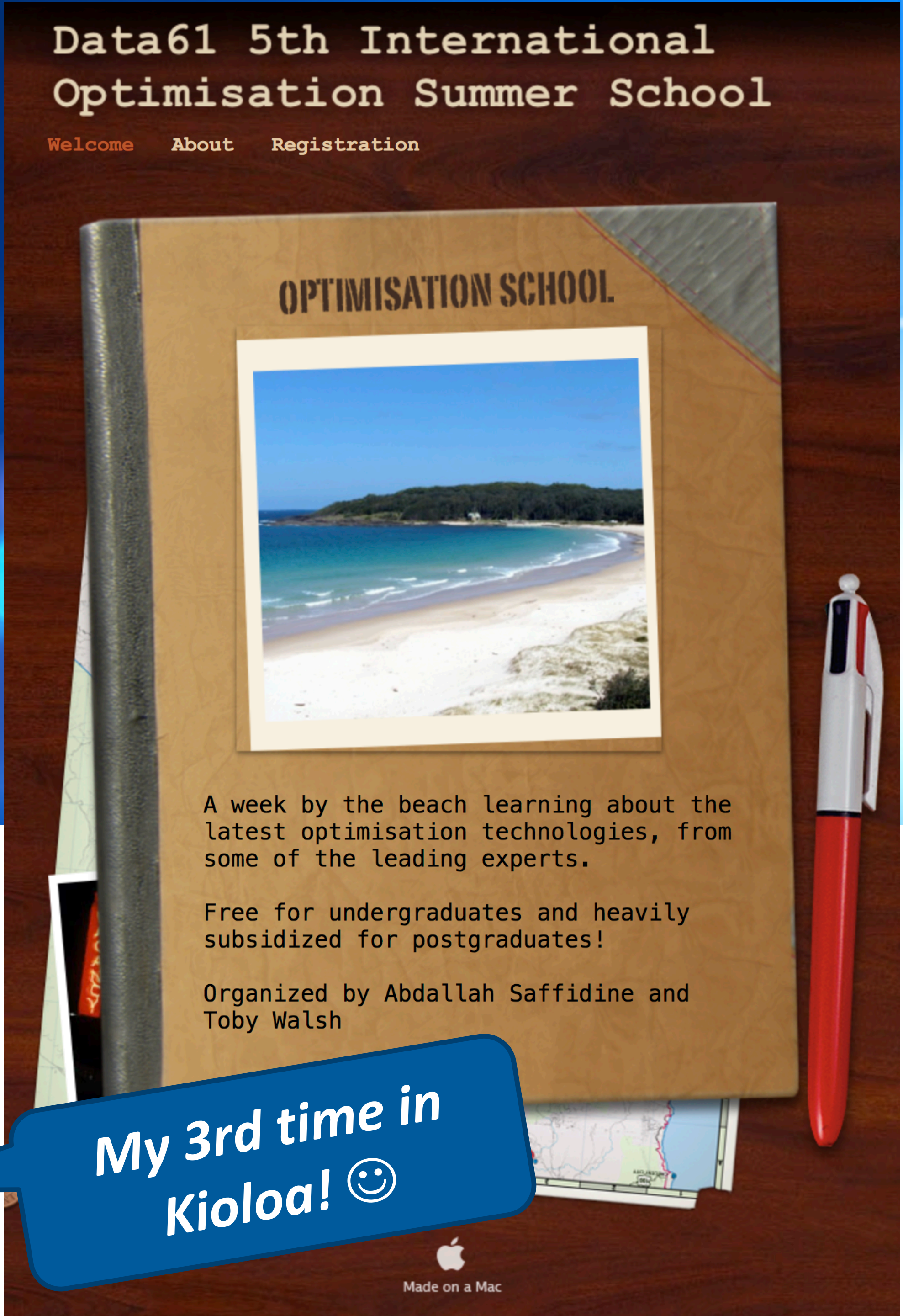
THE UNIVERSITY of ADELAIDE

Optimisation and Logistics

Markus Wagner

markus.wagner@adelaide.edu.au



My 3rd time in Kioloa! ☺

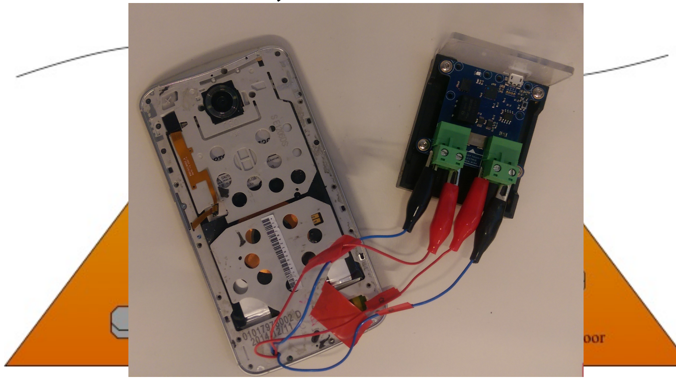
Optimisation and Logistics Group

Our research agenda

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

Group Leader: Prof. Frank Neumann
Staff: 8 faculty members, 1 postdoc, 10 PhD students

Topics: wind/wave energy, large-scale optimisation, theory of bio-inspired computing, computational economics, search-based software engineering, ...



http://cs.adelaide.edu.au/~optlog
http://cs.adelaide.edu.au/~markus
Online: code, articles, ...

Markus Wagner

2003-2009 UNIVERSITÄT KOBLENZ · LANDAU

2006-2007 The University of Georgia

2010-2013 max planck institut informatik

2013 THE UNIVERSITY of ADELAIDE

Senior Lecturer

Summary: 70+ papers/co-authors/reviews/events/...

1 best paper/poster/presentation/keynote/medal/...

3rd time in Kioloa ☺


Lead[redacted]g

Charles Robert Darwin

Follow

naturalist (1809-1882)
life sciences, evolution, biogeography, speciation, natural selection
Verified email at unr.edu.ar - Homepage

Title 1–20	Cited by	Year
On the Orngin of Species CR Darwin London: John Murray	34323 *	1859
Origin of species CMA Darwin DMP	~~34234~~ *	1978
R.(1859): On the Origin of Species by Means of Natural Selection C Darwin Murray. London	~~33919~~ *	1871
On the origin of species, facsimile of. 6th CD Darwin Cambridge: Harvard University Press	~~33642~~ *	1872
The decent of man, and selectionin relation to sex C Darwin Princeton, Princeton University Press	18807 *	1871
The descent ofman, andselection in relation to sex C Darwin London: Murray	~~18743~~ *	1871
The descent ofman, and selection in relation to sex (2 vols.) C Darwin London: Murray	~~18704~~ *	1871
The descent of man, and selection in relation to sex C Darwin Murray	~~18690~~	1888
The Descent of Man, and Selection in Relation to Sex.(1966) C Darwin	~~18686~~ *	1871

Google Scholar

Citation indices	All	Since 2012
Citations	131694	37156
h-index	104	59
i10-index	512	205

Co-authors View all...

Alfred Russel Wallace (1823-1913)

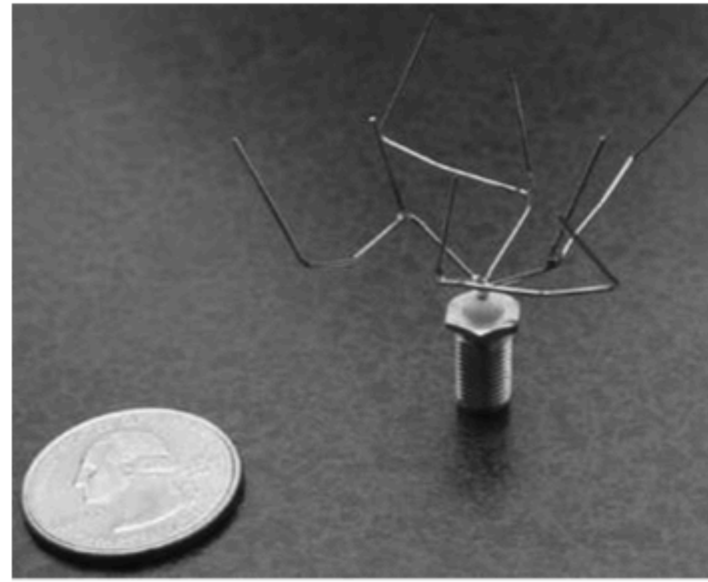
Introductory Example

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

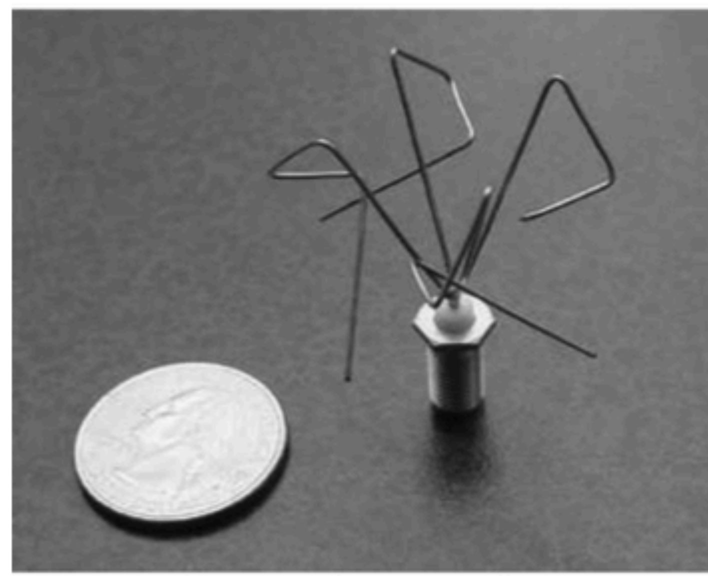
This evolved antenna design is the first computer-evolved antenna to be deployed for any application and is the first computer-evolved hardware in space.

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

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



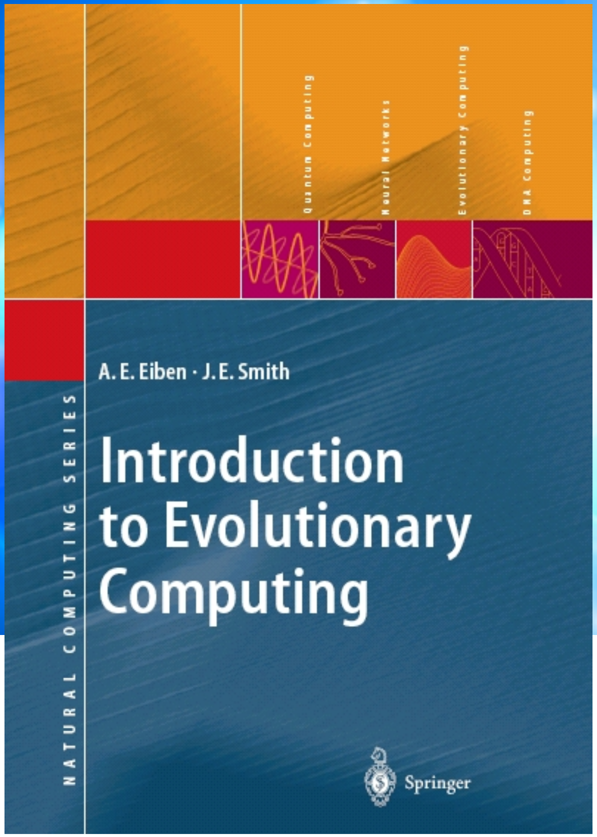
(a)



(b)

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

Evolutionary Computation



orange = 1st edition (preferred)
green = 2nd edition

Text Books

- A. E. Eiben, J. E. Smith: Introduction to Evolutionary Computing, Springer, 2003. [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. Free: http://www.bioinspiredcomputation.com
- F. Rothlauf: Design of Modern Heuristics - Principles and Applications, Springer, 2011. [strongly recommended]

Today's Contents

1st Session

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

2nd Session

- Multi-modal problems
- Multi-objective optimisation

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
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, 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

- 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:

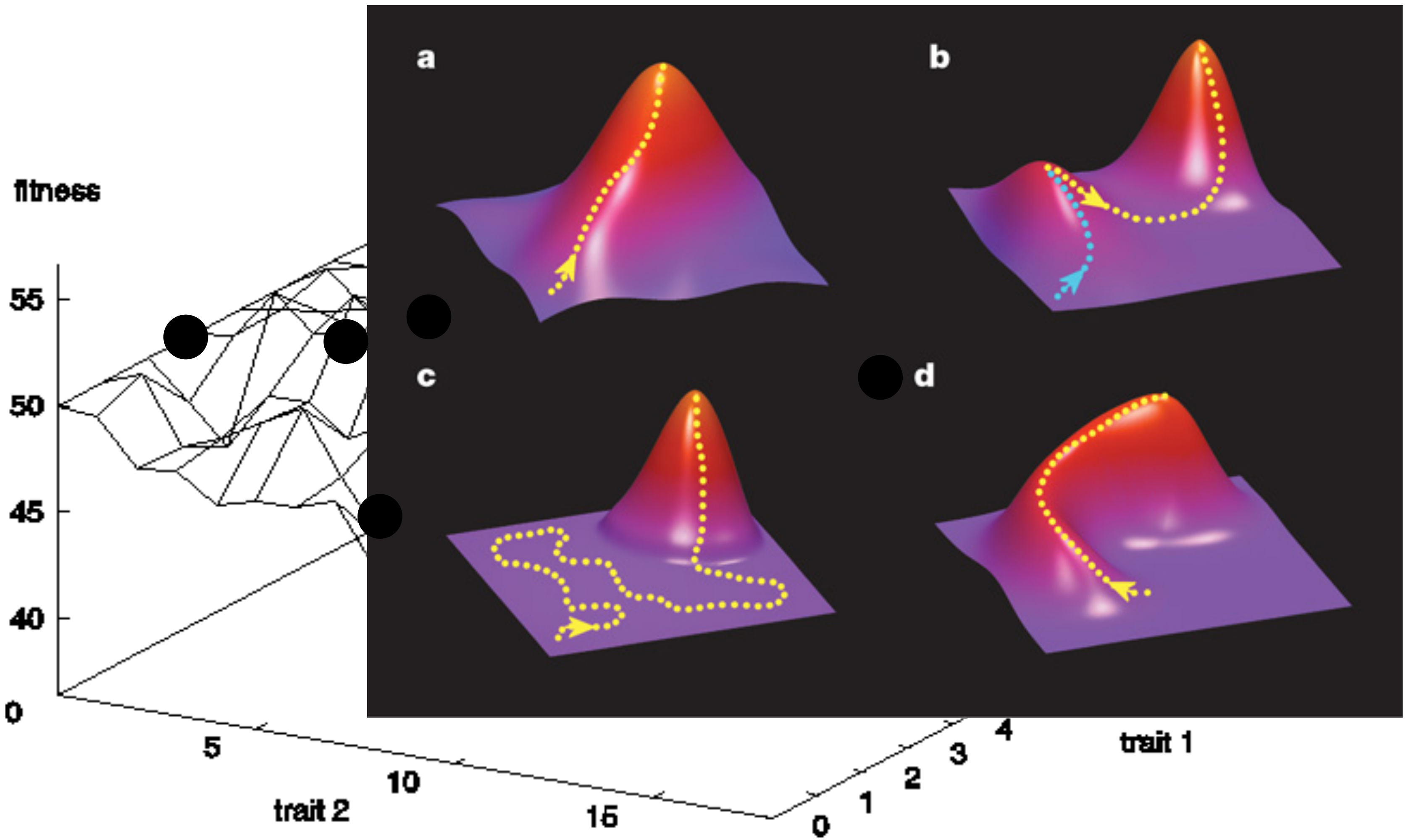
Population is the "unit of evolution"

- Note the absence of a "guiding force"

Adaptive landscape metaphor
(Wright, 1932)

- Can envisage population with n traits as existing in a n+1-dimensional space (landscape) with height corresponding to fitness
- Each different individual (phenotype) represents a single point on the landscape
- Population is therefore a “cloud” of points, moving on the landscape over time as it evolves - adaptation

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

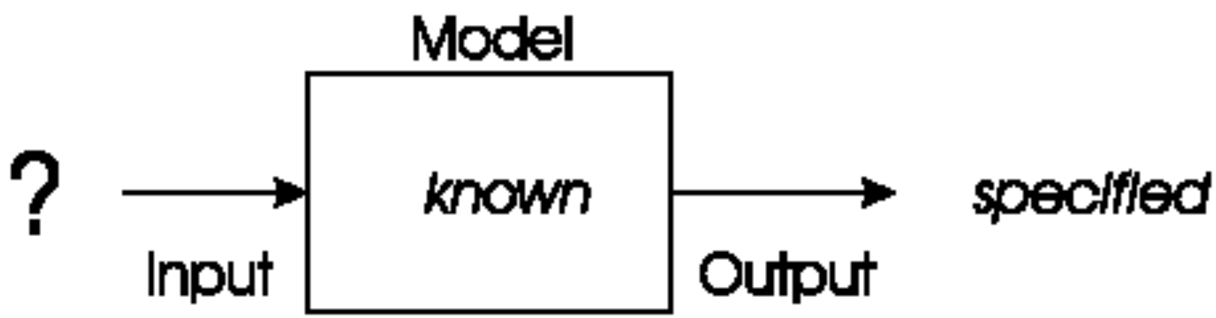
- Time for thorough problem analysis decreases

- 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



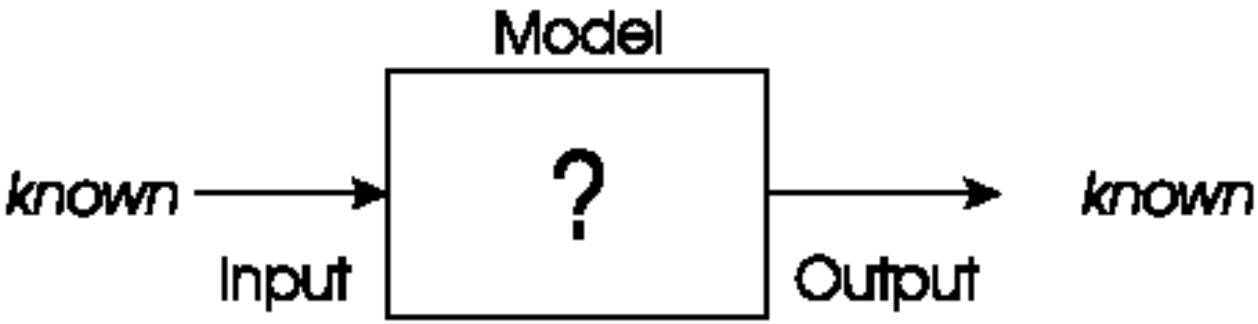
For example

– time tables for university, call centre, or hospital

– design specifications, etc.

Problem types 2: Modelling

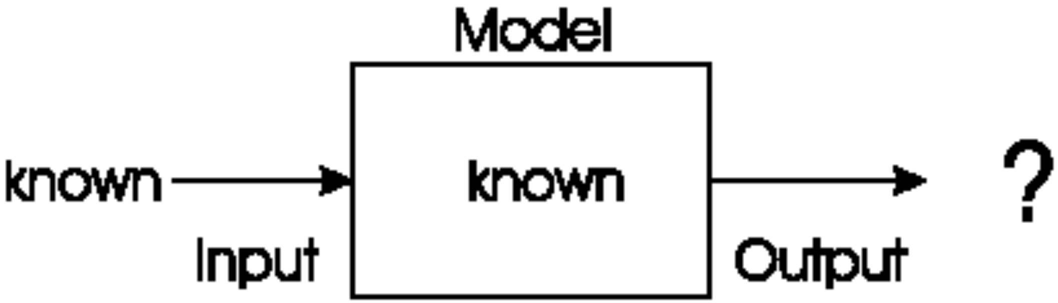
We have corresponding sets of inputs & outputs and seek model that delivers correct output for every known input



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

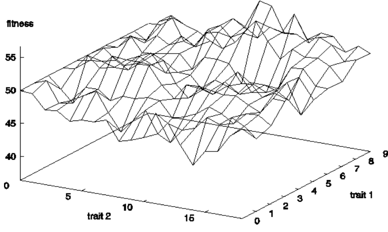
Recap of EC metaphor

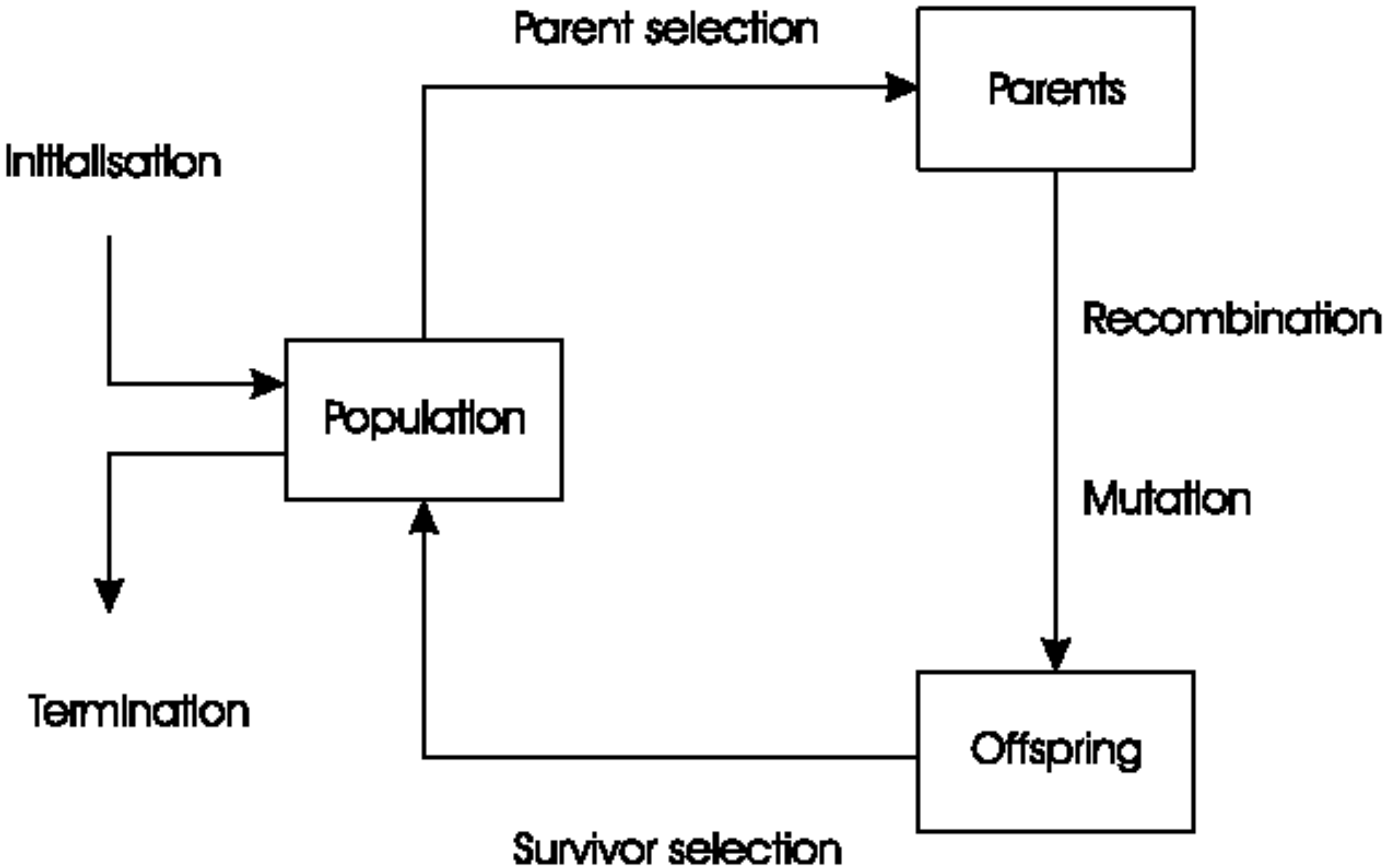
- A population of individuals exists in an environment with limited resources
- ***Competition*** for those resources causes selection of those ***fitter*** individuals that are better adapted to the environment
- These individuals act as seeds for the generation of new individuals through recombination 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



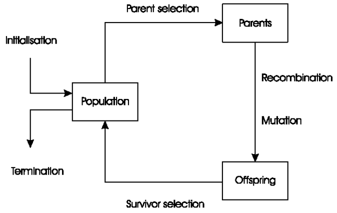


Pseudo-code for typical EA

```
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

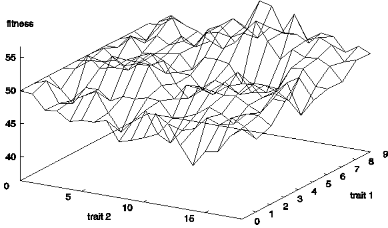


Representations

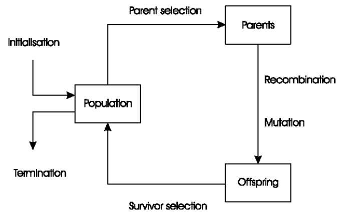
- Candidate solutions (individuals) exist in *phenotype* space
- They are encoded in chromosomes, which exist in *genotype* space
  - Encoding : phenotype => genotype (not necessarily one to one)
  - Decoding : genotype => phenotype (must be one to one)
- Chromosomes contain genes, which are in (usually 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

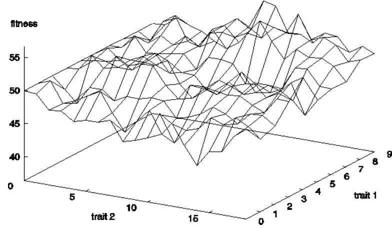
Evaluation (Fitness) Function



- 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



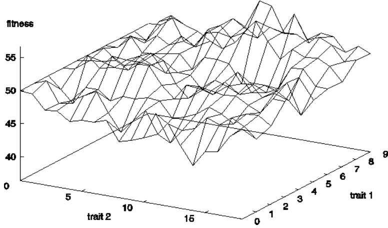
Population



- 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)

Quick Question
"Small" or "large" populations?

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

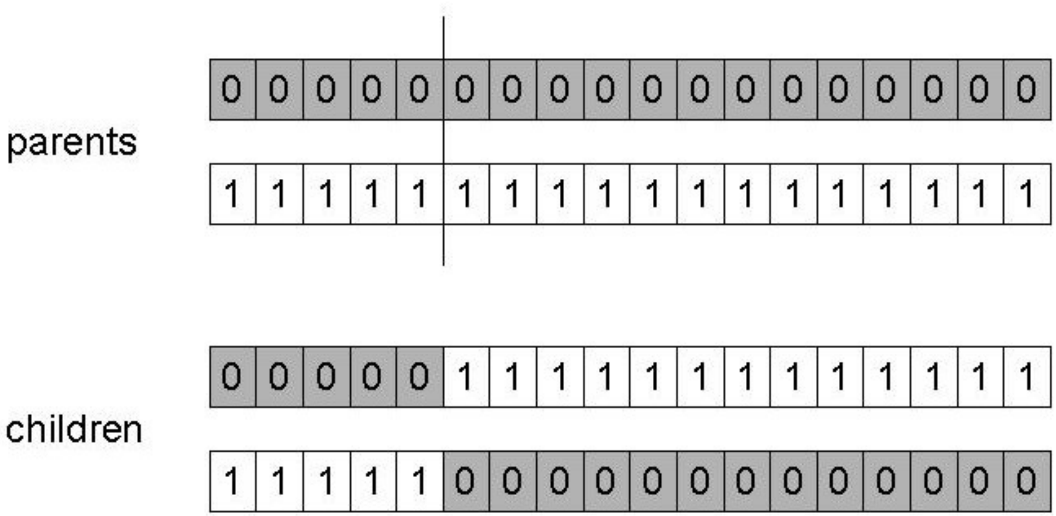
parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

Quick Question

How about TSP tours?

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



Quick Question
How about TSP tours?

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

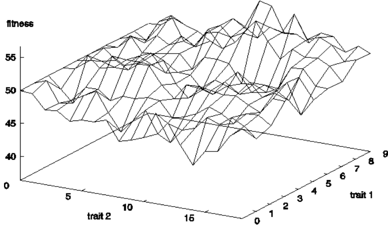
Quick Question

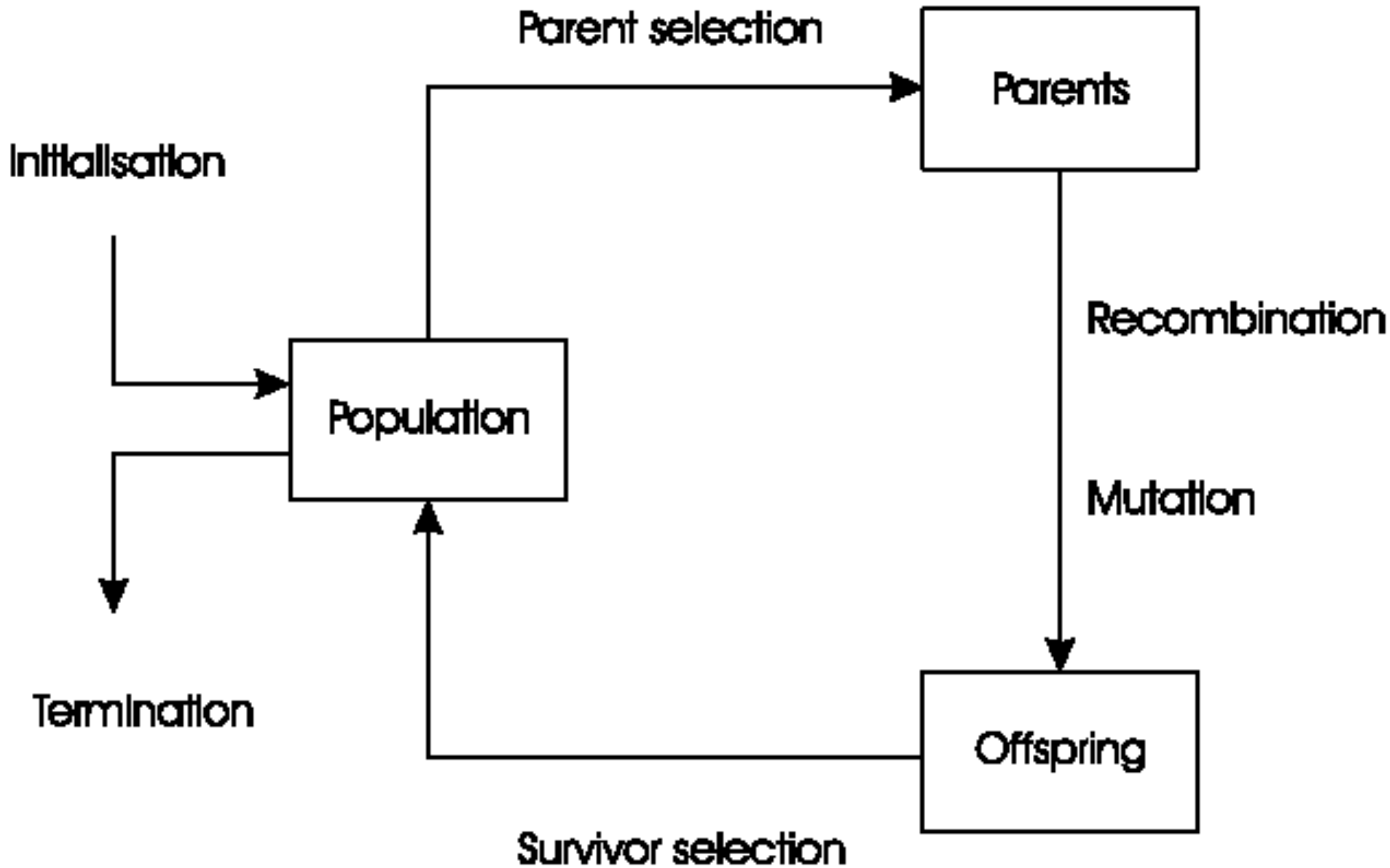
Seeding: what are disadvantages?

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

General Scheme of EAs





Gusz Eiben, Jim Smith, From evolutionary computation to the evolution of things (article in Nature 2015)
"... From the perspective of the underlying substrate in which the evolution takes place, the emergence of evolutionary computation can be considered as a major transition of the evolutionary principles from 'wetware', the realm of biology, to software, the realm of computers. Today the field is at an exciting stage. New developments in robotics and rapid prototyping (3D printing) are paving the way towards a second major transition: from software to hardware, going from digital evolutionary systems to physical ones. ..."
http://www.nature.com/nature/journal/v521/n7553/full/nature14544.html

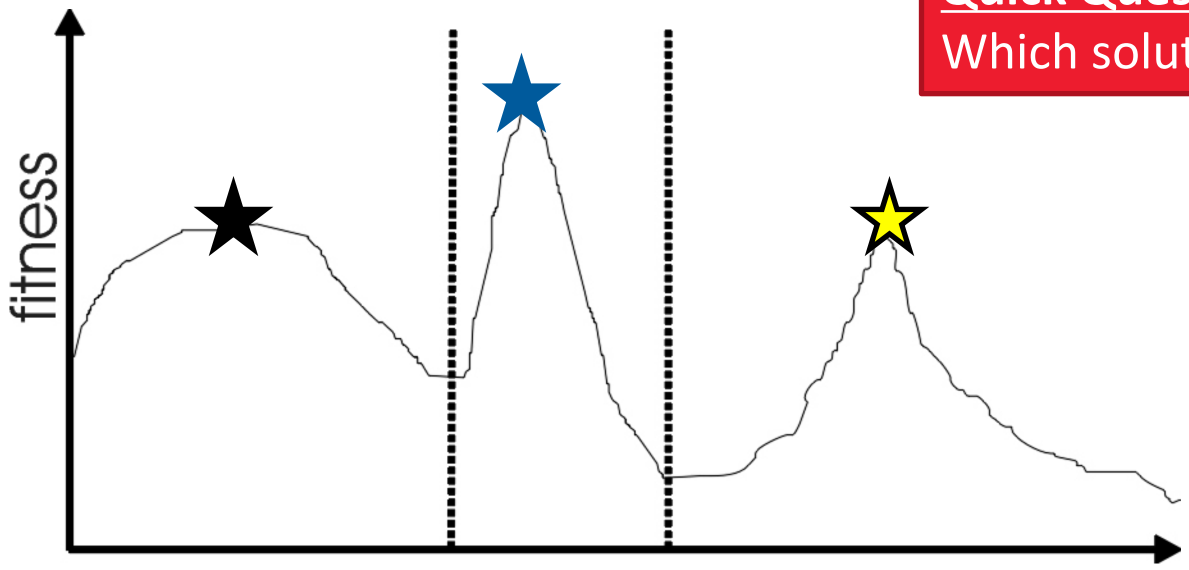
Multimodal Problems, Multi-Objective Optimization

Eiben/Smith Chapter 9

Multimodal Problems

Motivation 1: Multimodality

Most interesting problems have more than one locally optimal solution.



Quick Question

Which solution is best?

Motivation 2: Genetic Drift

- Finite population with global (panmictic) mixing and selection eventually convergences around one optimum
- Often might want to identify several possible peaks
- This can aid global optimisation when sub-optima has the largest basin of attraction

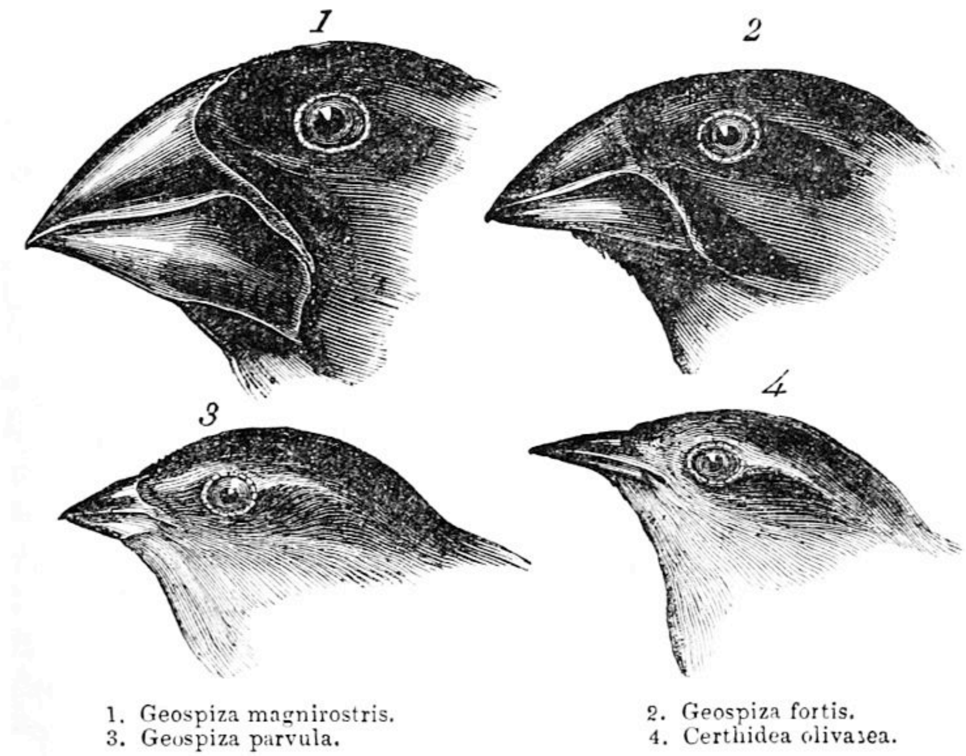
Biological Motivation 1: Speciation

- In nature different species adapt to occupy different environmental niches, which contain finite resources, so the individuals are in competition with each other
- Species only reproduce with other members of the same species (Mating Restriction)
- These forces tend to lead to phenotypic homogeneity within species, but differences between species

Biological Motivation 1: Speciation

"Darwin's finches" (19th century)

A group of ~15 species, only found on the Galapagos Islands



Biological Motivation 2: Punctuated Equilbria

- Theory that periods of stasis are interrupted by rapid growth when main population is “invaded” by individuals from previously spatially isolated group of individuals from the same species
- The separated sub-populations (demes) often show local adaptations in response to slight changes in their local environments

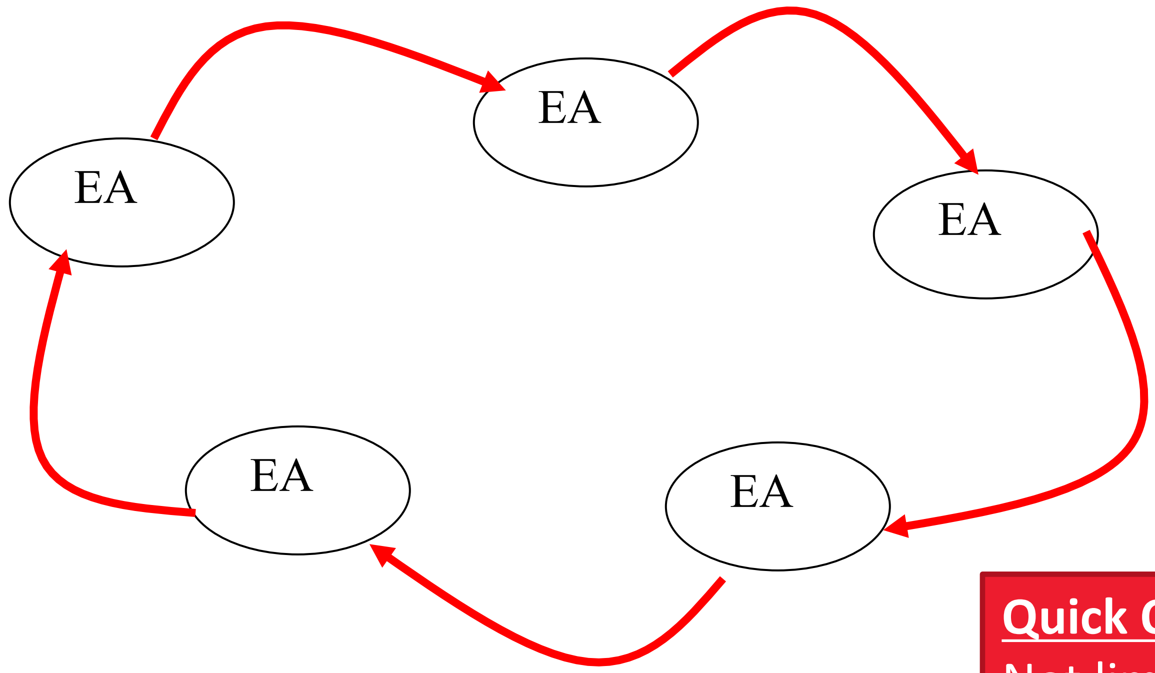
Additional reading:
Did wolf no. 93 (temporarily) save the wolf population on Isle Royale? (1997)
http://en.wikipedia.org/wiki/Wolves_and_moose_on_Isle_Royale
2011: “The wolf population on Isle Royale is small, averaging only about 23 wolves. By the end of his eight years of breeding, he produced 34 pups, those had produced an additional 45 pups.”
2016: “[The] wolf population is now nearly extinct with only two severely inbred wolves present.”

Implications for Evolutionary Optimisation

Two main approaches to diversity maintenance:

- Implicit approaches
  - Impose an equivalent of geographical separation
  - Impose an equivalent of speciation
- Explicit approaches
  - Make similar individuals compete for resources (fitness)
  - Make similar individuals compete with each other for survival

Implicit 1: “Island” Model Parallel EAs



Quick Question
Not limited to EAs…

Periodic migration of individual solutions between populations

Island Model EAs contd:

- Run multiple populations in parallel, in some kind of communication structure (usually a ring or a torus).
- After a (usually fixed) number of generations (an ***epoch***),
  exchange individuals with neighbours
- Repeat until ending criteria met
- Partially inspired by parallel/clustered systems

Island Model Parameters 1

- Could use different operators in each island
- How often to exchange individuals ?
  - too quick and all pops converge to same solution
  - too slow and waste time
  - most authors use range~ 25-150 gens
  - can do it adaptively (stop each EA when there is no improvement for (say) 25 generations)

Island Model Parameters 2

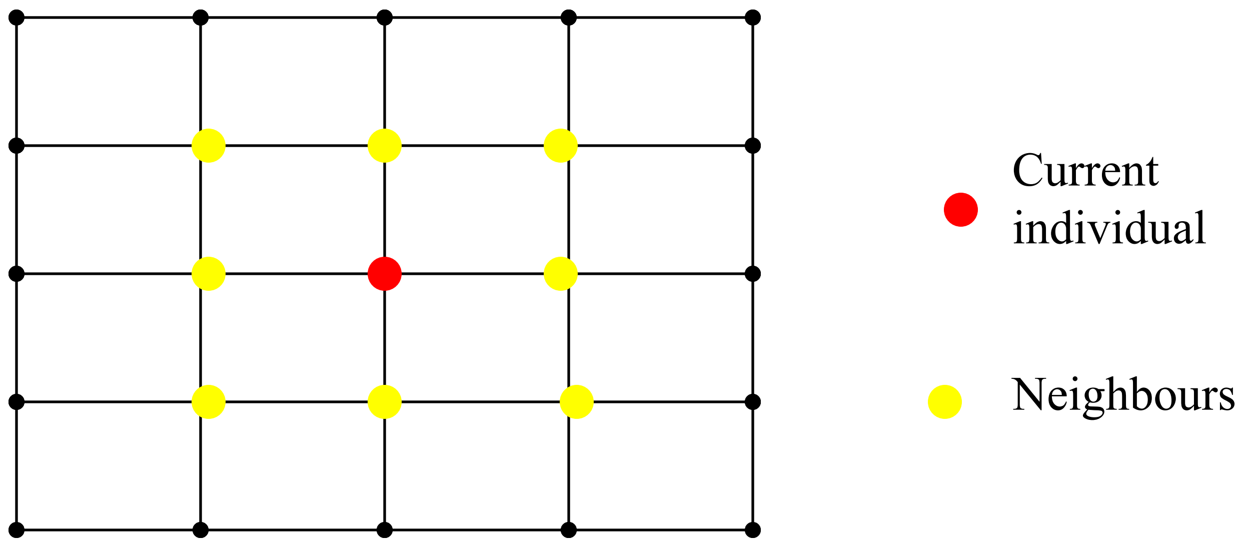
- How many, which individuals to exchange ?
  - usually ~2-5, but depends on population size.
  - more sub-populations usually gives better results but there can be a "critical mass", i.e., minimum size of each sub population needed
  - can select random/worst individuals to replace

Additional reading:

Dirk Sudholt's theoretical results on migration strategies (look for "migration" on http://dblp.uni-trier.de/pers/hd/s/Sudholt:Dirk, e.g. his 2015 book chapter http://staffwww.dcs.shef.ac.uk/people/D.Sudholt/parallel-eas.pdf)

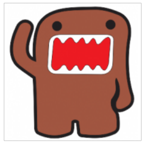
Implicit 2:
Diffusion Model Parallel EAs

- Impose spatial structure (usually grid) in 1 population



Diffusion Model EAs

- Consider each individual to exist on a point on a (usually rectangular toroid) grid
- Selection (hence recombination) and replacement happen using concept of a neighbourhood a.k.a. ***deme***



- Leads to different parts of grid searching different parts of space, good solutions diffuse across grid over a number of gens

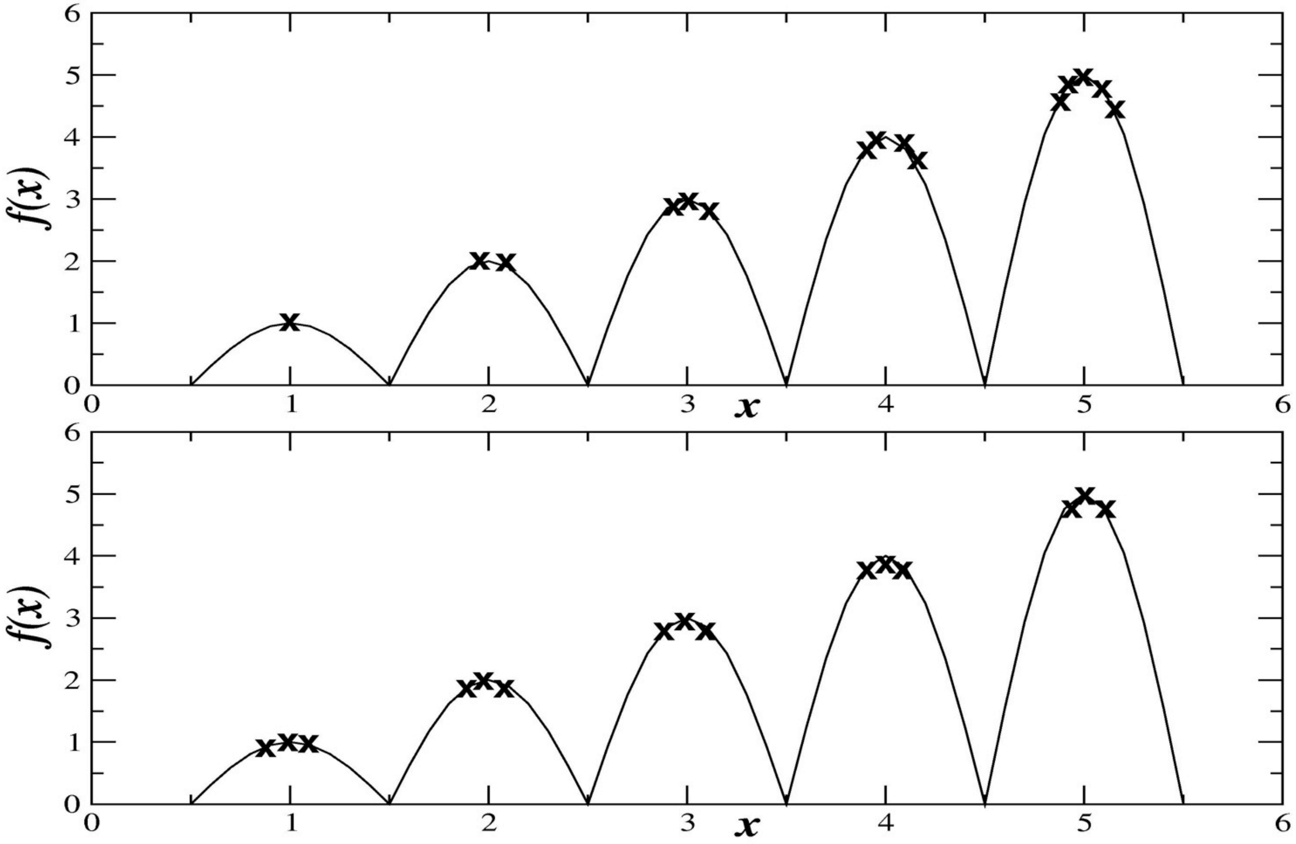
Diffusion Model Example

- Assume rectangular grid so each individual has 8 immediate neighbours
- equivalent of 1 generation is:
  - pick point in pop at random
  - pick one of its neighbours using roulette wheel
  - crossover to produce 1 child, mutate
  - replace individual if fitter
  - circle through population until done

Implicit 3: Automatic Speciation

- Either only mate with genotypically/phenotypically similar members or
- Add bits to problem representation
  - that are initially randomly set
  - subject to recombination and mutation
  - when selecting partner for recombination, only pick members with a good match
  - can also use tags to perform fitness sharing (see later) to try and distribute members amongst niches

Explicit Methods: Fitness Sharing vs. Crowding



Explicit 1: Fitness Sharing

- Restricts the number of individuals within a given niche by "sharing" their fitness, so as to allocate individuals to niches in proportion to the niche fitness
- need to set the size of the niche σ share in either genotype or phenotype space
- run EA as normal but after each gen set

$$f'(i) = \frac{f(i)}{\sum_{j=1}^{\mu} sh(d(i,j))}$$

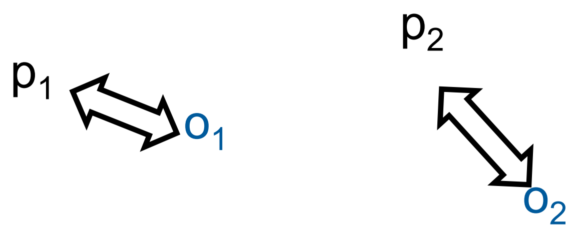
$$sh(d) = \begin{cases} 1-(d/\sigma)^{\alpha} & d < \sigma \\ 0 & otherwise \end{cases}$$

Nice idea, but difficult to master (for more details see http://www.cs.bham.ac.uk/~pkl/teaching/2009/ec/lecture_notes/l06-niching.pdf)

Explicit 2: Crowding

- Attempts to distribute individuals evenly amongst niches (new individuals replace similar existing ones)
- relies on the assumption that offspring will tend to be close to parents
- uses a distance metric in phenotype/genotype space
- randomly shuffle and pair parents, produce 2 offspring, then

  2 parent/offspring tournaments - pair so that
  $d(p_1,o_1)+d(p_2,o_2) < d(p_1,o_2) + d(p_2,o_1)$
  (subscript: the tournament number, e.g. p_1 is participates in tournament 1)



Multi-Objective Optimisation

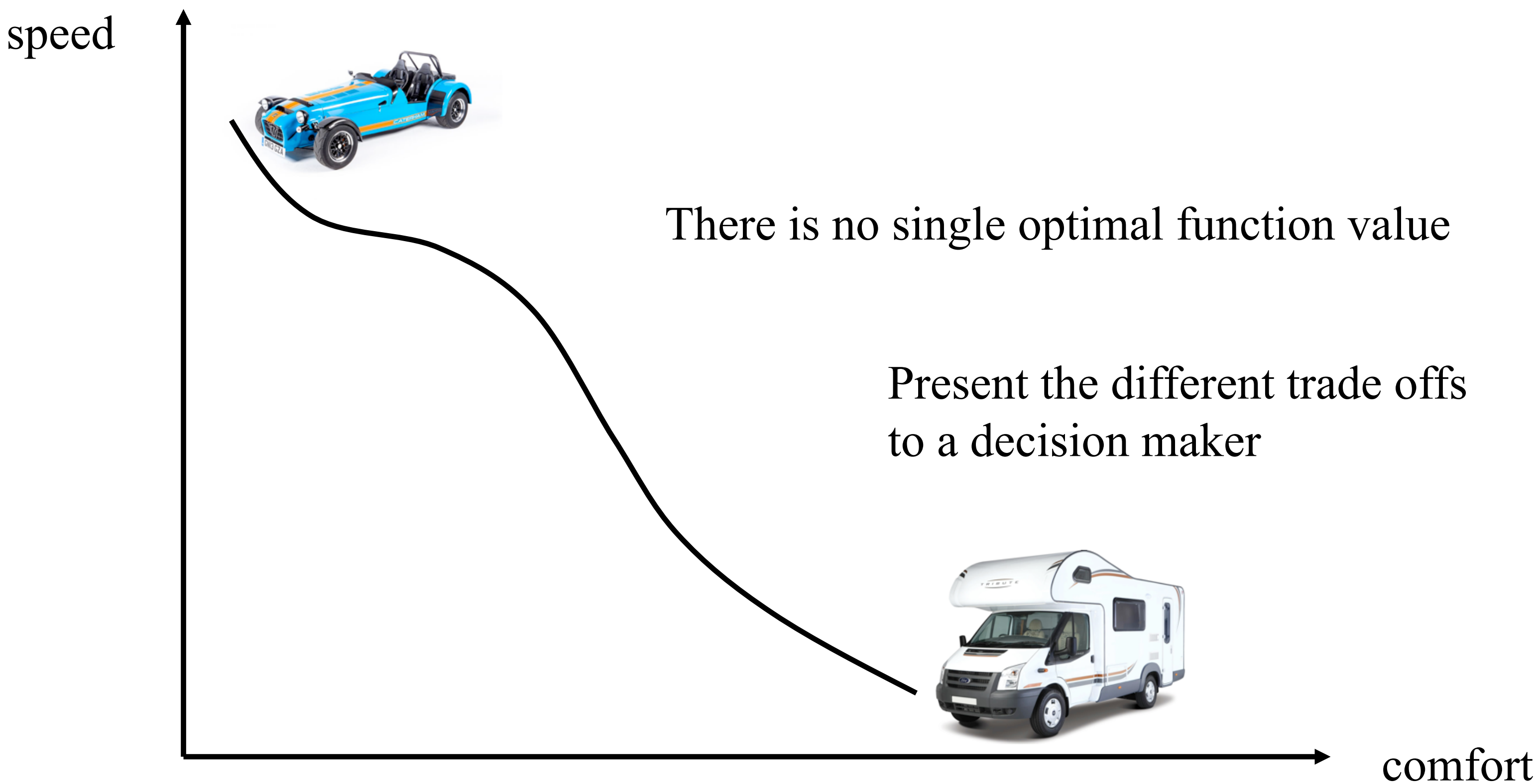
Multi-Objective Problems (MOPs)

- Wide range of problems can be categorised by the presence of a number of n possibly conflicting objectives:
  - buying a car: speed vs. price vs. reliability
  - engineering design: lightness vs strength
- Two part problem:
  - finding set of good solutions
  - choice of best for particular application

Multi-Objective Optimisation

Many problems have more than one goal function

Example: Buying a new car



There is no single optimal function value

Present the different trade offs
to a decision maker

MOPs 1: Conventional approaches

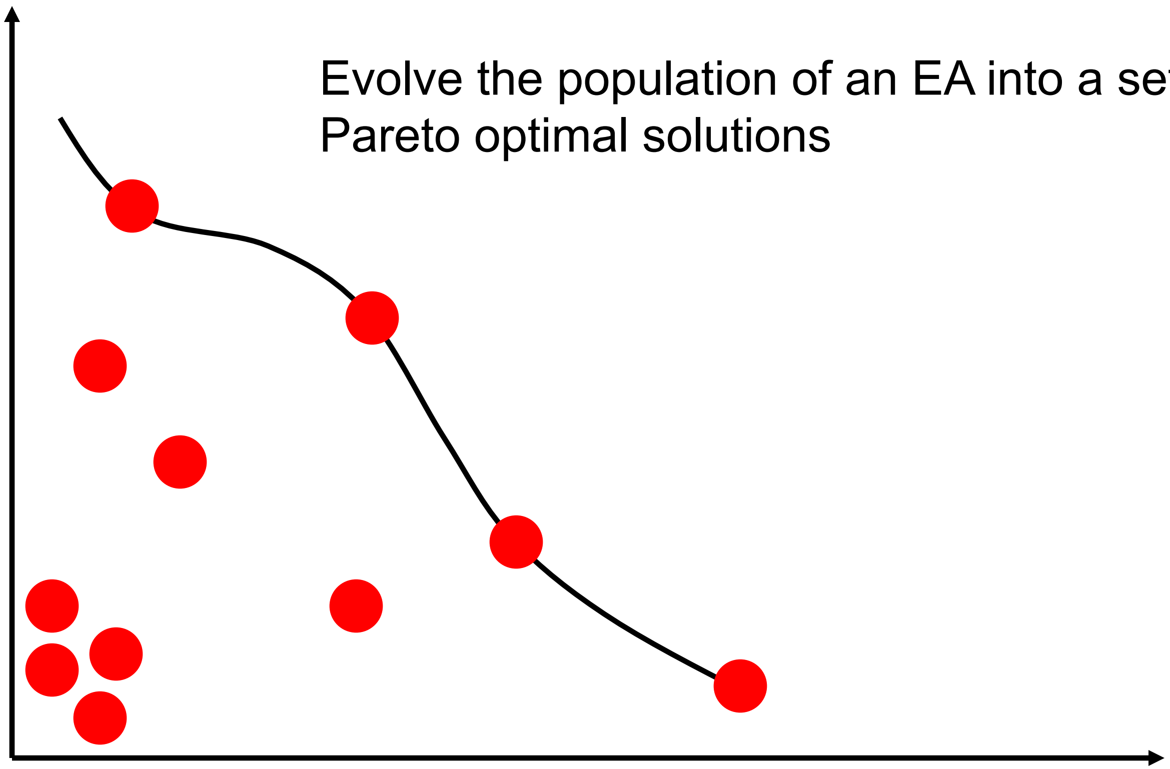
rely on using a weighting of objective function values to give a single scalar objective function which can then be optimised

$$f'(x) = \sum_{i=1}^{n} w_i f_i(x)$$

to find other solutions have to re-optimise with different w_i.

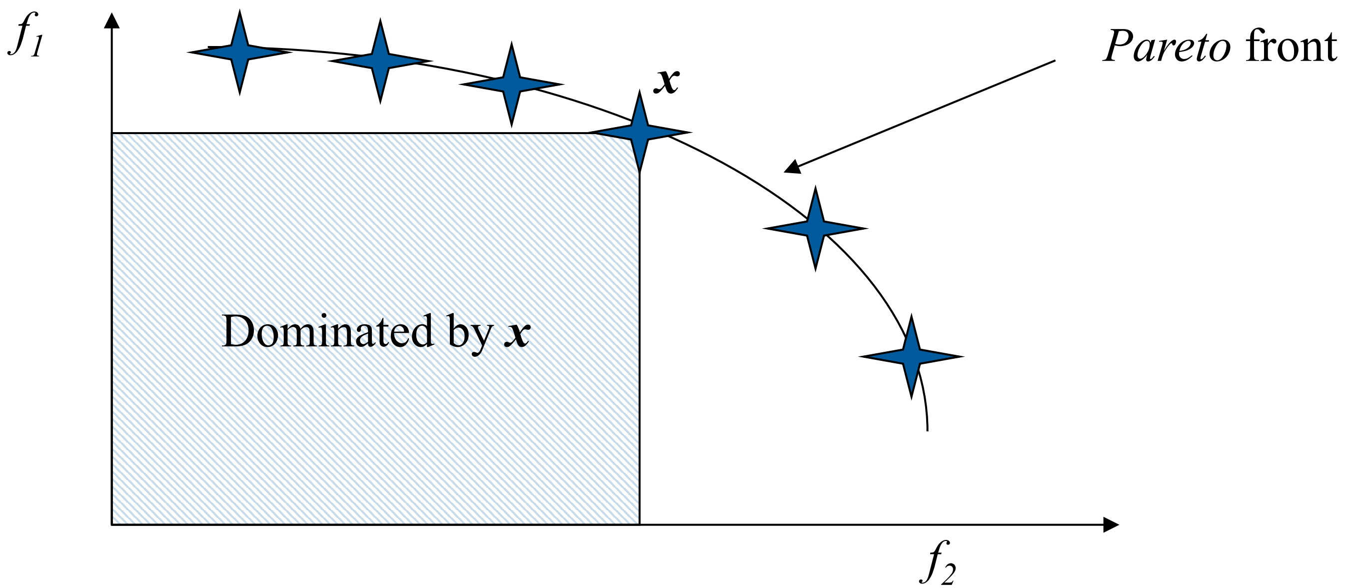
Evolutionary Multi-Objective Optimization

Try to compute/approximate the Pareto front by EAs



MOPs 2: Dominance

we say *x* dominates *y* if it is at least as good on all criteria and ***better*** on at least one



Multi-Objective Optimisation

$$f : X^n \rightarrow R^m$$

Dominance in the objective space

u *weakly dominates* v $(u \succeq v)$ iff $u_i \geq v_i$ for all $i \in \{1, \ldots, m\}$

u *dominates* v $(u \succ v)$ iff $u \succeq v$ and $u \neq v$.

Non-dominated objective vectors constitute the Pareto front

Classical goal:
Compute for each Pareto optimal objective vector
a corresponding solution

Single-Objective vs. Multi-Objective Optimization

General assumption

- Multi-objective optimization is more (as least as) difficult as single-objective optimization.
- True, if criteria to be optimized are independent.

Examples

- Minimum Spanning Tree Problem (MST) (in P).
- MST with at least 2 weight functions (NP-hard).
- Shortest paths (SP) (in P).
- SP with at least 2 weight functions (NP-hard).

MOPs 3: Advantages of EC approach

- Population-based nature of search means you can *simultaneously* search for a set of points approximating Pareto front
- Don't have to make guesses about which combinations of weights might be useful
- Makes no assumptions about shape of Pareto front, can be convex/discontinuous etc.

The fast non-dominated sorting algorithm (NSGA-II)

Non-dominated Sorting

Idea

- Search points that are non-dominated are really good.
- Search points that are just dominated by a few other search points are not that bad.
- Search points that are dominated by many other search points are really bad.

Procedure

Rank that individuals in a population according to the number of individuals that dominate it.

Non-dominated Sorting

Assume minimization of fitness function

$$f : X^n \to R^m$$

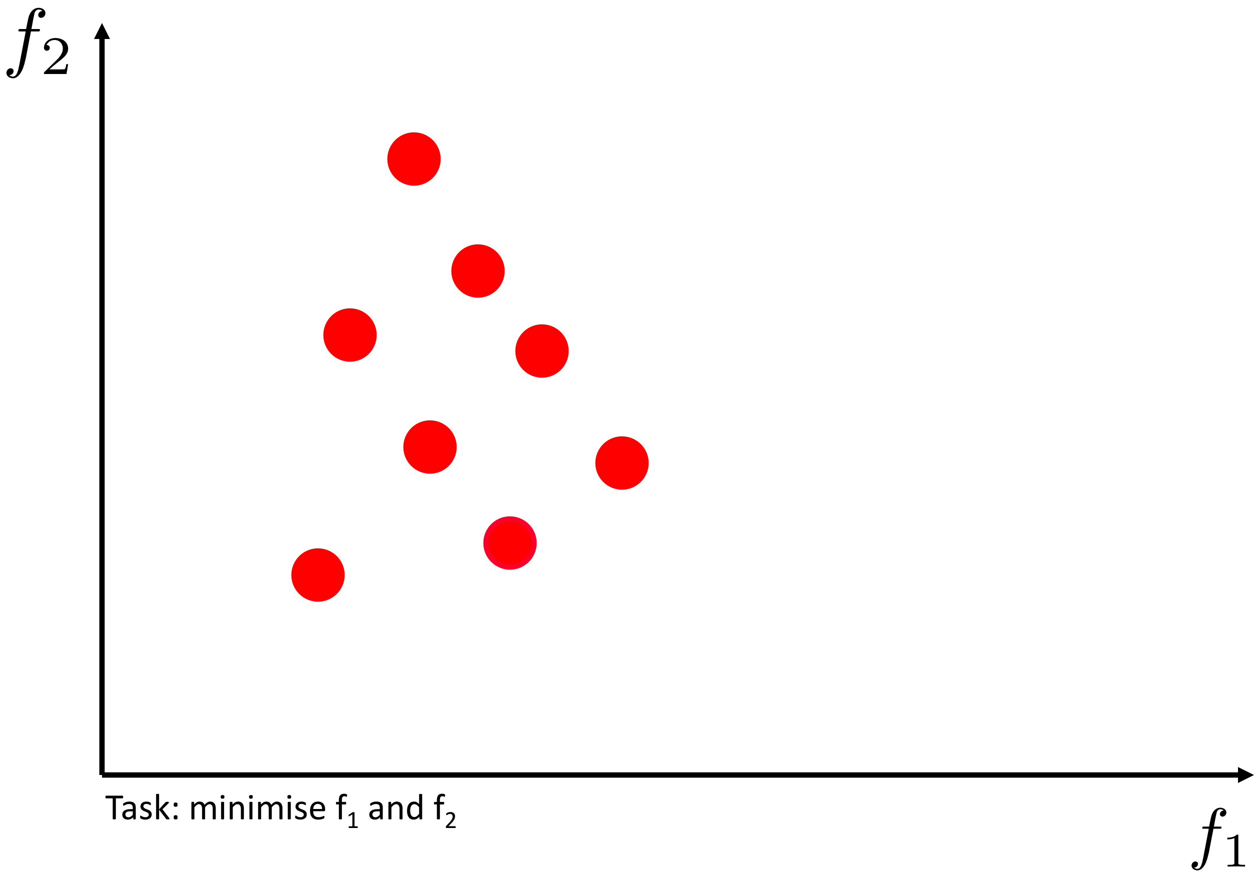
Dominance in objective space

$$x \preceq_{Par} y :\Leftrightarrow x_i \leq y_i \text{ for } 1 \leq i \leq m.$$

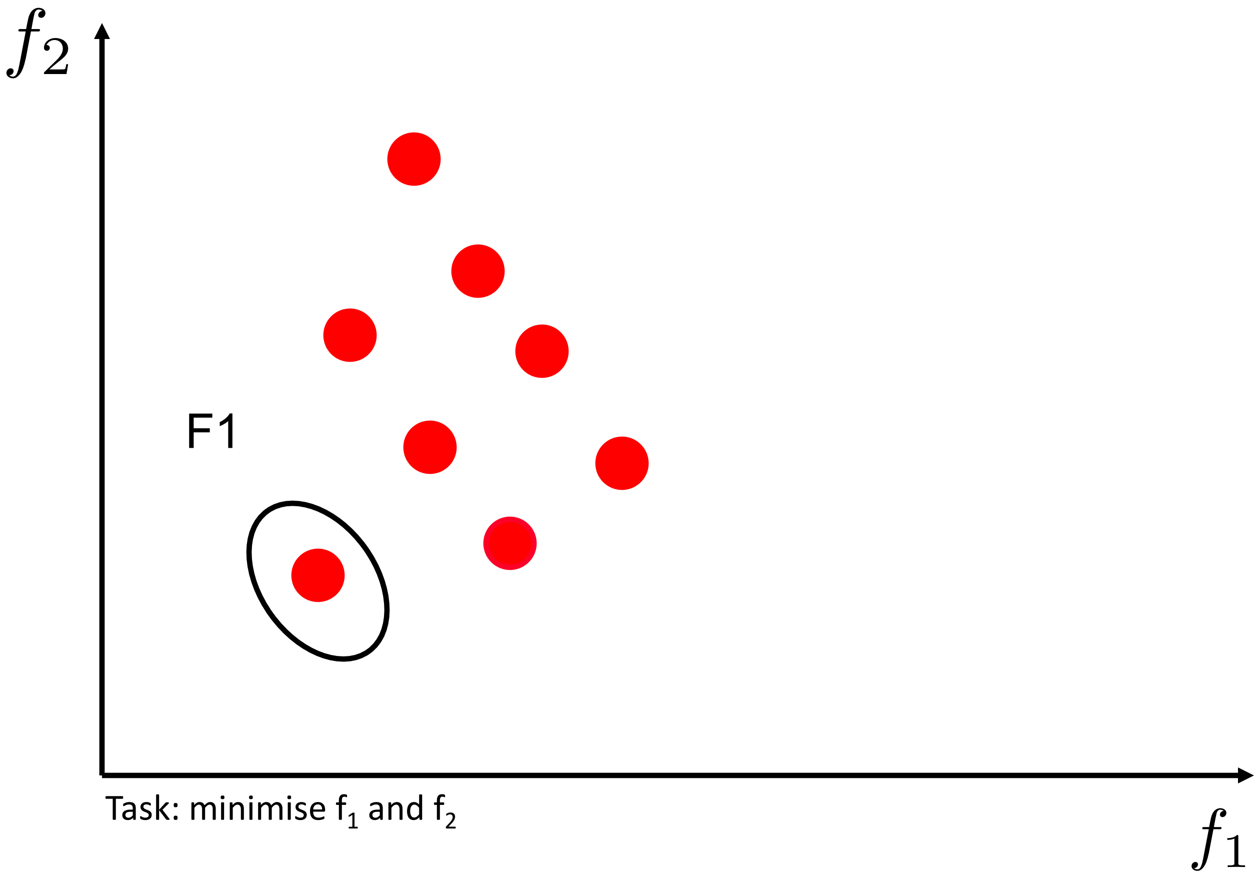
Dominance in search space

$$a \preceq_{Par} b :\Leftrightarrow f(a) \preceq_{Par} f(b).$$

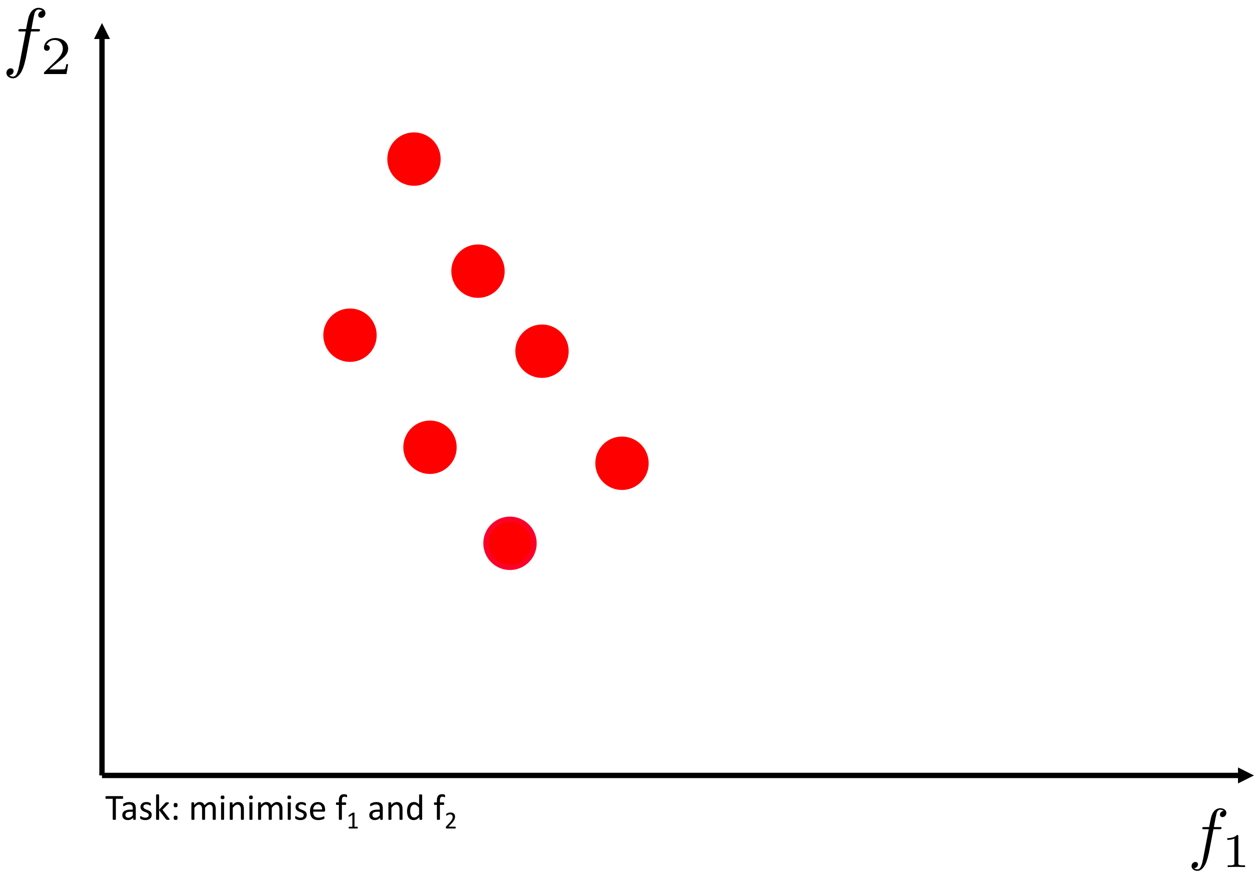
Non-dominated Sorting



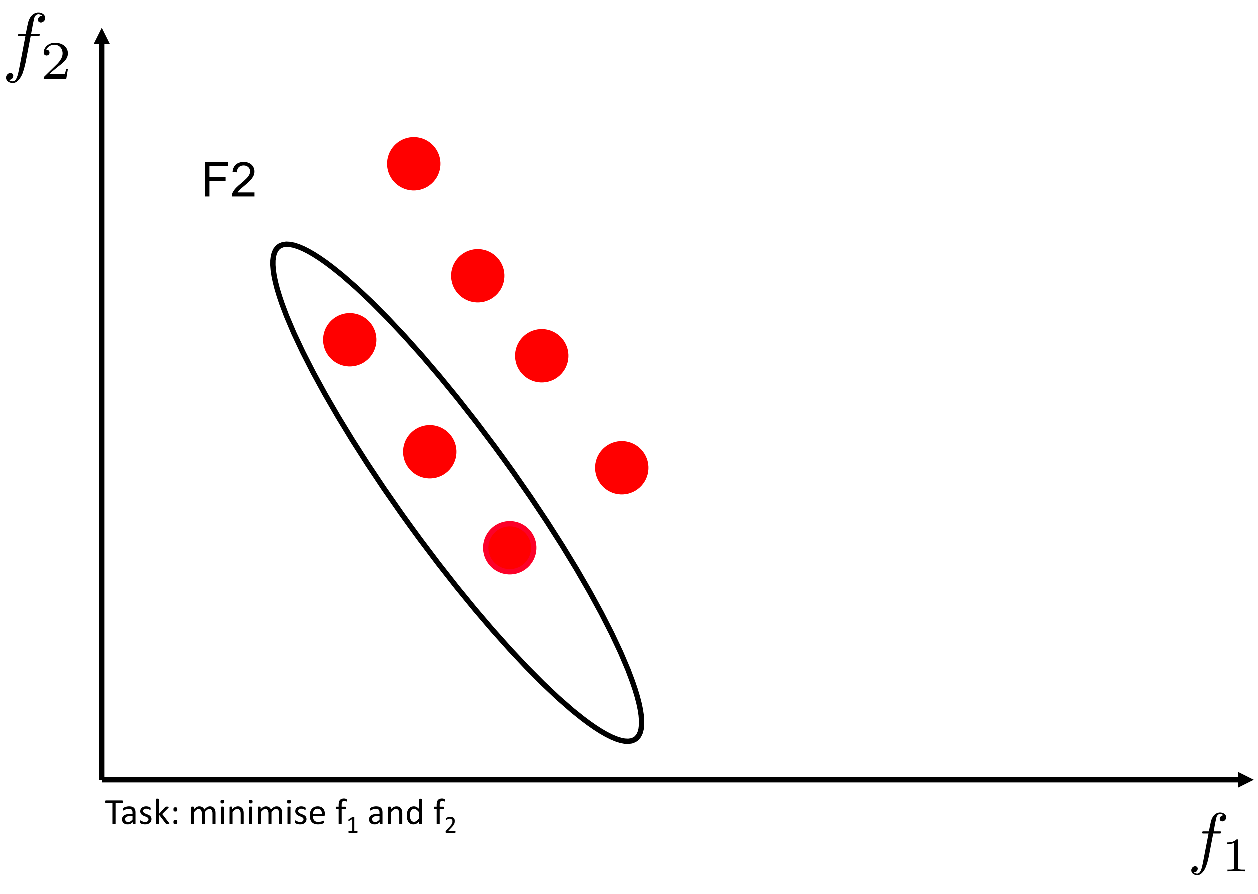
Non-dominated Sorting



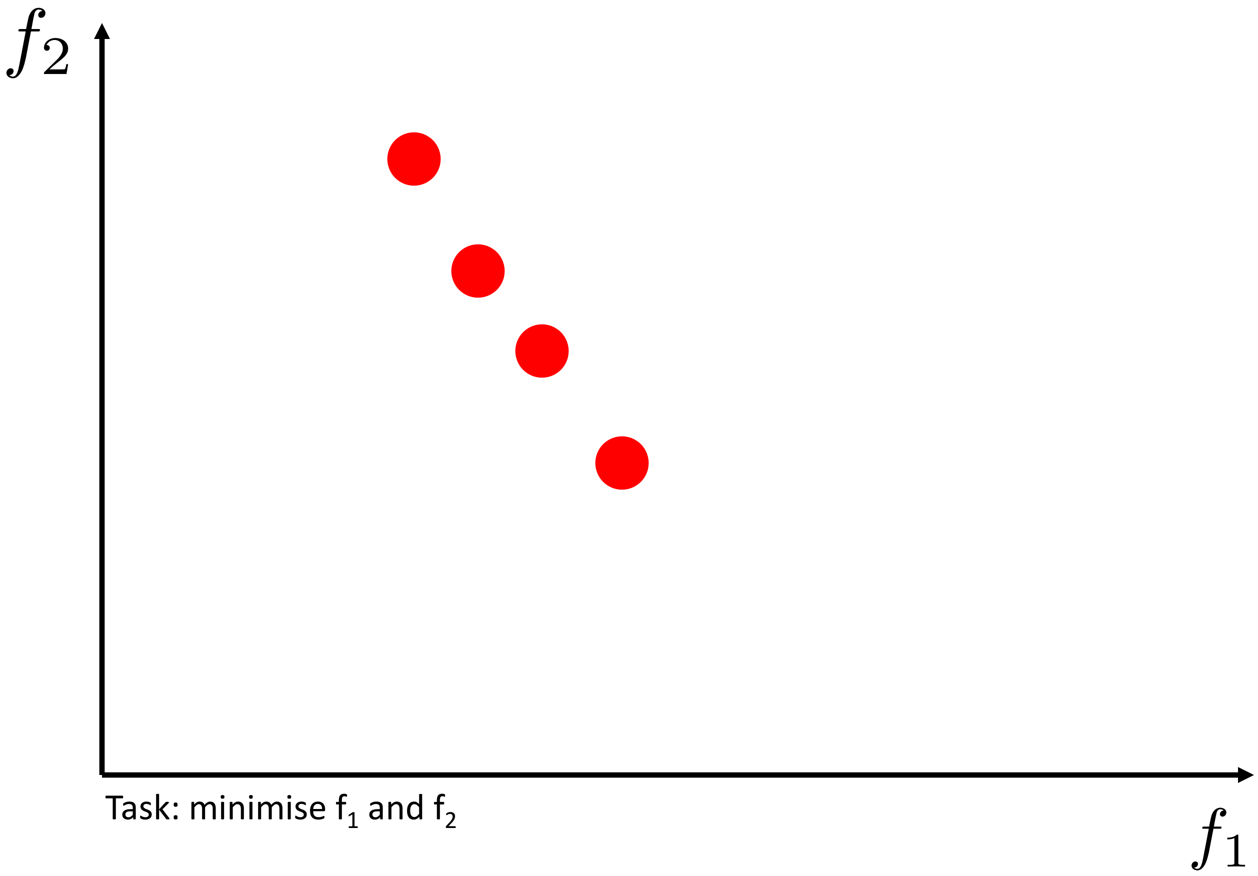
Non-dominated Sorting



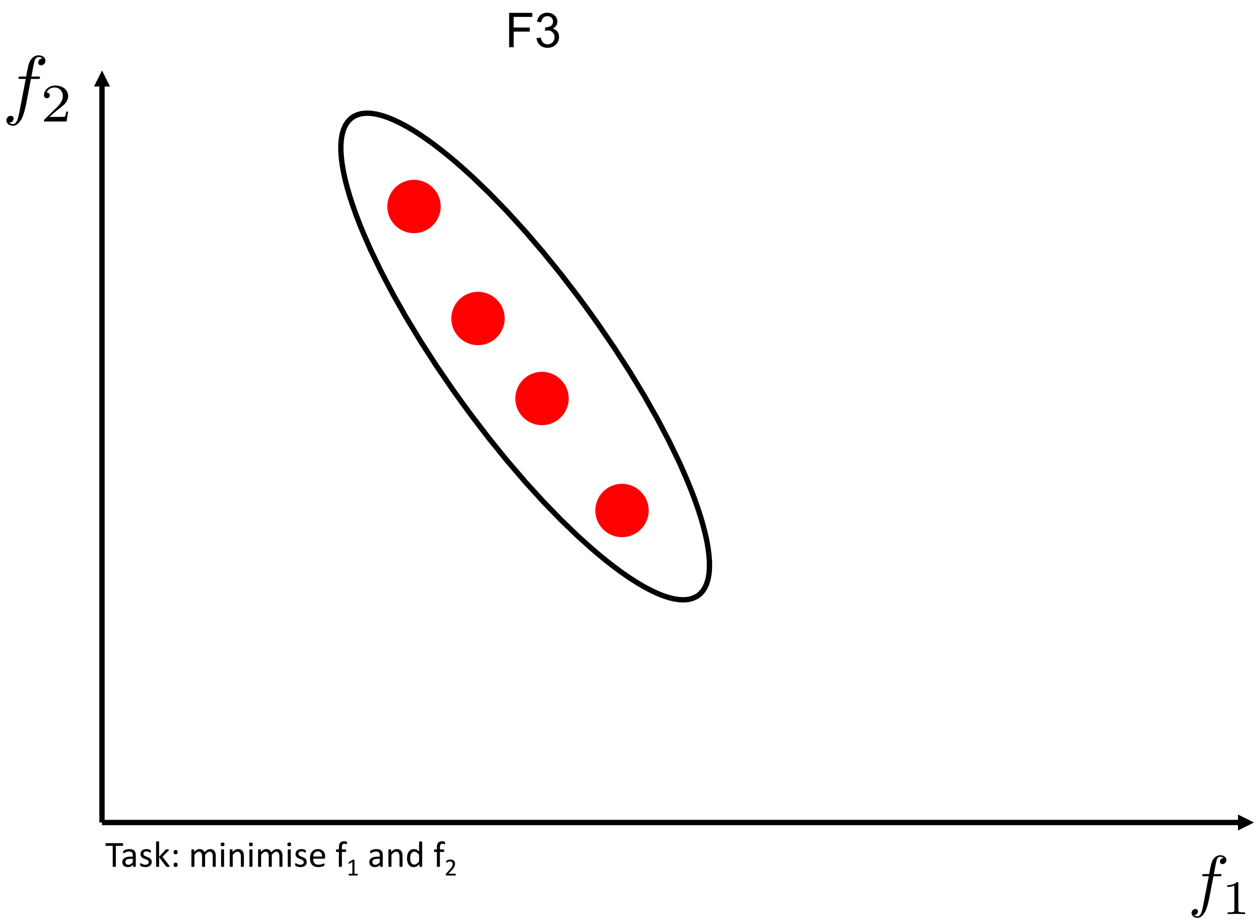
Non-dominated Sorting



Non-dominated Sorting



Non-dominated Sorting



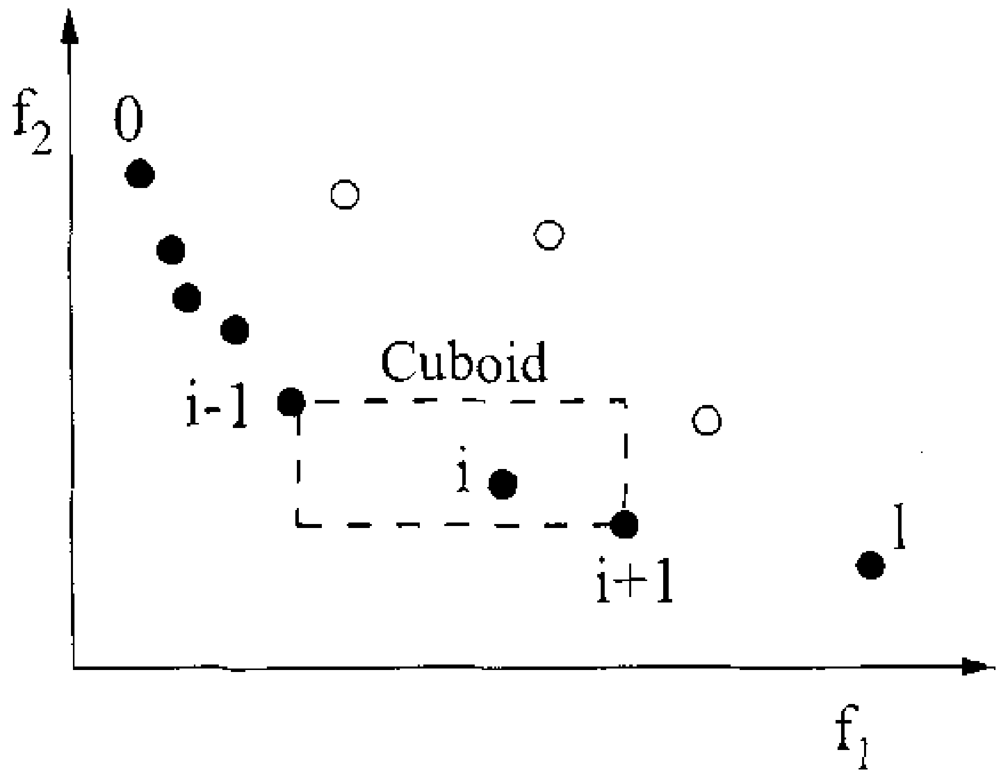
Fast non-dominated Sorting

```
fast-non-dominated-sort(P)
```

for each $p \in P$	
$\quad S_p = \emptyset$	
$\quad n_p = 0$	
$\quad$ for each $q \in P$	
$\quad\quad$ if $(p \prec q)$ then	If p dominates q
$\quad\quad\quad S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
$\quad\quad$ else if $(q \prec p)$ then	
$\quad\quad\quad n_p = n_p + 1$	Increment the domination counter of p
$\quad$ if $n_p = 0$ then	p belongs to the first front
$\quad\quad p_{\text{rank}} = 1$	
$\quad\quad \mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$	
$i = 1$	Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$	
$\quad Q = \emptyset$	Used to store the members of the next front
$\quad$ for each $p \in \mathcal{F}_i$	
$\quad\quad$ for each $q \in S_p$	
$\quad\quad\quad n_q = n_q - 1$	
$\quad\quad\quad$ if $n_q = 0$ then	q belongs to the next front
$\quad\quad\quad\quad q_{\text{rank}} = i + 1$	
$\quad\quad\quad\quad Q = Q \cup \{q\}$	
$\quad i = i + 1$	
$\quad \mathcal{F}_i = Q$	

Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

Crowding distance



Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

Crowding distance assignment

$\texttt{crowding-distance-assignment}(\mathcal{I})$	
$l = \vert\mathcal{I}\vert$	number of solutions in $\mathcal{I}$
for each i, set $\mathcal{I}[i]_{\text{distance}} = 0$	initialize distance
for each objective m	
$\quad \mathcal{I} = \text{sort}(\mathcal{I}, m)$	sort using each objective value
$\quad \mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$	so that boundary points are always selected
$\quad$ for $i = 2$ to $(l-1)$	for all other points
$\quad\quad \mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)/(f_m^{\max} - f_m^{\min})$	

Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

Crowded Comparison Operator

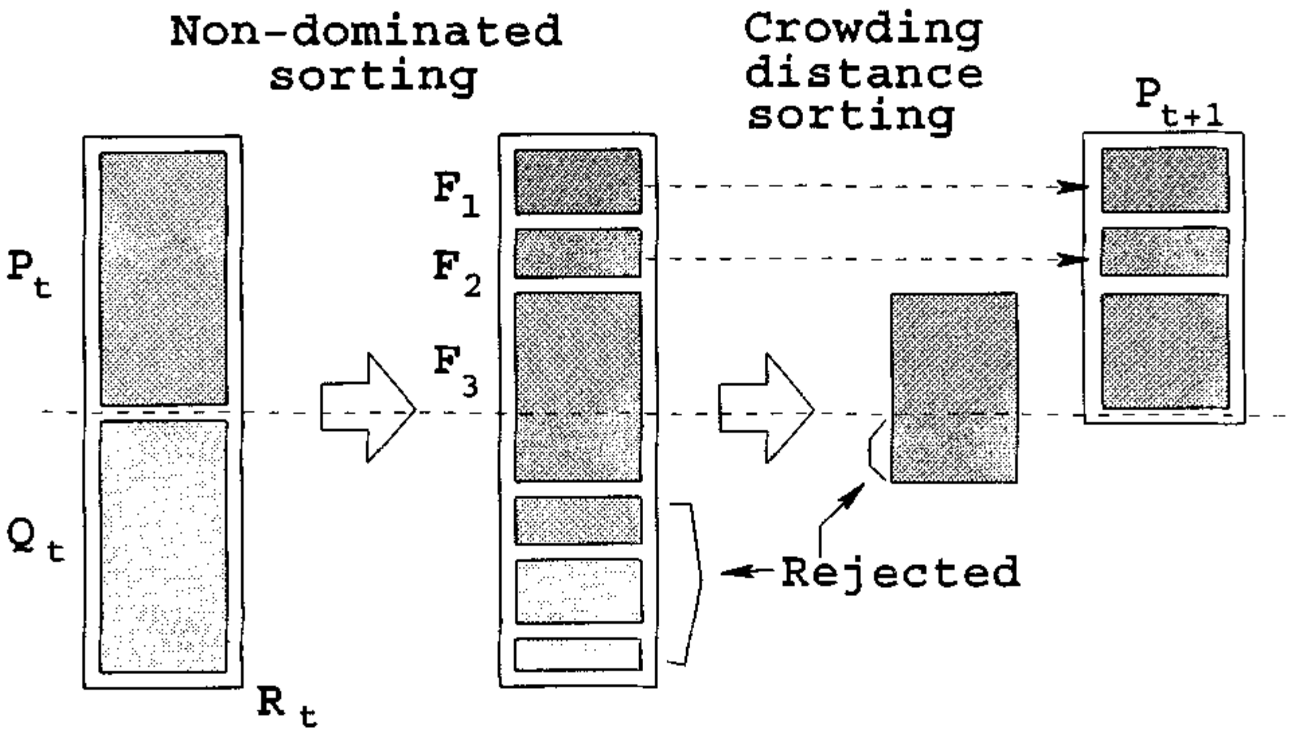
Crowded comparison operator $(\prec_n)$

1) nondomination rank (i_{rank});
2) crowding distance (i_{distance}).

We now define a partial order $\prec_n$ as

$$
\begin{aligned}
& i \prec_n j \quad \text{if } (i_{\text{rank}} < j_{\text{rank}}) \\
& \text{or } ((i_{\text{rank}} = j_{\text{rank}}) \\
& \text{and } (i_{\text{distance}} > j_{\text{distance}}))
\end{aligned}
$$

Selection



Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

1 Iteration for NSGA-II

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots),$ all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $\|P_{t+1}\| + \|\mathcal{F}_i\| \leq N$	until the parent population is filled
$\texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
Sort$(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \|P_{t+1}\|)]$	choose the first $(N - \|P_{t+1}\|)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

1 Iteration for NSGA-II

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $\lvert P_{t+1}\rvert + \lvert\mathcal{F}_i\rvert \leq N$	until the parent population is filled
~~$\texttt{crowding-distance-assignment}(\mathcal{F}_i)$~~	~~calculate crowding distance in $\mathcal{F}_i$~~
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
$\text{Sort}(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1}\rvert)]$	choose the first $(N - \lvert P_{t+1}\rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

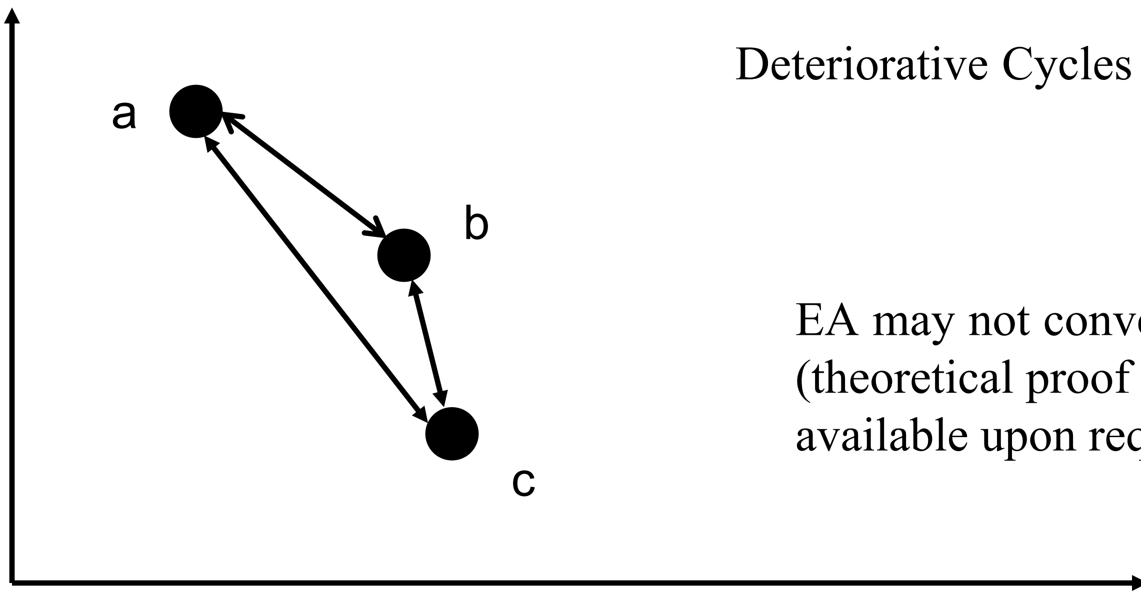
Compute crowding distance as part of this step

Source: Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T. Meyarivan:
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Trans. Evolutionary Computation 6(2): 182-197 (2002)

Deteriorative Cycles

Dominance relation on sets is not total.

We may move between solutions if they are incomparable



Deteriorative Cycles

EA may not converge!!!
(theoretical proof are
available upon request)

The Hypervolume Indicator

Set-Based Multi-Objective Optimisation

We are interested in sets of search points

Extend dominance relation to sets (assume maximization)

Let $A, B \in 2^X$ then

$$A \succeq_{dom} B :\Leftrightarrow (\forall b \in B\ \exists a \in A \colon a \succeq_{Par} b).$$

Let $A, B \in 2^X$ and $\preceq$ be an arbitrary relation on 2^X. Then

$$A \succ B :\Leftrightarrow (A \succeq B) \wedge (B \not\succeq A)$$

Goal in multi-objective optimization:

We denote the set of maximal elements containing exactly μ elements of X by $Max_\mu(2^X, \succeq)$, i.e

$$Max_\mu(2^X, \succeq) := Max\{R \mid R \in 2^X \wedge |R| = \mu\}.$$

Unary Indicators

We want to assign to a set of search points (population) a value that determines the quality of this set

$$I_1 \colon 2^X \to \mathbb{R}.$$

For an unary indicators I_1 we set

$$A \succeq_{I_1} B :\Leftrightarrow I_1(A) \geq I_1(B),$$
$$A \succ_{I_1} B :\Leftrightarrow I_1(A) > I_1(B).$$

MOPs 4: Requirements of EC approach

- Way of assigning fitness,
  - usually based on dominance
- Preservation of diverse set of points
  - similarities to multi-modal problems
- Remembering all the non-dominated points you've seen
  - usually using elitism or an archive

MOPs 5: Fitness Assignment

- Could use aggregating approach and change weights during evolution
  - no guarantees
- Different parts of pop use different criteria
  - e.g. VEGA, but no guarantee of diversity
- Dominance
  - ranking or depth based
  - fitness related to whole population

MOPs 6: Diversity Maintenance

- Usually done by niching techniques such as:
  - fitness sharing
  - adding amount to fitness based on inverse distance to nearest neighbour (minimisation)
  - (adaptively) dividing search space into boxes and counting occupancy
- All rely on some distance metric in genotype / phenotype space

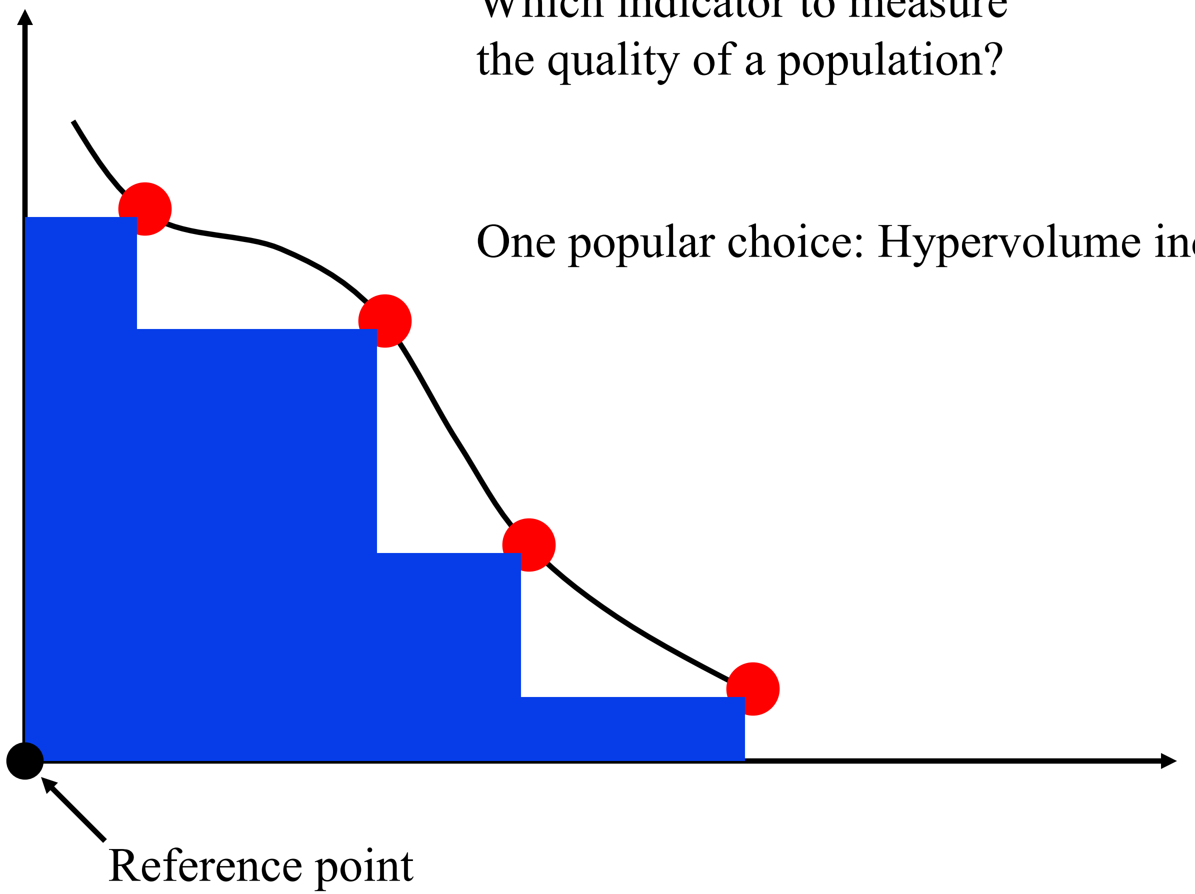
MOPs 7: Remembering Good Points

- Could just use elitist algorithm
  - e.g. ($\mu + \lambda$) replacement
- Common to maintain an archive of non-dominated points
  - some algorithms use this as second population that can be in recombination etc
  - others divide archive into regions too e.g. PAES

Set-Based Multi-Objective Optimisation

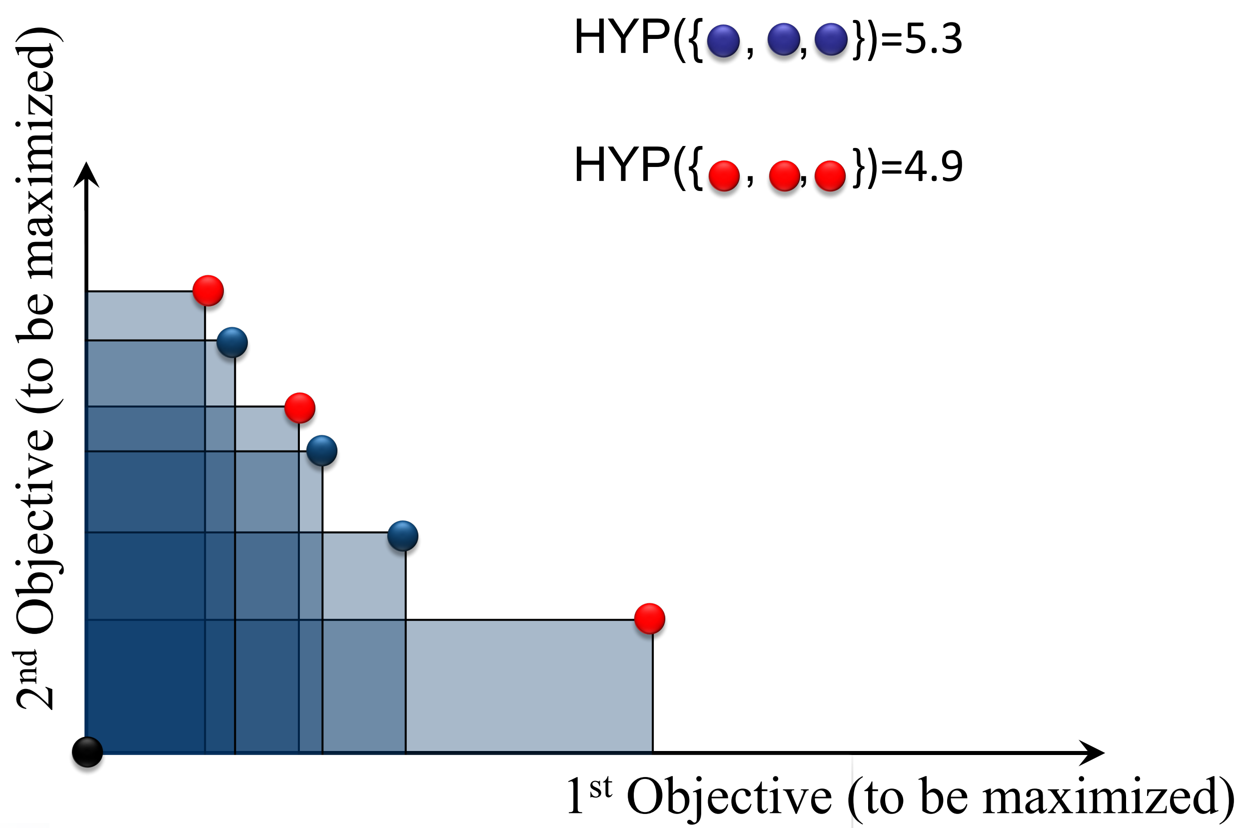
Which indicator to measure the quality of a population?

One popular choice: Hypervolume indicator



Hypervolume Indicator

Which population is better?



Hypervolume

Hypervolume of a set of points A with respect to a reference point

$$R = (R_1, R_2, \ldots, R_d) \in \mathbb{R}^d$$

is given by

$$I_{\mathrm{HYP}}^{R}(A) := \quad \mathrm{VOL}\left(\bigcup_{x \in A} [f_1(x), R_1] \times \ldots \times [f_d(x), R_d] \right)$$

Simple Indicator-based EA

Algorithm 1 Simple Indicator-Based Evolutionary Algorithm (SIBEA)

Given: population size μ; number of generations N

Step 1 (Initialization): Generate an initial set of decision vectors P of size μ; set the generation counter $m := 0$

Step 2 (Environmental Selection): Iterate the following three steps until the size of the population does no longer exceed μ:

1. Rank the population using dominance rank (number of dominating solutions) and determine the set of solutions $P' \subseteq P$ with the worst rank
2. For each solution $x \in P'$ determine the loss of hypervolume $d(x) = I_H(P') - I_H(P' \setminus \{x\})$ if it is removed from P'
3. Remove the solution with the smallest loss $d(x)$ from the population P (ties are broken randomly)

Step 3 (Termination): If $m \geq N$ then output P and stop; otherwise set $m := m + 1$.

Step 4 (Mating): Randomly select elements from P to form a temporary mating pool Q of size λ. Apply variation operators such as recombination and mutation to the mating pool Q which yields Q'. Set $P := P + Q'$ (multi-set union) and continue with Step 2.

Source: Dimo Brockhoff. Theoretical Aspects of Evolutionary Multiobjective Optimization—A Review

Hypervolume Indicator

Property of “strict Pareto compliance”:

- Consider two Pareto sets A and B:
- Hypervolume indicator values A higher than B if the Pareto set A dominates the Pareto set B

Input:	α	*(population size)*
	N	*(maximum number of generations)*
	κ	*(fitness scaling factor)*
Output:	A	*(Pareto set approximation)*

Step 1: ***Initialization:*** *Generate an initial population* P *of size* α*; set the generation counter* m *to* 0.

Step 2: ***Fitness assignment:*** *Calculate fitness values of individuals in* P*, i.e., for all* $\boldsymbol{x}^1 \in P$ *set* $F(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} -e^{-I(\{\boldsymbol{x}^2\},\{\boldsymbol{x}^1\})/\kappa}$.

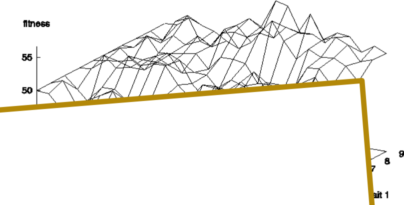
Step 3: ***Environmental selection:*** *Iterate the following three steps until the size of population* P *does not exceed* α*:*

1. *Choose an individual* $\boldsymbol{x}^* \in P$ *with the smallest fitness value, i.e.,* $F(\boldsymbol{x}^*) \leq F(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in P$.
2. *Remove* $\boldsymbol{x}^*$ *from the population.*
3. *Update the fitness values of the remaining individuals, i.e.,* $F(\boldsymbol{x}) = F(\boldsymbol{x}) + e^{-I(\{\boldsymbol{x}^*\},\{\boldsymbol{x}\})/\kappa}$ *for all* $\boldsymbol{x} \in P$.

Hypervolume Indicator

- Given: n axis-parallel boxes in d-dimensional space (boxes all have (0,...,0) as bottom corner)
- Task: Measure (volume) of their union
- Popular Algorithms:
  - HSO: $\mathcal{O}(n^d)$ [Zitzler'01, Knowles'02]
  - BR: $\mathcal{O}(n^{d/2} \log n)$ [Beume Rudolph'06]
  - Many (heuristical) improvements and specialized algorithms for small dimensions
  - Only Lower Bound: $\Omega(n \log n)$ [Beume et al.'07]

General Scheme of EAs



We touched upon:

1. Concepts from nature for heuristic problem solving
2. Multi-modal approaches, multi-objective approaches

There is so much more: self-adaptation, problem decomposition, hybrid approaches, matheuristics, applications, theoretical results, …

→ Super important: be systematic!
(there is room for creativity, though)

→ Markus: markus.wagner@adelaide.edu.au
http://cs.adelaide.edu.au/~markus/

Advertisement: Competition with AUD 1,000 prize

http://cs.adelaide.edu.au/~optlog/TTP2017Comp

3rd Competition on the Optimisation of Problems with Multiple Interdependent Components

June/July 2017

Motivation Evaluation Prize Important Dates Organizers

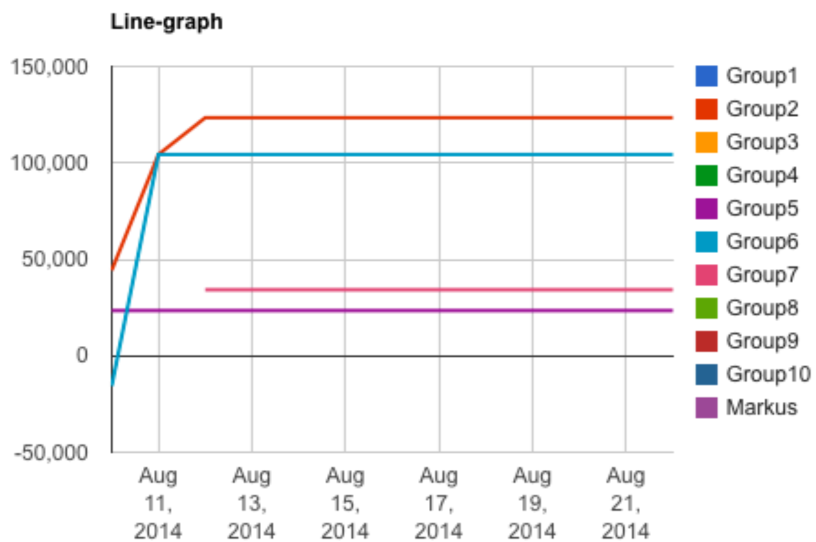
On our side, we will take your solution files and then evalute them. This will include the checking of constraints, such as, knapsack capacity considered, all cities visited (when applicable), starting and end city is city with ID 1, etc.

Online Leaderboard

We are currently developing an online leaderboard that will allow all participants to continuously compare their performances. Potentially, this system will replace your submission via email - we will inform you in time.

a280_n1395_uncorr-similar-weights_05.ttp

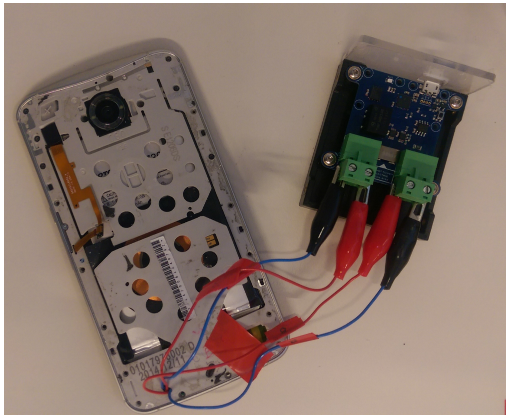
groupID,score,time
Group2,123422.23424234342,2014-08-12 21:11:23
Group6,104363.01731367875,2014-08-11 22:09:15
Group7,34225.1341415121321,2014-08-12 10:11:24
Group5,23442.254215435134,2014-08-10 11:11:43
Group1,-,-
Group3,-,-
Group4,-,-
Group8,-,-
Group9,-,-
Group10,-,-
Markus,-,-



Important Dates

1 May 2017 (anywhere on Earth): submission deadline
Please contact us if you want to receive notifications via email.

Applications in Adelaide

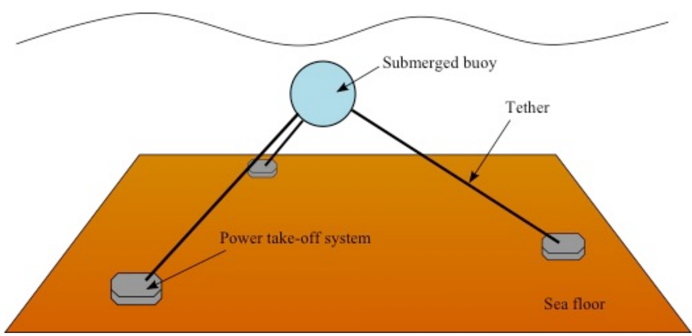


Energy consumption optimisation of apps on smart-phones. Our test bed:
https://www.youtube.com/watch?v=C7WHoLW1KYw

Optimisation of submerged wave energy converters:

http://cs.adelaide.edu.au/~optlog/research/energy.php

In collaboration with Carnegie Wave Pty Ltd and School of Mechanical Engineering



And much more:

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

http://cs.adelaide.edu.au/~markus/publications.html