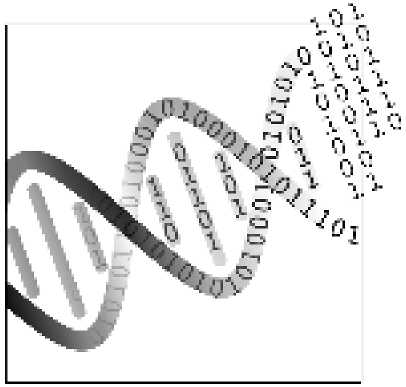
Multi-objective Evolutionary Optimization



What you will learn in this lecture

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

Single-objective optimization

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

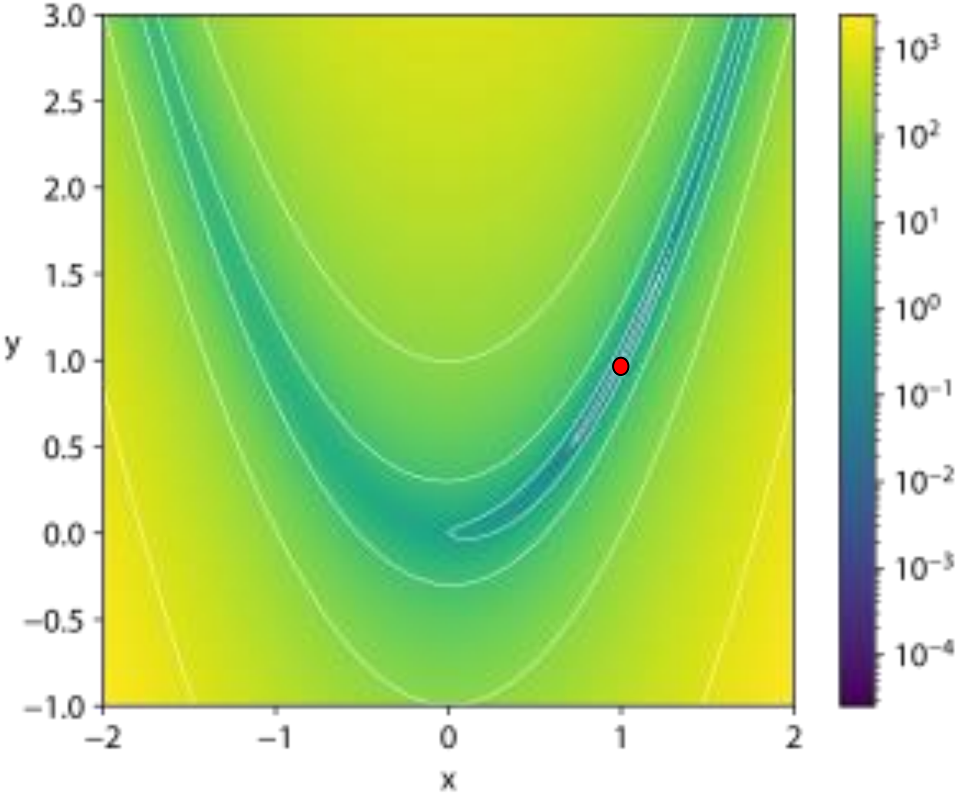
Rosenbrock function of 2 variables

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

Our goal is to find the optimum (minimum)

For a=1, b=100

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



Single-objective, multimodal function

- If there are several equal optima, all evolutionary algorithms seen so far will find only one optimum
- We may want to find all optimal (maximum) solutions to choose from based on additional criteria

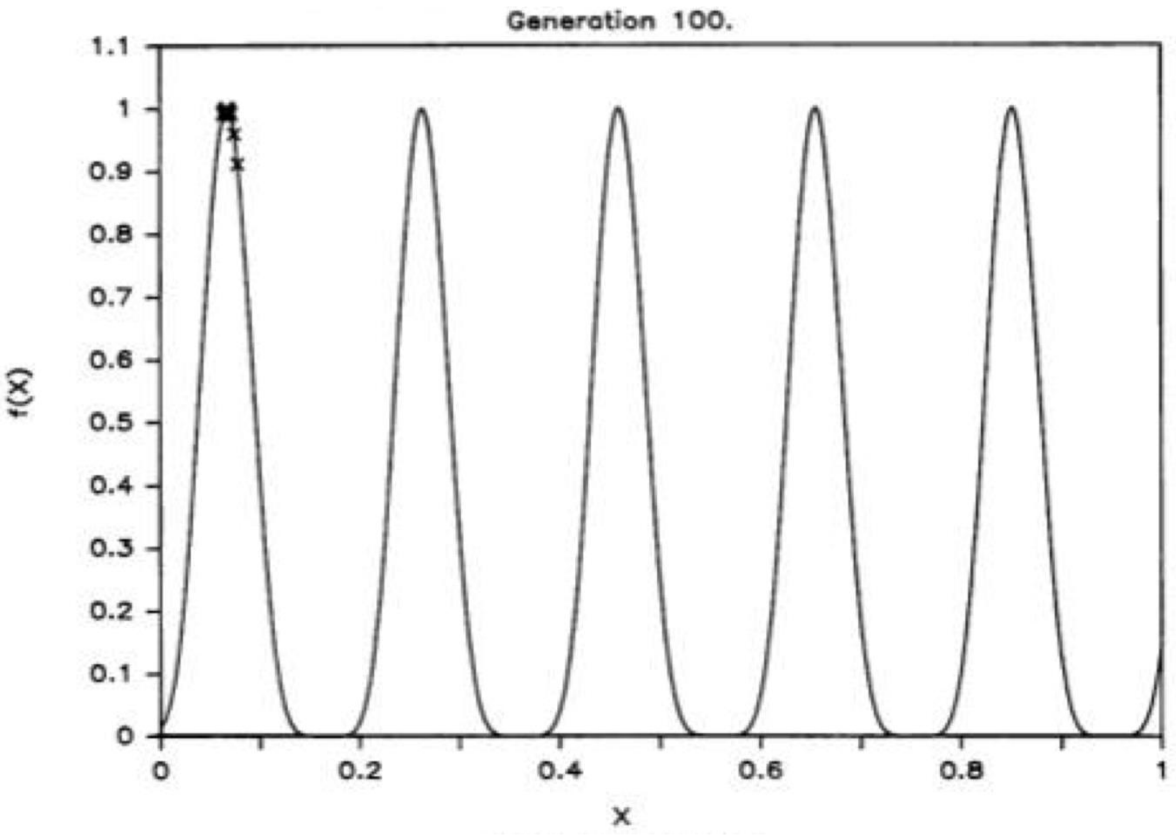


Image from Goldberg, D.E. (1989) Genetic Algorithms, Addison-Wesley

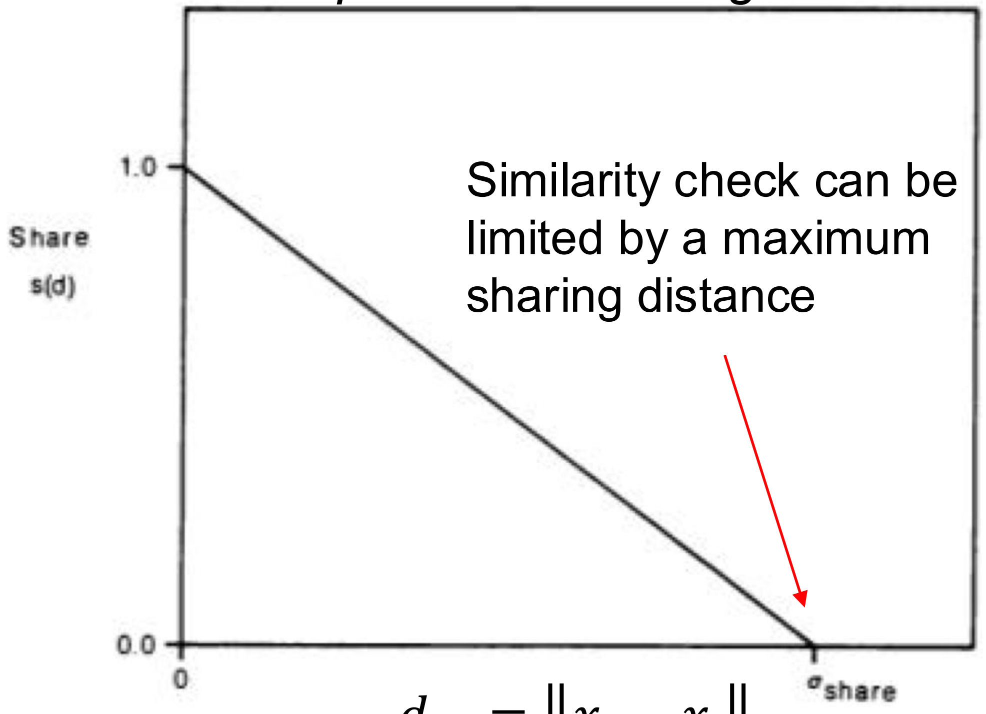
Fitness sharing *(Goldberg and Richardson, 1987)*

- 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

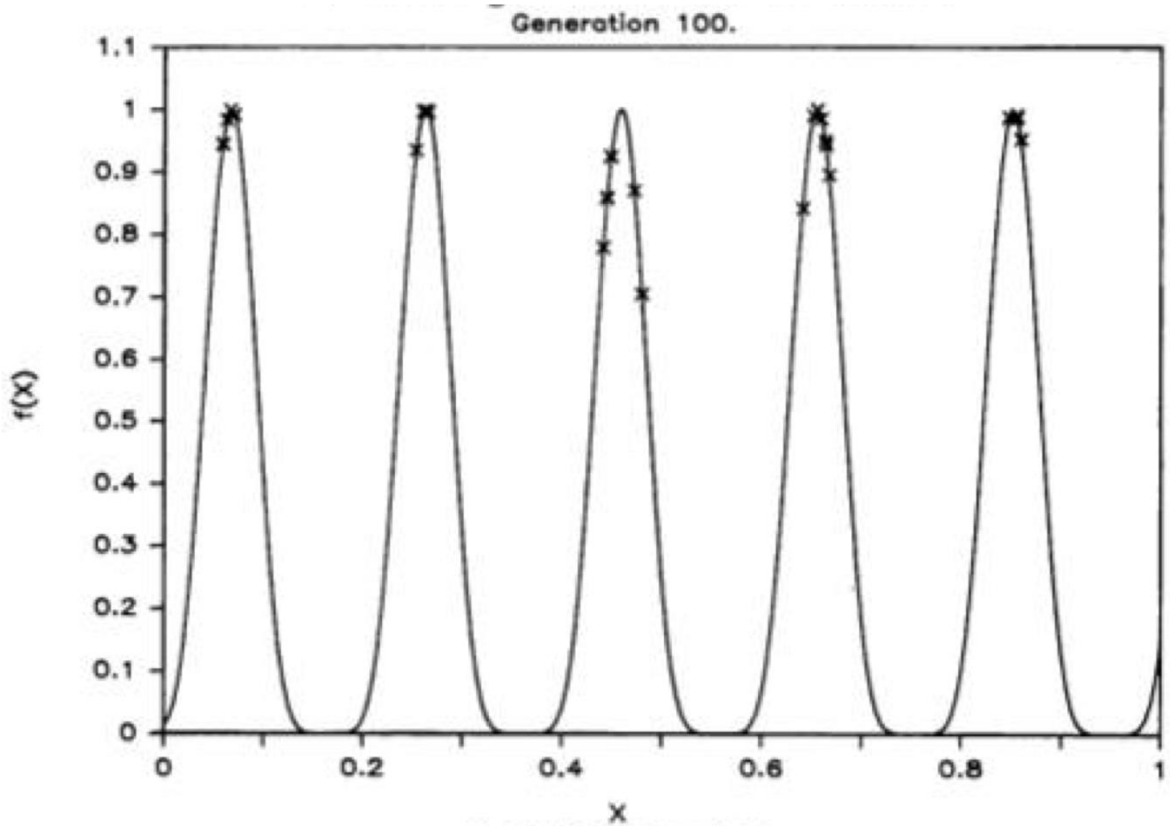


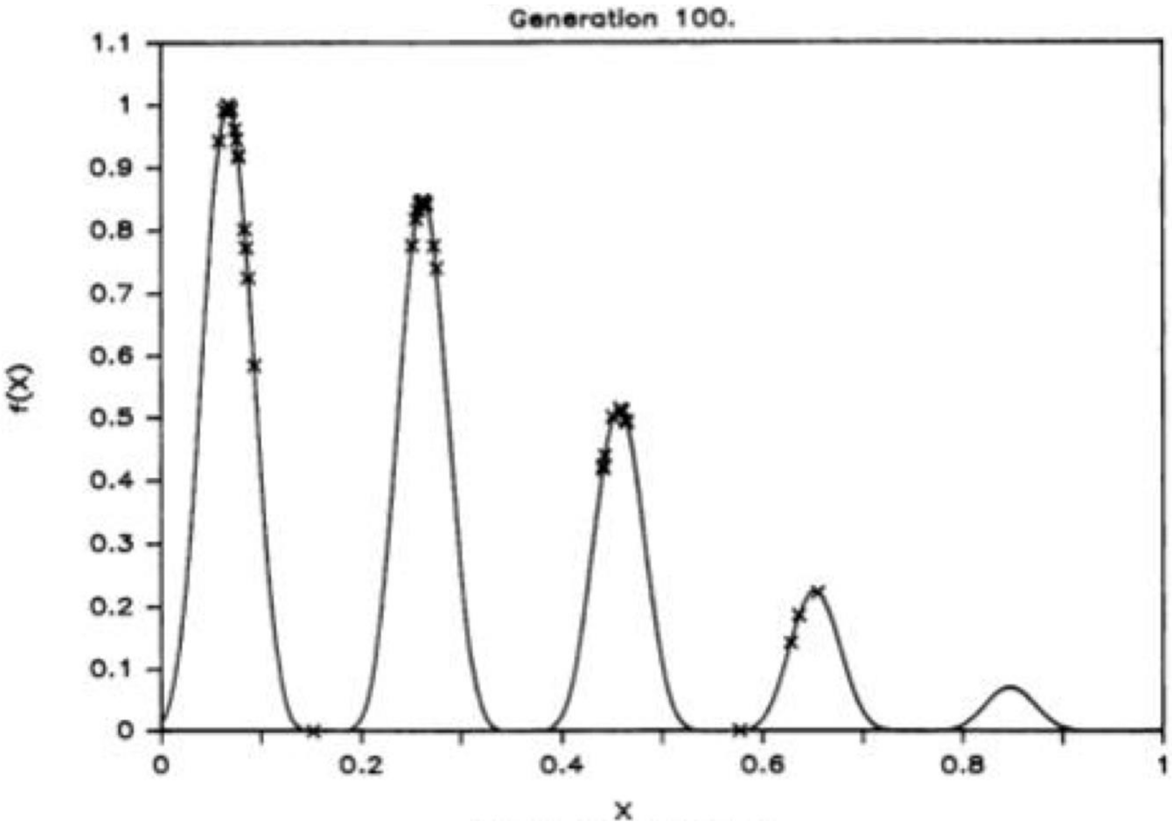
$$d_{i,j} = \left\|x_i - x_j\right\|$$

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





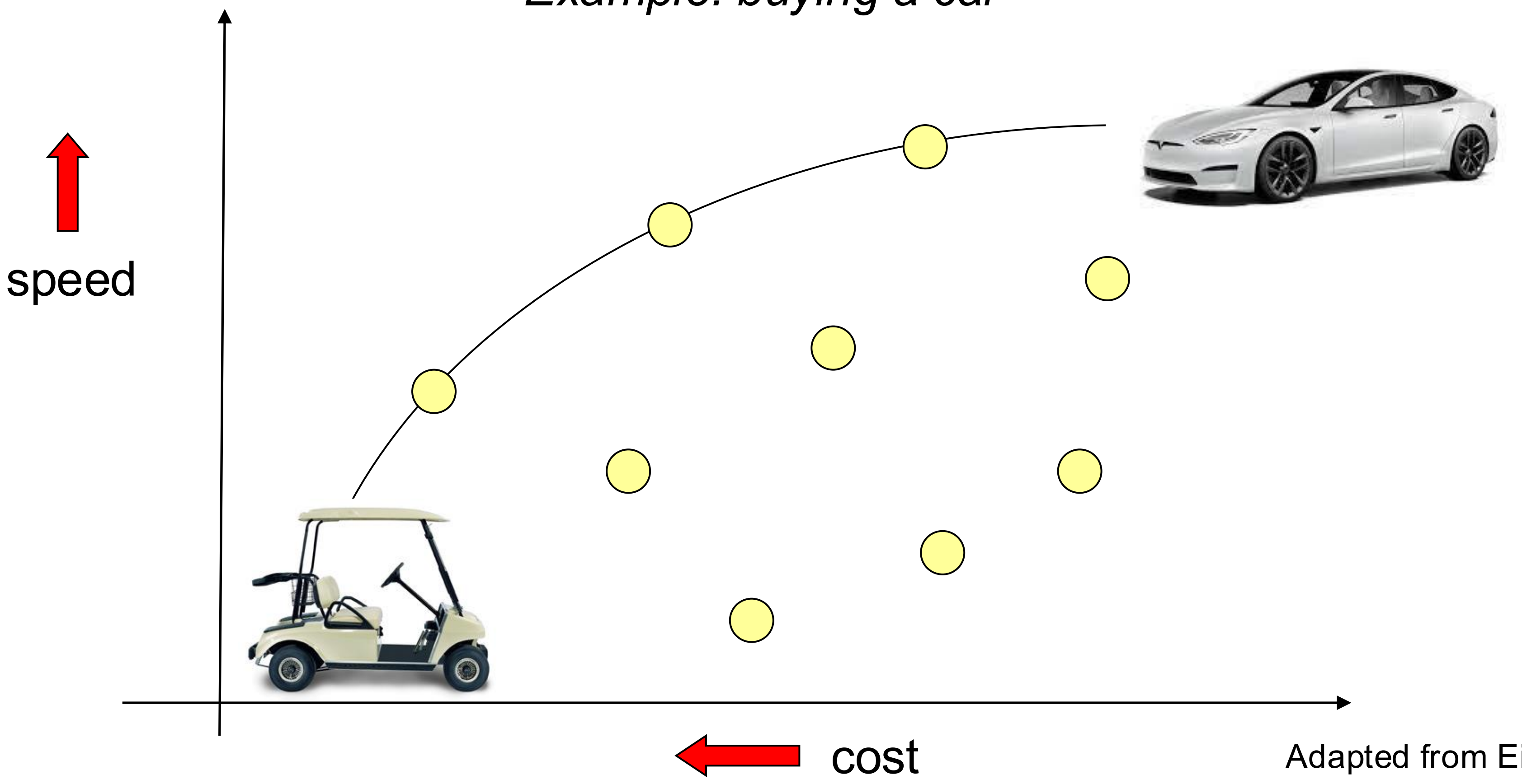
Number of solutions is proportional to the relative value of the local optima: more solutions are allocated to better optima

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

Multi-objective optimization

- Many real-world problems present <u>multiple objectives that are not commensurable or compatible</u>
- There is not a solution that is superior to others with respect to all objectives
- In this case, we want to identify the solutions that are the "best compromise"

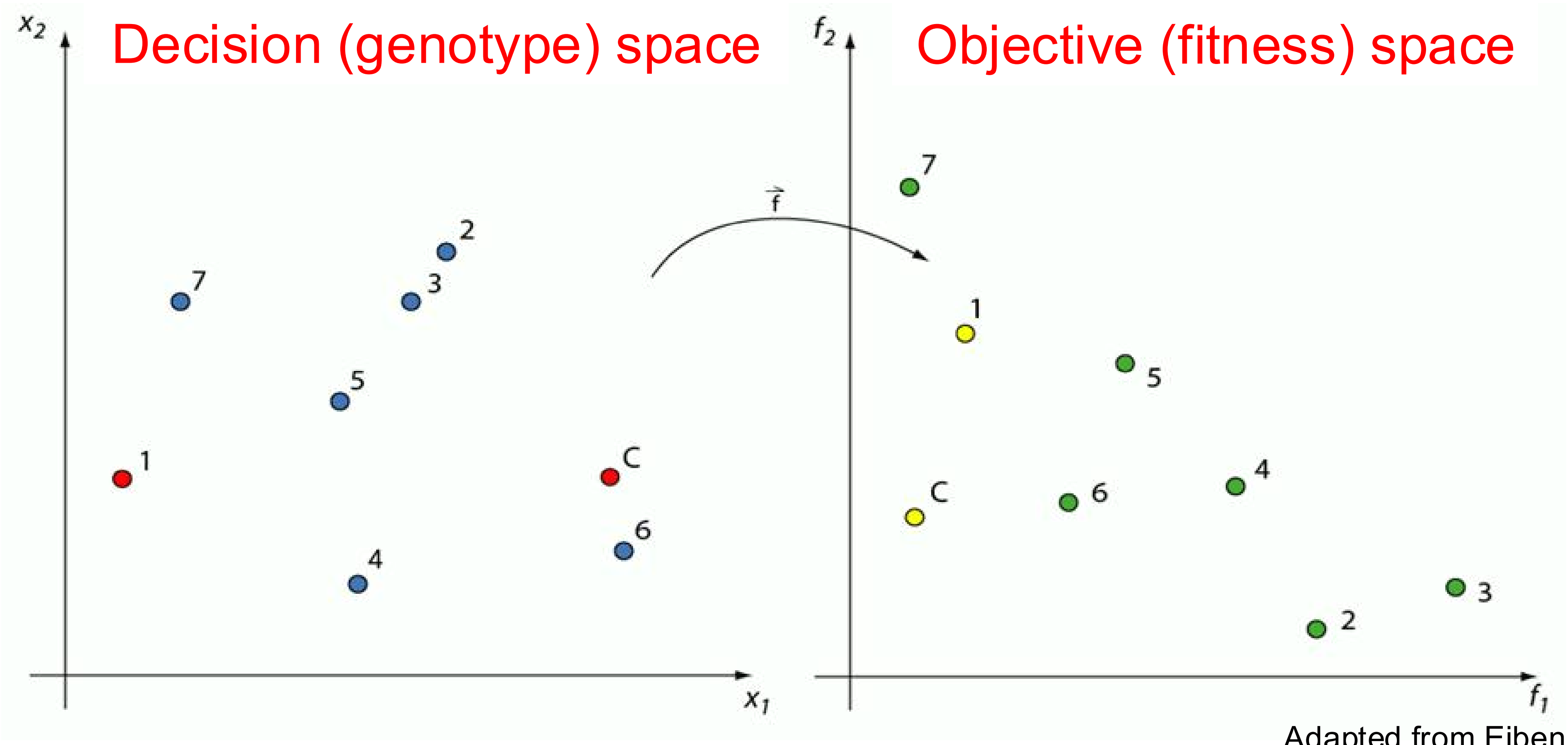
Example: buying a car



Adapted from Eiben & Smith, 2015

Decision and objective spaces

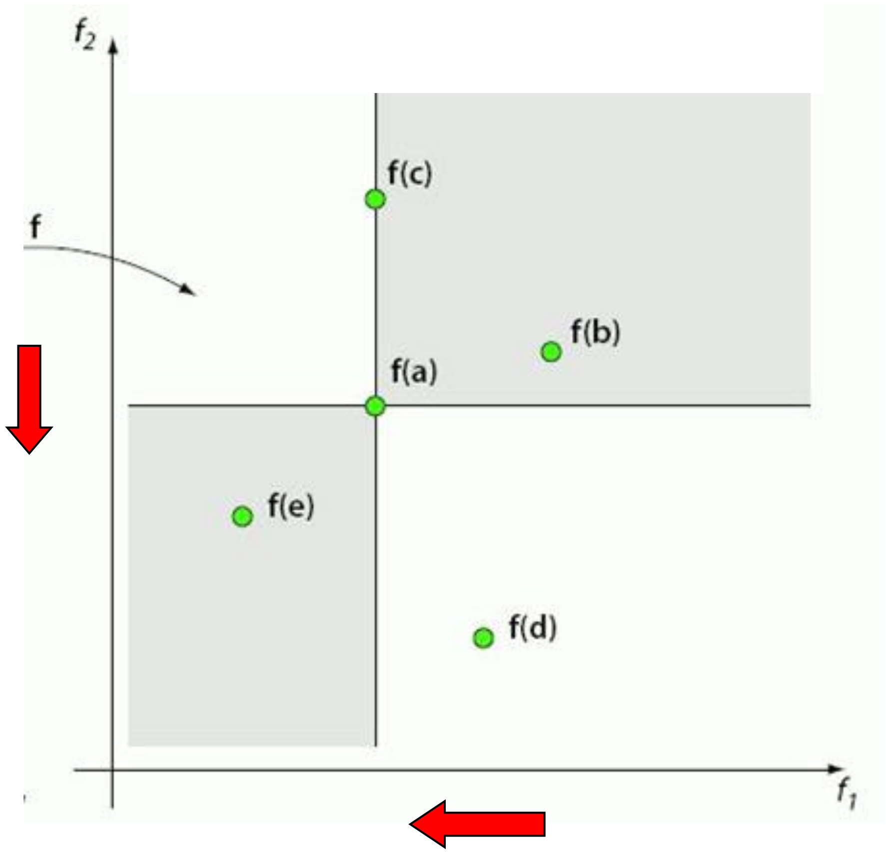
- It is convenient to distinguish between decision space (genotype) and objective space (fitness)
- We assume a one-to-one mapping between decision space and objective space



Adapted from Eiben & Smith, 2015

Comparing solutions in the objective space

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* between solutions in the objective space.

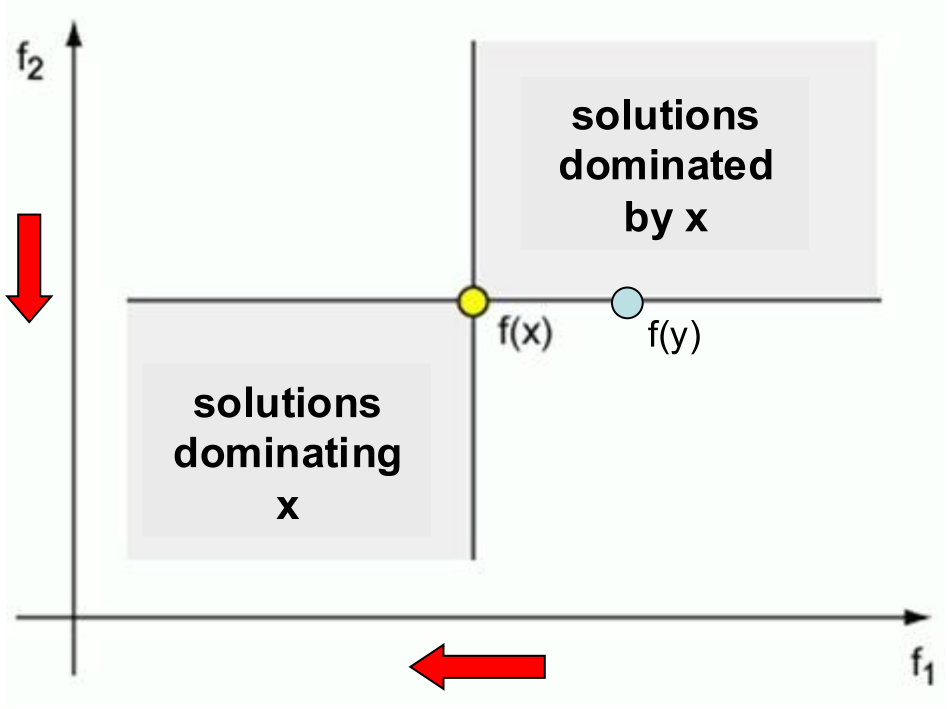
For a minimization problem with *k* objectives, solution *x* dominates solution *y* if:

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 on some objectives

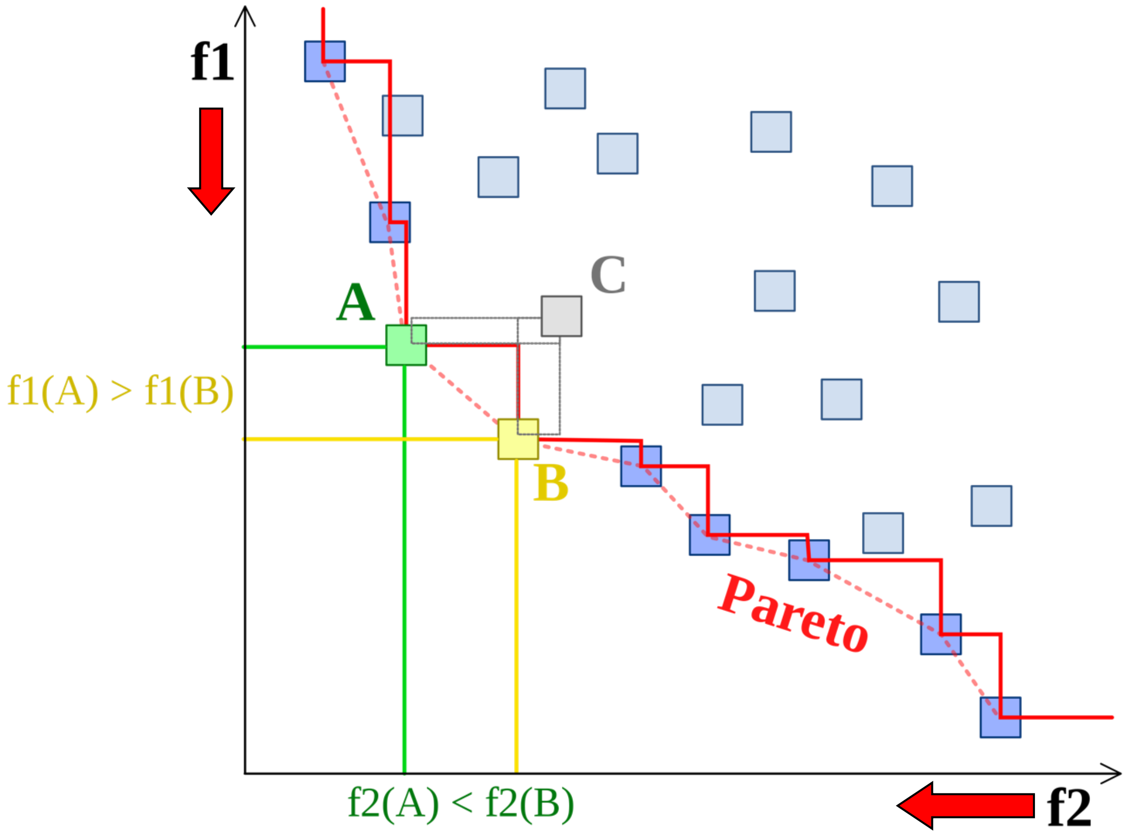


Image: Johann Dréo

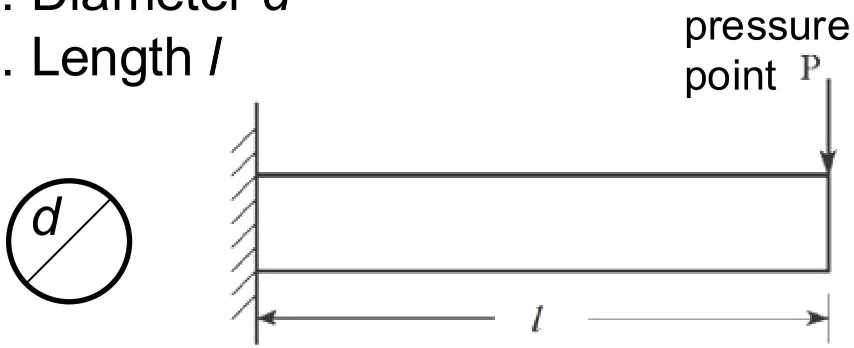
Minimize steel beam's weight and deflection (Deb, 2001)

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*

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

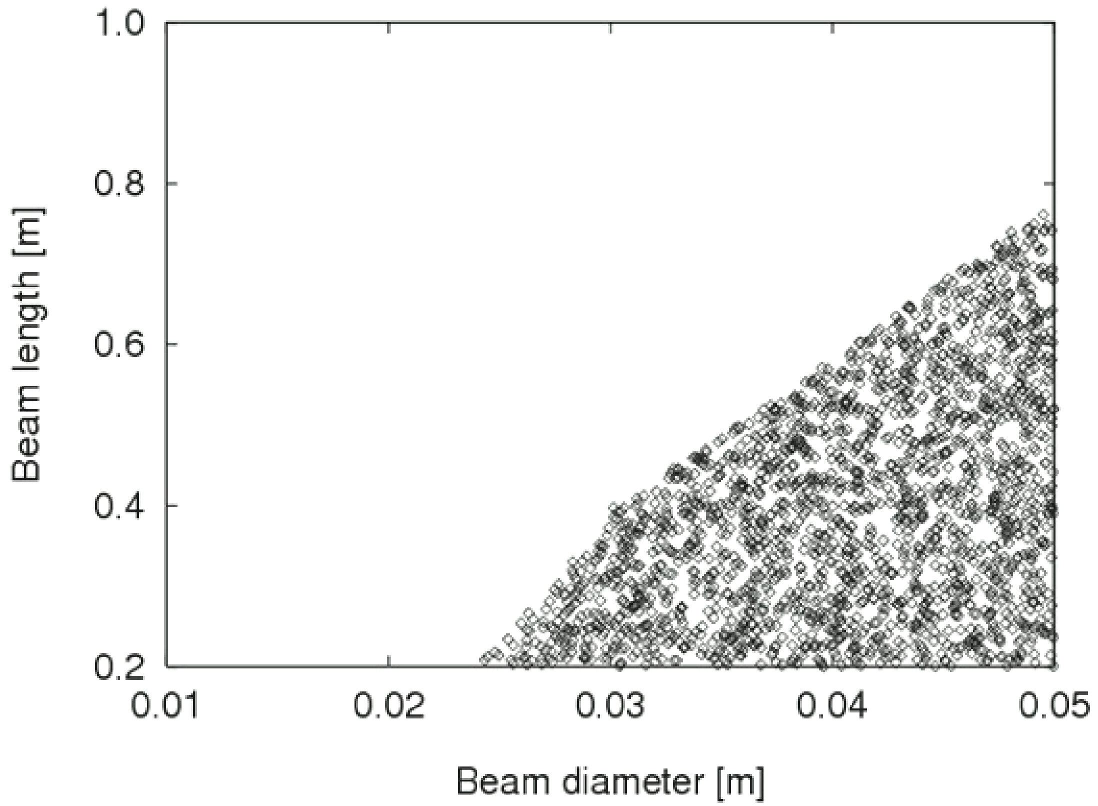


where
- $\rho = 7800 \text{ kg/m}^3, \; P = 2 \text{ kN}$
- $E = 207 \text{ GPa}$ *modulus of elasticity*
- $S_y = 300 \text{ MPa}, \; \delta_{\max} = 0.005 \text{ m}$

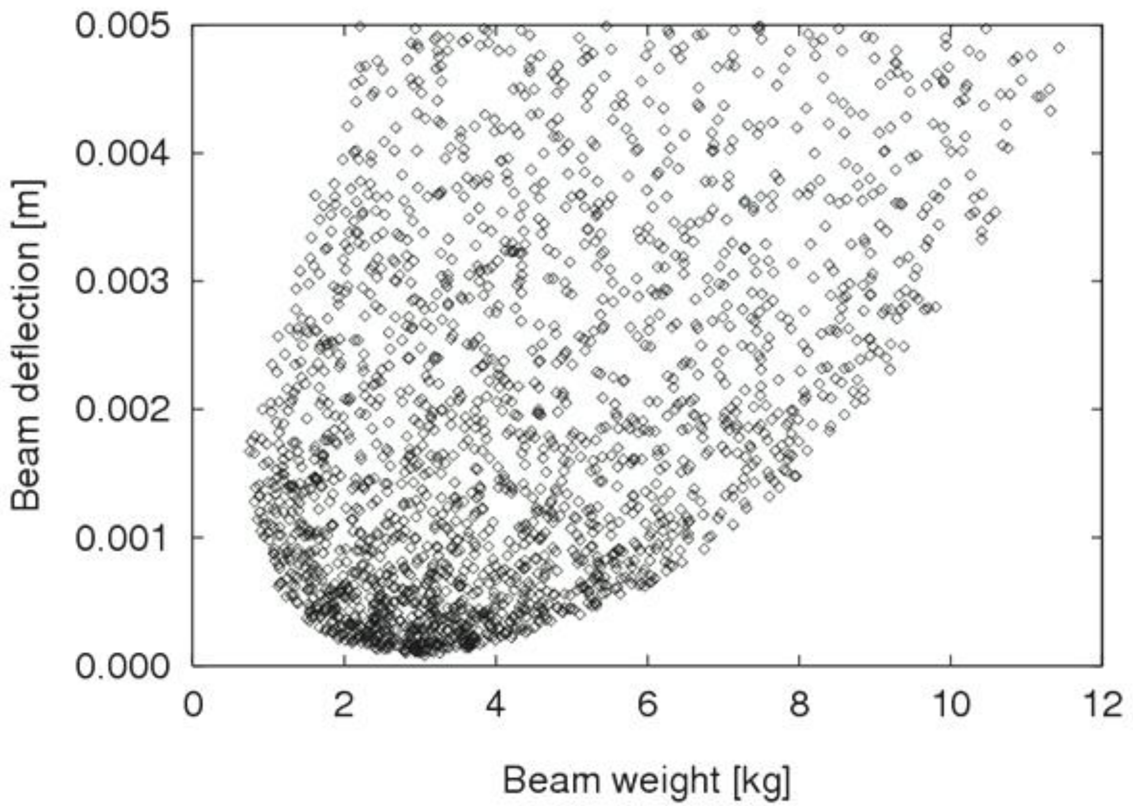
Feasible solutions

Feasible solutions are solutions that fulfil the constraints

Decision space



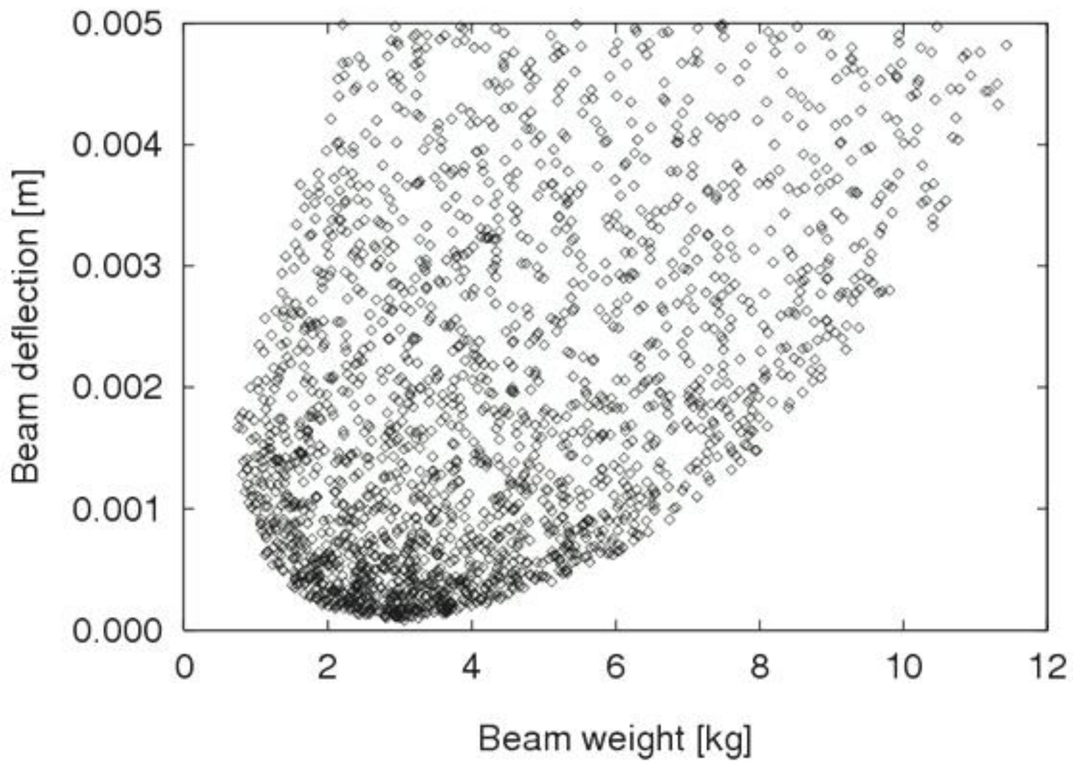
Objective space

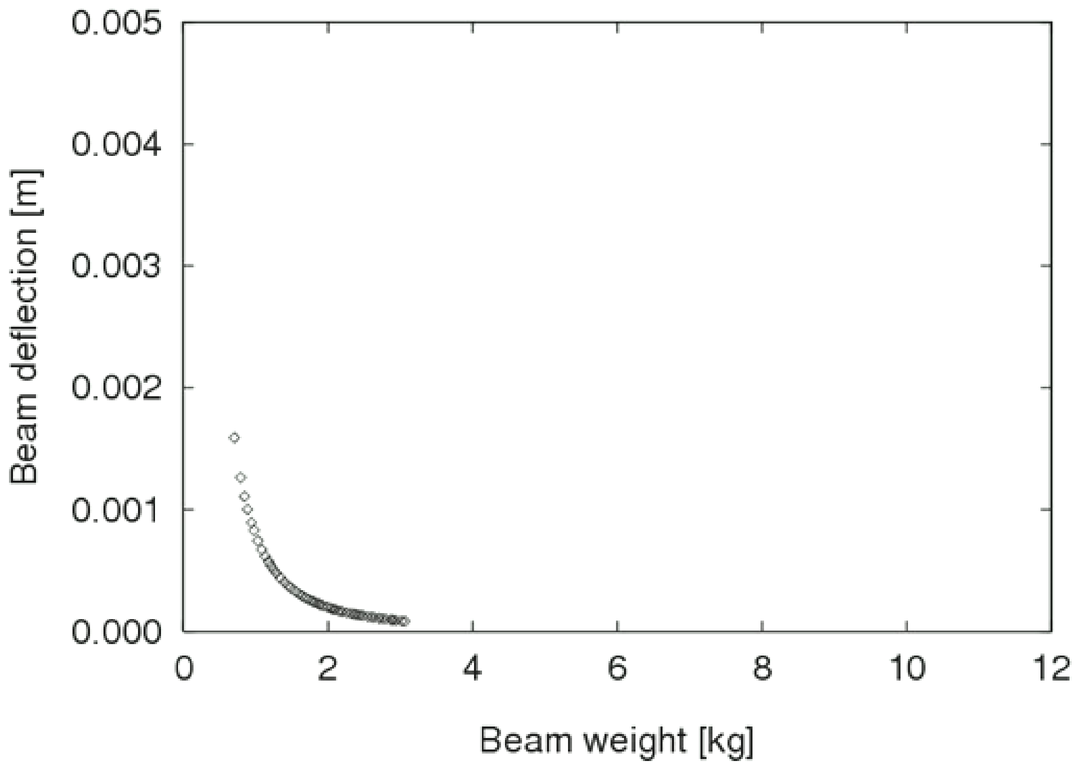


Goal: Finding non-dominated solutions

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

Characteristics	**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 **f**(**x**) into a single objective Z(**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 optimal solution, compute its value on each and all other 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 one Pareto optimal solution

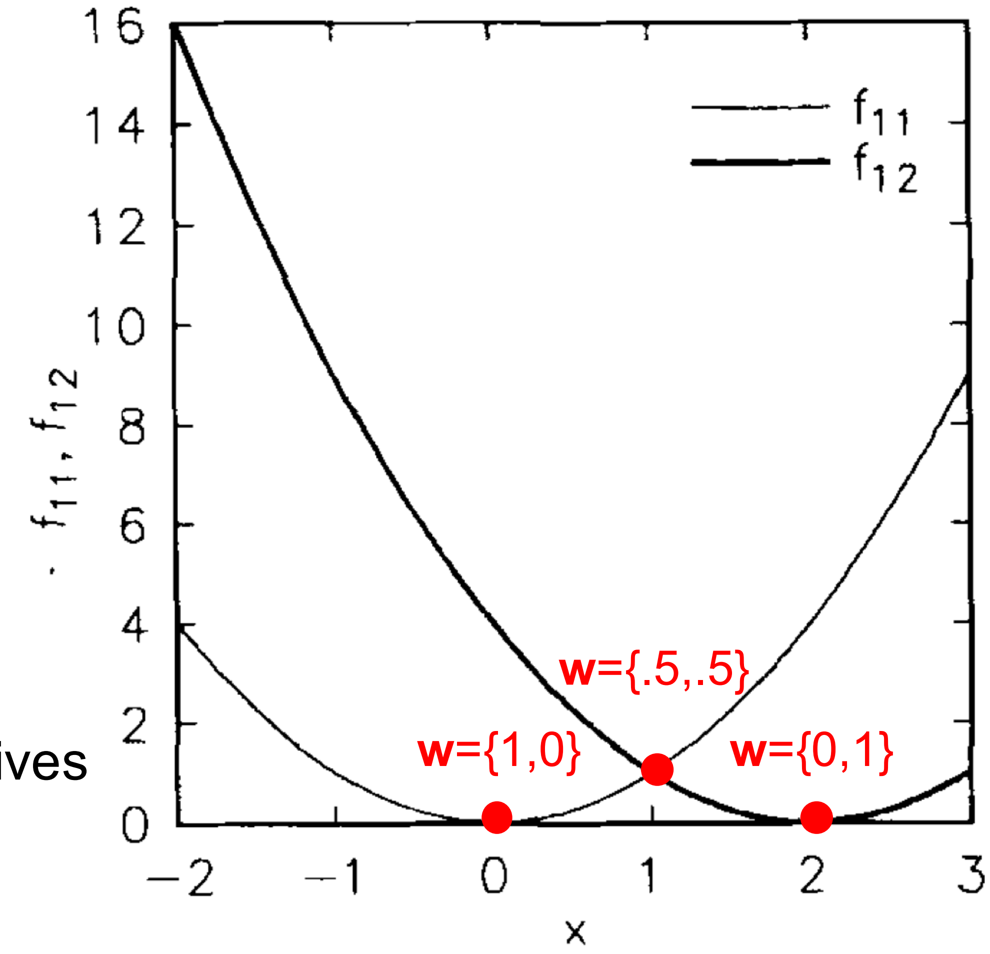
Example of Objective Weighting

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

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

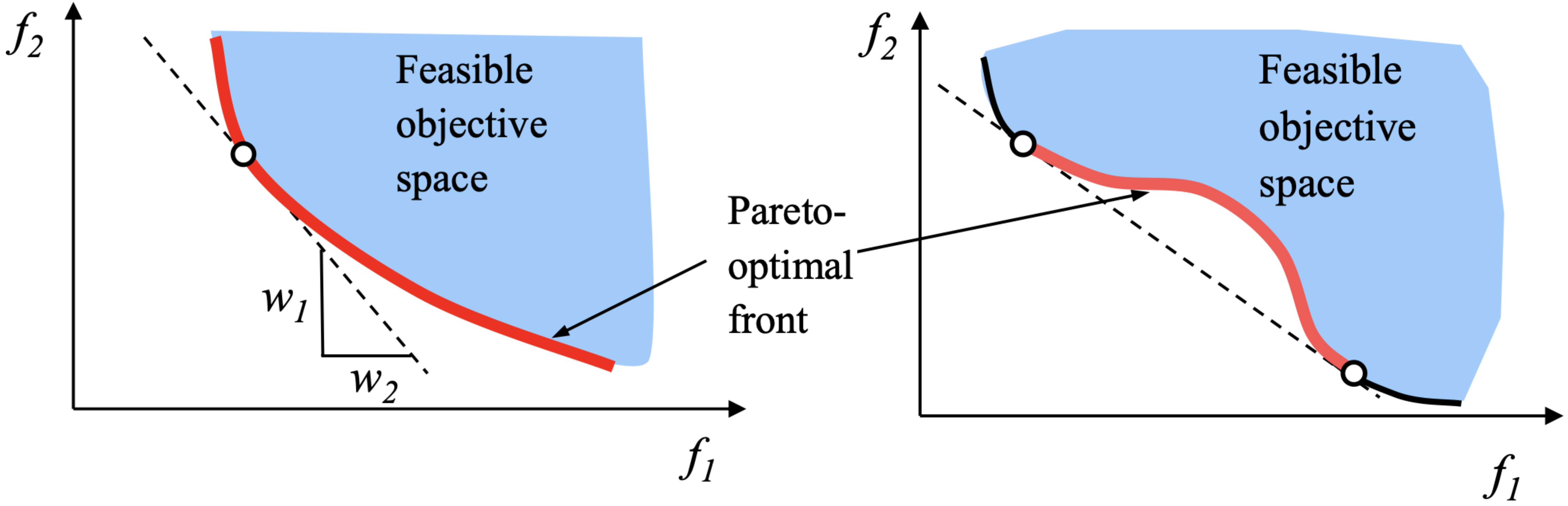
Pros:
- Each found solution is a Pareto optimum
- Weight vector allows control of objective priorities

Cons:
- For a given weight set, only one solution is found
- Not obvious how to choose **w** when there are many objectives



Objective weighting and Pareto front convexity

The choice of objective weights defines an optimal point on the tangent to the Pareto Front
Therefore, objective weighting can be used only when the Pareto Front is convex



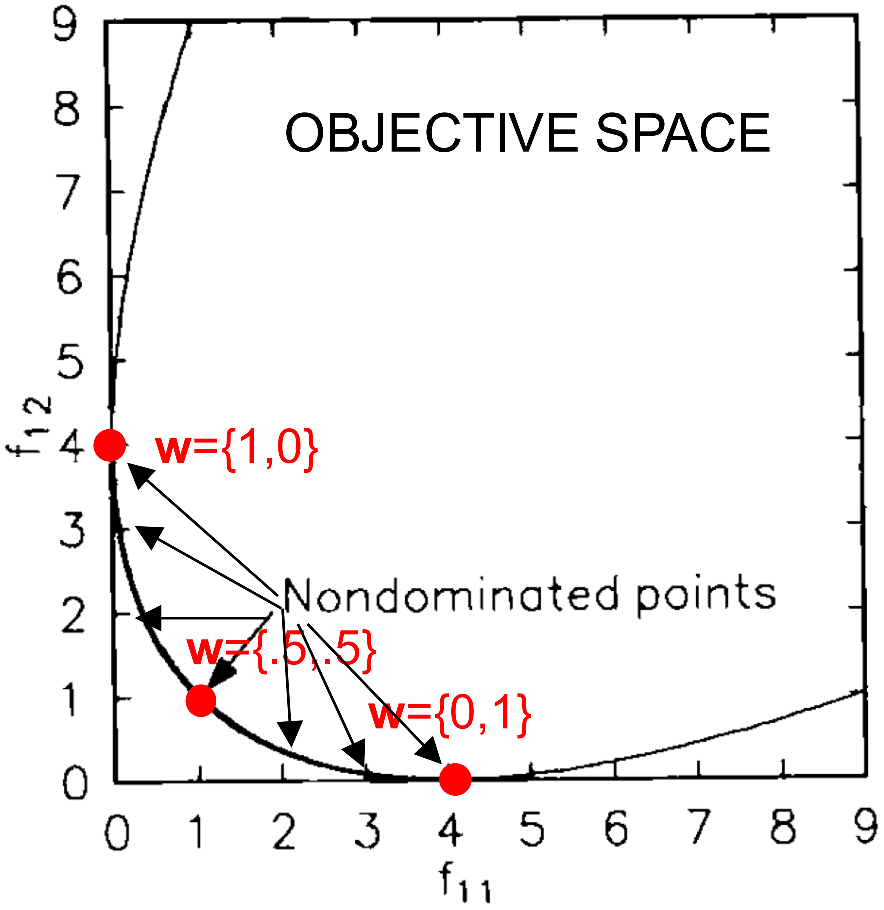
Source: K. Deb

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 can search for, and 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

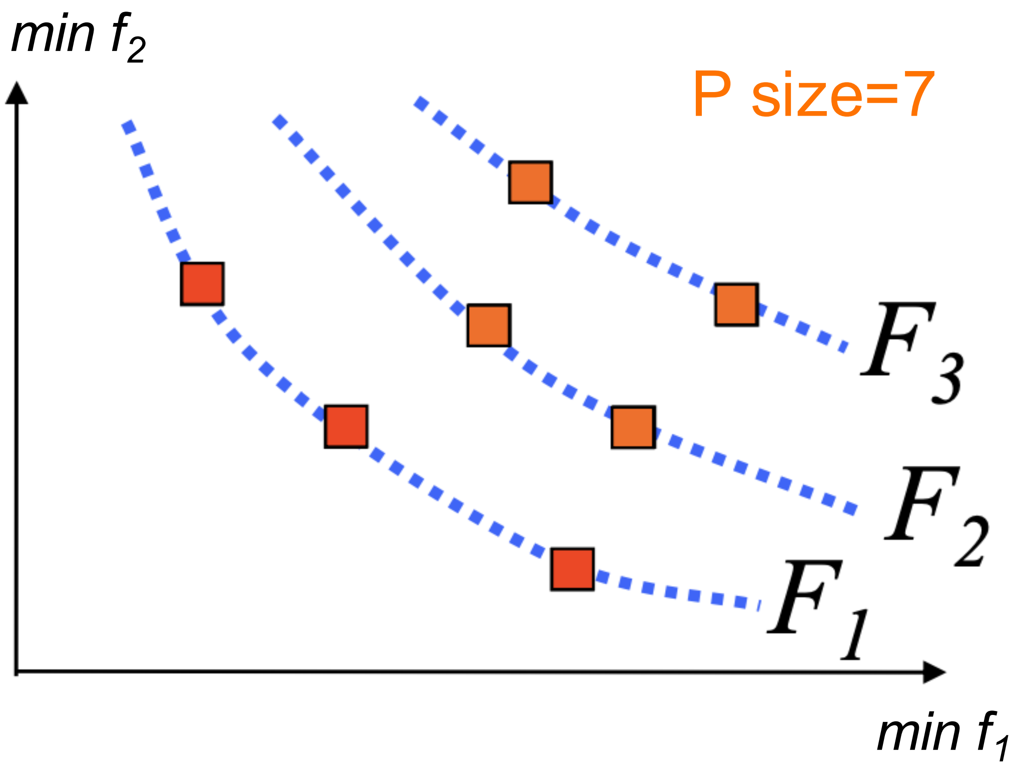
NSGA: Identification of F Pareto fronts

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

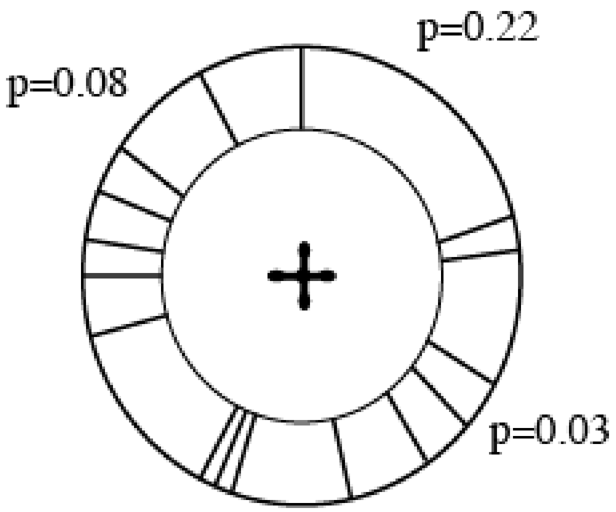
1. Assign identical high dummy fitness r to all solutions in F_i
2. Fitness sharing: compute shared fitness s of each solution (divide fitness of individual solutions in F_i by quantity proportional to number of other solutions in their neighborhood)
3. Set r to be smaller than smallest shared fitness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

NSGA: Create new population

1. Proportionate selection

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

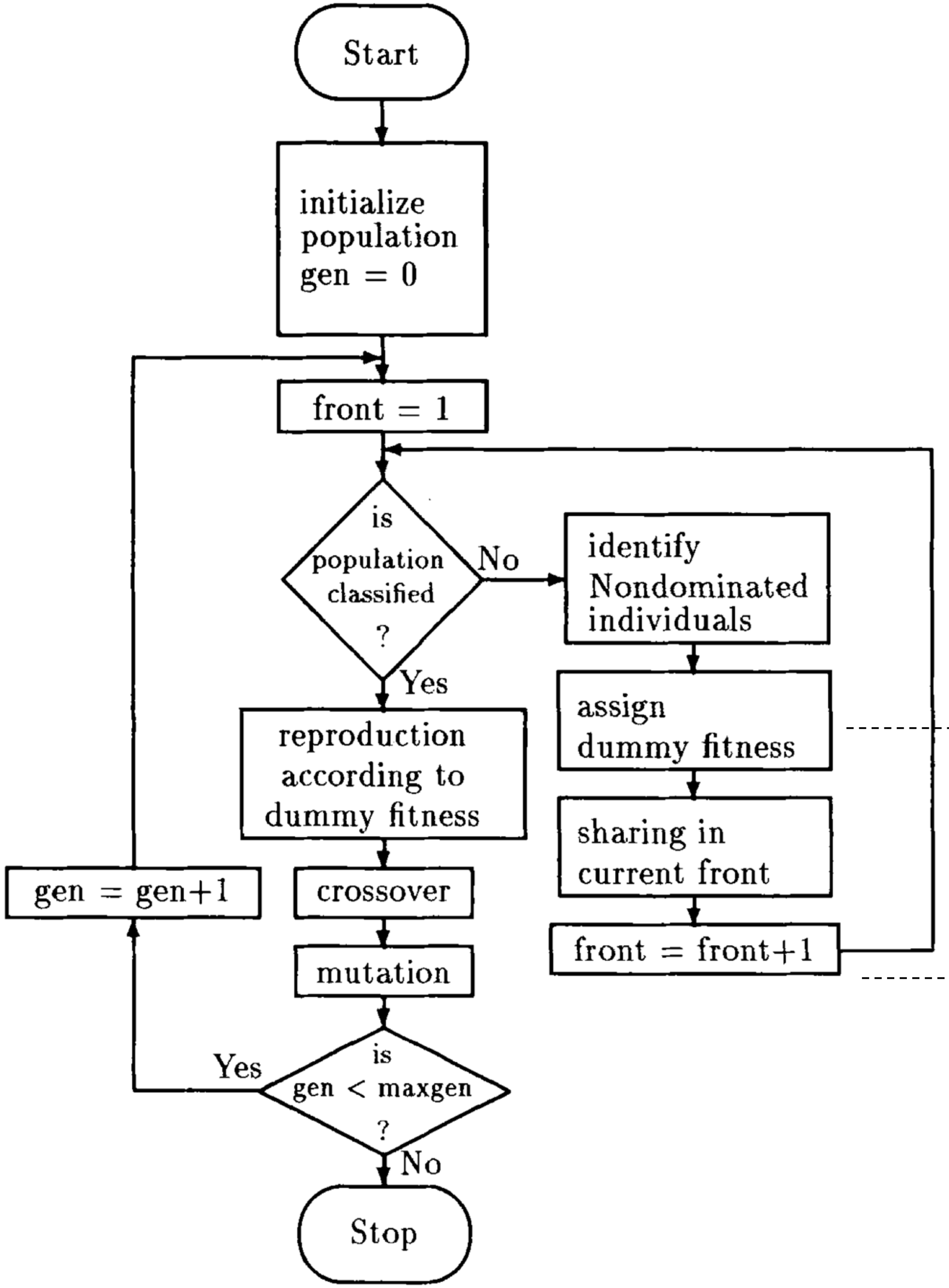
where s_i is the shared fitness of individual *i*



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

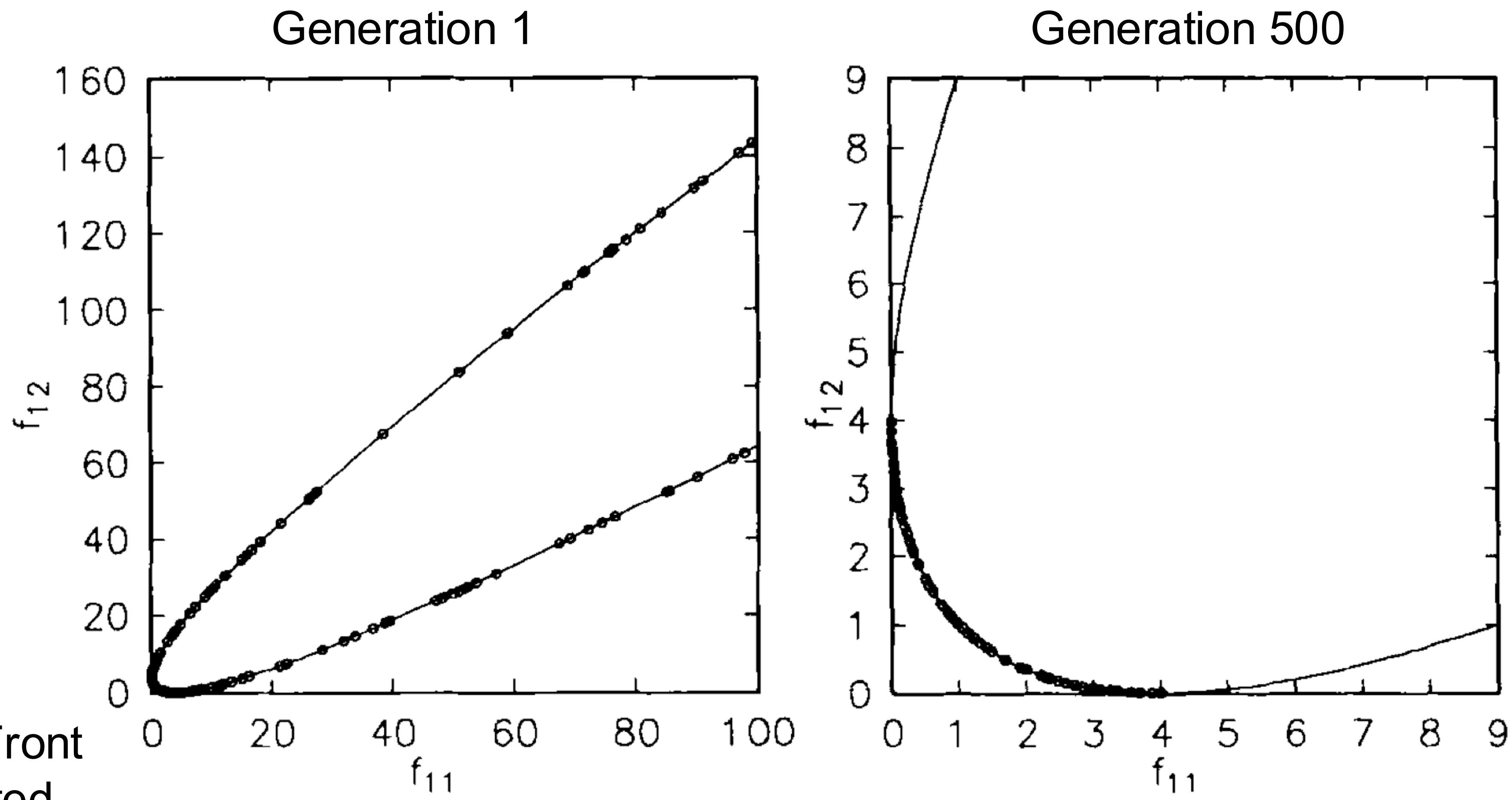
NSGA on 2-objective function of 1 variable

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

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

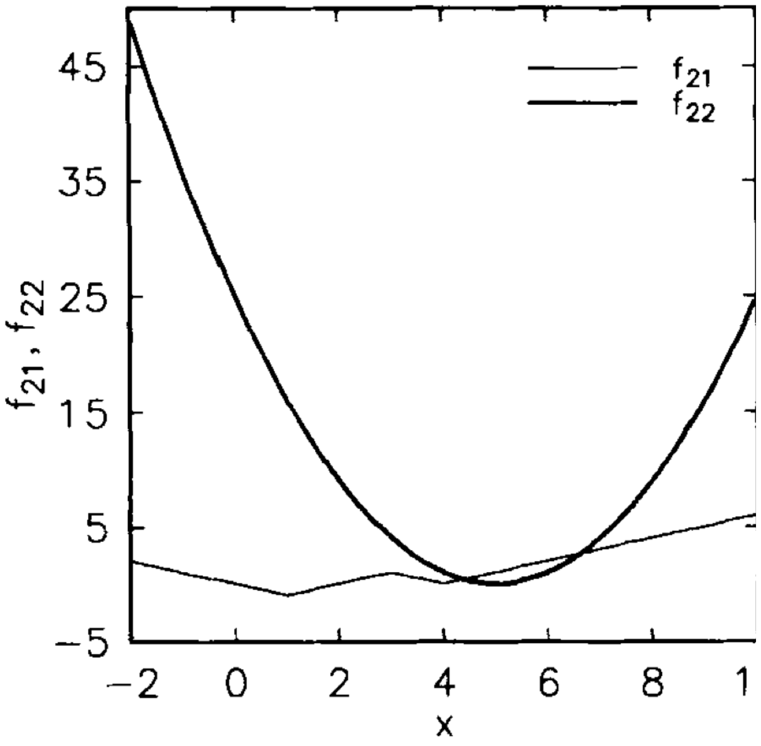


Source: Srinivas & Deb, 1994

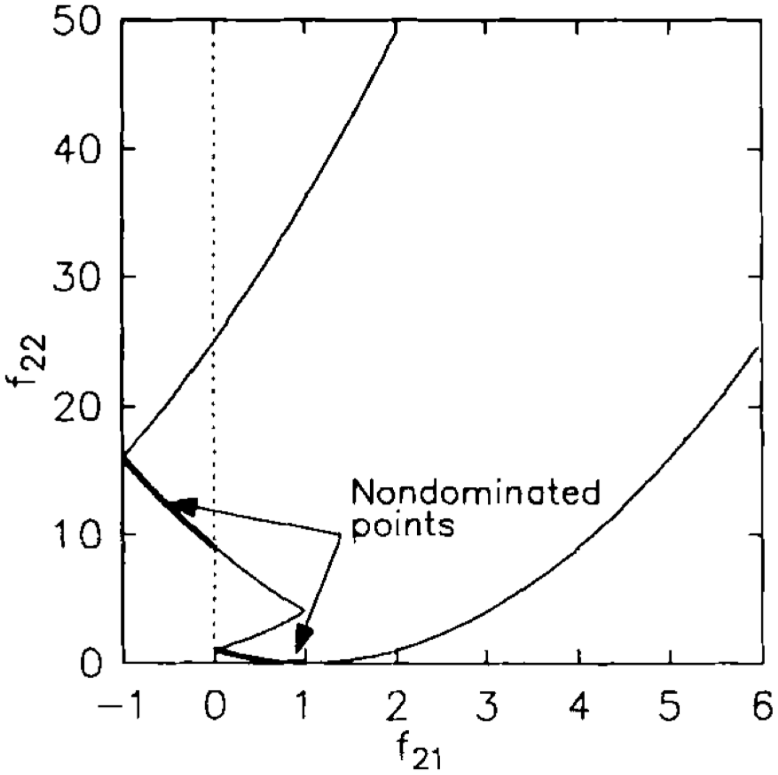
2-objective, 1 variable, non-convex and disjointed space

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

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



Objective space



PS: Remember that objective weighting does not work well on convex spaces

Source: Srinivas & Deb, 1994

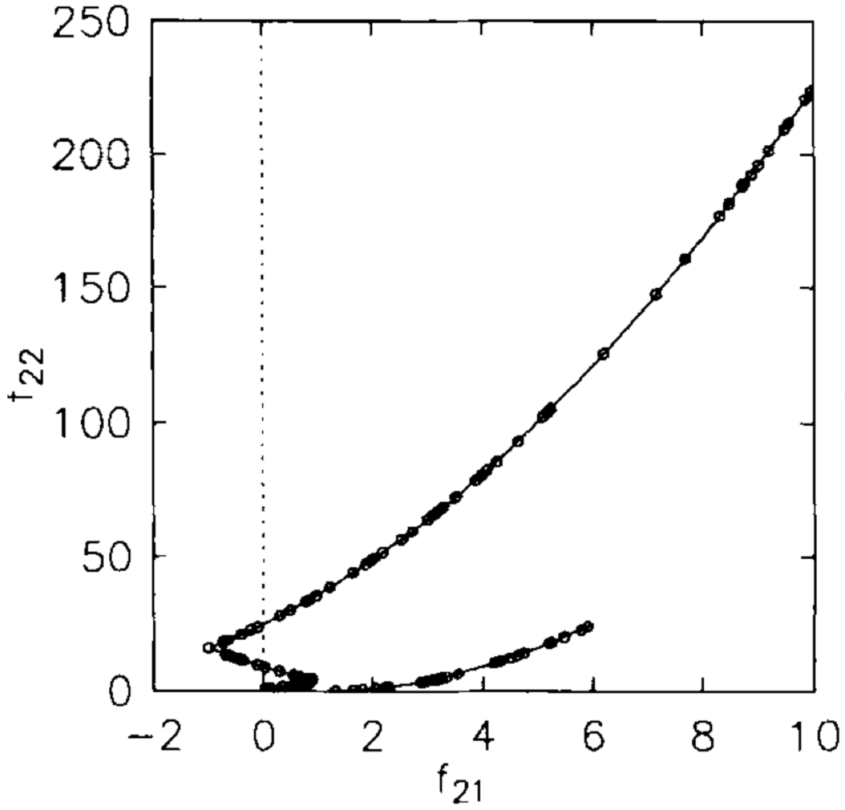
NSGA

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

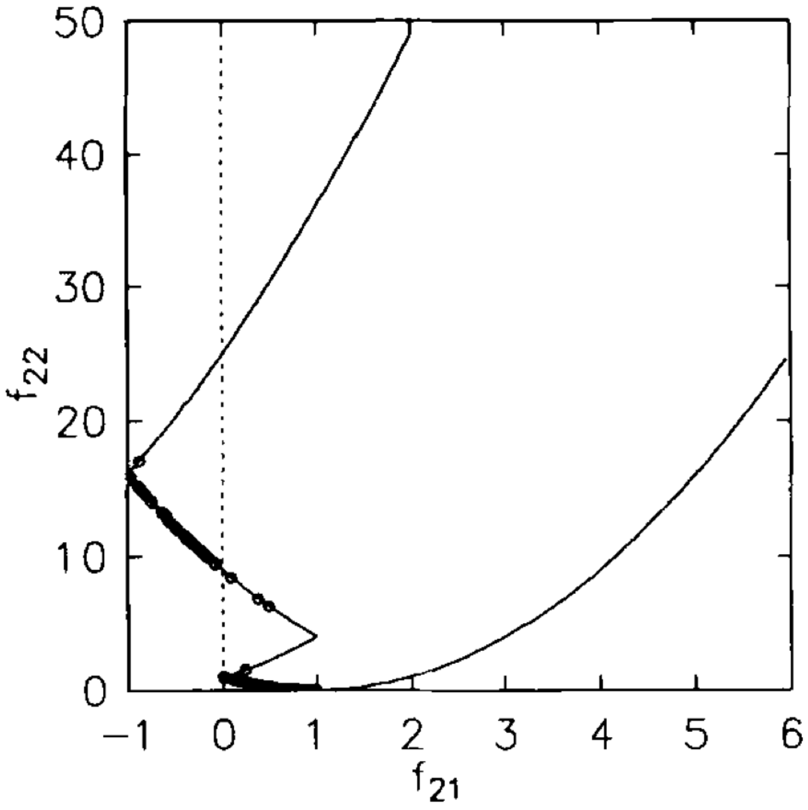
*mutation kept to 0 to highlight algorithm's search performance

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* (a.k.a. individual), 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 the remaining solutions and are assigned to Pareto front F_2

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

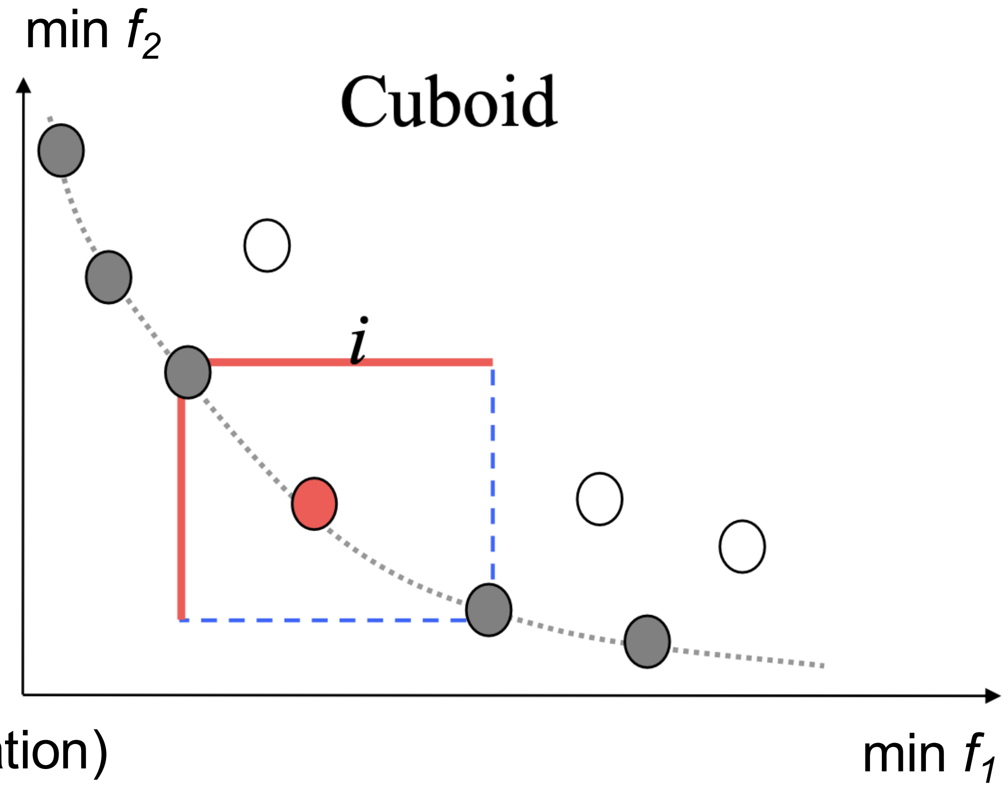
Computational complexity of this domination sorting is $O(MN^2)$

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.

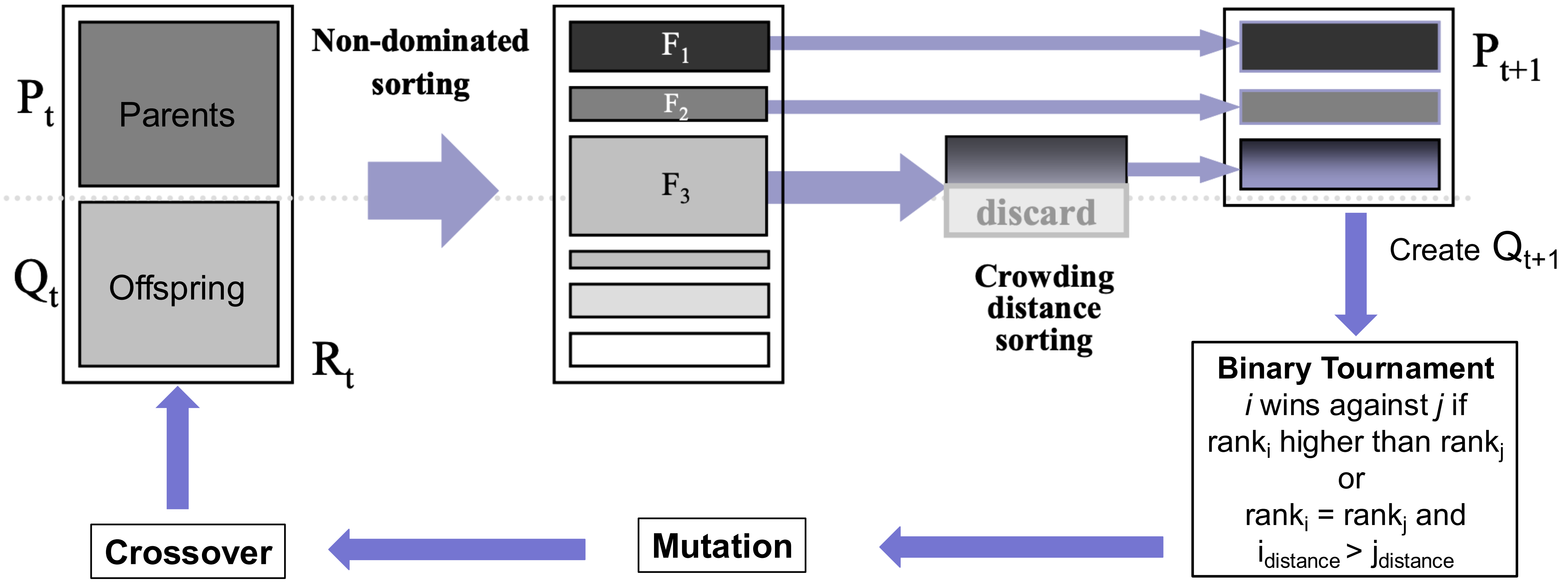
1. For each objective f_i, order solutions from best to worst and normalize their objective values
2. The normalized objective values of a solution define the position of the solution in the objective space
3. Assign infinite distance to first and last solutions to ensure high selection probability
4. 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)

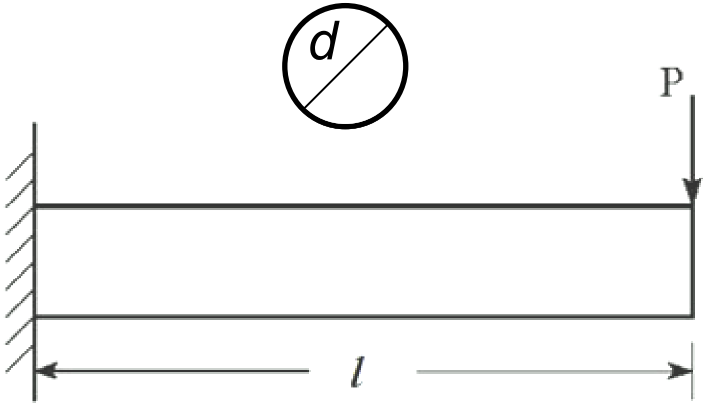
NSGA-II: Elitism by including previous parents during selection



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!

Checkpoints

- What is fitness sharing used for and how is it computed?
- Definitions of single-objective, multimodal, multi-objective functions
- Under what conditions a solution A dominates a solution B?
- What is a Pareto Frontier?
- How does Objective Weighting work in single-objective optimization?
- What are the limitations of Objective Weighting?
- What are the three steps of NSGA-II? (front identification, crowding distance, elitism)
- How does tournament selection work in NSGA-II?