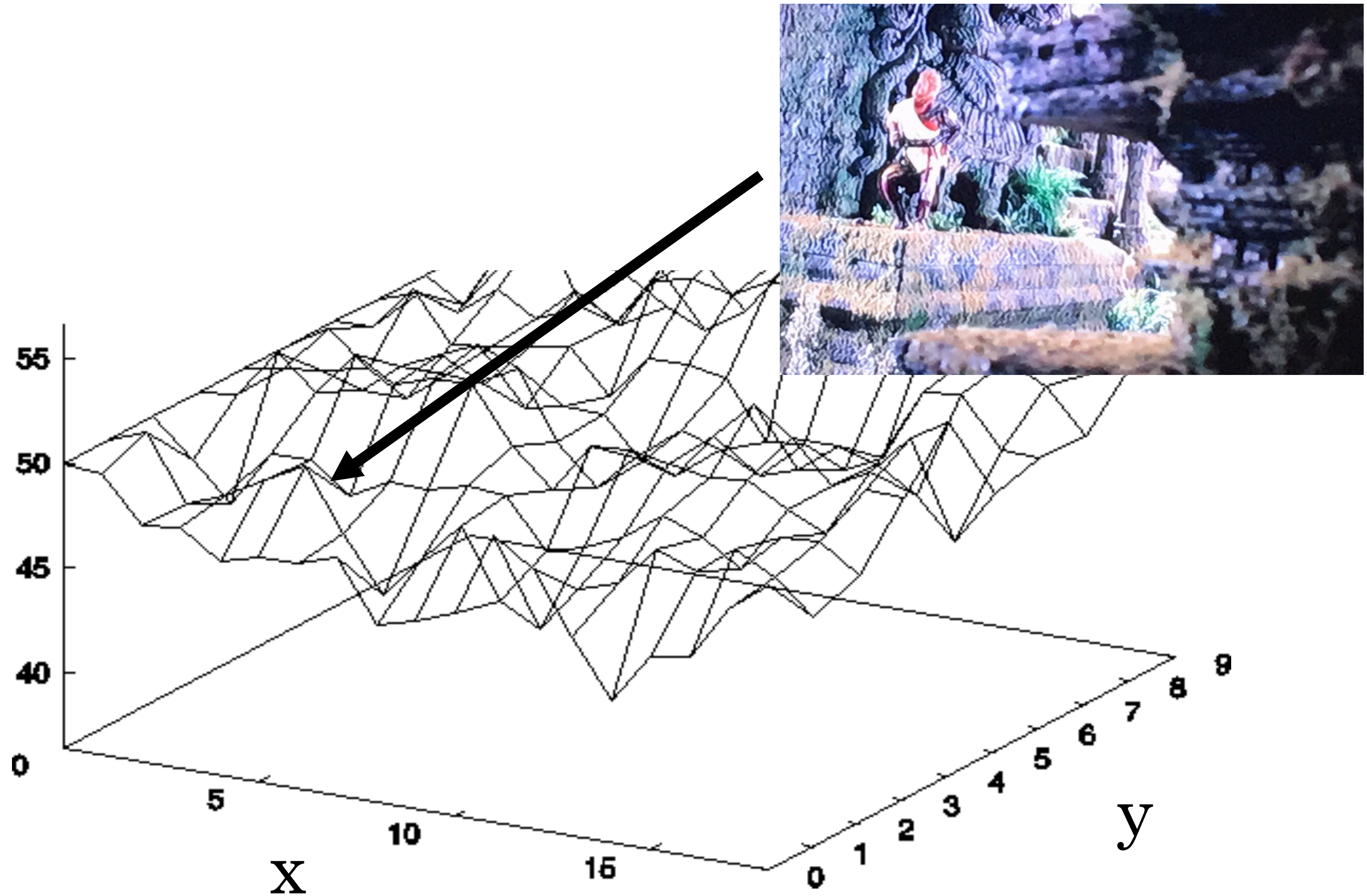
Monkey King: find the treasure on the highest mountain



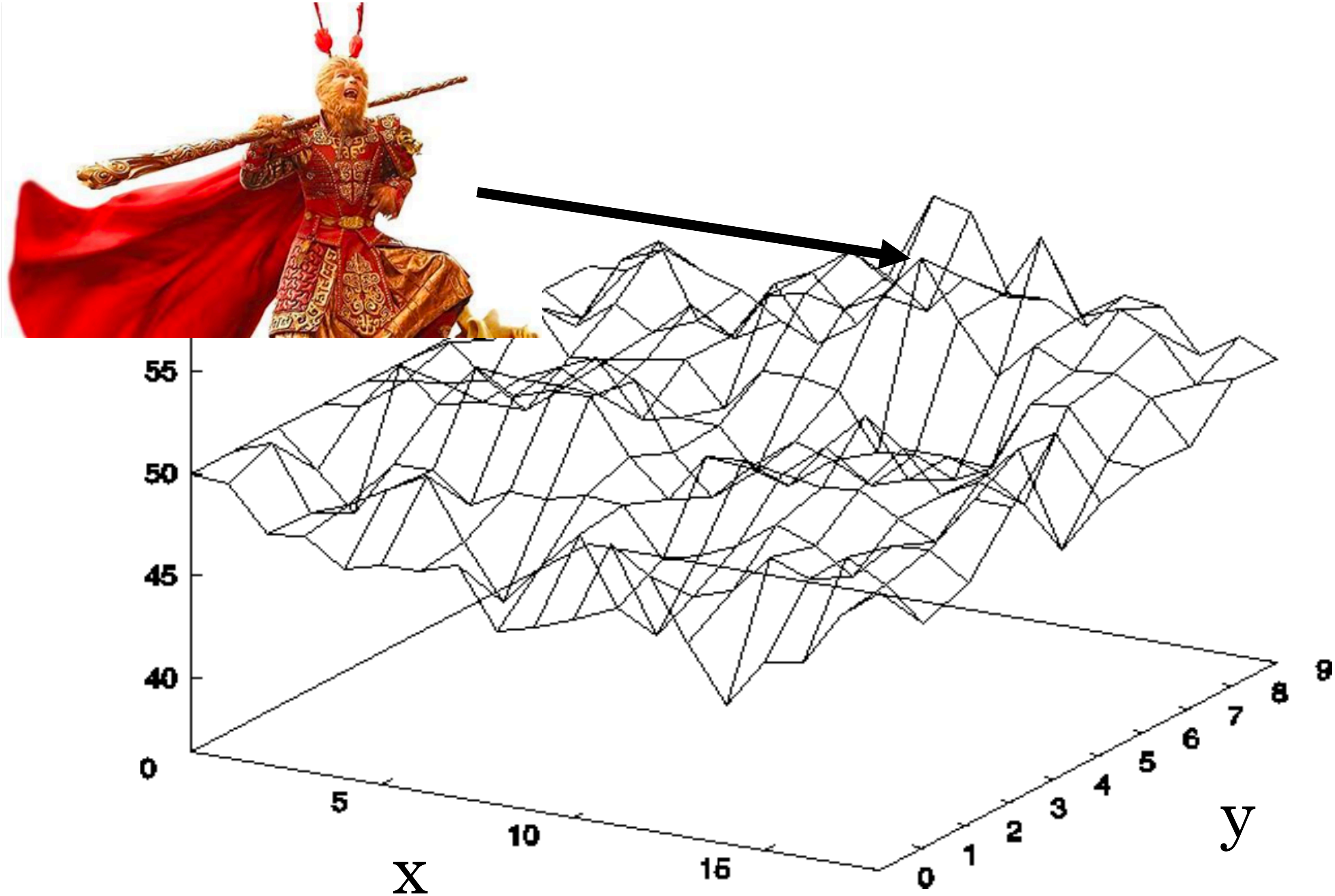
Monkey King: find the treasure on the highest mountain



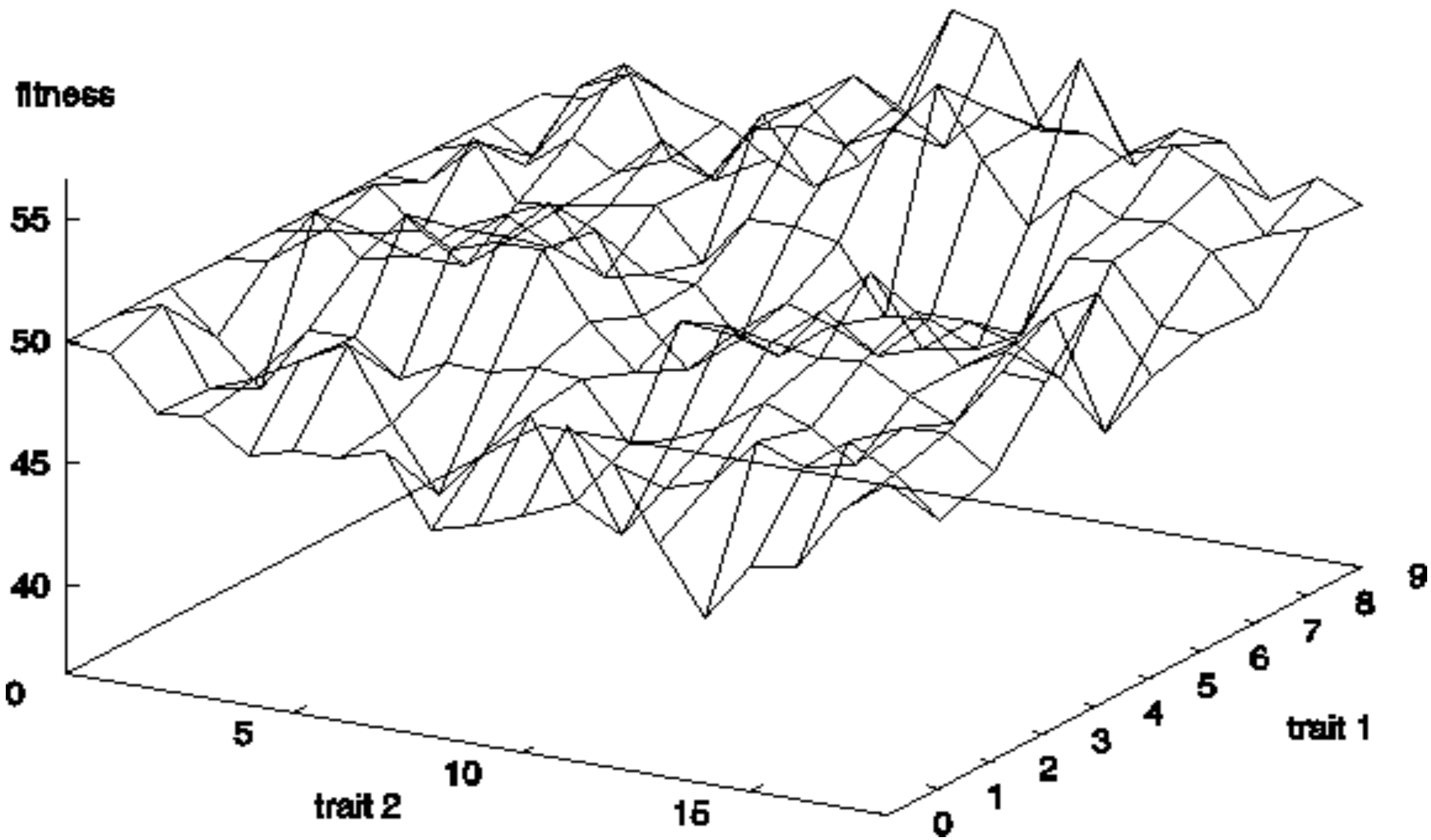
Monkey King: find the treasure on the highest mountain



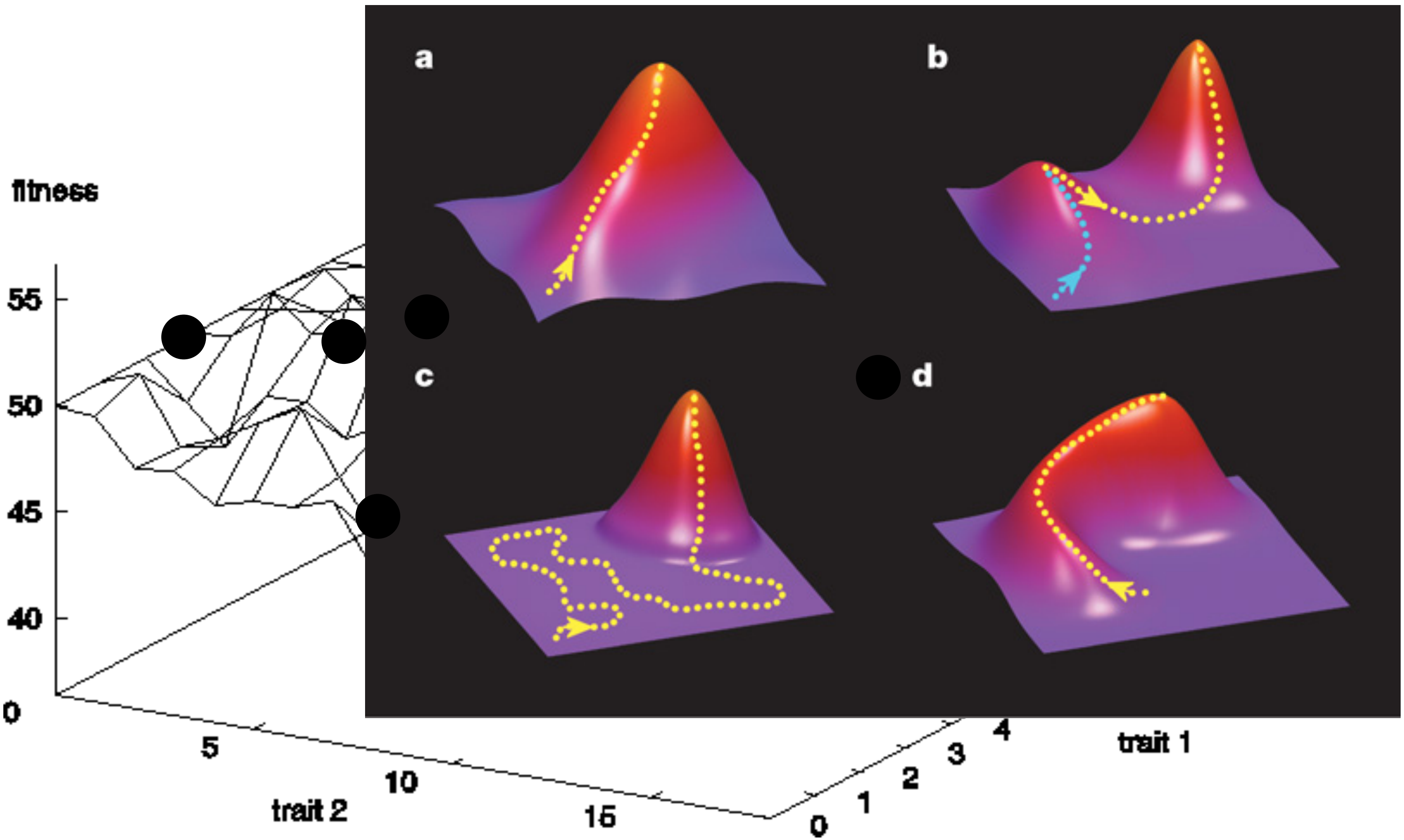
Monkey King: find the treasure on the highest mountain



Example with two traits



Example with two traits



Adaptive landscape metaphor (cont’d)

- Selection “pushes” population up the landscape

- Genetic drift:
  - random variations in feature distribution
    (+ or -) arising from sampling error
  - can cause the population to “melt down” hills, thus crossing valleys and leaving local optima

Motivations for EC: 1

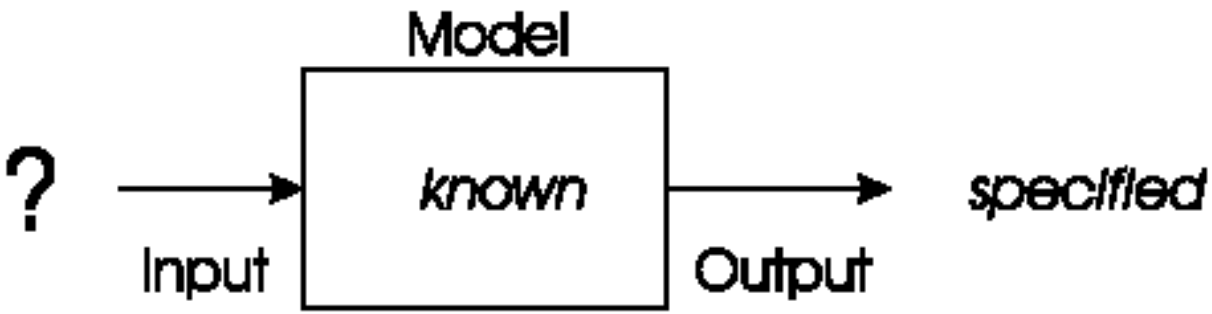
- Nature has always served as a source of inspiration for engineers and scientists
- The best problem solver known in nature is:
  - the (human) brain that created “the wheel, New York, wars and so on” (after Douglas Adams’ Hitch-Hikers Guide)
  - the evolution mechanism that created the human brain (after Darwin’s Origin of Species)
- Answer 1 → neurocomputing
- Answer 2 → evolutionary computing

Motivations for EC: 2

- Developing, analysing, applying problem solving methods a.k.a. algorithms is a central theme in mathematics and computer science
- Time for thorough problem analysis decreases
- Complexity of problems to be solved increases
- Consequence:
  Robust problem solving technology needed

Problem type 1: Optimisation

We have a model of our system and seek inputs that give us a specified goal

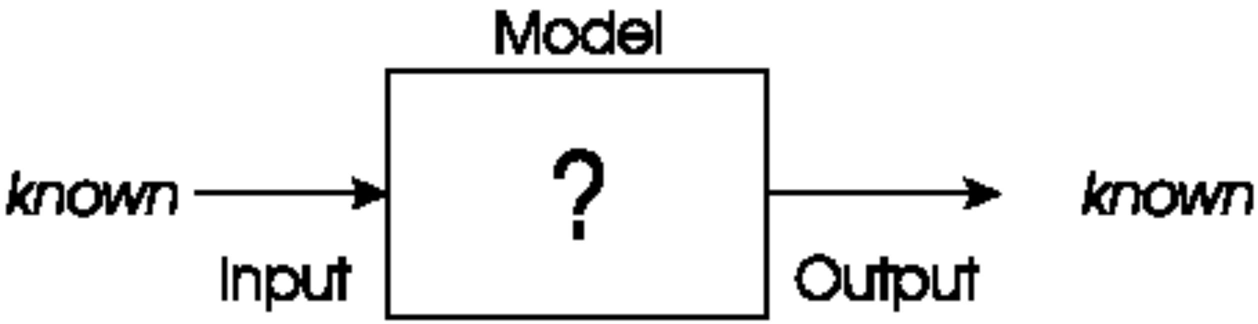


For example

- time tables for university, call centre, or hospital
- design specifications, etc.

Problem type 2: Modelling

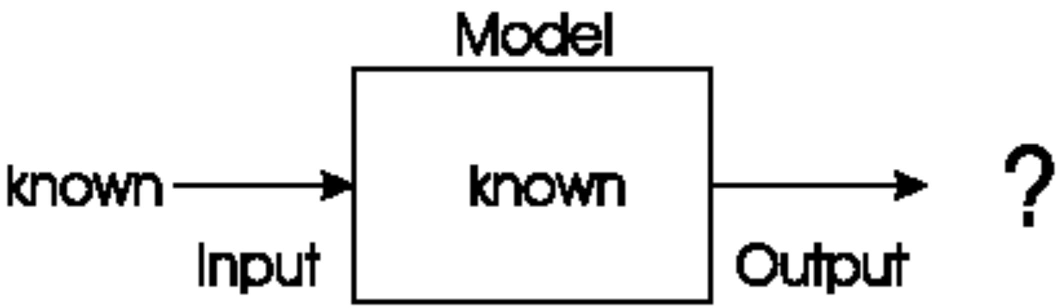
We have corresponding sets of inputs & outputs and seek model that delivers correct output for every known input



For example: evolutionary machine learning

Problem type 3: Simulation

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



Often used to answer “what-
environments, e.g. Evolutio

Evolutionary Algorithms: Overview

- Recap of Evolutionary Metaphor
- Basic scheme of an EA
- Basic Components:
  - Representation / Evaluation / Population / Parent Selection / Recombination / Mutation / Survivor Selection / Termination
- Example: eight queens

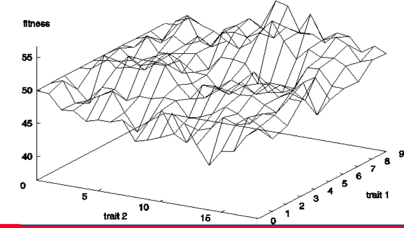
Recap of EC metaphor

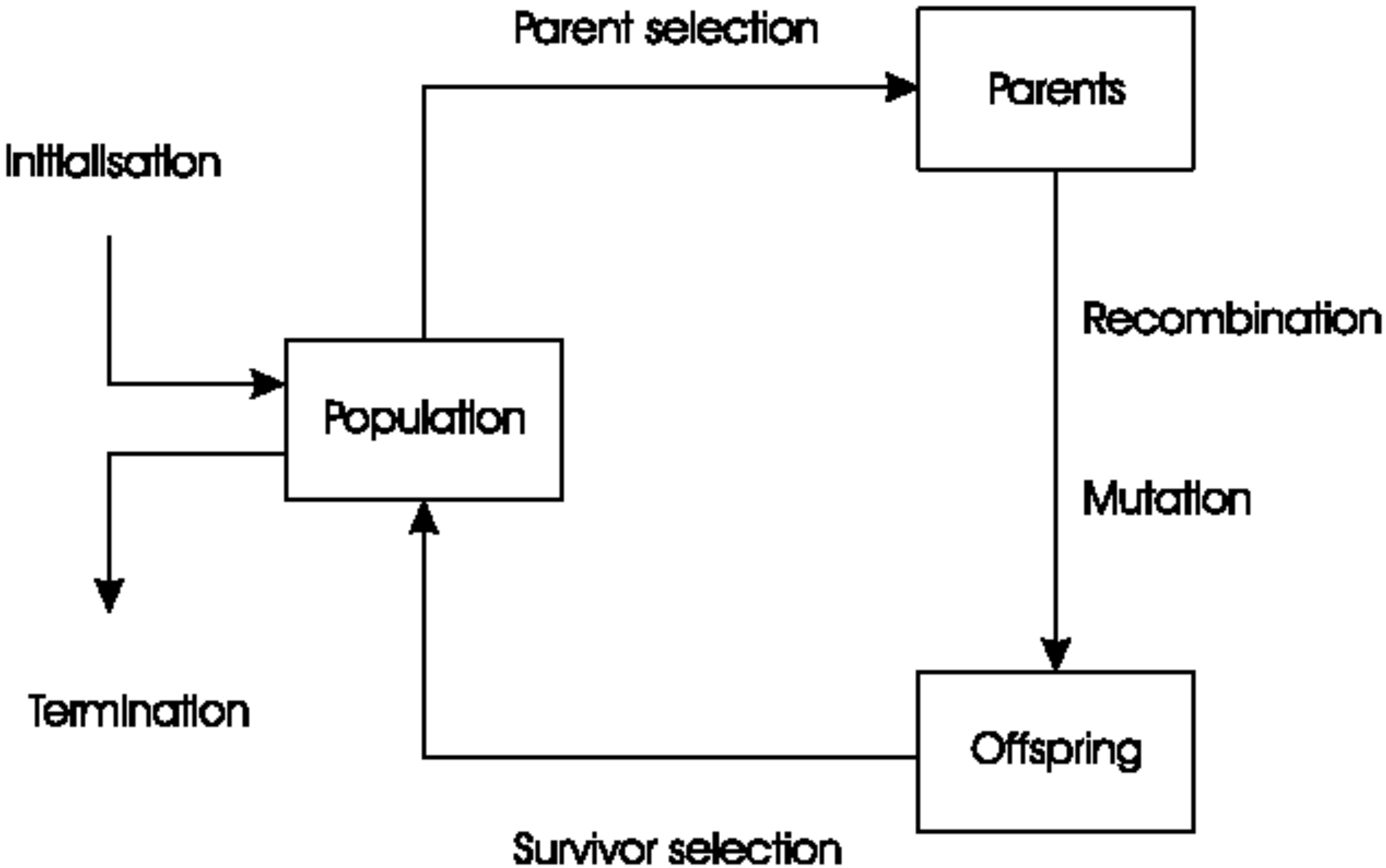
- A population of individuals exists in an environment with limited resources
- ***Competition*** for those resources causes selection of those ***fitter*** individuals that are better adapted to the environment
- These individuals act as seeds for the generation of new individuals through recombination and mutation
- The new individuals have their fitness evaluated and compete (possibly also with parents) for survival.
- Over time ***Natural selection*** causes a rise in the fitness of the population

Recap 2:

- EAs fall into the category of "generate and test" algorithms
- They are stochastic, population-based algorithms
- Variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty
- Selection reduces diversity and acts as a force pushing quality

General Scheme of EAs





Pseudo-code for typical EA

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

What are the different types of EAs

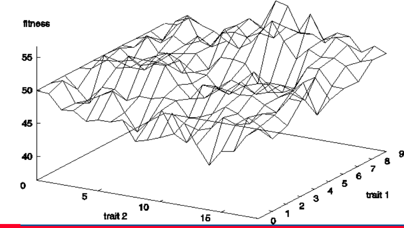
- Historically different flavours of EAs have been associated with different representations
  - Binary strings: Genetic Algorithms
  - Real-valued vectors: Evolution Strategies
  - Finite state Machines: Evolutionary Programming
  - LISP trees: Genetic Programming
- These differences are largely irrelevant, best strategy
  - choose representation to suit problem
  - choose variation operators to suit representation
- Selection operators only use fitness and so are independent of representation

Representations

- Candidate solutions (individuals) exist in *phenotype* space
- They are encoded in chromosomes, which exist in *genotype* space
  - Encoding : phenotype => genotype (not necessarily one to one)
  - Decoding : genotype => phenotype (must be one to one)
- Chromosomes contain genes, which are in (usually fixed) positions called loci (sing. locus) and have a value (allele)

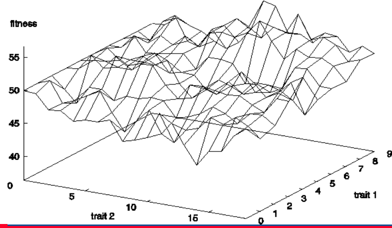
In order to find the global optimum, every feasible solution must be represented in genotype space

Evaluation (Fitness) Function



- Represents the requirements that the population should adapt to
- a.k.a. ***quality*** function or ***objective*** function
- Assigns a single real-valued fitness to each phenotype which forms the basis for selection
  - So the more discrimination (different values) the better
- Typically we talk about fitness being maximised
  - Some problems may be best posed as minimisation problems, but conversion is trivial

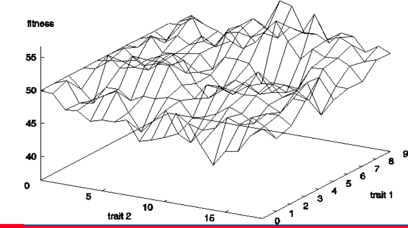
Population



- Holds (representations of) possible solutions
- Usually has a fixed size and is a *multiset* of genotypes
- Some sophisticated EAs also assert a spatial structure on the population, e.g., a grid.
- Selection operators usually take whole population into account i.e., reproductive probabilities are *relative* to *current* generation
- Diversity of a population refers to the number of different fitnesses / phenotypes / genotypes present (note not the same thing)

Onlineclicker.org: 8889
What is best:
1. "Small" populations? (e.g. 10 individuals)
2. "Large" populations? (e.g. 10 000 individuals)

Parent Selection Mechanism



- Assigns variable probabilities of individuals acting as parents depending on their fitnesses
- Usually probabilistic
  - high quality solutions more likely to become parents than low quality
  - but not guaranteed
  - even worst in current population usually has non-zero probability of becoming a parent
- This ***stochastic*** nature can aid escape from local optima

Variation Operators

- Role is to generate new candidate solutions
- Usually divided into two types according to their arity (number of inputs):
  - Arity 1 : mutation operators
  - Arity >1 : Recombination operators
  - Arity = 2 typically called crossover
- There has been much debate about relative importance of recombination and mutation
  - Nowadays most EAs use both
  - Choice of particular variation operators is representation dependant

Mutation

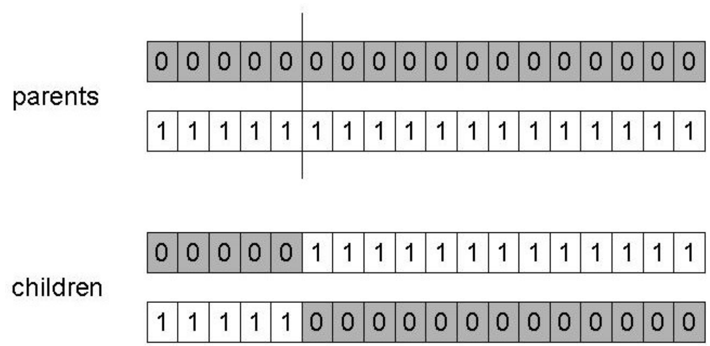
- Acts on one genotype and delivers another
- Element of randomness is essential and differentiates it from other unary heuristic operators
- Importance ascribed depends on representation and dialect:
  - Binary GAs – background operator responsible for preserving and introducing diversity
  - EP for FSM’s/continuous variables – only search operator
  - GP – hardly used
- May guarantee connectedness of search space and hence convergence proofs

parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

Question
How about TSP tours?
(Travelling Salesperson Problem)

Recombination

- Merges information from parents into offspring
- Choice of what information to merge is stochastic
- Most offspring may be worse, or the same as the parents
- Hope is that some are better by combining elements of genotypes that lead to good traits
- Principle has been used for millennia by breeders of plants and livestock



Question
How about TSP tours?
(Travelling Salesperson Problem)

Survivor Selection

- a.k.a. ***replacement***
- Most EAs use fixed population size so need a way of going from (parents + offspring) to next generation
- Often deterministic
  - Fitness based : e.g., rank parents+offspring and take best
  - Age based: make as many offspring as parents and delete all parents
- Sometimes do combination (elitism)

Quick Question
Why is Survivor Selection needed?

Initialisation / Termination

Initialisation usually done at random

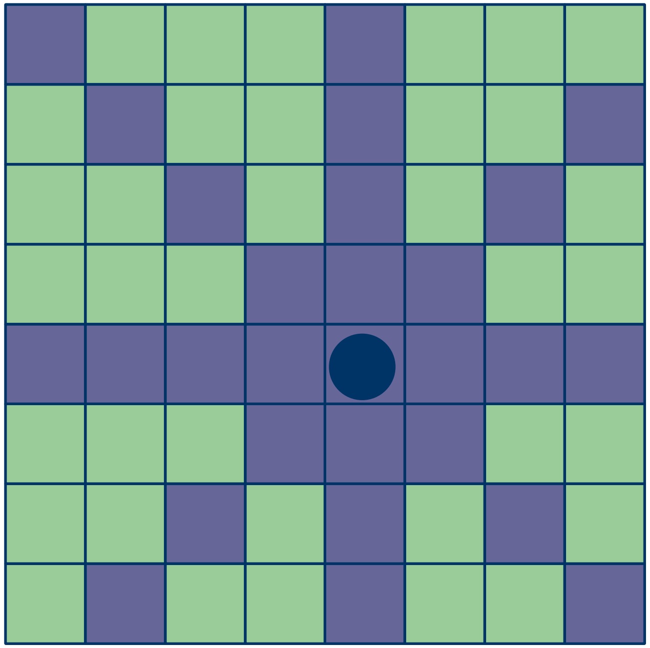
- Need to ensure even spread and mixture of possible allele values
- Can include existing solutions, or use problem-specific heuristics, to "seed" the population

Quick Question
Seeding: what are disadvantages?

Termination condition checked every generation

- Reaching some (known/hoped for) fitness
- Reaching some maximum allowed number of generations
- Reaching some minimum level of diversity
- Reaching some specified number of generations without fitness improvement

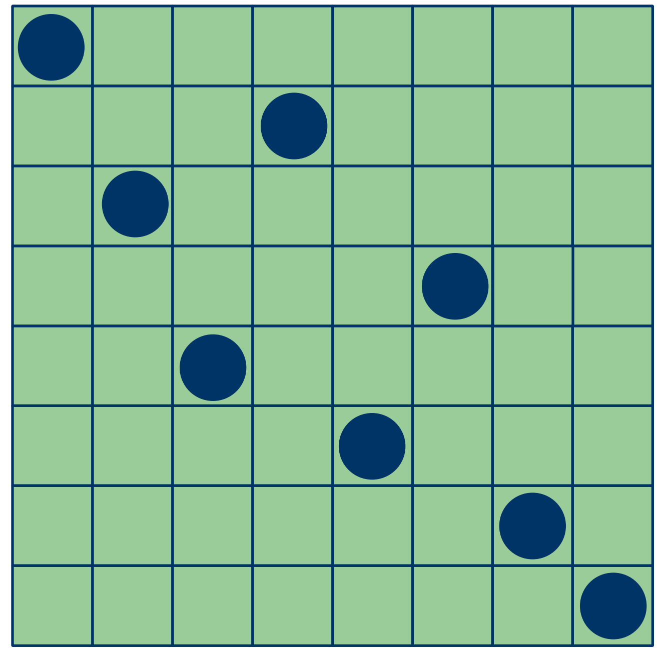
Example: the 8 queens problem



Place 8 queens on an 8x8 chessboard in such a way that they cannot check each other

The 8 queens problem: representation

Phenotype:
a board configuration



Obvious mapping

1	3	5	2	6	4	7	8

Genotype:
a permutation of
the numbers 1 - 8

The 8 queens problem: fitness evaluation

- Penalty of one queen:
  the number of queens she can check.

- Penalty of a configuration:
  the sum of the penalties of all queens.

- Note: penalty is to be minimized

- Fitness of a configuration:
  inverse penalty to be maximized

The 8 queens problem: mutation

Small variation in one permutation, e.g.:

- swapping values of two randomly chosen positions,

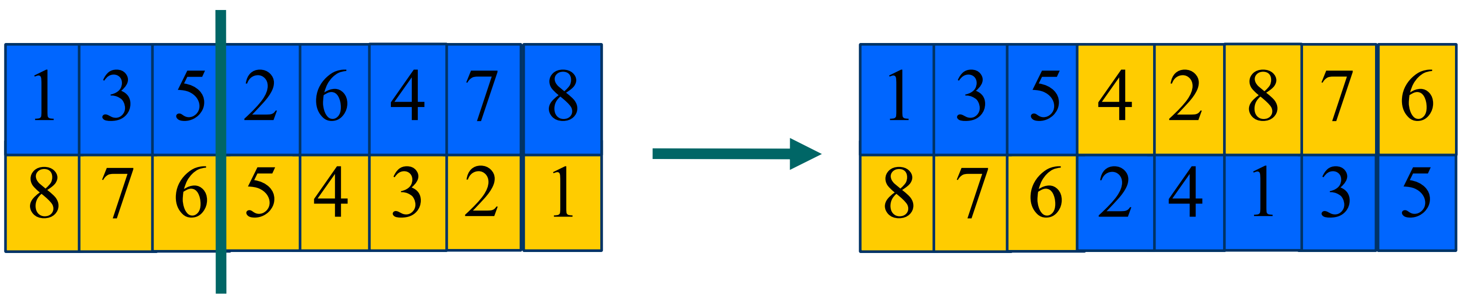
1	3	5	2	6	4	7	8

$\longrightarrow$

1	3	7	2	6	4	5	8

The 8 queens problem: recombination

Combining two permutations into two new permutations:
- choose random crossover point
- copy first parts into children
- create second part by inserting values from other parent:
    - in the order they appear there
    - beginning after crossover point
    - skipping values already in child



The 8 queens problem: selection

Parent selection

- Pick 5 parents and take best two to undergo crossover

Survivor selection (replacement)

- When inserting a new child into the population, choose an existing member to replace by:
- sorting the whole population by decreasing fitness
- enumerating this list from high to low
- replacing the worst individual in the population

The 8 queens problem: summary

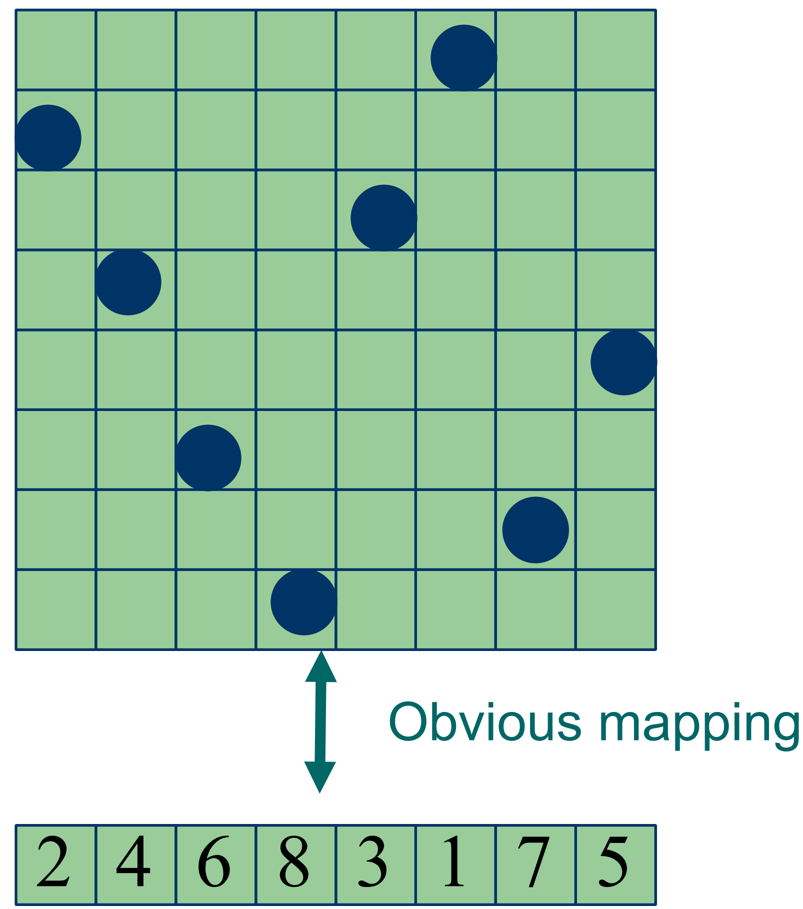
Representation	Permutations
Recombination	"Cut-and-crossfill" crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	80%
Parent selection	Best 2 out of random 5
Survival selection	Replace worst
Population size	100
Number of Offspring	2
Initialisation	Random
Termination condition	Solution or 10,000 fitness evaluation

Note: this is ***only one possible***
set of choices of operators and parameters

The 8 queens problem: example solution

Phenotype:
a board configuration

Genotype:
a permutation of
the numbers 1 - 8



Evolutionary Computation

December 2020

Course page

http://linoit.com/users/markuswagnerlinoit/canvases/HIT%20Evolutionary%20Comp.%20Dec%202020