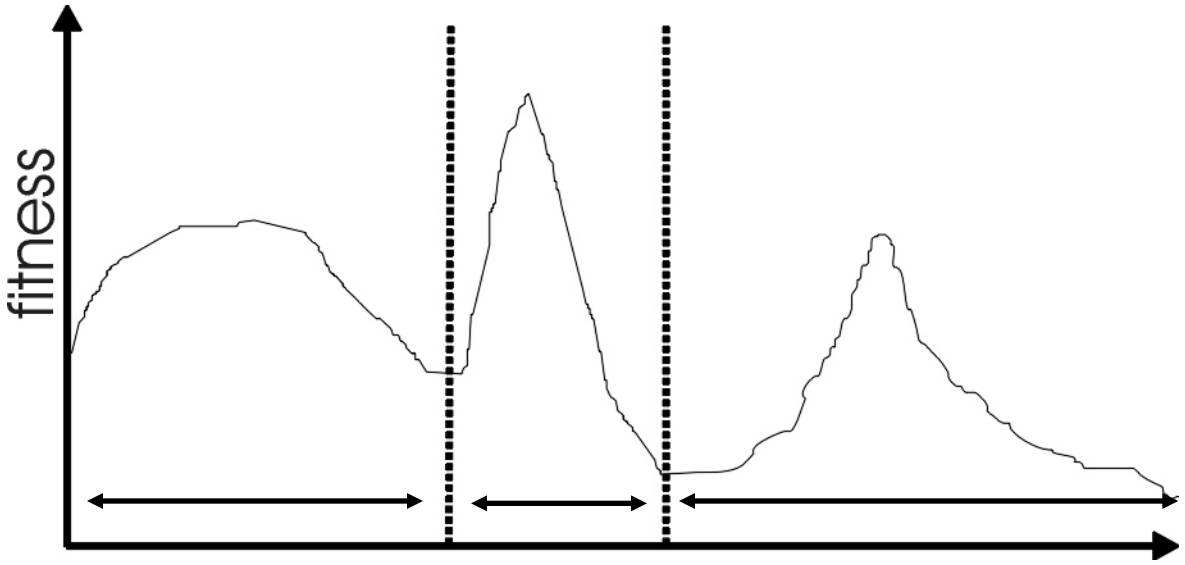
Multimodal Problems, Multi-Objective Optimization

Eiben/Smith Chapter 9

Multimodal Problems

Motivation 1: Multimodality

Most interesting problems have more than one locally optimal solution.



three basins of attraction

Motivation 2: Genetic Drift

- Finite population with global (panmictic) mixing and selection eventually convergence 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

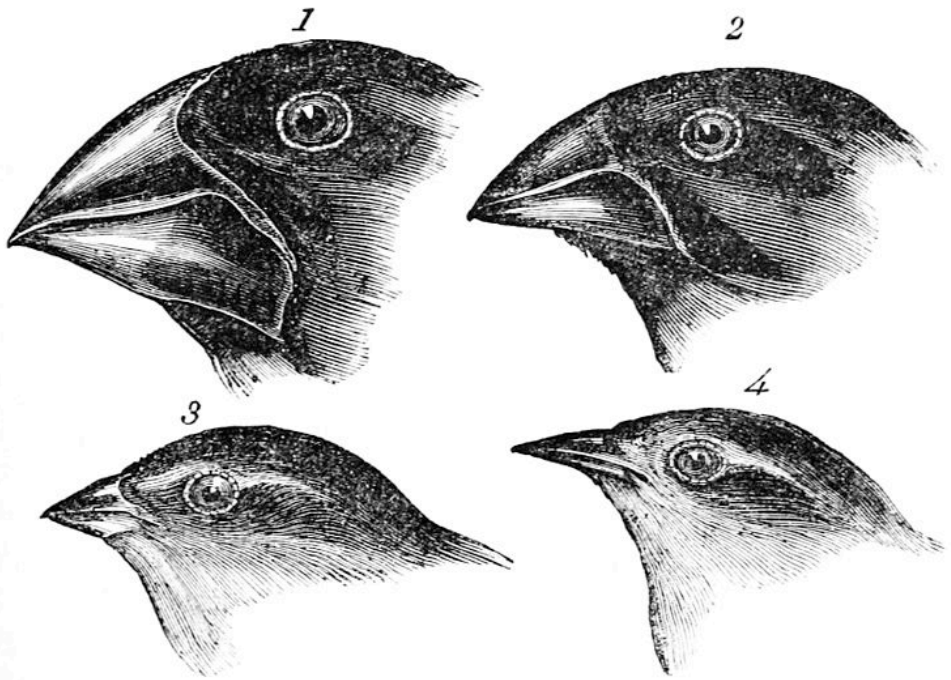
Biological Motivation 1: Speciation

- 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



1. Geospiza magnirostris.
2. Geospiza fortis.
3. Geospiza parvula.
4. Certhidea olivacea.

Biological Motivation 2: Punctuated Equilbria

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- 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."

2014: decision "no wolves will be transferred to attempt a genetic rescue"
2016: "[The] wolf population is now nearly extinct with only two severely inbred wolves present.

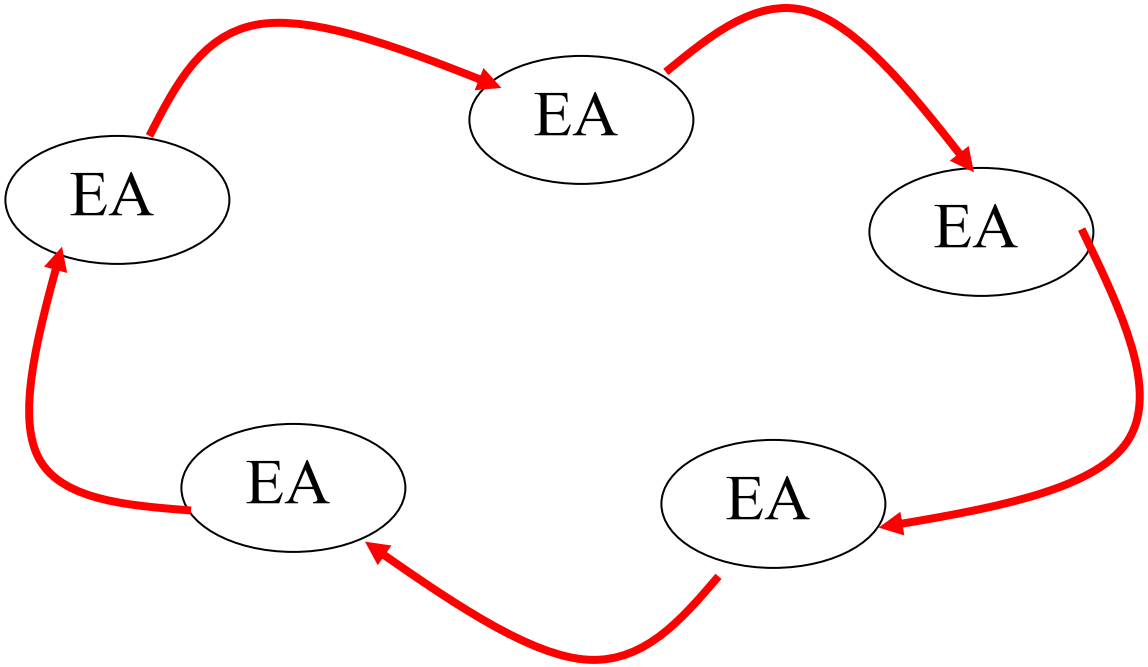
2018: four wolves were transferred

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



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

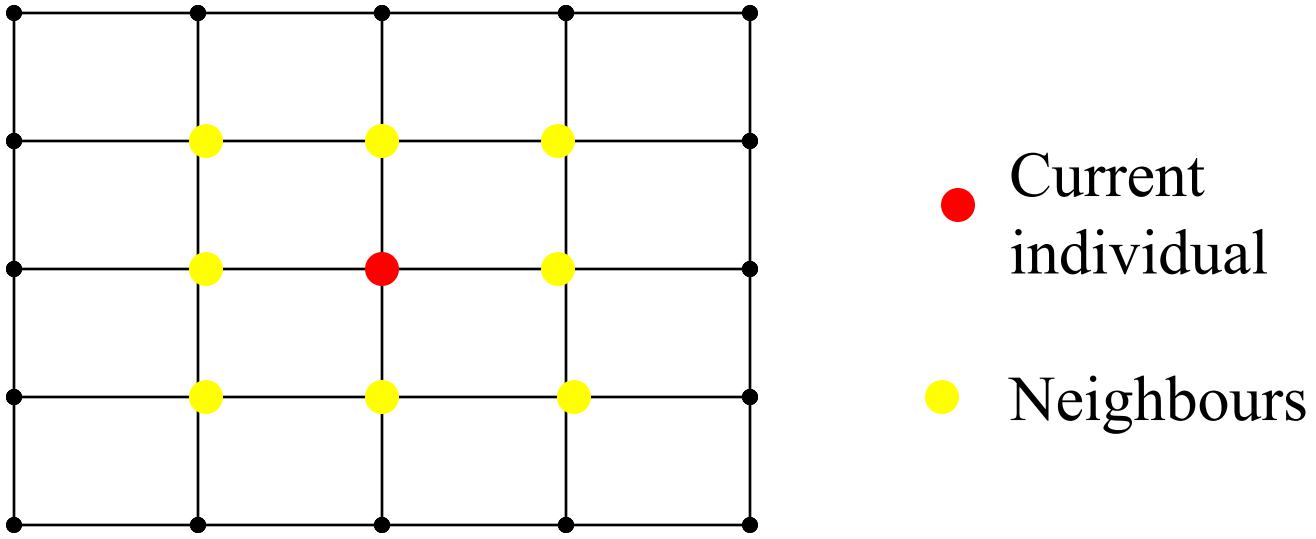
- How many, which individuals to exchange ?
  - usually ~1-5, but depends on population size.
  - more sub-populations usually gives better results but there can be a "critical mass", i.e., minimum size of each sub population needed
  - Sometimes: it is better to exchange randomly selected individuals than best
  - can select random/worst individuals to replace

Additional reading:

Dirk Sudholts 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 pop



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

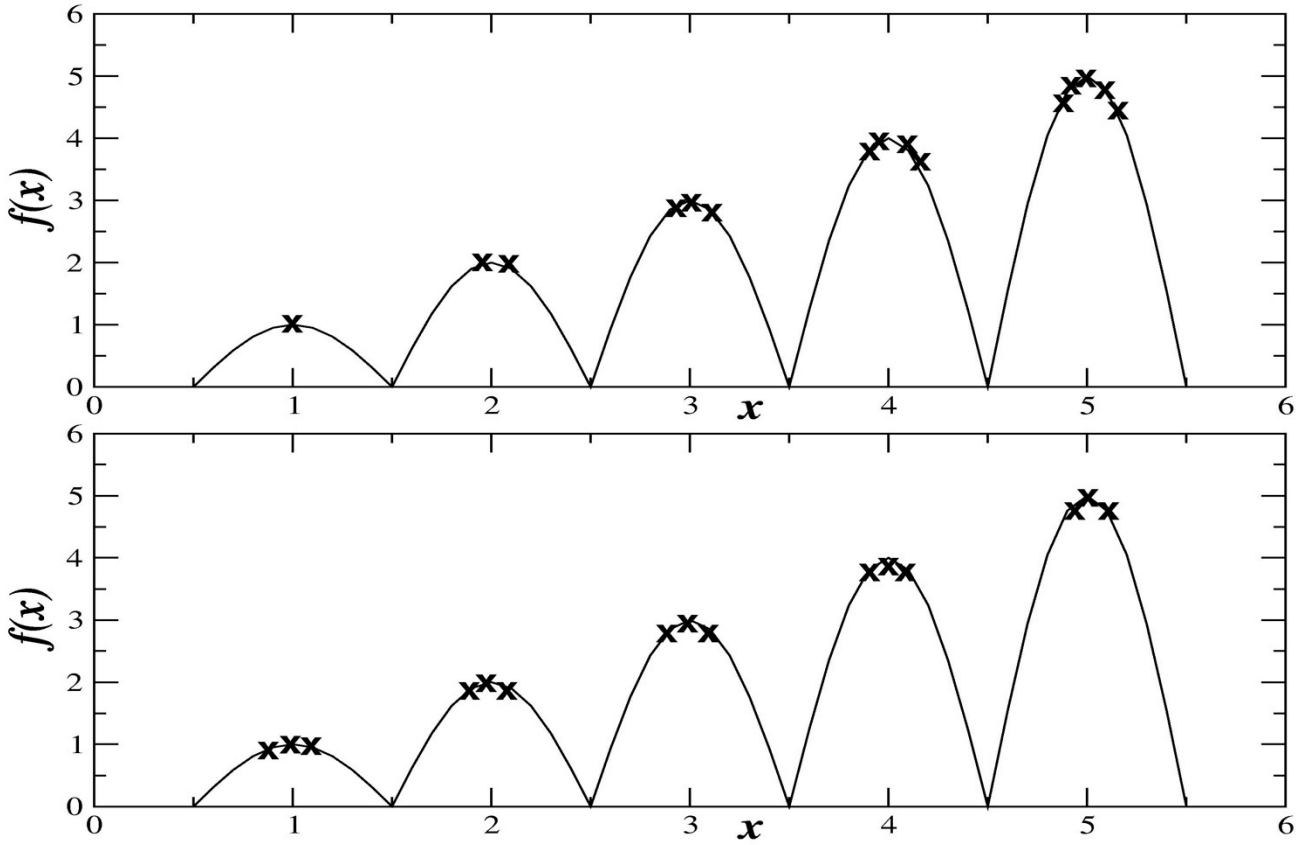
- Assume rectangular grid so each individual has 8 immediate neighbours
- equivalent of 1 generation is:
  - pick point in pop at random
  - pick one of its neighbours using roulette wheel
  - crossover to produce 1 child, mutate
  - replace individual if fitter
  - circle through population until done

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 (i.e., the match does not have to be perfect
  - can also use tags to perform fitness sharing (see later) to try and distribute members amongst niches

solution 1	**x1 x2 x3 x4 … xn**	0 1 1 0
solution 2	**x1 x2 x3 x4 … xn**	0 0 1 0
solution 3	**x1 x2 x3 x4 … xn**	1 1 1 1

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 participates in tournament 1)

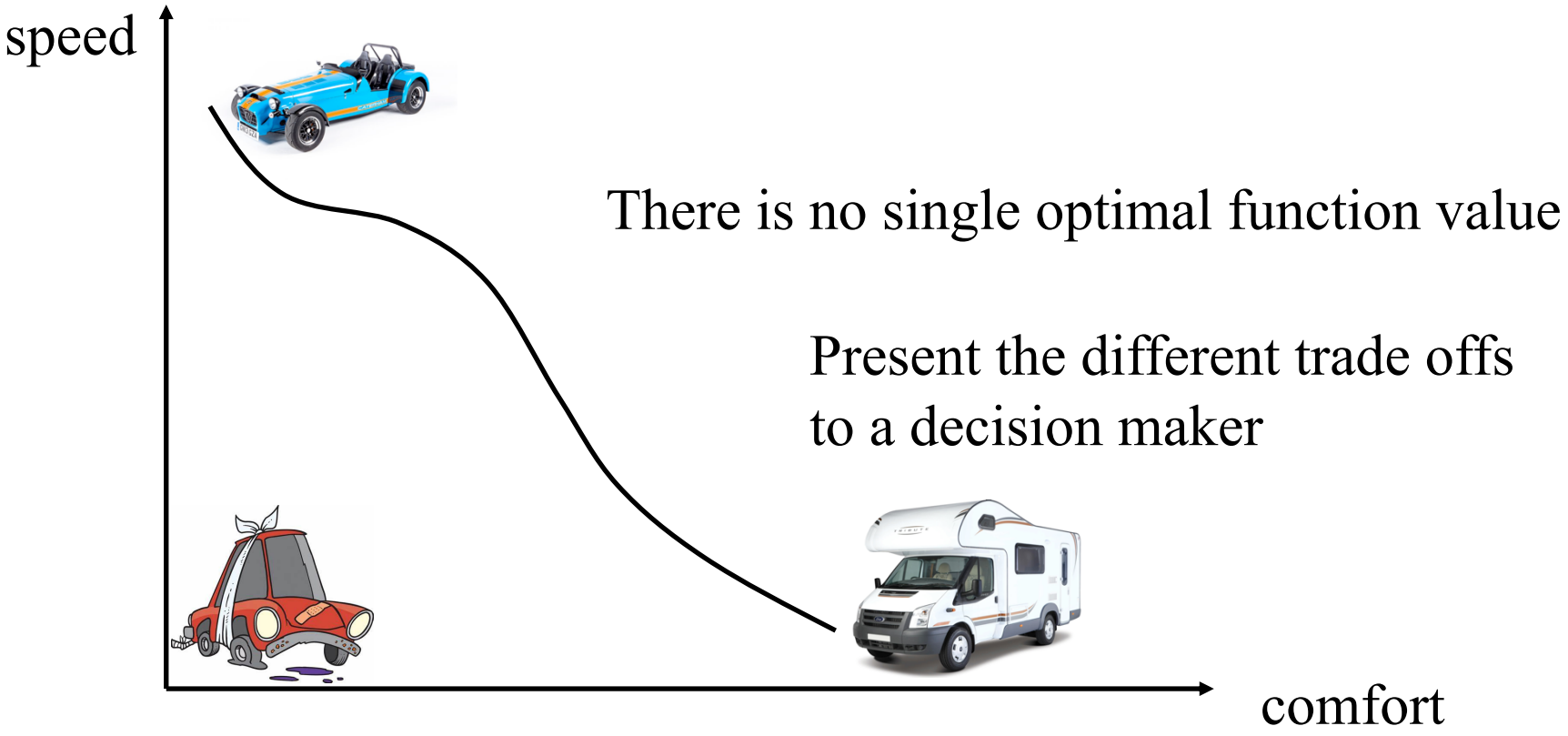
Multi-Objective Optimisation

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



MOPs 1: Conventional approaches

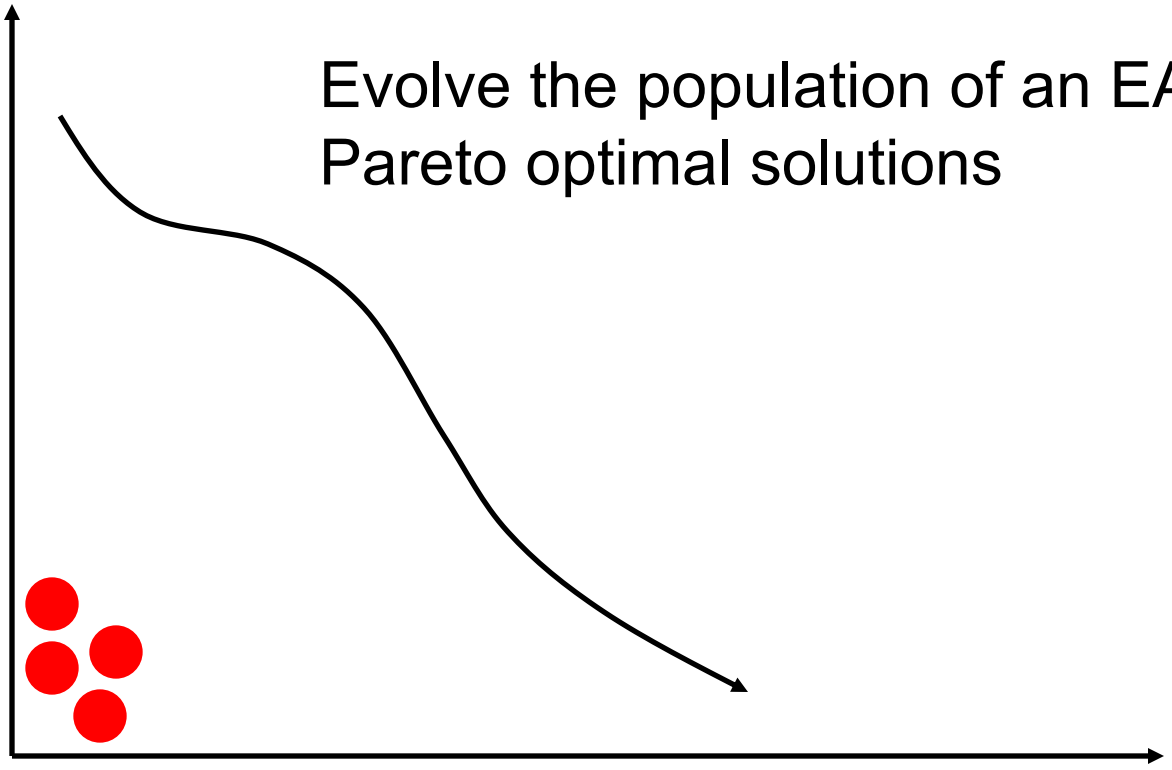
rely on using a weighting of objective function values to give a single scalar objective function which can then be optimised

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

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

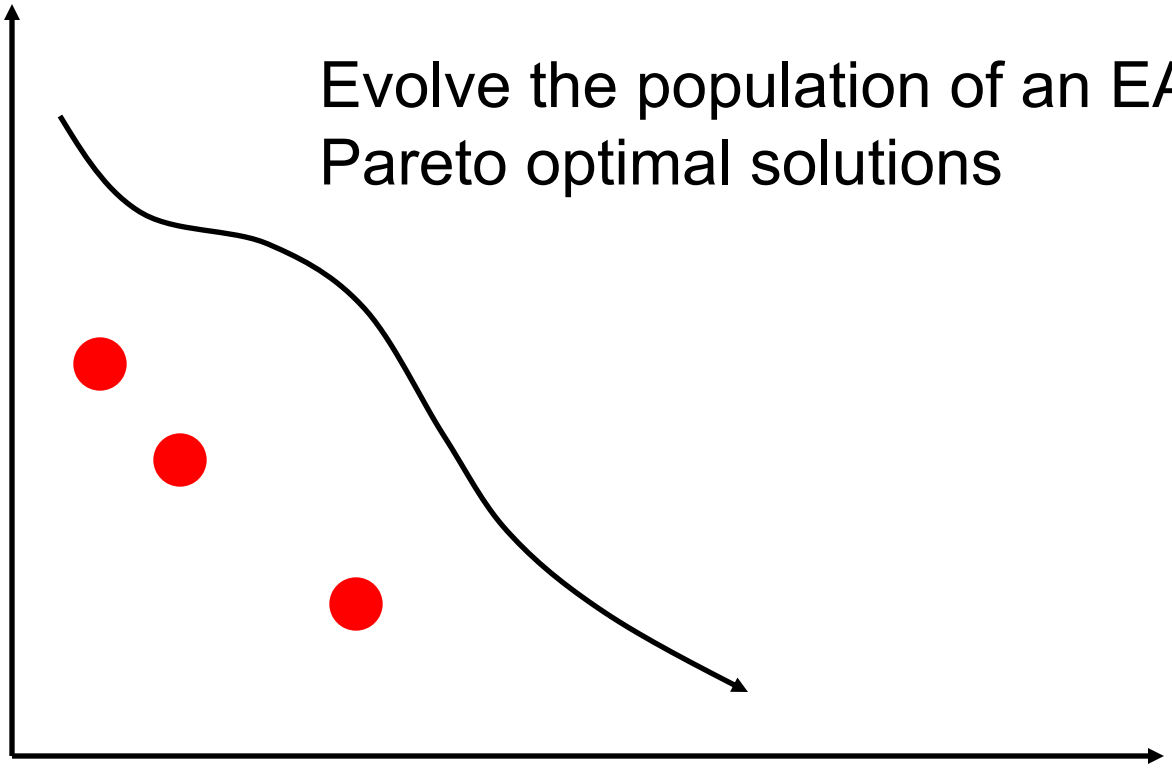
Evolutionary Multi-Objective Optimization

Try to compute/approximate the Pareto front by EAs



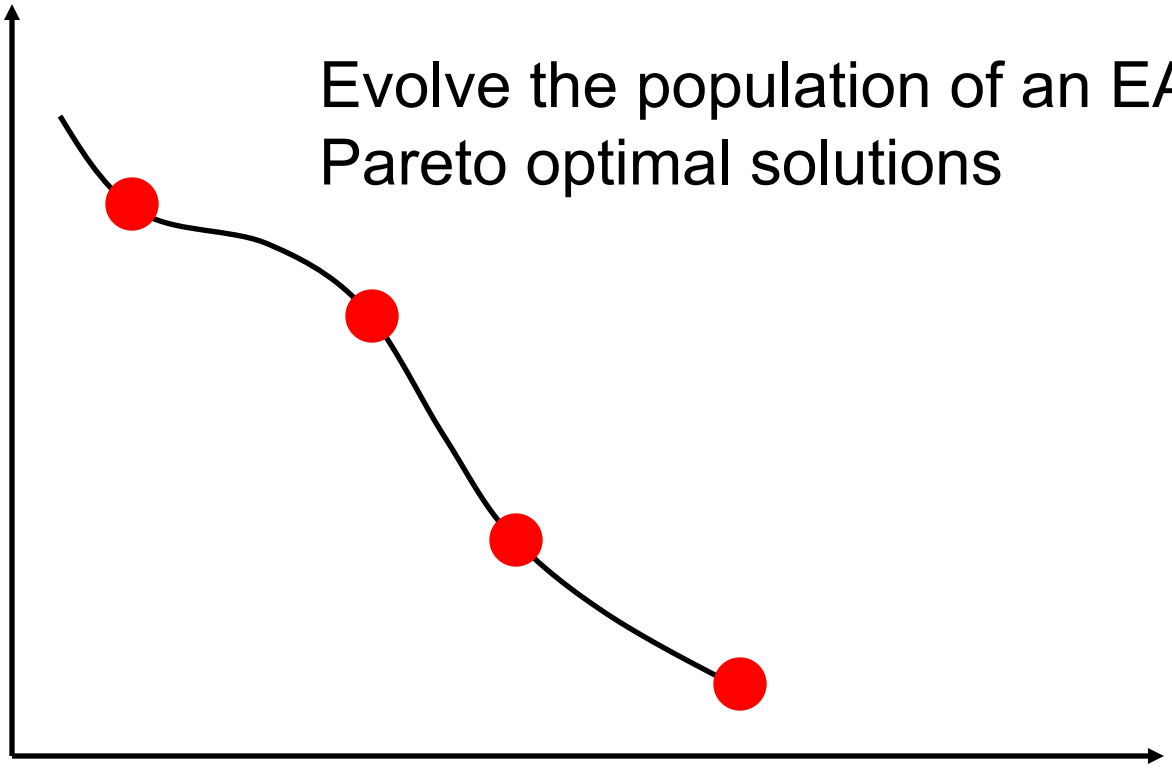
Evolutionary Multi-Objective Optimization

Try to compute/approximate the Pareto front by EAs



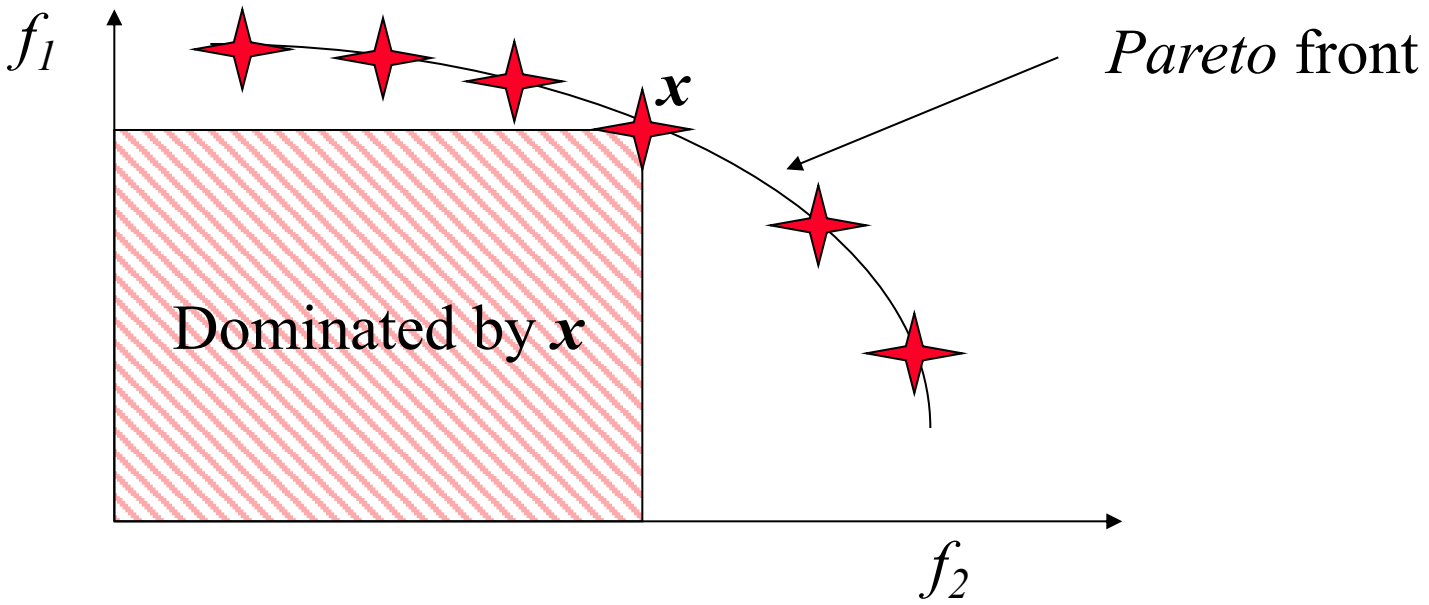
Evolutionary Multi-Objective Optimization

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

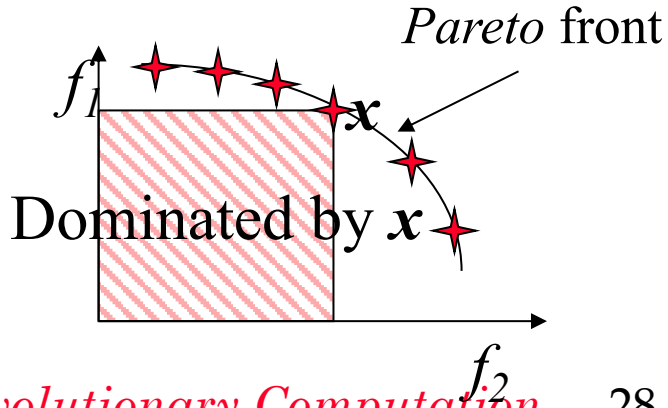
u *weakly dominates* v $(u \succeq v)$ iff $u_i \geq v_i$ for all $i \in \{1, \ldots, m\}$

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

THE UNIVERSITY OF ADELAIDE AUSTRALIA

SUB CRUCE LUMEN

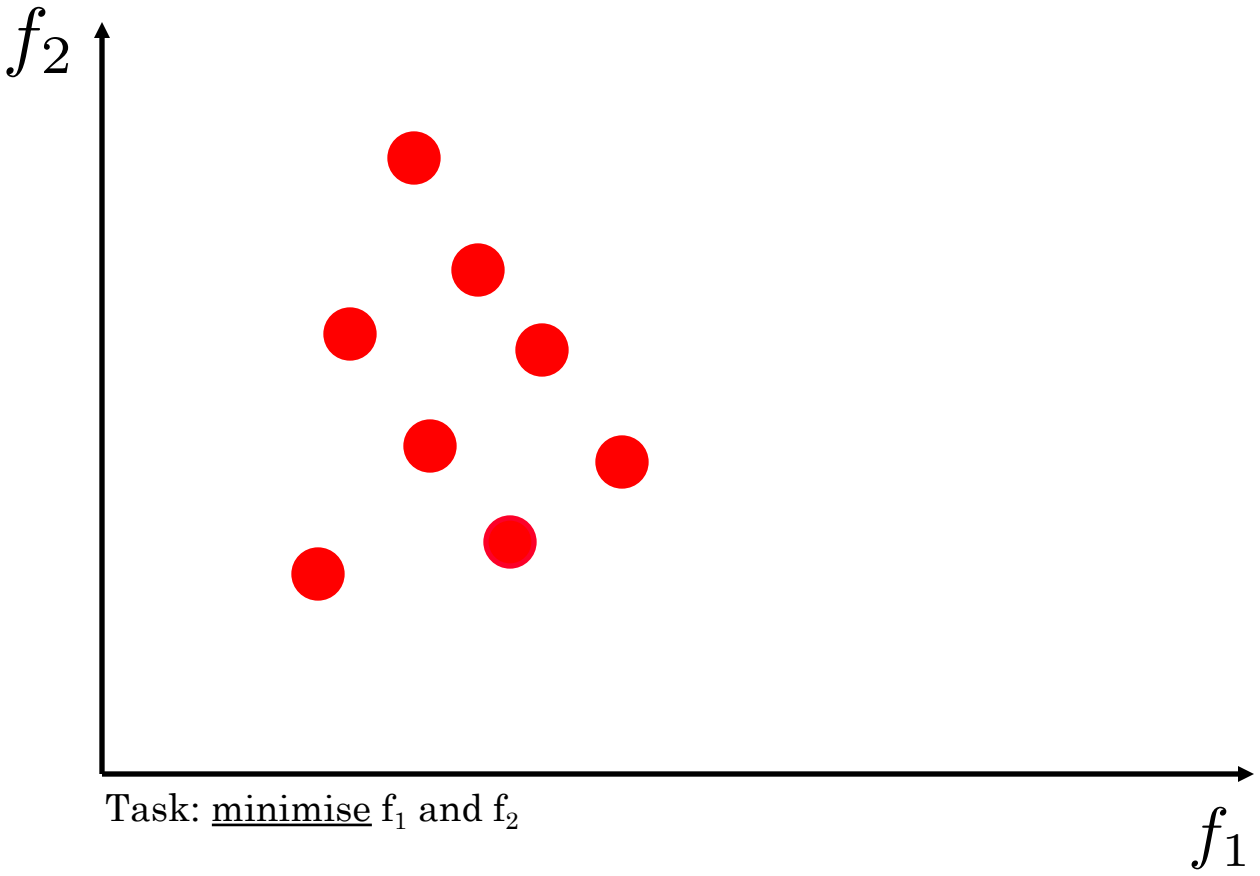
Idea

- 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.

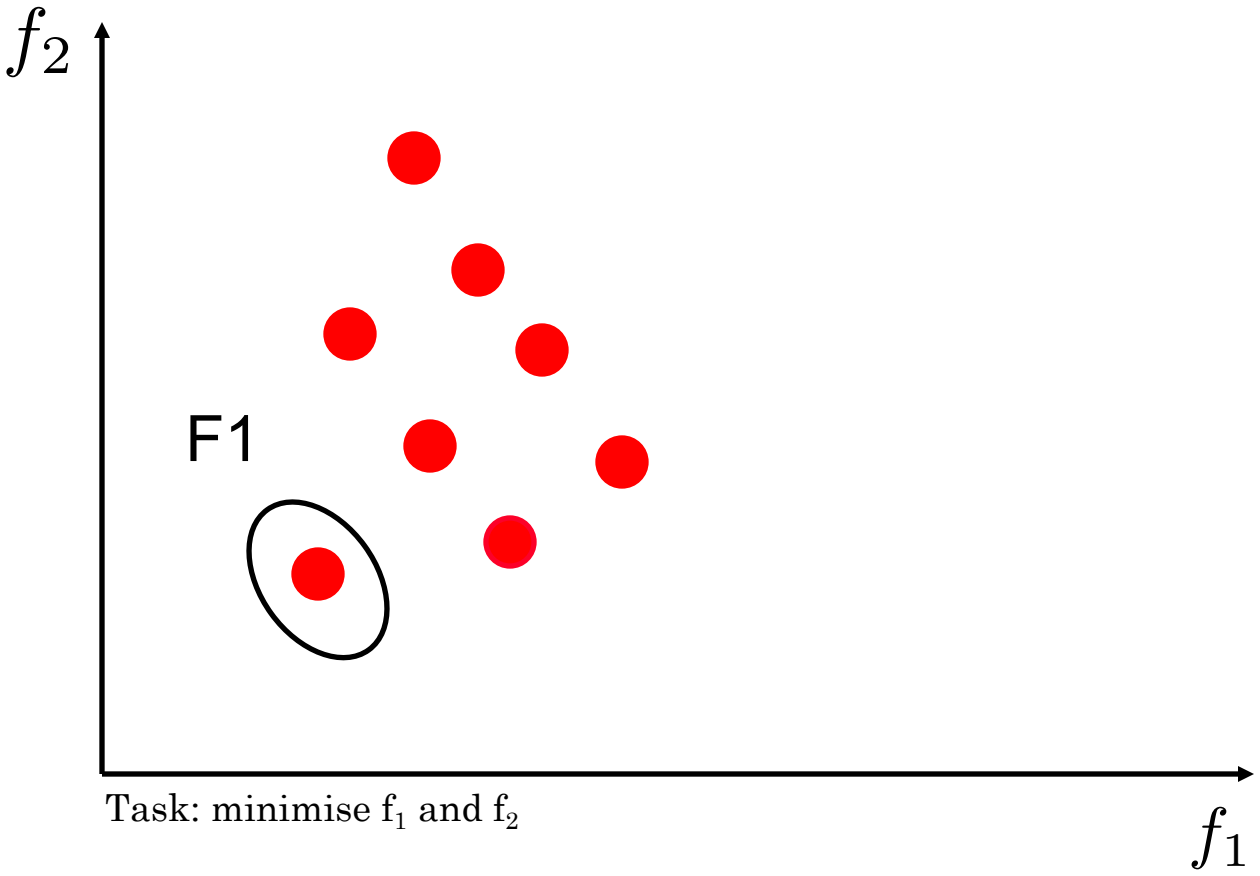
Procedure

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

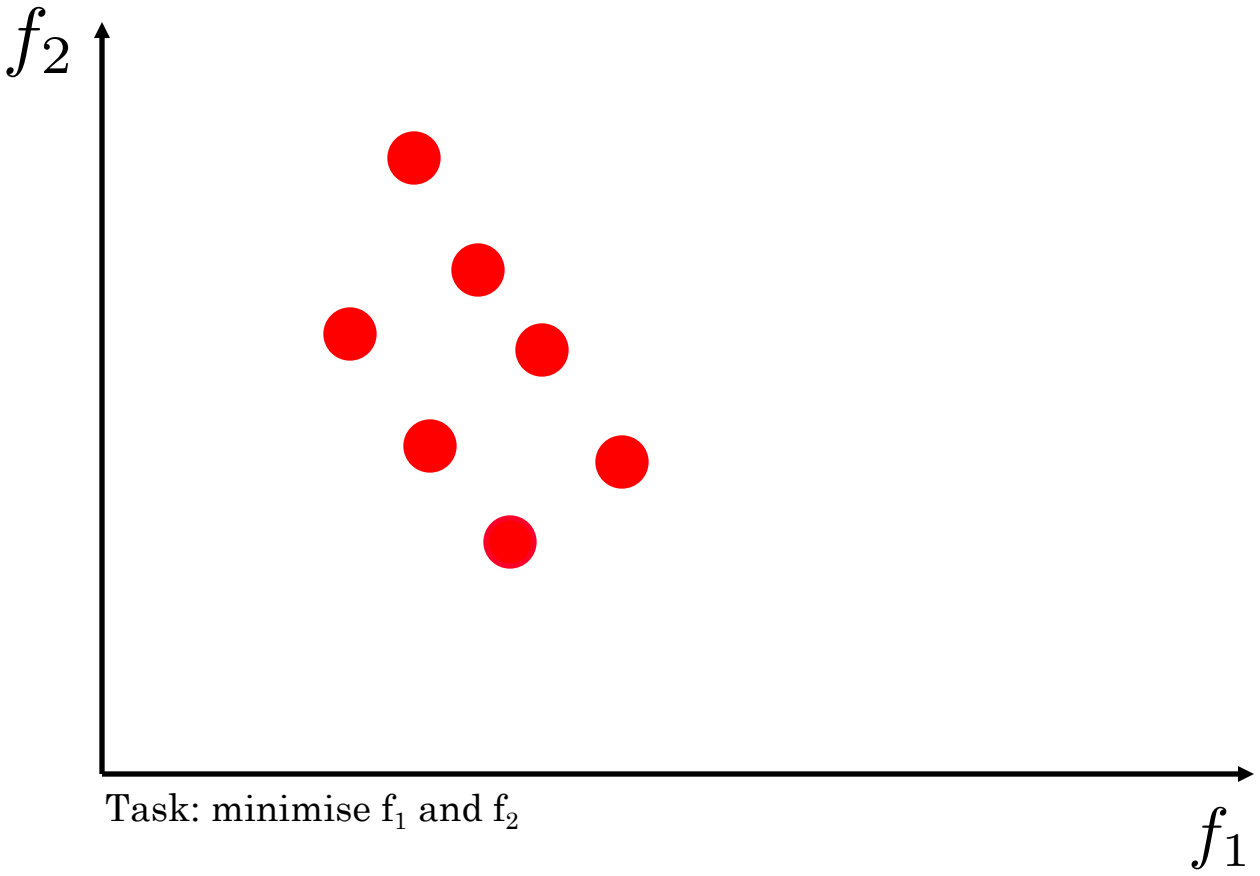
Non-dominated Sorting



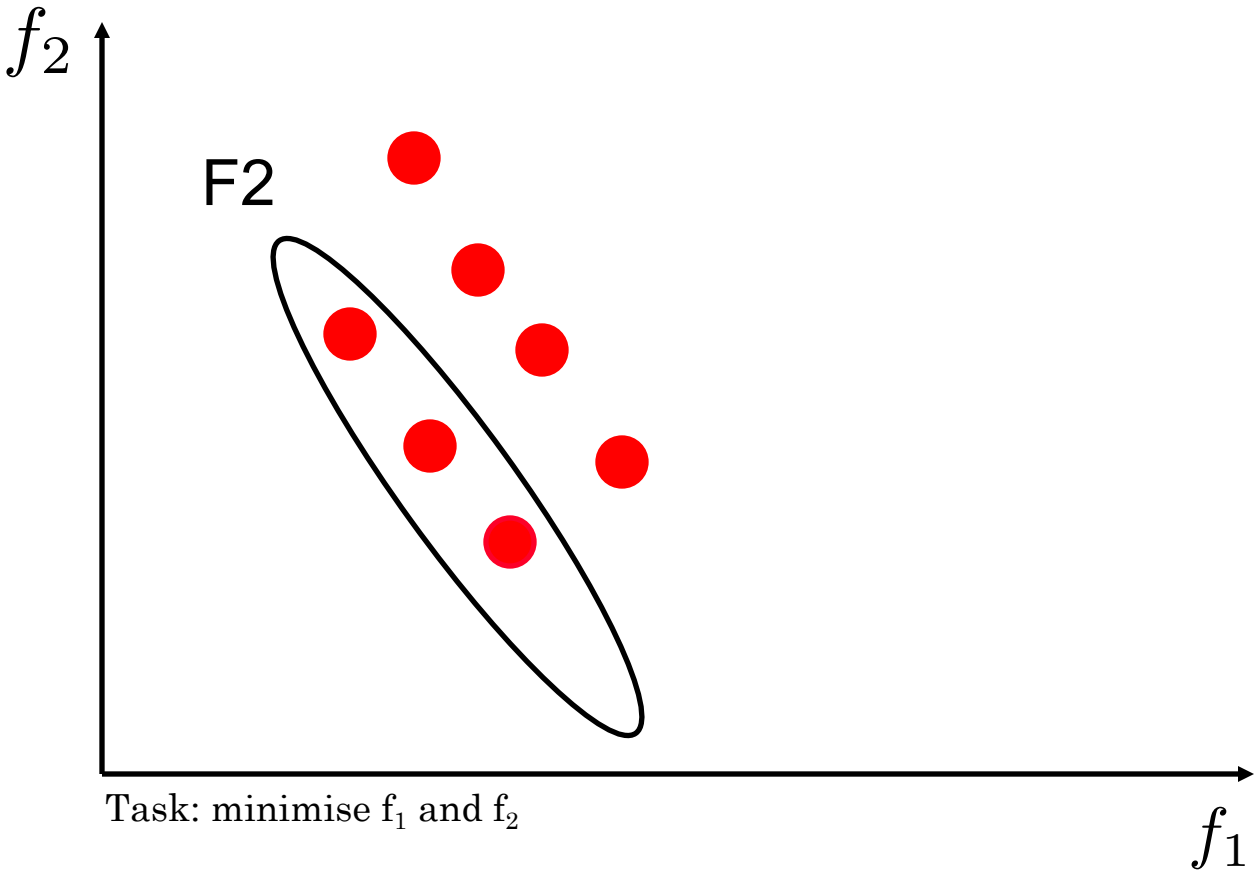
Non-dominated Sorting



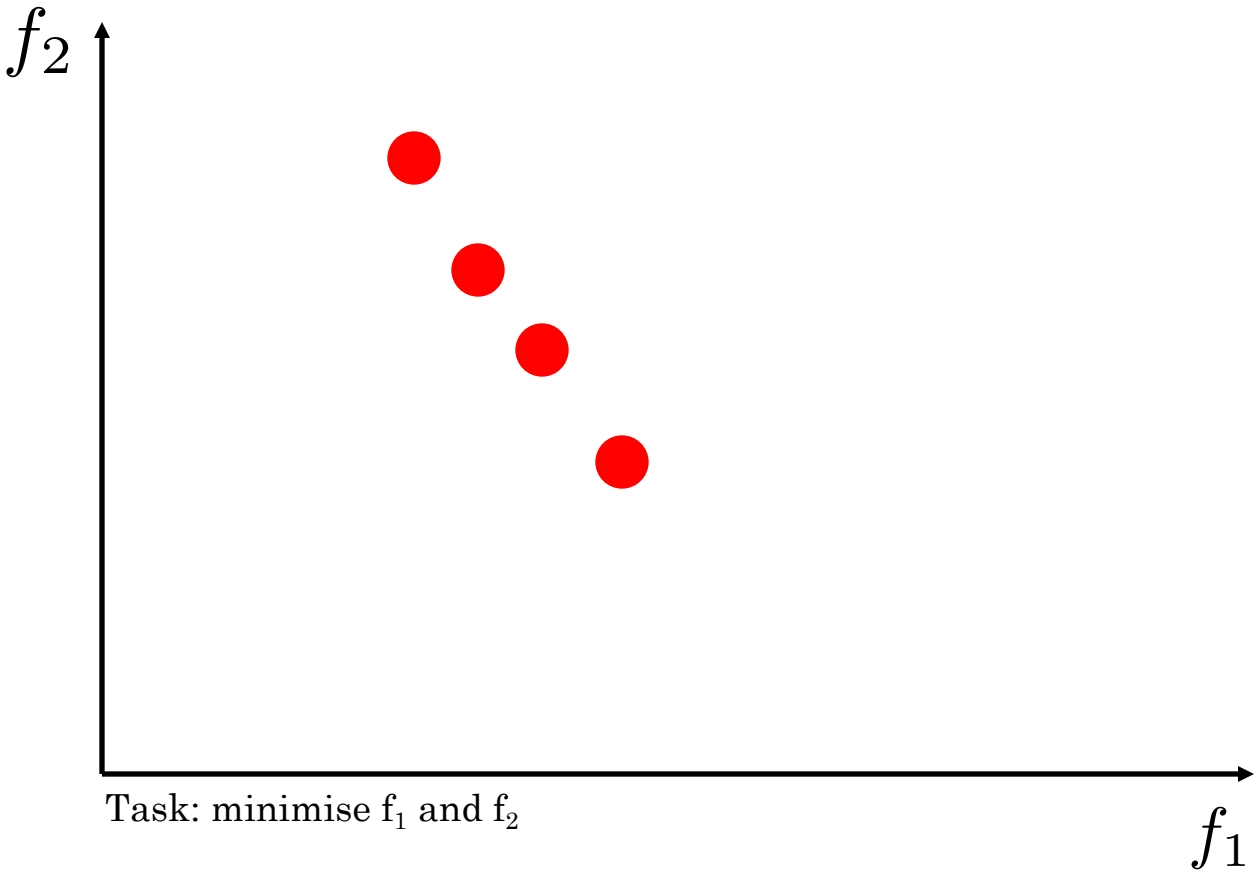
Non-dominated Sorting



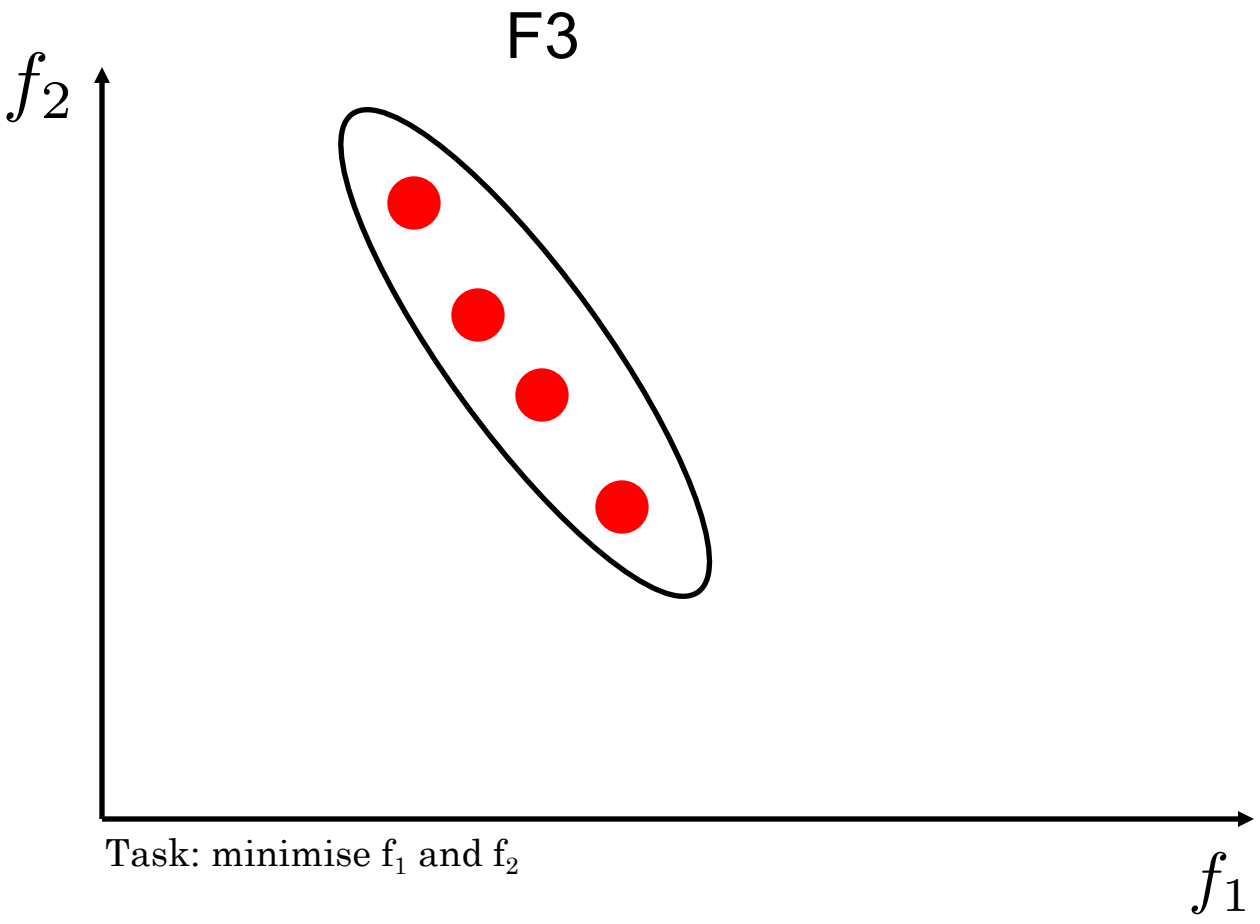
Non-dominated Sorting



Non-dominated Sorting



Non-dominated Sorting



Fast non-dominated Sorting

THE UNIVERSITY OF ADELAIDE AUSTRALIA

SUB CRUCE LUMEN

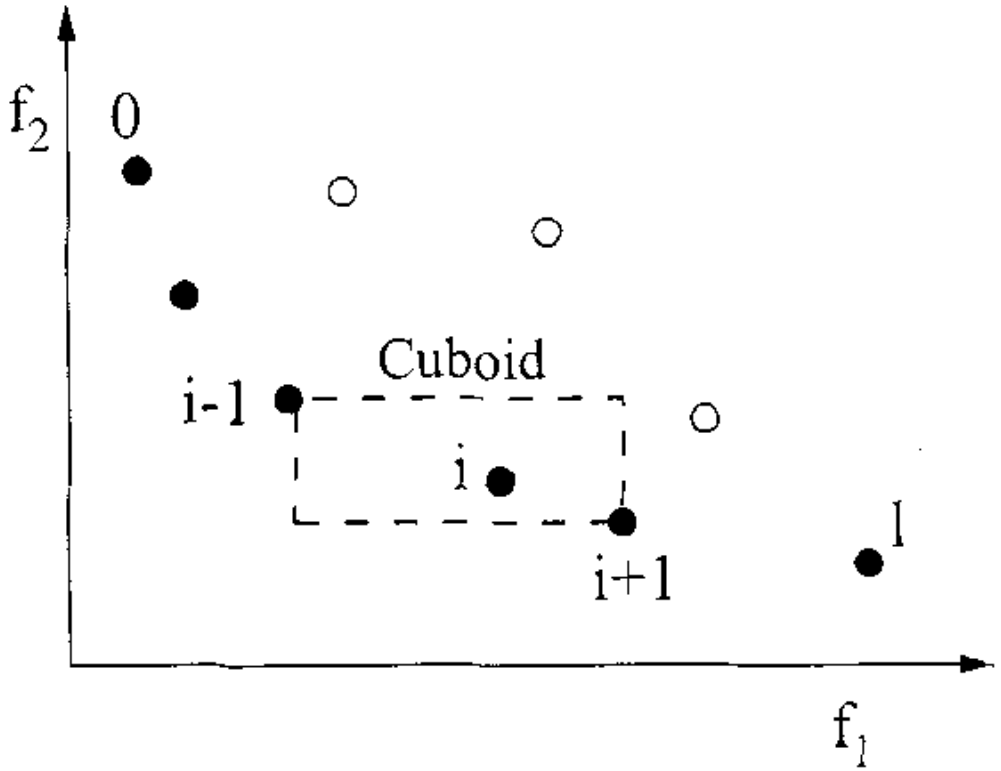
$\texttt{fast-non-dominated-sort}(P)$

for each $p \in P$
  $S_p = \emptyset$
  $n_p = 0$
  for each $q \in P$
    if $(p \prec q)$ then — If p dominates q
      $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
    else if $(q \prec p)$ then
      $n_p = n_p + 1$ — Increment the domination counter of p
  if $n_p = 0$ then — p belongs to the first front
    $p_{\text{rank}} = 1$
    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$

$i = 1$ — Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$
  $Q = \emptyset$ — Used to store the members of the next front
  for each $p \in \mathcal{F}_i$
    for each $q \in S_p$
      $n_q = n_q - 1$
      if $n_q = 0$ then — q belongs to the next front
        $q_{\text{rank}} = i + 1$
        $Q = Q \cup \{q\}$
  $i = i + 1$
  $\mathcal{F}_i = Q$

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

THE UNIVERSITY OF ADELAIDE AUSTRALIA

SUB CRUCE LUMEN

$\underline{\texttt{crowding-distance-assignment}(\mathcal{I})}$

$l = |\mathcal{I}|$ — number of solutions in $\mathcal{I}$

for each i, set $\mathcal{I}[i]_{\text{distance}} = 0$ — initialize distance

for each objective m

$\mathcal{I} = \text{sort}(\mathcal{I}, m)$ — sort using each objective value

$\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$ — so that boundary points are always selected

for $i = 2$ to $(l - 1)$ — for all other points

$\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)/(f_m^{\max} - f_m^{\min})$

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

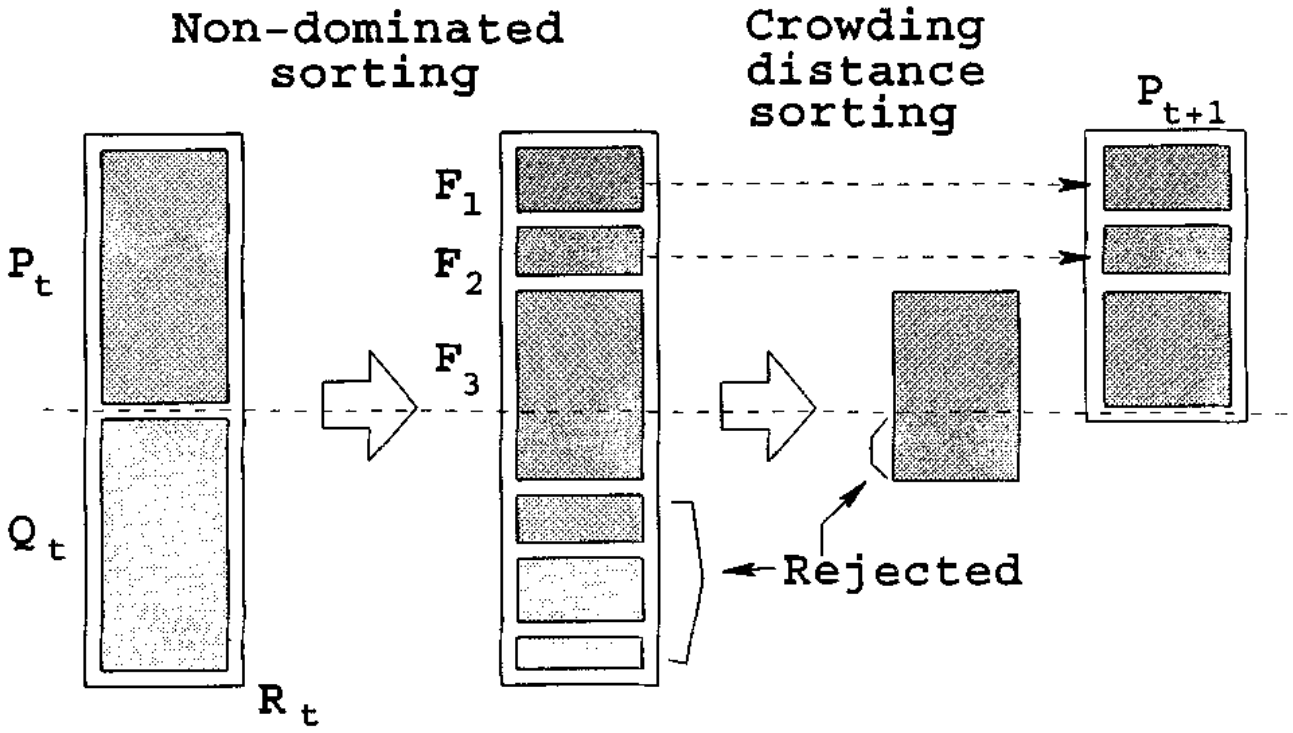
Crowded comparison operator $(\prec_n)$

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

We now define a partial order $\prec_n$ as

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

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 $\lvert P_{t+1}\rvert + \lvert\mathcal{F}_i\rvert \leq N$	until the parent population is filled
$\quad\texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in $\mathcal{F}_i$
$\quad P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$\quad i = i + 1$	check the next front for inclusion
Sort$(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1}\rvert)]$	choose the first $(N - \lvert P_{t+1}\rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	$\quad$ use selection, crossover and mutation to create $\quad\quad$ 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

THE UNIVERSITY OF ADELAIDE AUSTRALIA

$R_t = P_t \cup Q_t$ — 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

$\text{Sort}(\mathcal{F}_i, \prec_n)$ — sort in descending order using $\prec_n$

$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ — 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

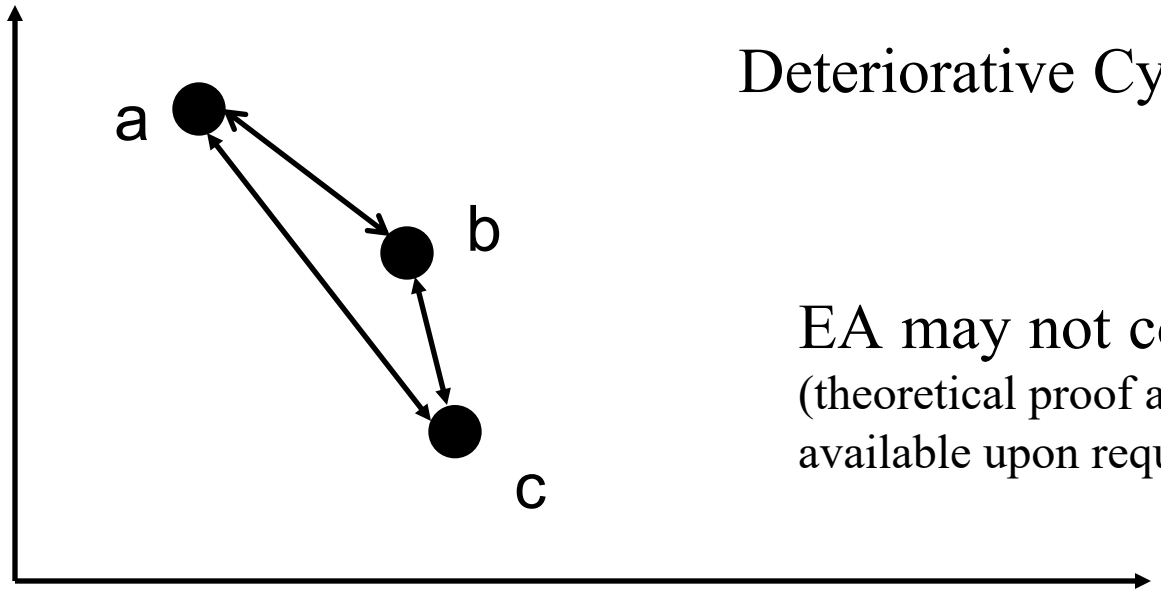
Compute crowding distance as part of this step

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.

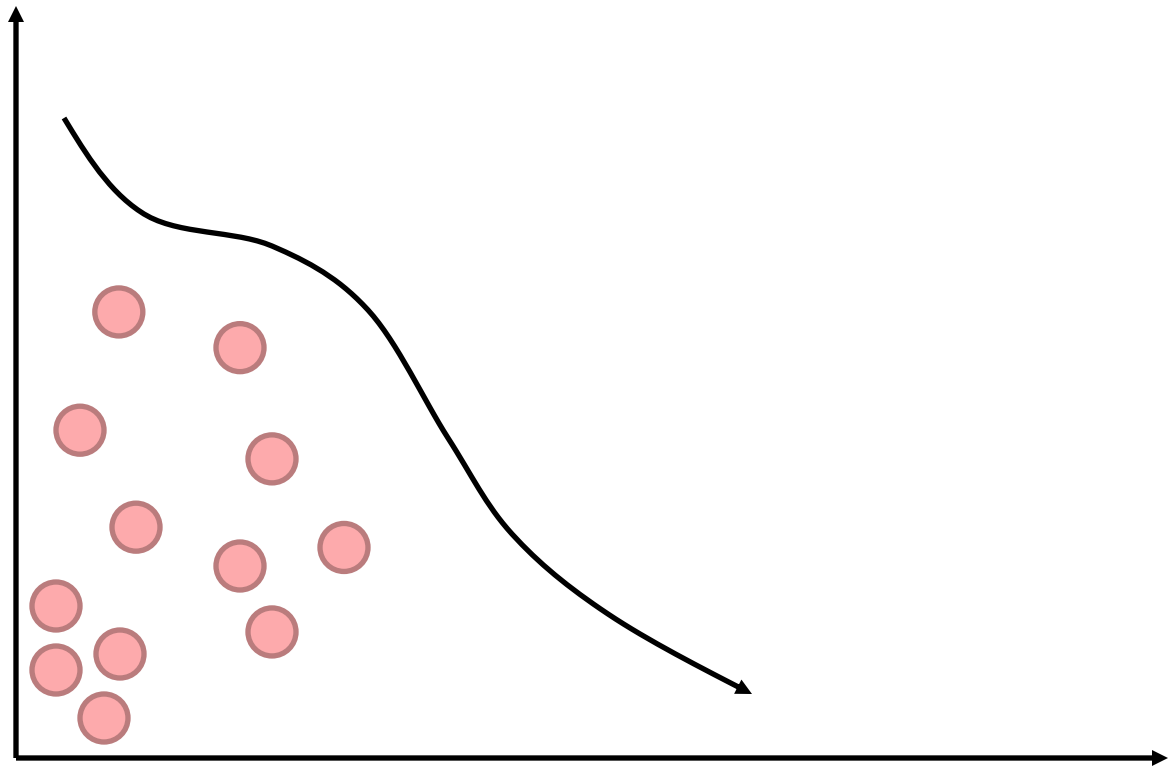
We may move between solutions if they are incomparable



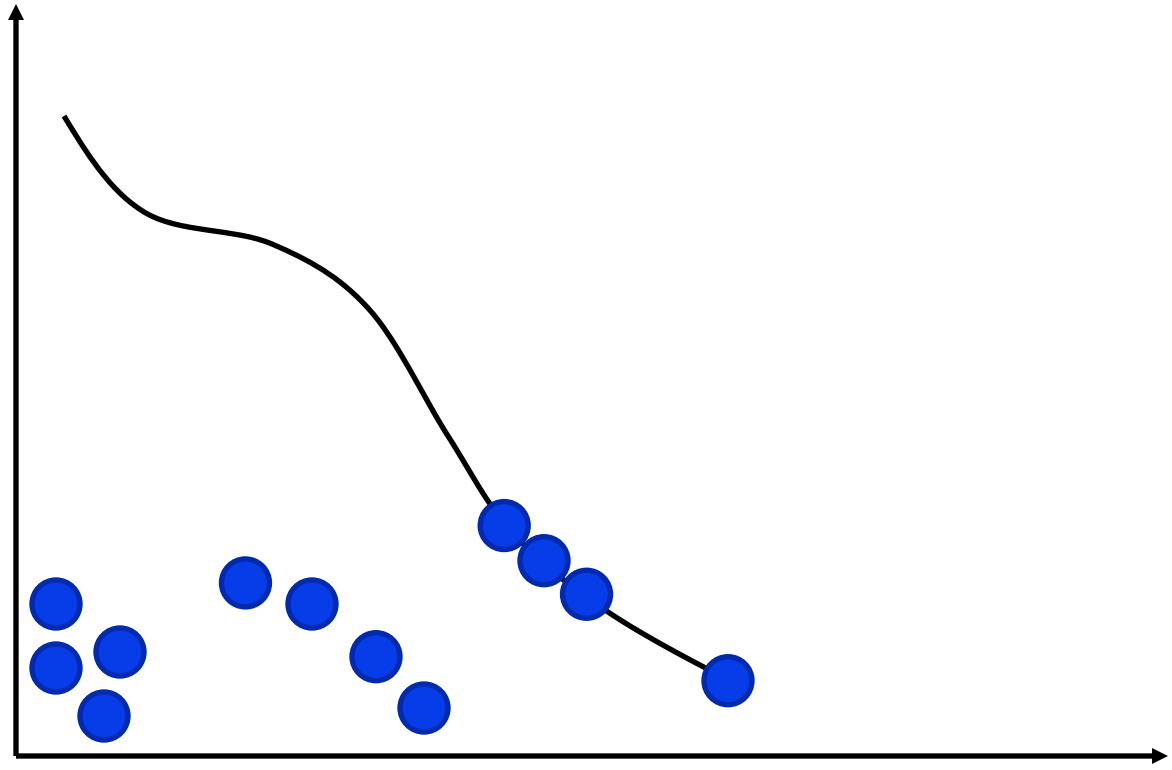
Deteriorative Cycles

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

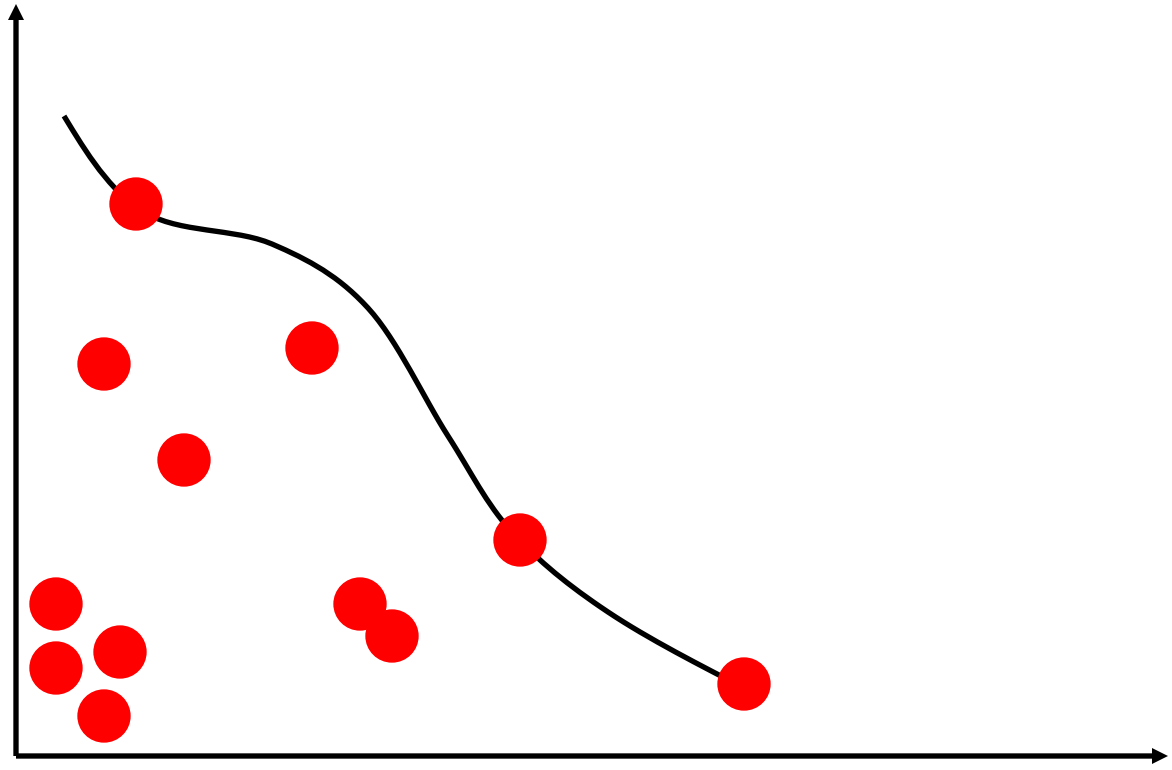
Set-Based Multi-Objective Optimisation



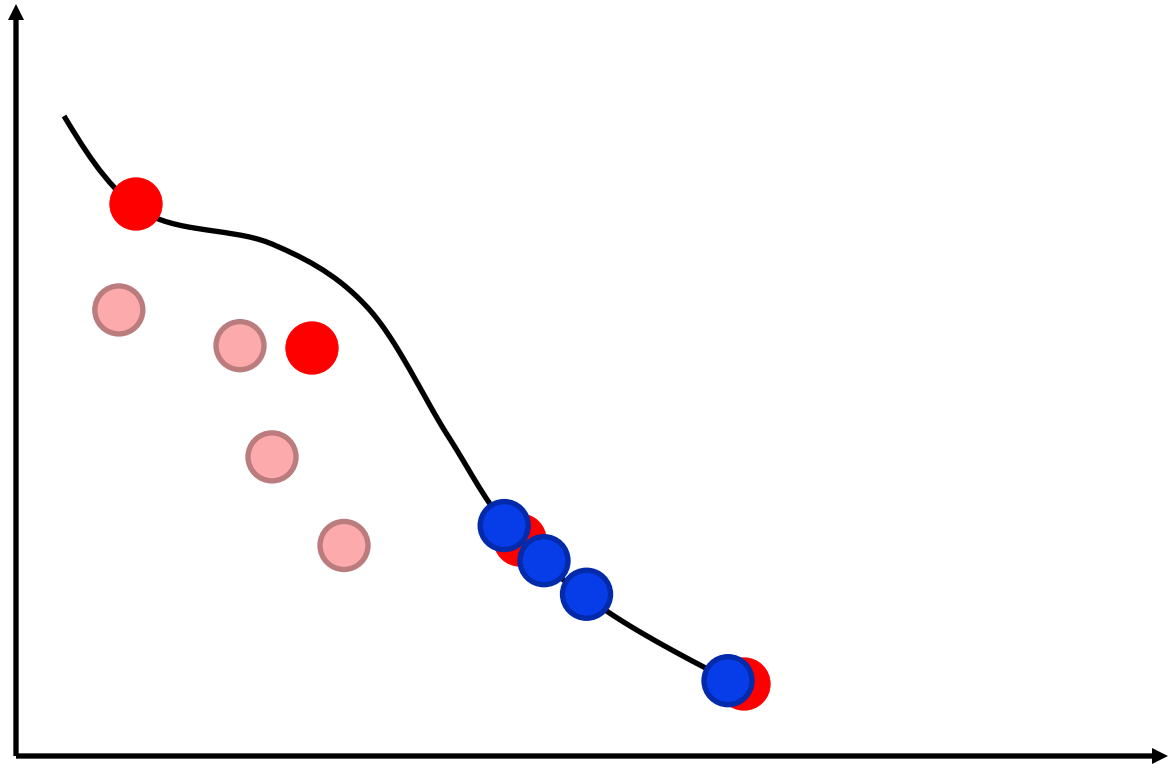
Set-Based Multi-Objective Optimisation



Set-Based Multi-Objective Optimisation

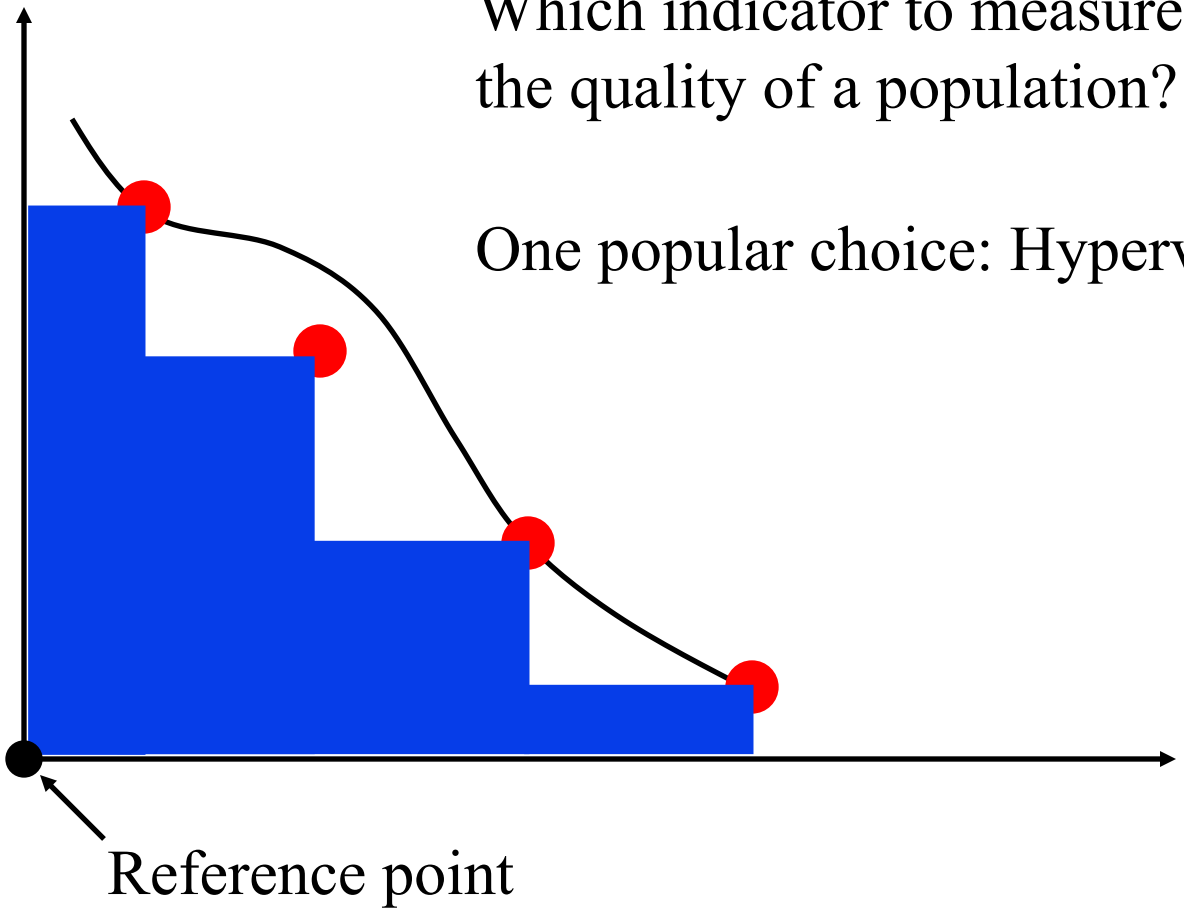


Set-Based Multi-Objective Optimisation



Question
Which algorithm performed best?

Set-Based Multi-Objective Optimisation

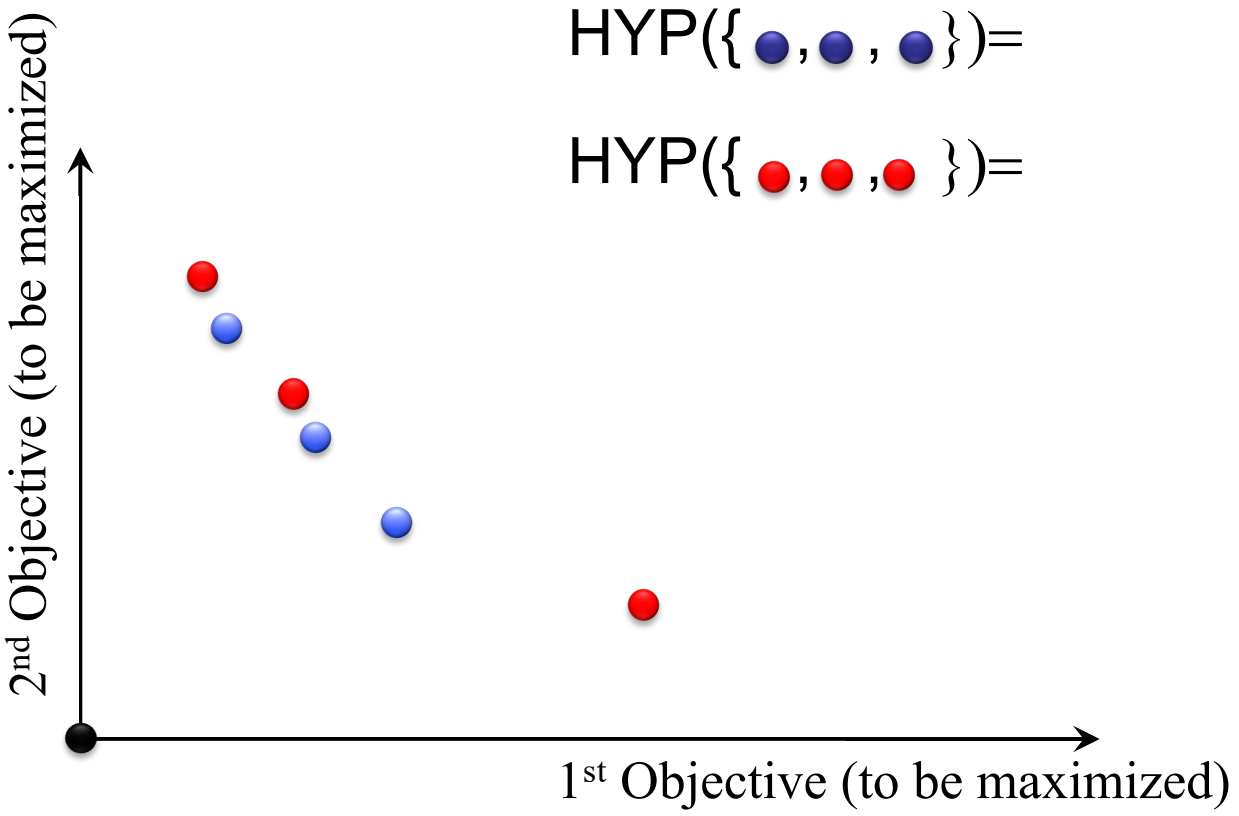


Which indicator to measure the quality of a population?

One popular choice: Hypervolume indicator

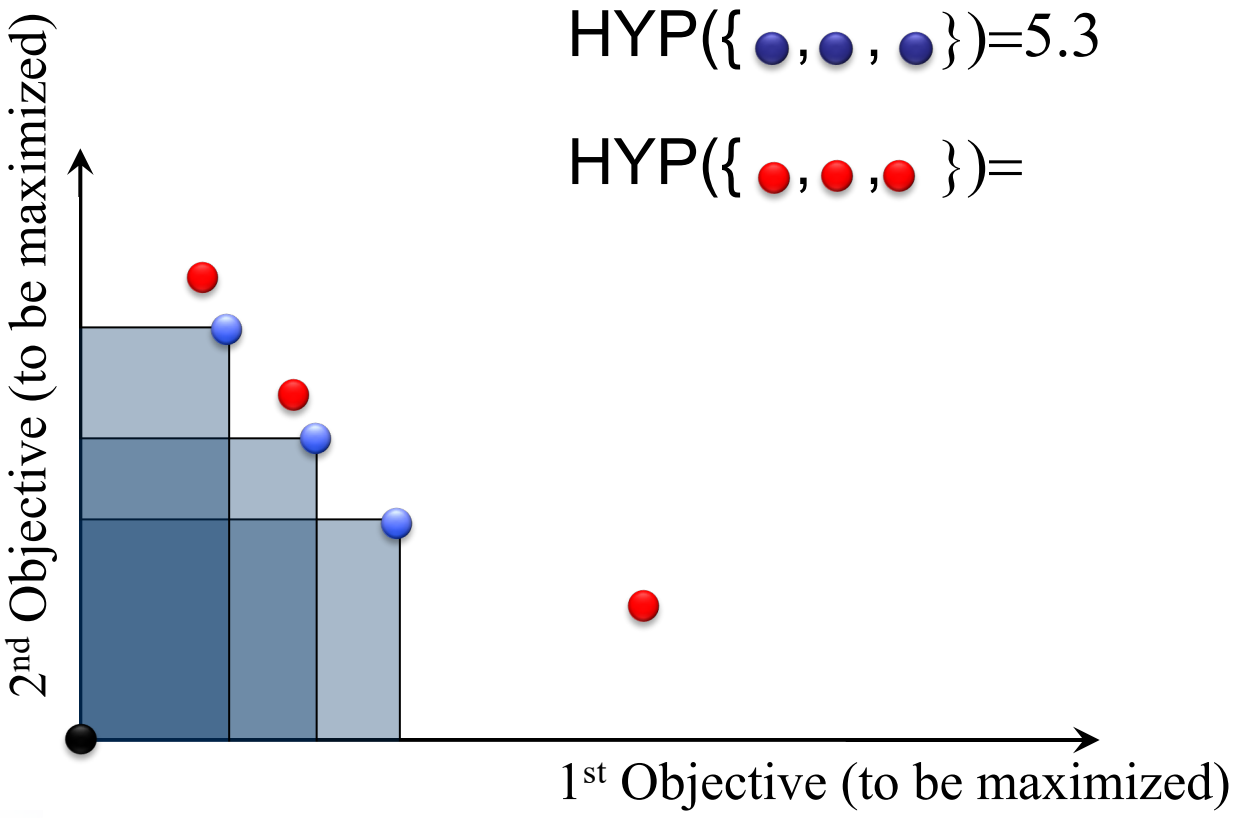
Hypervolume Indicator

Which population is better?



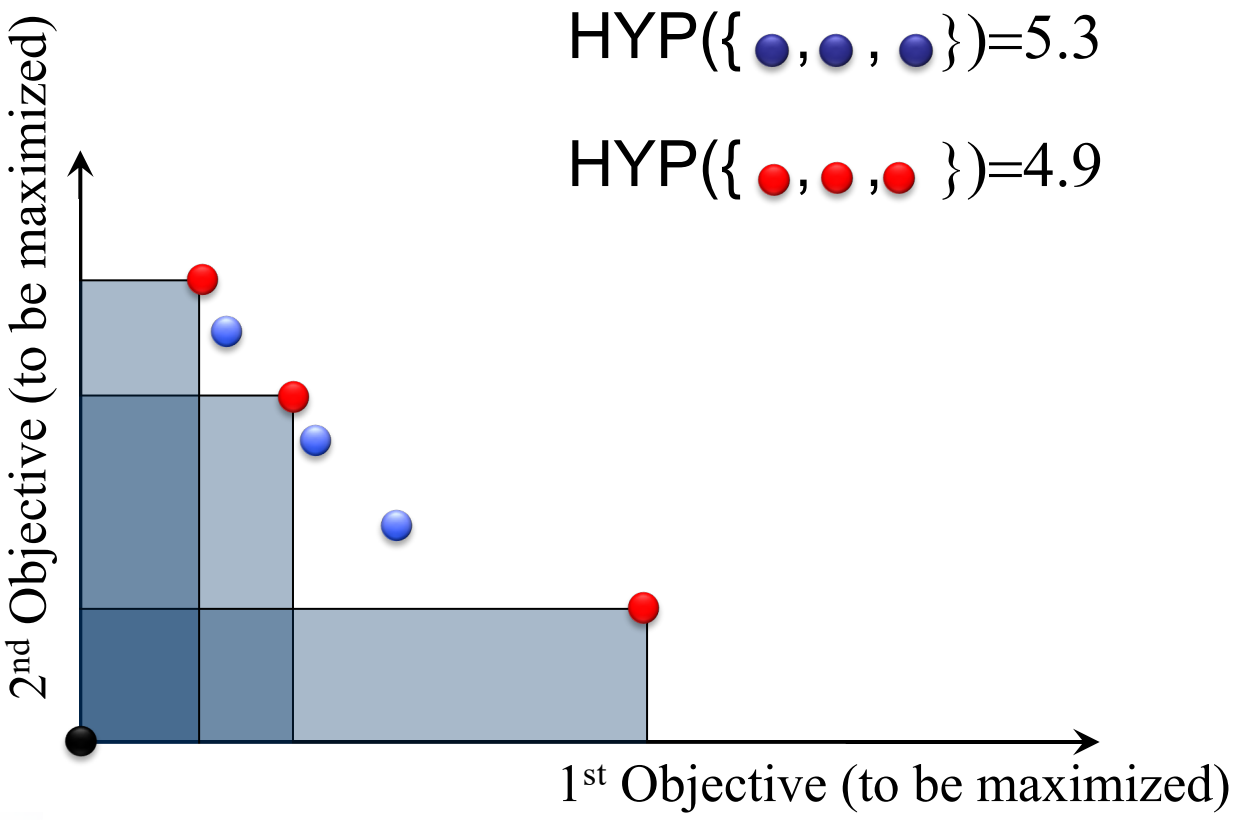
Hypervolume Indicator

Which population is better?



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

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

The computation is costly, and in general exponential in the number of objectives.

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)

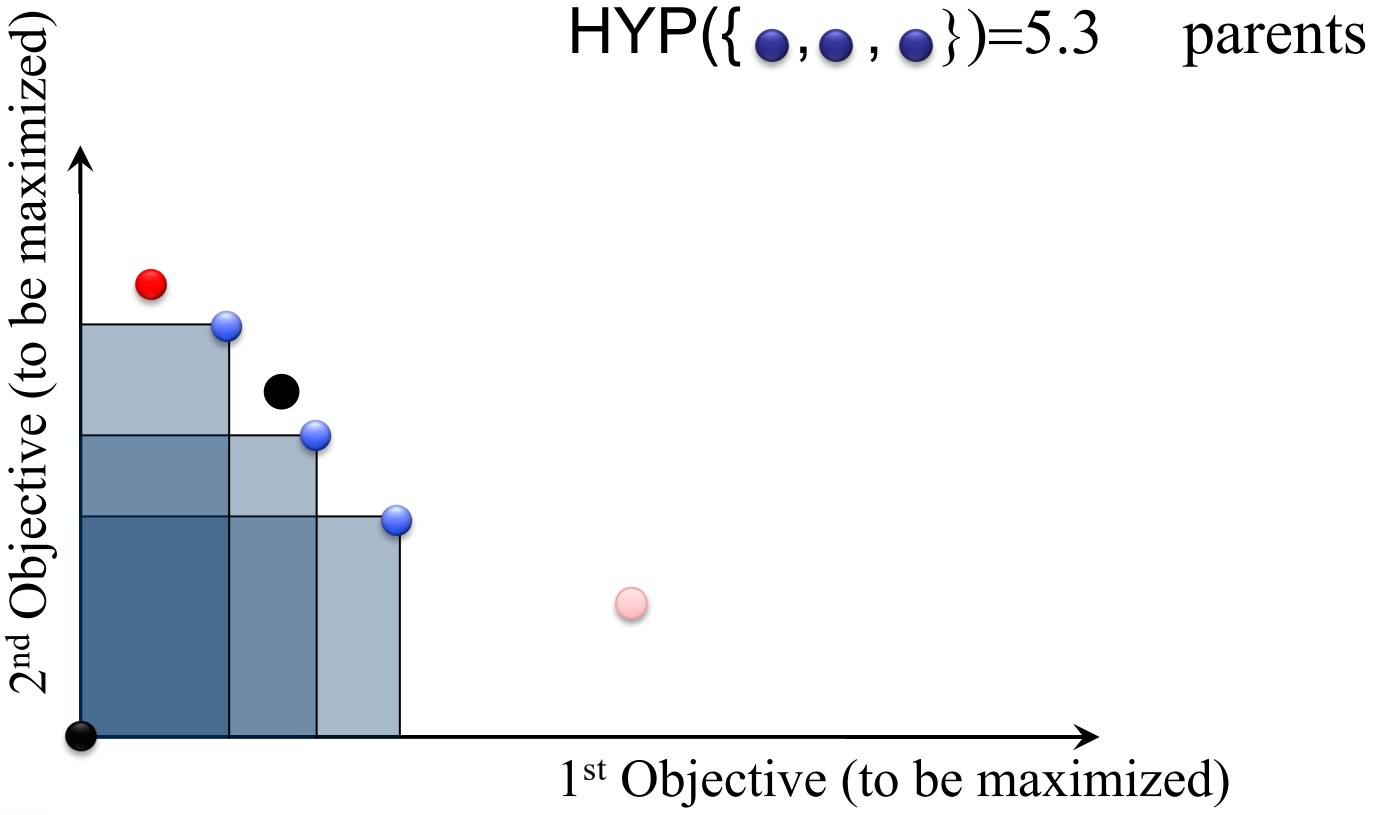
Step 3 (Termination): If $m \geq N$ then output P and stop; otherwise set $m := m+1$.

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

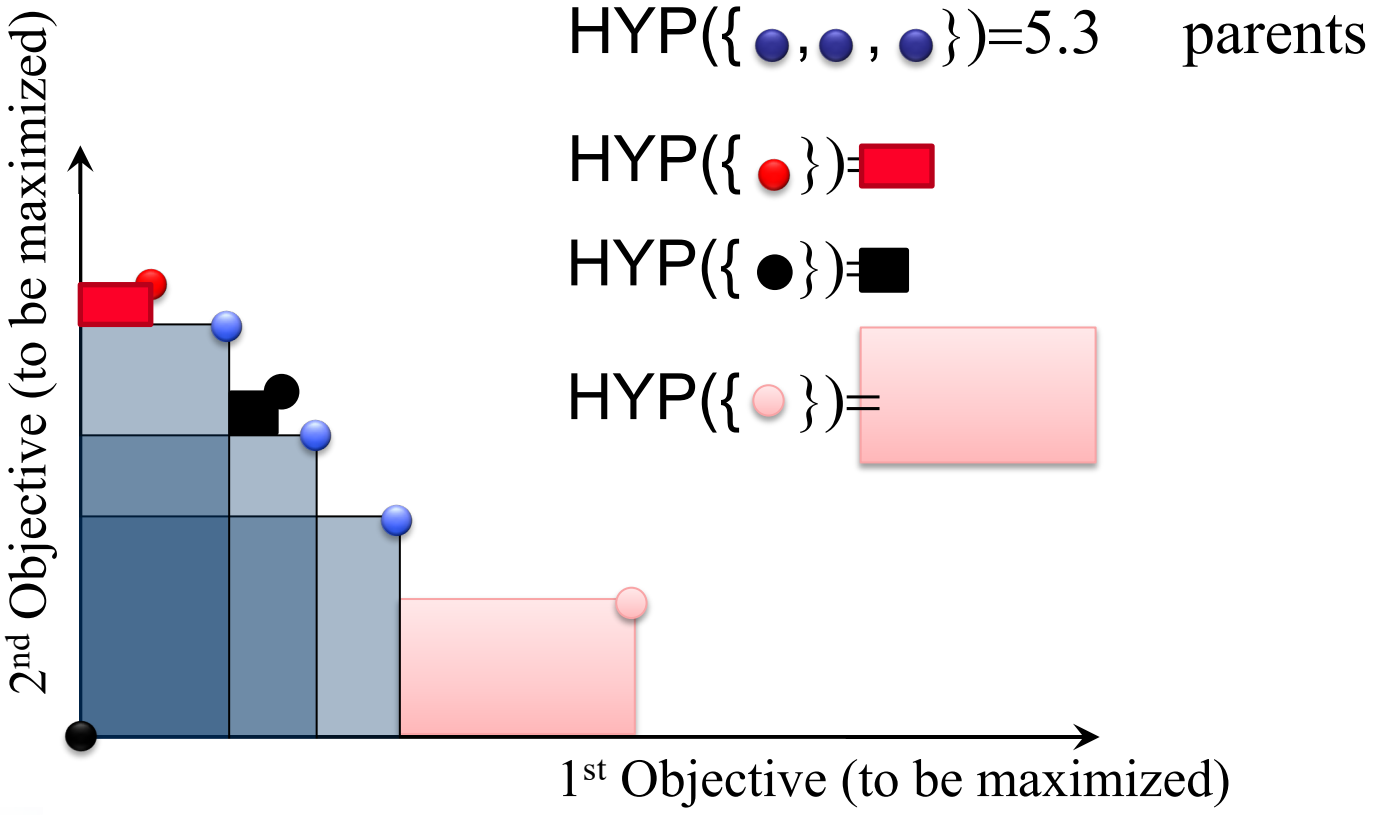
Simple Indicator-based EA

Which child is best?

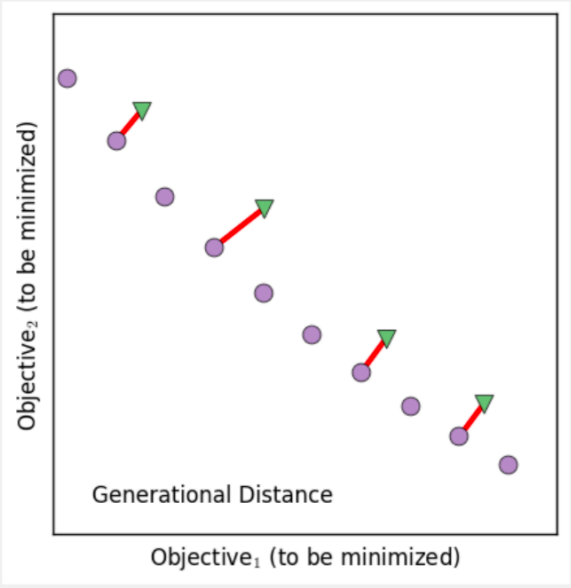


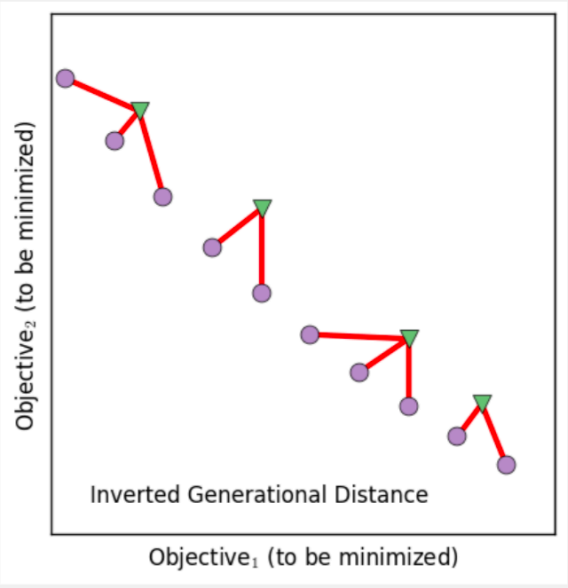
Simple Indicator-based EA

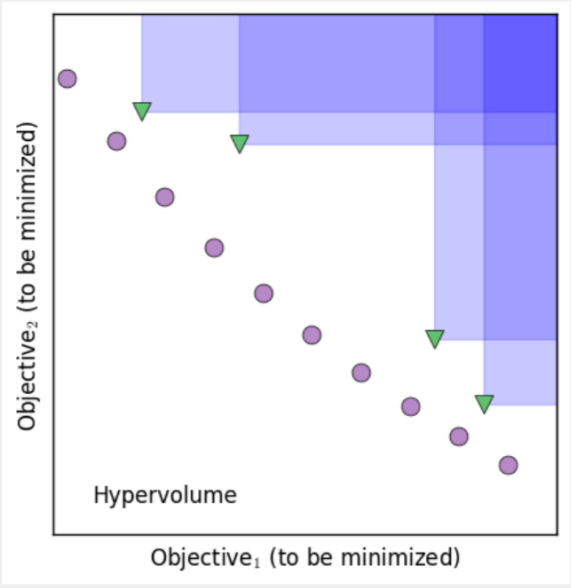
Which child is worst?

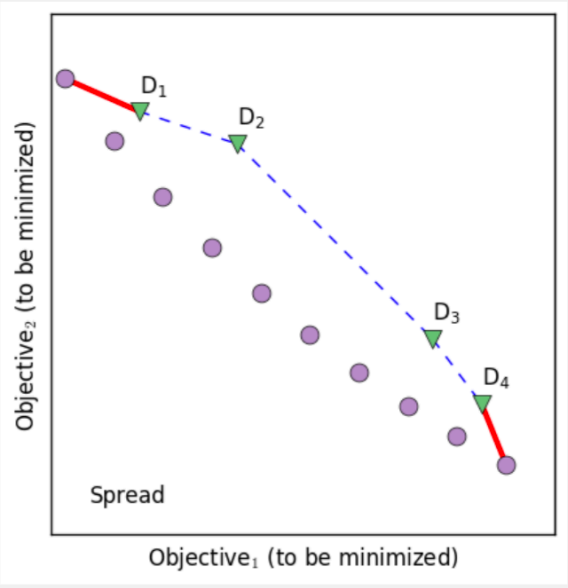


Other indicators









For more information, see page 3 of https://cs.adelaide.edu.au/~markus/pub/2018msr-dse.pdf

Recommendation:
use IGD or Hypervolume or Approximation (not shown here)