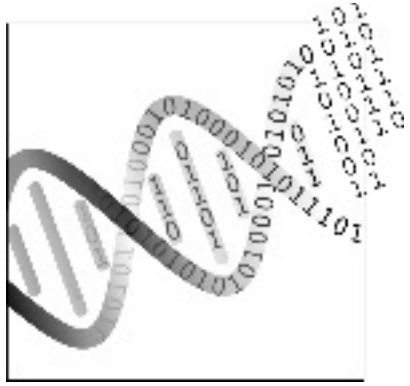
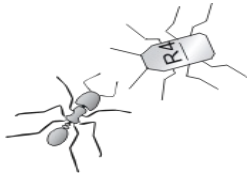
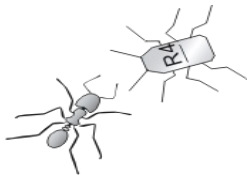
Multi-objective Evolutionary Optimization





What you will learn today

- Single-objective, multimodal, and multiobjective problems
- How to spread solutions on problem space by means of fitness sharing
- Reducing multi-objective problems to single-objective problems by objective weighting
- Caveats of objective weighting
- Non-dominated Sorting Genetic Algorithm (NSGA) algorithm
- NSGA-II algorithm
- How to handle objective constraints in NSGA-II
- Viability Evolution algorithm



Single-objective optimization

The parameter space can be multi-dimensional, and the fitness (objective) function can be polynomial, but the search space admits only one solution that is better or equal to other solutions.

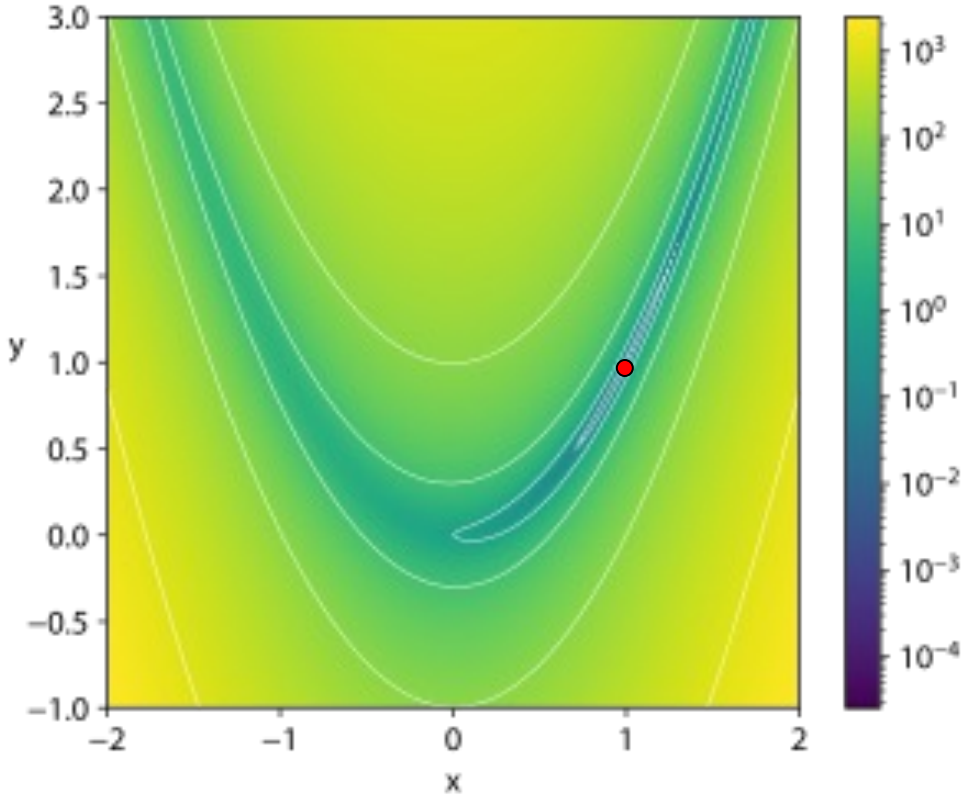
Rosenbrock function of 2 variables

$$f(x,y) = (a-x)^2 + b(y-x^2)^2$$

Our goal is to converge to the optimal solution

For a=1, b=100

$f(x,y)$=0 at x=1, y=1



Single-objective, multimodal function

- If there are several equal optima, simple evolutionary algorithms will converge towards only one.
- We may want to find all solutions with optimal fitness to choose from based on additional criteria.

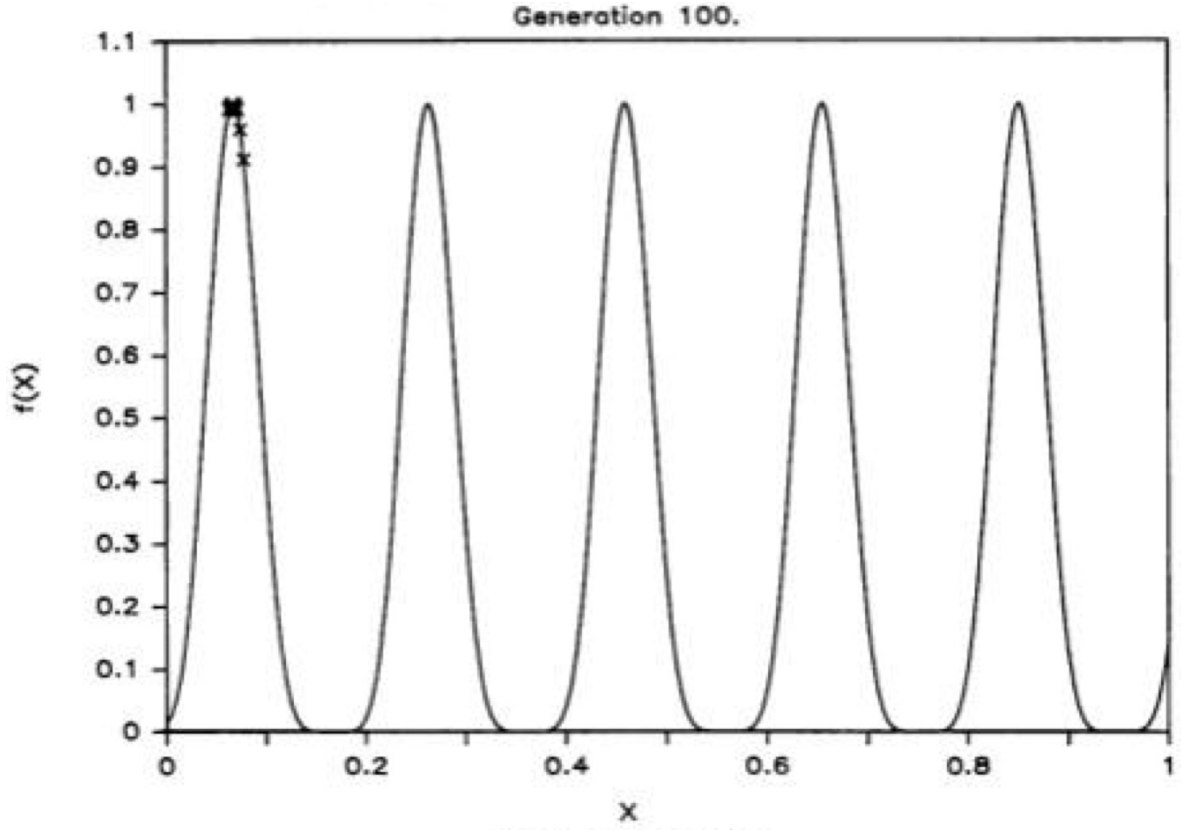
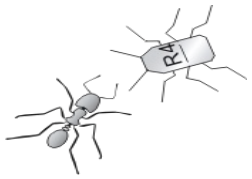


Image from Goldberg, D.E. (1989) Genetic Algorithms, Addison-Wesley



Fitness sharing (Goldberg and Richardson, 1987)

- Maintaining population diversity helps to discover and retain diverse solutions
- Fitness sharing penalizes fitness of an individual as a function of the number of similar individuals.

$$f(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} s\left(d\left(x_i, x_j\right)\right)}$$

Distance can be computed at the phenotype level or at the genotype level (for example, Euclidian or Hamming distance).

Example: linear sharing function

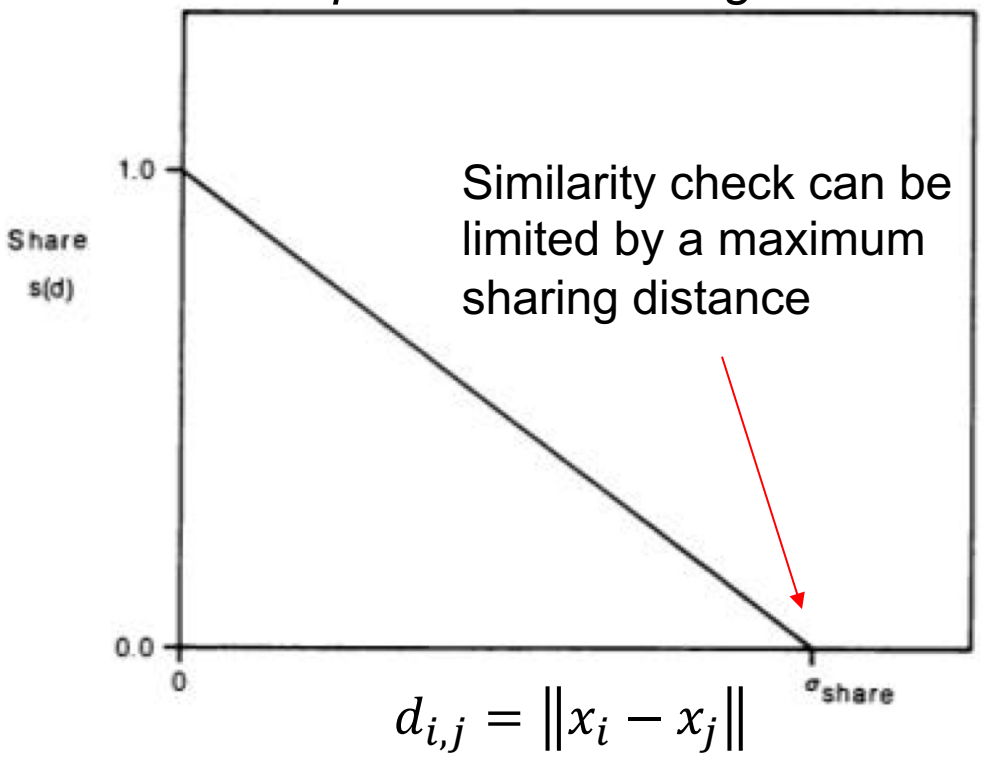
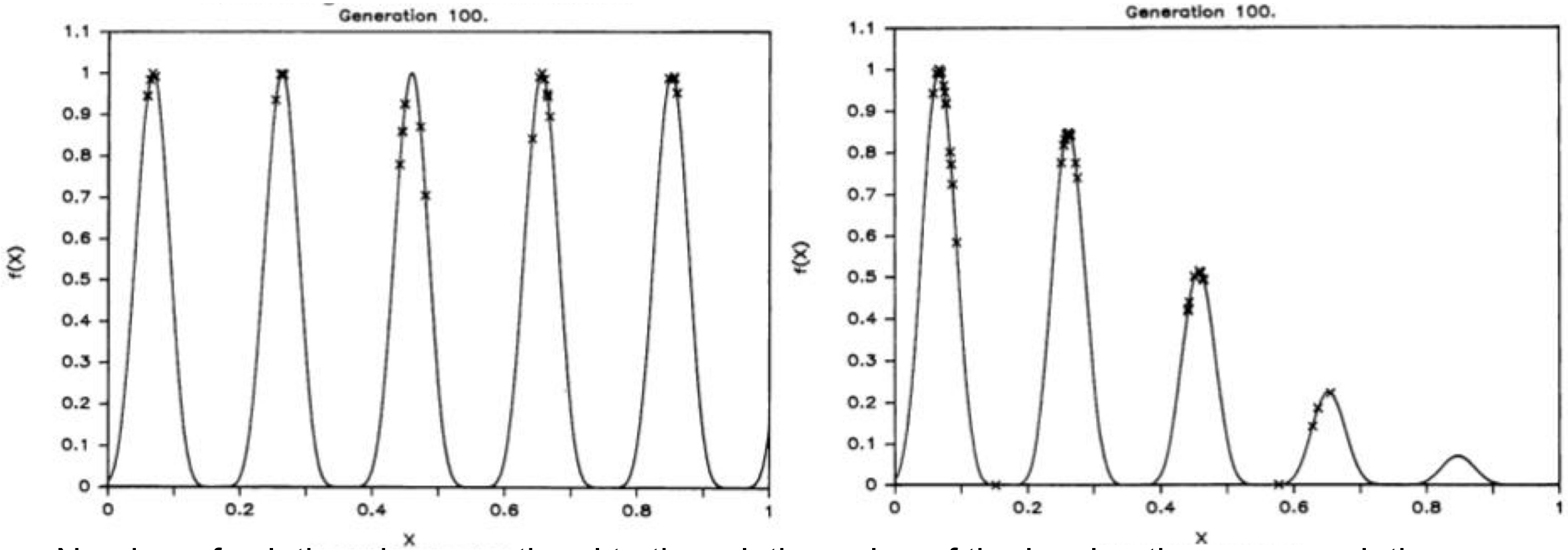


Image from Goldberg, D.E. (1989) Genetic Algorithms, Addison-Wesley

Fitness sharing in multimodal functions

Fitness sharing maintains population diversity and finds multiple solutions in multimodal functions



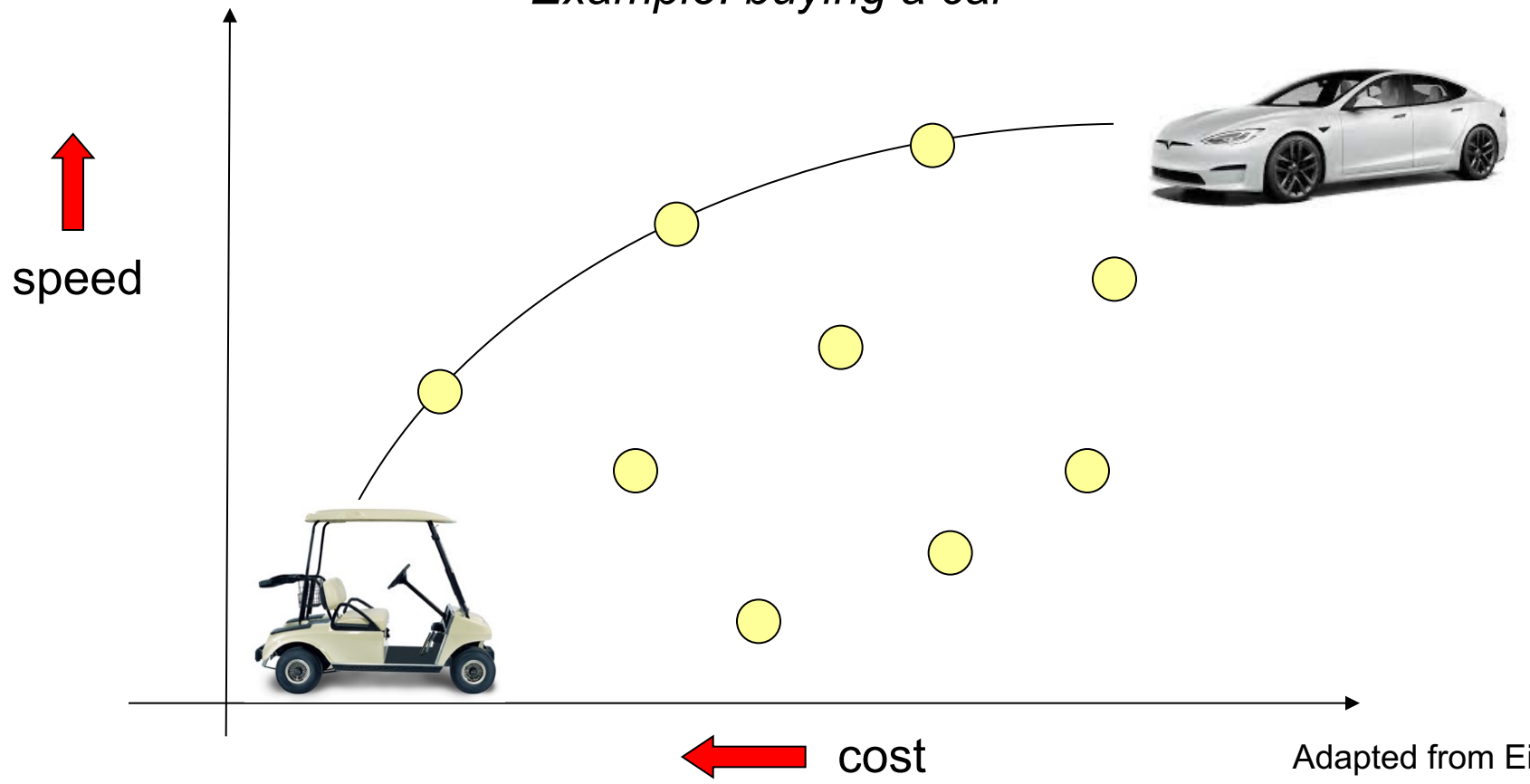
Number of solutions is proportional to the relative value of the local optima: more solutions are allocated to better optima

Image from Goldberg, D.E. (1989) Genetic Algorithms, Addison-Wesley

Multi-objective optimization

- Many real-world problems present multiple objectives that are not commensurable or compatible
- There may not exist one solution that is superior to others with respect to all objectives
- We want to identify all the solutions that are the “best compromise”

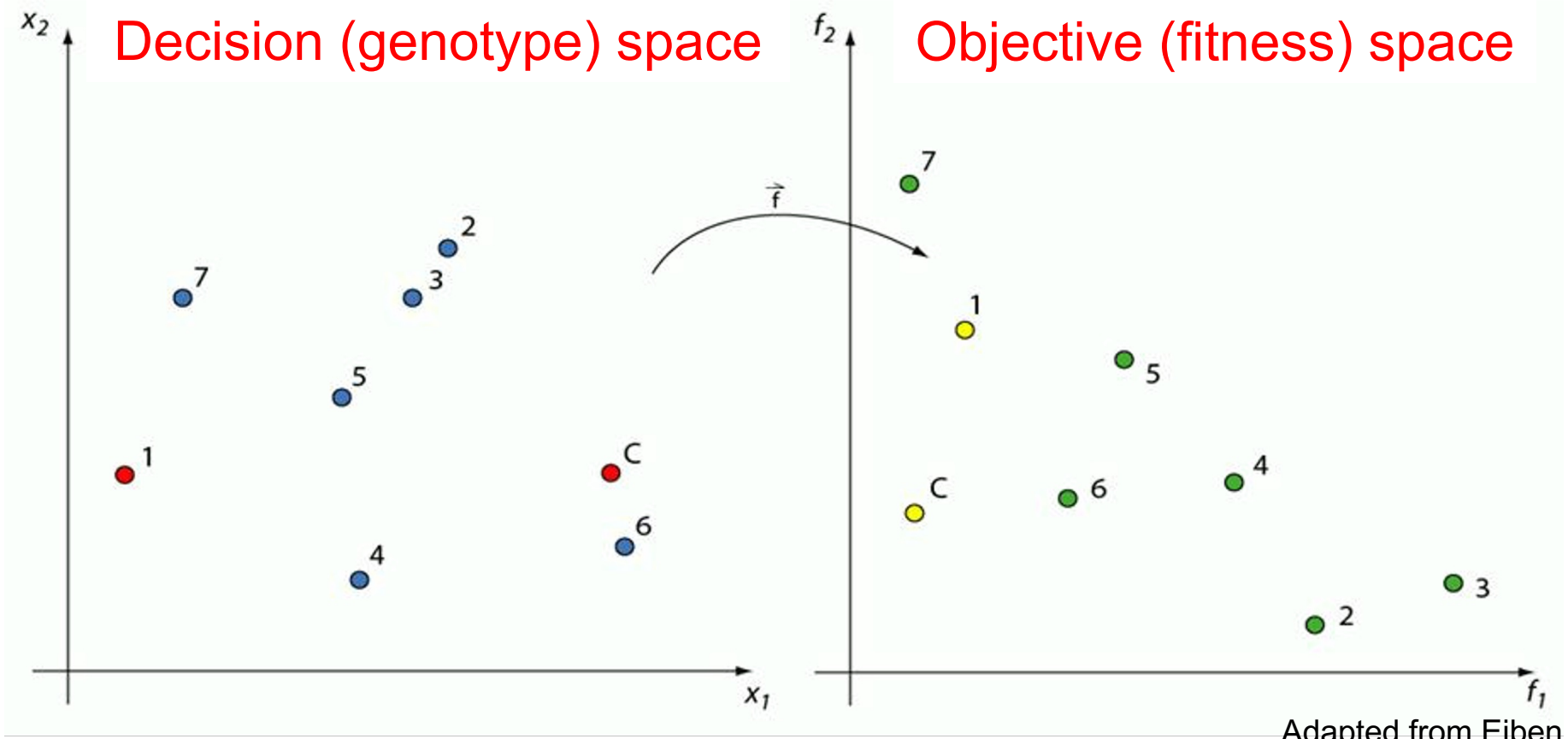
Example: buying a car



Adapted from Eiben & Smith, 2015

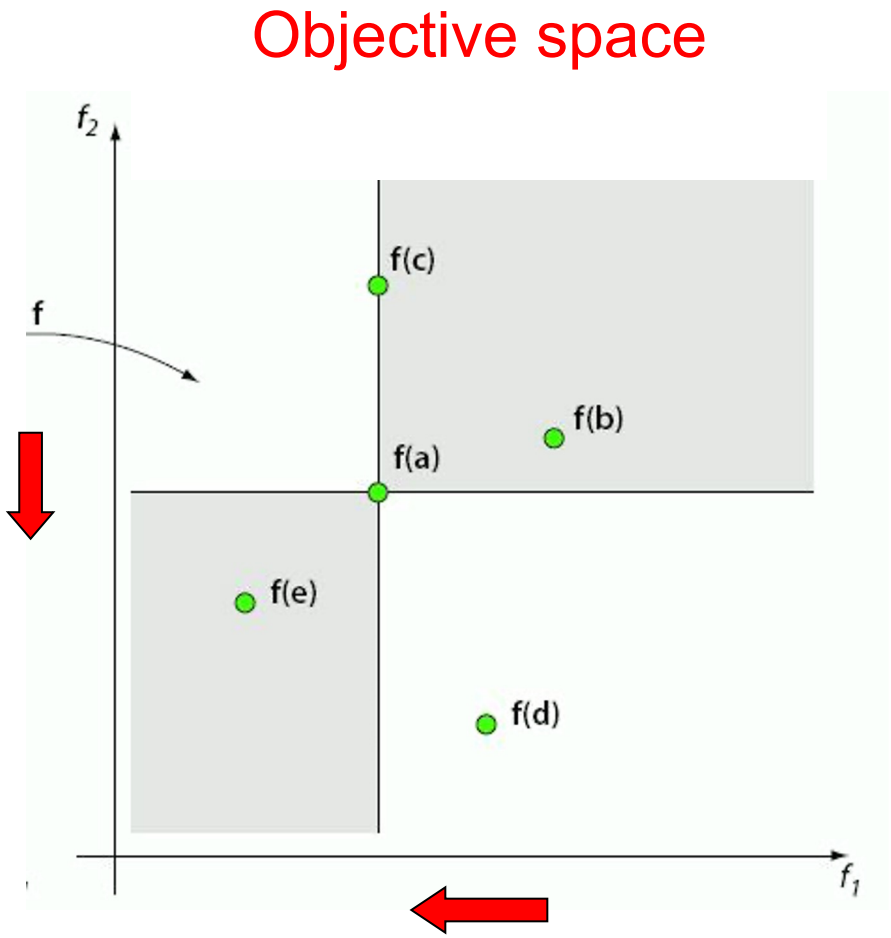
Decision and objective spaces

- It is convenient to distinguish between decision space (genotype) and objective space (fitness)
- We assume a one-to-one mapping between decision space and objective space



Adapted from Eiben & Smith, 2015

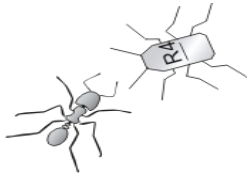
Comparing solutions in the objective space



- Optimisation task:
  Minimize both f_1 and f_2

- Consider solution a:
  a is better than b
  a is better than c
  a is worse than e
  a and d are not comparable

Adapted from Eiben & Smith, 2015



Dominance relation

In multi-objective optimization we use the concept of dominance relation between solutions in the objective space.

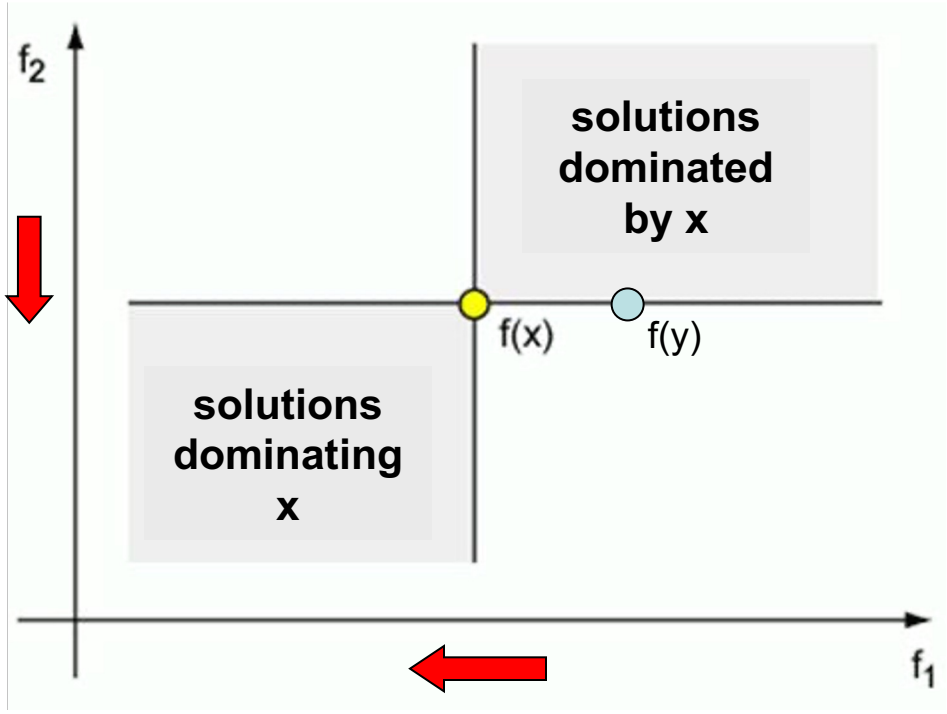
For a minimization problem with *k* objectives, solution *x* dominates solution *y* if:

1. $\forall i \in \{1, \ldots, k\}, f_i(x) \leq f_i(y)$

f(x) is better or equal to f(y) for all *k* objectives

2. $\exists i \in \{1, \ldots, k\}, f_i(x) < f_i(y)$

f(x) is better than f(y) in at least one objective



Adapted from Eiben & Smith, 2015

Pareto Optimal Set and Pareto Frontier

Given a set of feasible solutions Q, we want to find solutions that are **not dominated** by any other solution in the Q set.

Example
A and B are not-dominated because there is no other solution that dominates them
C is dominated by both A and B

- The set of non-dominated solutions belong to the Pareto Optimal Set

- These solutions define the Pareto Frontier because there is no other solution that can improve without degrading some objectives

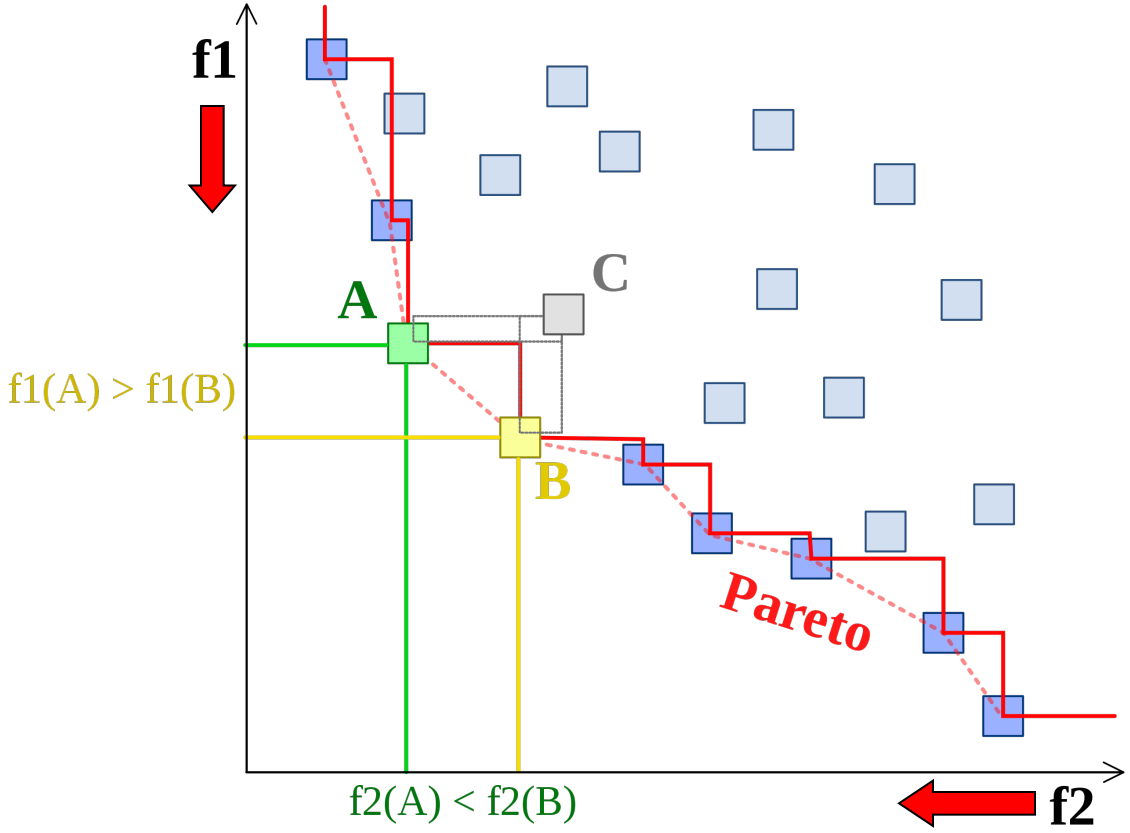
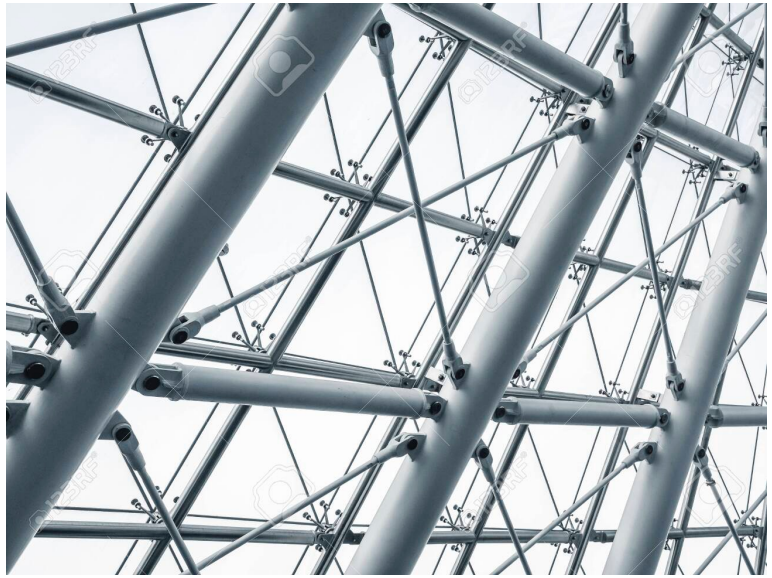
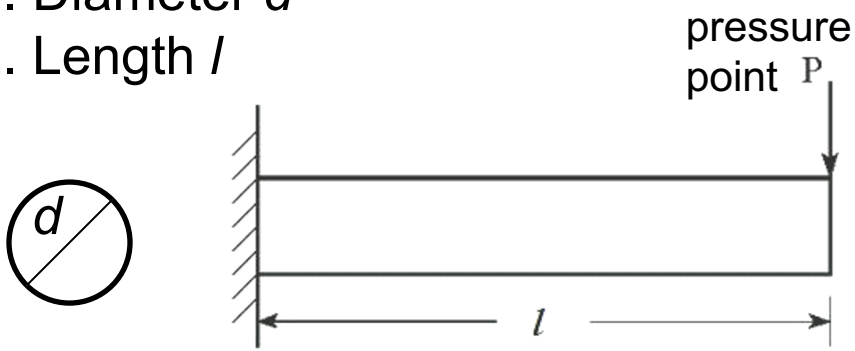


Image: Johann Dréo

Minimize steel beam's weight and deflection (Deb, 2001)



Two decision variables:
1. Diameter *d*
2. Length *l*



$$\text{minimize} \quad f_1(d,l) = \rho \frac{\pi d^2}{4} l \qquad \textit{weight}$$

$$\text{minimize} \quad f_2(d,l) = \delta = \frac{64 P l^3}{3 E \pi d^4} \qquad \textit{deflection}$$

constraints

$$0.01 \text{ m} \le d \le 0.05 \text{ m} \qquad \textit{diameter}$$

$$0.2 \text{ m} \le l \le 1.0 \text{ m} \qquad \textit{length}$$

$$\sigma_{\max} = \frac{32 P l}{\pi d^3} \le S_y \qquad \textit{max stress}$$

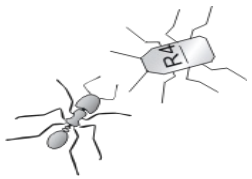
$$\delta \le \delta_{\max} \qquad \textit{max deflection}$$

where

$$\rho = 7800 \text{ kg/m}^3, \; P = 2 \text{ kN}$$

$$E = 207 \text{ GPa} \qquad \textit{modulus of elasticity}$$

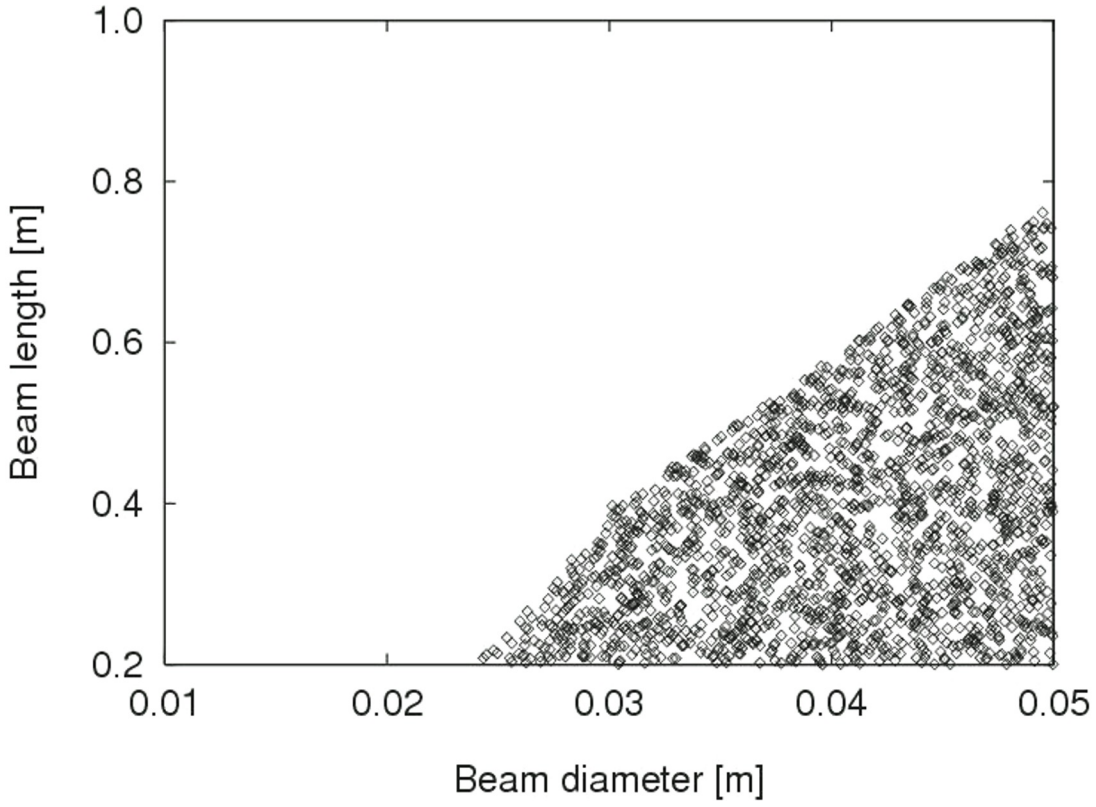
$$S_y = 300 \text{ MPa}, \; \delta_{\max} = 0.005 \text{ m}$$



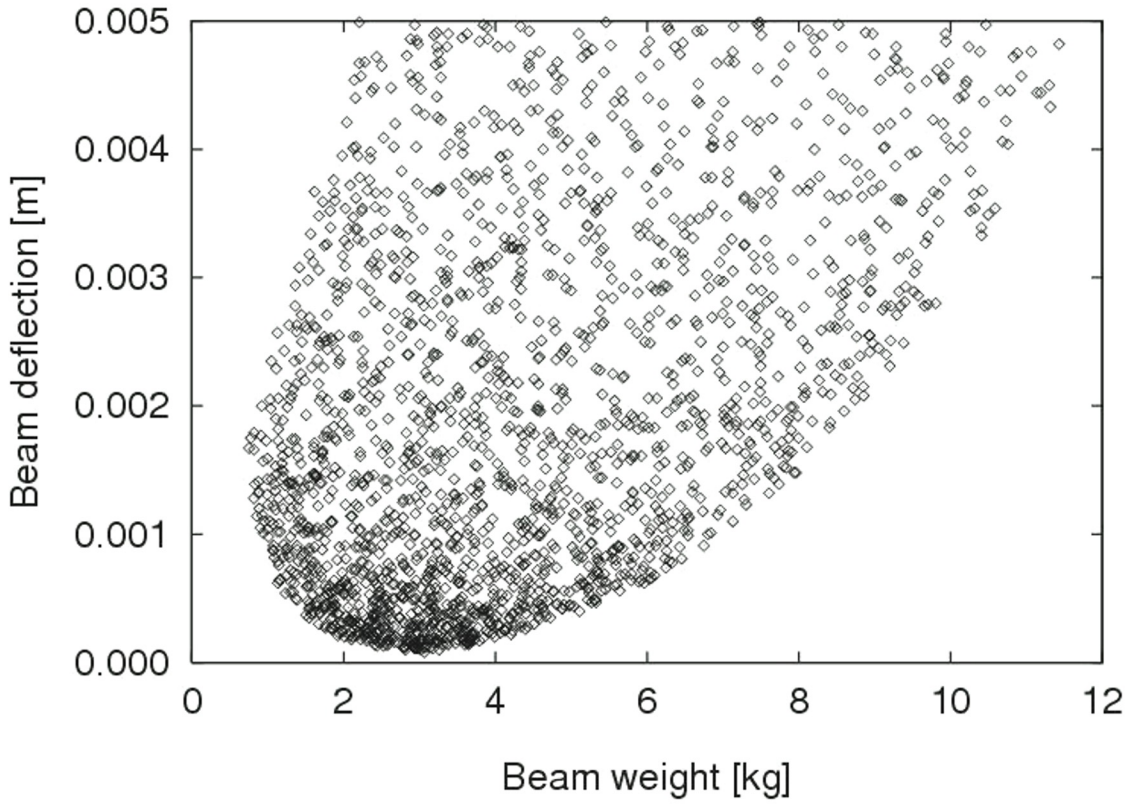
Feasible solutions

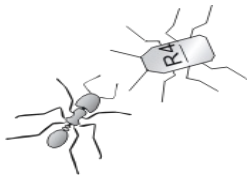
Feasible solutions are solutions that fulfil the constraints

Decision space



Objective space

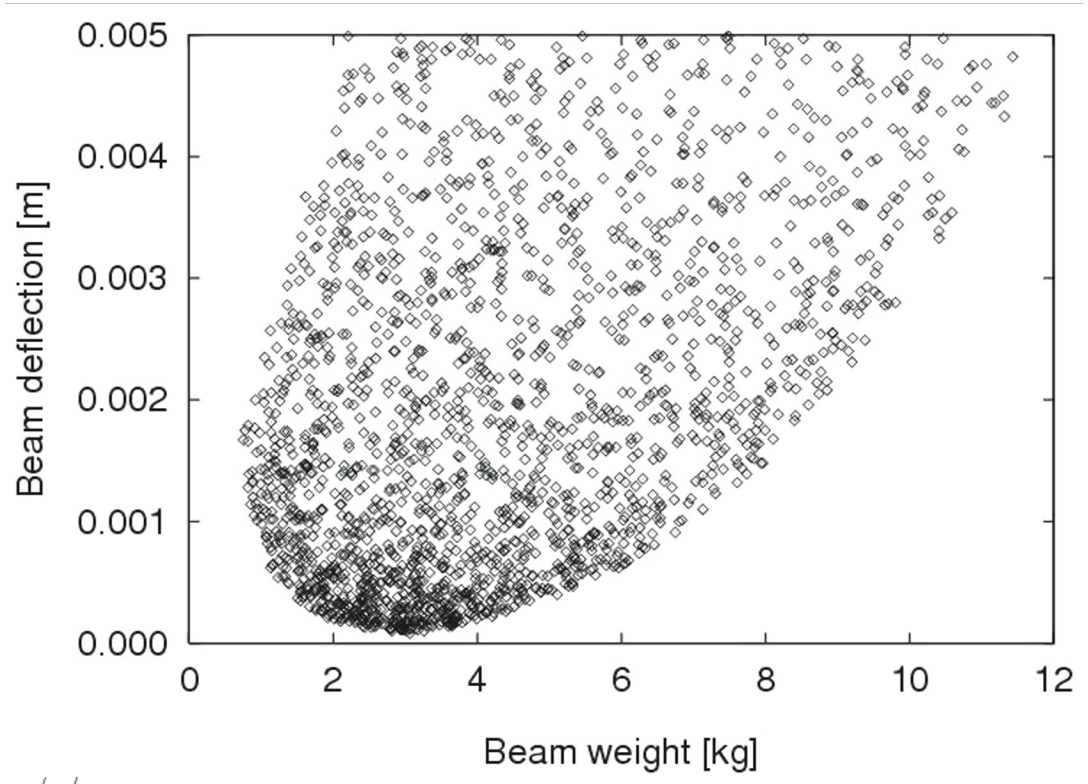


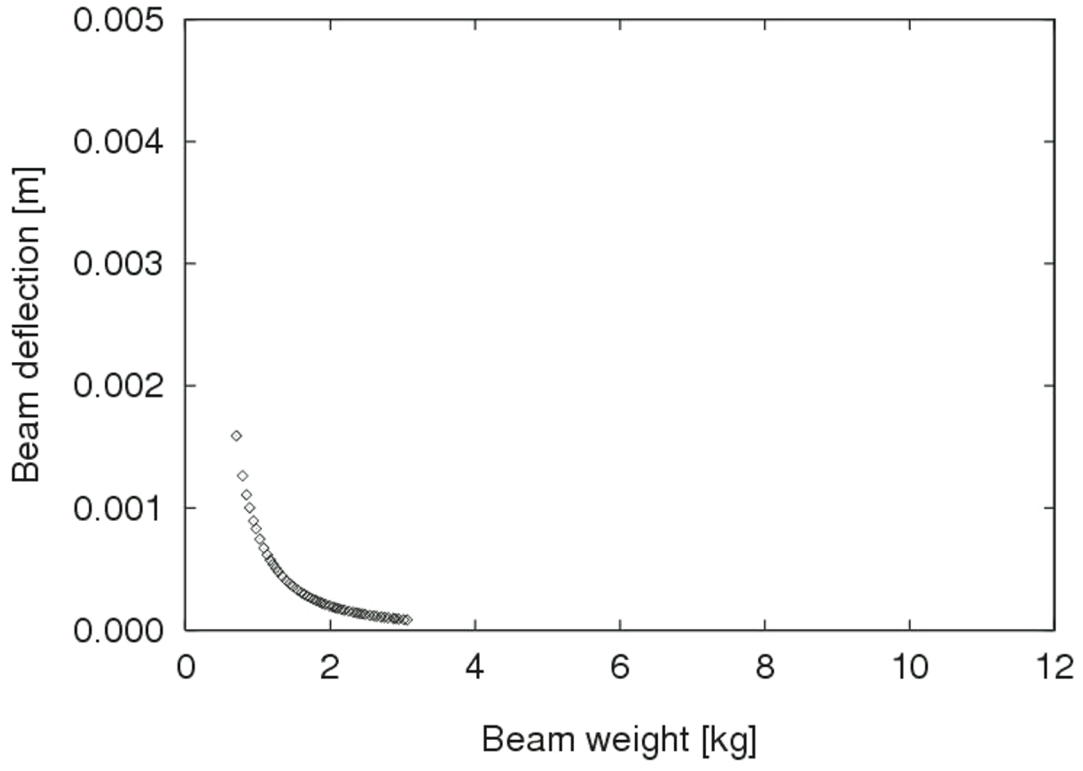


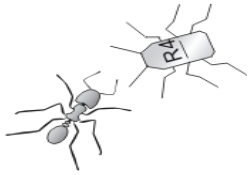
Goal: Finding non-dominated solutions

Find a set of non-dominated solutions following the criteria of:

- convergence (as close as possible to the Pareto-optimal front),
- diversity (spread over the Pareto front)



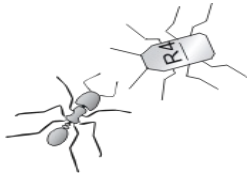




Single- vs. Multi-objective optimisation

Characteristic	Single-objective optimisation	Multi-objective optimisation
Number of objectives	one	more than one
Spaces	one	two: decision space, objective space
Comparison of candidate solutions	*x* is better than *y*	*x* dominates *y*
Result	one (or several equally good) solution(s)	Pareto-optimal set
Algorithm‘s goals	convergence	convergence, diversity

Adapted from Eiben & Smith, 2015



Classical Multi-objective Optimization: Objective Weighting

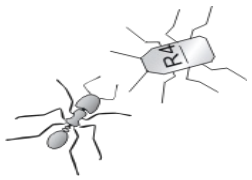
Classical methods scalarize the objective vector $\mathbf{f}(\mathbf{x})$ into a single objective $Z(\mathbf{x})$ and find a compromise solution subjected to some constraints.

$$Z(\mathbf{x}) = \sum_{i=1}^{N} w_i f_i(\mathbf{x})$$
$$\text{for } (\mathbf{0} \leq w_i \leq 1), \sum_{i=1}^{N} w_i = 1$$

where $\mathbf{x} \in \mathbf{X}$ represents the set of feasible solutions

1. Find optimal solution for each objective function separately
2. For ech solution, compute values of all objective functions
3. Find desired solution by objective weighting, e.g. $Z(\mathbf{x}) = 0.7 f_1(\mathbf{x}) + 0.3\ f_2(\mathbf{x})$

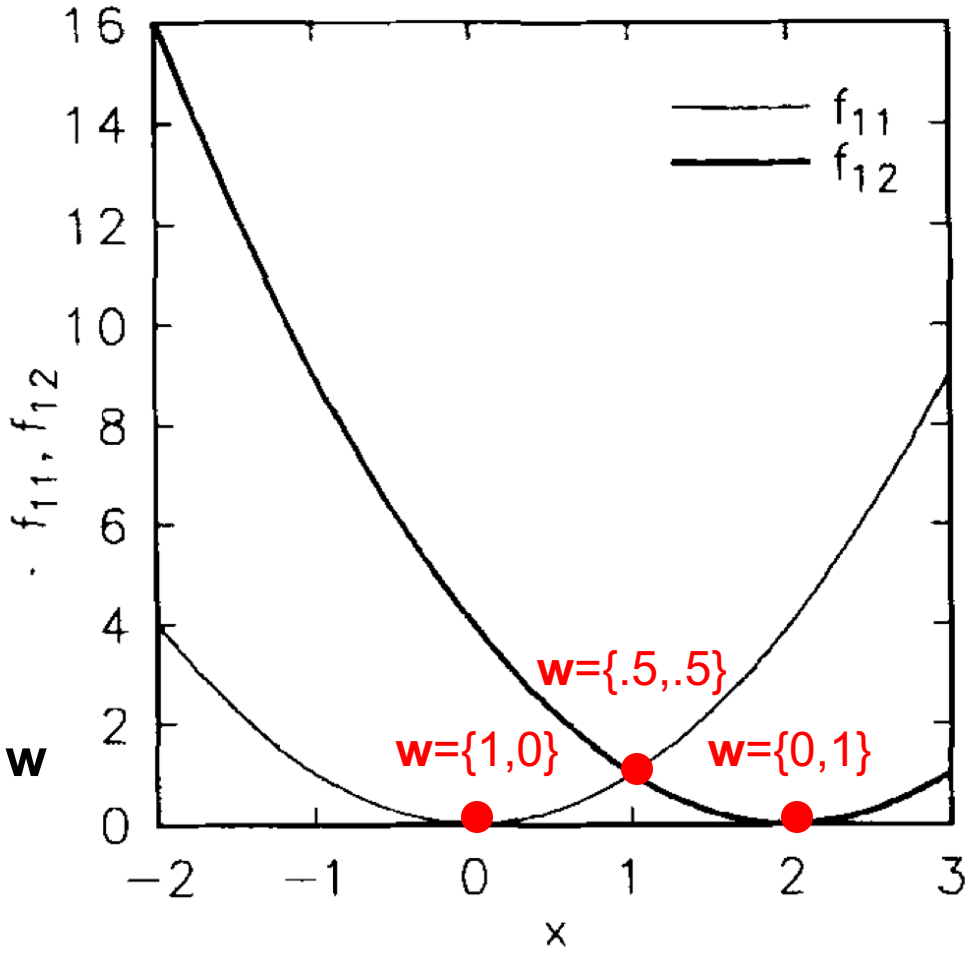
Objective Weighting may find a Pareto optimal solution



Example: Objective Weighting

Let's consider a simple problem with two objective functions of one variable

minimize $f_{11} = x^2$
minimize $f_{12} = (x-2)^2$

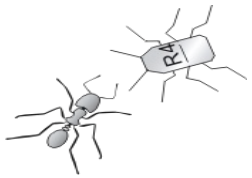


Pros:

- Each found solution is a Pareto optimum
- Weight vector allows control of objective priorities

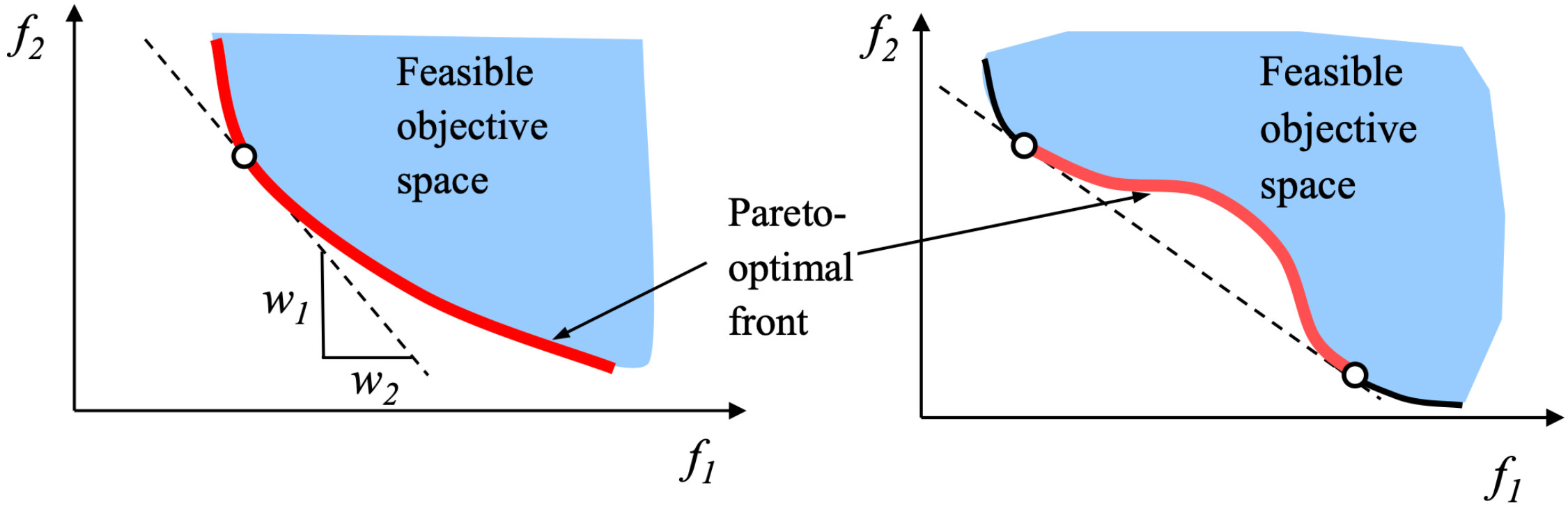
Cons:

- Only one solution at a time is found
- For several objectives it is not obvious how to choose **w**



Objective weighting and Pareto front convexity

The choice of objective weights defines the optimal point on the Pareto Front tangent. Therefore, objective weighting can work only in convex Pareto Fronts.



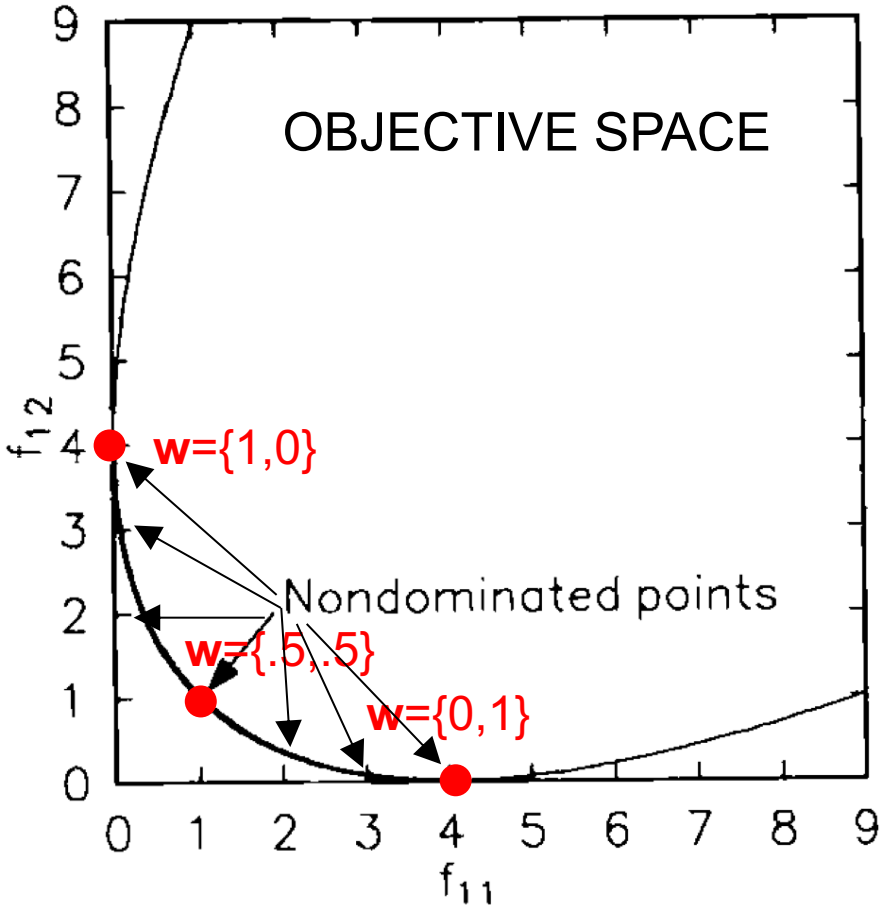
Source: K. Deb

Exploring the Pareto front(ier)

1. We want to find at once several solutions that lie on the Pareto front

2. We want the solutions to be equally distributed on the Pareto front

Evolutionary algorithms are perfect candidates for exploring the Pareto front because they maintain a population of solutions



Nondominated Sorting Genetic Algorithm (NSGA)

Srinivas & Deb, 1994

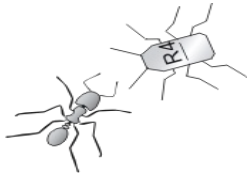
1. Identify F Pareto fronts of decreasing order until all individuals are allocated to a front F_i

2. Assign higher dummy fitness to higher fronts; all individuals in a front have same dummy fitness

3. To maintain diversity, apply fitness sharing to individuals in a front

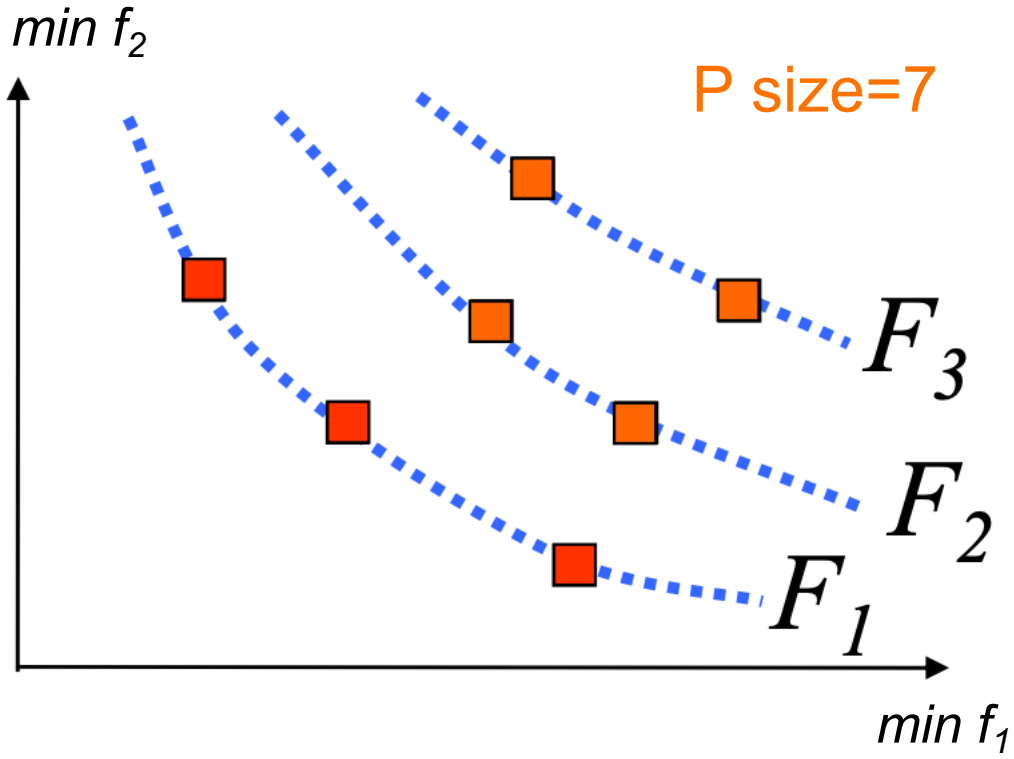
4. Apply proportionate selection to ensure that individuals in higher order fronts make proportionally more offspring

5. Apply mutation and crossover to offspring



NSGA: Identification of F Pareto fronts

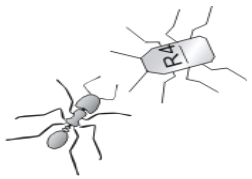
1. Find list of non-dominated solutions in Population and assign them to F_1
2. Temporarily remove F_1 solutions from Population
3. Find list of non-dominated solutions in Population and assign them to F_2
4. Repeat steps 2-3 until all F fronts have been identified (all solutions in Population have been allocated to a front F_i)



NSGA: Fitness Sharing

n = number of identified fronts in current population
i=1 // let's start with solutions in highest ranking Front F_1
r=999 // arbitrarily high dummy fitness
While $i \leq n$ // all fronts have been visited

1. Assign identical high dummy fitness r to all solutions in F_i
2. Fitness sharing: compute shared fitness s of each solution (divide fitness of individual solutions in F_i by quantity proportional to number of other solutions in their neighborhood)
3. Set r to be smaller than smallest shared fitness in F_i
4. i++

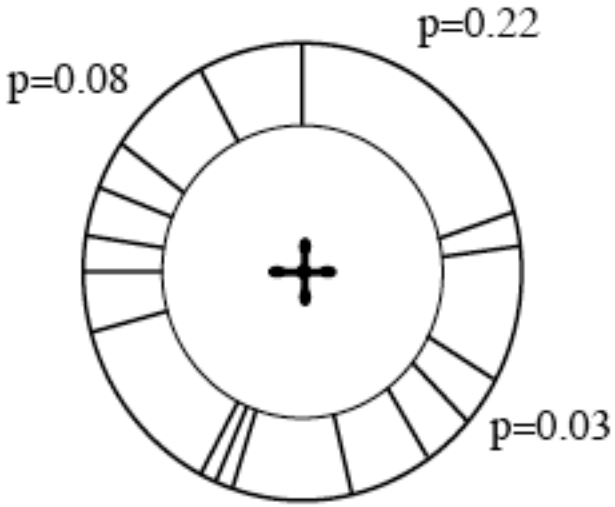


NSGA: Create new population

1. Proportionate selection

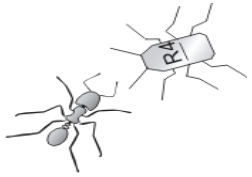
$$p_i = \frac{s_i}{\sum_{i=1}^{N} s_i}$$

where s_i is the shared fitness of individual *i*

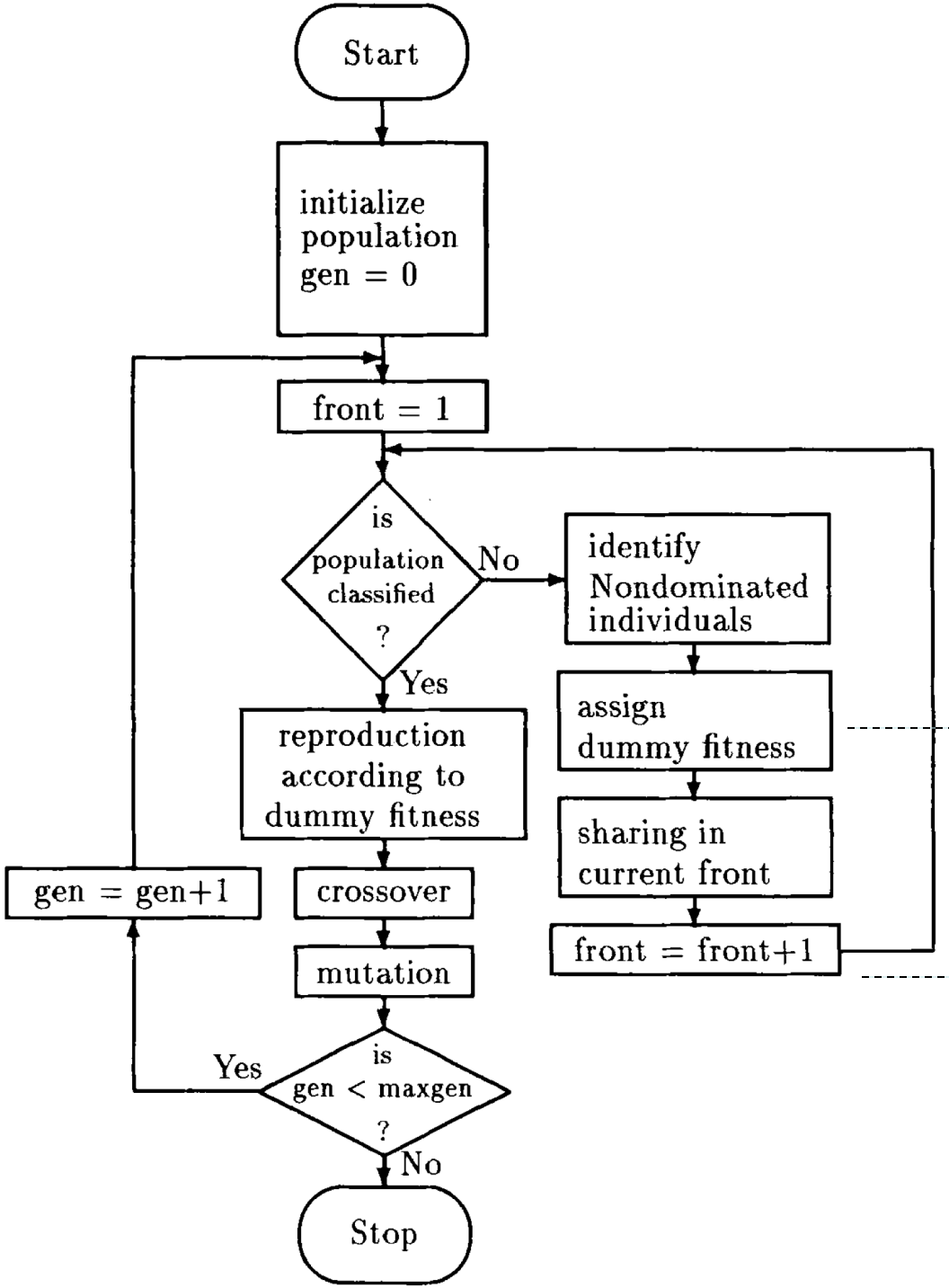


2. Mutation of newly created individuals

3. Crossover of newly created individuals



NSGA: Flowchart



If front>1, dummy fitness < smallest shared fitness of previous front

Temporarily remove nondominated individuals from population

Adapted from: Srinivas & Deb, 1994

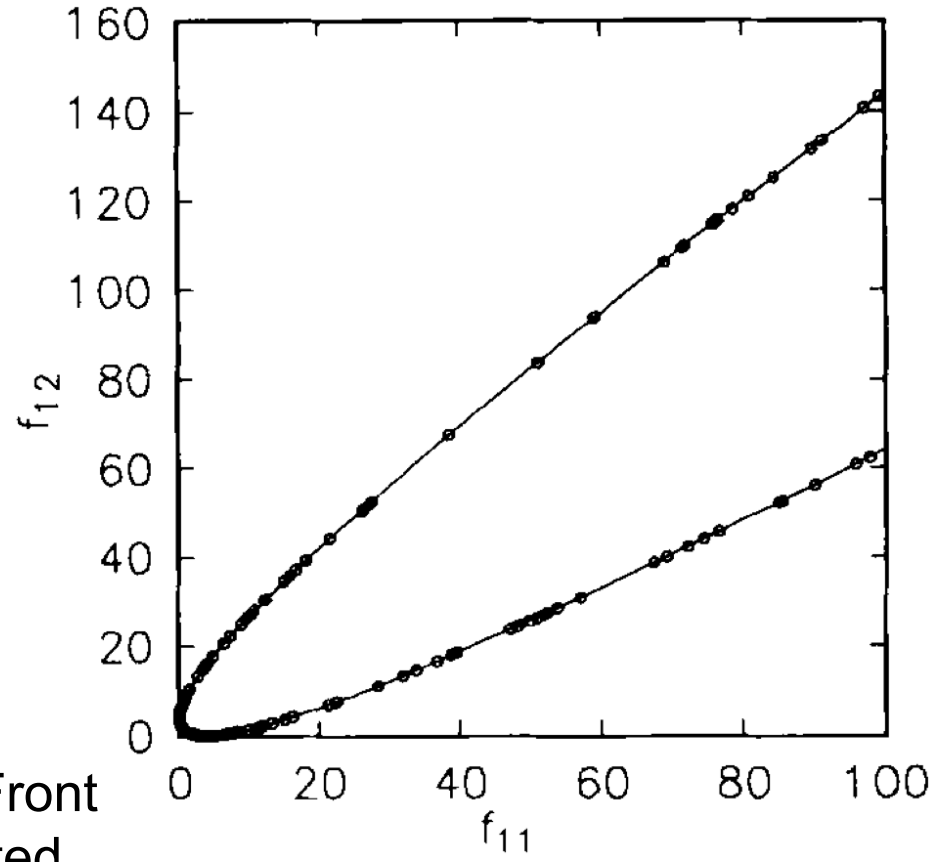
NSGA on 2-objective function of 1 variable

minimize $f_{11} = x^2$
minimize $f_{12} = (x-2)^2$

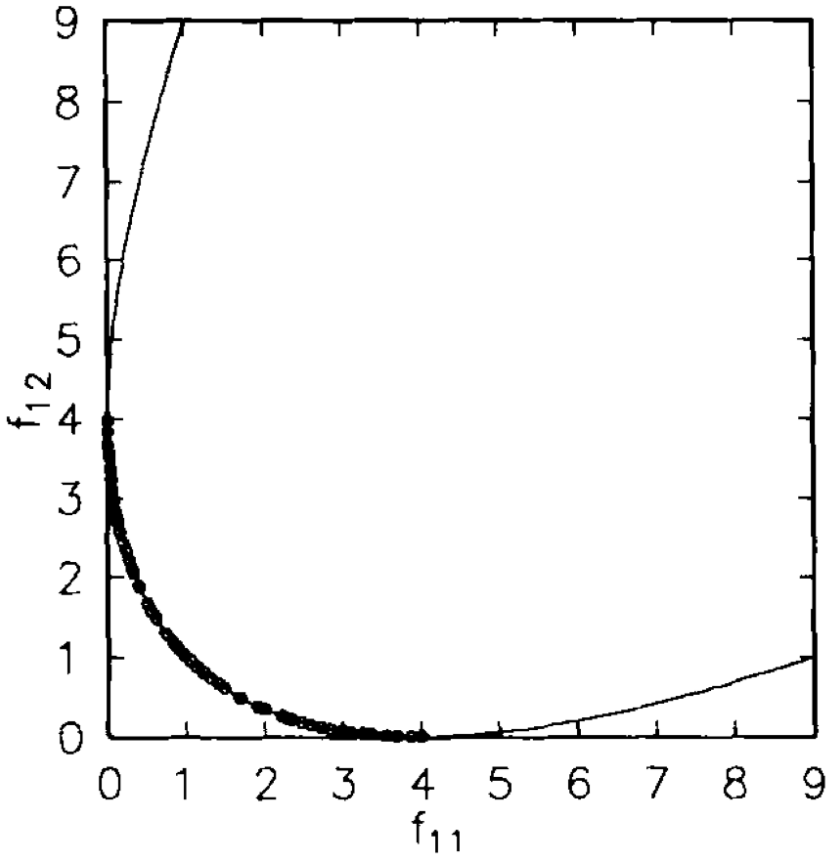
Max gen	500
Population size	100
String length (binary)	32
Probability crossover	1.0
Probability mutation	0.0*

*mutation kept to 0 to highlight algorithm's search performance

Generation 1

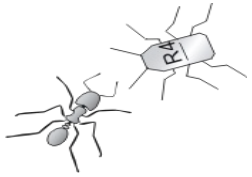


Generation 500



Results:
- Solutions aligned on Pareto Front
- Solutions are equally distributed

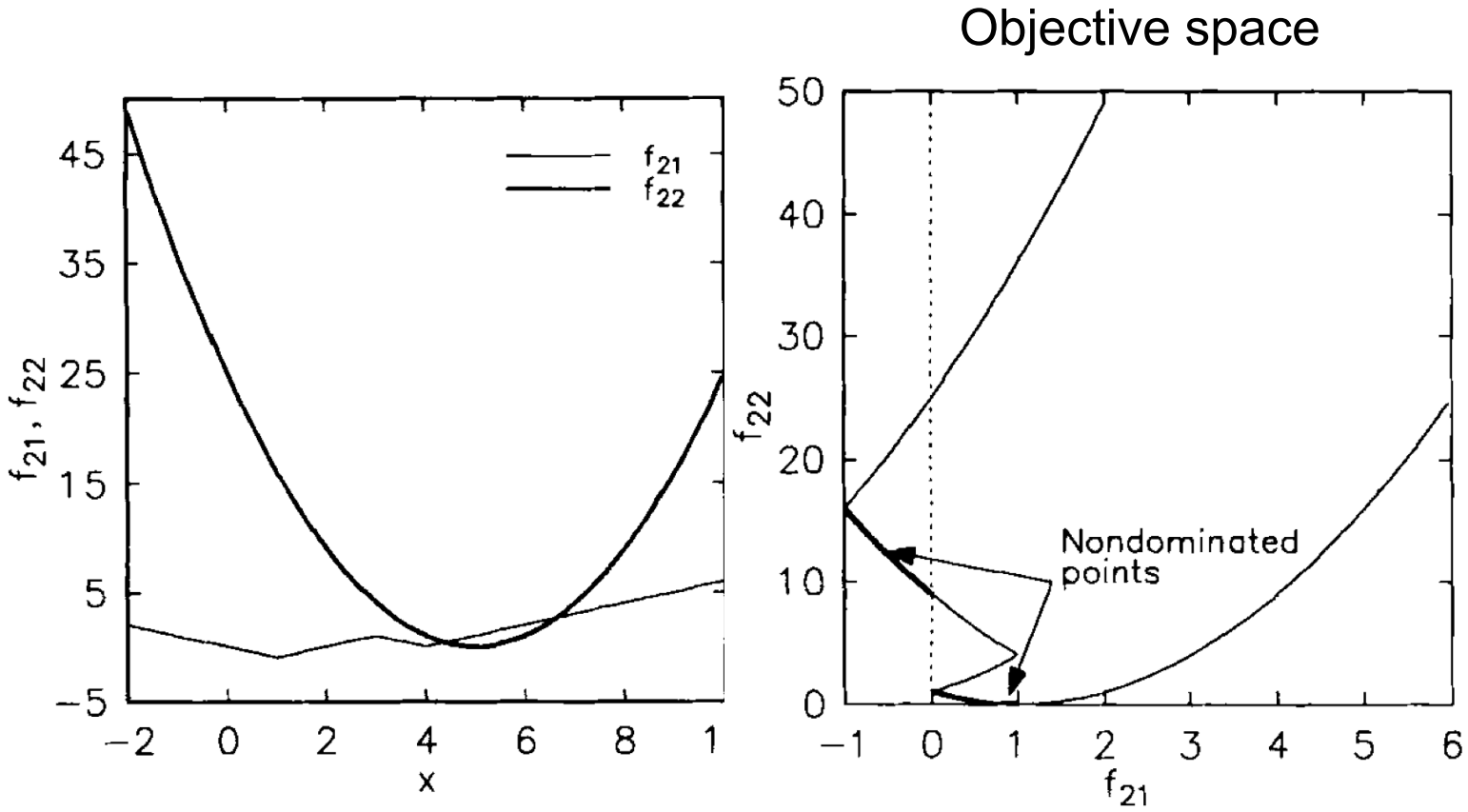
Source: Srinivas & Deb, 1994



2-objective, 1 variable, non-convex and disjointed space

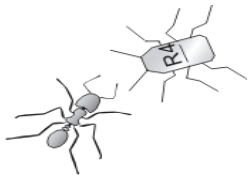
$$\text{Min } f_{21}(x) = \begin{cases} -x, & x \leq 1 \\ -2 + x, & 1 < x \leq 3 \\ 4 - x, & 3 < x \leq 4 \\ -4 + x, & x > 4 \end{cases}$$

$$\text{Min } f_{22}(x) = (x - 5)^2$$



Remember that objective weighting does not work well on this space

Source: Srinivas & Deb, 1994



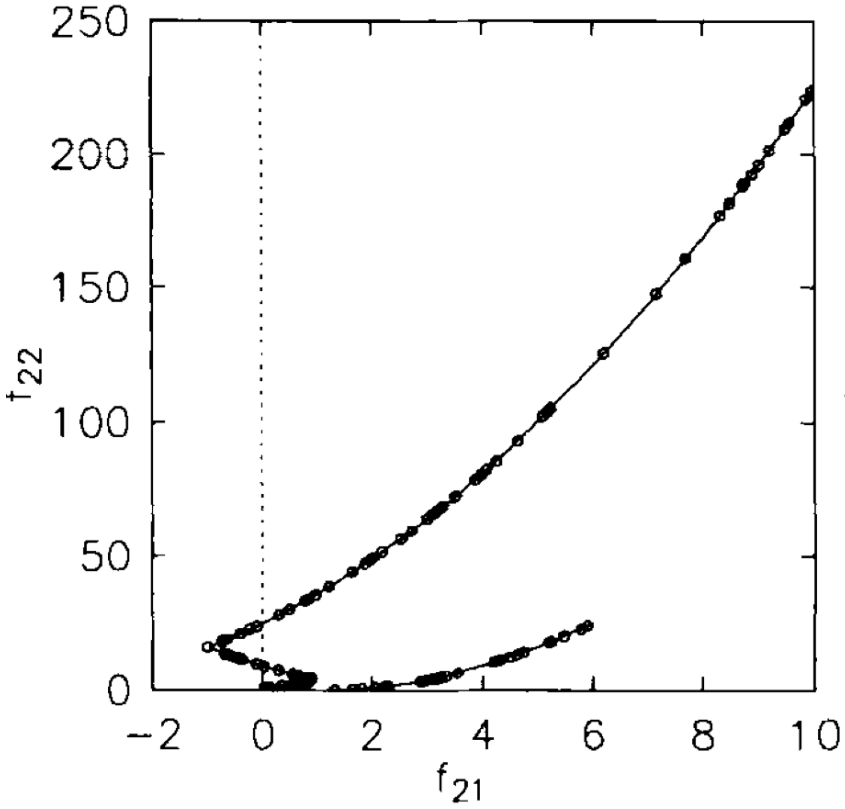
NSGA

Max gen	500
Population size	100
String length (binary)	32
Probability crossover	1.0
Probability mutation	0.0*

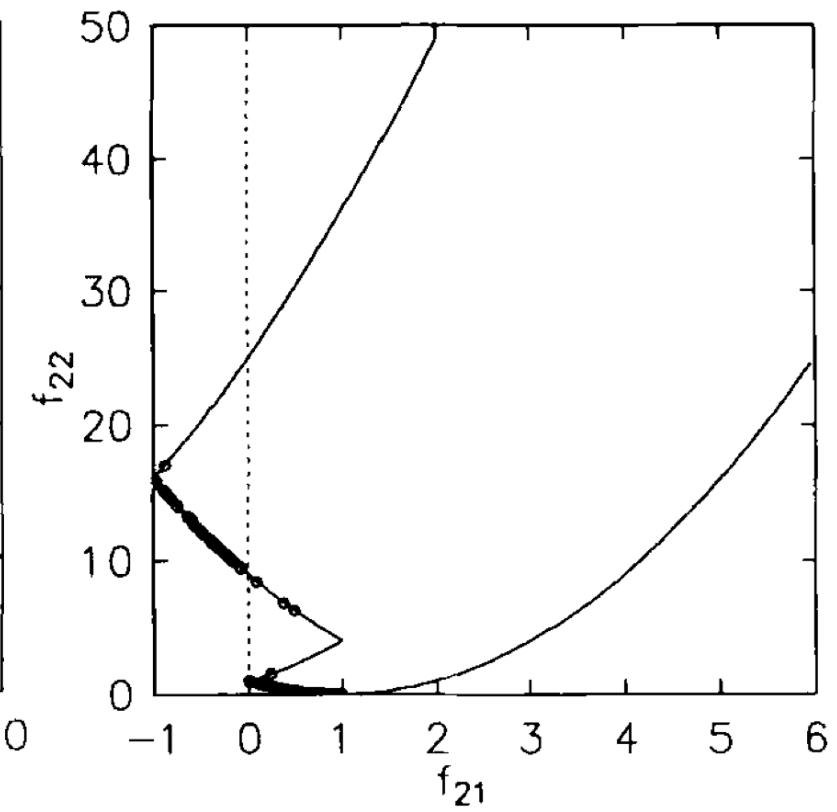
*mutation kept to 0 to highlight algorithm's search performance

Results:
- Solutions aligned on Pareto Front
- Solutions are equally distributed

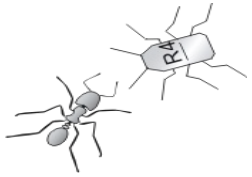
Generation 1



Generation 500



Source: Srinivas & Deb, 1994

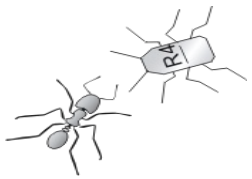


NSGA limitations

1. High computational complexity of nondominated sorting, up to $O(MN^3)$, where M is the number of objectives and N is the population size, when each front is occupied by only 1 solution

2. Strong sensitivity to choice of fitness sharing neighborhood

3. Lack of elitism may lead to loss of Pareto-optimal solutions



NSGA-II: Fast domination sorting

For each solution *p*, two entities are computed:

1. Scalar: Dominance count n_p = number of solutions that dominate solution *p*
2. Matrix: Set S_p = set of solutions *q* that *p* dominates

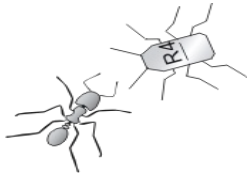
- As a result, all solutions that are nondominated will have $n_p=0$ and are assigned to the Pareto front F_1

1. For each solution with $n_p=0$, visit the q solutions in its S_p and reduce their n_p count by 1

2. As a result, all q solutions with $n_p=0$ are non dominated by remaining solutions and are assigned to Pareto front F_2

- Repeat steps 1-2 until all fronts are identified.

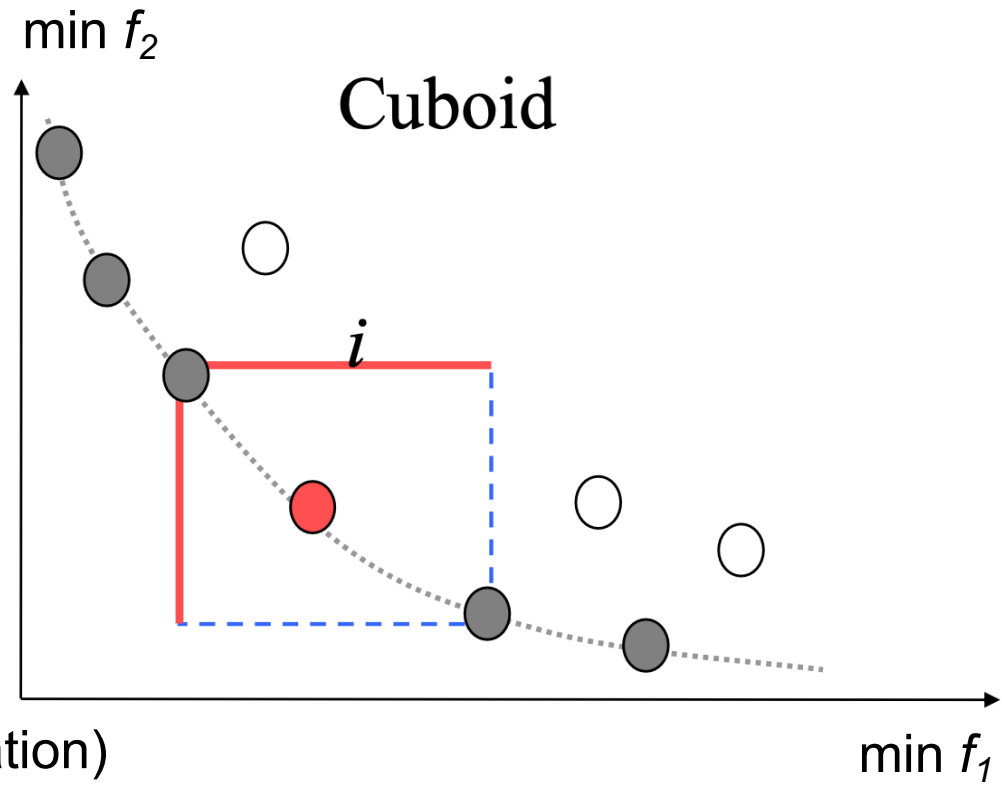
Computational complexity of this domination sorting is $O(MN^2)$



NSGA-II: Crowding distance index

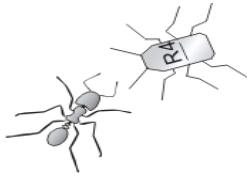
Average distance (side length) of cuboid in *M* dimensions (M is the number of objectives) from nearest neighbours: smaller distance value indicates higher crowding.

1. For each objective, order solutions from best to worst and normalize objective values
2. The normalized objective values of a solution define its position in the objective space
3. Assign infinite distance to first and last solutions
4. For each other solution, compute crowding distance
5. Normalize crowding distance values

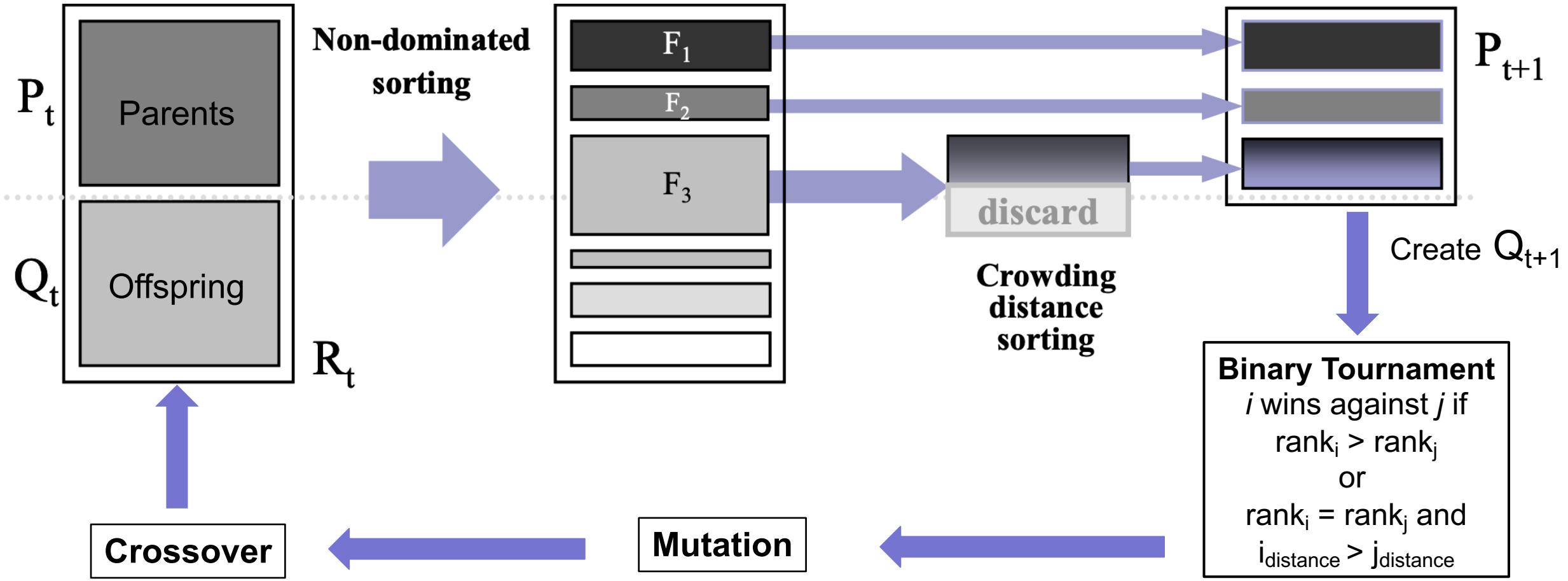


No need to set neighborhood parameter!

Each solution in the population is now characterized by:
- its nondominance rank (front number)
- its crowding distance (witht respect to the entire population)



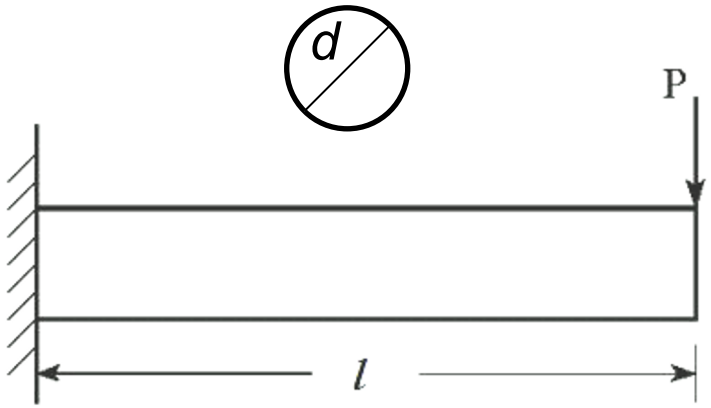
NSGA-II: Elitism by including offspring population in selection



PS: only dominance rank is used to create the first offspring population Q_1

NSGA-II: Constraint handling

Two decision variables:
1. Diameter *d*
2. Length *l*



minimize $f_1(d,l) = \rho \frac{\pi d^2}{4} l$ *weight*

minimize $f_2(d,l) = \delta = \frac{64Pl^3}{3E\pi d^4}$ *deflection*

constraints

$0.01\ \text{m} \le d \le 0.05\ \text{m}$ *diameter*

$0.2\ \text{m} \le l \le 1.0\ \text{m}$ *length*

$\sigma_{\max} = \frac{32Pl}{\pi d^3} \le S_y$ *max stress*

$\delta \le \delta_{\max}$ *max deflection*

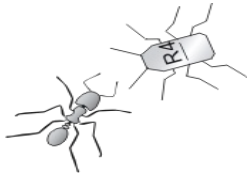
where

$\rho = 7800\ \text{kg/m}^3,\ P = 2\ \text{kN}$

$E = 207\ \text{GPa}$

$S_y = 300\ \text{MPa},\ \delta_{\max} = 0.005\ \text{m}$

If a solution is within the constraints it is said to be *feasible*, otherwise it is *unfeasible*



NSGA-II: Constraint handling

Redefine dominance relation by taking into account the constraints

A solution *i* is said to constrained-dominate a solution *j* if any of the following conditions is true:

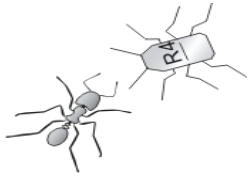
1) Solution *i* is feasible and solution *j* is not
2) Solutions *i* and *j* and are both infeasible, but solution *i* has a smaller overall constraint violation
3) Solutions *i* and *j* and are both feasible and solution *i* dominates solution *j*

Fast domination sorting becomes:

For each solution *p*, two entities are computed:

1. Scalar: Dominance count n_p = number of solutions that constrained-dominate solution *p*
2. Matrix: Set S_p = set of solutions *q* that *p* constrained-dominates

Try to change the constraint values in the exercises!



Revisiting selection

I should premise that I use the term Struggle for Existence in a large and metaphorical sense . . . . Two canine animals in a time of dearth, may be truly said to struggle with each other... But a plant on the edge of a desert is said to struggle for life against the drought, though more properly it should be said to depend on the moisture **(Darwin, On the Origin of Species, 1860)**

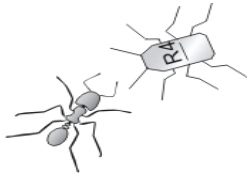
Viability

To stay alive, an organism must keep physiological parameters within upper and lower viability boundaries

Cannon, 1939
The wisdom of the body

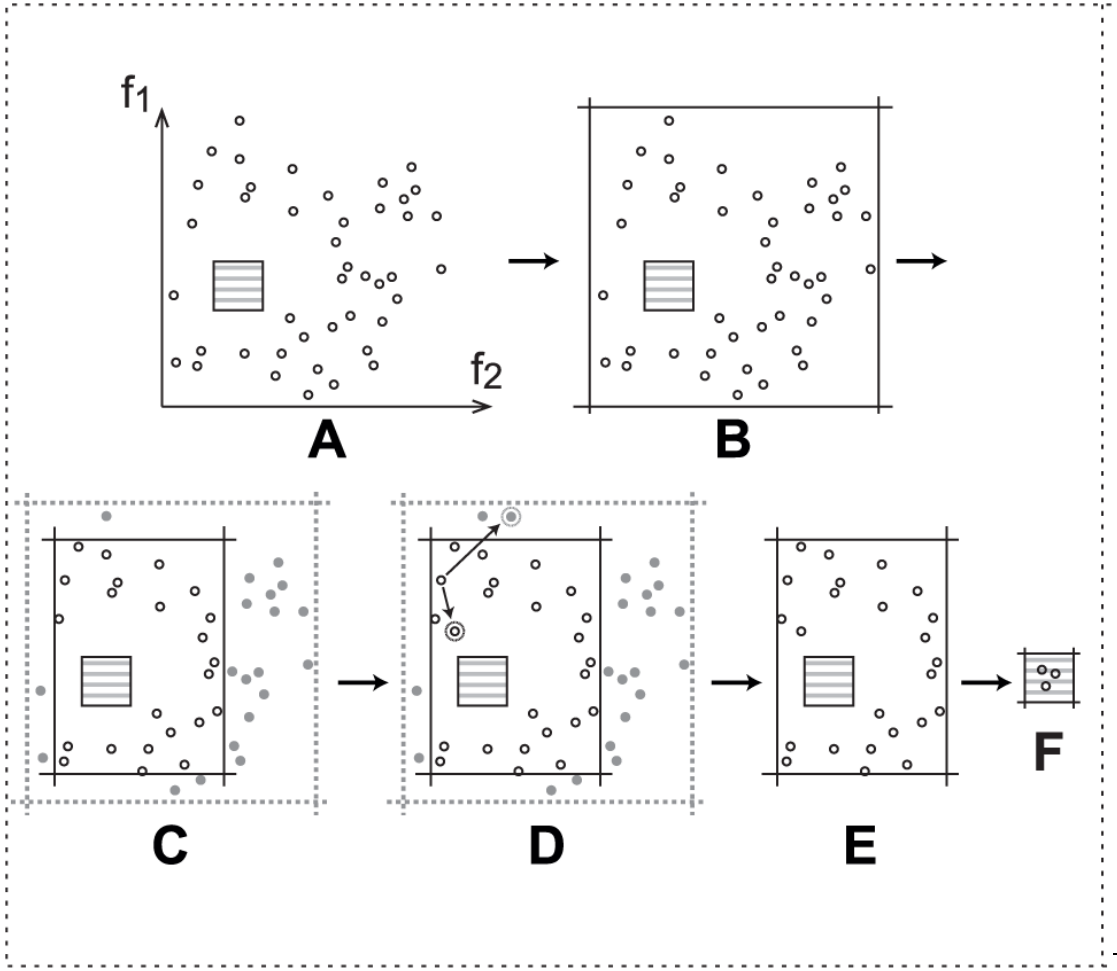
Ashby, 1960
Design for a Brain

Aubin, 1991
Viability Theory



Viability Evolution (ViE) *Maesani, Mattiussi, Floreano 2014*

Reproduce all viable solutions, where viability is defined by minimum fitness and constraint feasibility. Population size can vary, and there can be population extinctions



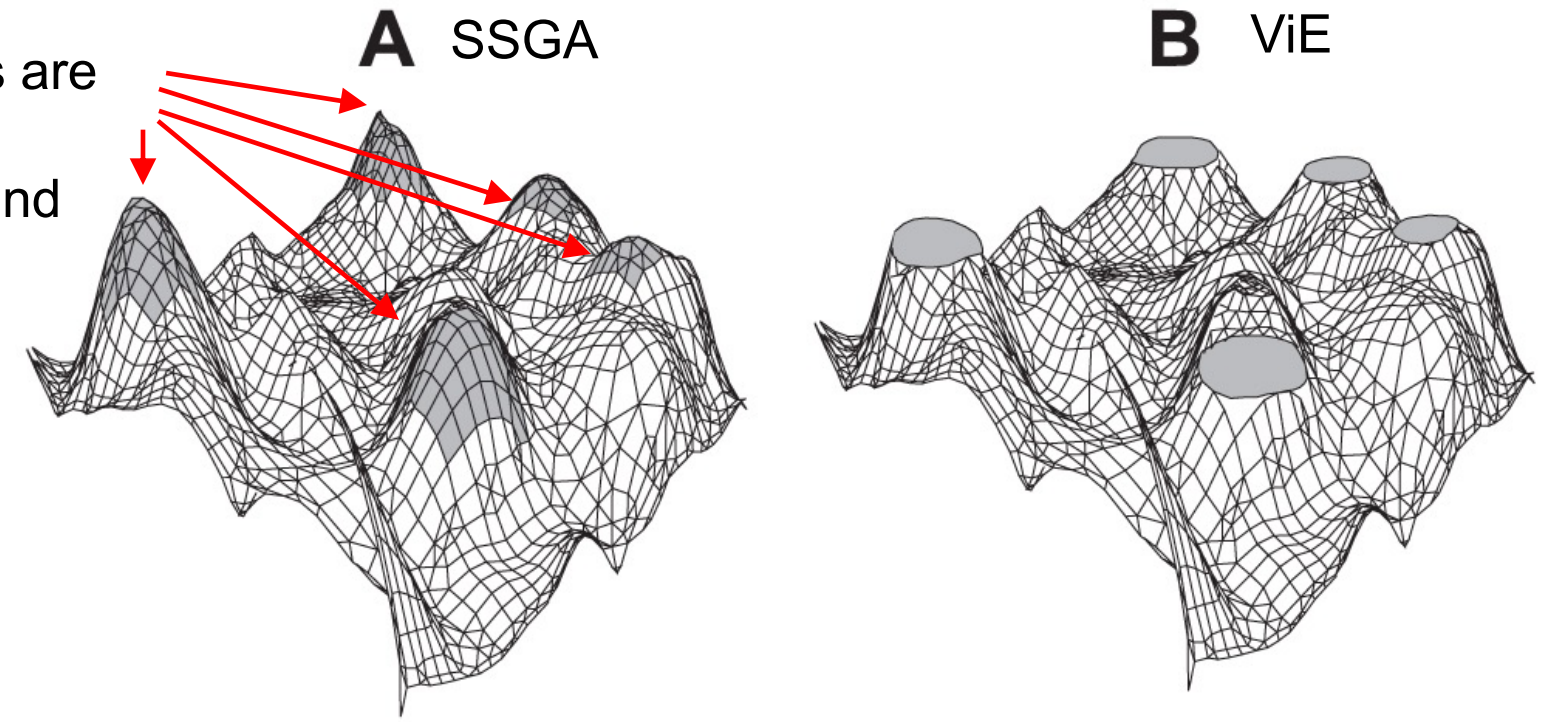
Max_size= maximum population size
t=0
(**A**) Set *Target_boundaries* for each objective and create initial population (*Max_size)*
(**B**) Enlarge *boundaries(t)* to encompass all solutions

while t≤*T* or *boundaries(t)=Target_boundaries*
(**C**) Shrink *boundaries(t)* towards *Target_boundaries*
(**D**) Reproduce viable individuals with mutations (up to *Max_size)*
(**D**) Eliminate non-viable individuals

Standard Simple GA and ViE on Multimodal functions

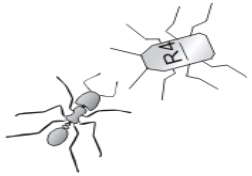
To make the two algorithms comparable, we must put an arbitrarily chosen limit to the objective function in SSGA all individuals and add fitness sharing to SSGA to promote diversity

Final solutions are expected to distribute around peak regions

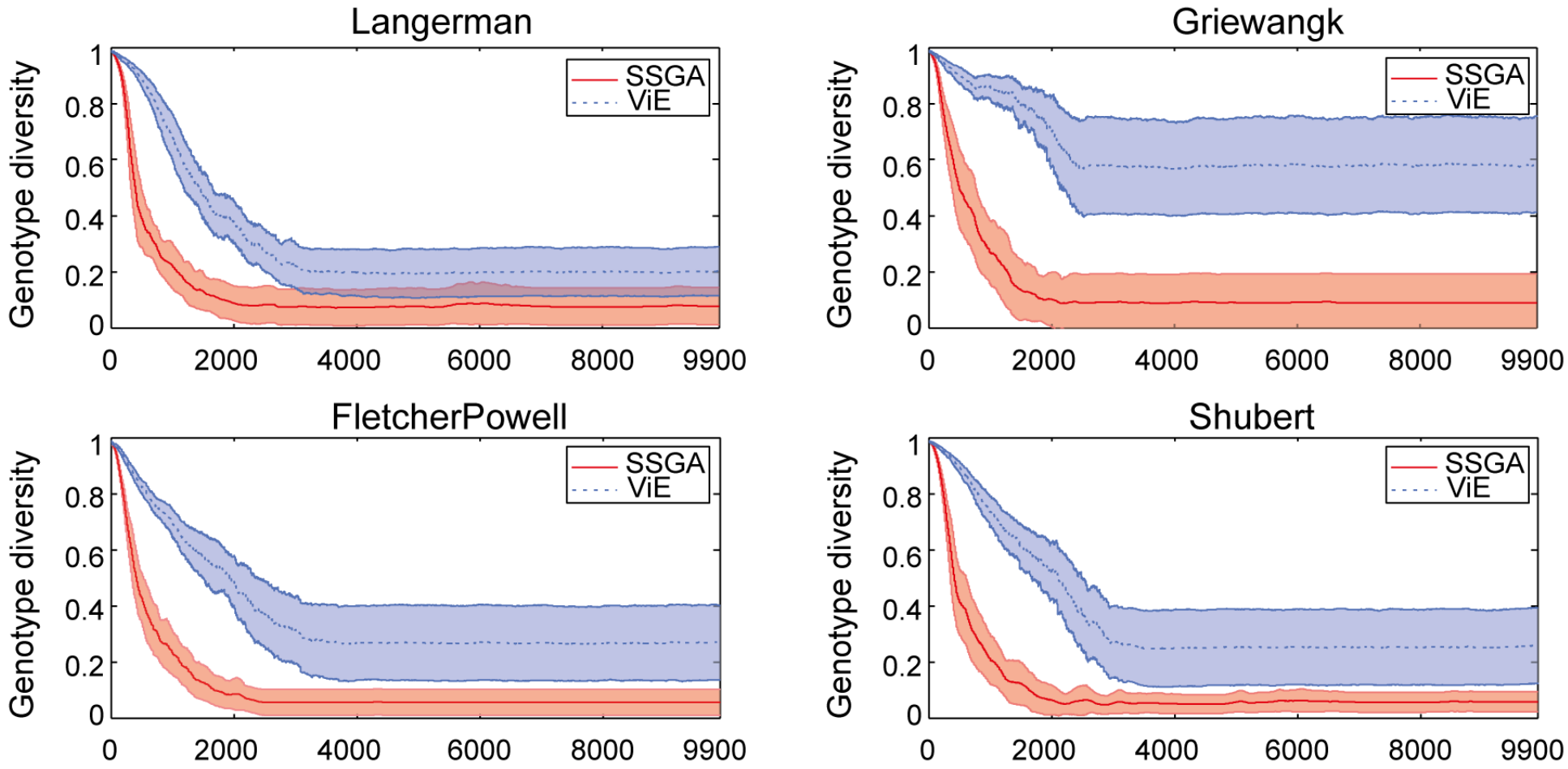


Target regions match peak regions of single-objective optimization

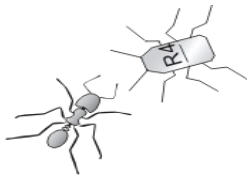
ViE population can grow up to size of SSGA population



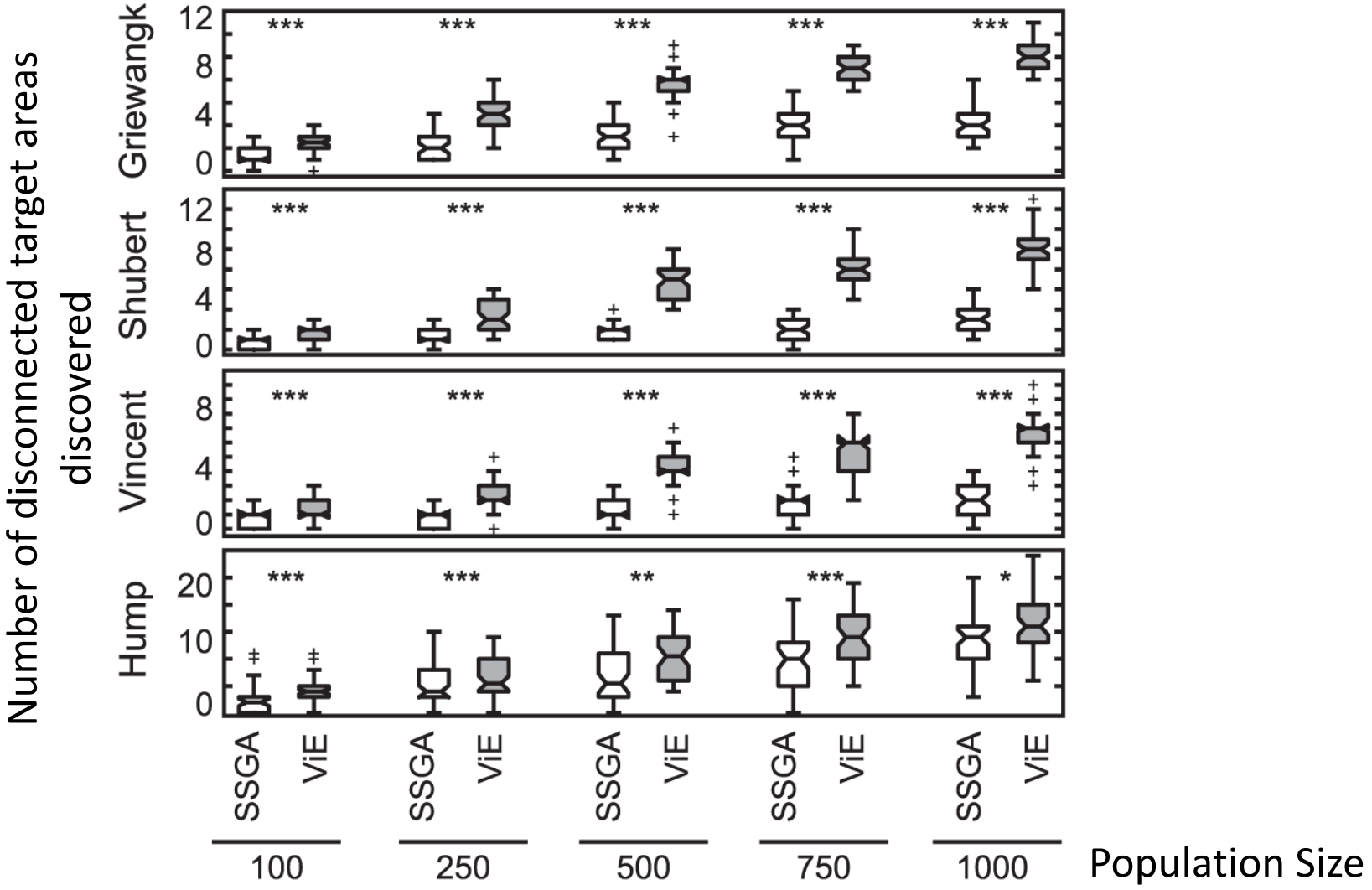
Viability can maintain higher genetic diversity



Selected plots from 10 benchmark multimodal functions

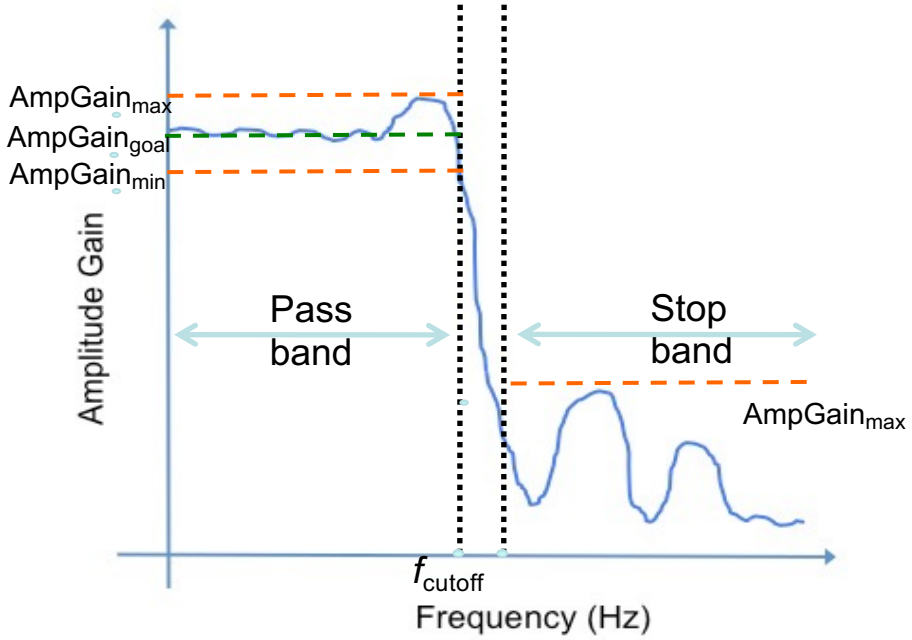


Vie can find solutions in more disconnected target areas



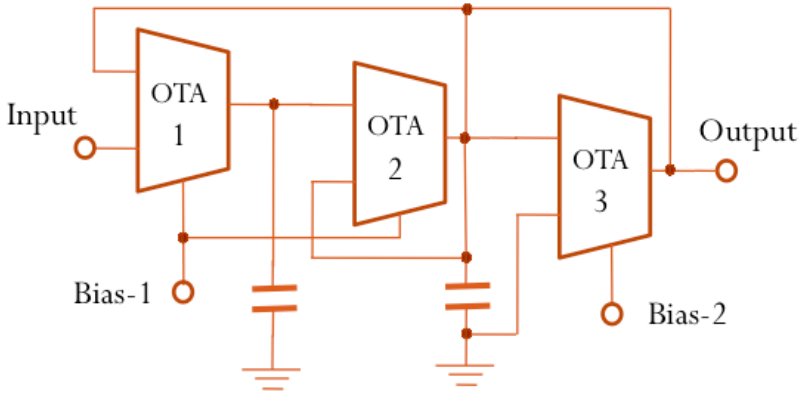
Multi-objective Optimization of Electronic Low Pass Filter

Low pass filters are analog circuits used in all amplifiers and are critical for amplification quality



1. Gain-Bandwidth (GBW) product
2. Pass Band Flatness
3. Stop Band Attenuation

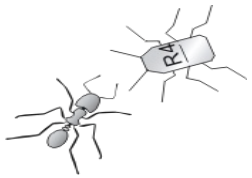
OTA – Operational Transconductance Amplifier



Tunable 2nd Order Low Pass Filter

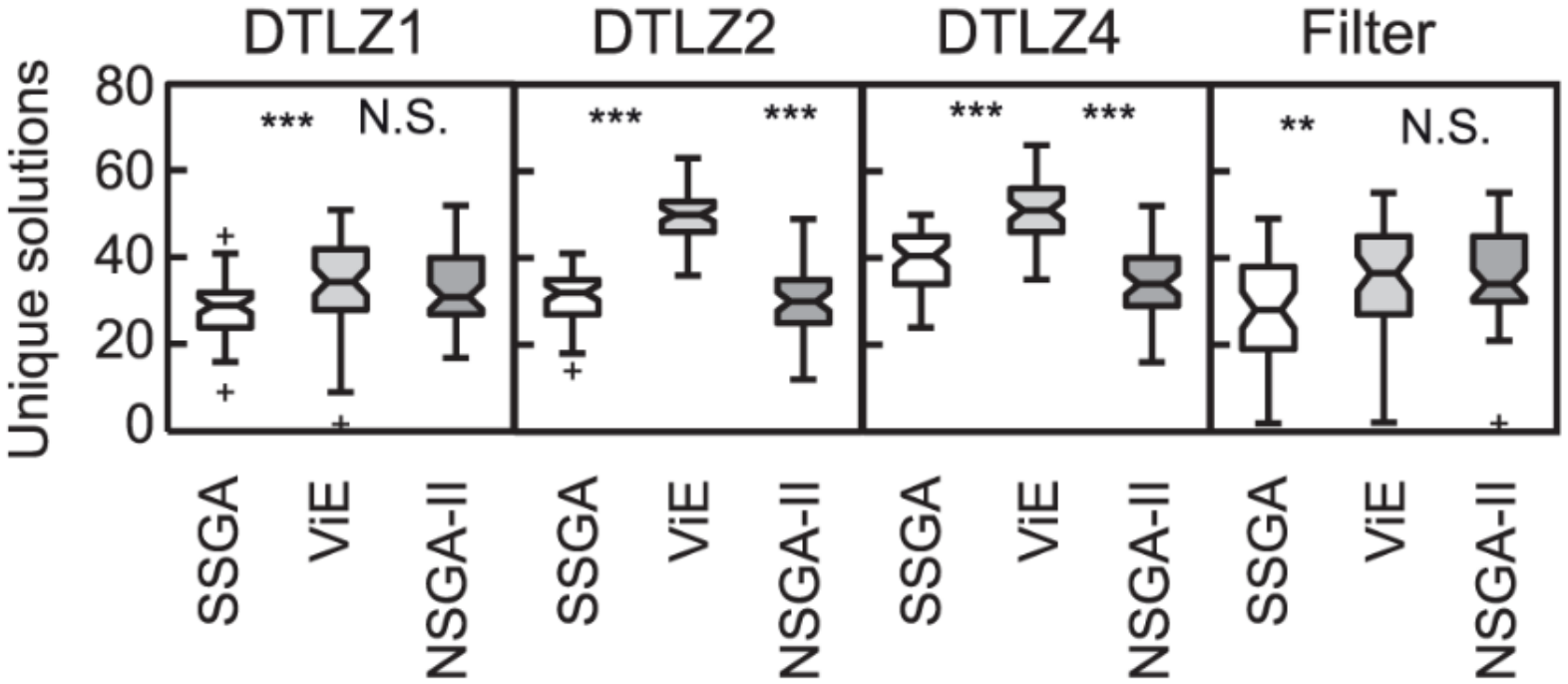
(Redrawn from "Active Filter Design using Operational Transconductance Amplifiers: A Tutorial", R.L. Geiger and E.Sanchez-Sinencio, IEEE Circuits & Devices Magazine, Vol. 1, pp.20-32, 1985)

The values of the filter's characteristics can be modified using the voltages at the two bias points

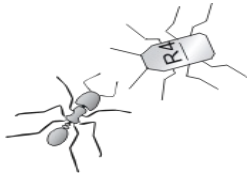


Multi-objective Optimization: SSGA, NSGA-II and ViE

ViE discovers more unique solutions than SSGA with fitness sharing and objective weighting, and more or equal number of unique solutions than NSGA-II

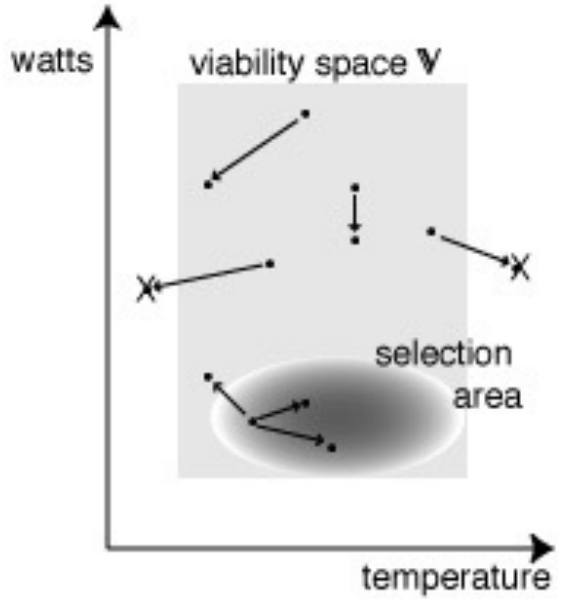


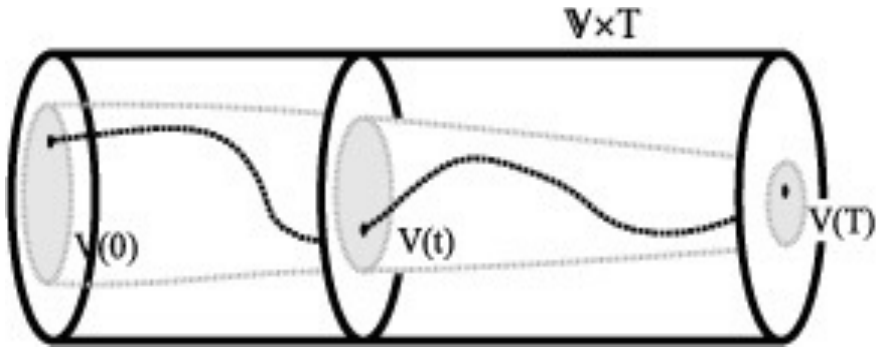
DTLZ is a standard set of multiobjective problems for benchmarking algorithms



Memetic Evolution: Viability with Fitness-based selection

A combination of evolutionary algorithms is called **Memetic Evolution.**
Viability evolution allows for regions of competitive selection if desired: individuals in those areas can use proportionate (or tournament selection) to reproduce.





Evolution over time

Memetic Viability Evolution has beaten all best single-objective and multi-objective optimization algorithms in a class of standard benchmark problems in terms of performance and computation time (Maesani, Iacca, Floreano, 2016)