Evolutionary Computing

Chapter 12

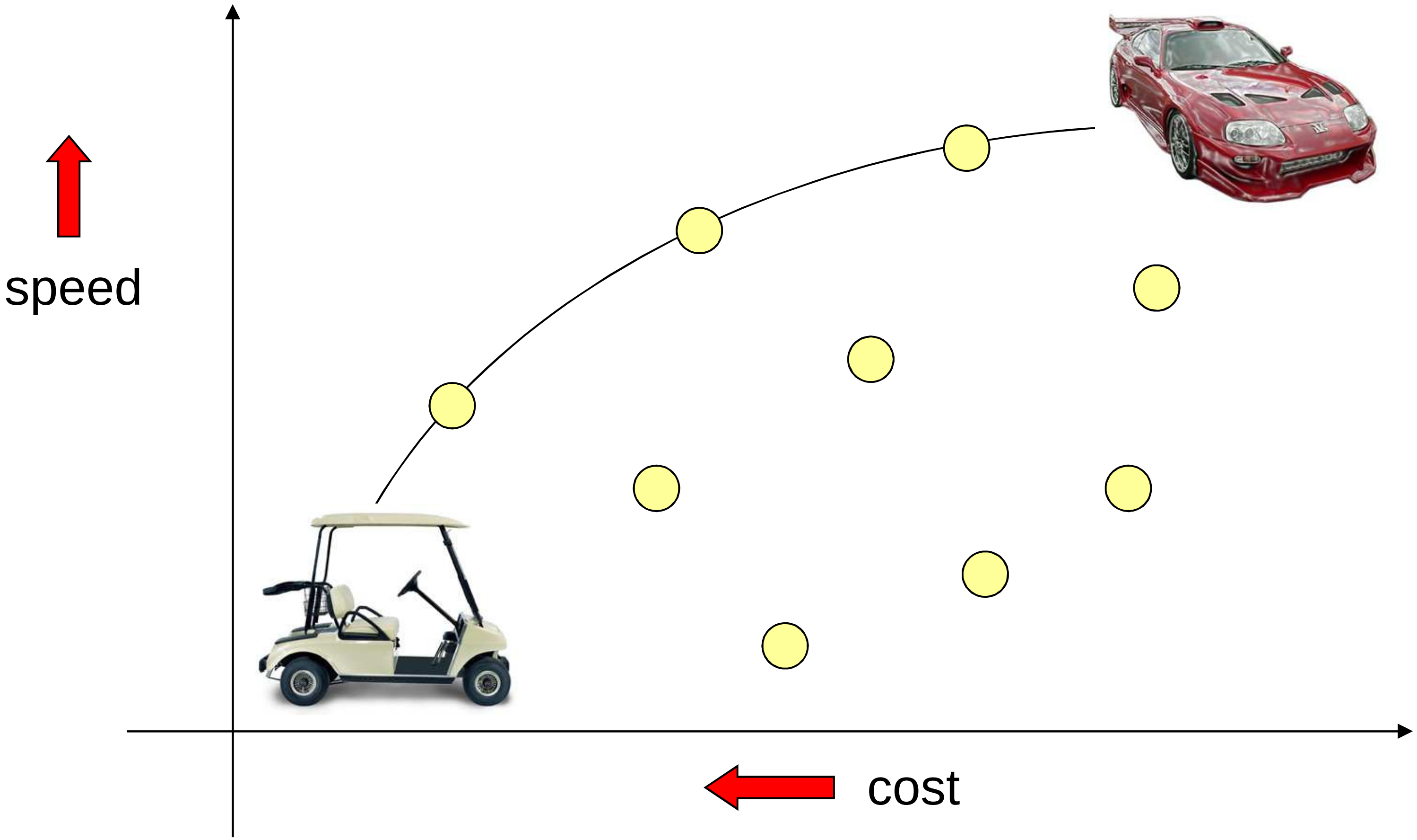
Chapter 12: Multiobjective Evolutionary Algorithms

- Multiobjective optimisation problems (MOP)
  - Pareto optimality
- EC approaches
  - Evolutionary spaces
  - Preserving diversity
- Examples of MOEAs

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 problems:
  - finding set of good solutions
  - choice of best for particular application

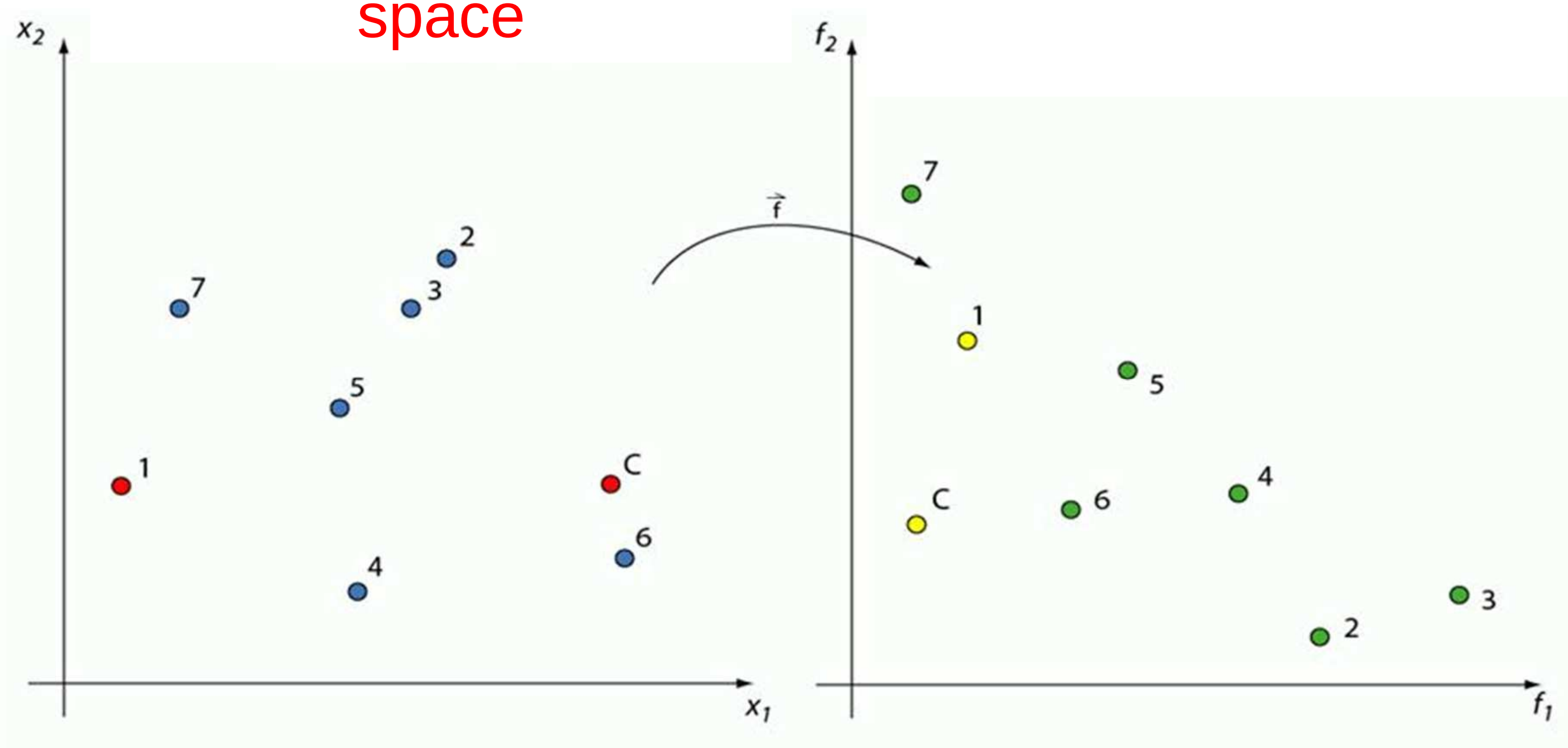
An example: Buying a car



Two spaces

Decision (variable) space

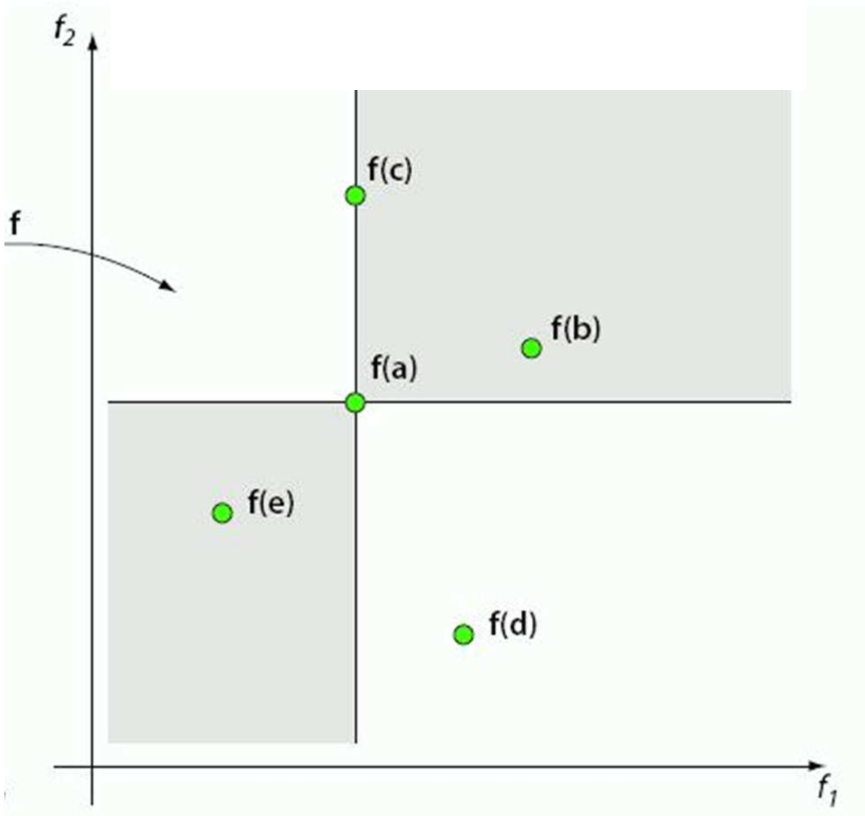
Objective space



Adapted from A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing 2014

Comparing solutions

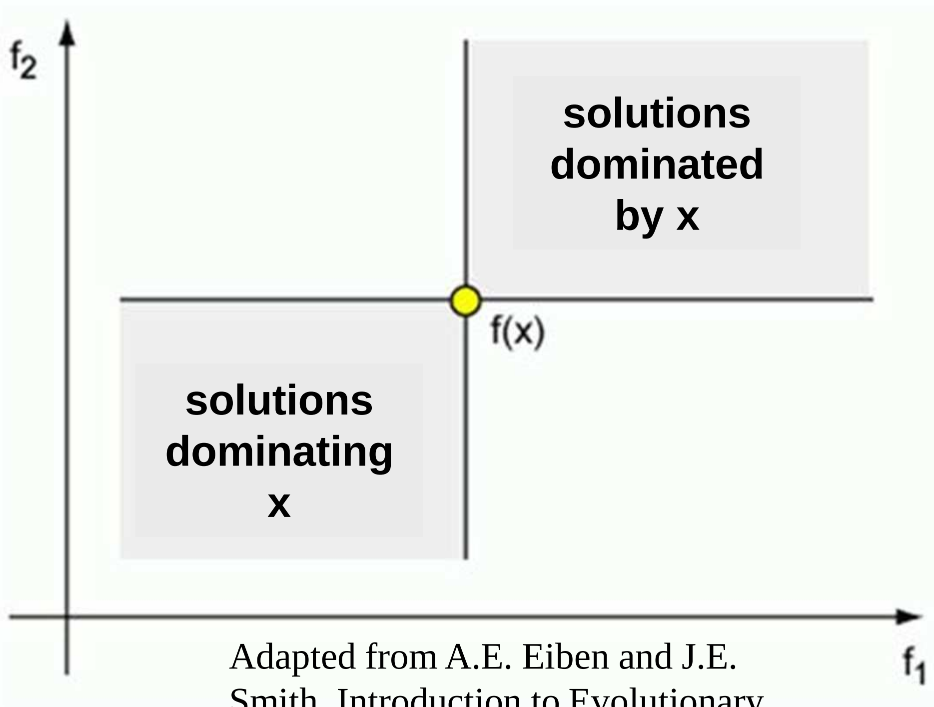
Objective space



- Optimisation task: Minimize both f_1 and f_2
- Then:
  a is better than b
  a is better than c
  a is worse than e
  a and d are incomparable

Dominance relation

- Solution x dominates solution y, ($x \preceq y$), if:
  - x is better than y in at least one objective,
  - x is not worse than y in all other objectives



Adapted from A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing 2014

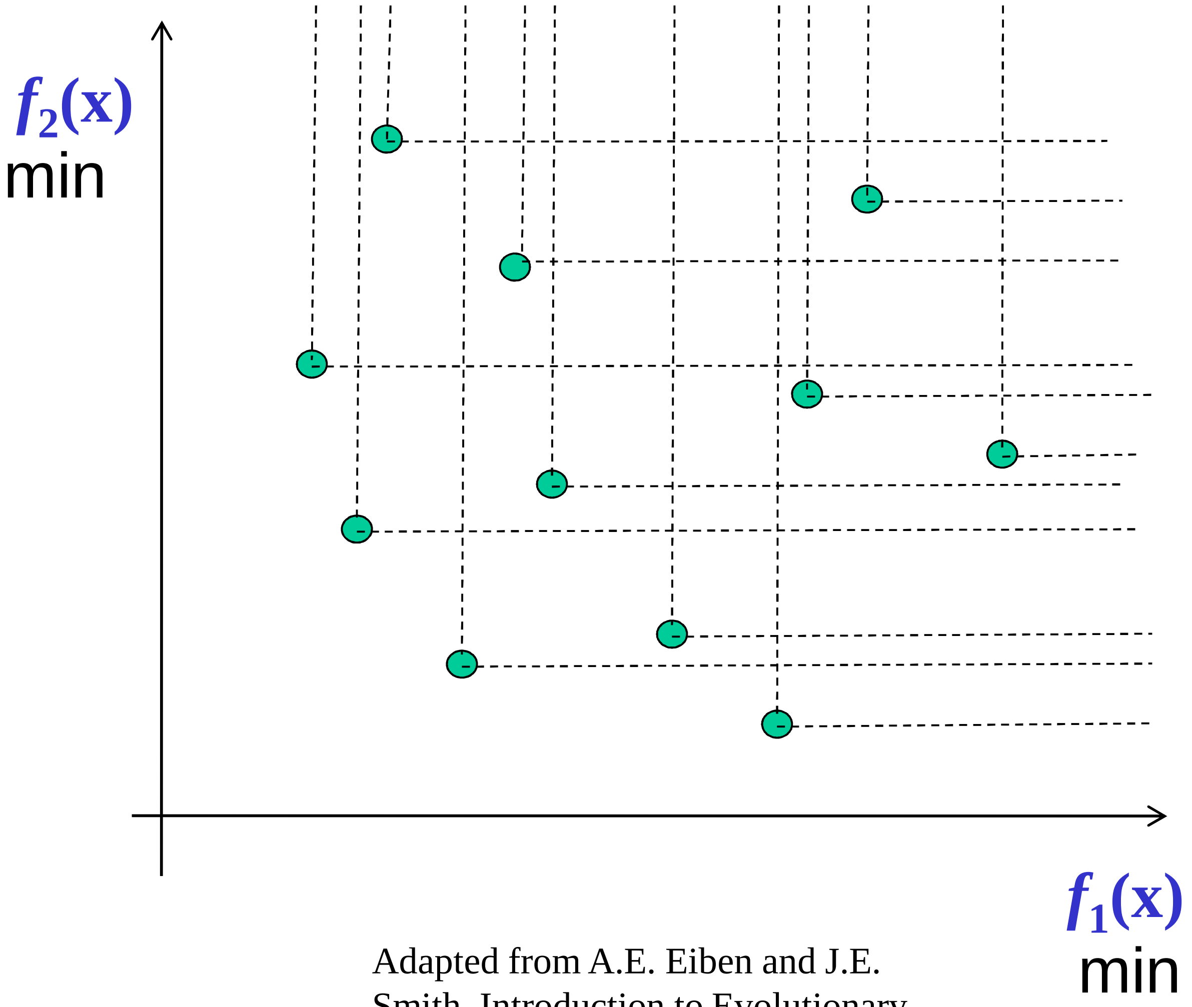
Pareto optimality

- Solution x is non-dominated among a set of solutions Q if no solution from Q dominates x

- A set of non-dominated solutions from the entire feasible solution space is the Pareto-optimal set, its members Pareto-optimal solutions

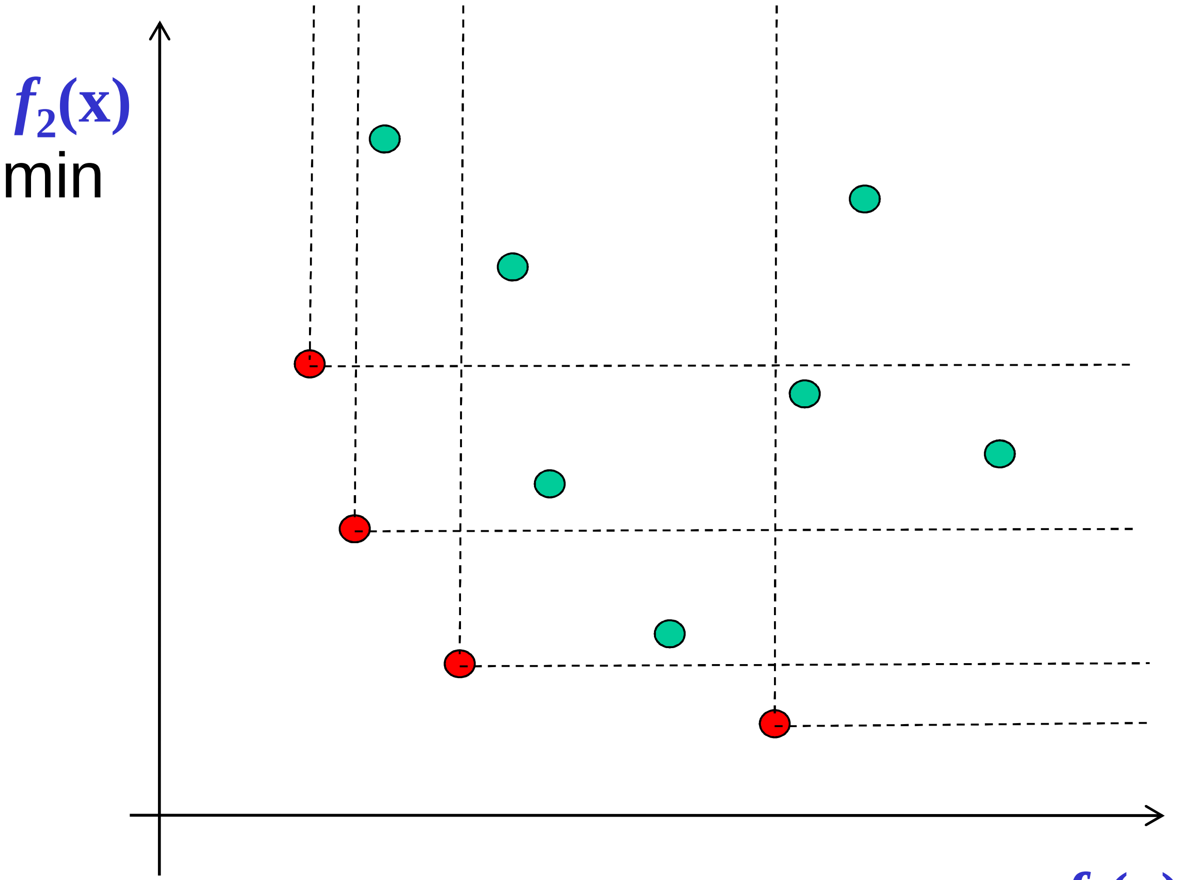
- Pareto-optimal front: an image of the Pareto-optimal set in the objective space

Illustration of the concepts



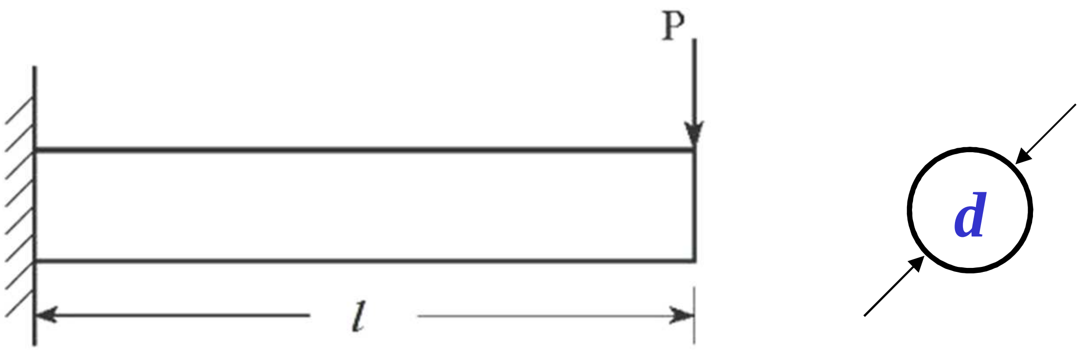
Adapted from A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing 2014

Illustration of the concepts



A practical example:
The beam design problem

Minimize weight and deflection of a beam (Deb, 2001):



Formal definition

- Minimize $f_1(d,l) = \rho \dfrac{\pi d^2}{4} l$ (beam weight)
- minimize $f_2(d,l) = \delta = \dfrac{64Pl^3}{3E\pi d^4}$ (beam deflection)
- subject to $0.01 \text{ m} \leq d \leq 0.05 \text{ m}$

$$0.2 \text{ m} \leq l \leq 1.0 \text{ m}$$

$$\sigma_{\max} = \frac{32Pl}{\pi d^3} \leq S_y$$ (maximum stress)

$$\delta \leq \delta_{\max}$$

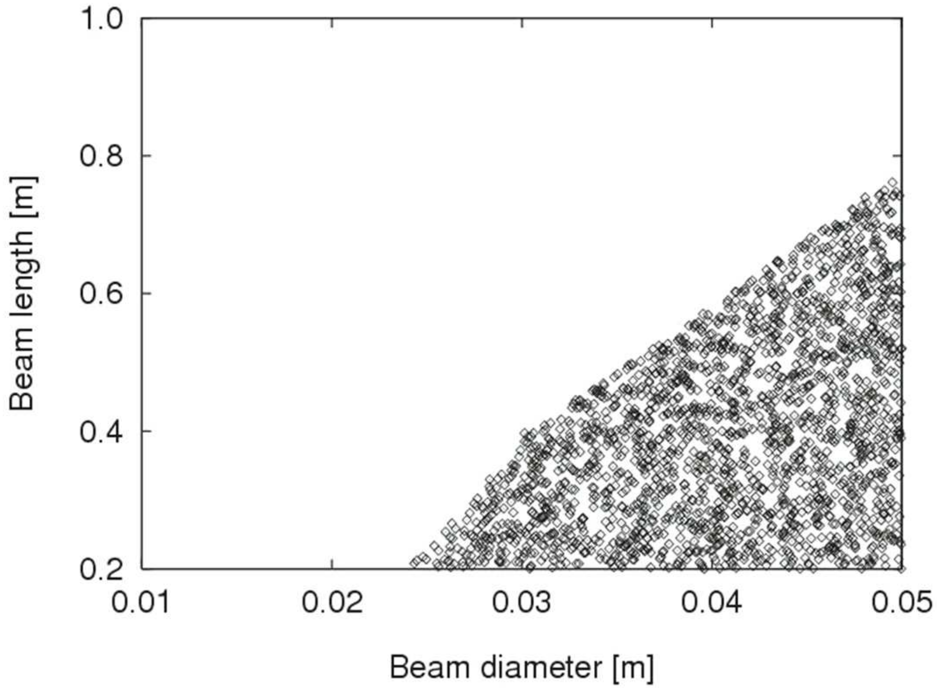
where $\rho = 7800 \text{ kg/m}^3,\ P = 2 \text{ kN}$

$$E = 207 \text{ GPa}$$

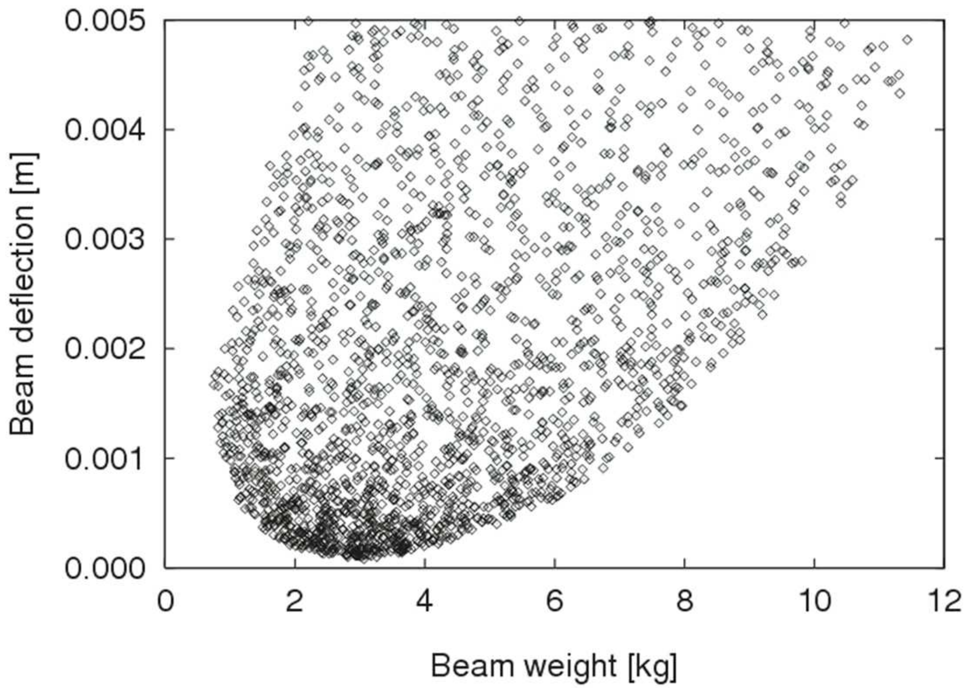
$$S_y = 300 \text{ MPa},\ \delta_{\max} = 0.005 \text{ m}$$

Feasible solutions

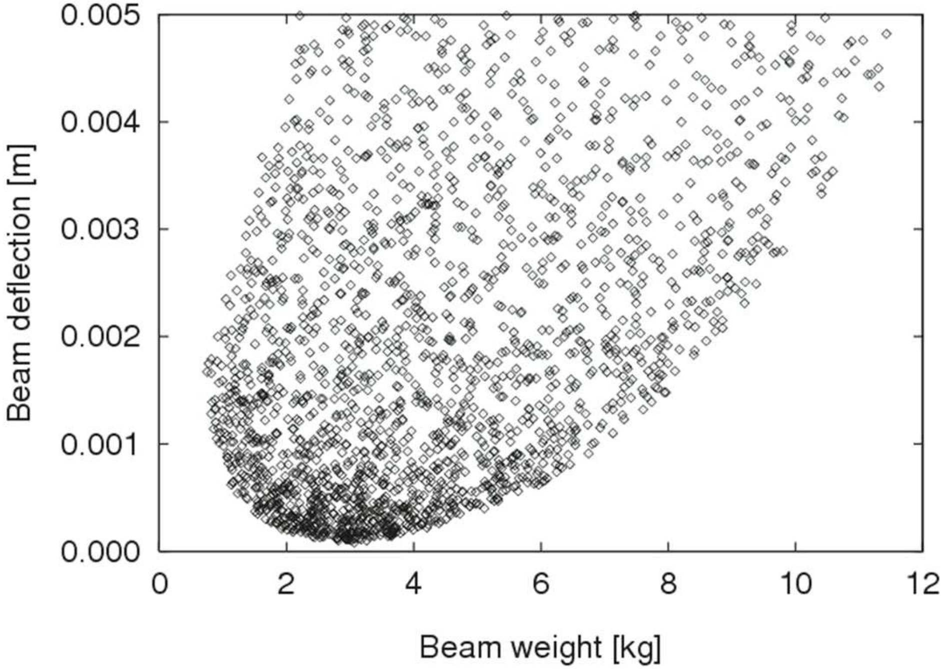
Decision (variable) space

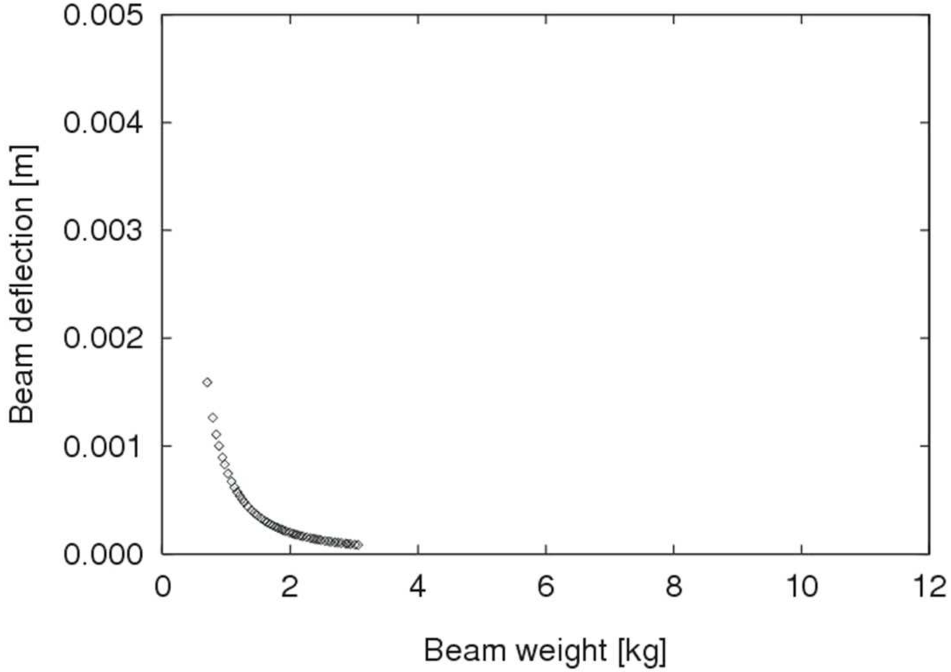


Objective space



Goal: Finding non-dominated solutions

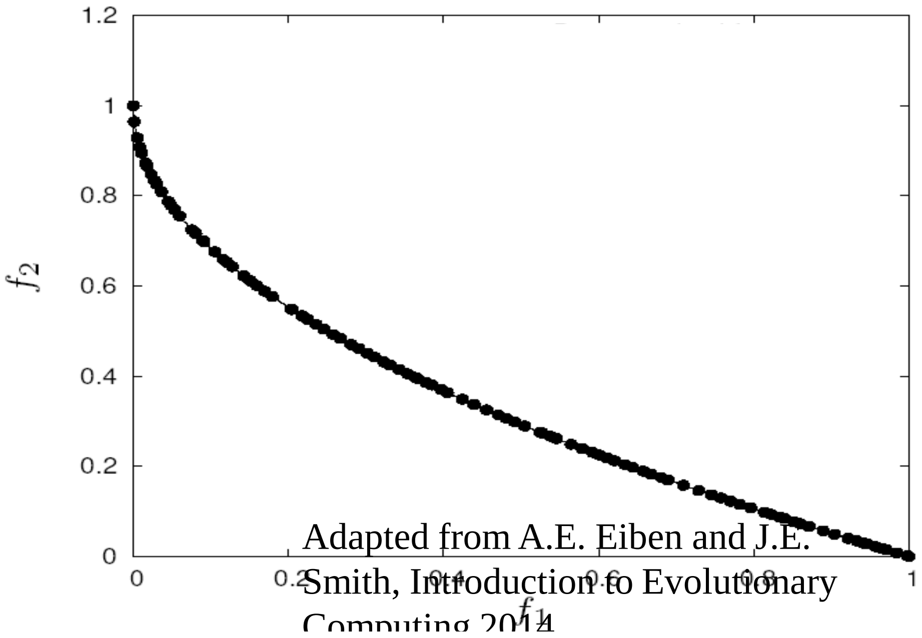




Adapted from A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing 2014

Goal of multiobjective optimisers

- Find a set of non-dominated solutions (approximation set) following the criteria of:
  - convergence (as close as possible to the Pareto-optimal front),
  - diversity (spread, distribution)



Single- vs. multiobjective optimisation

Characteristic	**Singleobjective optimisation**	**Multiobjective optimisation**
Number of objectives	one	more than one
Spaces	single	two: decision (variable) 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 goals	convergence	convergence, diversity

Two approaches to multiobjective optimisation

- Preference-based:
  traditional, using single objective optimisation methods

- Ideal:
  possible with novel multiobjective optimisation techniques, enabling better insight into the problem

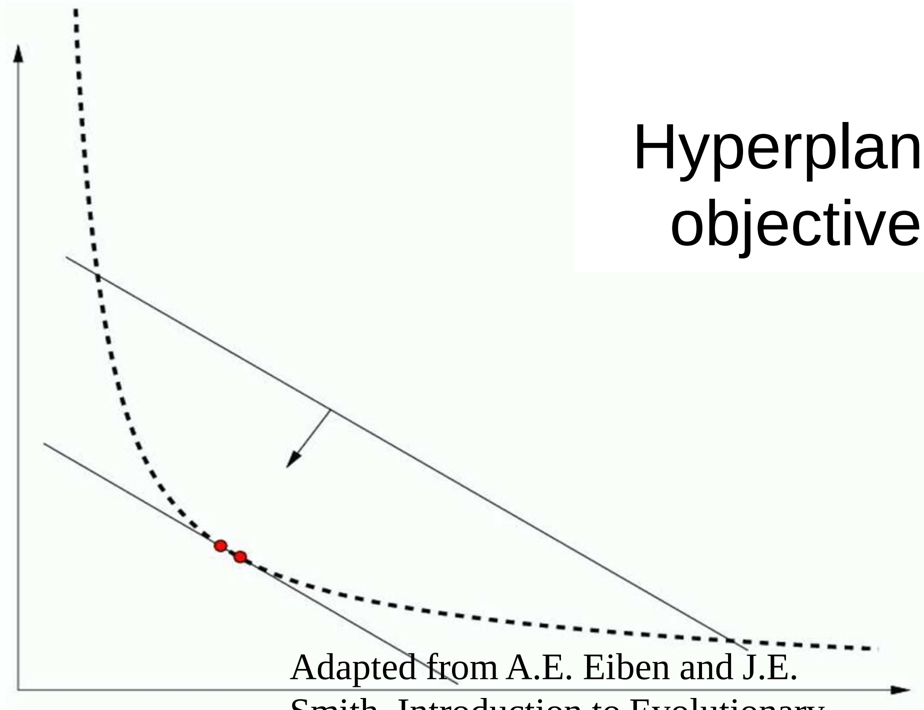
Preference-based approach

- Given a multiobjective optimisation problem,
- use higher-level information on importance of objectives
- to transform the problem into a singleobjective one,
- and then solve it with a single objective optimisation method
- to obtain a particular trade-off solution.

An example approach: Weighted-sum

- Modified problem:

$$F(\mathrm{X}) = \sum_{m=1}^{M} w_m f_m(\mathrm{X}), \quad w_m \in [0,1], \quad \sum_{m=1}^{M} w_m = 1$$



Hyperplanes in the objective space!

Adapted from A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing 2014

Ideal approach

- Given a multiobjective optimisation problem,
- solve it with a multiobjective optimisation method
- to find multiple trade-off solutions,
- and then use higher-level information
- to obtain a particular trade-off solution.

Multiobjective optimisation with evolutionary algorithms

- Population-based method
- Can return a set of trade-off solutions (approximation set) in a single run
- Allows for the ideal approach to multiobjective optimisation

EC approach: Advantages

- Population-based nature of search means you can *simultaneously* search for 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.

EC approach: Requirements

- 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 have seen
  - usually using elitism or an archive

EC approach: Fitness Assignment

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

EC approach: 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

EC approach: 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

MOP - summary

- MO problems occur very frequently
- EAs are very good in solving MO problems
- MOEAs are one of the most successful EC subareas

References

- http://pt.slideshare.net/paskorn/rnsgaii-presentation