Multi-Objective Evolutionary Algorithms

Kalyanmoy Deb [a]
Kanpur Genetic Algorithm Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, Pin 208016 INDIA
deb@iitk.ac.in
http://www.iitk.ac.in/kangal/deb.html

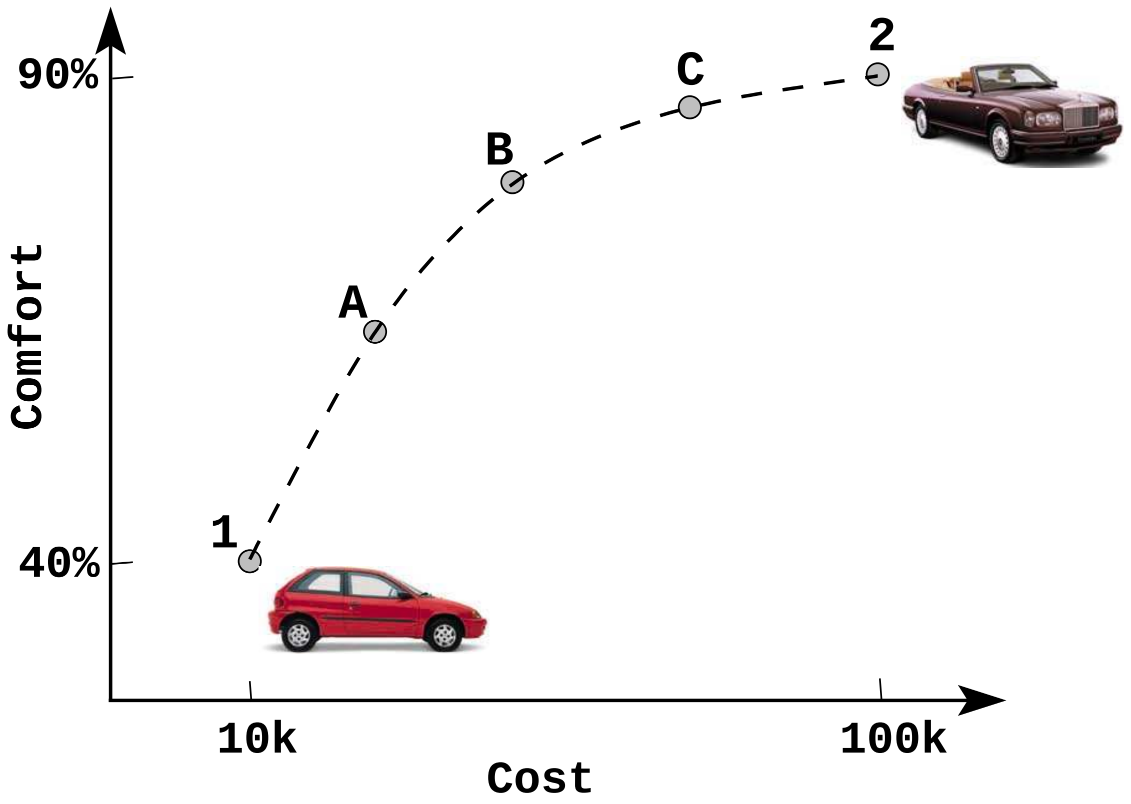
[a] Currently visiting TIK, ETH Zürich

Overview of the Tutorial

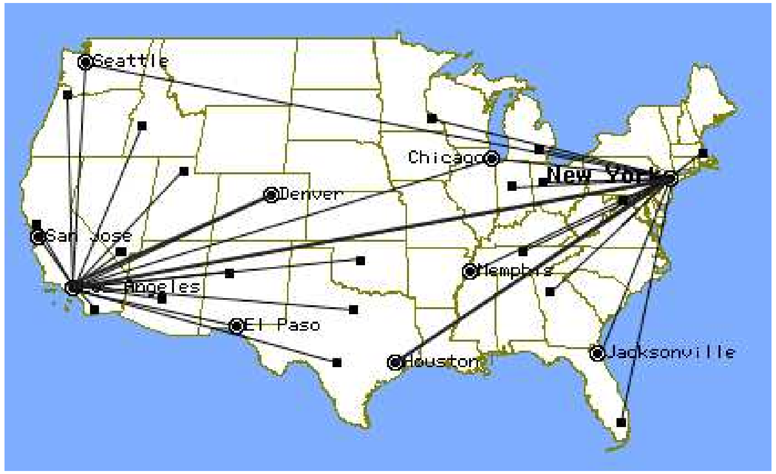
- Multi-objective optimization
- Classical methods
- History of multi-objective evolutionary algorithms (MOEAs)
- Non-elitst MOEAs
- Elitist MOEAs
- Constrained MOEAs
- Applications of MOEAs
- Salient research issues

Multi-Objective Optimization

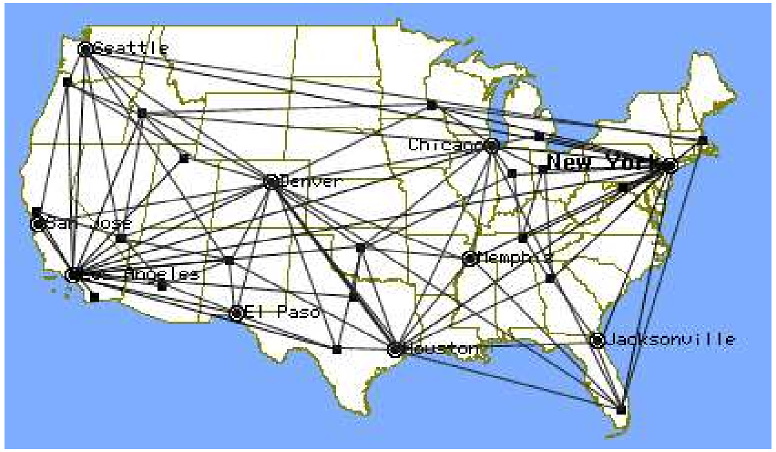
- We often face them



More Examples

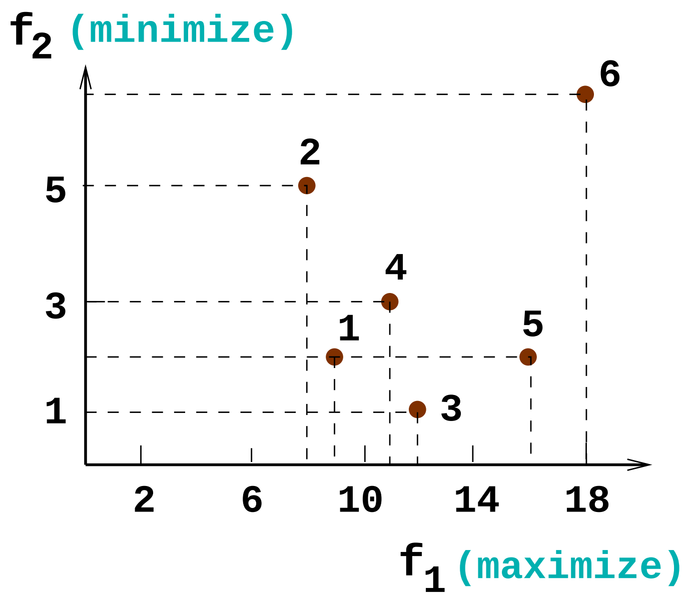


A cheaper but inconvenient flight



A convenient but expensive flight

Which Solutions are Optimal?



Domination:

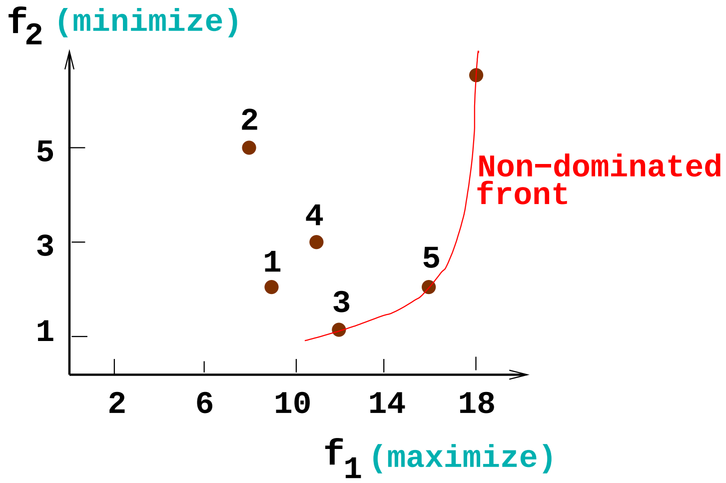
$\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$ if

1. $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ in all objectives
2. $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective

Pareto-Optimal Solutions

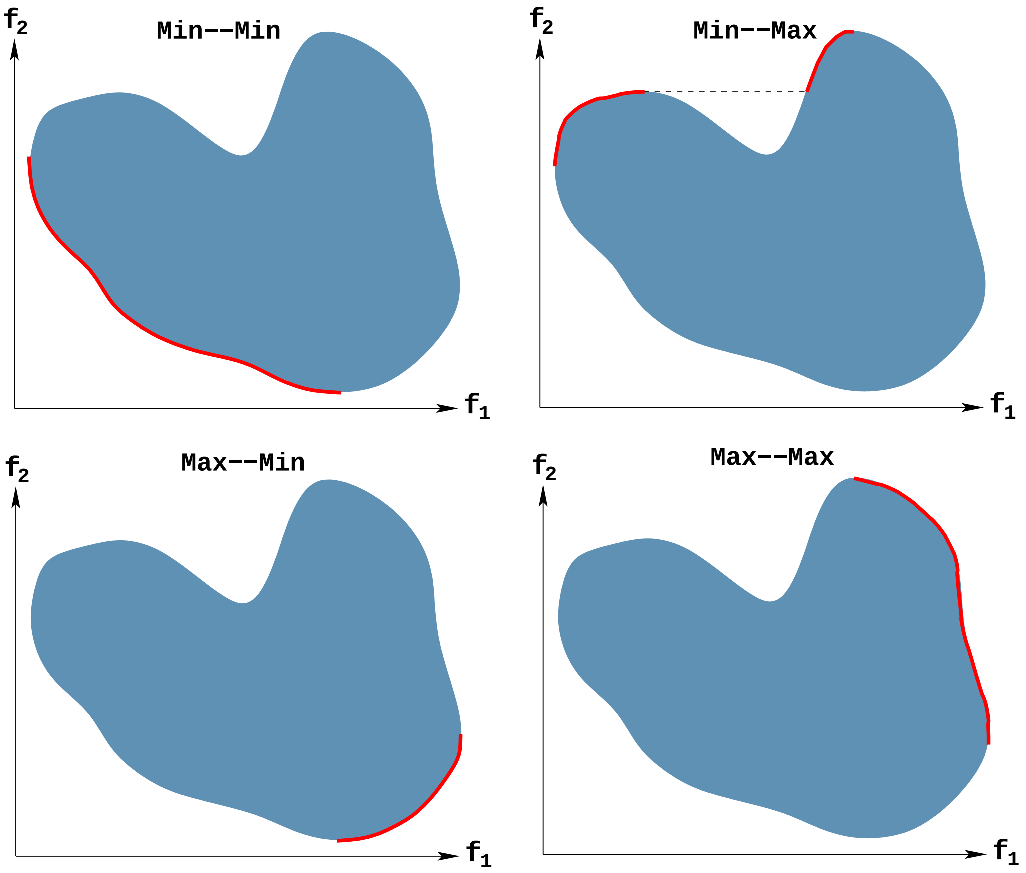
Non-dominated solutions: Among a set of solutions P, the non-dominated set of solutions P' are those that are not dominated by any member of the set P. $O(MN^2)$ algorithms exist.

Pareto-Optimal solutions: When $P = \mathcal{S}$, the resulting P' is Pareto-optimal set

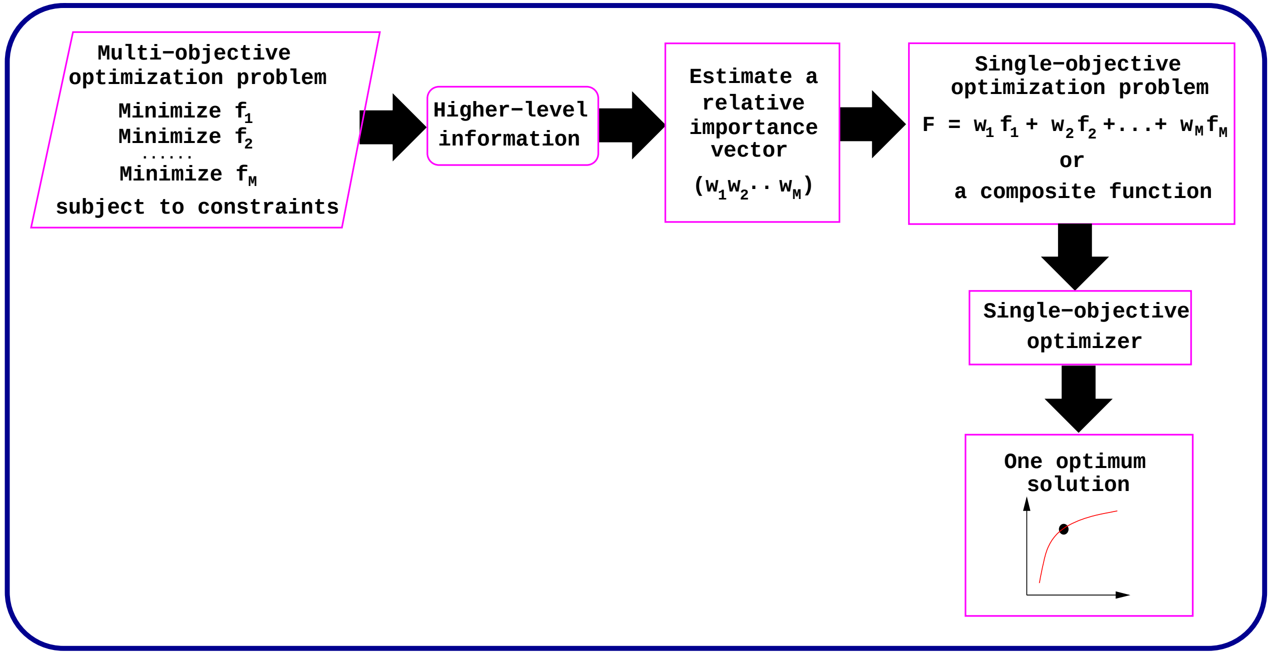


A number of solutions are optimal

Pareto-Optimal Fronts



Preference-Based Approach



- Classical approaches follow it

Classical Approaches

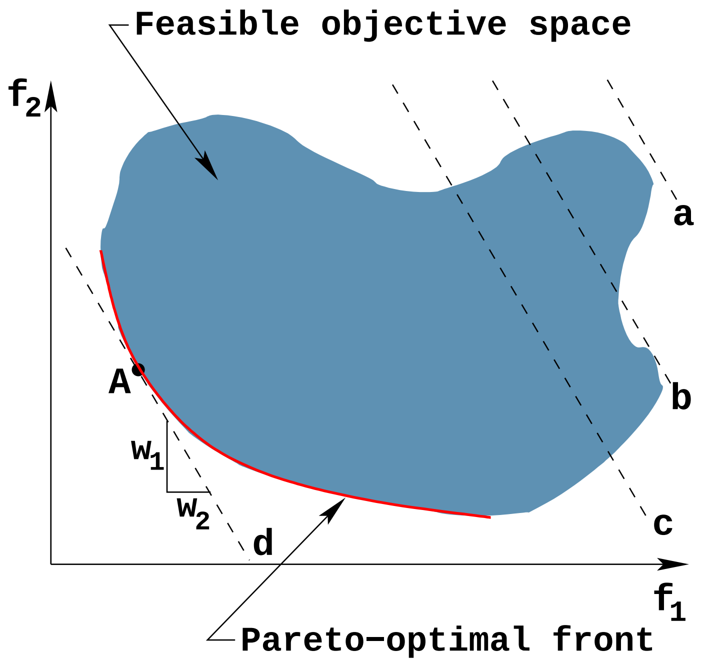
- No Preference methods (heuristic-based)
- Posteriori methods (generating solutions)
- A priori methods (one preferred solution)
- Interactive methods (involving a decision-maker)

Weighted Sum Method

- Construct a weighted sum of objectives and optimize

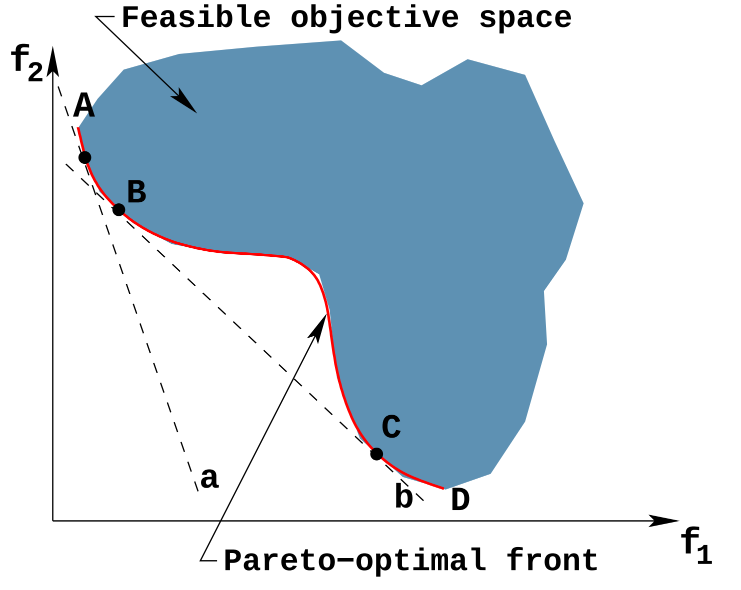
$$F(\mathbf{x}) = \sum_{m=1}^{M} w_m f_m(\mathbf{x}).$$

- User supplies weight vector **w**



Difficulties with Weighted Sum Method

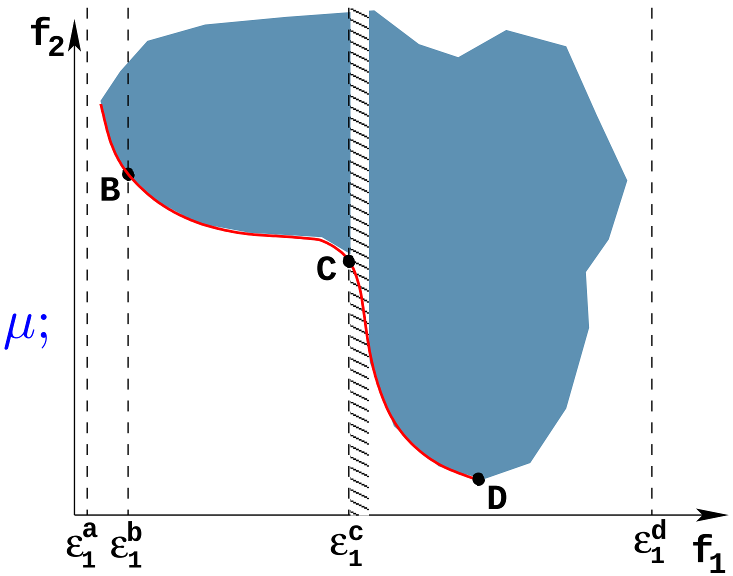
- Need to know **w**
- Non-uniformity in Pareto-optimal solutions
- Inability to find some Pareto-optimal solutions



ϵ-Constraint Method

- Optimize one objective, constrain all other

$$\text{Minimize} \quad f_\mu(\mathbf{x}),$$
$$\text{subject to} \quad f_m(\mathbf{x}) \le \epsilon_m, \; m \neq \mu;$$



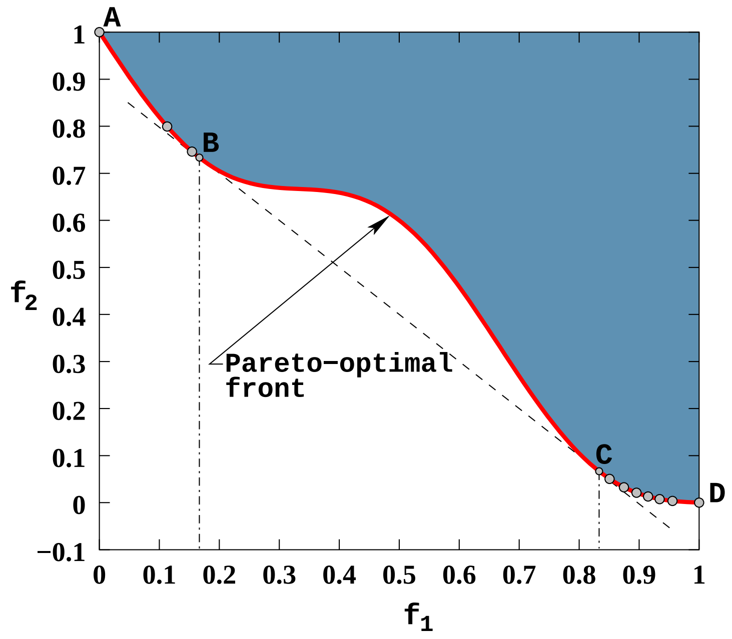
- User supplies a ϵ vector

- Need to know relevant ϵ vectors

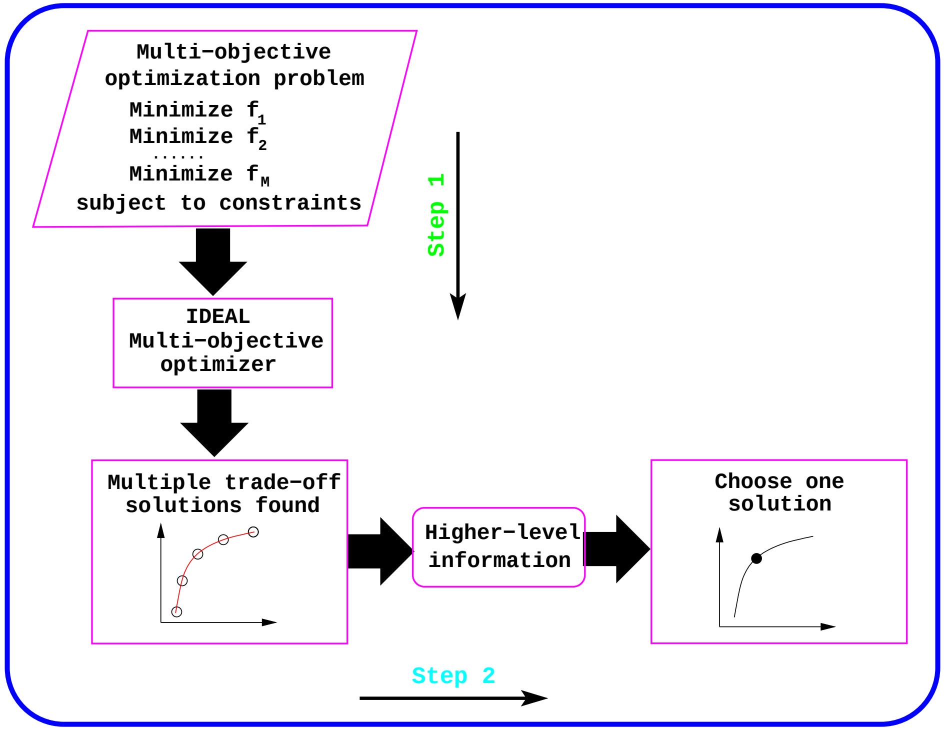
- Non-uniformity in Pareto-optimal solutions

Difficulties with Most Classical Methods

- Need to run a single-objective optimizer many times
- Expect a lot of problem knowledge
- Even then, good distribution is not guaranteed
- Multi-objective optimization as an application of single-objective optimization



Ideal Multi-Objective Optimization

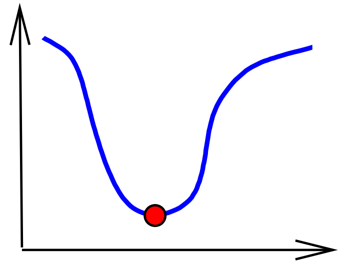


Step 1 Find a set of Pareto-optimal solutions

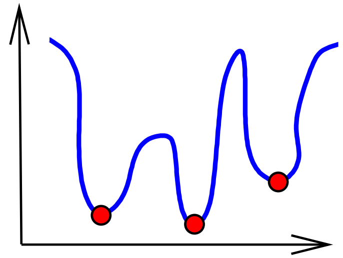
Step 2 Choose one from the set

Advantages of Ideal Multi-Objective Optimization

- Decision-making becomes easier and less subjective



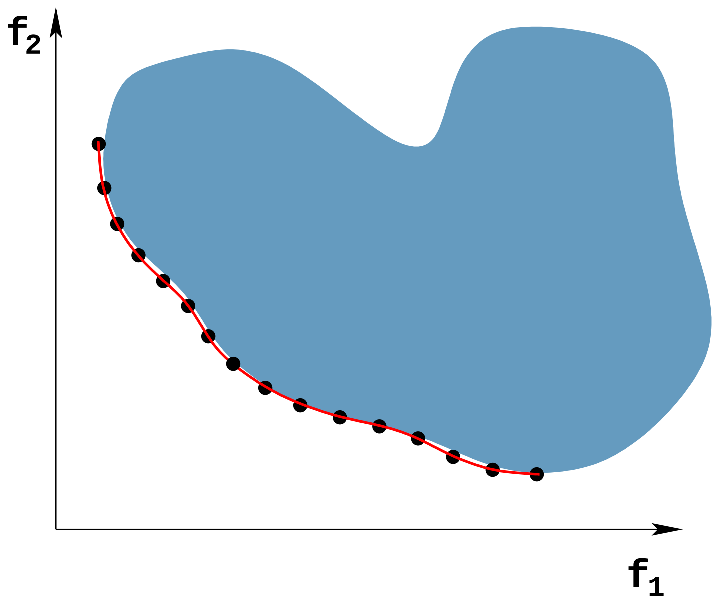
- Single-objective optimization is a degenerate case of multi-objective optimization
  - Step 1 finds a single solution
  - No need for Step 2



- Multi-modal optimization is a special case of multi-objective optimization

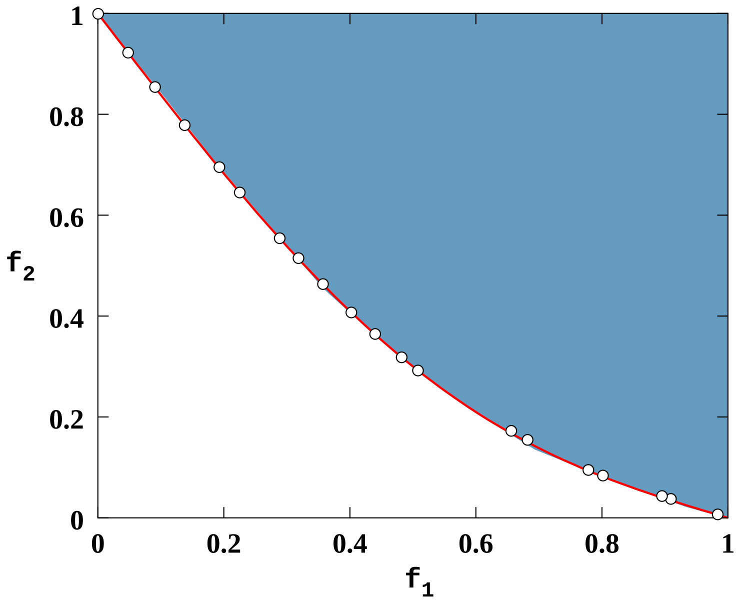
Two Goals in Ideal Multi-Objective Optimization

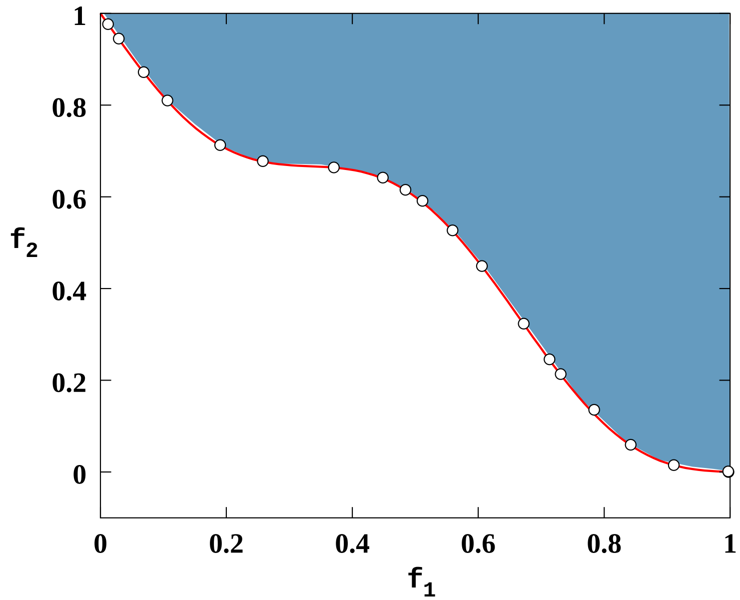
1. Converge on the Pareto-optimal front
2. Maintain as diverse a distribution as possible



Why Evolutionary?

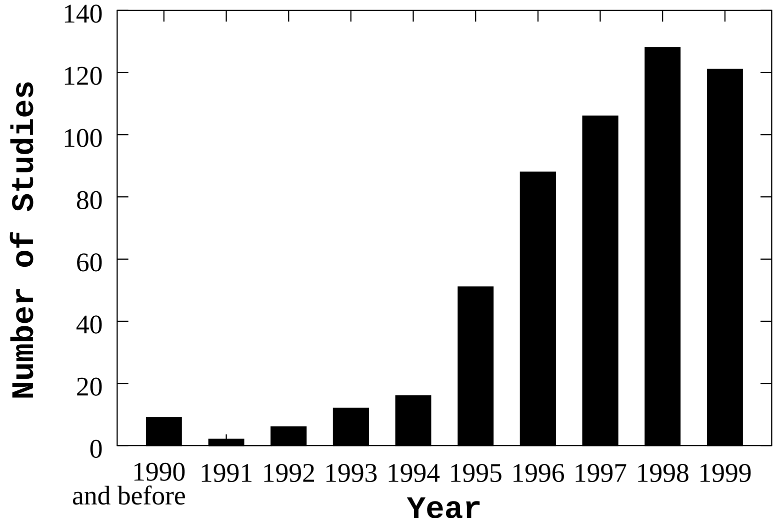
- Population approach suits well to find multiple solutions
- Niche-preservation methods can be exploited to find diverse solutions

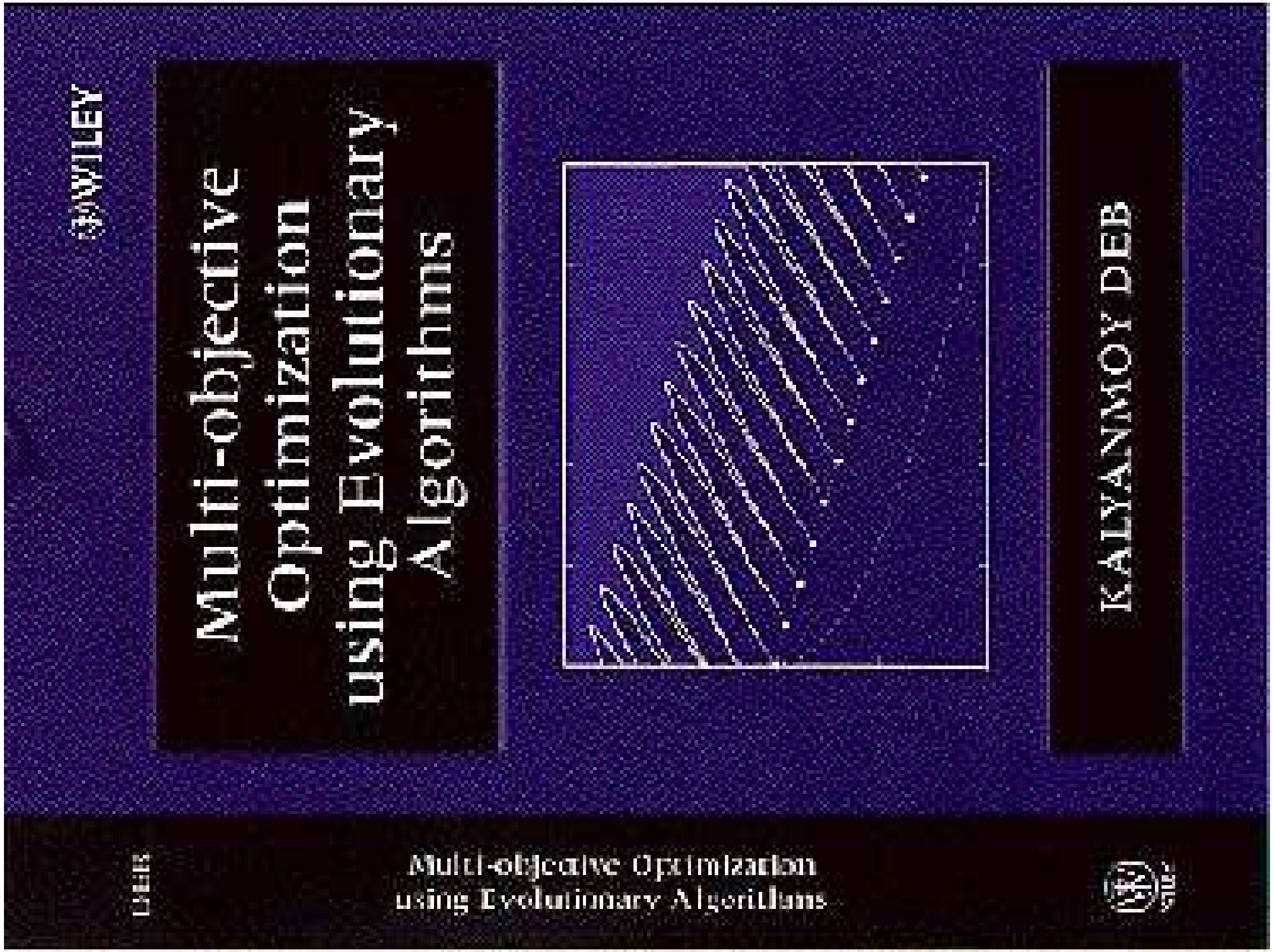




History of Multi-Objective Evolutionary Algorithms (MOEAs)

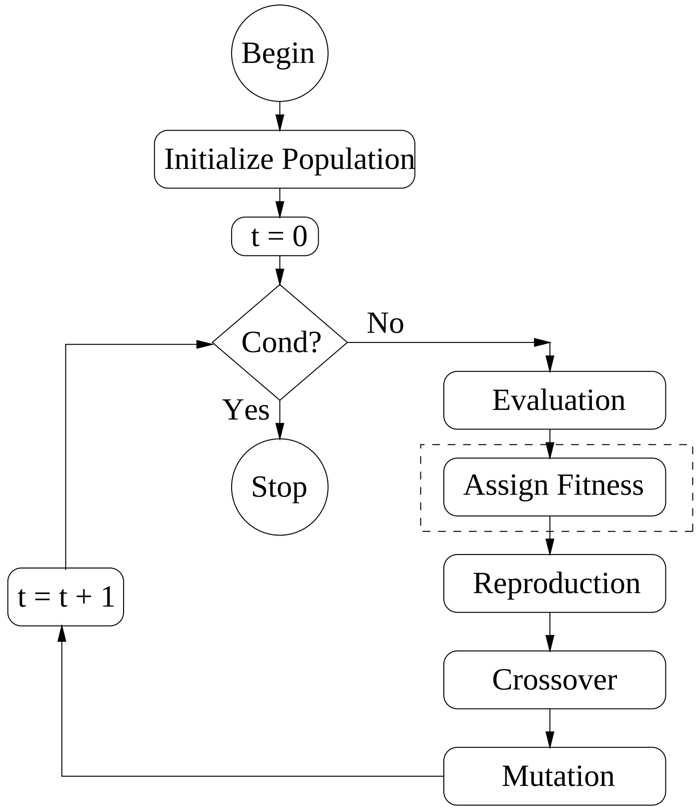
- Early penalty-based approaches
- VEGA (1984)
- Goldberg's suggestion (1989)
- MOGA, NSGA, NPGA (1993-95)
- Elitist MOEAs (SPEA, NSGA-II, PAES, MOMGA etc.) (1998 – Present)




WILEY
Multi-objective Optimization using Evolutionary Algorithms
KALYANMOY DEB
DEB
Multi-objective Optimization using Evolutionary Algorithms

What to Change in a Simple GA?

- Modify the fitness computation



Identifying the Non-dominated Set

Step 1 Set $i = 1$ and create an empty set P'.

Step 2 For a solution $j \in P$ (but $j \neq i$), check if solution j dominates solution i. If yes, go to Step 4.

Step 3 If more solutions are left in P, increment j by one and go to Step 2; otherwise, set $P' = P' \cup \{i\}$.

Step 4 Increment i by one. If $i \leq N$, go to Step 2; otherwise stop and declare P' as the non-dominated set.

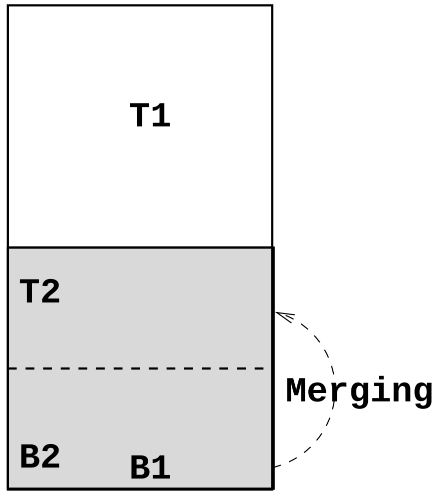
$O(MN^2)$ computational complexity

An Efficient Approach

Kung et al.'s algorithm (1975)

Step 1 Sort the population in descending order of importance of f_1

Step 2, Front(P) If $|P| = 1$, return P as the output of **Front(P)**. Otherwise, $T = \textbf{Front}(P^{(1)} - - P^{(|P|/2)})$ and $B = \textbf{Front}(P^{(|P|/2+1)} - - P^{(|P|)})$. If the i-th solution of B is not dominated by any solution of T, create a merged set $M = T \cup \{i\}$. Return M as the output of **Front(P)**.



$O\left(N(\log N)^{M-2}\right)$ for $M \geq 4$ and $O(N \log N)$ for $M = 2$ and 3

A Simple Non-dominated Sorting Algorithm

- Identify the best non-dominated set
- Discard them from population
- Identify the next-best non-dominated set
- Continue till all solutions are classified
- We discuss a $O(MN^2)$ algorithm later

Non-Elitist MOEAs

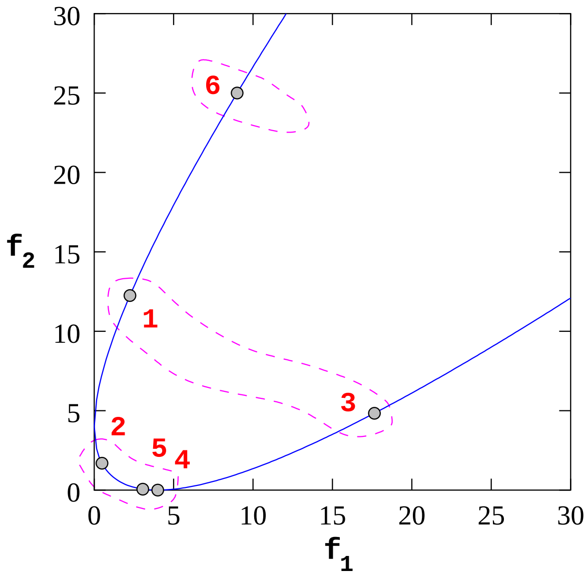
- Vector evaluated GA (VEGA) (Schaffer, 1984)
- Vector optimized EA (VOES) (Kursawe, 1990)
- Weight based GA (WBGA) (Hajela and Lin, 1993)
- Multiple objective GA (MOGA) (Fonseca and Fleming, 1993)
- Non-dominated sorting GA (NSGA) (Srinivas and Deb, 1994)
- Niched Pareto GA (NPGA) (Horn et al., 1994)
- Predator-prey ES (Laumanns et al., 1998)
- Other methods: Distributed sharing GA, neighborhood constrained GA, Nash GA etc.

Non-Dominated Sorting GA (NSGA)

- A non-dominated sorting of the population
- First front: Fitness $F = N$ to all
- Niching among all solutions in first front
- Note worst fitness (say F_w^1)
- Second front: Fitness $F_w^1 - \epsilon_1$ to all
- Niching among all solutions in second front
- Continue till all fronts are assigned a fitness

Non-Dominated Sorting GA (NSGA)

x	f_1	f_2	Front	Fitness before	Fitness after
−1.50	2.25	12.25	2	3.00	3.00
0.70	0.49	1.69	1	6.00	6.00
4.20	17.64	4.84	2	3.00	3.00
2.00	4.00	0.00	1	6.00	3.43
1.75	3.06	0.06	1	6.00	3.43
−3.00	9.00	25.00	3	2.00	2.00



- Niching in *parameter* space
- Non-dominated solutions are emphasized
- Diversity among them is maintained

Vector-Evaluated GA (VEGA)

- Divide population into M equal blocks
- Each block is reproduced with one objective function
- Complete population participates in crossover and mutation
- Bias towards to individual best objective solutions
- A non-dominated selection: Non-dominated solutions are assigned more copies
- Mate selection: Two distant (in parameter space) solutions are mated
- Both necessary aspects missing in one algorithm

Multi-Objective GA (MOGA)

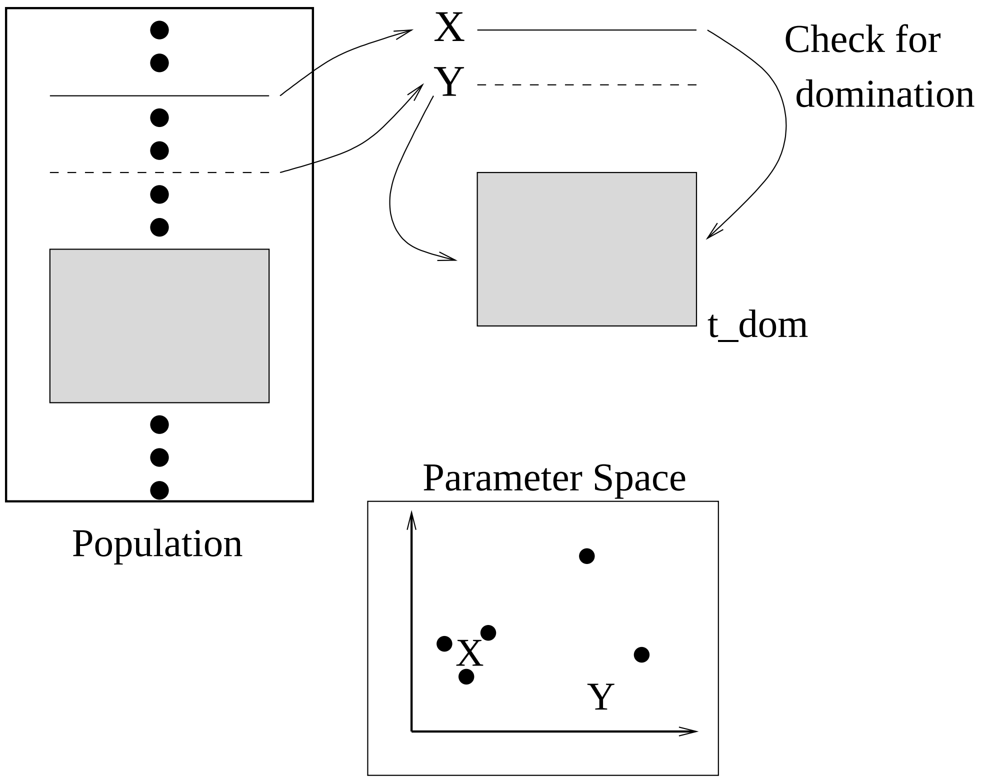
- Count the number of dominated solutions (say n)
- Fitness: $F = n + 1$
- A fitness ranking adjustment
- Niching in *fitness* space
- Rest all are similar to NSGA

	F	Asgn.	Fit.
1	2	3	2.5
2	1	6	5.0
3	2	2	2.5
4	1	5	5.0
5	1	4	5.0
6	3	1	1.0

Niched Pareto GA (NPGA)

- Solutions in a tournament are checked for domination with respect to a small subpopulation (t_{dom})
- If one dominated and other non-dominated, select second
- If both non-dominated or both dominated, choose the one with smaller niche count in the subpopulation
- Algorithm depends on t_{dom}
- Nevertheless, it has both necessary components

NPGA (cont.)

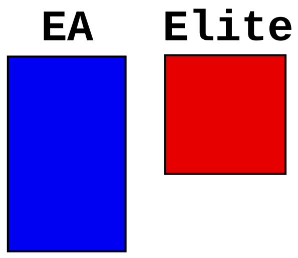


X
Y
Check for
domination
t_dom
Population
Parameter Space
X
Y

Shortcoming of Non-Elitist MOEAs

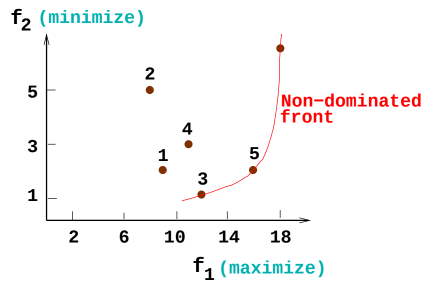
- Elite-preservation is missing
- Elite-preservation is important for proper convergence in SOEAs
- Same is true in MOEAs
- Three tasks
  - Elite preservation
  - Progress towards the Pareto-optimal front
  - Maintain diversity among solutions

Elitist MOEAs



Elite-preservation:

- Maintain an archive of non-dominated solutions

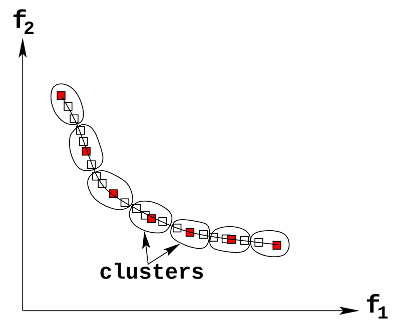


Progress towards Pareto-optimal front:

- Preferring non-dominated solutions

Maintaining spread of solutions:

- Clustering, niching, or grid-based competition for a place in the archive



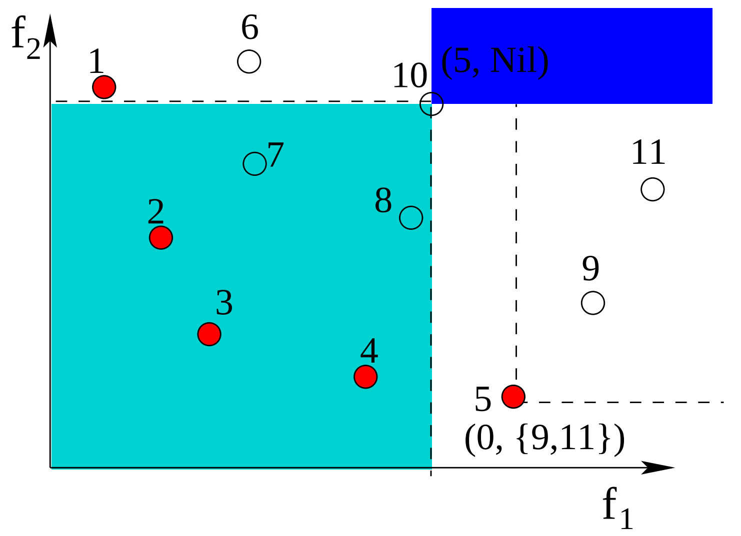
Elitist MOEAs (cont.)

- Distance-based Pareto GA (DPGA) (Osyczka and Kundu, 1995)
- Thermodynamical GA (TDGA) (Kita et al., 1996)
- Strength Pareto EA (SPEA) (Zitzler and Thiele, 1998)
- Non-dominated sorting GA-II (NSGA-II) (Deb et al., 1999)
- Pareto-archived ES (PAES) (Knowles and Corne, 1999)
- Multi-objective Messy GA (MOMGA) (Veldhuizen and Lamont, 1999)
- Other methods: Pareto-converging GA, multi-objective micro-GA, elitist MOGA with coevolutionary sharing

Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)

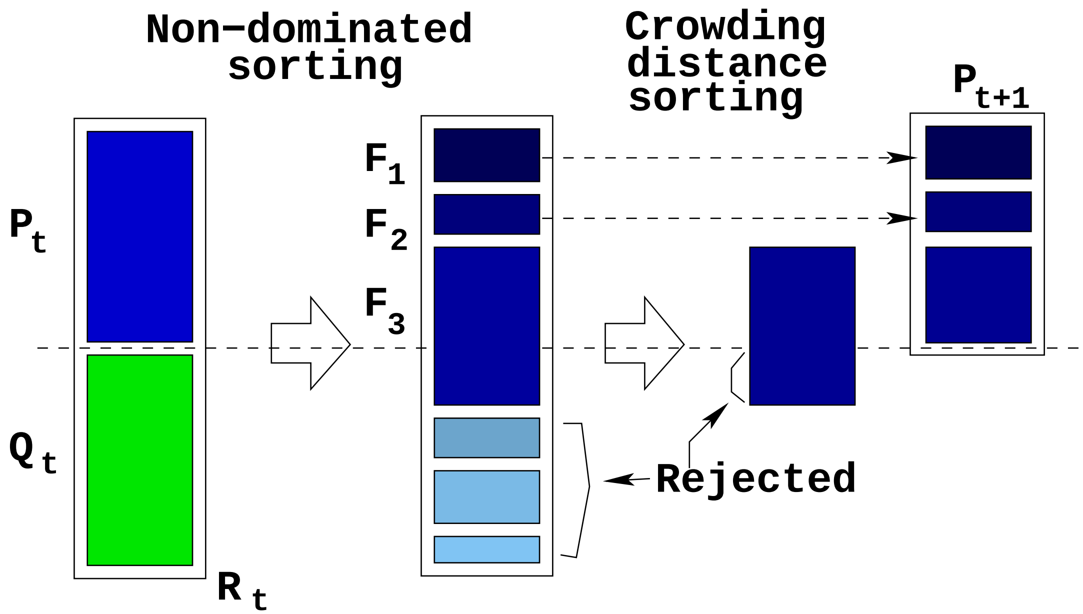
Non-dominated sorting: $O(MN^2)$

- Calculate (n_i, S_i) for each solution i
- n_i: Number of solutions dominating i
- S_i: Set of solutions dominated by i



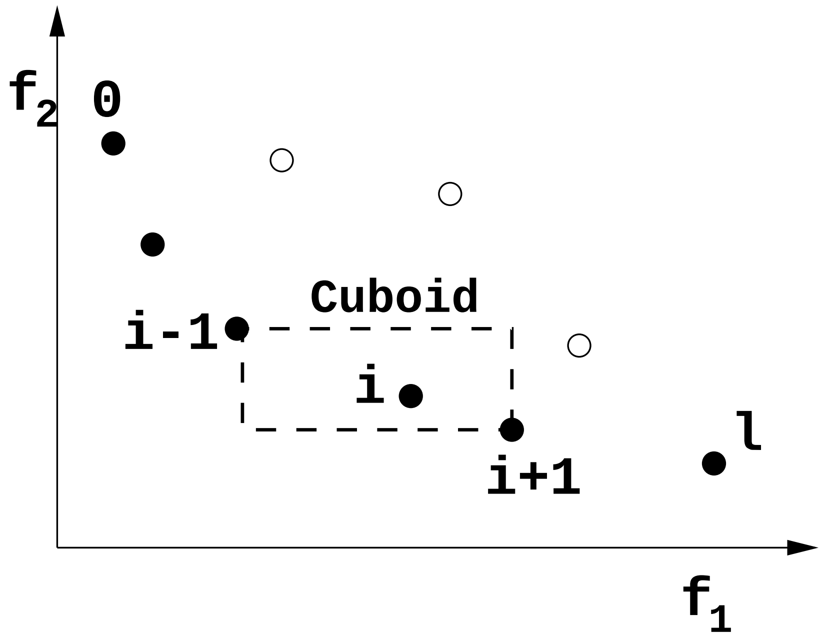
NSGA-II (cont.)

Elites are preserved



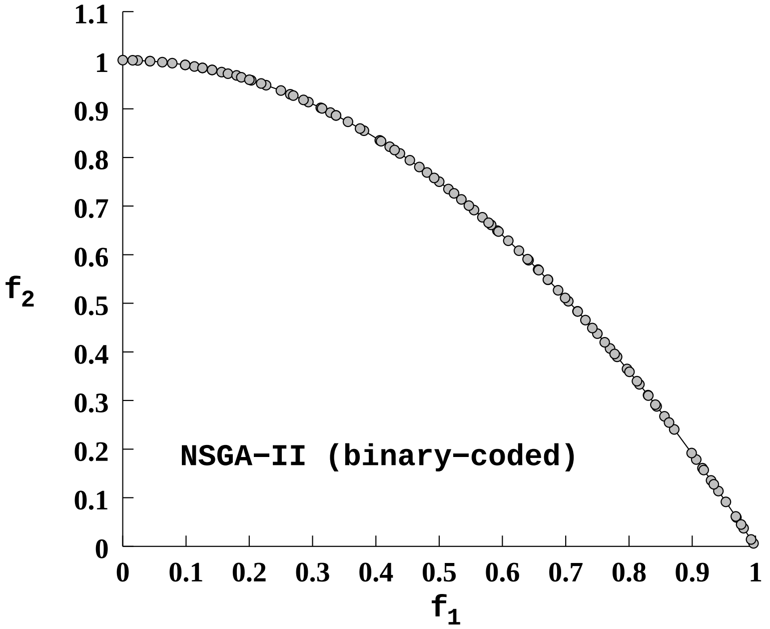
NSGA-II (cont.)

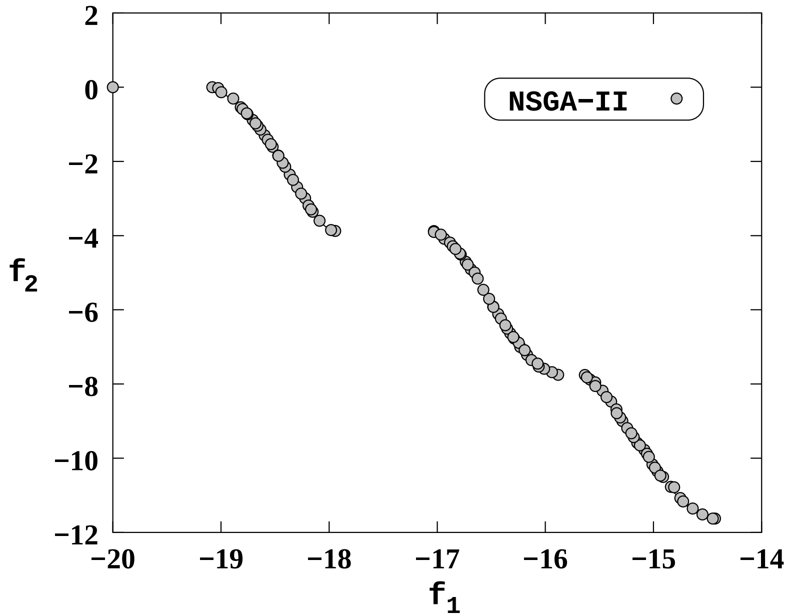
Diversity is maintained: $O(MN \log N)$



Overall Complexity: $O(MN^2)$

NSGA-II Simulation Results



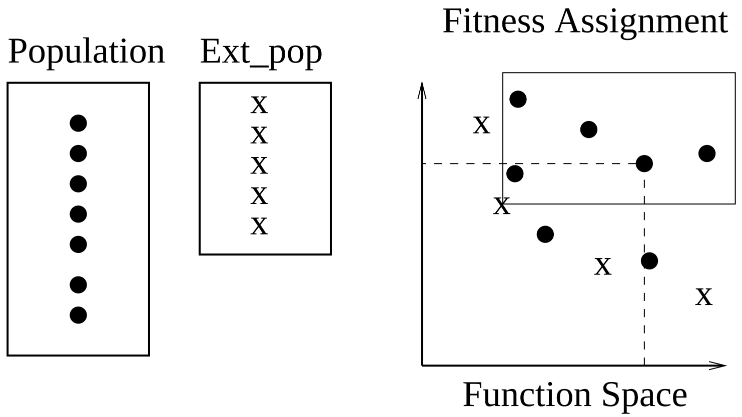


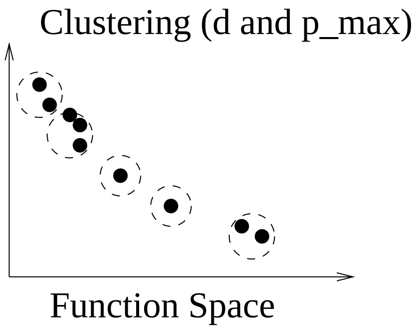
Strength Pareto EA (SPEA)

- Stores non-dominated solutions externally
- Pareto-dominance to assign fitness
  - External members: Assign number of dominated solutions in population (smaller, better)
  - Population members: Assign sum of fitness of external dominating members (smaller, better)
- Tournament selection and recombination applied to combined current and elite populations
- A clustering technique to maintain diversity in updated external population, when size increases a limit

SPEA (cont.)

- Fitness assignment and clustering methods



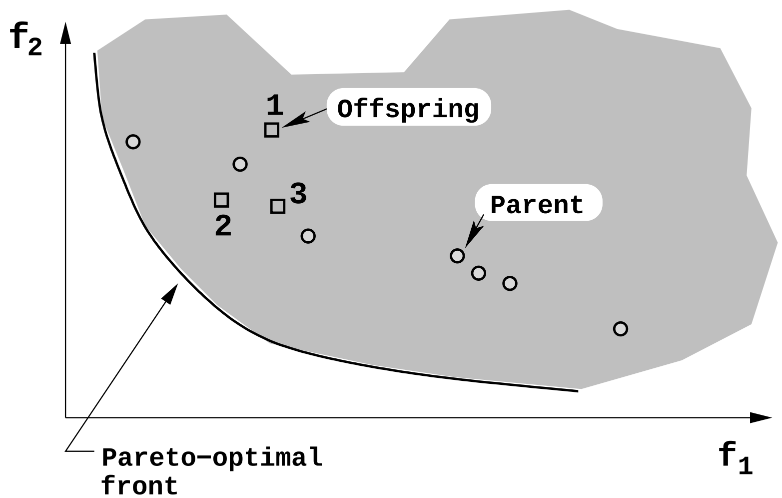


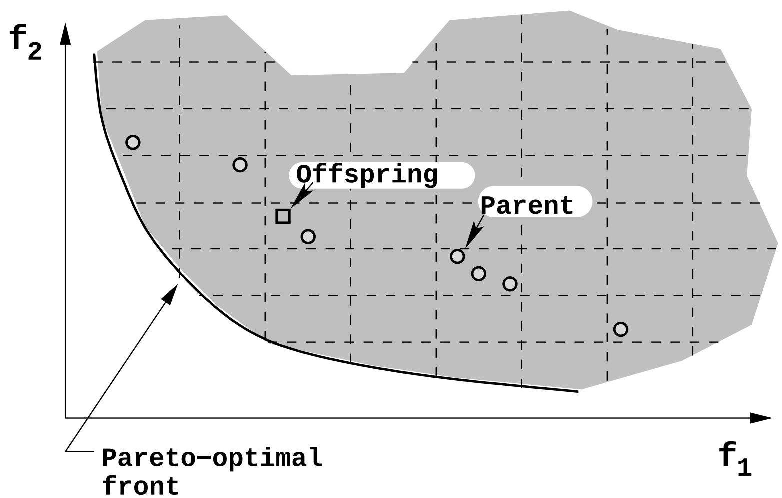
Pareto Archived ES (PAES)

- An (1+1)-ES
- Parent p_t and child c_t are compared with an external archive A_t
- If c_t is dominated by A_t, $p_{t+1} = p_t$
- If c_t dominates a member of A_t, delete it from A_t and include c_t in A_t and $p_{t+1} = c_t$
- If $|A_t| < N$, include c_t and $p_{t+1} = winner(p_t, c_t)$
- If $|A_t| = N$ and c_t does not lie in highest count hypercube H, replace c_t with a random solution from H and $p_{t+1} = winner(p_t, c_t)$.

The winner is based on *least* number of solutions in the hypercube

Niching in PAES-(1+1)





Constrained Handling

- Penalty function approach

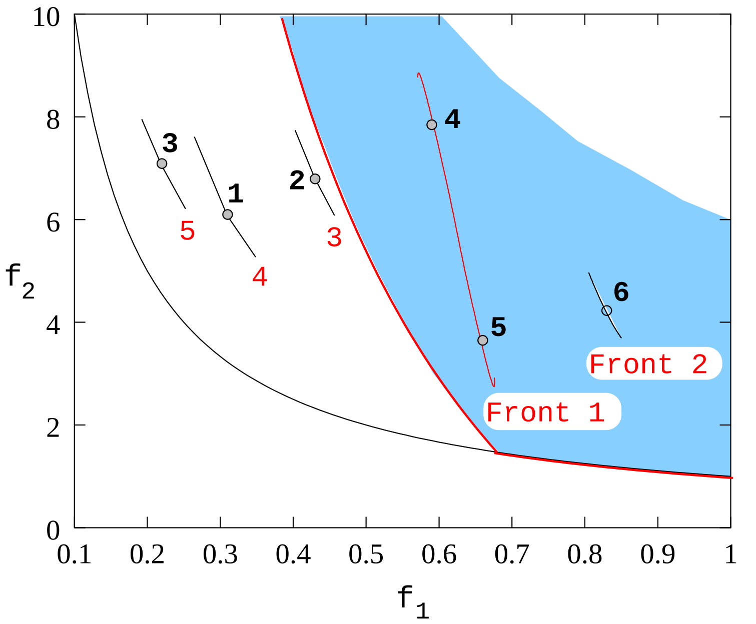
$$F_m = f_m + R_m \Omega(\vec{g}).$$

- Explicit procedures to handle infeasible solutions
  - Jimenez's approach
  - Ray-Tang-Seow's approach
- Modified definition of domination
  - Fonseca and Fleming's approach
  - Deb et al.'s approach

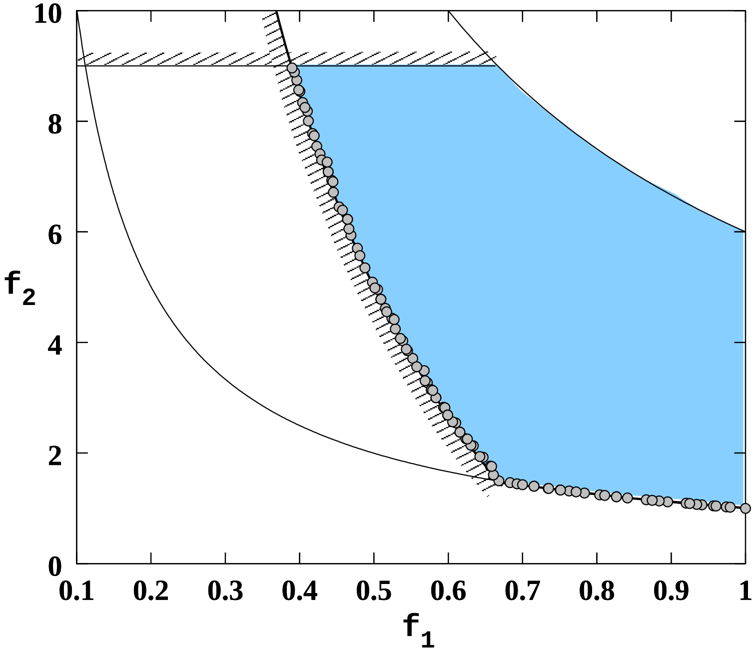
Constrain-Domination Principle

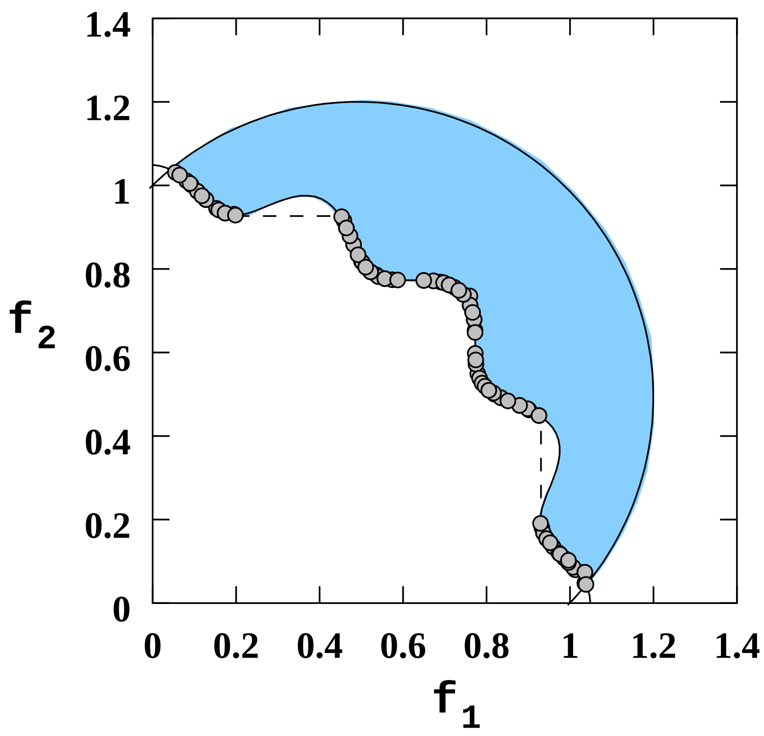
A solution i constrained-dominates a solution j, if any is true:

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



Constrained NSGA-II Simulation Results





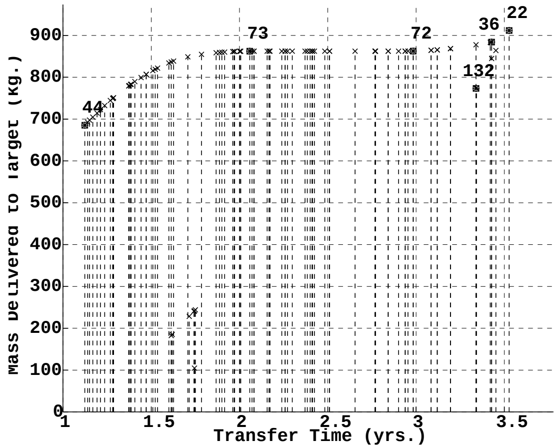
Applications of MOEAs

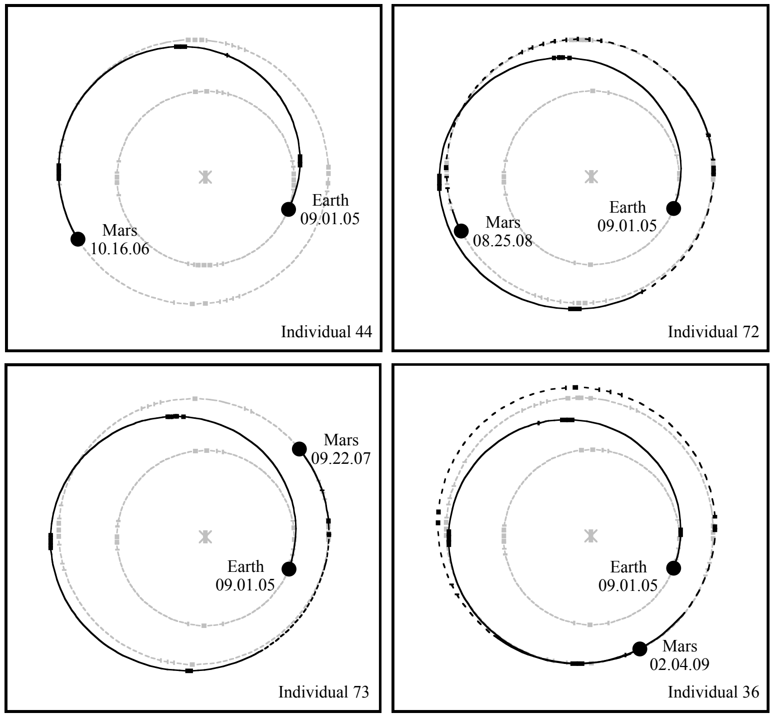
- Space-craft trajectory optimization
- Engineering component design
- Microwave absorber design
- Ground-water monitoring
- Extruder screw design
- Airline scheduling
- VLSI circuit design
- Other applications (refer Deb, 2001 and EMO-01 proceedings)

Spacecraft Trajectory Optimization

- Coverstone-Carroll et al. (2000) with JPL Pasadena
- Three objectives for inter-planetary trajectory design
  - Minimize time of flight
  - Maximize payload delivered at destination
  - Maximize heliocentric revolutions around the Sun
- NSGA invoked with SEPTOP software for evaluation

Earth–Mars Rendezvous





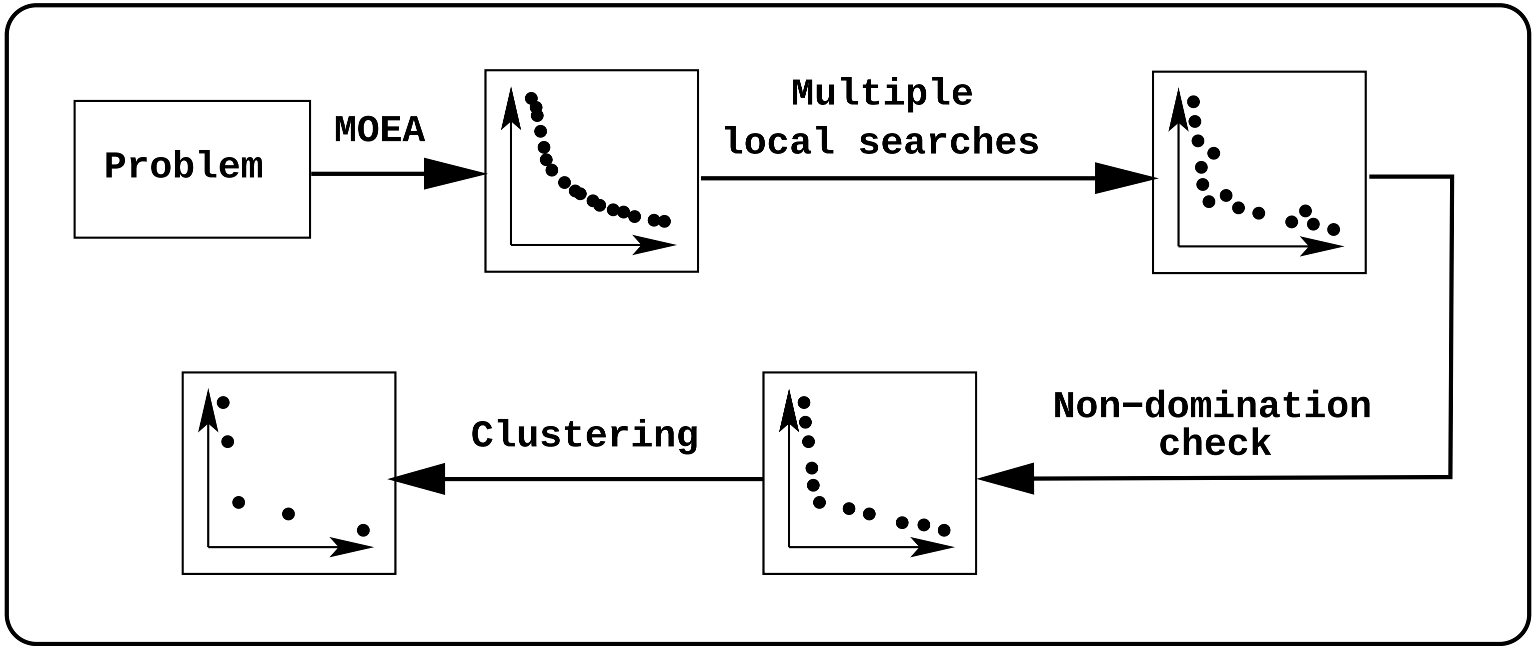
Salient Research Tasks

- Scalability of MOEAs to handle more than two objectives
- Mathematically convergent algorithms with guaranteed spread of solutions
- Test problem design
- Performance metrics and comparative studies
- Controlled elitism
- Developing practical MOEAs – Hybridization, parallelization
- Application case studies

Hybrid MOEAs

- Combine EAs with a local search method
  - Better convergence
  - Faster approach
- Two hybrid approaches
  - Local search to update each solution in an EA population (Ishubuchi and Murata, 1998; Jaskiewicz, 1998)
  - First EA and then apply a local search

Posteriori Approach in an MOEA



- Which objective to use in local search?

Proposed Local Search Method

- Weighted sum strategy (or a Tchebycheff metric)

$$F = \sum_i w_i * f_i$$

- f_i is scaled
- Weight w_i chosen based on location of i in the obtained front

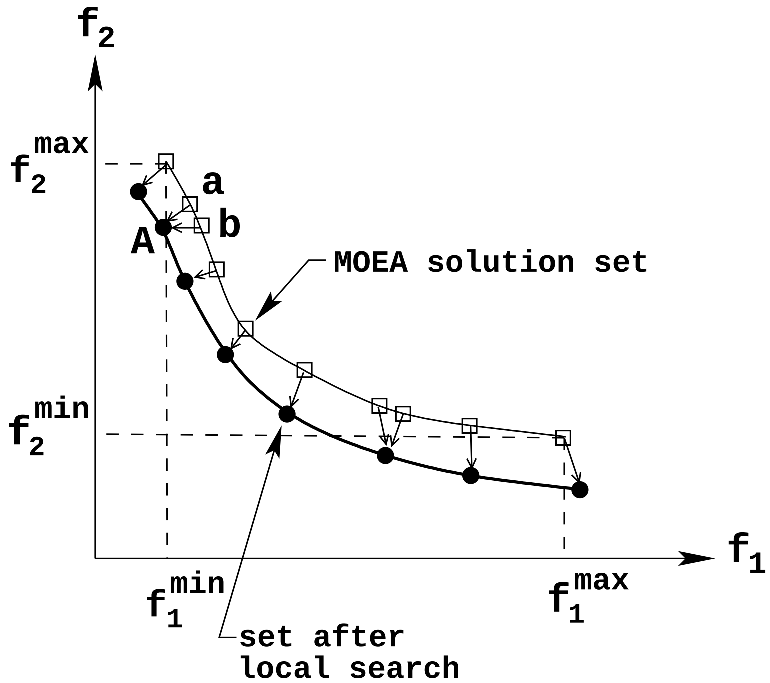
$$\bar{w}_j = \frac{(f_j^{\max} - f_j(\mathbf{x}))/(f_j^{\max} - f_j^{\min})}{\sum_{k=1}^{M}(f_k^{\max} - f_k(\mathbf{x}))/(f_k^{\max} - f_k^{\min})}$$

- Weights are normalized

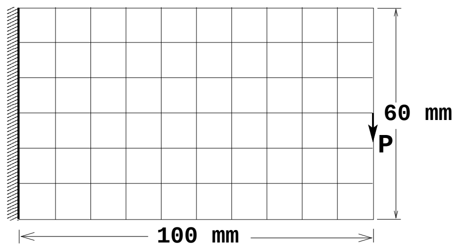
$$\sum_i w_i = 1$$

Fixed Weight Strategy

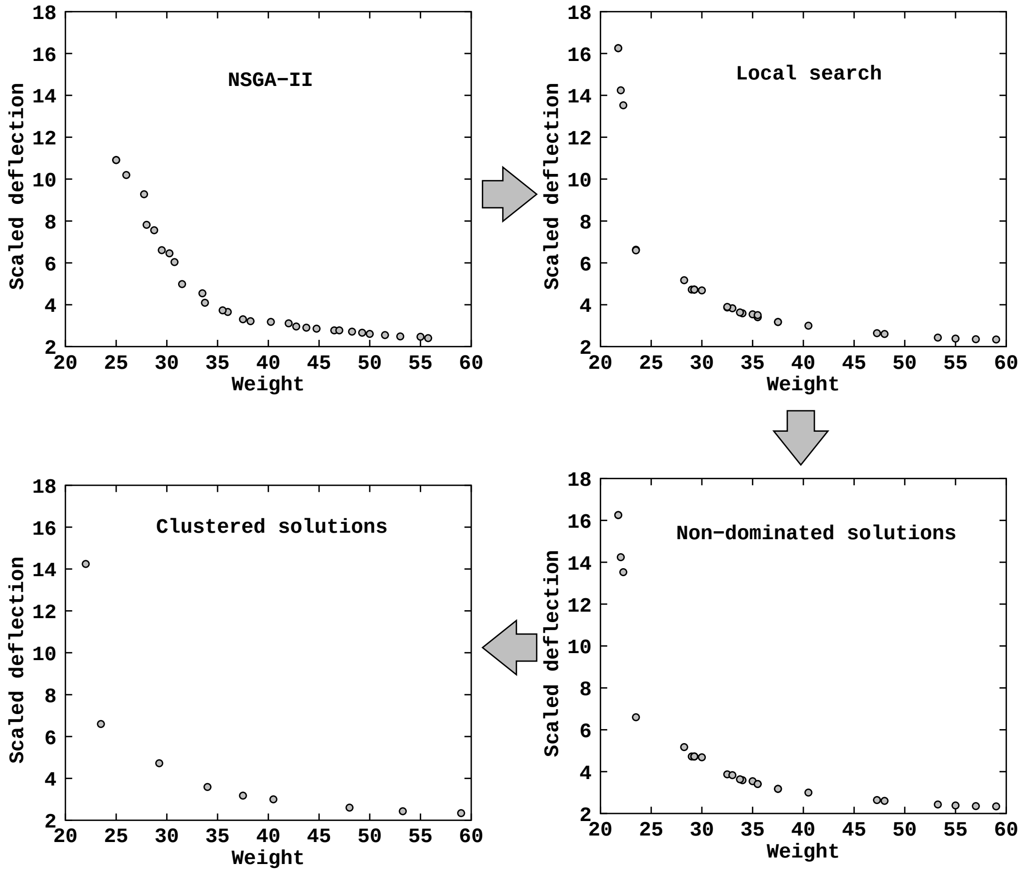
- Extreme solutions are assigned extreme weights
- Linear relation between weight and fitness
- Many solution can converge to same solution after local search



Design of a Cantilever Plate

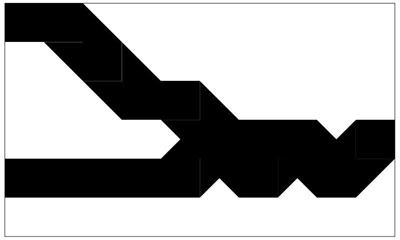


Base plate

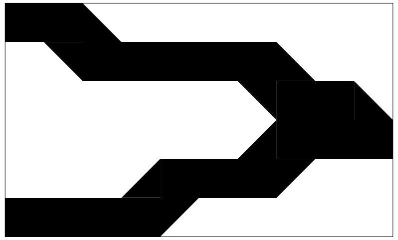


Nine trade-off solutions are chosen

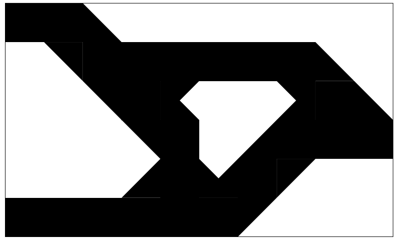
Trade-off Solutions



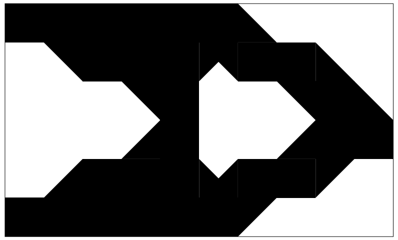
(1.00, 0.00)



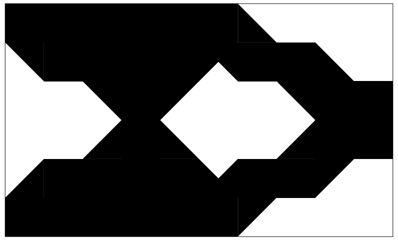
(0.60, 0.40)



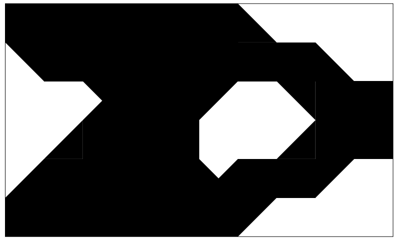
(0.50, 0.50)



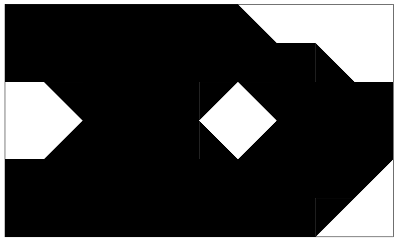
(0.43, 0.57)



(0.38, 0.62)



(0.35, 0.65)



(0.23, 0.77)

(0.14, 0.86)

(0.00, 1.00)

Conclusions

- Ideal multi-objective optimization is generic and pragmatic
- Evolutionary algorithms are ideal candidates
- Many efficient algorithms exist, more efficient ones are needed
- With some salient research studies, MOEAs will revolutionize the act of optimization
- EAs have a definite edge in multi-objective optimization and should become more useful in practice in coming years