GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3207856

Decomposition Multi-Objective Optimisation

Current Developments and Future Opportunities

Ke Li k.li@exeter.ac.uk
Department of Computer Science, University of Exeter
Qingfu Zhang qingfu.zhang@cityu.edu.hk
Department fo Computer Science, City University of Hong Kong

Outline

Part I: Basics

- Basic Concepts
- Ideas in MOEA/D
- A Simple Variant

Part II: Advanced Topics

- Current Developments
  - Decomposition methods
  - Search methods
  - Collaboration
- Resources
- Future Directions

Outline

- **Part I: Basics**
  - Basic Concepts
  - Ideas in MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
  - Resources
  - Future Directions

Multi-objective Optimisation Problem (MOP)

objective space

$$
\begin{array}{ll}
\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} & g_j(\mathbf{x}) \geq a_j, \quad j = 1, \cdots, q \\
& h_j(\mathbf{x}) = b_j, \quad j = q+1, \cdots, \ell \\
& \mathbf{x} \in \Omega
\end{array}
$$

search space

Multi-objective Optimisation Problem (MOP)

objective space

$$
\begin{array}{ll}
\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} & g_j(\mathbf{x}) \geq a_j, \quad j = 1, \cdots, q \\
 & h_j(\mathbf{x}) = b_j, \quad j = q+1, \cdots, \ell \\
 & \mathbf{x} \in \Omega
\end{array}
$$

search space

- Why multi-objective:

Multi-objective Optimisation Problem (MOP)

objective space

$$
\begin{array}{ll}
\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} & g_j(\mathbf{x}) \geq a_j, \quad j = 1, \cdots, q \\
 & h_j(\mathbf{x}) = b_j, \quad j = q+1, \cdots, \ell \\
 & \mathbf{x} \in \Omega
\end{array}
$$

search space

- Why multi-objective:
  - By nature, many real-life problems have multiple objectives.

Multi-objective Optimisation Problem (MOP)

objective space

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & g_j(\mathbf{x}) \geq a_j, \quad j = 1, \cdots, q \\ & h_j(\mathbf{x}) = b_j, \quad j = q+1, \cdots, \ell \\ & \mathbf{x} \in \Omega \end{array}$$

search space

- Why multi-objective:
  - By nature, many real-life problems have multiple objectives.
  - Decision makers (DM) or modellers don't know how to combine them into one.

Multi-objective Optimisation Problem (MOP)

objective space

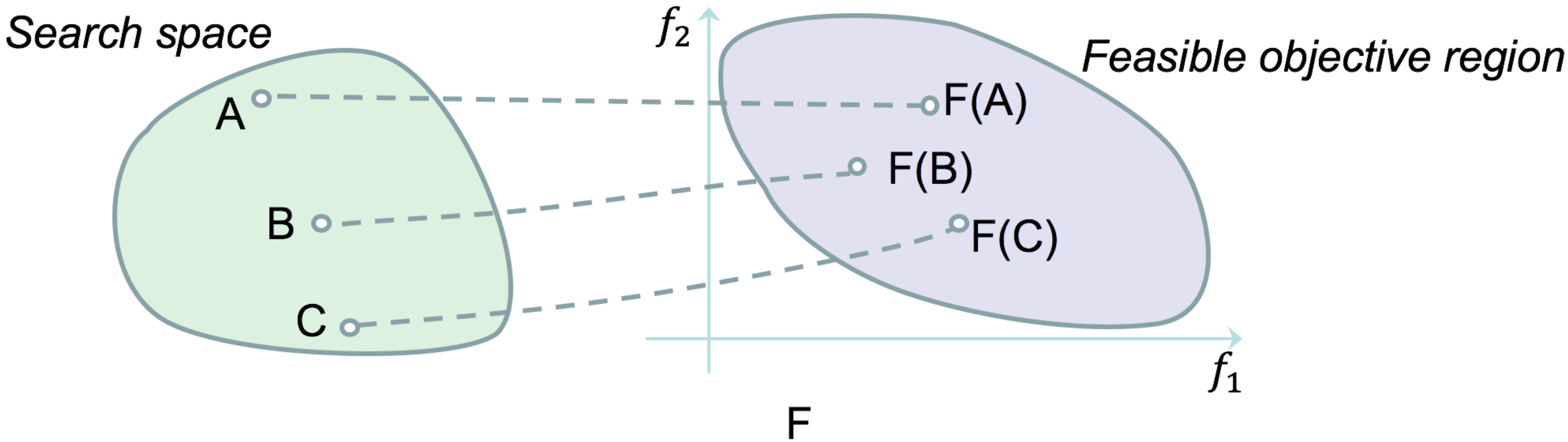
$$
\begin{array}{lll}
\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T & \\
\text{subject to} & g_j(\mathbf{x}) \geq a_j, & j = 1, \cdots, q \\
& h_j(\mathbf{x}) = b_j, & j = q+1, \cdots, \ell \\
& \mathbf{x} \in \Omega &
\end{array}
$$

search space

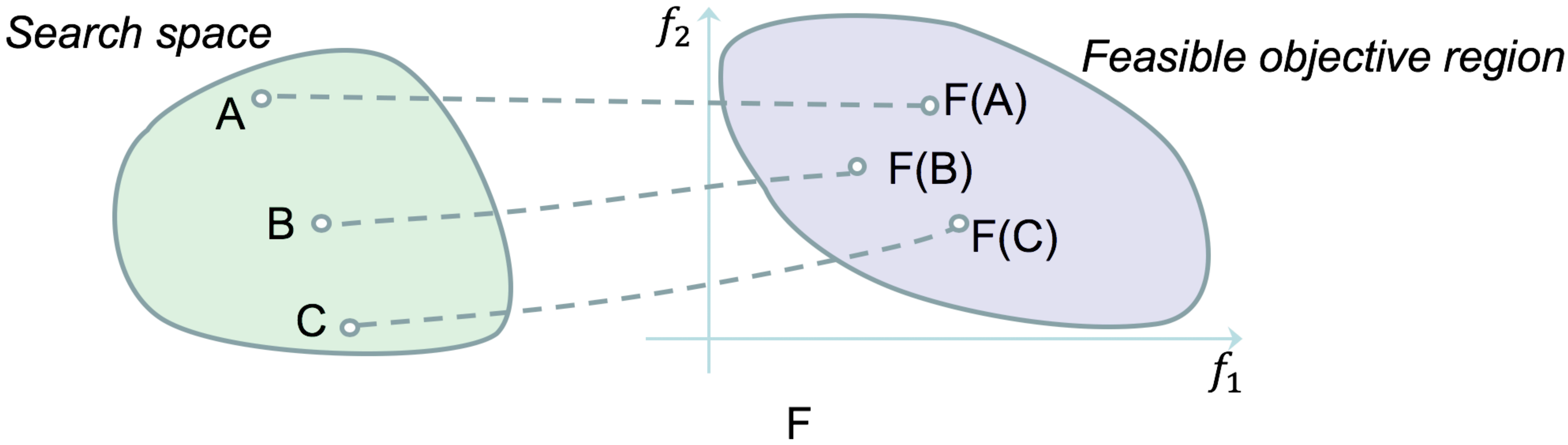
- Why multi-objective:
  - By nature, many real-life problems have multiple objectives.
  - Decision makers (DM) or modellers don’t know how to combine them into one.
  - DMs want to know trade-off relationship among these objectives.

Dominance : How to compare two solutions

Dominance : How to compare two solutions

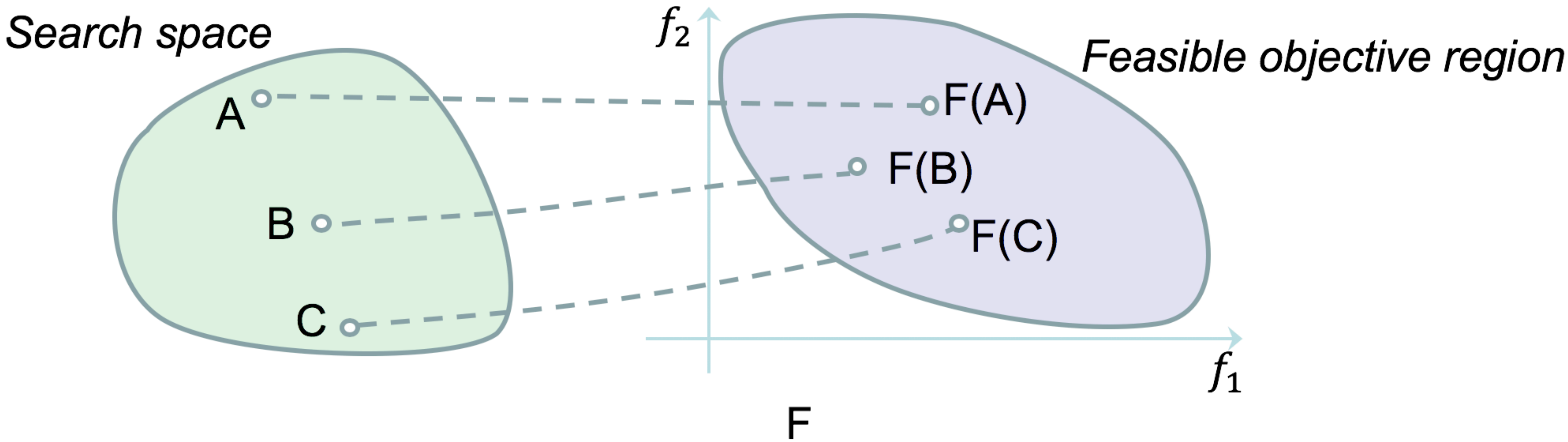


Dominance : How to compare two solutions



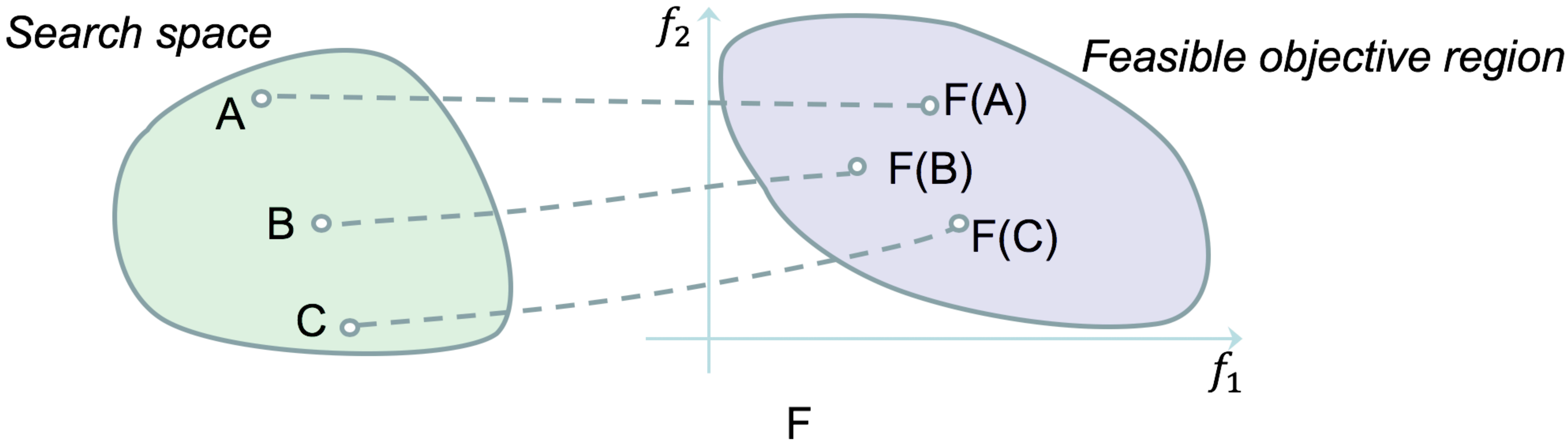
- y **dominates** x iff

Dominance : How to compare two solutions



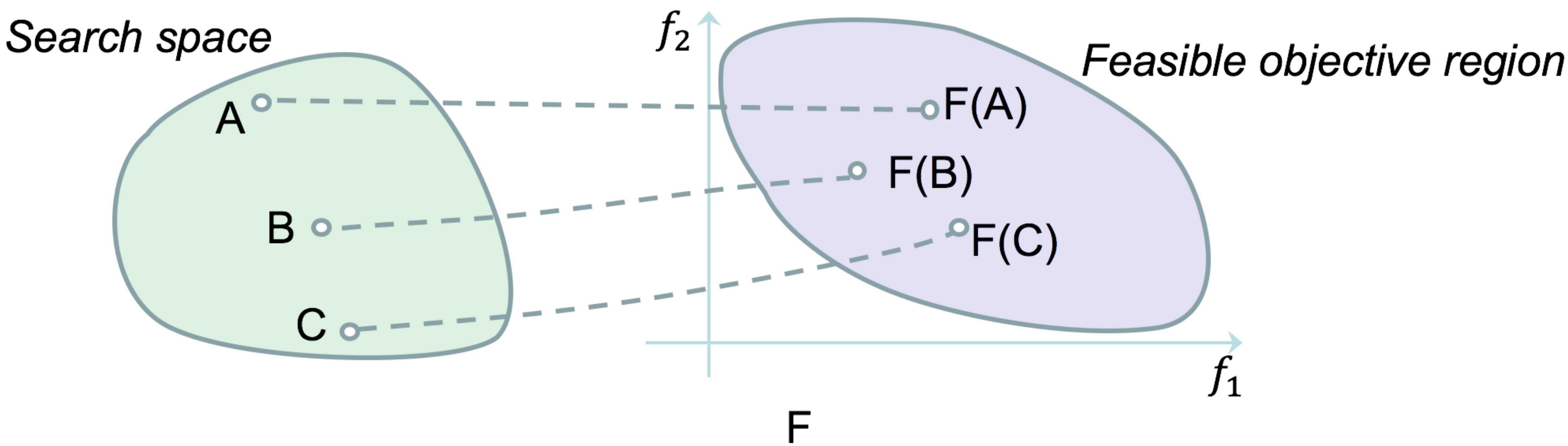
- y **dominates** x iff
  - y is no worse than x in any objs, and

Dominance : How to compare two solutions



- y **dominates** x iff
  - y is no worse than x in any objs, and
  - y is better than x in at least one obj.

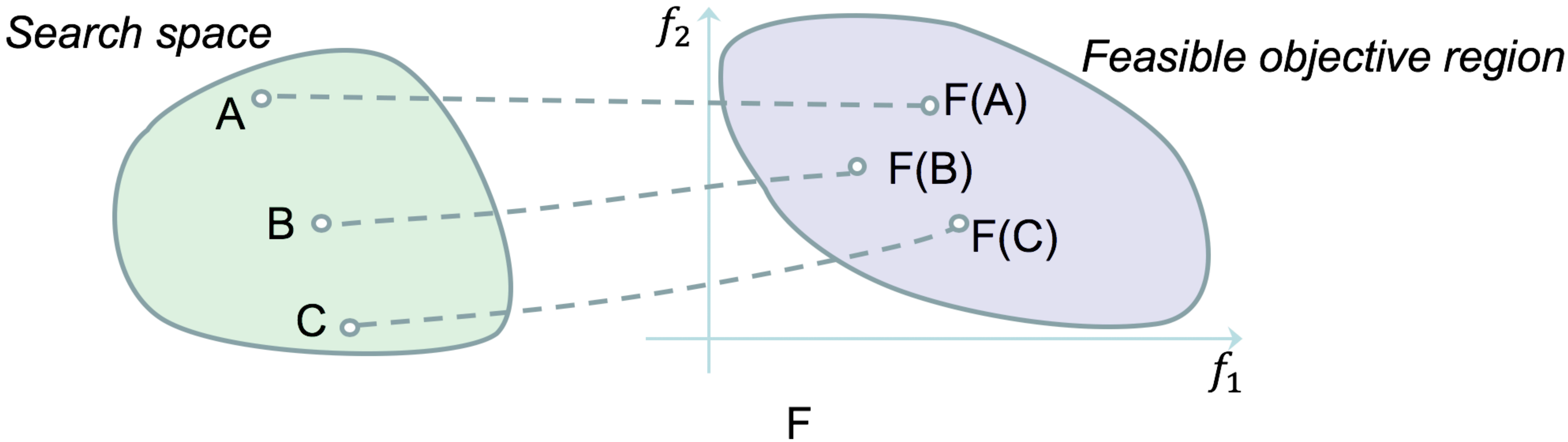
Dominance : How to compare two solutions



- y **dominates** x iff
  - y is no worse than x in any objs, and
  - y is better than x in at least one obj.

In this example:

Dominance : How to compare two solutions

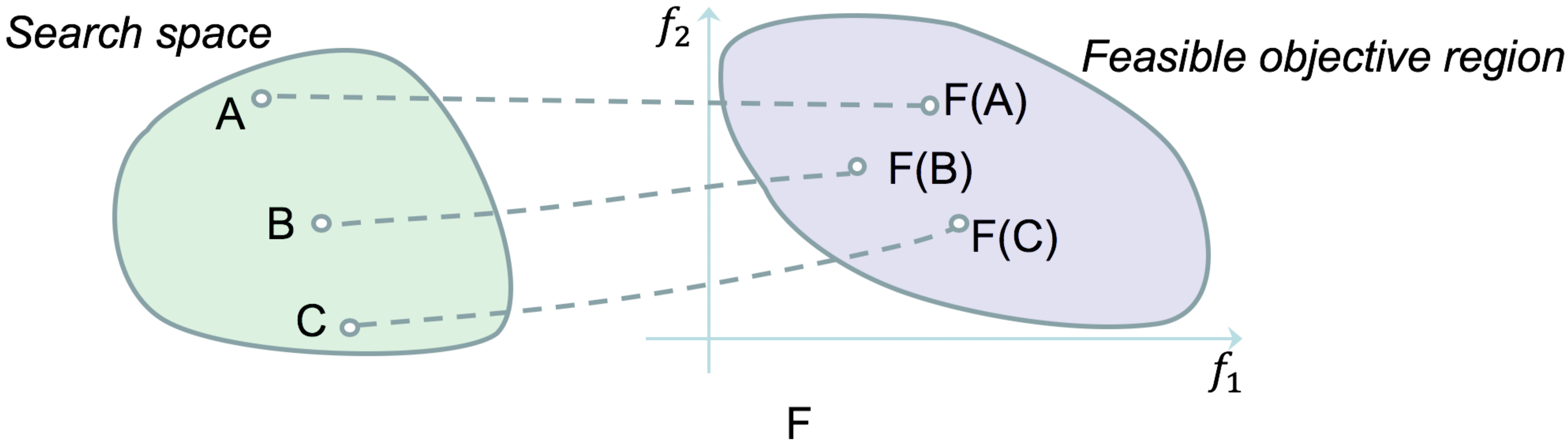


- y **dominates** x iff
  - y is no worse than x in any objs, and
  - y is better than x in at least one obj.

In this example:

B dominates A.

Dominance : How to compare two solutions



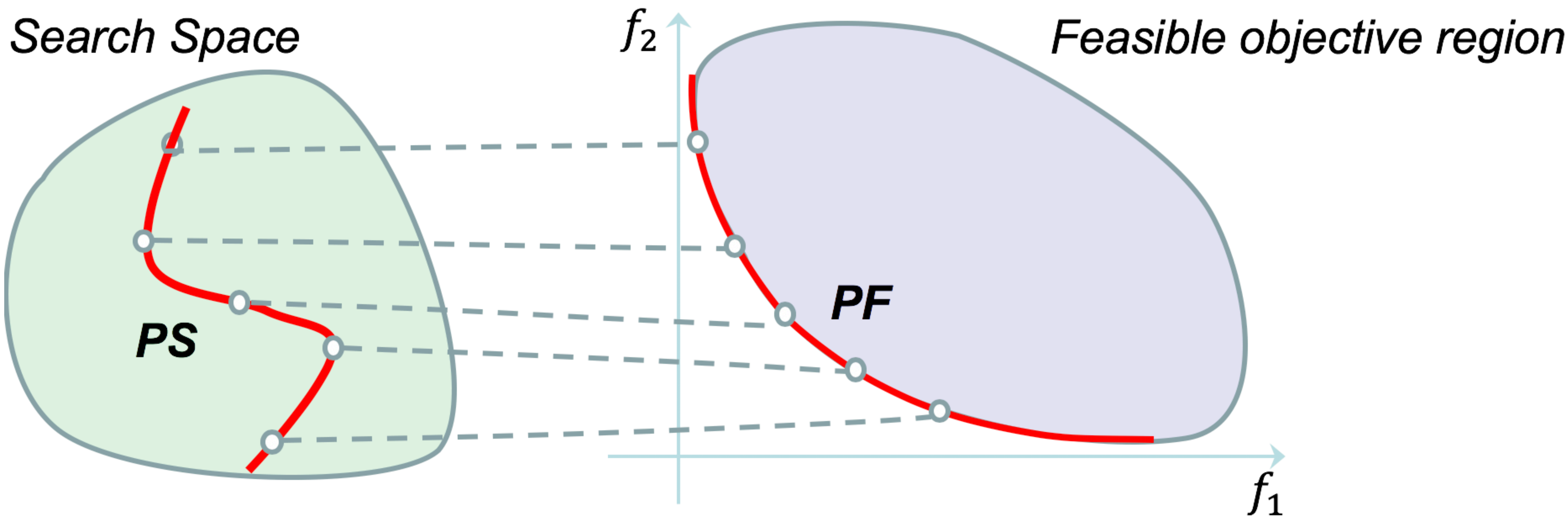
- y **dominates** x iff
  - y is no worse than x in any objs, and
  - y is better than x in at least one obj.

In this example:

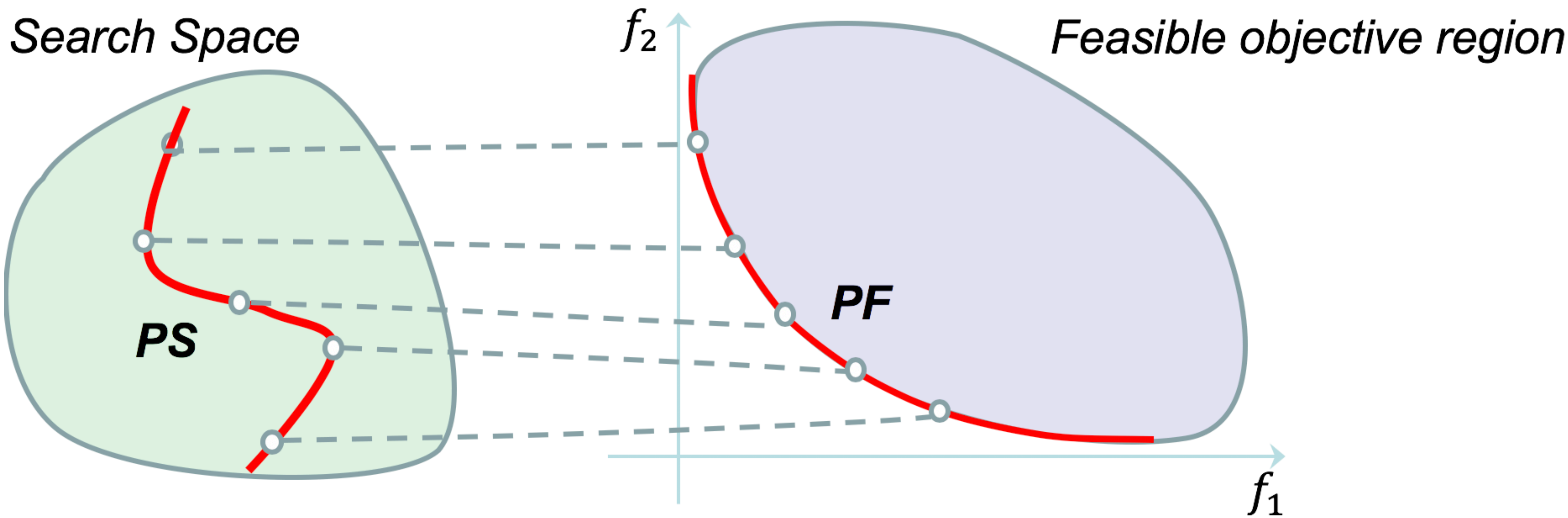
B dominates A.

B and C are not comparable

Pareto-Optimal Solutions = Best Trade-off Candidates

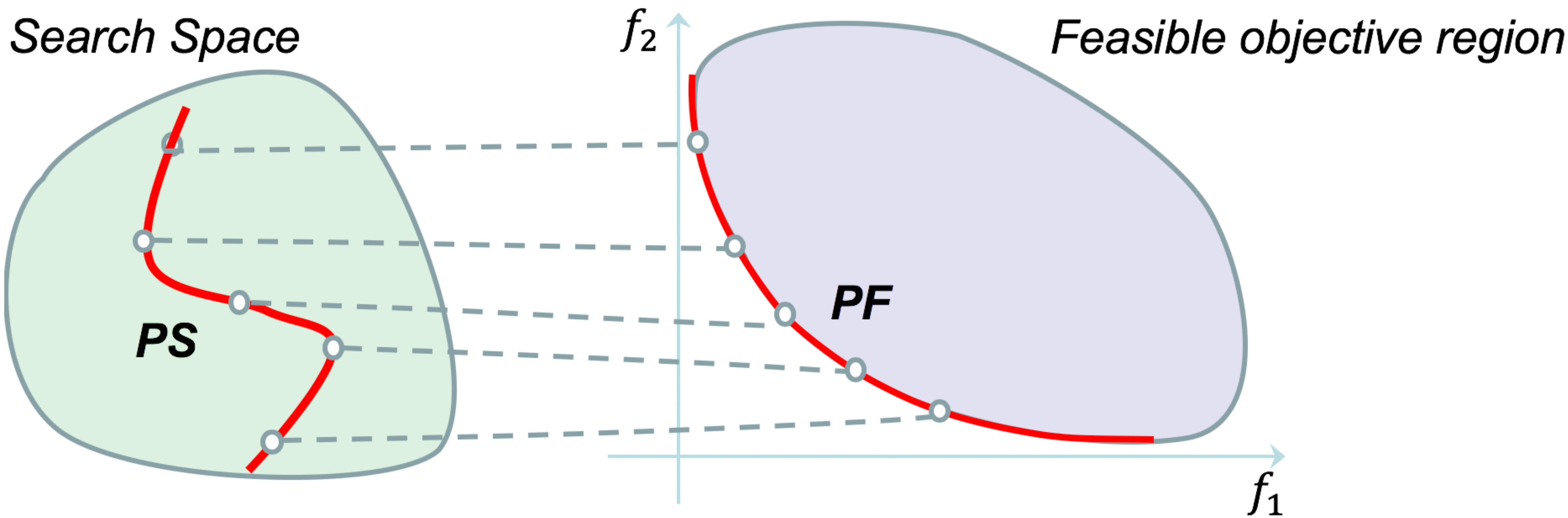


Pareto-Optimal Solutions = Best Trade-off Candidates



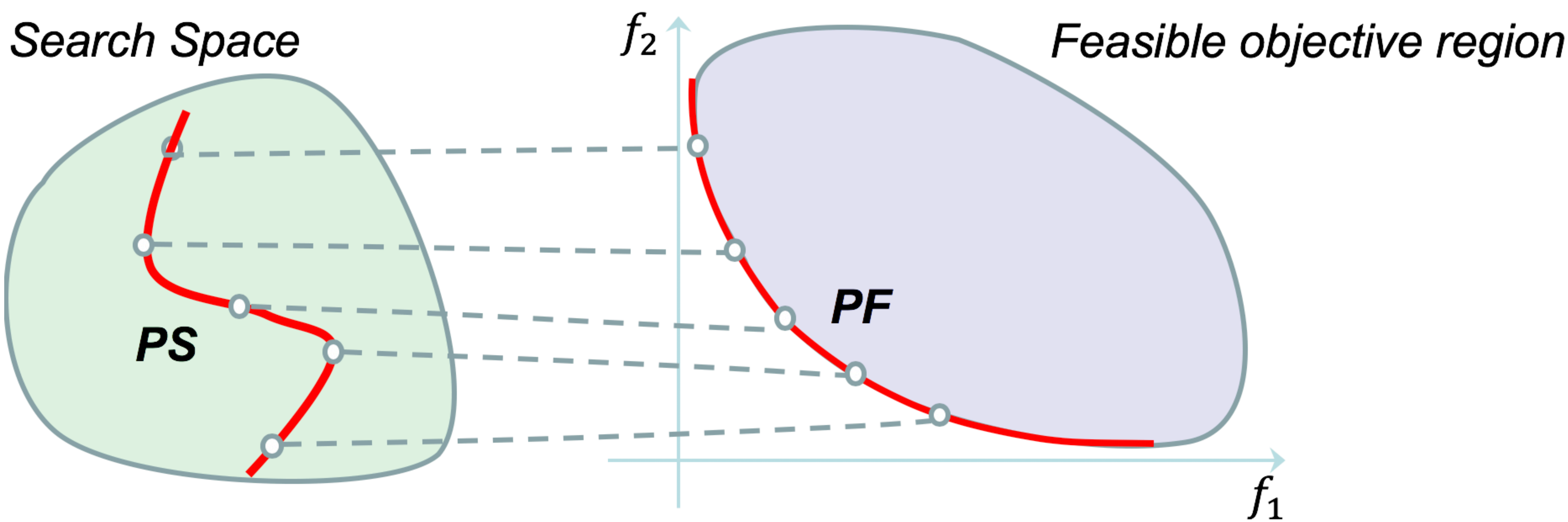
- x is Pareto optimal iff no other solution dominates it.

Pareto-Optimal Solutions = Best Trade-off Candidates



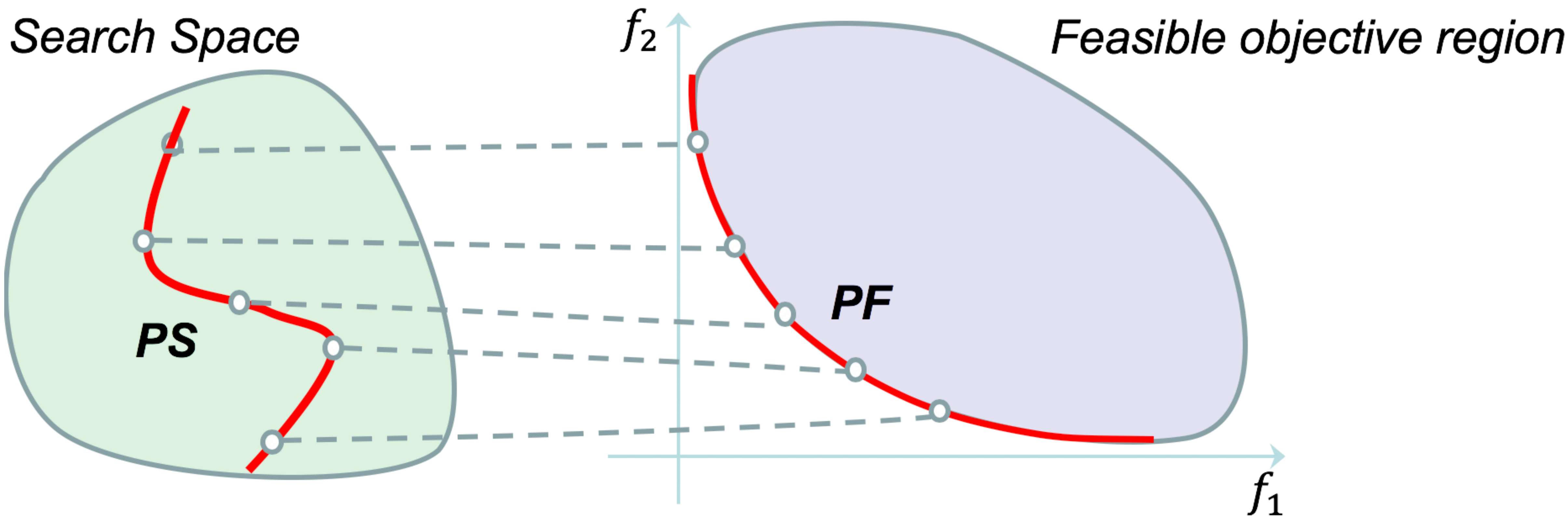
- x is Pareto optimal iff no other solution dominates it.
- A (rational) decision maker doesn't like non-Pareto optimal solutions.

Pareto-Optimal Solutions = Best Trade-off Candidates



- x is Pareto optimal iff no other solution dominates it.
- A (rational) decision maker doesn't like non-Pareto optimal solutions.
- Pareto set (PS) = the set of all Pareto optimal solutions in the x-space.

Pareto-Optimal Solutions = Best Trade-off Candidates



- x is Pareto optimal iff no other solution dominates it.
- A (rational) decision maker doesn't like non-Pareto optimal solutions.
- Pareto set (PS) = the set of all Pareto optimal solutions in the x-space.
- Pareto front (PF) = the image of the PS in the F-space.

Basic Strategies in MCDM

Basic Strategies in MCDM

- A-priori

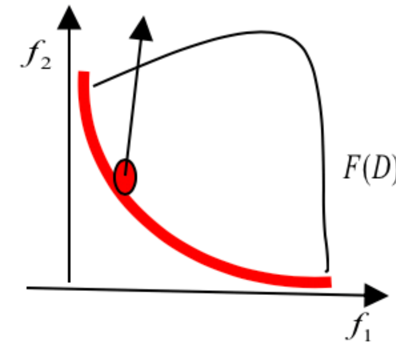
Basic Strategies in MCDM

- A-priori

→ $f(x) = 0.3 f_1(x) + 0.7 f_2(x)$ →

→

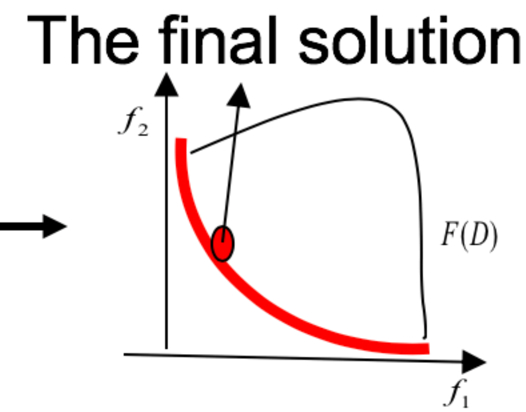
The final solution



Basic Strategies in MCDM

- A-priori

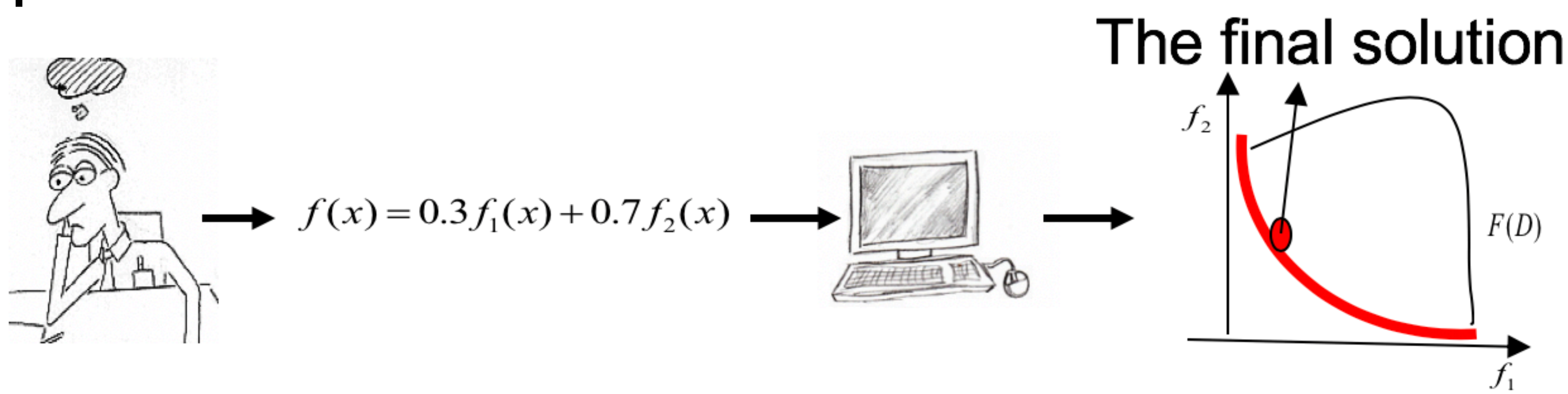
$$f(x) = 0.3 f_1(x) + 0.7 f_2(x)$$



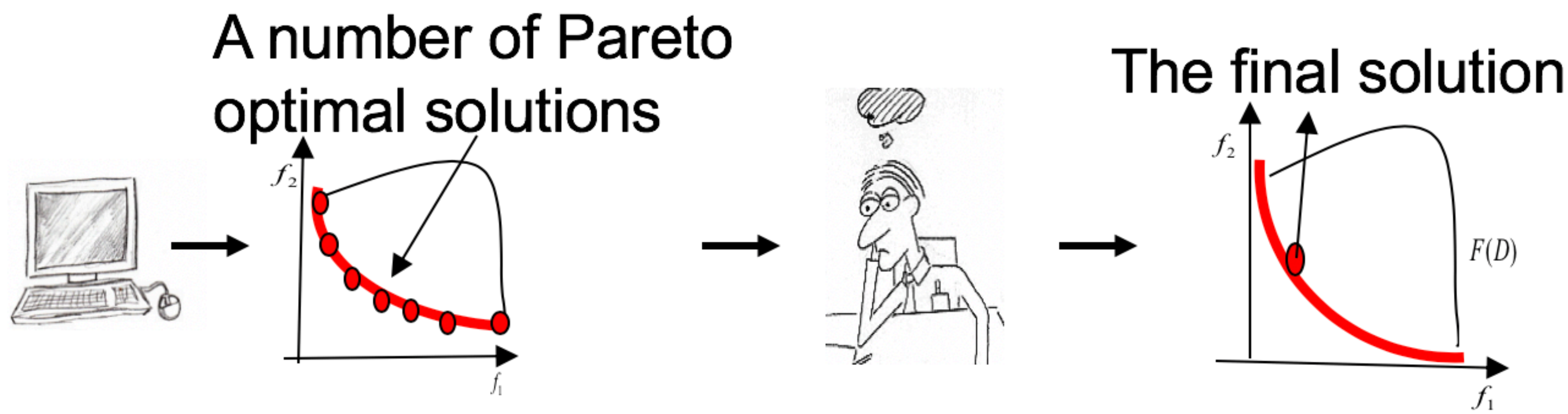
- A-posteriori

Basic Strategies in MCDM

- A-priori

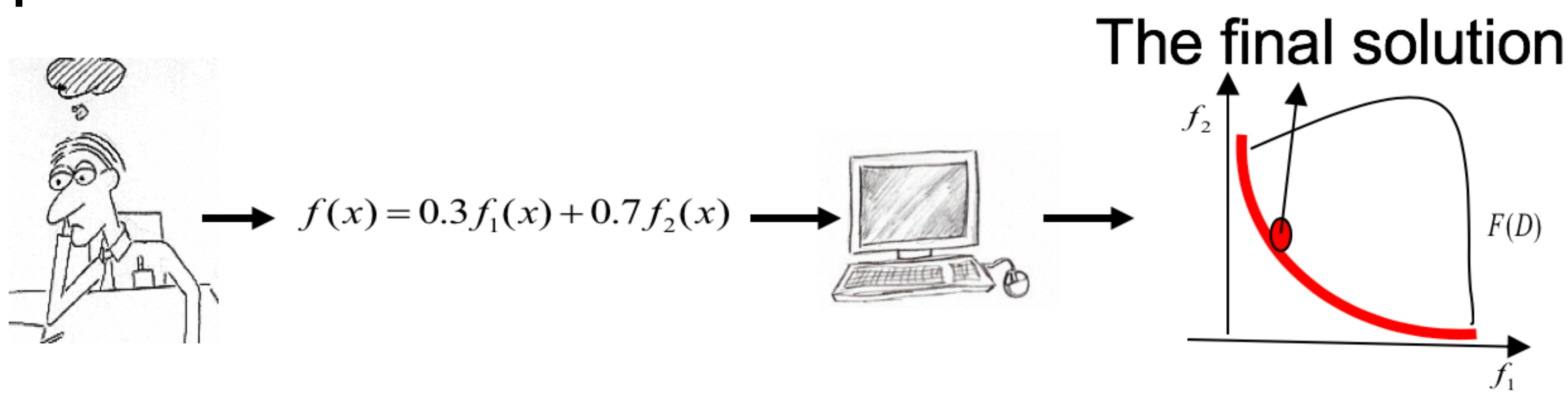


- A-posteriori

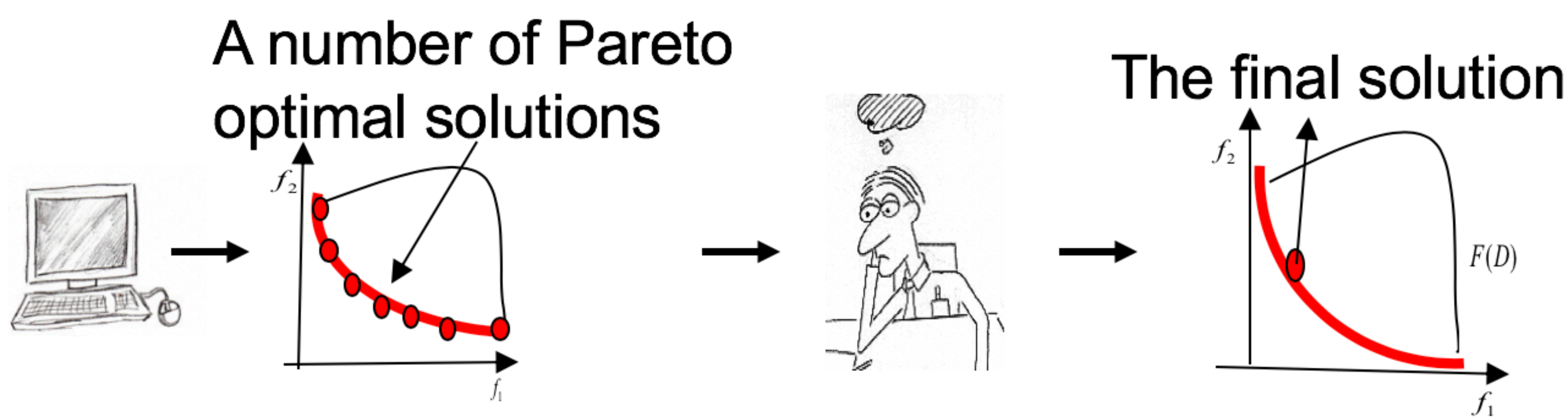


Basic Strategies in MCDM

- A-priori



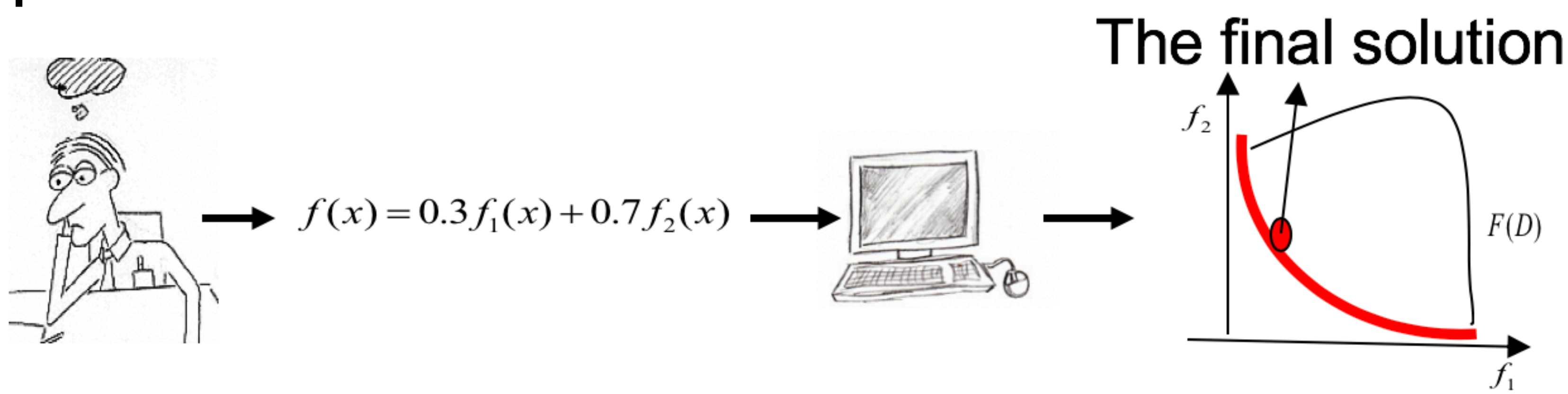
- A-posteriori



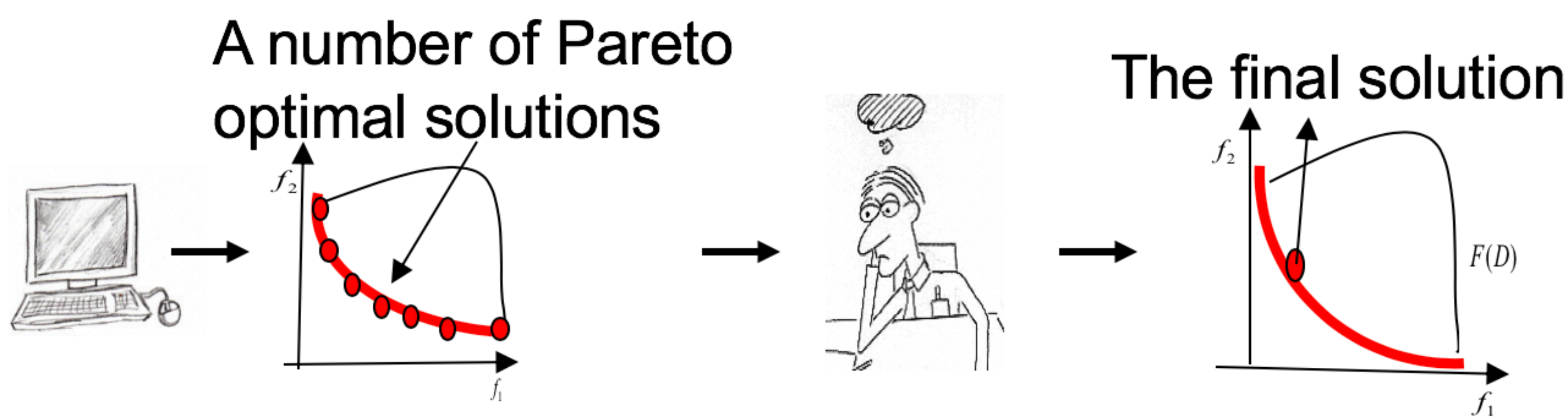
- Interactive

Basic Strategies in MCDM

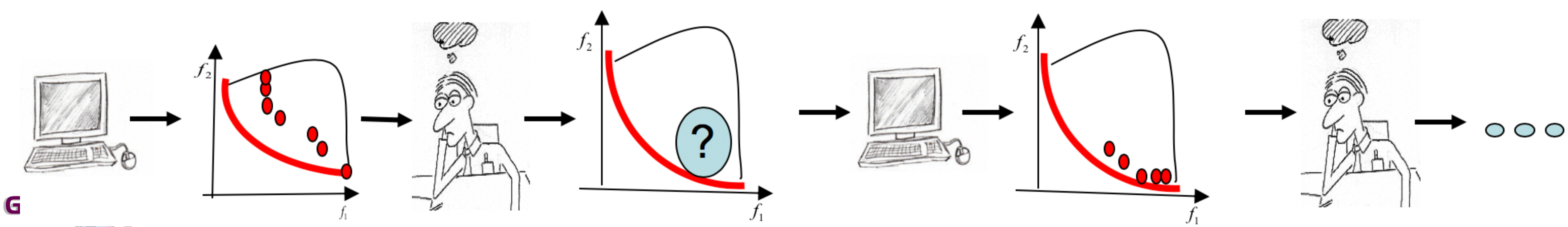
- A-priori



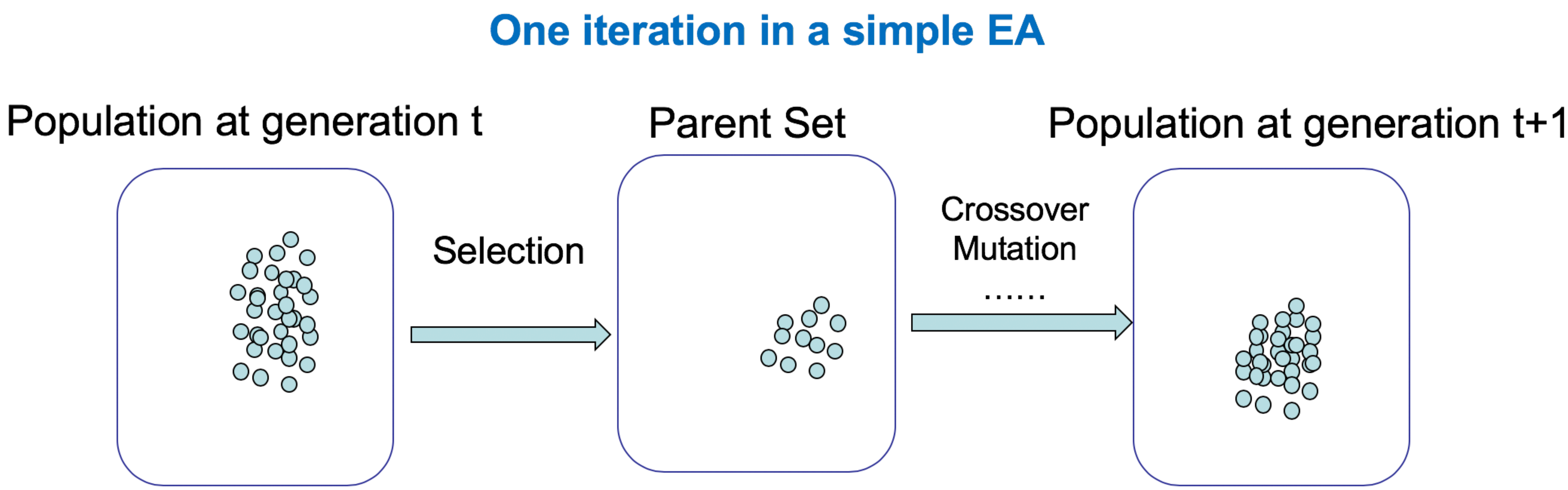
- A-posteriori



- Interactive



Evolutionary Algorithm (EA) = Population Based Iterative Search Method



- Population: a set of candidate solutions.
- Selection: select fittest solutions to be parents for next generation
- Crossover: mix two parent solutions to produce new solutions.
- Mutation: modify a parent solution to produce a new solution.

Evolutionary Multi-objective Optimisation (EMO)

- Generate a number of solutions to approximate the PF (PS) (zero-order approximation).
- Help the DM to understand her MOP and find her preferred solutions.
  - Popular and well respected in multi-criterion decision making area.
  - Hottest research area EC.

Major EMO Algorithms

- Pareto Dominance Based: *NSGA-II*, *SPEA2*, *PAES2* ….
- Performance Indicator Based: *SMS-EMOA*, *HyPE* ….
- Decomposition Based:

Outline

- **Part I: Basics**
  - Basic Concepts
  - Ideas in MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
  - Resources
  - Future Directions

Decomposition in EMO

- Decomposition has been used to some extent in EMO area for many years.

- Examples includes:
  - MOGLS (Ishibuchi, et al, 1998, Jaszkiewicz, 2002)
  - MOSA[Ulungu ,et al, 1999]
  - MOTS [Hansen, 1997].
  - 2PLS [Paquete & Stutzle, 2003]
  - AWA (Jin et al, 2001)
  - MSOPS (Hughes, 2003)
  - MOTGA (Alves, 2007)
  - CMOGA (Murata, et al, 2001)
  - ….

Decomposition in EMO

- Decomposition has been used to some extent in EMO area for many years.

- Examples includes:
  - MOGLS (Ishibuchi, et al, 1998, Jaszkiewicz, 2002)
  - MOSA[Ulungu ,et al, 1999]
  - MOTS [Hansen, 1997].
  - 2PLS [Paquete & Stutzle, 2003]
  - AWA (Jin et al, 2001)
  - MSOPS (Hughes, 2003)
  - MOTGA (Alves, 2007)
  - CMOGA (Murata, et al, 2001) → MOEA/D
  - ….

Decomposition in EMO

- Decomposition has been used to some extent in EMO area for many years.

- Examples includes:
  - MOGLS (Ishibuchi, et al, 1998, Jaszkiewicz, 2002)
  - MOSA[Ulungu ,et al, 1999]
  - MOTS [Hansen, 1997].
  - 2PLS [Paquete & Stutzle, 2003]
  - AWA (Jin et al, 2001)
  - MSOPS (Hughes, 2003)
  - MOTGA (Alves, 2007)
  - CMOGA (Murata, et al, 2001) → MOEA/D
  - ….

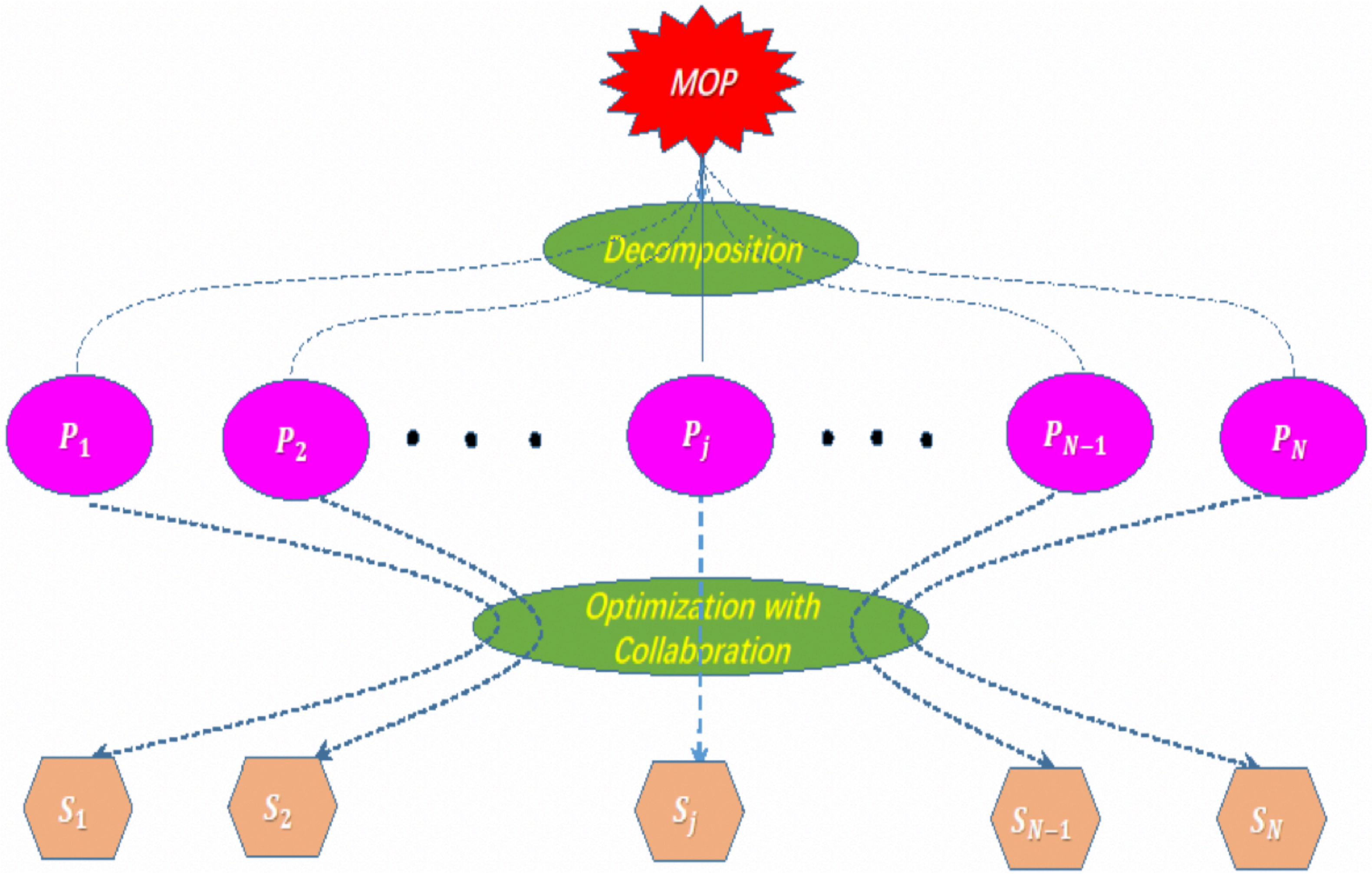
These algorithms use traditional aggregation approaches.

MOEA/D =Decomposition + Collaboration

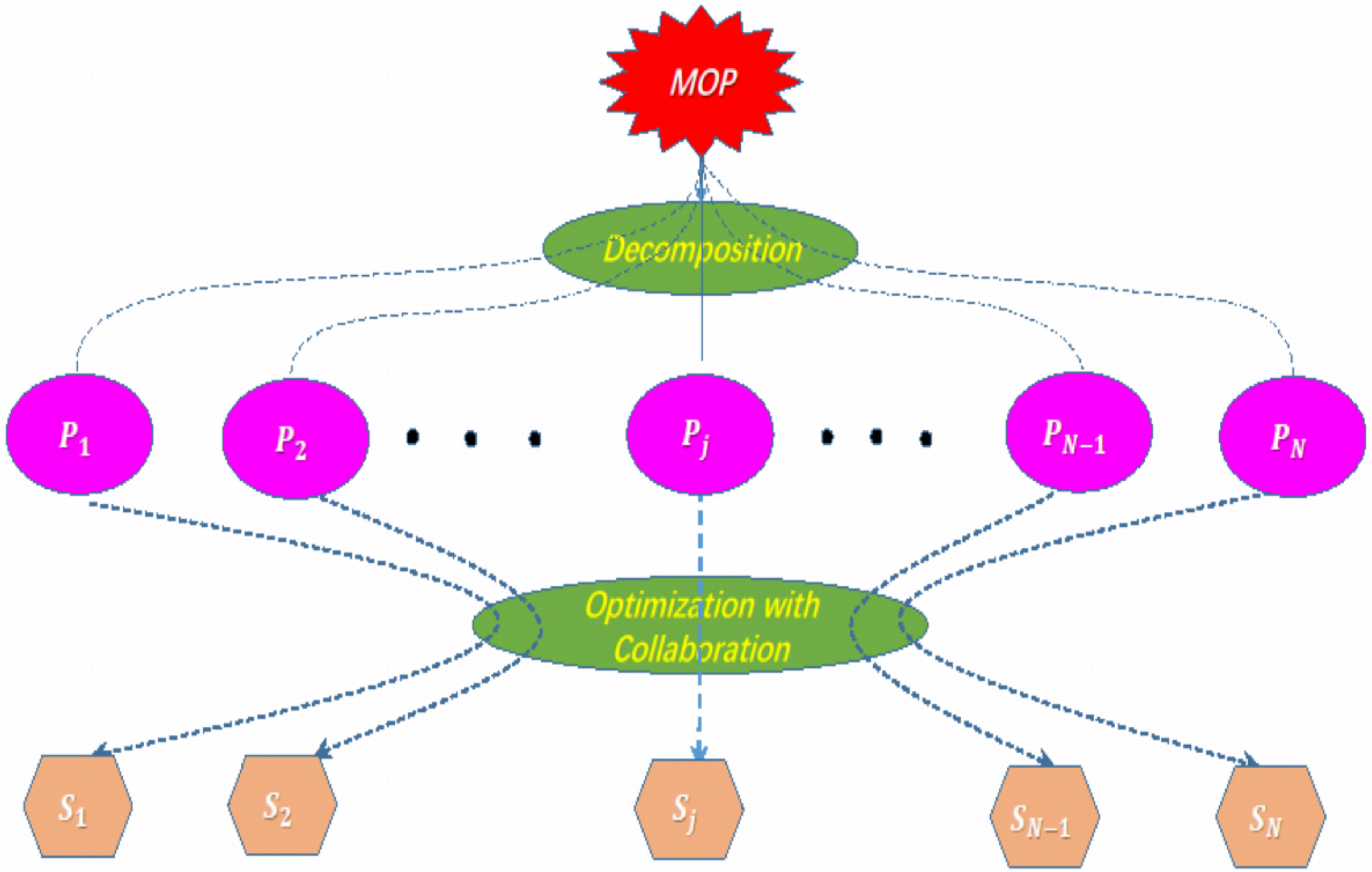
- Decomposition (from traditional optimisation)
  - Decompose the task of approximating the PF into *N* subtasks. Each subproblem can be single objective or multi-objective.

- Collaboration (from EC)
  - *N* agents (procedures) are used. Each agent is for a different subproblem.
  - These *N* subproblems are related to one another. *N* agents can solve these subproblems in a collaborative manner.

Major Issues

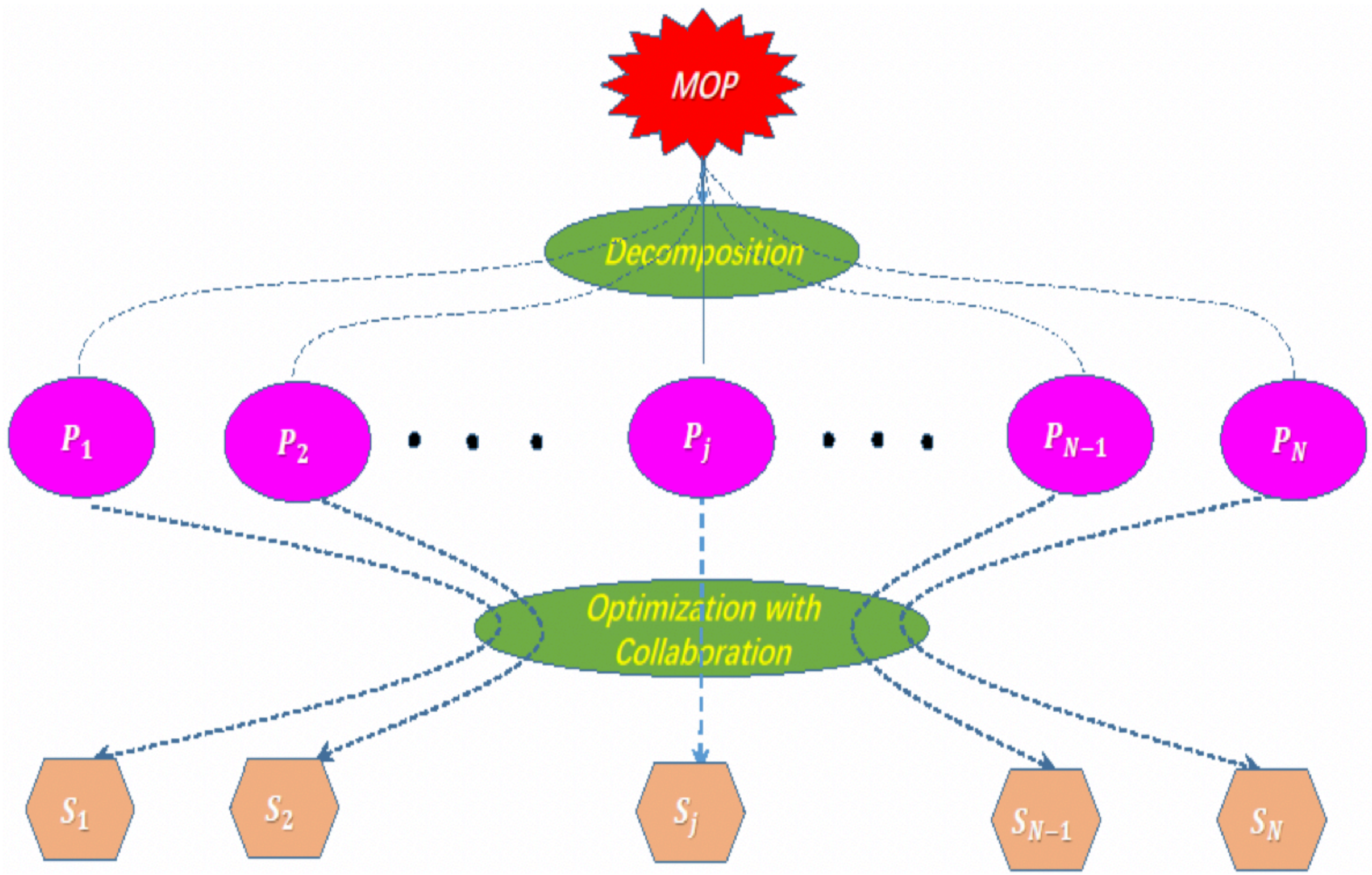


Major Issues



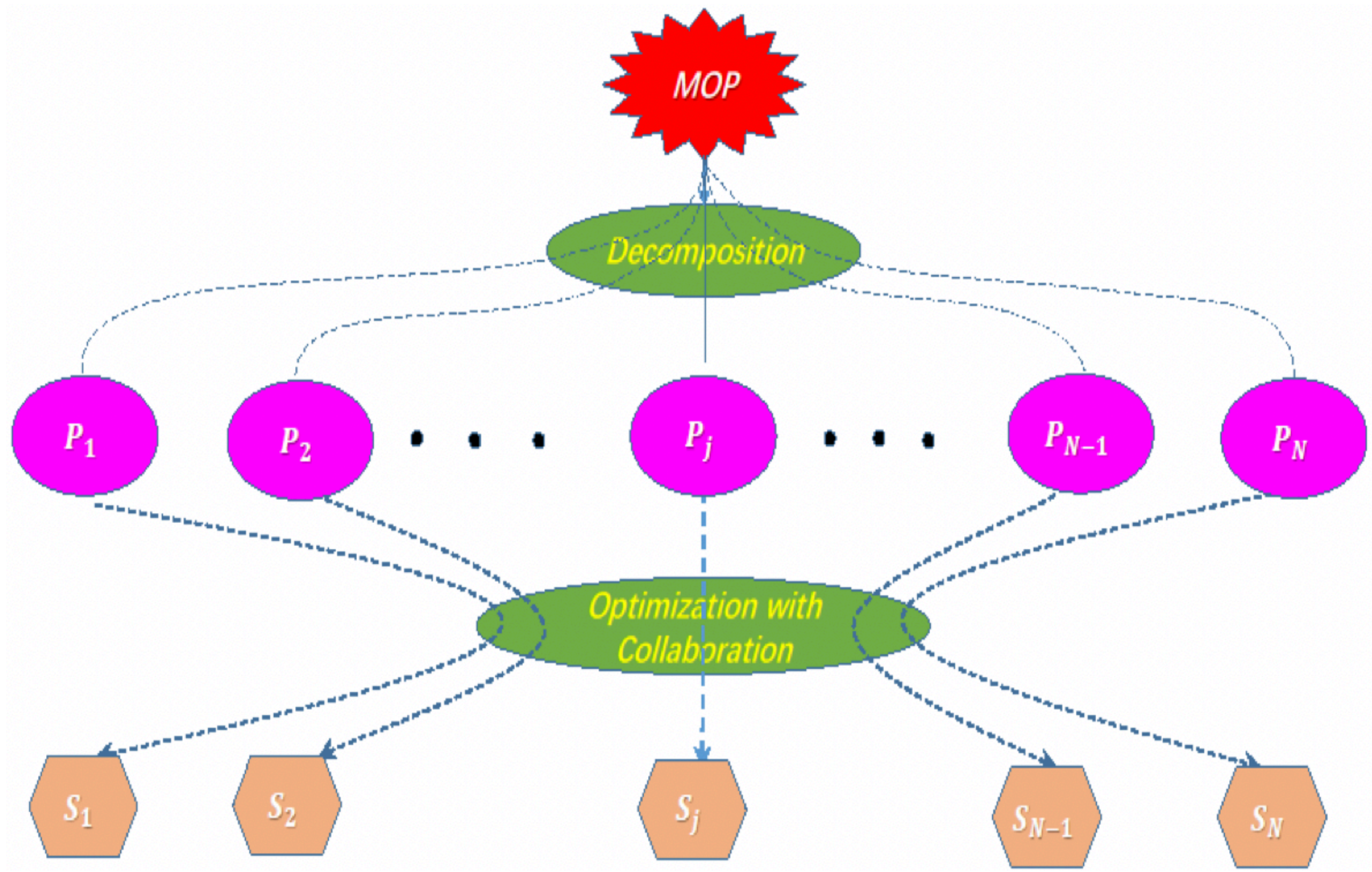
- Problem decomposition and neighbourhood structure.

Major Issues



- Problem decomposition and neighbourhood structure.
- Search method used by each agent: memory and reproduction operators.

Major Issues



- Problem decomposition and neighbourhood structure.
- Search method used by each agent: memory and reproduction operators.
- Collaboration: mating selection and replacement.

Outline

- **Part I: Basics**
  - Basic Concepts
  - Ideas in MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
  - Resources
  - Future Directions

Problem Decomposition

- Weighted Sum Approach

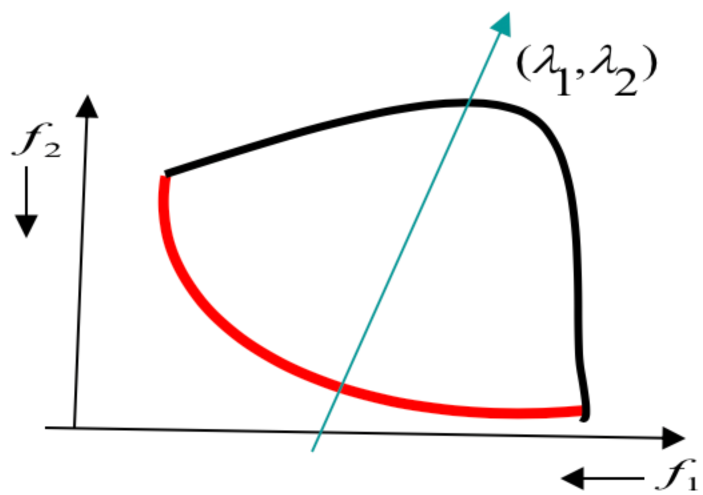
Problem Decomposition

- Weighted Sum Approach

$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

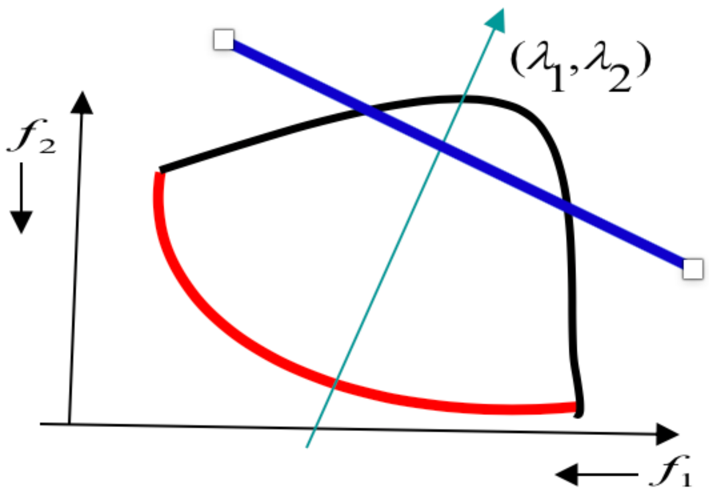
- Weighted Sum Approach



$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

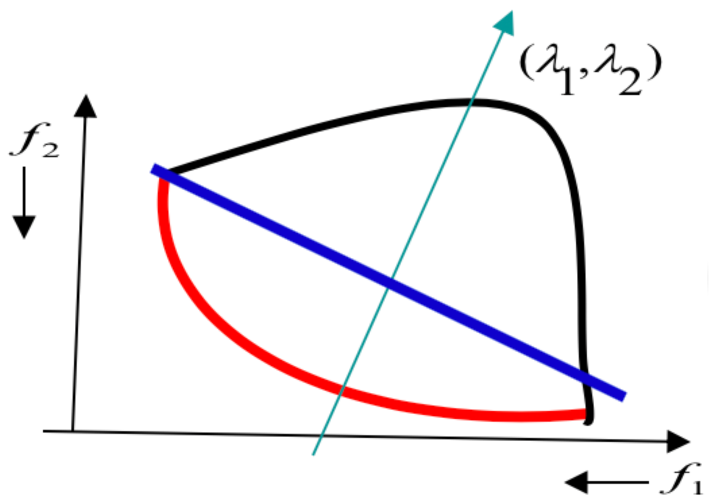
- Weighted Sum Approach



$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

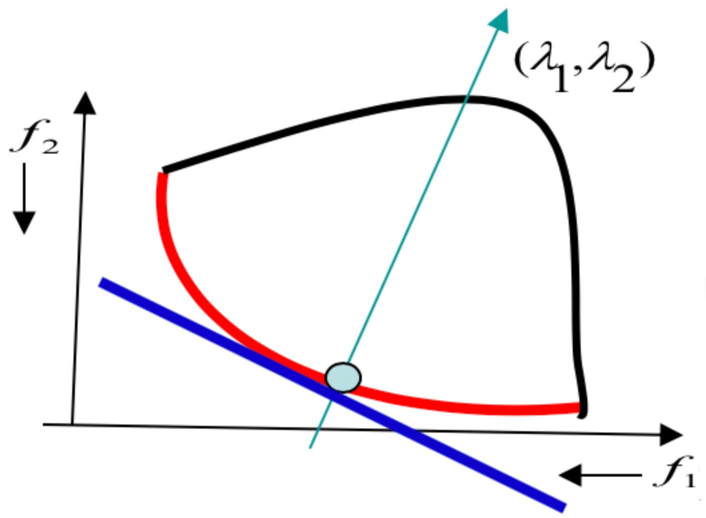
- Weighted Sum Approach



$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

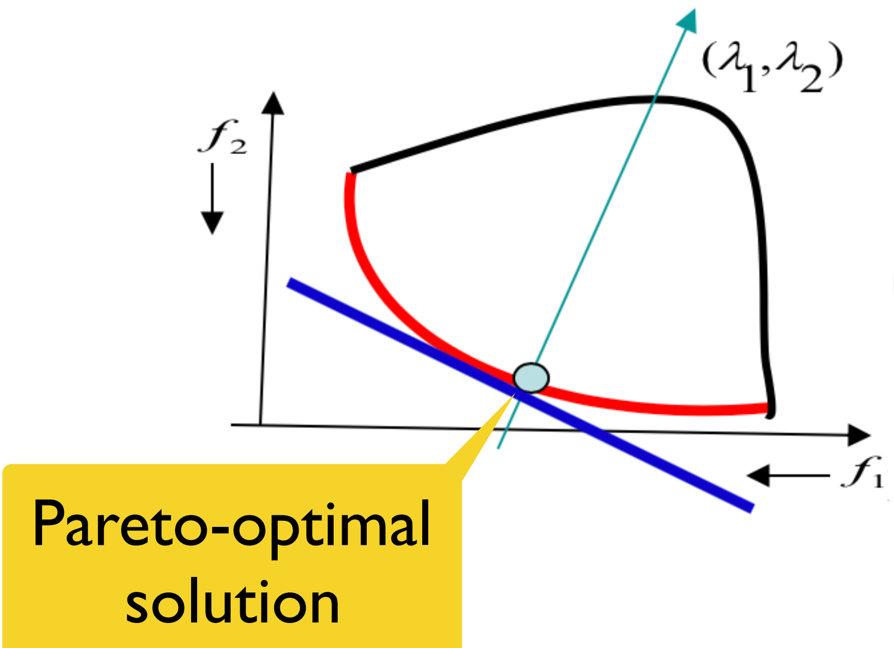
- Weighted Sum Approach



$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

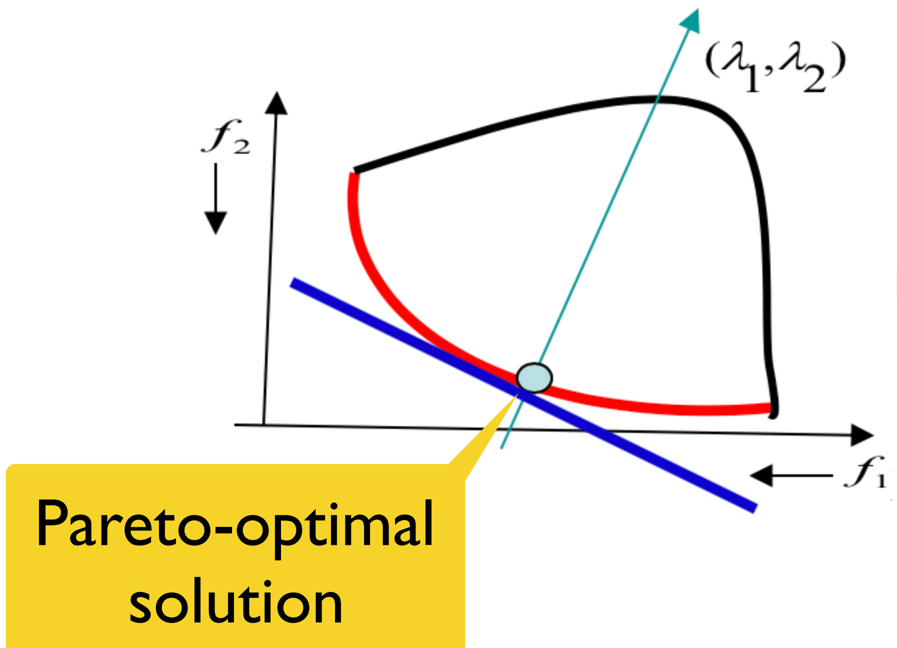
- Weighted Sum Approach



$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

Problem Decomposition

- Weighted Sum Approach

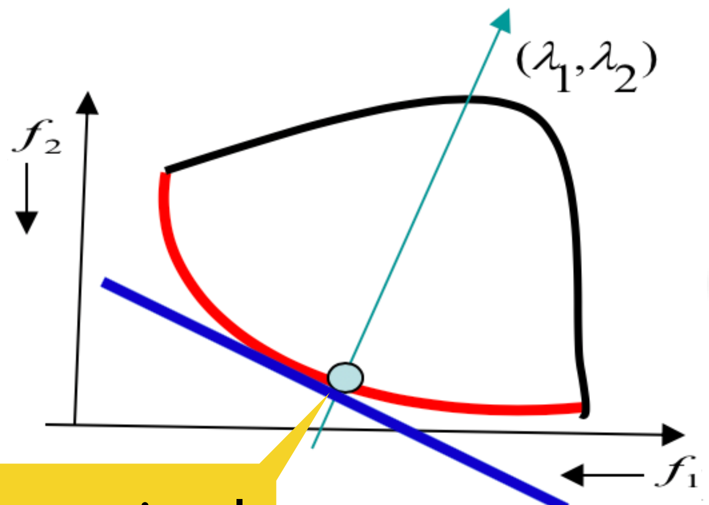


$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

It works for convex PF.

Problem Decomposition

- Weighted Sum Approach



Pareto-optimal solution

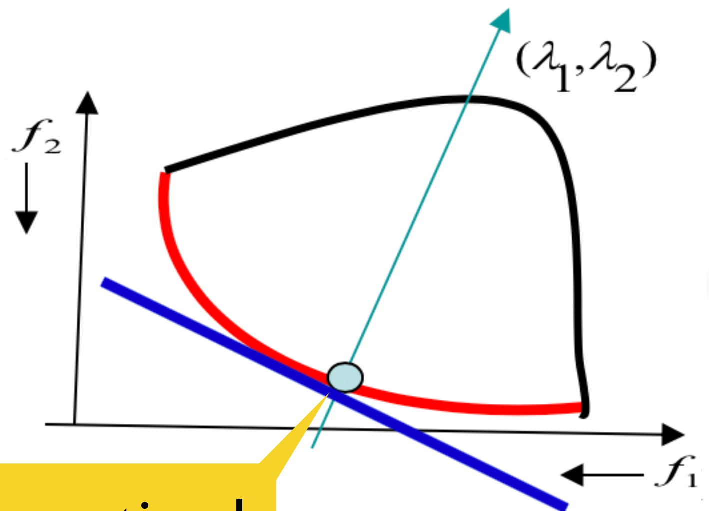
$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

It works for convex PF.

Approximation of the PF → *N* single obj opt suproblems

Problem Decomposition

- Weighted Sum Approach

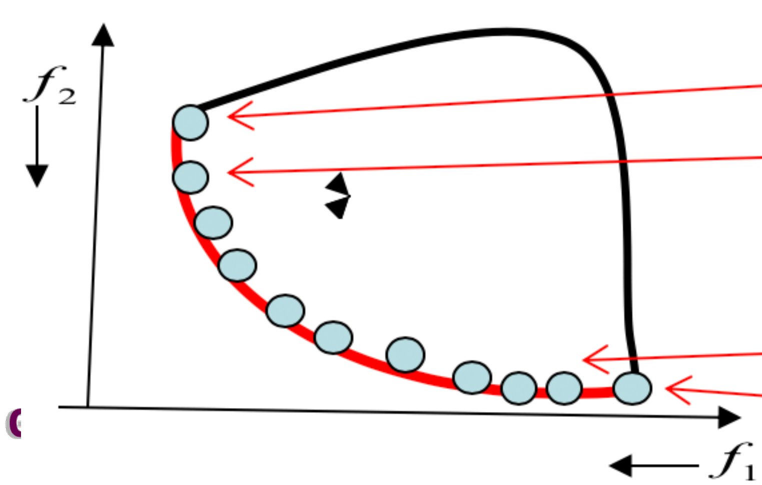


Pareto-optimal solution

$$\min g^{ws}(x \mid \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$
$$\text{where } \lambda_1 + \lambda_2 = 1 \text{ and } \lambda_1, \lambda_2 \geq 0.$$

It works for convex PF.

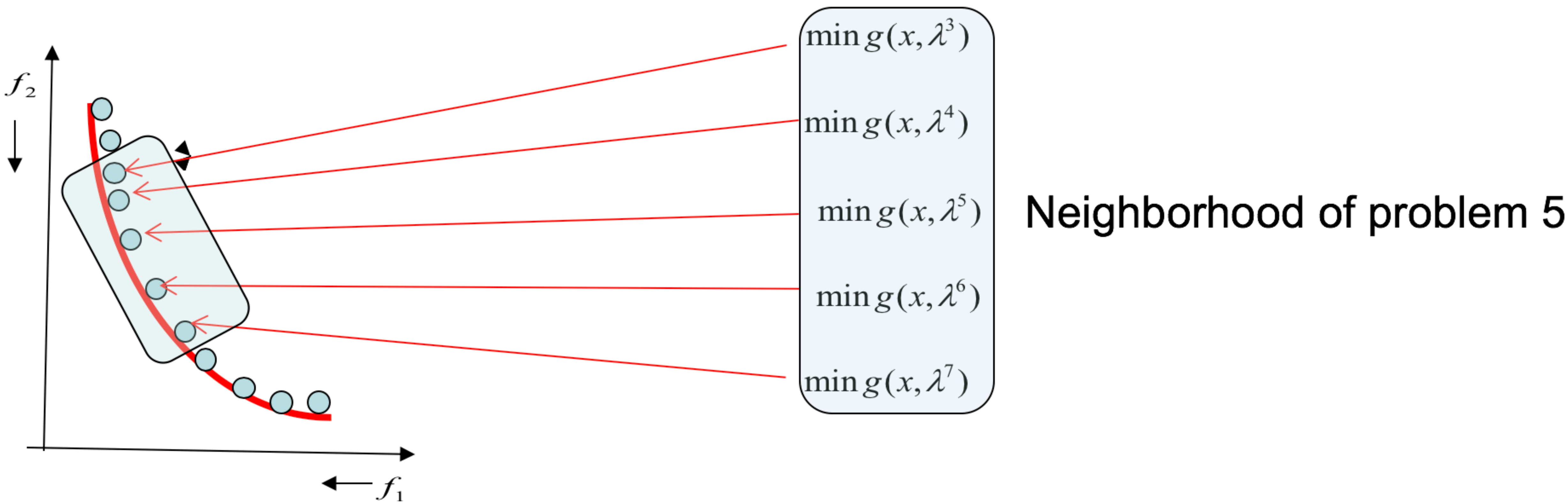
Approximation of the PF → *N* single obj opt suproblems



$$\min g^{ws}(x \mid \lambda^1) = 1 \times f_1(x) + 0 \times f_2(x) \qquad \lambda^1 = (1, 0)$$
$$\min g^{ws}(x \mid \lambda^2) = 0.9 \times f_1(x) + 0.1 \times f_2(x) \qquad \lambda^2 = (0.9, 0.1)$$
$$\vdots$$
$$\min g^{ws}(x \mid \lambda^{10}) = 0.1 \times f_1(x) + 0.9 \times f_2(x) \qquad \lambda^{10} = (0.1, 0.9)$$
$$\min g^{ws}(x \mid \lambda^{11}) = 0 \times f_1(x) + 1 \times f_2(x) \qquad \lambda^{11} = (0, 1)$$

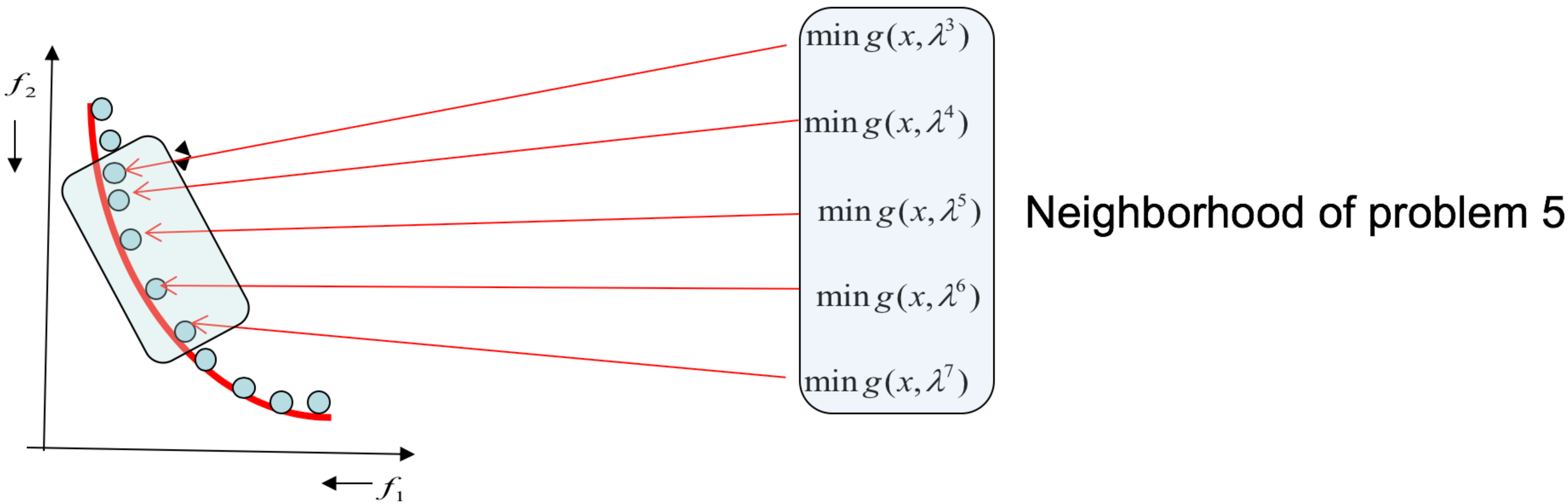
Neighbourhood Structure

- **Neighbourhood**: measure the relationship among subtasks
  - Two subproblems are neighbours if their weight vectors are close.
  - Neighbouring subproblems should have similar objective functions and thus similar optimal solutions with high probability.



Neighbourhood Structure

- **Neighbourhood**: measure the relationship among subtasks
  - Two subproblems are neighbours if their weight vectors are close.
  - Neighbouring subproblems should have similar objective functions and thus similar optimal solutions with high probability.



Many different ways for defining neighbourhood structure.

Search Method Used by Each Agent

Search Method Used by Each Agent

- Memory

Search Method Used by Each Agent

- Memory
  - At each generation, agent i records only one candidate solution $\mathbf{x}^i$, the best solution found so far for its subproblem.

Search Method Used by Each Agent

- Memory
  - At each generation, agent i records only one candidate solution $\mathbf{x}^i$, the best solution found so far for its subproblem.

- Reproduction Operators

Search Method Used by Each Agent

- Memory
  - At each generation, agent i records only one candidate solution $\mathbf{x}^i$, the best solution found so far for its subproblem.

- Reproduction Operators
  - genetic operators: Crossover/mutation

Collaboration among different agents

Collaboration among different agents

- Mating selection

Collaboration among different agents

- Mating selection
  - Each agent borrows solutions from its neighbours for generating new solutions.

Collaboration among different agents

- Mating selection
  - Each agent borrows solutions from its neighbours for generating new solutions.

- Replacement

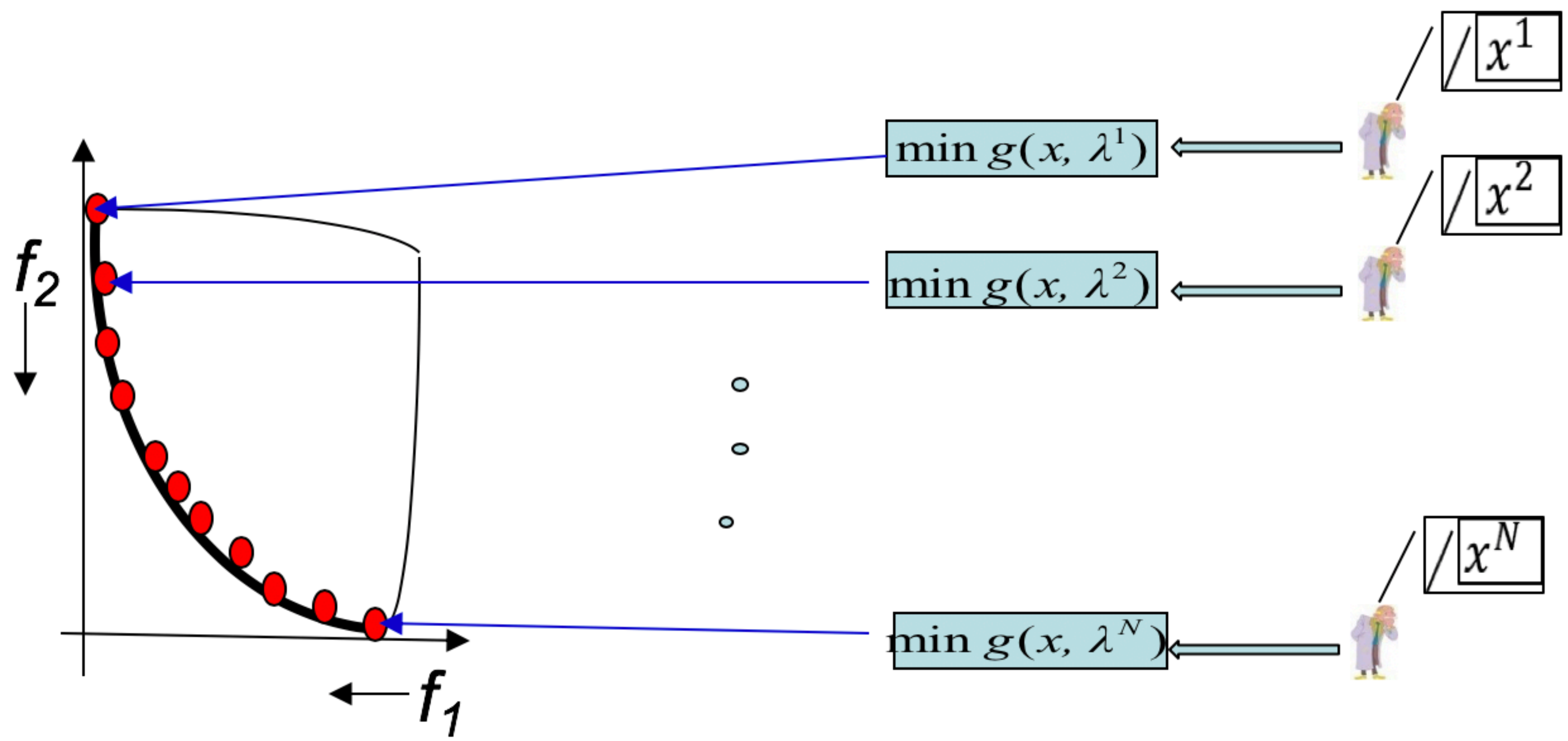
Collaboration among different agents

- Mating selection
  - Each agent borrows solutions from its neighbours for generating new solutions.

- Replacement
  - Each agent sends new solutions to its neighbours.

General Working Principle of MOEA/D

Decomposition, Neighbourhood, Memory



- At each generation, each agent does the following:
  - **Mating selection**: obtain the current solutions of some neighbours (collaboration).
  - **Reproduction**: **generate** a new solution by applying *reproduction operators* on its own solution and borrowed solutions.
  - **Replacement**:
    - replace its old solution by the new one if the new one is better than old one for its objective.
    - pass the new solution on to some of its neighbours, each of them replaces its old solution by this new one if it is better for its objective. (*collaboration*, *neighbourhood*)

Experimental Results

- Continuous MOP test instances.
- Same population size and same Xover and mutation.
- Same number of objective function evaluations.

D-METRIC VALUES OF THE SOLUTIONS FOUND BY MOEA/D WITH THE TCHEBYCHEFF APPROACH AND NSGA-II. THE NUMBERS IN PARENTHESES REPRESENT THE STANDARD DEVIATION

		NSGA-II	MOEA/D
Instance	ZDT1	0.0050 (0.0002)	0.0055 (0.0039)
	ZDT2	0.0049 (0.0002)	0.0079 (0.0109)
	ZDT3	0.0065 (0.0054)	0.0143 (0.0091)
	ZDT4	0.0182 (0.0237)	0.0076 (0.0023)
	ZDT6	0.0169 (0.0028)	0.0042 (0.0003)
	DTLZ1	0.0648 (0.1015)	0.0317 (0.0005)
	DTLZ2	0.0417 (0.0013)	0.0389 (0.0001)

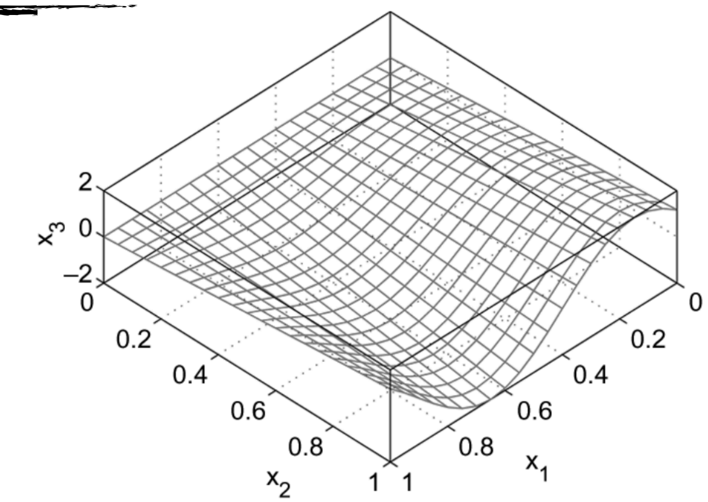
AVERAGE CPU TIME (IN SECONDS) USED BY NSGA-II AND MOEA/D WITH THE TCHEBYCHEFF APPROACH

		NSGA-II	MOEA/D
Instance	ZDT1	1.03	0.60
	ZDT2	1.00	0.47
	ZDT3	1.03	0.57
	ZDT4	0.77	0.33
	ZDT6	0.73	0.27
	DTLZ1	10.27	1.20
	DTLZ2	8.37	1.10

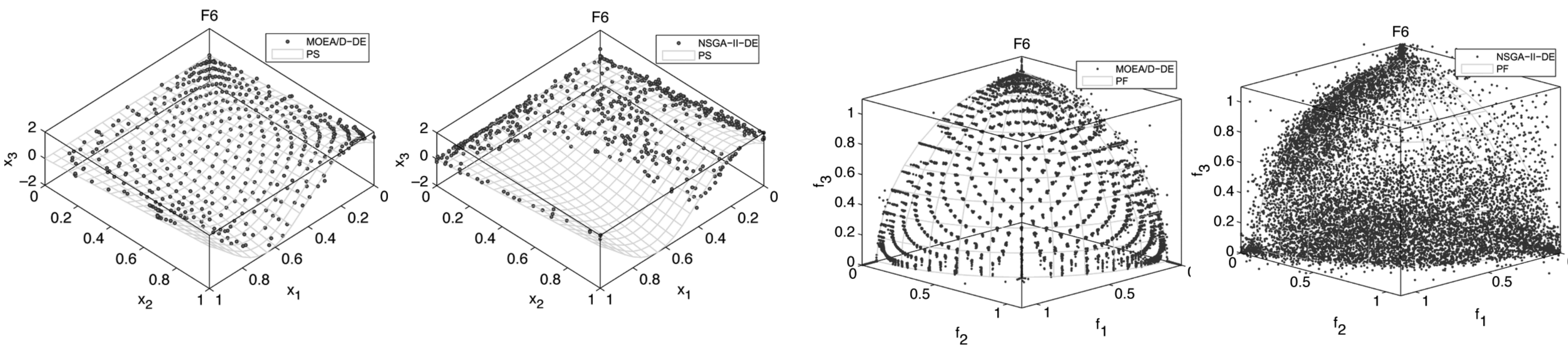
- Observation: It works.
  - Solution quality: MOEA/D ≈NSGA-II
  - CPU time: MOEA/D is better.

Test Instances with complicated PS shapes

F6	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_3}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ where $J_1 = \{j \mid 3 \le j \le n$, and $j-1$ is a multiplication of $3\}$, $J_2 = \{j \mid 3 \le j \le n$, and $j-2$ is a multiplication of $3\}$, $J_3 = \{j \mid 3 \le j \le n$, and j is a multiplication of $3\}$ Its PS is $x_j = 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}), j = 3, \ldots, n$.	$[0,1]^2 \times [-2,2]^{n-2}$



F6 ($n = 10$)



- MOEA/D is better.
- Combination of MOEA/D and NSGAII (Cai, et al): Champion in 2017 contest

Some Remarks

- Diversity among subproblems leads to diversity among $\{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$
- MOEA/D has a well-organised memory.
- It deals with a population of subtasks, related to recent proposed evolutionary multi-task optimisation (Y-S Ong et al, 2016).

Outline

- **Part I: Basics**
  - Basic Concepts
  - Simple MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
  - Resources
  - Future Directions

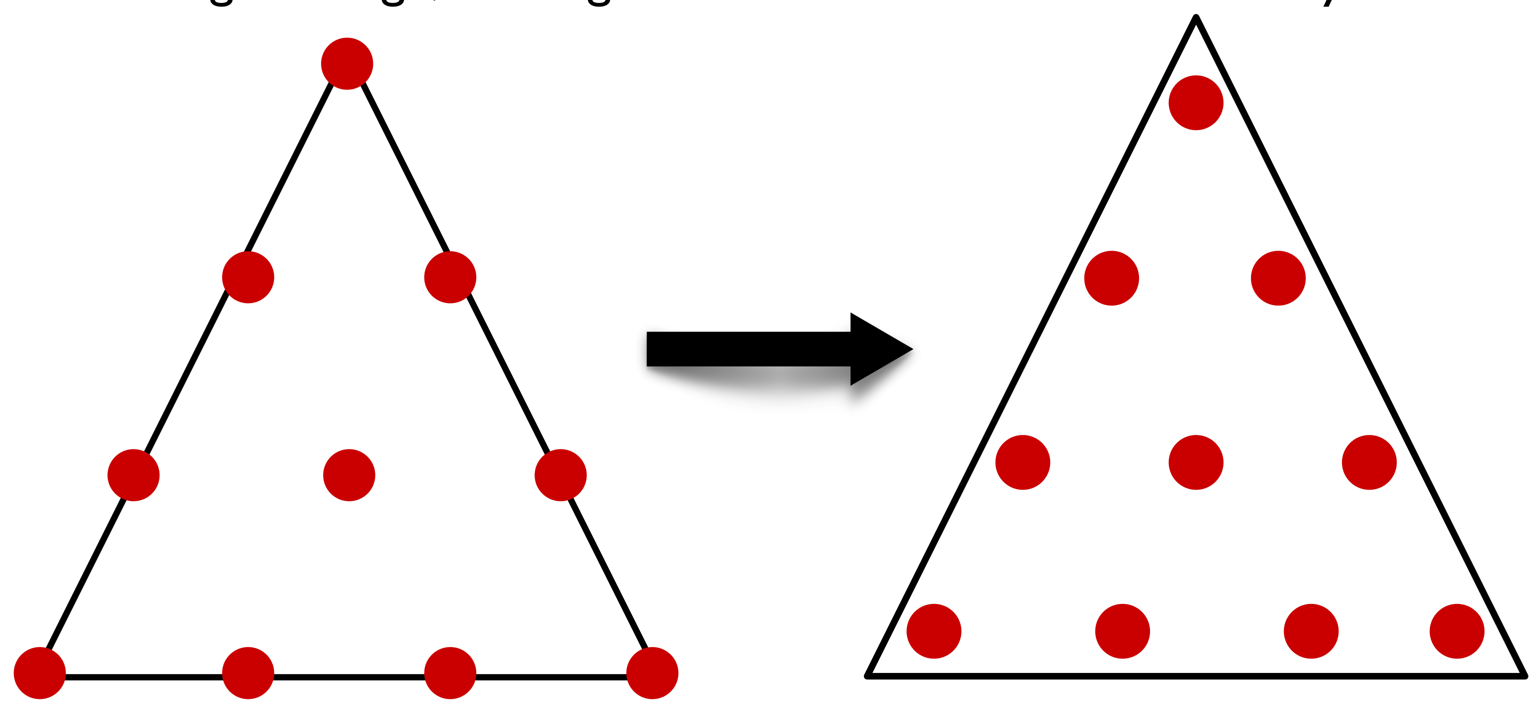
Setting of Weight Vectors

- Drawbacks of the Das and Dennis's method [1]
  - Not very '*uniform*' [2]
  - Number of weights is restricted to $N = \binom{H+m-1}{m-1}$,
    - N increases nonlinearly with m
    - If N is not large enough, all weight vectors will be at the boundary of the simplex

[1] I. Das et. al., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", SIAM J. Optim, 8(3): 631-657, 1998.
[2] Y-Y Tan, et al., "MOEA/D + Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives", Comput & OR, 40: 1648-1660, 2013.
[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", IEEE Trans. Evol. Comput., 18(4): 577-601, 2014.
[4] **K. Li**, K. Deb, et al., "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition", IEEE Trans. Evol. Comput., 19(5): 694-716, 2015.

Setting of Weight Vectors

- Drawbacks of the Das and Dennis's method [1]
  - Not very '*uniform*' [2]
  - Number of weights is restricted to $N = \binom{H+m-1}{m-1}$,
    - N increases nonlinearly with m
    - If N is not large enough, all weight vectors will be at the boundary of the simplex



simplex-lattice design   uniform design [9]

[1] I. Das et. al., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", SIAM J. Optim, 8(3): 631-657, 1998.
[2] Y-Y Tan, et al., "MOEA/D + Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives", Comput & OR, 40: 1648-1660, 2013.
[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", IEEE Trans. Evol. Comput., 18(4): 577-601, 2014.
[4] **K. Li**, K. Deb, et al., "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition", IEEE Trans. Evol. Comput., 19(5): 694-716, 2015.

Setting of Weight Vectors

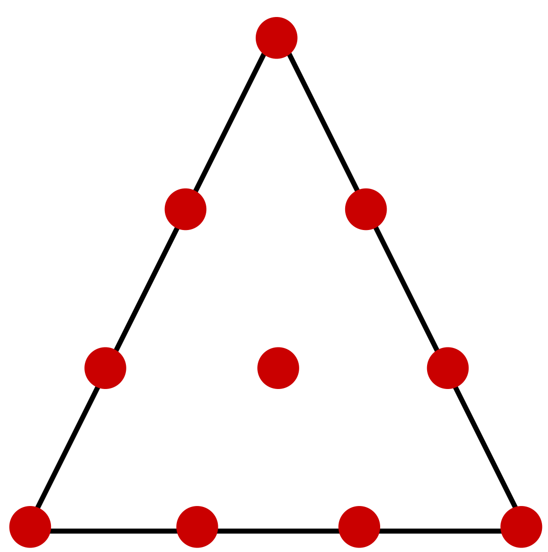
- Drawbacks of the Das and Dennis's method [1]
  - Not very '*uniform*' [2]
  - Number of weights is restricted to $N = \binom{H+m-1}{m-1}$,
    - N increases nonlinearly with m
    - If N is not large enough, all weight vectors will be at the boundary of the simplex

---

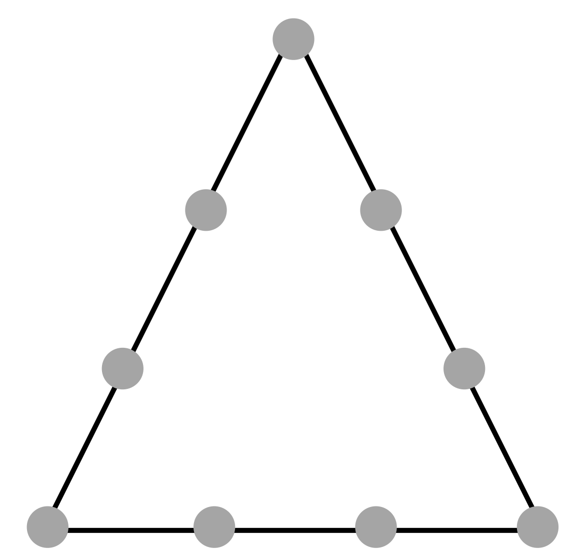
[1] I. Das et. al., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", SIAM J. Optim, 8(3): 631-657, 1998.
[2] Y-Y Tan, et al., "MOEA/D + Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives", Comput & OR, 40: 1648-1660, 2013.
[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", IEEE Trans. Evol. Comput., 18(4): 577-601, 2014.
[4] **K. Li**, K. Deb, et al., "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition", IEEE Trans. Evol. Comput., 19(5): 694-716, 2015.

Setting of Weight Vectors

- Drawbacks of the Das and Dennis's method [1]
  - Not very '*uniform*' [2]
  - Number of weights is restricted to $N = \binom{H+m-1}{m-1}$,
    - N increases nonlinearly with m
    - If N is not large enough, all weight vectors will be at the boundary of the simplex



1st layer



2nd layer

[1] I. Das et. al., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", SIAM J. Optim, 8(3): 631-657, 1998.

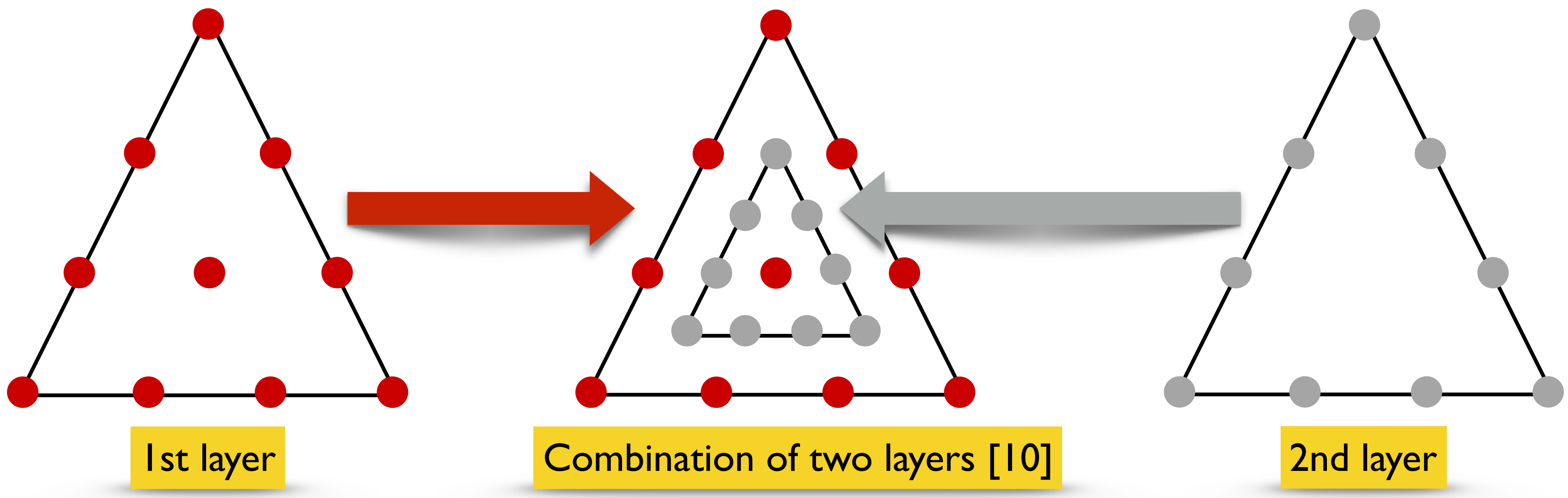
[2] Y-Y Tan, et al., "MOEA/D + Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives", Comput & OR, 40: 1648-1660, 2013.

[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", IEEE Trans. Evol. Comput., 18(4): 577-601, 2014.

[4] **K. Li**, K. Deb, et al., "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition", IEEE Trans. Evol. Comput., 19(5): 694-716, 2015.

Setting of Weight Vectors

- Drawbacks of the Das and Dennis's method [1]
  - Not very '*uniform*' [2]
  - Number of weights is restricted to $N = \binom{H+m-1}{m-1}$,
    - N increases nonlinearly with m
    - If N is not large enough, all weight vectors will be at the boundary of the simplex



[1] I. Das et. al., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems", SIAM J. Optim, 8(3): 631-657, 1998.

[2] Y-Y Tan, et al., "MOEA/D + Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives", Comput & OR, 40: 1648-1660, 2013.

[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints", IEEE Trans. Evol. Comput., 18(4): 577-601, 2014.

[4] **K. Li**, K. Deb, et al., "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition", IEEE Trans. Evol. Comput., 19(5): 694-716, 2015.

Setting of Weight Vectors (cont.)

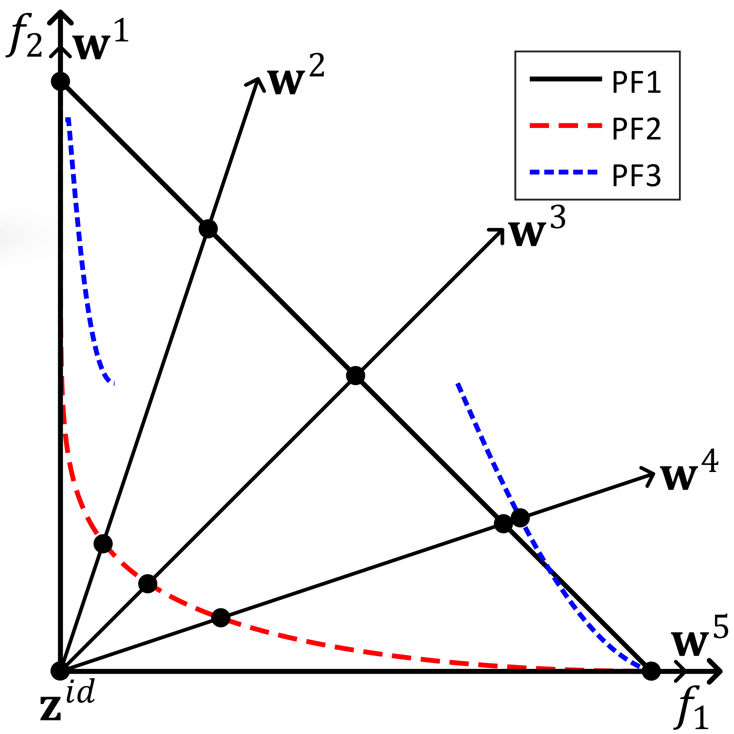
- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …

[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

Setting of Weight Vectors (cont.)

- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …

If the PF meets $\sum_{i=1}^{m} f_i = 1$, that's fine; otherwise …



[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

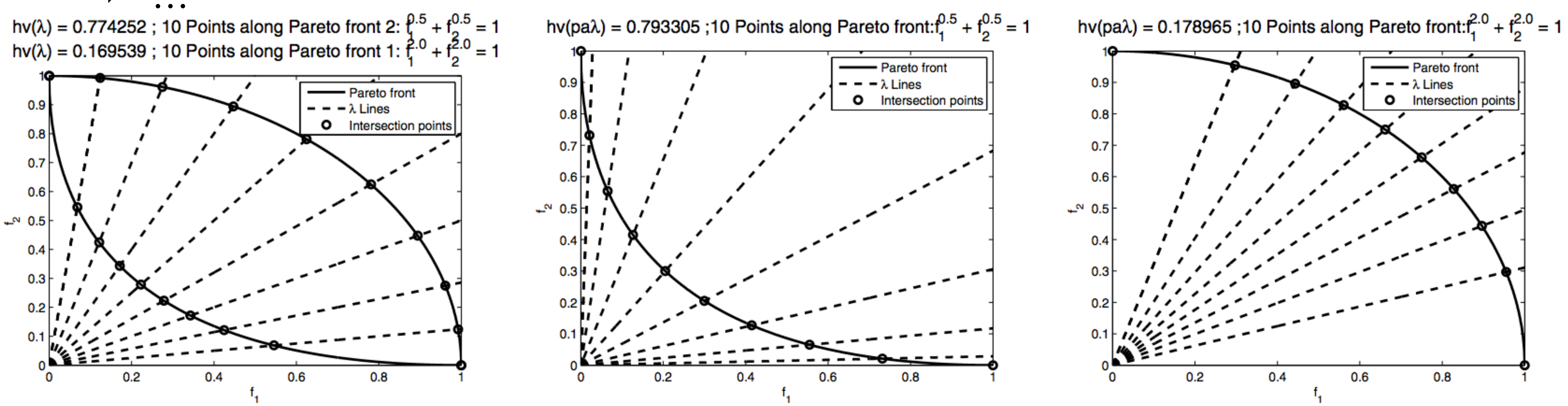
Setting of Weight Vectors (cont.)

- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …

[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

Setting of Weight Vectors (cont.)

- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …



(a) λ (MOEA/D) for convex and concave PF (b) $pa\lambda$ for convex PF (c) $pa\lambda$ for concave PF

Assume PF as $\sum_{i=1}^{m} f_i^p = 1$, estimate p according to the current non-dominated solutions [5]

[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

Setting of Weight Vectors (cont.)

- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …

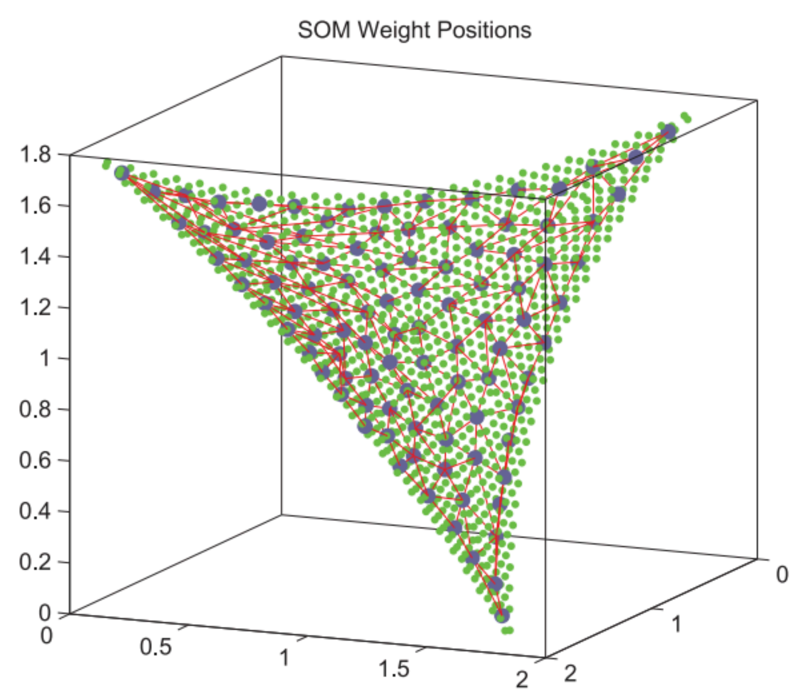
[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

Setting of Weight Vectors (cont.)

- Drawbacks of evenly distributed weight vectors
  - Do NOT always lead to evenly distributed solutions
  - Do NOT support all PF shapes
    - Disconnected PF
    - Inverted PF
    - …

<u>Adaptive weight vectors adjustment</u>

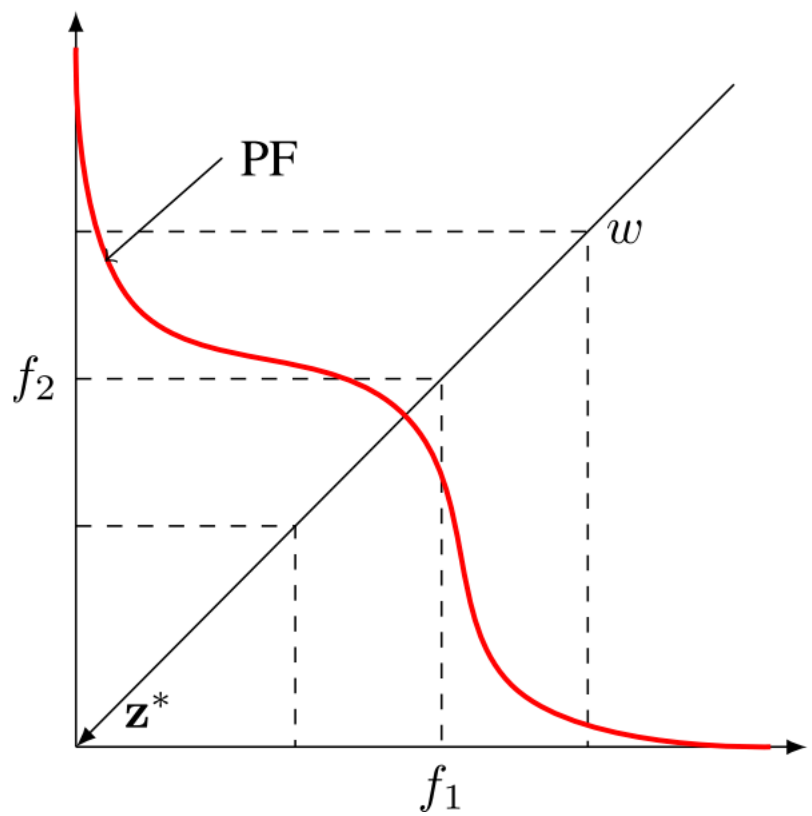
- Estimate the PF shape progressively according to the current population
- Resample a set of weight vectors according to the estimated PF
  - ✓ Add new ones in feasible regions, and remove useless ones from infeasible regions [6]
  - ✓ Sampling from some estimated model, e.g. GP [7] and SOM [8]
- Construct new subproblems with respect to newly sampled weight vectors



[5] S. Jiang, et al., "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", ICNC'11, 1260-1264, 2011.
[6] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.
[7] M. Wu, **K. Li**, et al., "Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization", GECCO'17, 641–648, 2017.
[8] F. Gu, et al., "Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 22(2): 211-225, 2018.

Revisit Weighted Tchebycheff

- Weighted Tchebycheff



weighted Tchebycheff

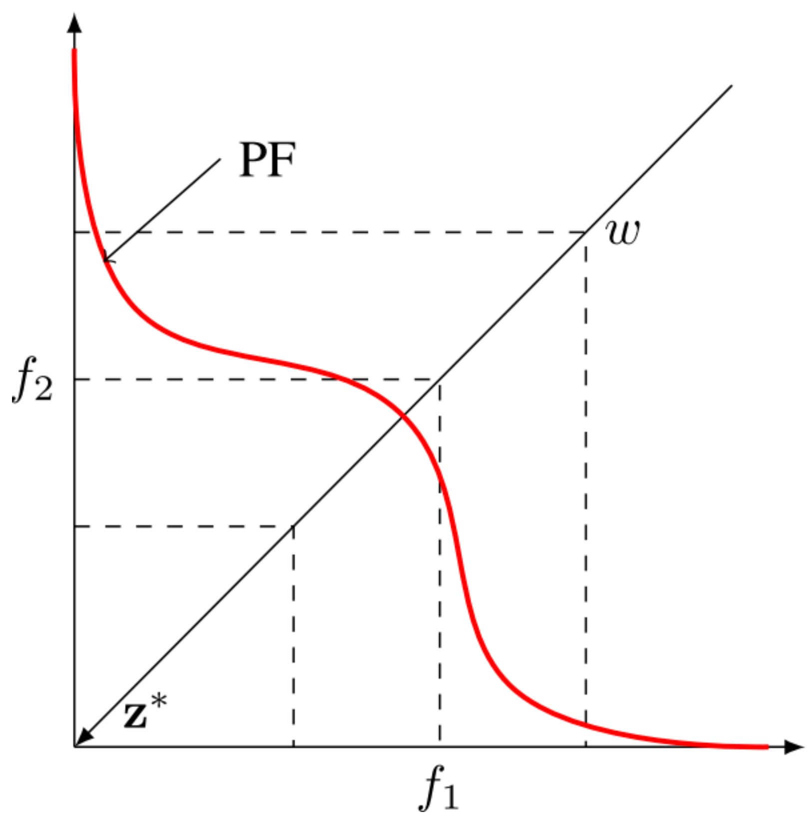
$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity

Revisit Weighted Tchebycheff

- Weighted Tchebycheff



weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

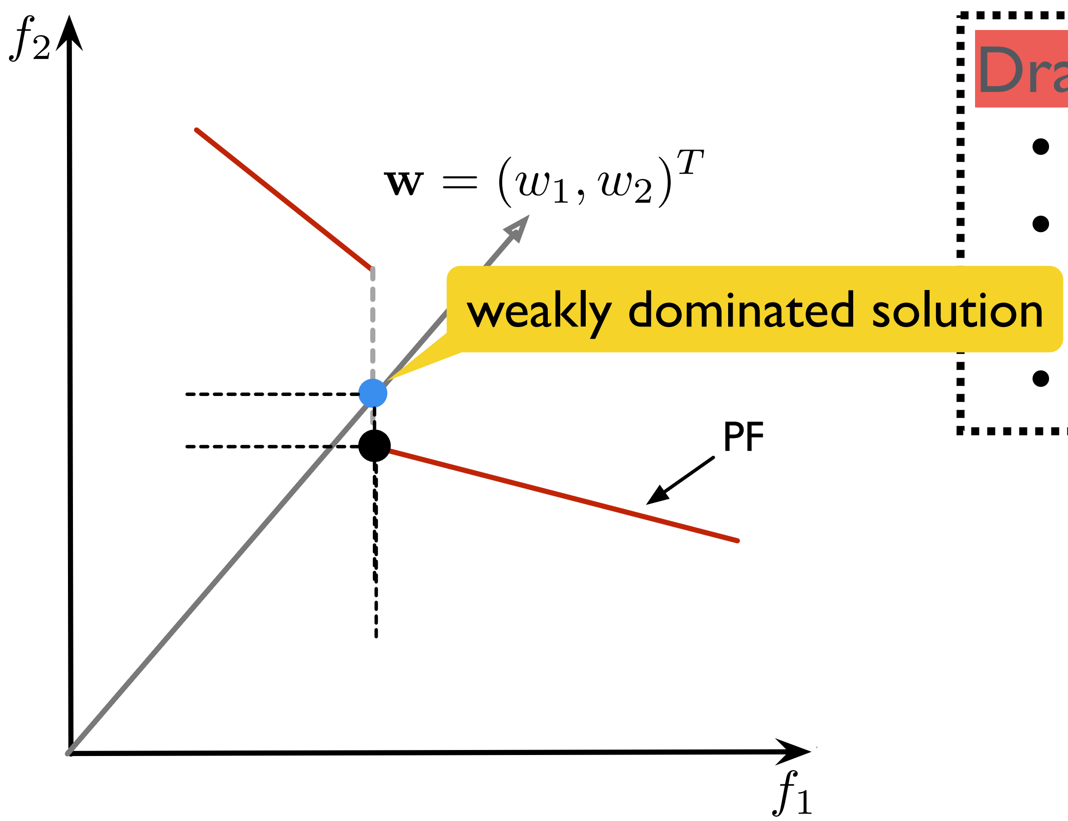
Drawbacks:

- non-smooth, weakly dominated solution [8]
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity

[8] K. Miettinen, et al., "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Boston, 1999

Revisit Weighted Tchebycheff

- Weighted Tchebycheff



weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

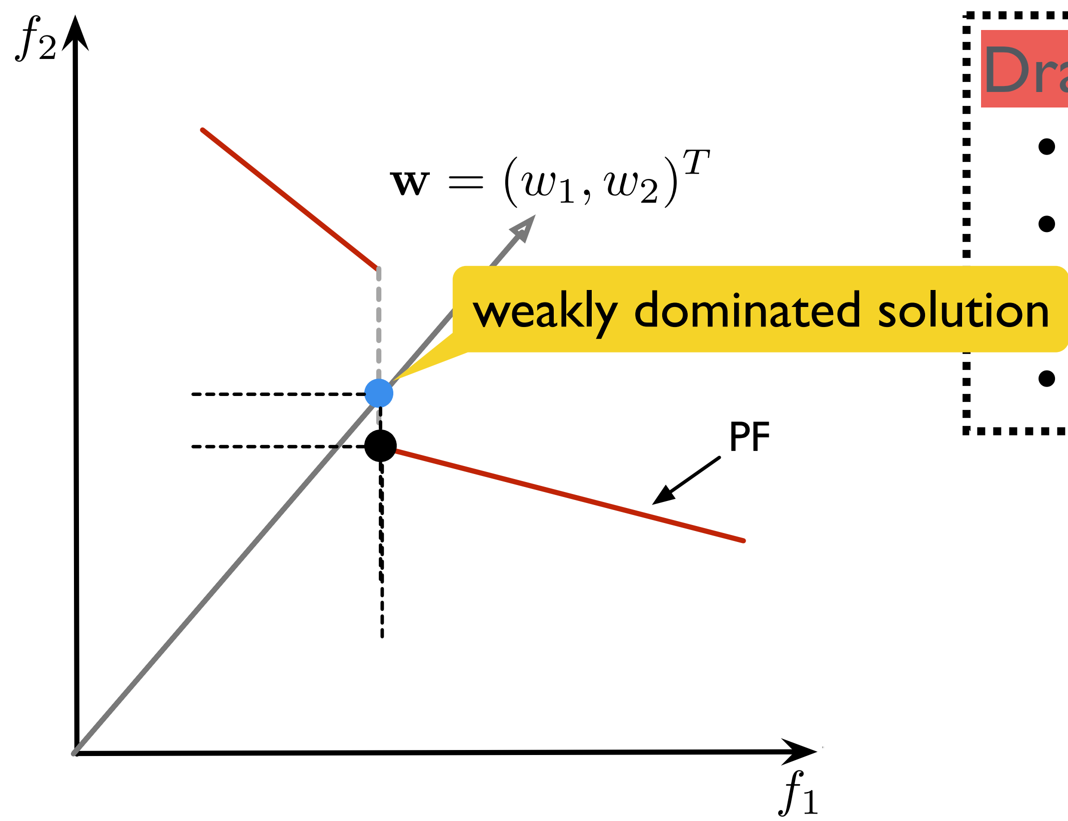
Drawbacks:

- non-smooth, weakly dominated solution [8]
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity

[8] K. Miettinen, et al., "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Boston, 1999

Revisit Weighted Tchebycheff

- Weighted Tchebycheff



Drawbacks:

- non-smooth, weakly dominated solution [8]
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity

weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x}-z_i^*|$$

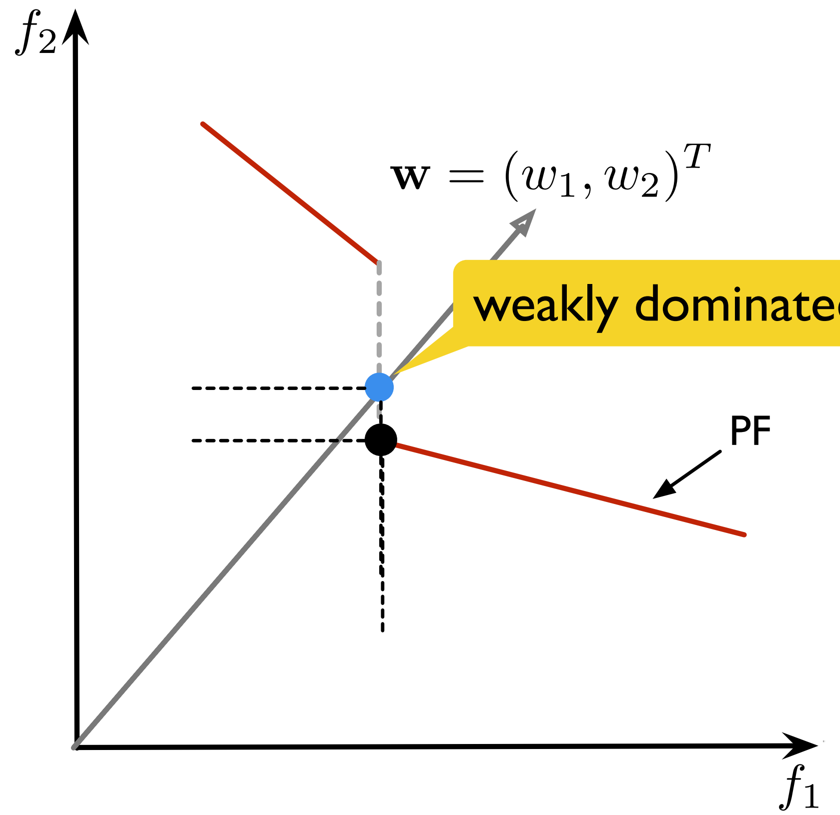
augmented scalarizing function

$$g^a(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m}(\frac{f_i(\mathbf{x}-z_i^*}{w_i}) + \rho\sum_{i=1}^{m}(\frac{f_i(\mathbf{x}-z_i^*}{w_i})$$

[8] K. Miettinen, et al., "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Boston, 1999

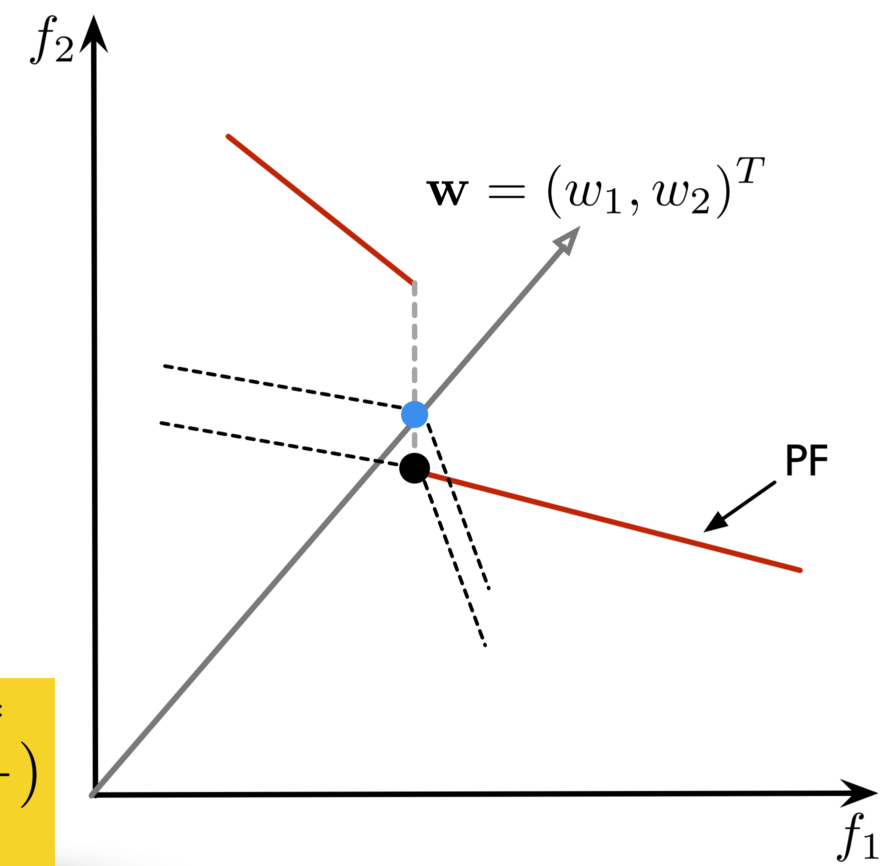
Revisit Weighted Tchebycheff

- Weighted Tchebycheff



Drawbacks:

- non-smooth, weakly dominated solution [8]
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity



weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

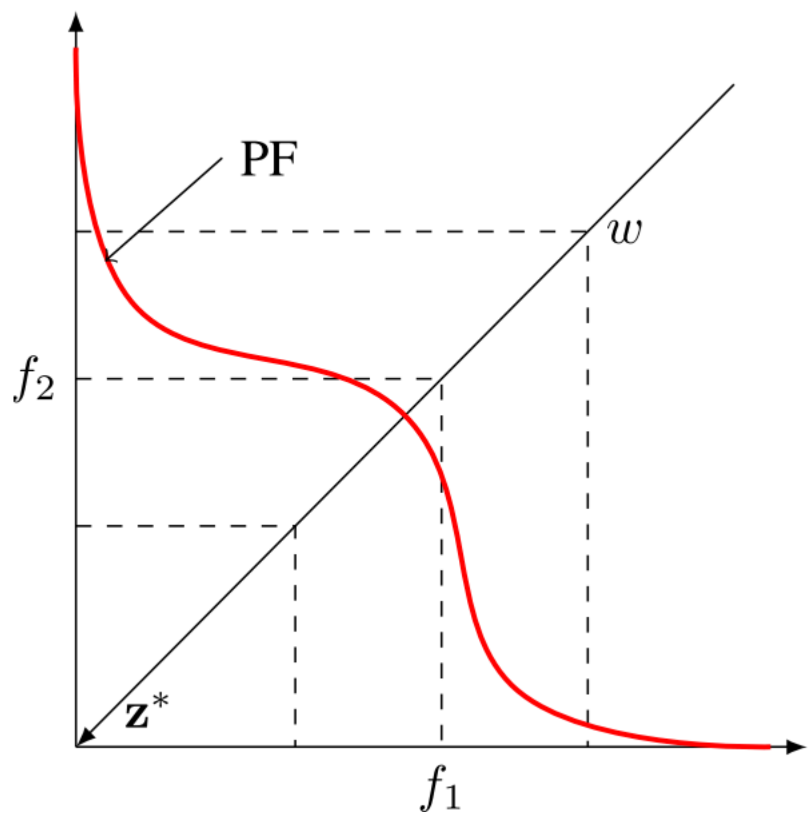
augmented scalarizing function

$$g^a(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} (\frac{f_i(\mathbf{x} - z_i^*}{w_i}) + \rho\sum_{i=1}^{m}(\frac{f_i(\mathbf{x} - z_i^*}{w_i})$$

[8] K. Miettinen, et al., "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Boston, 1999

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff



weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1 \le i \le m} w_i |f_i(\mathbf{x} - z_i^*|$$

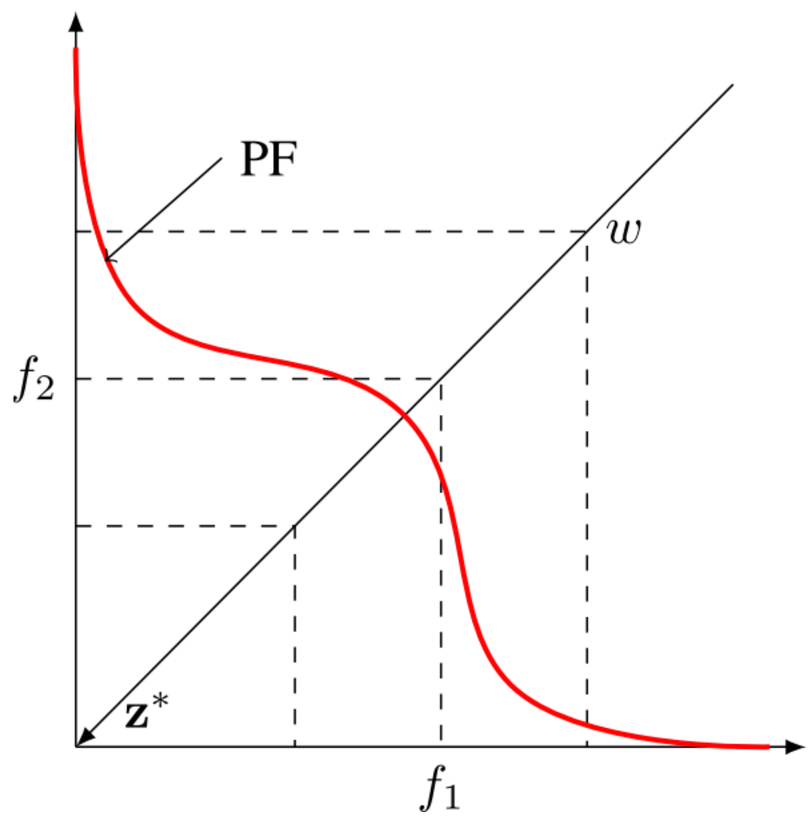
Drawbacks:

- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions [9]
- might easily loose diversity

[9] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff

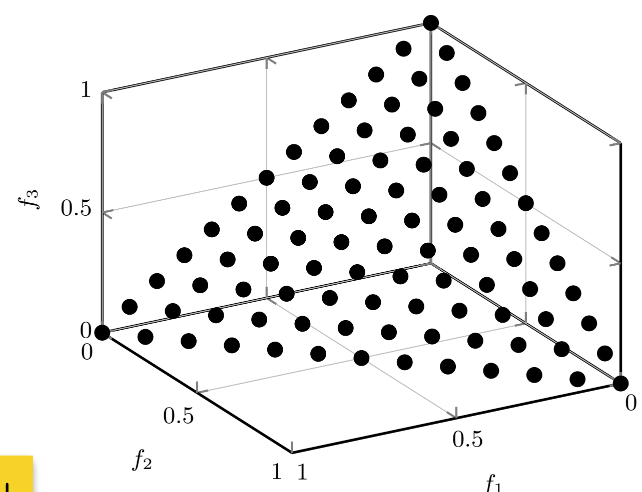


weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

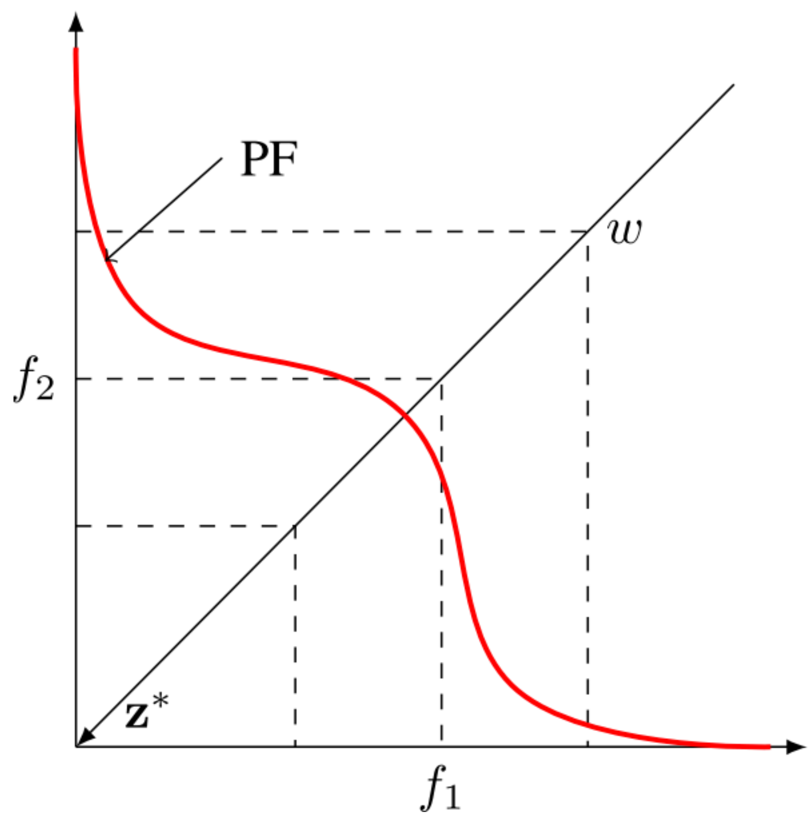
- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions [9]
- might easily loose diversity



[9] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff

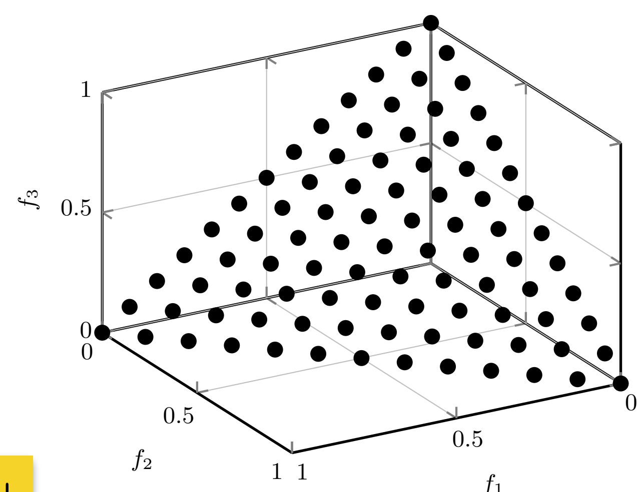


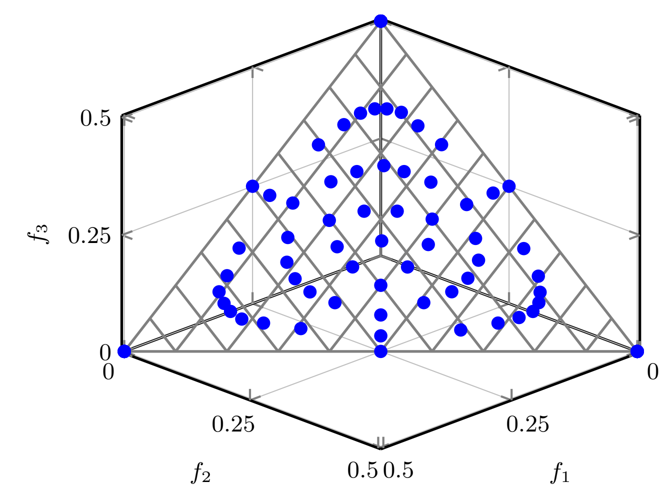
weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions [9]
- might easily loose diversity

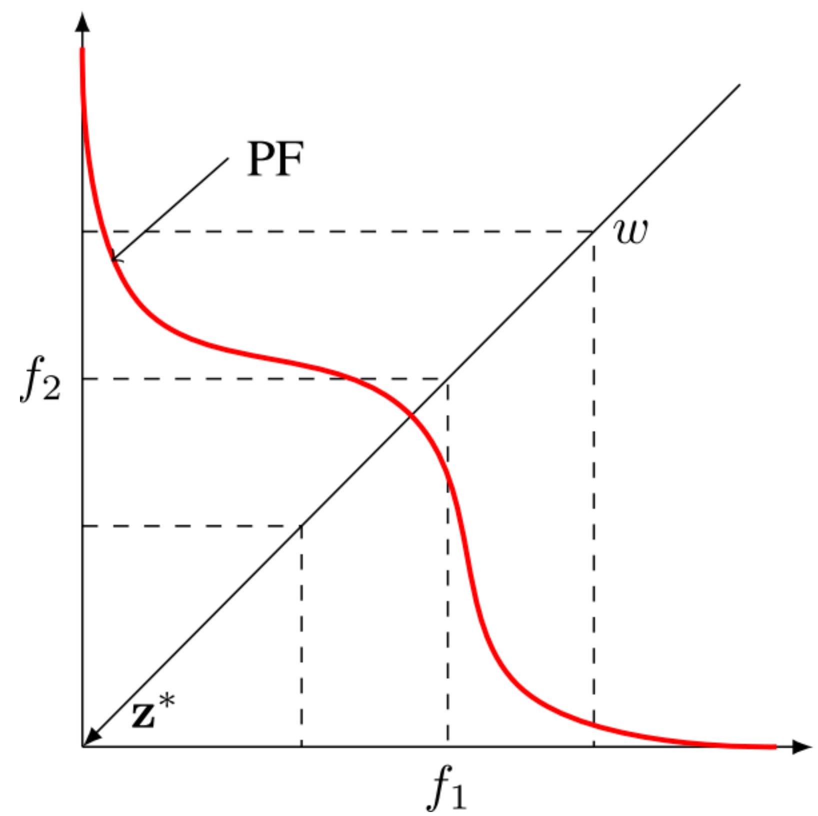




[9] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff

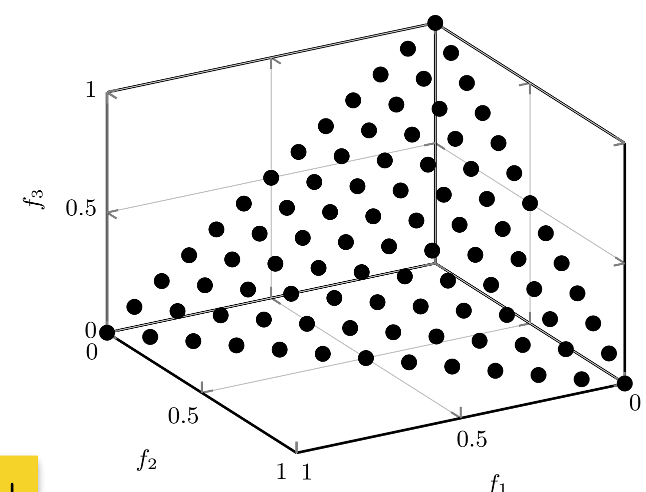


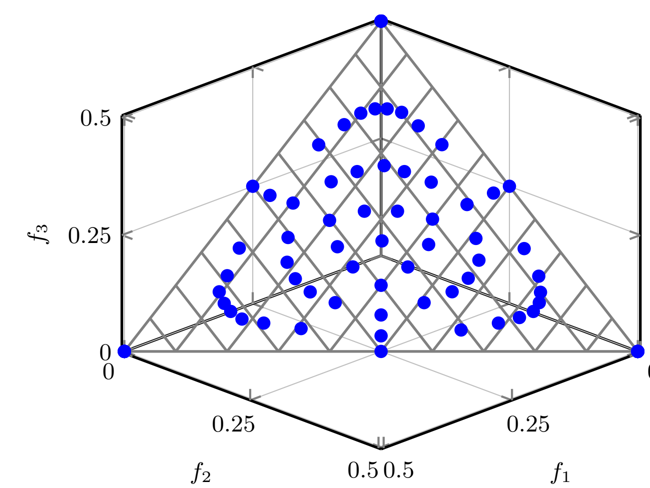
weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions [9]
- might easily loose diversity



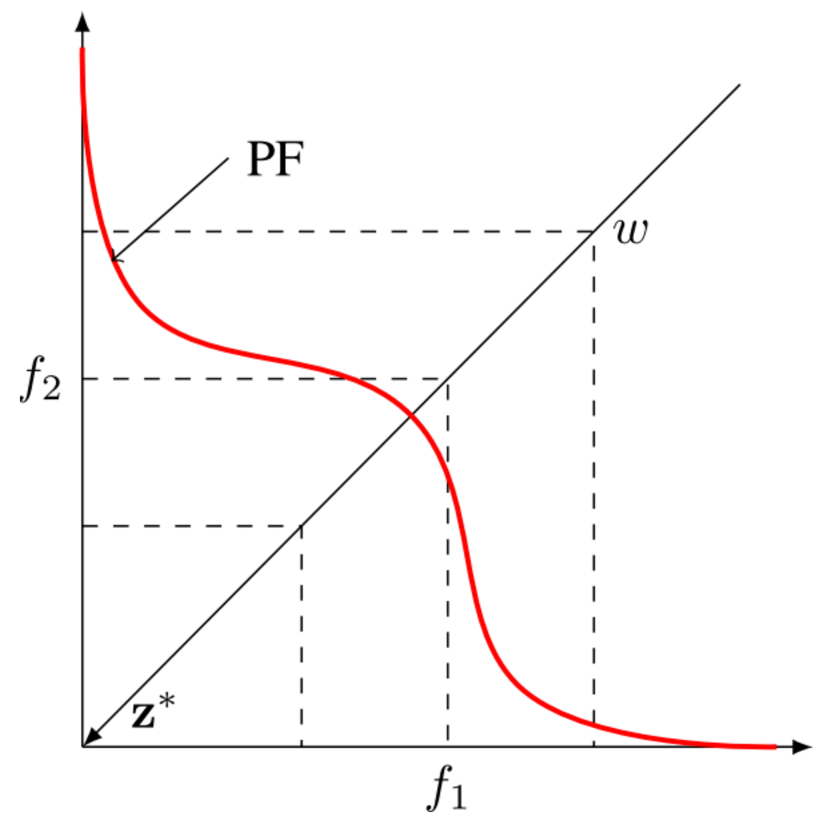


The search direction for $\mathbf{w} = (w_1, \cdots, w_m)^T$ is $\mathbf{w} = (\frac{1/w_1}{\sum_{i=1}^m 1/w_i}, \cdots, \frac{1/w_m}{\sum_{I=1}^m 1/w_i})^T$

[9] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff

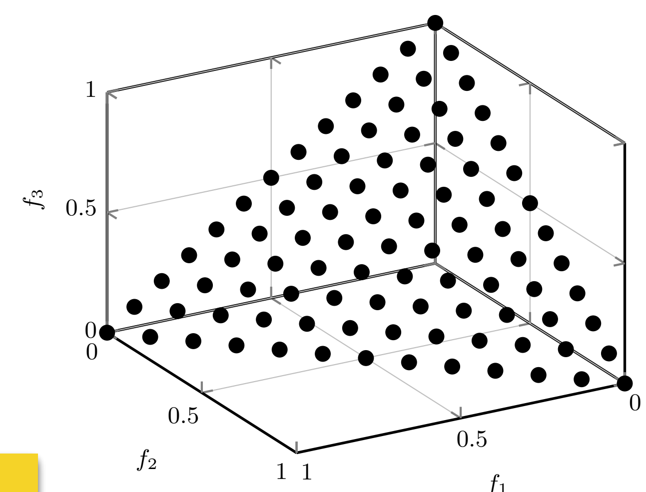


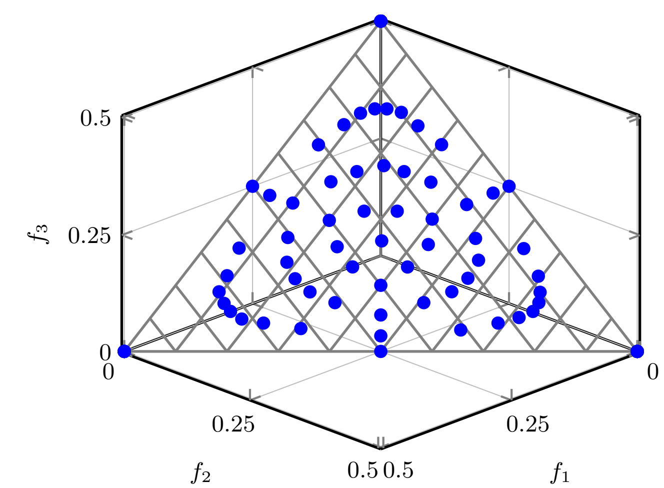
weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} \frac{1}{w_i} |f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions [9]
- might easily loose diversity



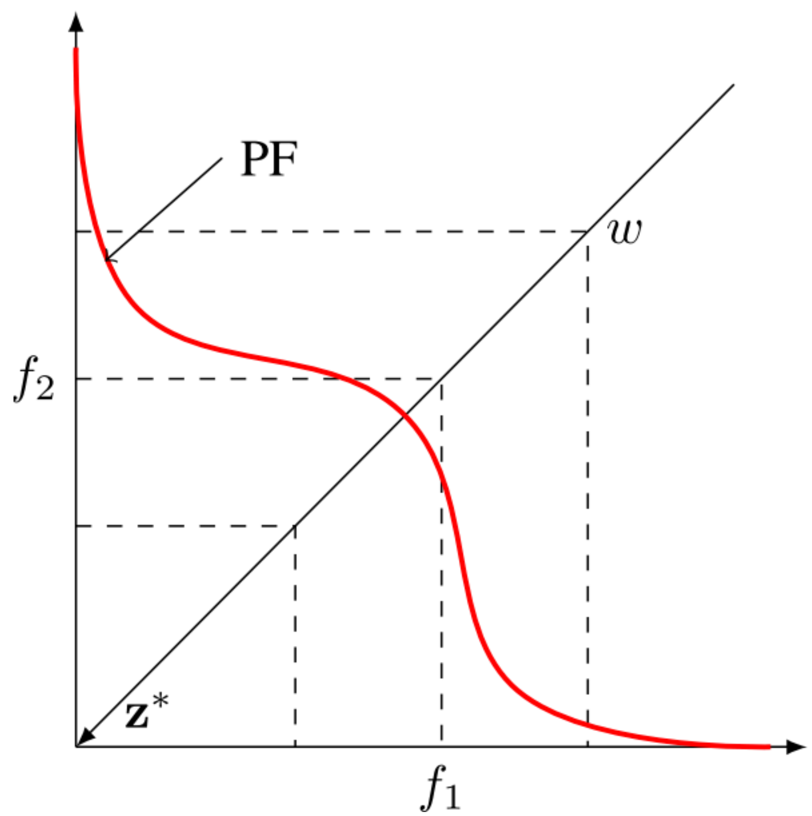


The search direction for $\mathbf{w} = (w_1, \cdots, w_m)^T$ is $\mathbf{w} = (\frac{1/w_1}{\sum_{i=1}^{m} 1/w_i}, \cdots, \frac{1/w_m}{\sum_{I=1}^{m} 1/w_i})^T$

[9] Y. Qi, et al., "MOEA/D with Adaptive Weight Adjustment", Evol. Comput. 22(2): 231–264, 2014.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff

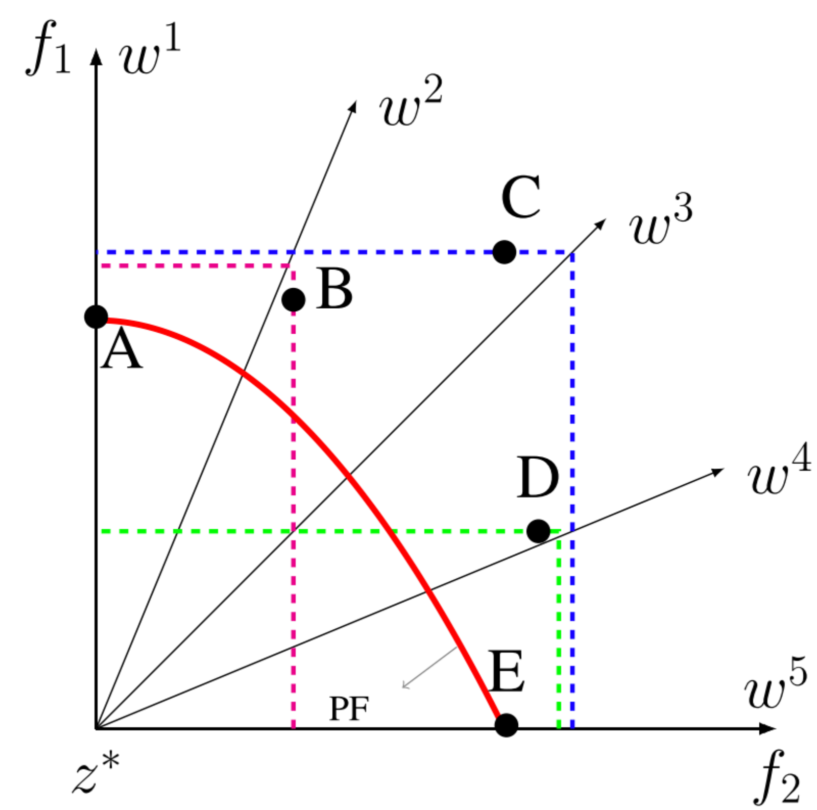


weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

Drawbacks:

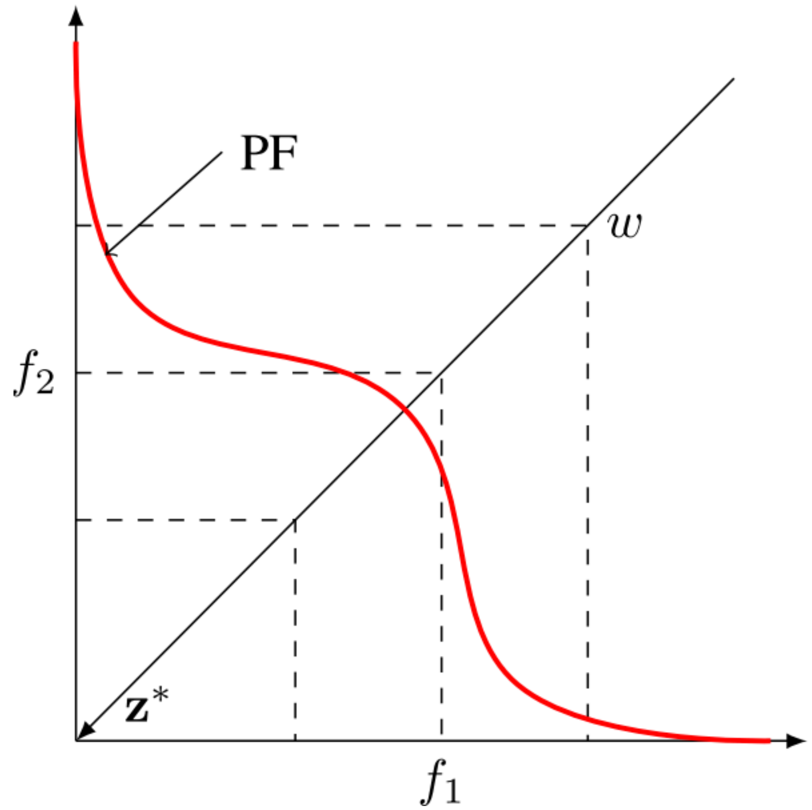
- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity [10]



[10] S. Jiang, et al., "Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms", IEEE Trans. Evol. Comput., 22(2): 296-313, 2018.

Revisit Weighted Tchebycheff (cont.)

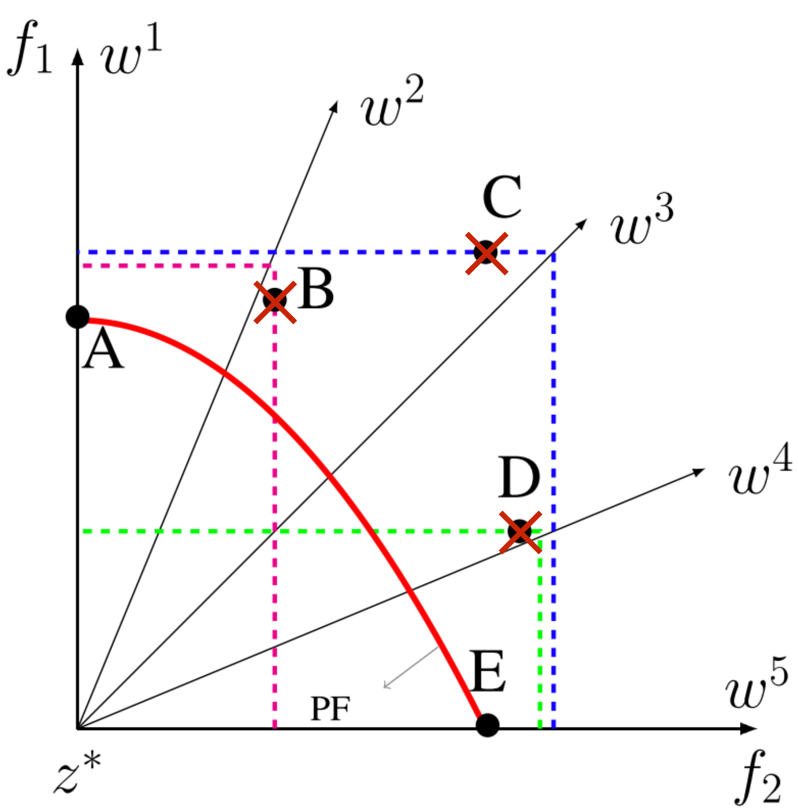
- Weighted Tchebycheff



weighted Tchebycheff

$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\leq i\leq m} w_i|f_i(\mathbf{x} - z_i^*|$$

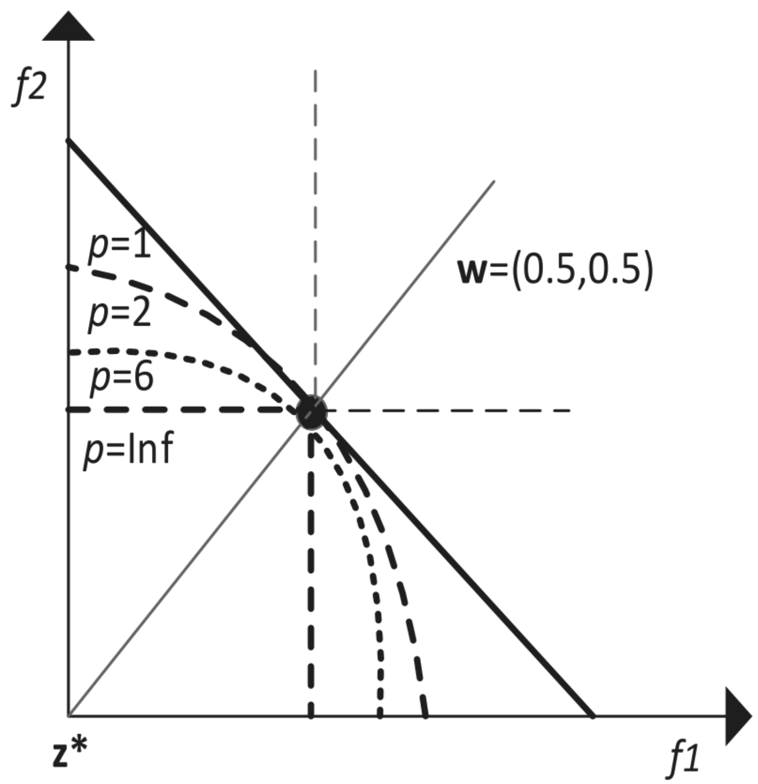
Drawbacks:
- non-smooth, weakly dominated solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity [10]



[10] S. Jiang, et al., "Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms", IEEE Trans. Evol. Comput., 22(2): 296-313, 2018.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff



weighted L_p scalarizing [12]

$$g^{wd}(\mathbf{x}|\mathbf{w}) = (\sum_{I=1}^{m} \lambda_i (f_i(\mathbf{x}) - z_i^*)^p)^{\frac{1}{p}}$$

$$\lambda_i = (\frac{1}{w_i}), p \geq 1$$

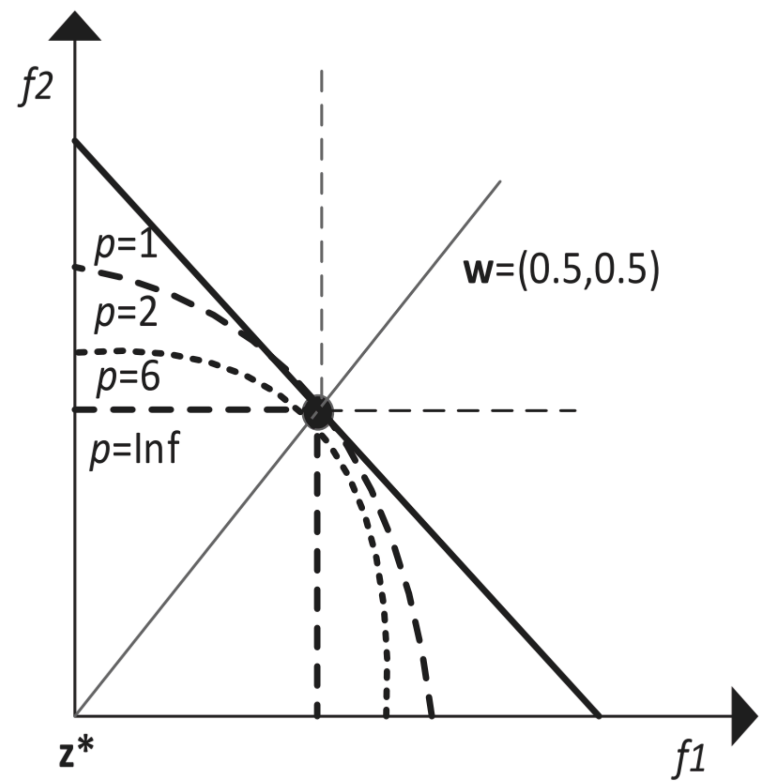
Drawbacks:

- non-smooth, weakly dominate solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity [11]

[11] R. Wang, **Q. Zhang**, et al., "Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods", IEEE Trans. Evol. Comput., 20(6): 821-837, 2016.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff



weighted L_p scalarizing [12]

$$g^{wd}(\mathbf{x}|\mathbf{w}) = (\sum_{I=1}^{m} \lambda_i (f_i(\mathbf{x}) - z_i^*)^p)^{\frac{1}{p}}$$

$$\lambda_i = (\frac{1}{w_i}), p \geq 1$$

Drawbacks:

- non-smooth, weakly dominate solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity [11]

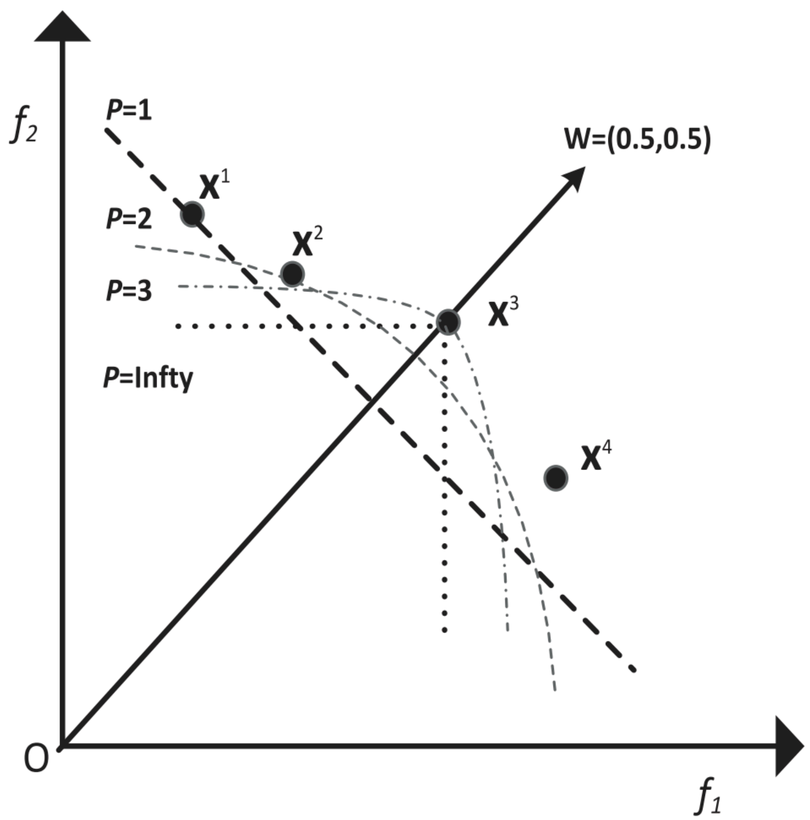
Pareto adaptive scalarising to choose p

$$\begin{aligned} \text{minimize} \quad & p, \quad p \in P \\ \text{subject to} \quad & \forall \mathbf{x}^k : g^{wd}(\mathbf{x}^*|\mathbf{w}, \mathbf{z}^*, p) \\ & \leq g^{wd}(\mathbf{x}^k|\mathbf{w}, \mathbf{z}^*, p) \end{aligned}$$

[11] R. Wang, **Q. Zhang**, et al., "Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods", IEEE Trans. Evol. Comput., 20(6): 821-837, 2016.

Revisit Weighted Tchebycheff (cont.)

- Weighted Tchebycheff



Drawbacks:

- non-smooth, weakly dominate solution
- evenly distributed weights do NOT lead to evenly distributed solutions
- might easily loose diversity [11]

weighted L_p scalarizing [12]

$$g^{wd}(\mathbf{x}|\mathbf{w}) = (\sum_{I=1}^{m} \lambda_i (f_i(\mathbf{x}) - z_i^*)^p)^{\frac{1}{p}}$$

$$\lambda_i = (\frac{1}{w_i}), p \geq 1$$

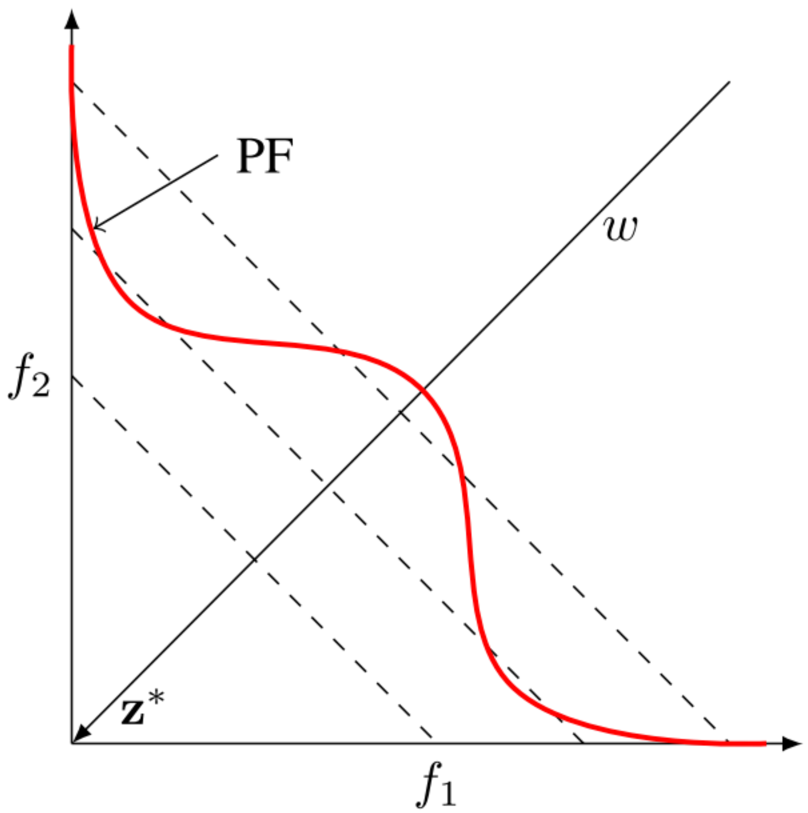
Pareto adaptive scalarising to choose p

$$\begin{aligned} &\text{minimize} \quad p, \quad p \in P \\ &\text{subject to} \quad \forall \mathbf{x}^k : g^{wd}(\mathbf{x}^*|\mathbf{w}, \mathbf{z}^*, p) \\ &\qquad\qquad\qquad \leq g^{wd}(\mathbf{x}^k|\mathbf{w}, \mathbf{z}^*, p) \end{aligned}$$

[11] R. Wang, **Q. Zhang**, et al., "Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods", IEEE Trans. Evol. Comput., 20(6): 821-837, 2016.

Revisit Weighted Sum

- Weighted sum



weighted sum

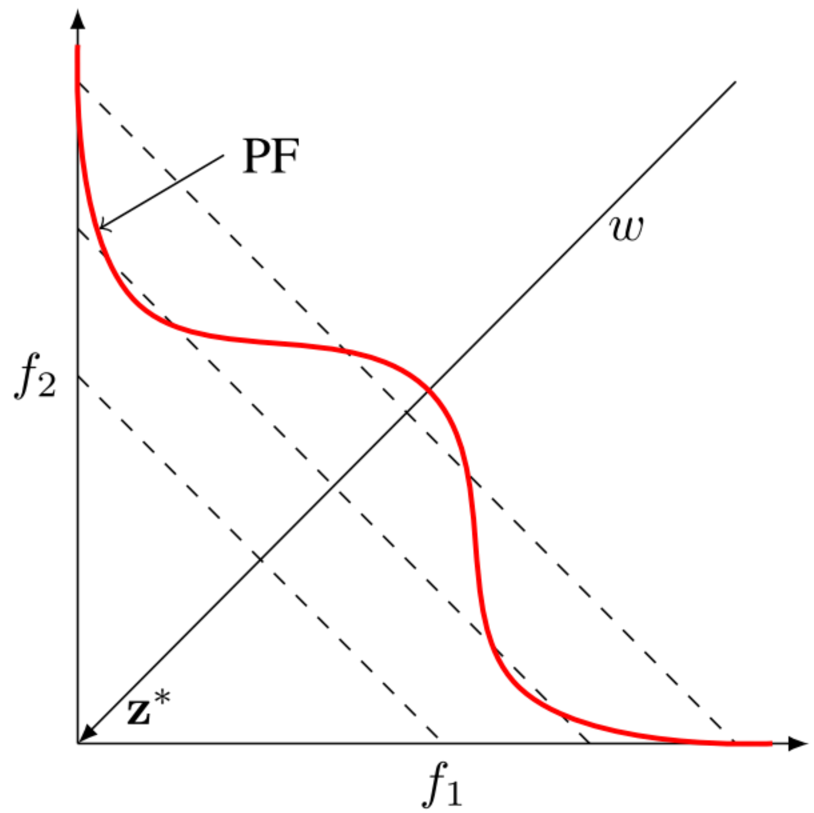
$$g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x})$$

Drawbacks:

- only useful for convex PFs while not all Pareto-optimal solutions can be found if the PF is not convex.
- …

Revisit Weighted Sum

- Weighted sum

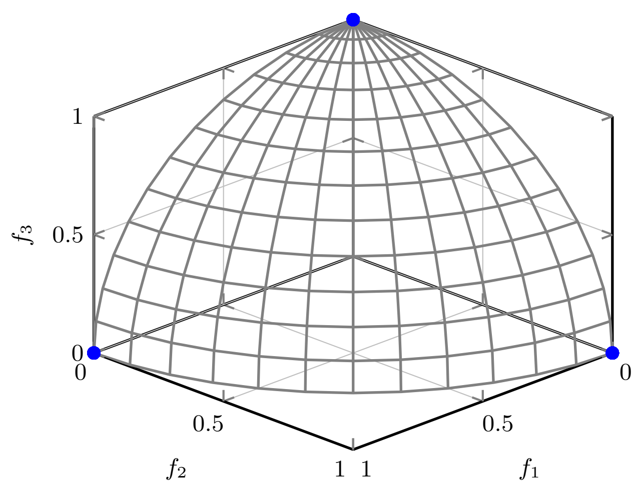


weighted sum

$$g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x})$$

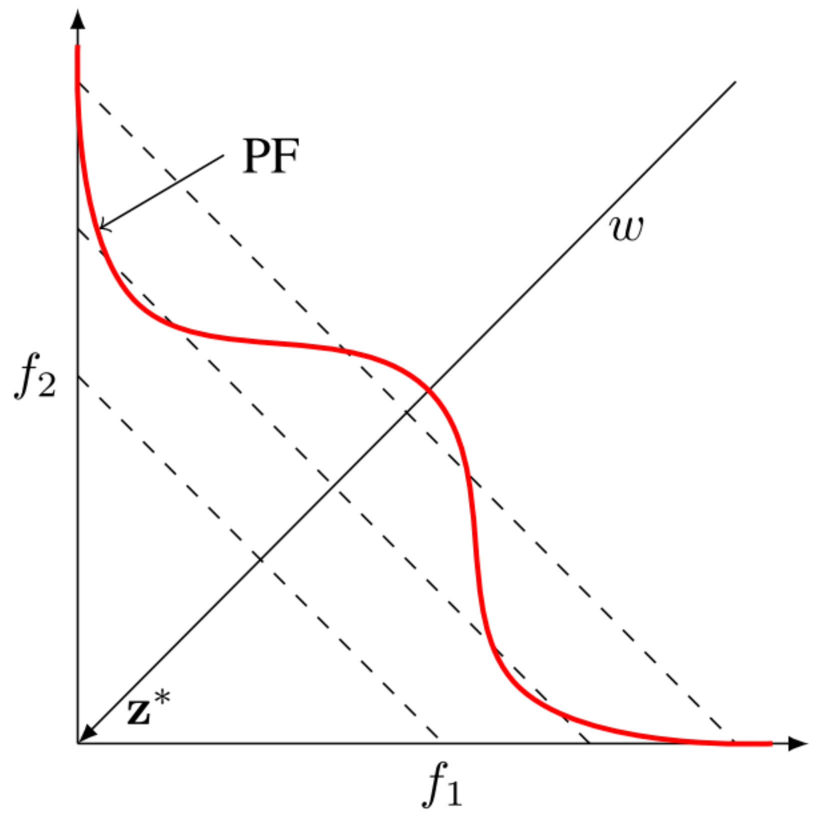
Drawbacks:

- only useful for convex PFs while not all Pareto-optimal solutions can be found if the PF is not convex.
- …



Revisit Weighted Sum

- Weighted sum



weighted sum

$$g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x})$$

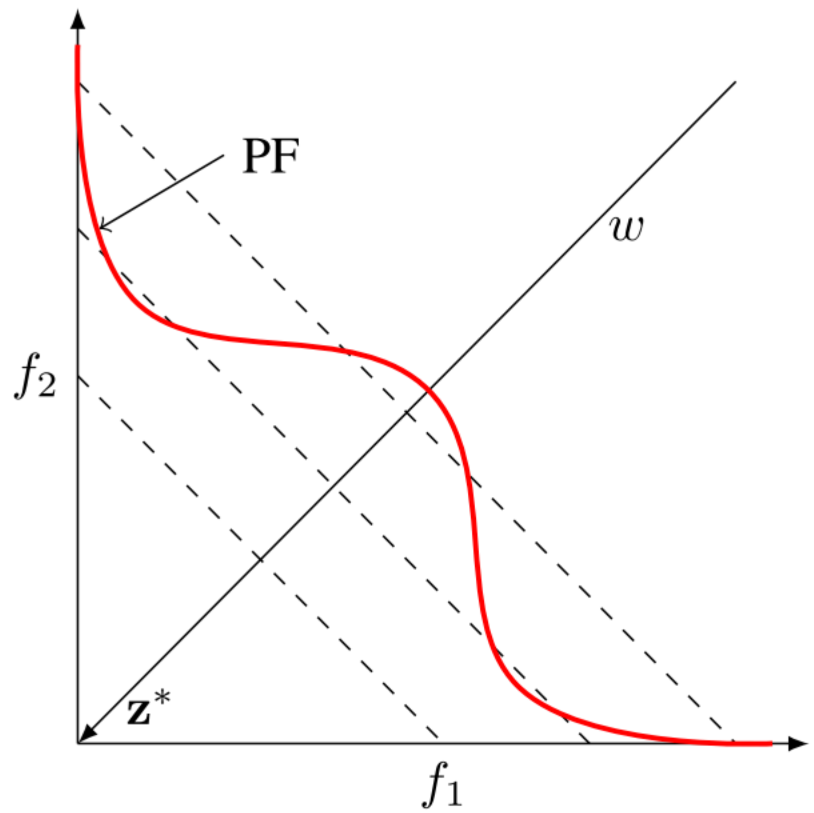
Is weighted sum really that bad?

- The superior region is constantly $1/2$, whereas it is $1/2^m$ for the Lp scalarizing
- MOEA/D with weighted sum have better convergence (given convex PF)

[12] R. Wang, et al., "Localized Weighted Sum Method for Many-Objective Optimization", IEEE Trans. Evol. Comput., 22(1): 3-18, 2018.

Revisit Weighted Sum

- Weighted sum

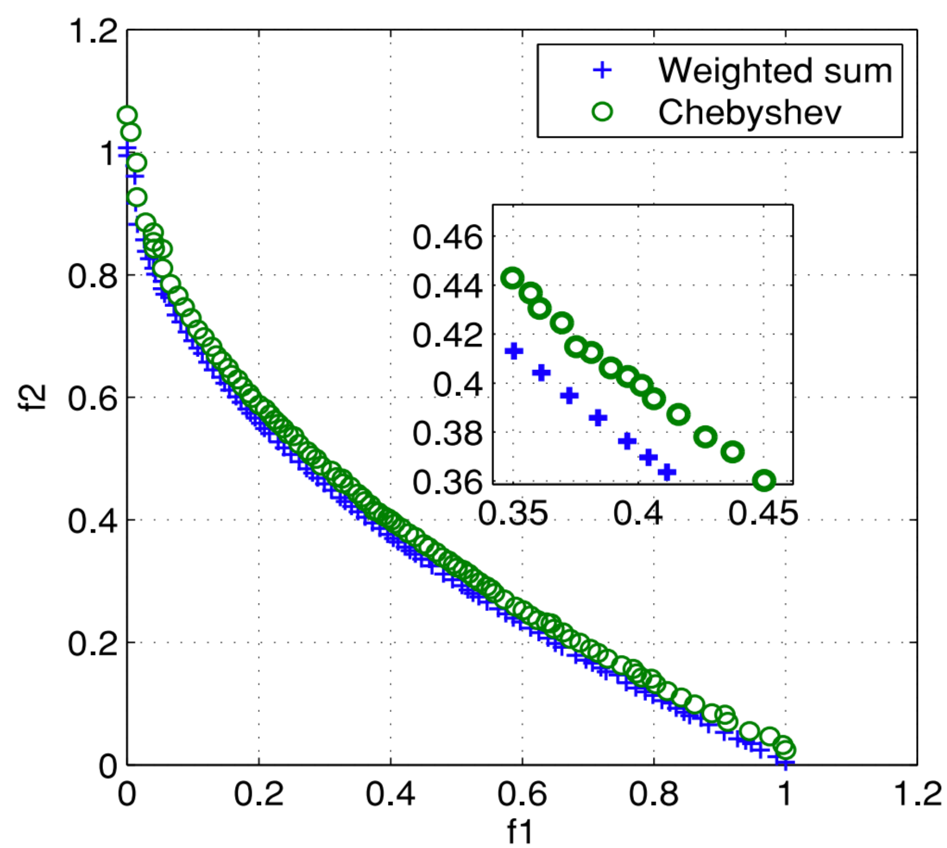


weighted sum

$$g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x})$$

Is weighted sum really that bad?

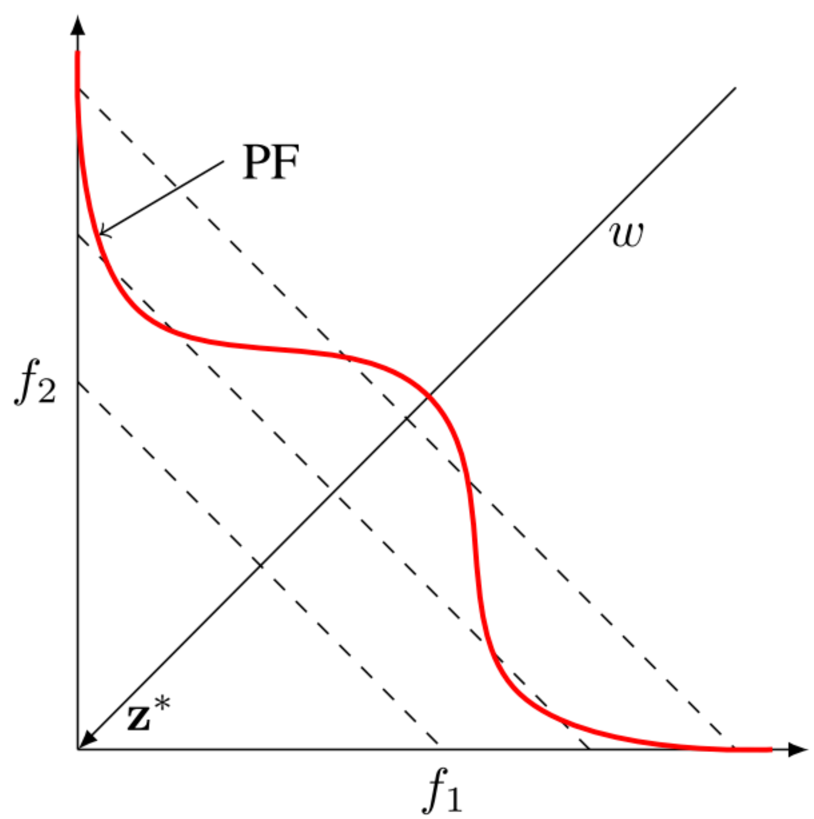
- The superior region is constantly $1/2$, whereas it is $1/2^m$ for the Lp scalarizing
- MOEA/D with weighted sum have better convergence (given convex PF)



[12] R. Wang, et al., "Localized Weighted Sum Method for Many-Objective Optimization", IEEE Trans. Evol. Comput., 22(1): 3-18, 2018.

Revisit Weighted Sum

- Weighted sum



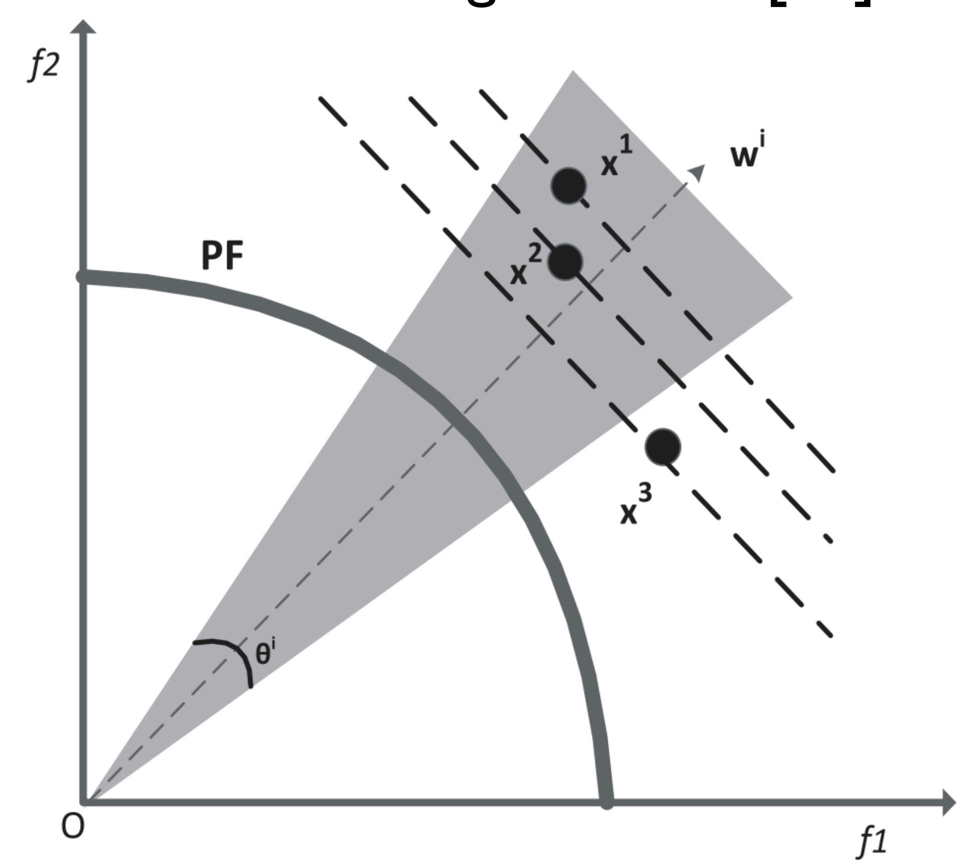
weighted sum

$$g(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x})$$

Is weighted sum really that bad?

- The superior region is constantly $1/2$, whereas it is $1/2^m$ for the Lp scalarizing
- MOEA/D with weighted sum have better convergence (given convex PF)

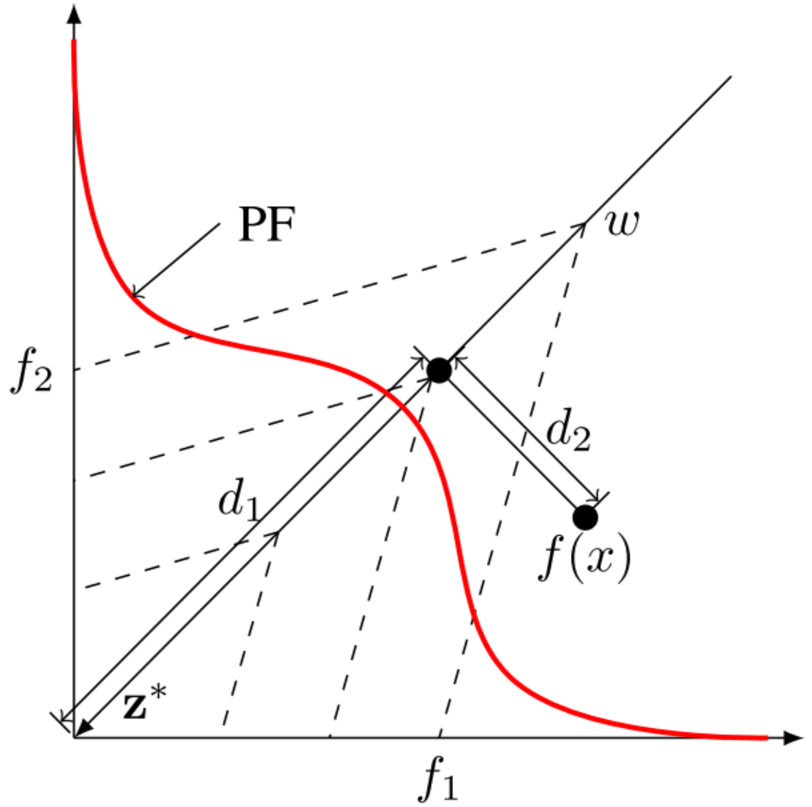
Localised weighted sum [12]



[12] R. Wang, et al., "Localized Weighted Sum Method for Many-Objective Optimization", IEEE Trans. Evol. Comput., 22(1): 3-18, 2018.

Boundary Intersection

- Penalty-Based Intersection (PBI) [13]



$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = d_1 + \theta d_2$$

$$d_1 = \frac{\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T\mathbf{w}\|}{\|\mathbf{w}\|}$$

$$d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^* + d_1\frac{\mathbf{w}}{\|\mathbf{w}\|})\|$$

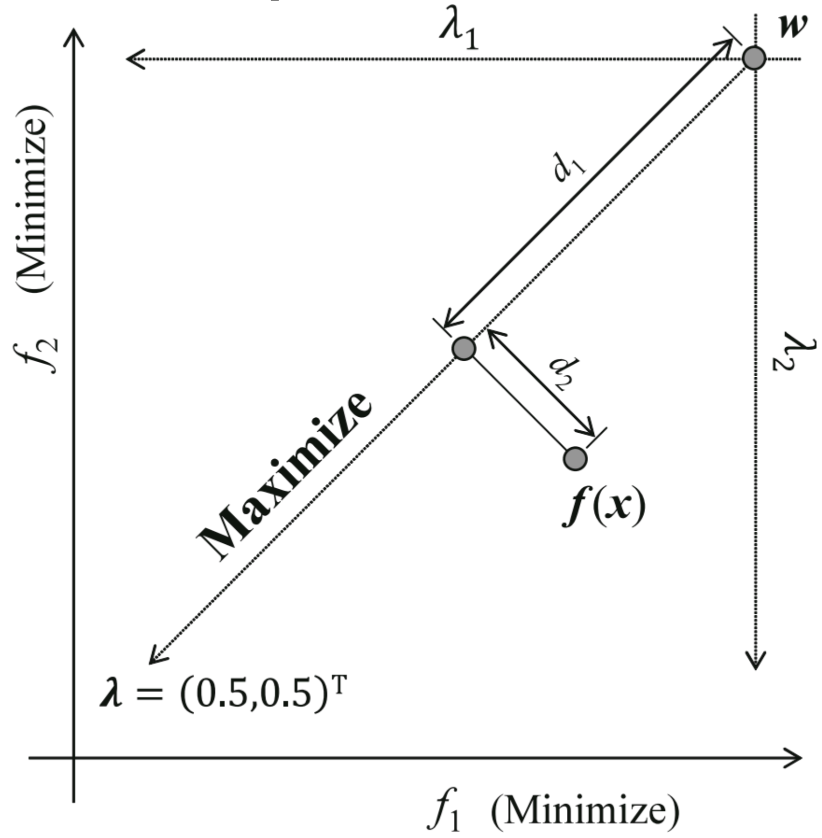
Characteristics:

- d_1 'measures' the <u>convergence</u>
  - ➡can be replaced by other measure [14]
- d_2 'measures' the <u>diversity</u>
  - ➡can be replaced by angle [14, 15]
- θ controls the contour and trade-offs

[13] **Q. Zhang** and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", IEEE Trans. Evol. Comput., 11(6): 712-731, 2007.
[14] R. Cheng, et al., "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization", IEEE Trans. Evol. Comput., 20(5): 773-791, 2016.
[15] Y. Xiang, et al., "A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization", IEEE Trans. Evol. Comput., 21(1): 131-152, 2017.
[16] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs", J. Heuristics, 21: 819-849, 2015.

Boundary Intersection

- Penalty-Based Intersection (PBI) [13]



$$g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = d_1 + \theta d_2$$

$$d_1 = \frac{\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T\mathbf{w}\|}{\|\mathbf{w}\|}$$

$$d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^* + d_1\frac{\mathbf{w}}{\|\mathbf{w}\|})\|$$

Inverted PBI [16]

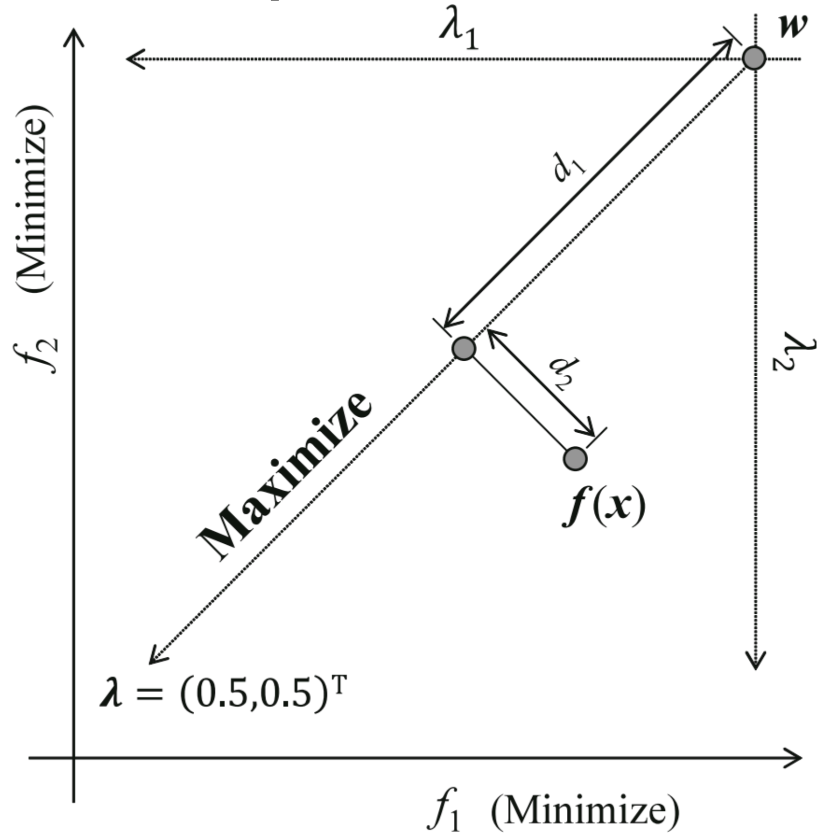
Characteristics:

- d_1 'measures' the convergence
  ➡can be replaced by other measure [14]
- d_2 'measures' the diversity
  ➡can be replaced by angle [14, 15]
- θ controls the contour and trade-offs

[13] **Q. Zhang** and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", IEEE Trans. Evol. Comput., 11(6): 712-731, 2007.
[14] R. Cheng, et al., "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization", IEEE Trans. Evol. Comput., 20(5): 773-791, 2016.
[15] Y. Xiang, et al., "A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization", IEEE Trans. Evol. Comput., 21(1): 131-152, 2017.
[16] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs", J. Heuristics, 21: 819-849, 2015.

Boundary Intersection

- Penalty-Based Intersection (PBI) [13]



$$g(\mathbf{x}|\mathbf{w}, \mathbf{z}^{nad}) = d_1 - \theta d_2$$

$$d_1 = \frac{\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^{nad})^T \mathbf{w}\|}{\|\mathbf{w}\|}$$

$$d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^{nad} + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|})\|$$

Inverted PBI [16]

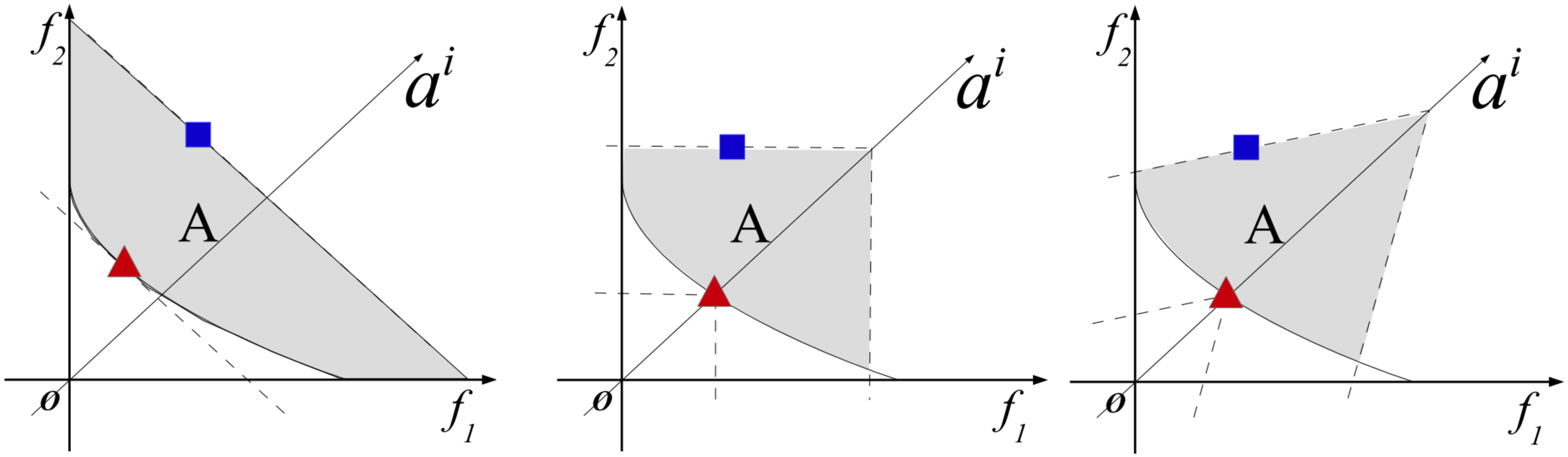
Characteristics:

- d_1 'measures' the convergence
  ➡can be replaced by other measure [14]
- d_2 'measures' the diversity
  ➡can be replaced by angle [14, 15]
- θ controls the contour and trade-offs

[13] **Q. Zhang** and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", IEEE Trans. Evol. Comput., 11(6): 712-731, 2007.
[14] R. Cheng, et al., "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization", IEEE Trans. Evol. Comput., 20(5): 773-791, 2016.
[15] Y. Xiang, et al., "A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization", IEEE Trans. Evol. Comput., 21(1): 131-152, 2017.
[16] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs", J. Heuristics, 21: 819-849, 2015.

Constrained Decomposition

- The improvement region of WS, TCH and PBI is too large
  - Gives a solution large chance to update many agents: hazard to diversity



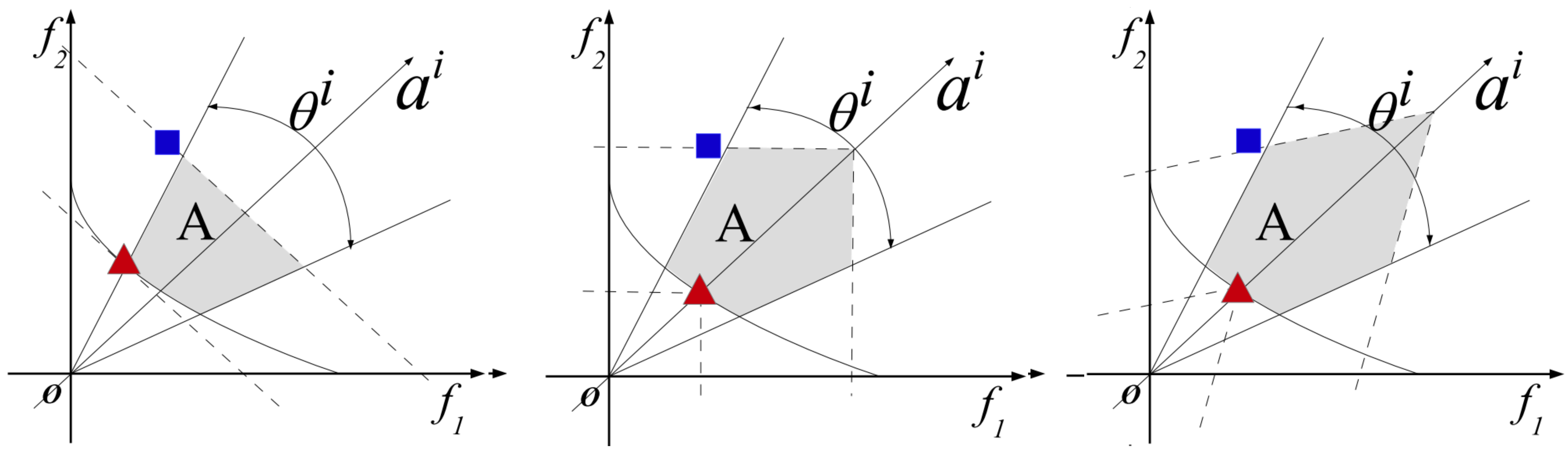
  - Add a <u>constraint</u> to the subproblem to reduce the improvement region [17]

$$\begin{array}{ll} \text{minimize} & g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) \\ \text{subject to} & \langle \mathbf{a}^i, \mathbf{F}(\mathbf{x}) - \mathbf{z}^* \rangle \le 0.5\theta^i \end{array}$$

[17] L. Wang, **Q. Zhang**, et al., "Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 20(3): 475-480, 2016.

Constrained Decomposition

- The improvement region of WS, TCH and PBI is too large
  - Gives a solution large chance to update many agents: hazard to diversity



  - Add a <u>constraint</u> to the subproblem to reduce the improvement region [17]

$$\begin{array}{ll} \text{minimize} & g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) \\ \text{subject to} & \langle \mathbf{a}^i, \mathbf{F}(\mathbf{x}) - \mathbf{z}^* \rangle \leq 0.5\theta^i \end{array}$$

[17] L. Wang, **Q. Zhang**, et al., "Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm", IEEE Trans. Evol. Comput., 20(3): 475-480, 2016.

Subproblem Can Be Multi-Objective …

- MOP to MOP (M2M)
  - Decompose a MOP into K $(K > 1)$ constrained MOPs [18].

$$
\begin{array}{ll}
\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} & \mathbf{x} \in \Omega
\end{array}
$$

[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

Subproblem Can Be Multi-Objective …

- MOP to MOP (M2M)
  - Decompose a MOP into K $(K > 1)$ constrained MOPs [18].

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}$$

$$\Rightarrow$$

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \\ & \mathbf{F}(\mathbf{x}) \in \Omega_k \end{array}$$

[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

Subproblem Can Be Multi-Objective …

- MOP to MOP (M2M)
  - Decompose a MOP into K $(K > 1)$ constrained MOPs [18].

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}$$

$$\Rightarrow$$

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \\ & \mathbf{F}(\mathbf{x}) \in \Omega_k \end{array}$$

$$\Omega_k = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^i \rangle \leq \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle \text{ for any } j = 1, \cdots, K\}$$

[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

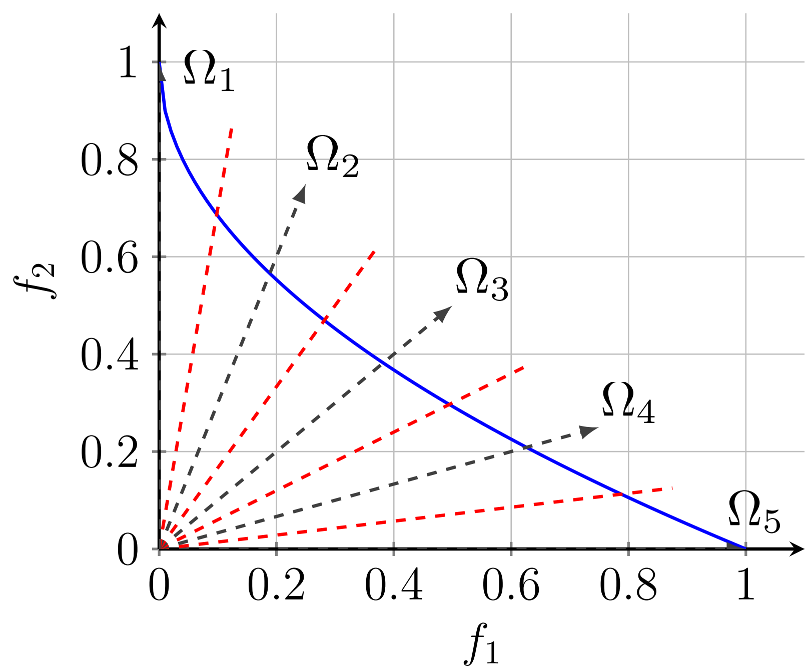
Subproblem Can Be Multi-Objective …

- MOP to MOP (M2M)
  - Decompose a MOP into K $(K > 1)$ constrained MOPs [18].

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}$$

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \\ & \mathbf{F}(\mathbf{x}) \in \Omega_k \end{array}$$

$$\Omega_k = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^i \rangle \le \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle \;\text{ for any }\; j = 1, \cdots, K\}$$



[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

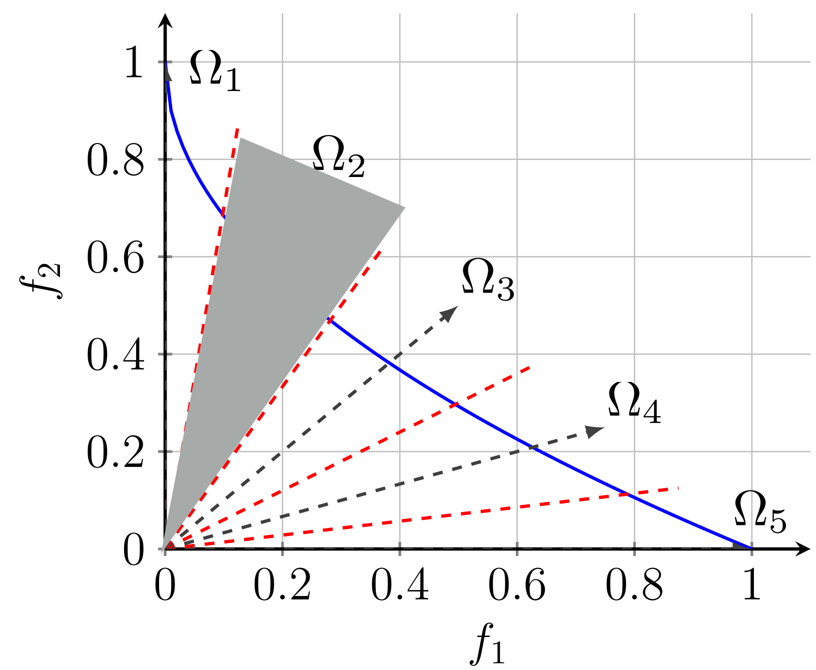
Subproblem Can Be Multi-Objective …

- MOP to MOP (M2M)
  - Decompose a MOP into K ($K > 1$) constrained MOPs [18].

$$\begin{array}{ll}\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega\end{array}$$

$$\begin{array}{ll}\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \\ & \mathbf{F}(\mathbf{x}) \in \Omega_k\end{array}$$

$$\Omega_k = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^i \rangle \leq \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle \ \text{ for any } \ j = 1, \cdots, K\}$$



[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

Subproblem Can Be Multi-Objective …

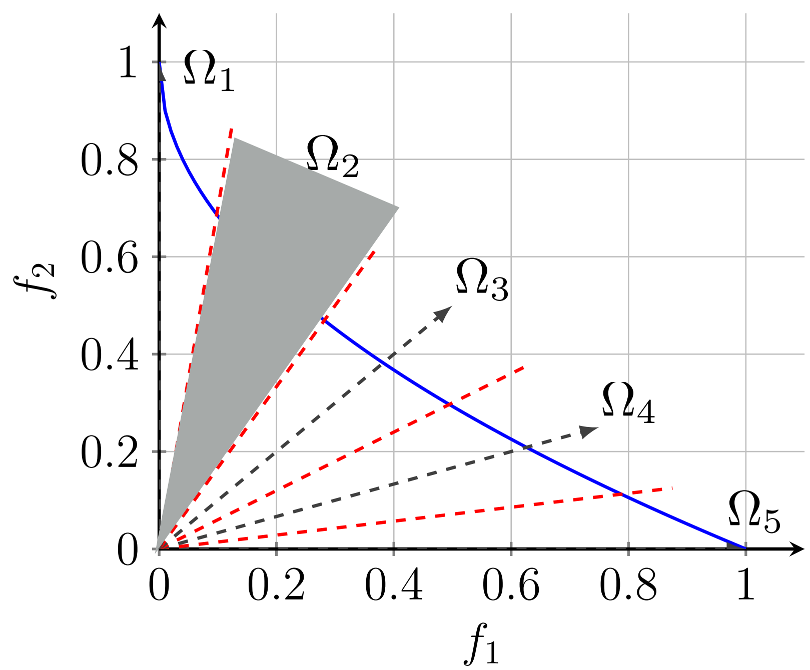
- MOP to MOP (M2M)
  - Decompose a MOP into K $(K > 1)$ constrained MOPs [18].

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}$$

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \\ & \mathbf{F}(\mathbf{x}) \in \Omega_k \end{array}$$

$$\Omega_k = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^i \rangle \leq \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle \ \text{ for any } \ j = 1, \cdots, K\}$$

  - Each agent is an EMO algorithm.



[18] H. Liu, F. Gu and **Q. Zhang**, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems", IEEE Trans. Evol. Comput., 18(3): 450-455, 2014.

Dynamic Resource Allocation

- Are all subproblems equally important?
  - Some regions in the PF/PS are easier than the others.
  - Different agents require different amounts of computational resources.
- Dynamic resource allocation (DRA) in MOEA/D [24]
  - Utility function to measure the likelihood of improvement
    - e.g. fitness improvement over ΔT

$$u^I = \frac{g^i(\mathbf{x^i_{t-\Delta T}}) - g^I(\mathbf{x}^i_t)}{g^i(\mathbf{x^i_{t-\Delta T}})}$$

  - Allocation mechanism
    - e.g. probability of improvement

$$p^i = \frac{u^i + \epsilon}{\max_{j=1,\cdots,N}\{u^j\} + \epsilon}$$

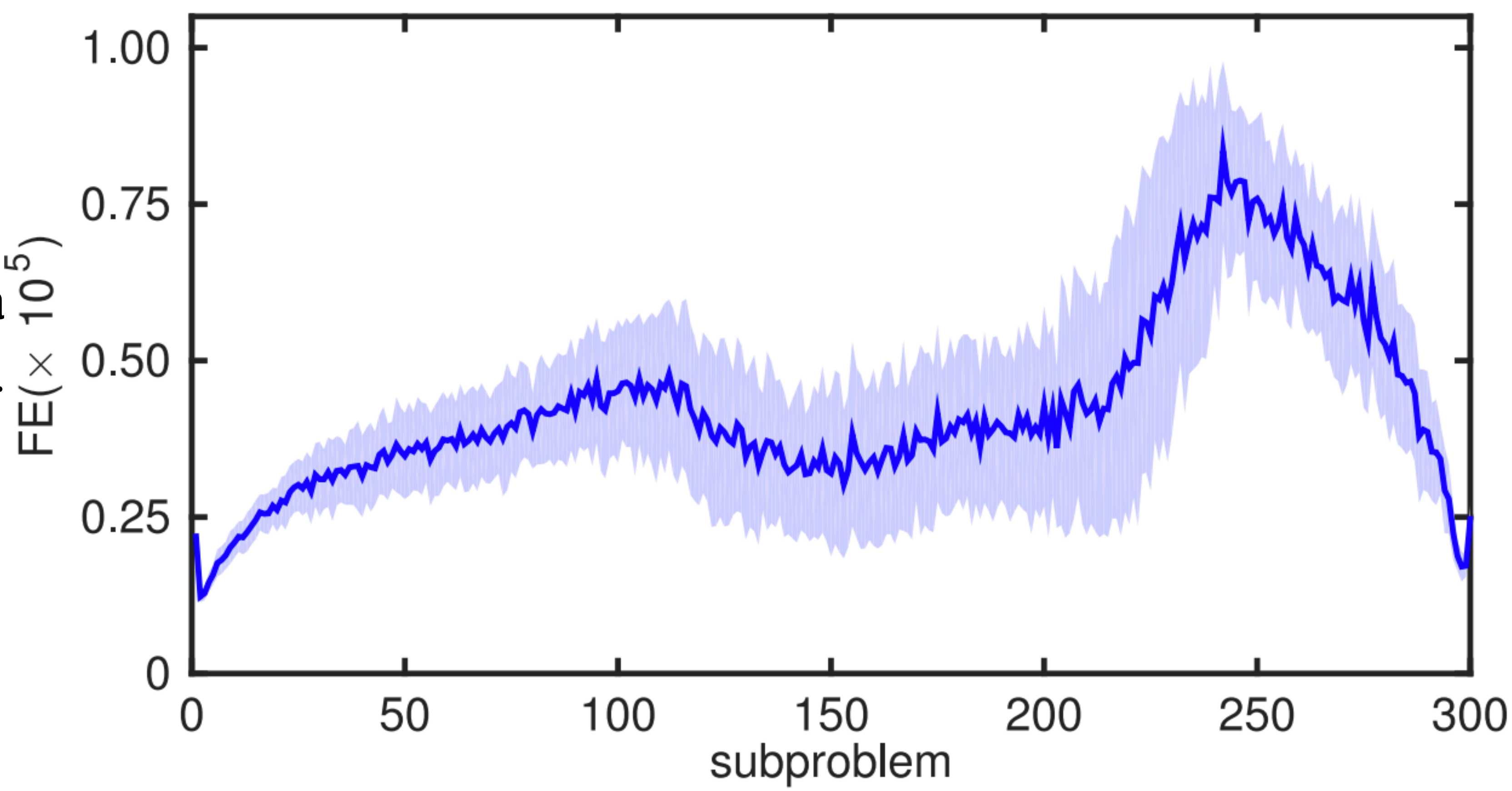
[19] A. Zhou and **Q. Zhang**, "Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms", IEEE TEVC, 20(1): 52-64, 2016.

Dynamic Resource Allocation

- Are all subproblems equally important?
  - Some regio
  - Different ag ces.
- Dynamic re
  - Utility funct
    - e.g. fitness improvement over
  - Alloca
    - e.g.

$$f_1(\mathbf{x}) = x1 + \frac{2}{|J_1|}\sum_{j\in J_1}(x_j - \sin(6\pi x_1 + \frac{j}{n}\pi))$$

$$f_2(\mathbf{x}) = 1 - \sqrt{x1} + \frac{2}{|J_2|}\sum_{j\in J_2}(x_j - \sin(6\pi x_1 + \frac{j}{n}\pi))$$



[19] A. Zhou and **Q. Zhang**, "Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms", IEEE TEVC, 20(1): 52-64, 2016.

Dynamic Resource Allocation

- Are all subproblems equally important?
  - Some regions in the PF/PS are easier than the others.
  - Different agents require different amounts of computational resources.
- Dynamic resource allocation (DRA) in MOEA/D [24]
  - Utility function to measure the likelihood of improvement
    - e.g. fitness improvement over ΔT

$$u^I = \frac{g^i(\mathbf{x^i_{t-\Delta T}}) - g^I(\mathbf{x}^i_t)}{g^i(\mathbf{x^i_{t-\Delta T}})}$$

  - Allocation mechanism
    - e.g. probability of improvement

$$p^i = \frac{u^i + \epsilon}{\max_{j=1,\cdots,N}\{u^j\} + \epsilon}$$

GECCO

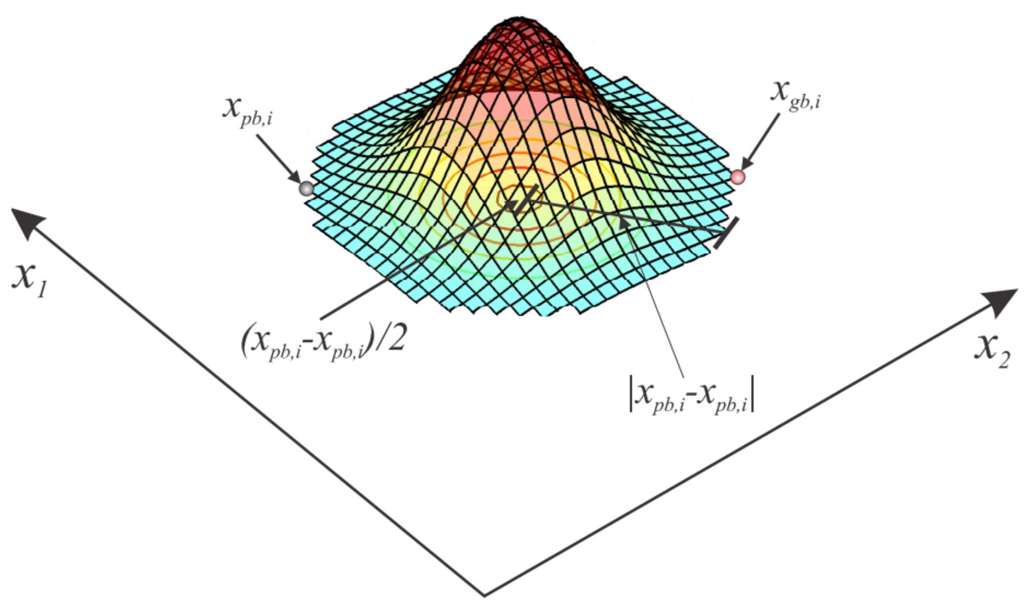
[19] A. Zhou and **Q. Zhang**, "Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms", IEEE TEVC, 20(1): 52-64, 2016.

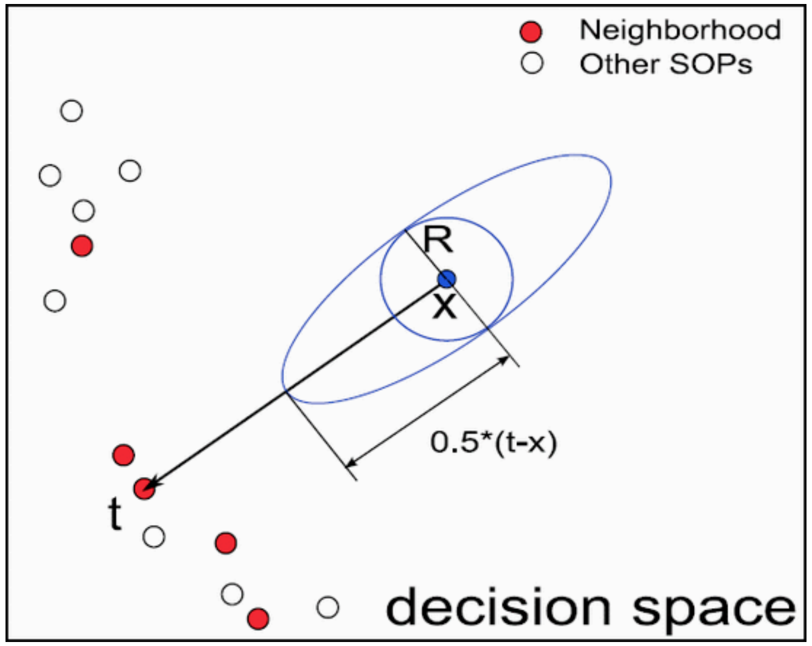
Outline

- **Part I: Basics**
  - Basic Concepts
  - Simple MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
  - Resources
  - Future Directions

Search Methods

- Offspring reproduction in MOEA/D
  - Neighbourhood defines where to find mating parents
  - Any genetic operator can be used
    - GA [13], DE [20], PSO [21], guided mutation [22], …

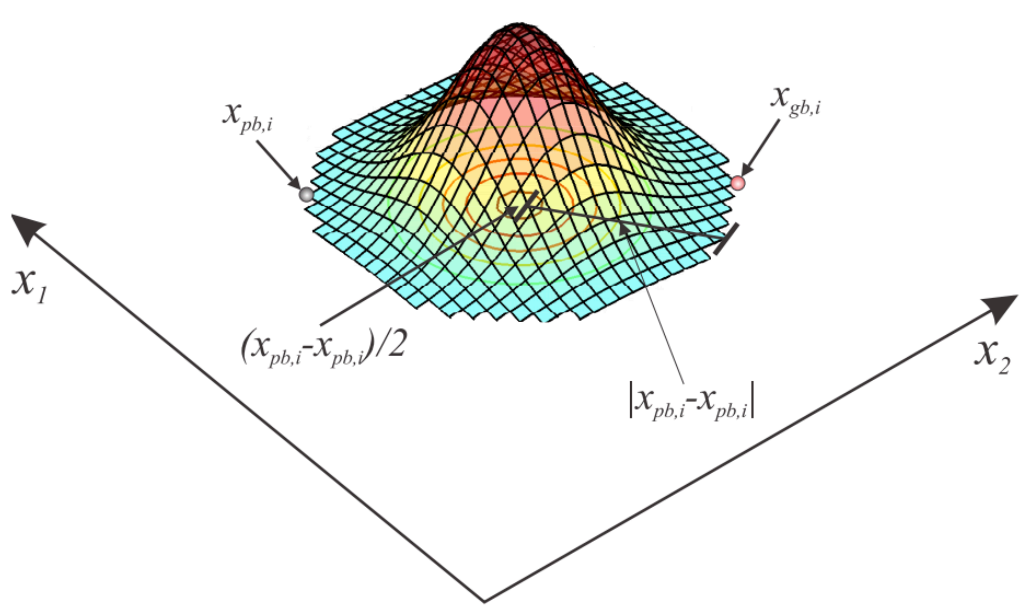


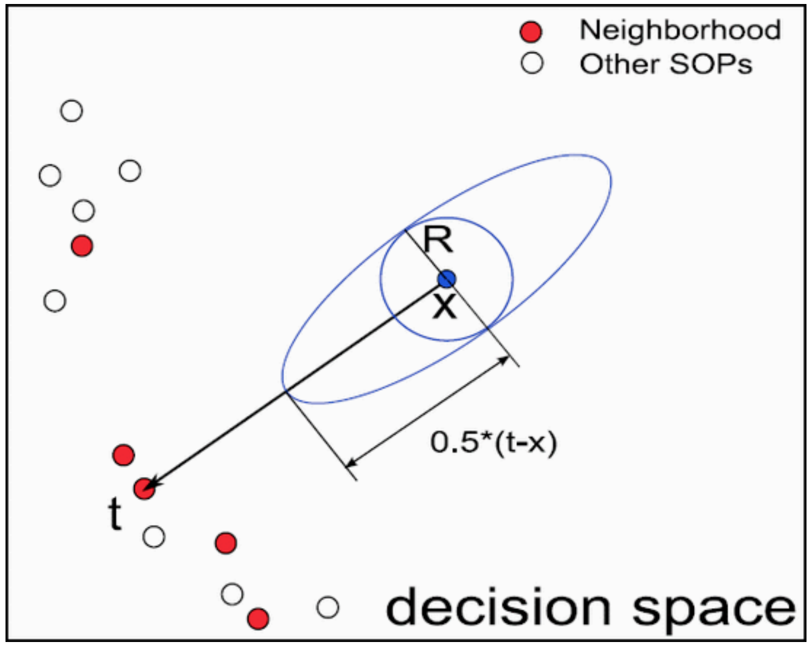


[13] **Q. Zhang** and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", IEEE Trans. Evol. Comput., 11(6): 712-731, 2007.
[20] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.
[21] S. Martínez, et al., "A multi-objective PSO based on decomposition, in GECCO 2011.
[22] C. Chen, et al., "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization", CEC 2009

Search Methods

- Offspring reproduction in MOEA/D
  - Neighbourhood defines where to find mating parents
  - Any genetic operator can be used
  - Any local search can be used
    - simulated annealing [23], interpolation [24], tabu search [25], GRASP [26], Nelder-Mead [27], …





[23] H. Li, et al., "An adaptive evolutionary multi-objective approach based on simulated annealing", Evol. Comput. 19(4): 561-595, 2011.

[24] K. Sindhya, "A new hybrid mutation operator for multiobjective optimization with differential evolution", Soft Comput., 15:2041–2055, 2011.

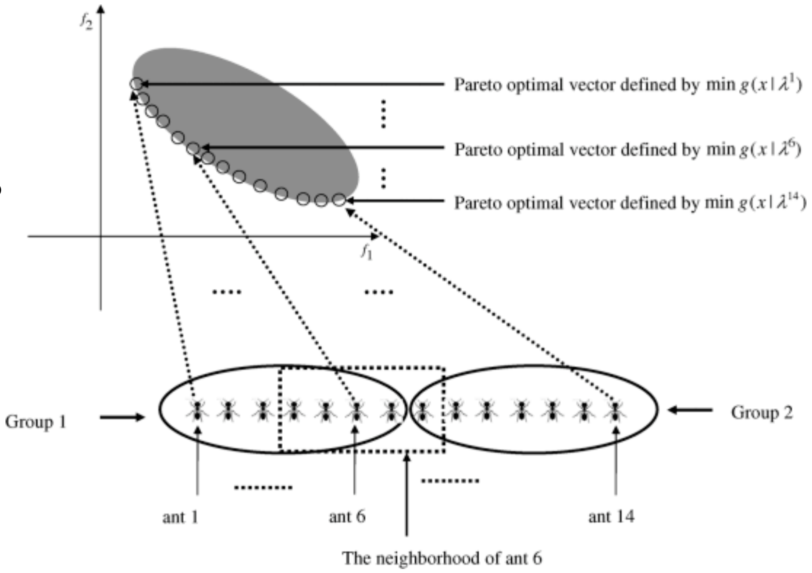
[25] A. Alhindi and **Q. Zhang**, "Hybridisation of decomposition and GRASP for combinatorial multiobjective optimisation", UKCI 2014.

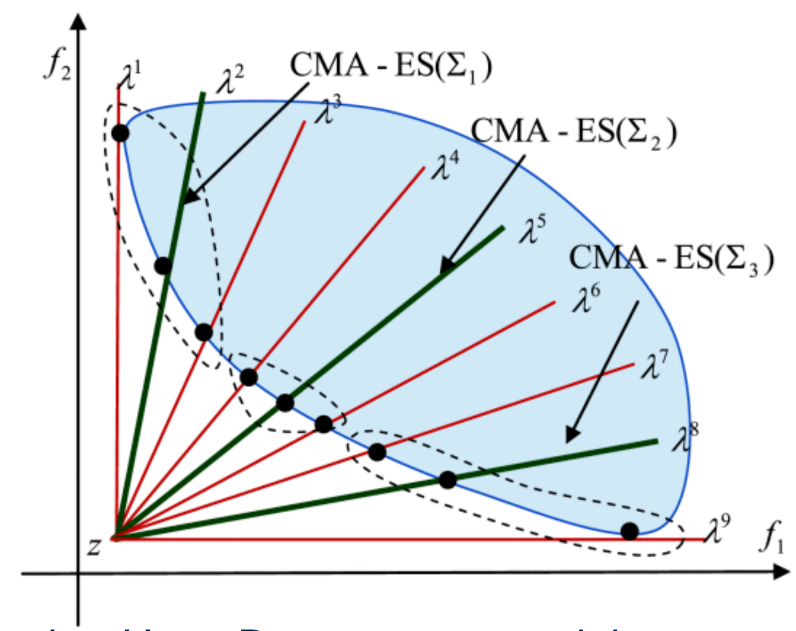
[26] A. Alhindi and **Q. Zhang**, "MOEA/D with Tabu Search for multiobjective permutation flow shop scheduling problems", CEC 2014.

[27] H. Zhang, et al., "Accelerating MOEA/D by Nelder-Mead method", CEC 2017.

Search Methods

- Offspring reproduction in MOEA/D
  - Neighbourhood defines where to find mating parents
  - Any genetic operator can be used
  - Any local search can be used
  - Probabilistic model can be used
    - Memory
      - ➡ Each agent records historical information, i.e. elites
    - Model building and solution construction
      - ➡ Each agent can build 'local model', e.g. ACO [28], EDA [29], cross entropy [30], graphical model [31], CMA-ES [32], based on memory of itself and its neighbour
      - ➡ New solutions are sampled from these models
      - ➡ NOTE: too many models may be too expensive
    - Memory update
      - ➡ Offspring update each agent's and its neighbour's memory





[28] L. Ke, **Q. Zhang**, et al., "MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony", IEEE Trans. Cybern., 43(6): 1845-1859, 2013.

[29] A. Zhou, **Q. Zhang**, et al., "A Decomposition based Estimation of Distribution Algorithm for Multiobjective Traveling Salesman Problems", Computers & Mathematics with Applications, 66(10): 1857-1868, 2013.

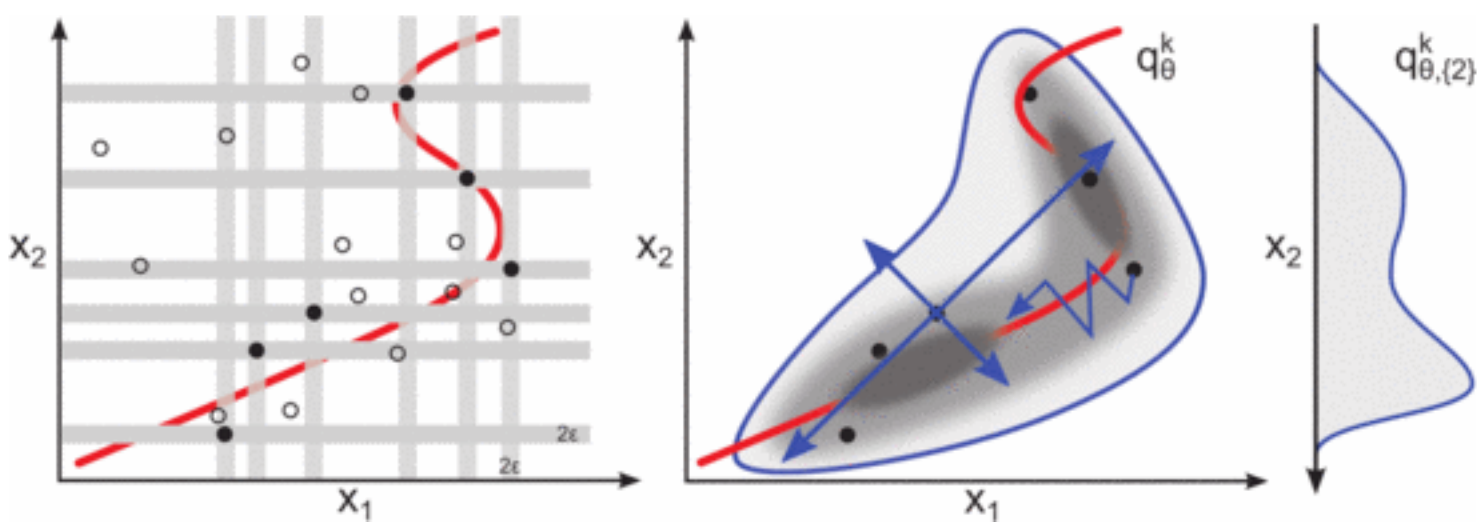
[30] I. Giagkiozis, et al., "Generalized decomposition and cross entropy methods for many-objective optimization", Inf. Sci., 282: 363-387, 2014.

[31] M. de Souza, et al., "MOEA/D-GM: Using probabilistic graphical models in MOEA/D for solving combinatorial optimization problems", arXiv:1511.05625, 2015.

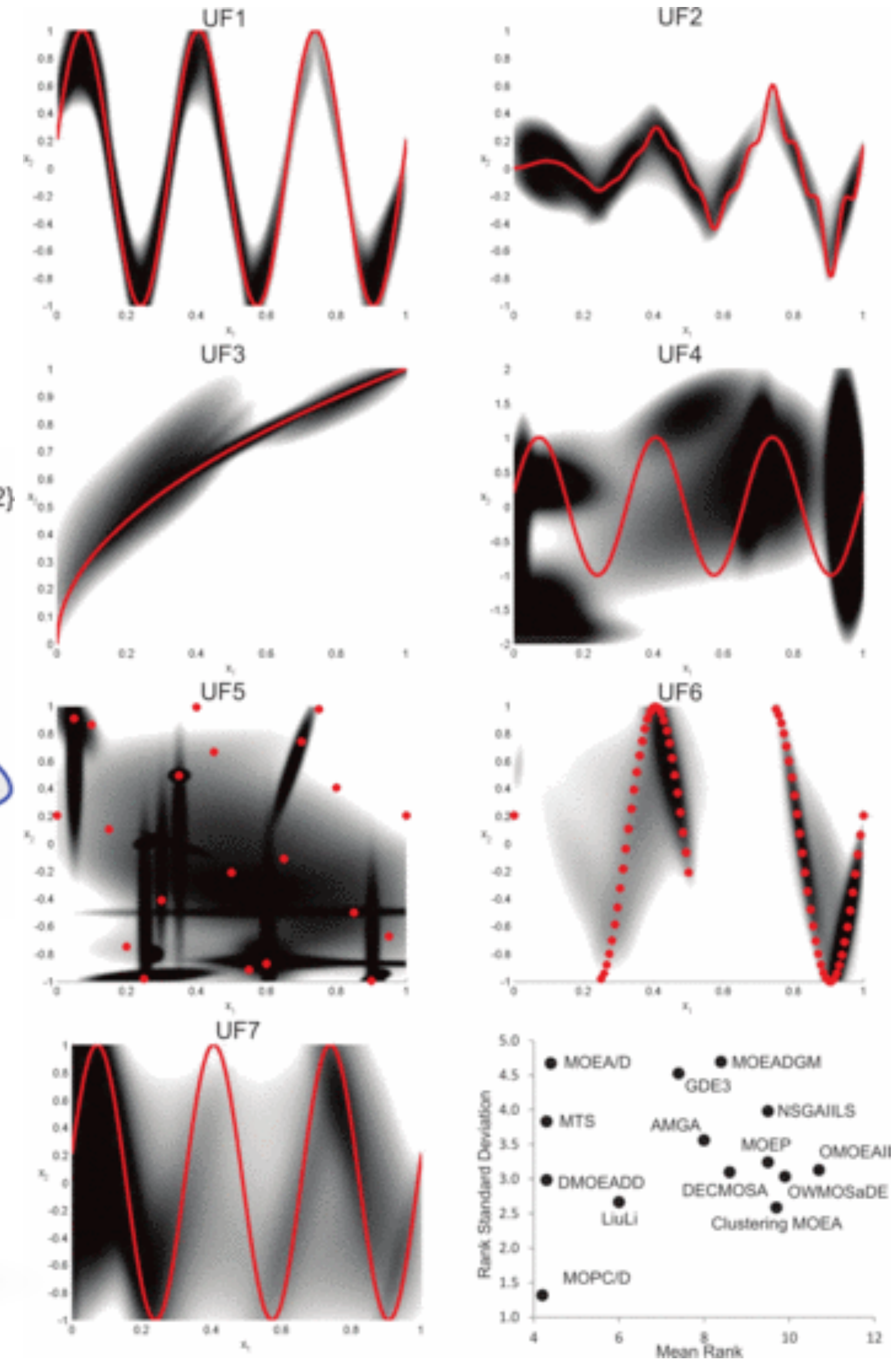
[32] H. Li and **Q. Zhang**, "Biased Multiobjective Optimization and Decomposition Algorithm", IEEE Trans. Cybern., 47(1): 52-66, 2016.

Search Methods (cont.)

- Using Probability Collective in MOEA/D
  - Instead of a point-based search, probability collective aims to fit a probability distribution highly peaked around the neighbourhood of PS



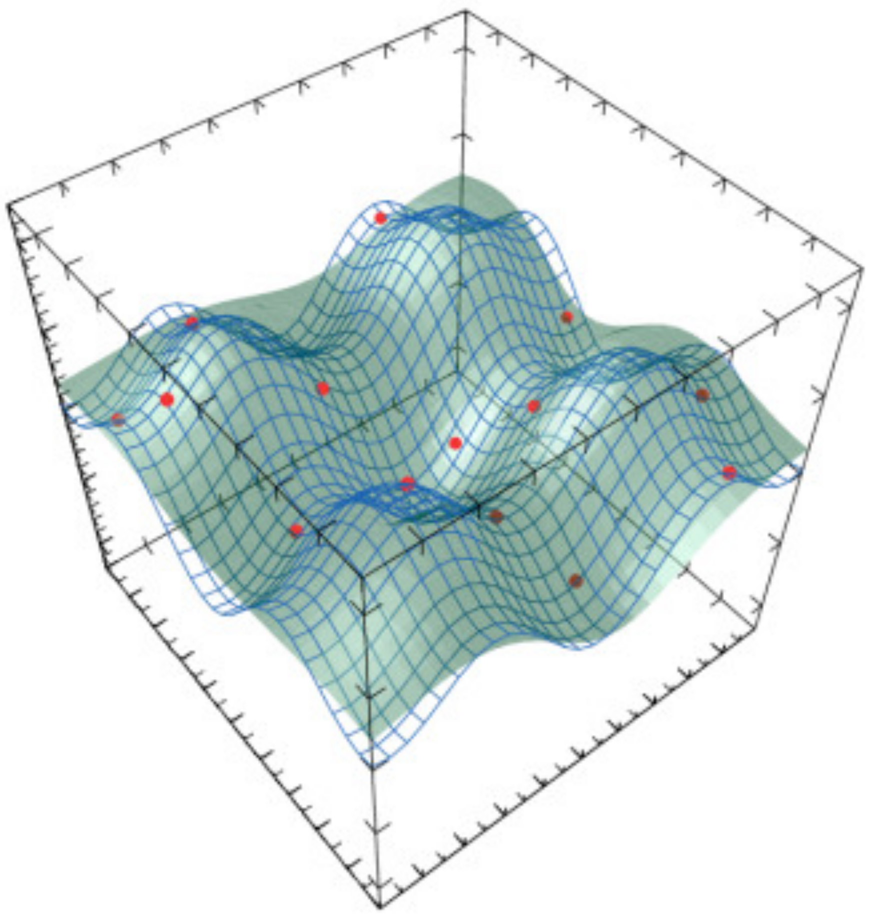
- Fit a Gaussian mixture model using solutions associated with each subproblem
- Search is based one sampling or local search upon the fitted model



[33] D. Morgan, et al., "MOPC/D: A new probability collectives algorithm for multiobjective optimisation", MCDM'13, 17-24, 2013

Search Methods (cont.)

- Expensive optimisation
  - Building surrogate model for expensive objective function
    - e.g. Gaussian processes (Kriging) [34, 35], RBF [36], …



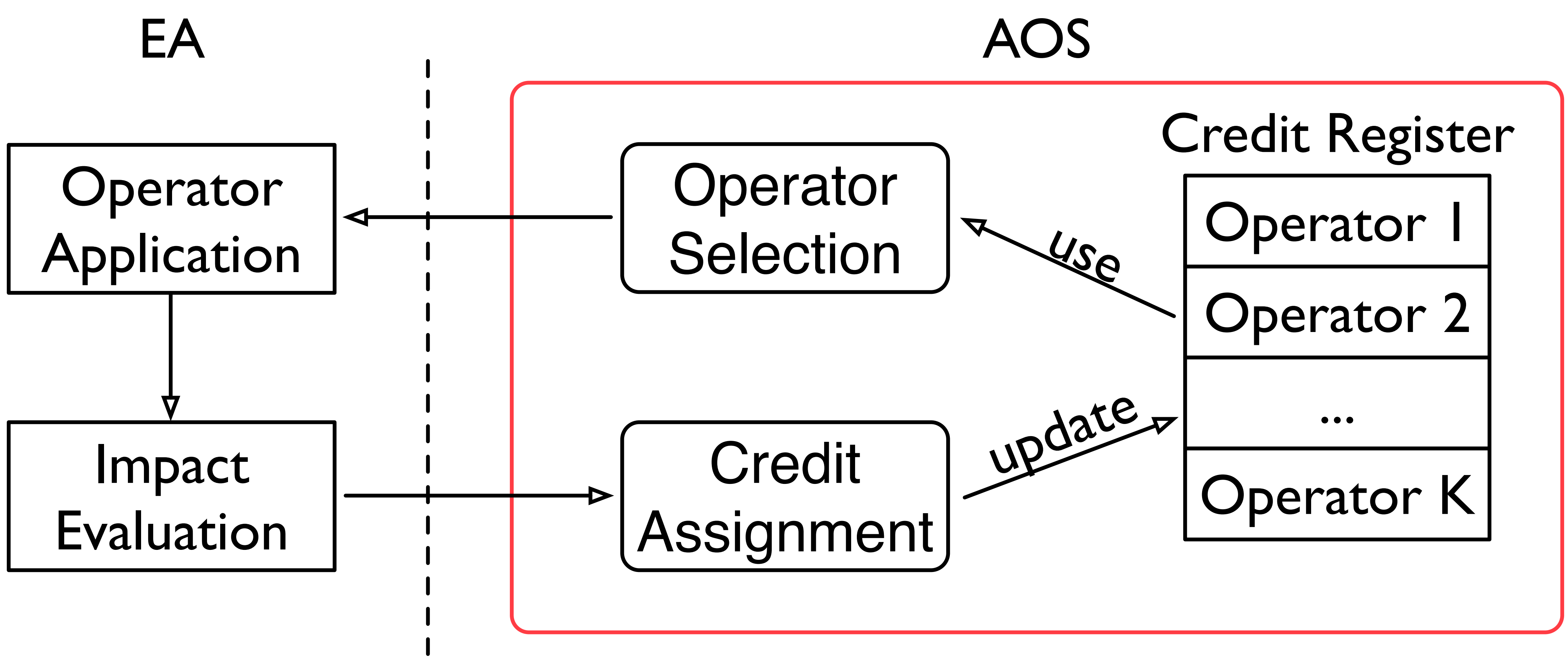
[34] **Q. Zhang**, et al., "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model", IEEE Trans. Evol. Comput., 14(3): 456-474, 2010.

[35] T. Chugh, et al., "A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization", 22(1): 129-142, 2018.

[36] S. Martínez, et al., "MOEA/D assisted by RBF Networks for Expensive Multi-Objective Optimization Problems", GECCO 2013.

Search Methods (cont.)

- Adaptive operator selection as a multi-armed bandits [37]
  - Strike the balance between the exploration and exploitation
    - Exploration: acquire new information (diversity)
    - Exploitation: capitalise on the available knowledge (convergence)



[37] K. Li, et al., "Adaptive operator selection with bandits for multiobjective evolutionary algorithm based on decomposition", IEEE Trans. Evol. Comput., 18(1): 114-130, 2014.

Outline

Part I: Basics

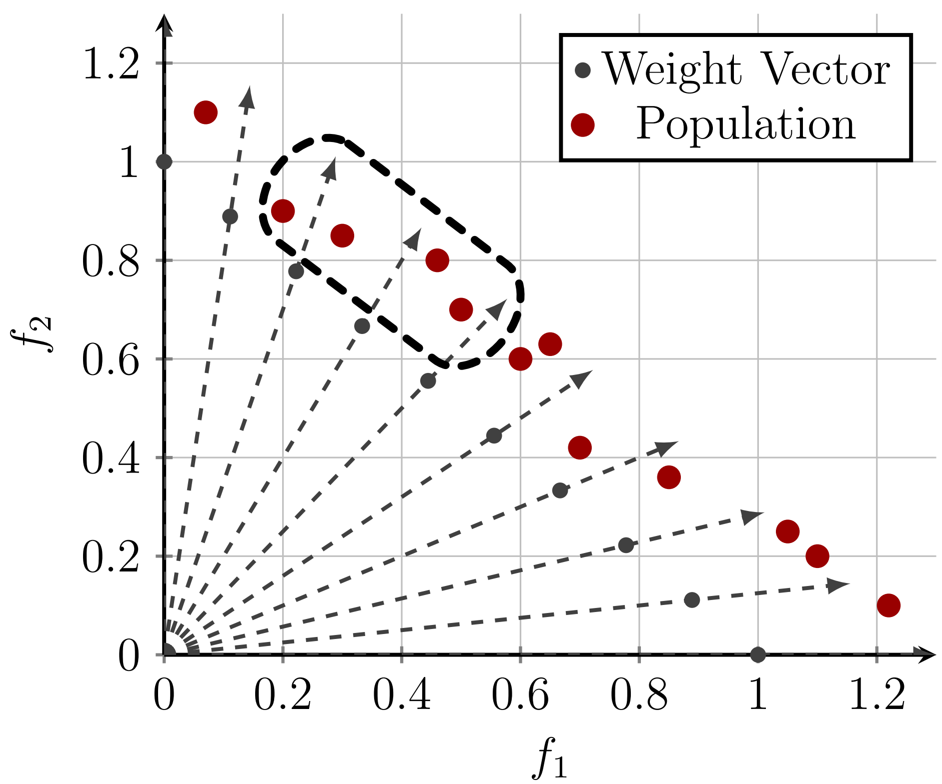
- Basic Concepts
- Simple MOEA/D
- A Simple Variant

Part II: Advanced Topics

- Current Developments
  - Decomposition methods
  - Search methods
  - Collaboration
    - Mating selection
    - Replacement
- Resources
- Future Directions

Mating Selection

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

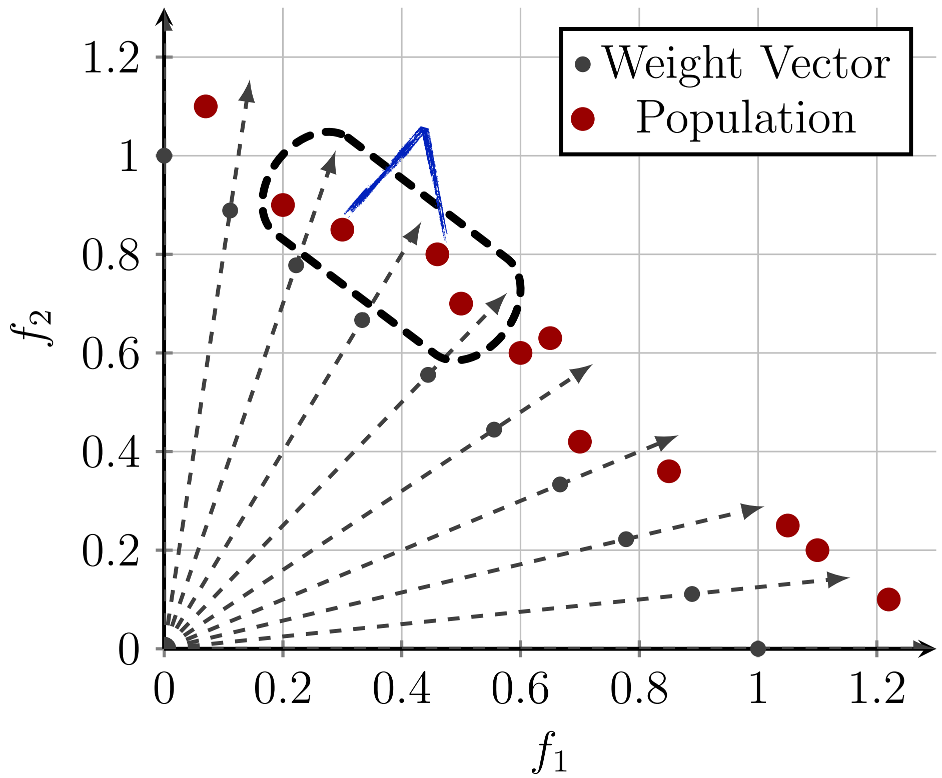


- Focusing on the neighbourhood is too much exploited
- Give some chance to explore in the whole population [38]

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Mating Selection

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

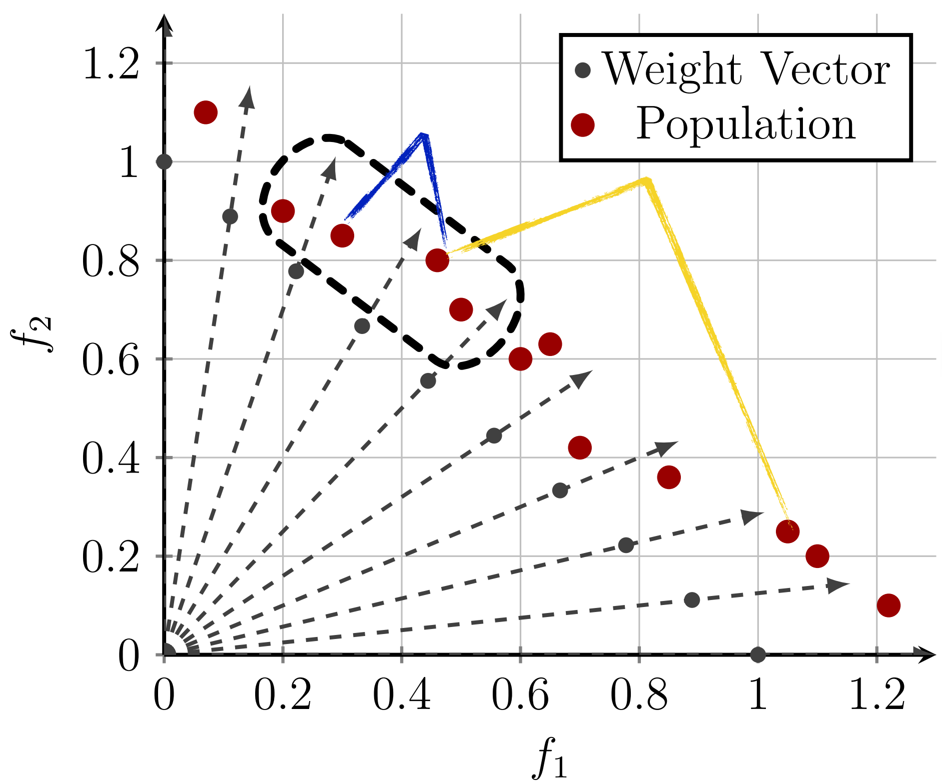


- Focusing on the neighbourhood is too much exploited
- Give some chance to explore in the whole population [38]

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Mating Selection

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

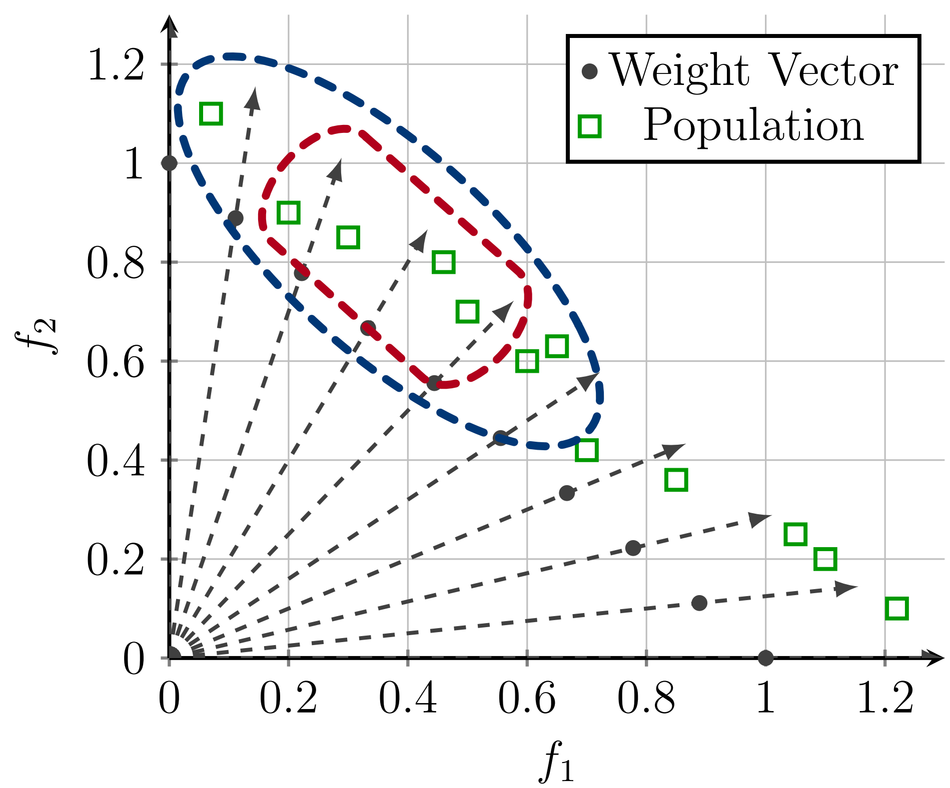


- Focusing on the neighbourhood is too much exploited
- Give some chance to explore in the whole population [38]

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Mating Selection (cont.)

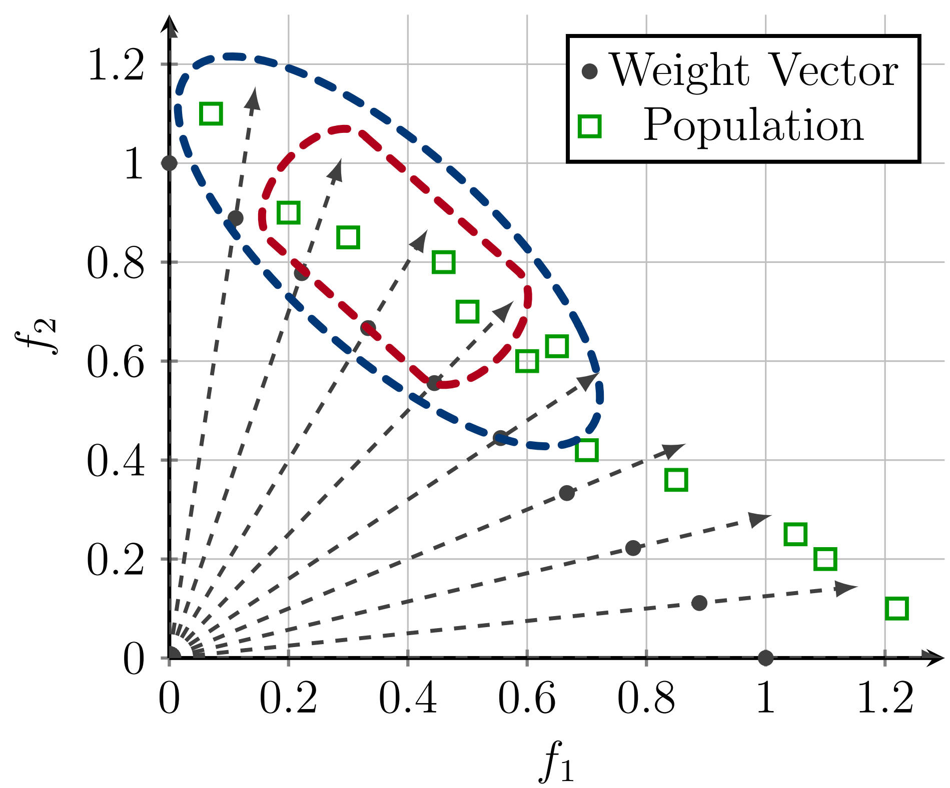
- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)



[39] S. Zhao, **Q. Zhang**, et al., "Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes", IEEE Trans. Evol. Comput., 16(3): 442-446, 2013.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)



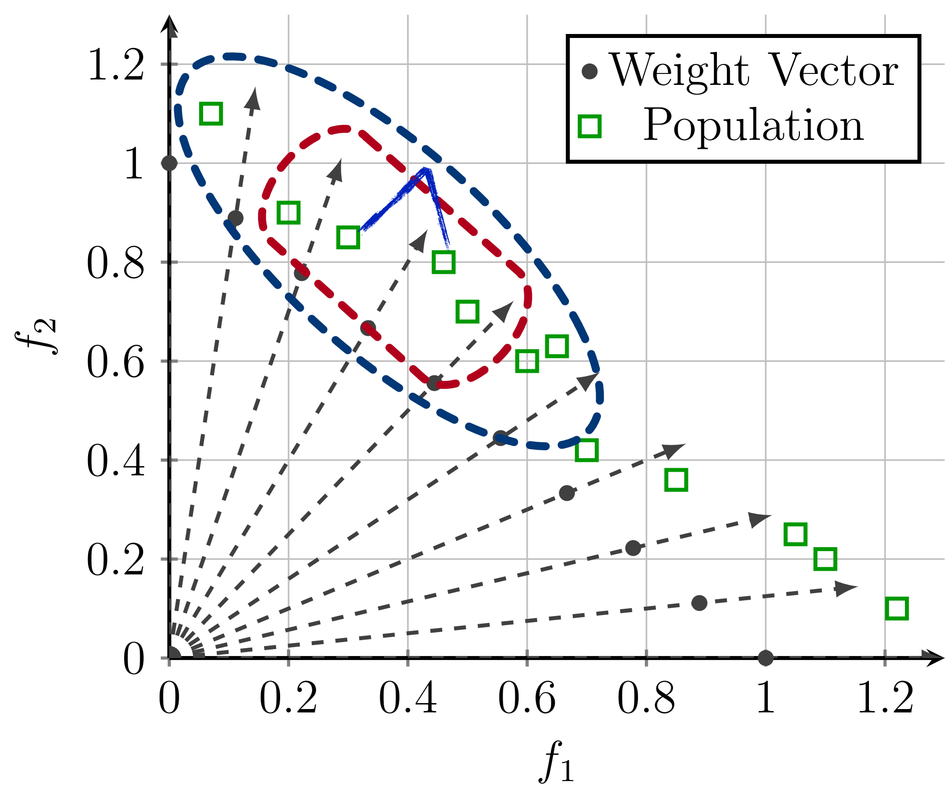
Effects of neighbourhood size (NS)

- Large neighbourhood makes the search globally
- Small neighbourhood encourages local search

[39] S. Zhao, **Q. Zhang**, et al., "Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes", IEEE Trans. Evol. Comput., 16(3): 442-446, 2013.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

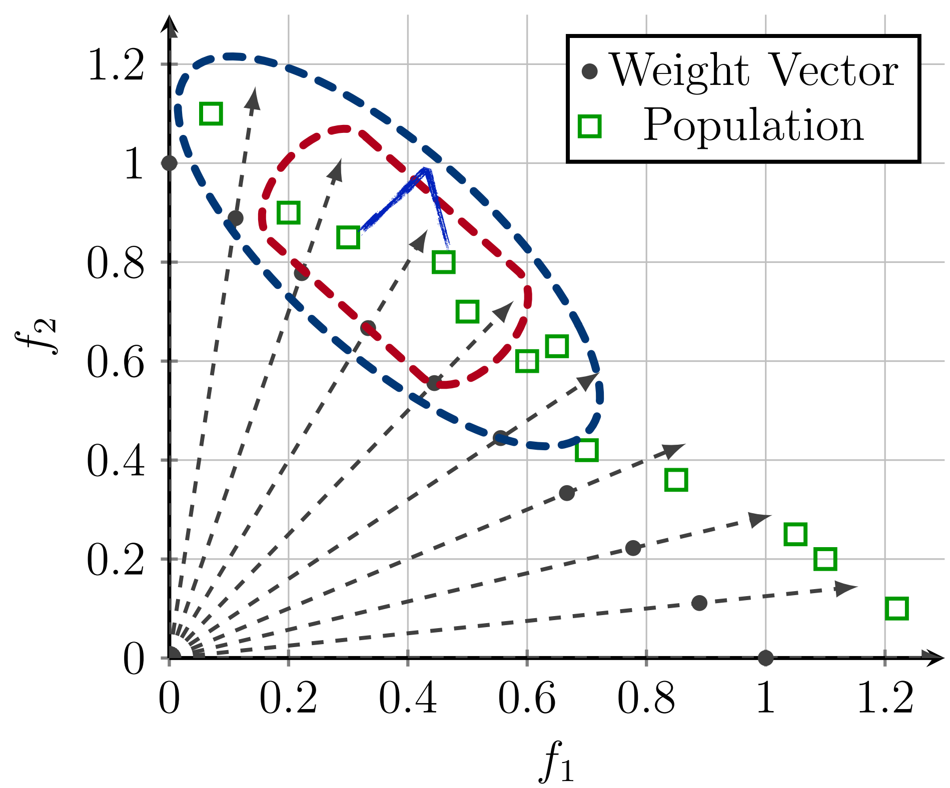


Effects of neighbourhood size (NS)
- Large neighbourhood makes the search globally
- Small neighbourhood encourages local search

[39] S. Zhao, **Q. Zhang**, et al., "Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes", IEEE Trans. Evol. Comput., 16(3): 442-446, 2013.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)



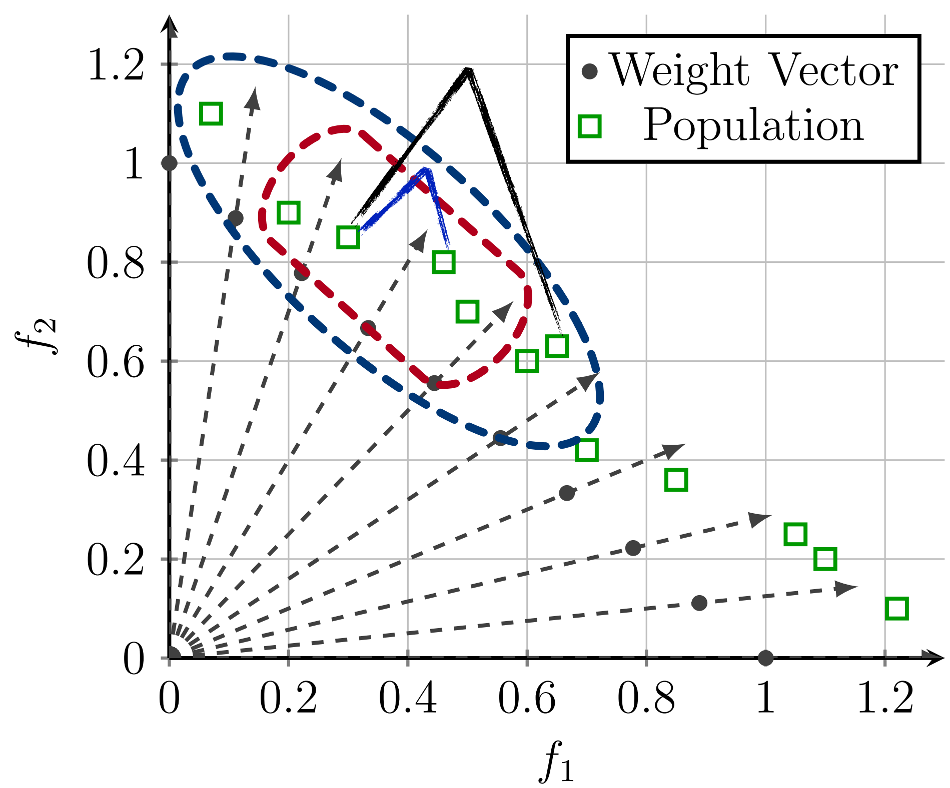
Effects of neighbourhood size (NS)
- Large neighbourhood makes the search globally
- Small neighbourhood encourages local search

Build an ensemble of neighbourhood sizes and chooses the appropriate one based on their historical performance. [39]

[39] S. Zhao, **Q. Zhang**, et al., "Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes", IEEE Trans. Evol. Comput., 16(3): 442-446, 2013.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)



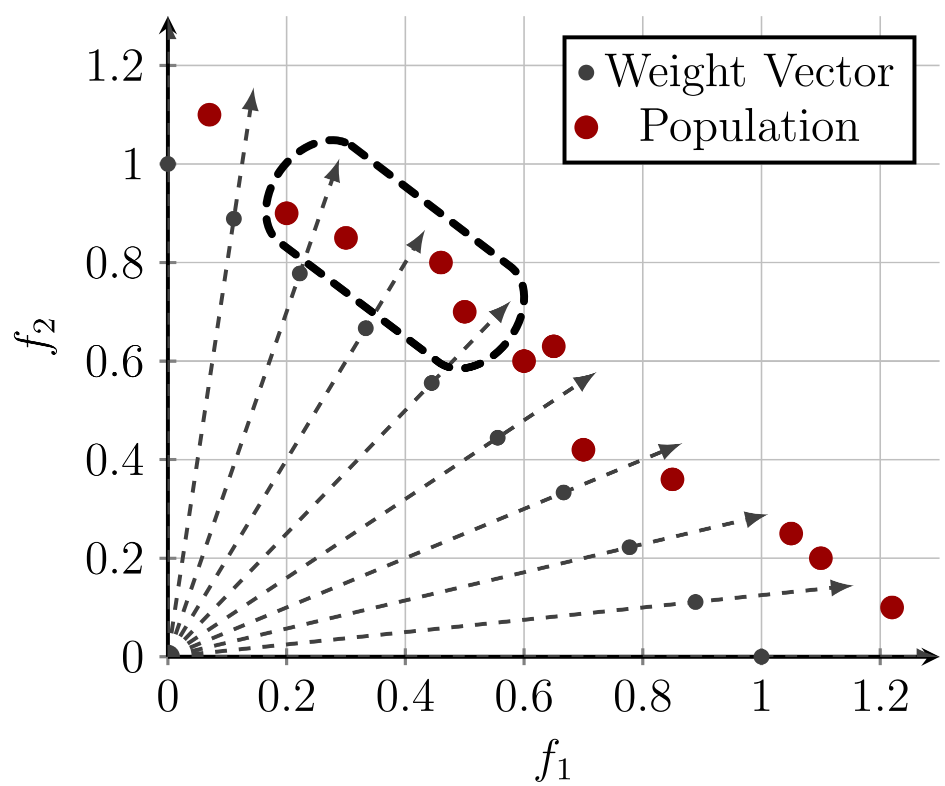
Effects of neighbourhood size (NS)
- Large neighbourhood makes the search globally
- Small neighbourhood encourages local search

Build an ensemble of neighbourhood sizes and chooses the appropriate one based on their historical performance. [39]

[39] S. Zhao, **Q. Zhang**, et al., "Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes", IEEE Trans. Evol. Comput., 16(3): 442-446, 2013.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

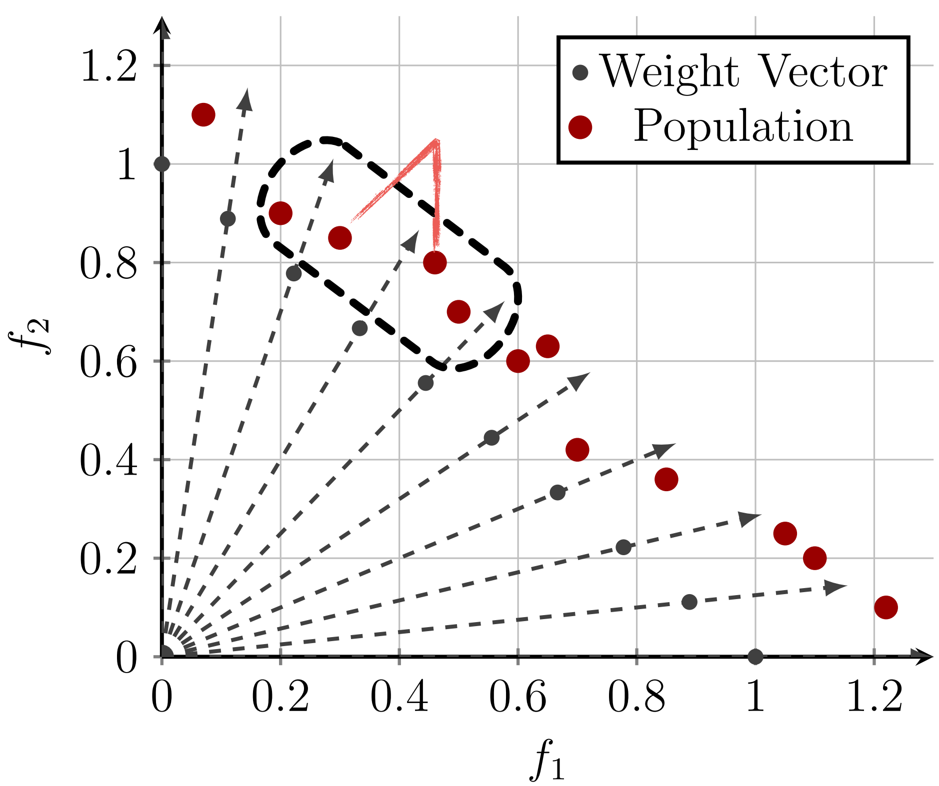


Take crowdedness into consideration [40]
- Compute the niche count of each solution within agent i's neighbour
- Select mating parents from outside of the neighbour if solutions are overly crowded

[40] S. Jiang, et al., "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts", IEEE Trans. Cybern, 46(2): 421-437, 2016.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - <u>Assumption</u>: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)

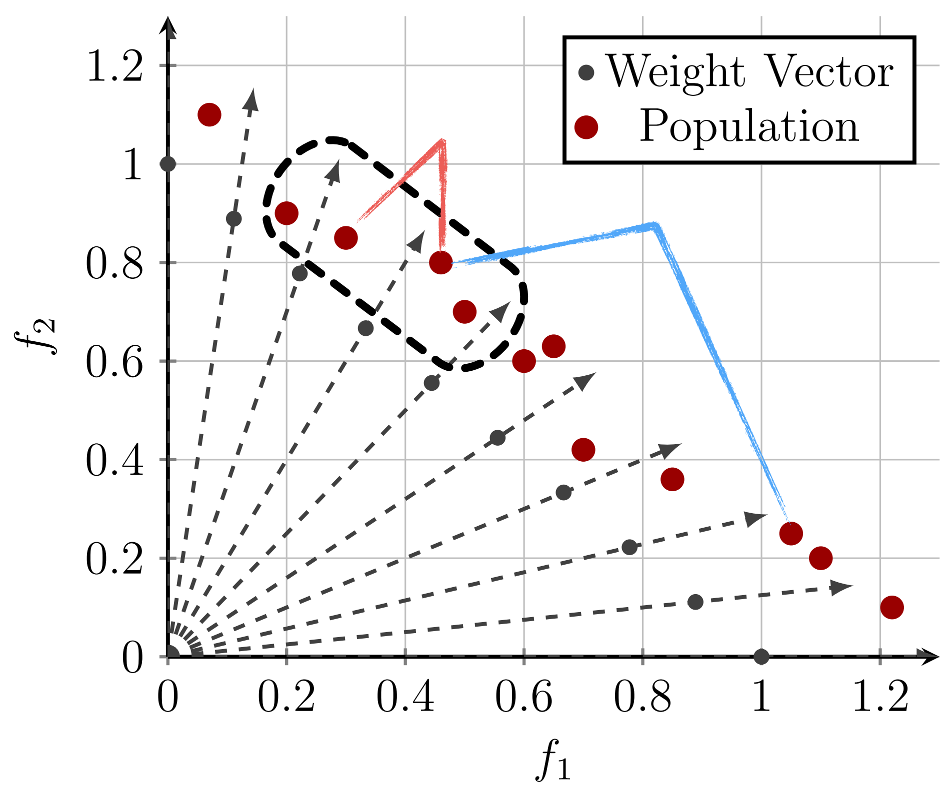


Take <u>crowdedness</u> into consideration [40]
- Compute the niche count of each solution within agent i's neighbour
- Select mating parents from outside of the neighbour if solutions are overly crowded

[40] S. Jiang, et al., "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts", IEEE Trans. Cybern, 46(2): 421-437, 2016.

Mating Selection (cont.)

- Mating selection: how to select parents for offspring reproduction?
  - Tournament selection, genotype neighbours, …
  - MOEA/Ds leverage the neighbourhood structure of weight vectors
    - Assumption: neighbouring subproblems have similar structure
    - Select mating parents purely from neighbouring agents (simple MOEA/D)



Take crowdedness into consideration [40]
- Compute the niche count of each solution within agent i's neighbour
- Select mating parents from outside of the neighbour if solutions are overly crowded

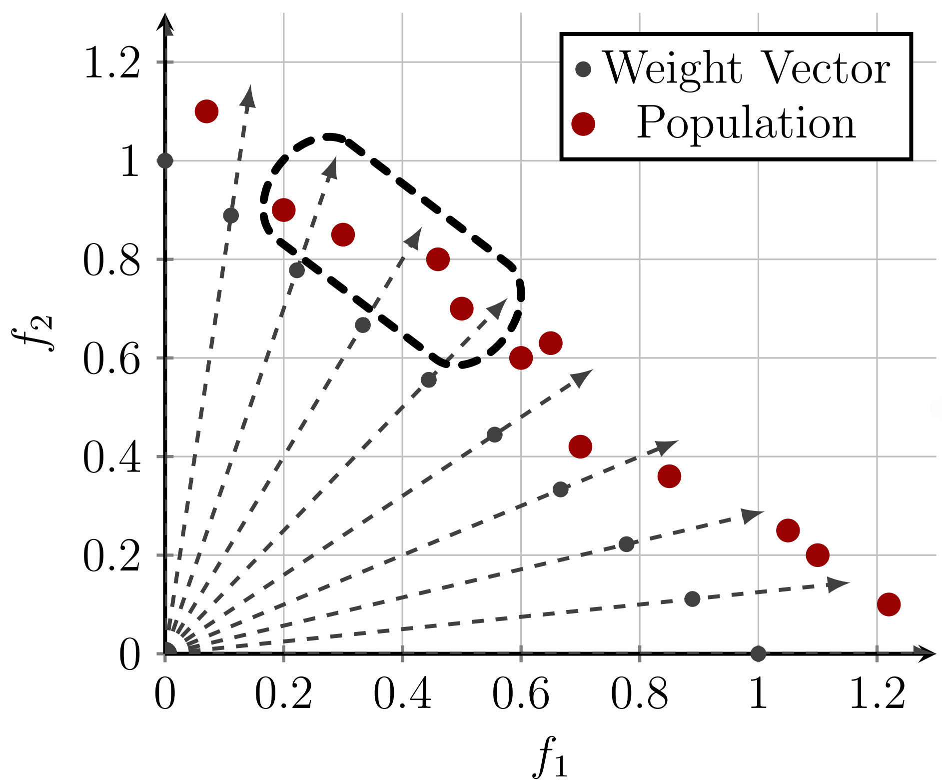
[40] S. Jiang, et al., "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts", IEEE Trans. Cybern, 46(2): 421-437, 2016.

Outline

- Part I: Basics
  - Basic Concepts
  - Simple MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
      - Mating selection
      - Replacement
  - Resources
  - Future Directions

Replacement

- Replacement: update the parent population
  - Steady-state evolution model (oracle MOEA/D)
  - Update as many neighbouring subproblems as it can (oracle MOEA/D)

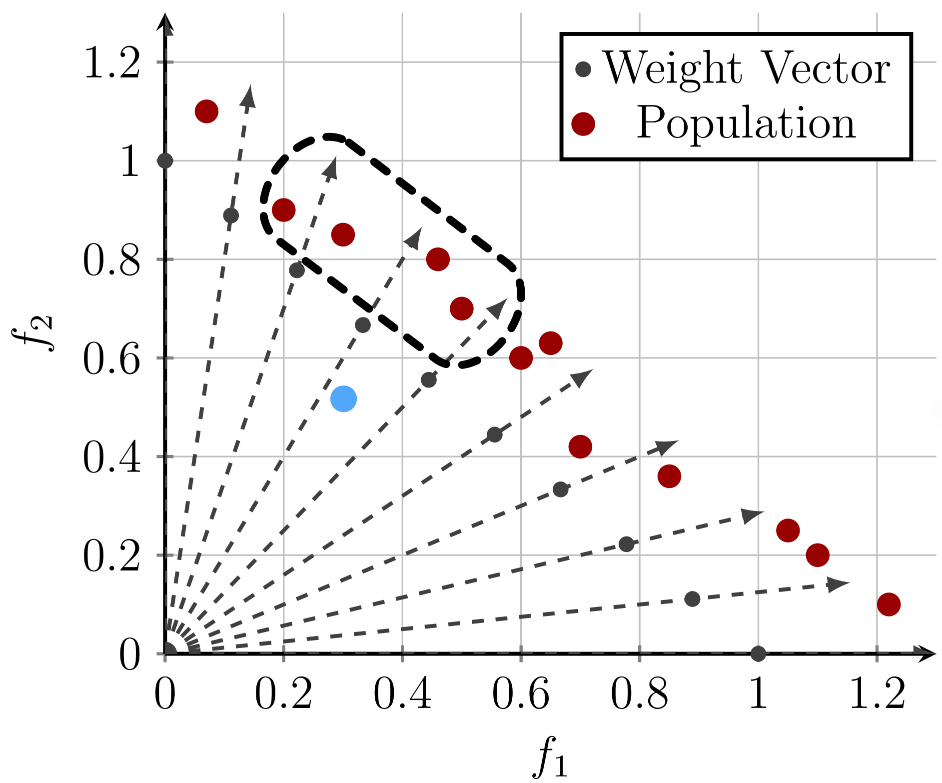


- The replacement strategy of the oracle MOEA/D is too greedy
- Offspring is only allowed to replace a limited number of parents [38]
  - Pros: Good for diversity
  - Cons: convergence may be slow

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Replacement

- Replacement: update the parent population
  - Steady-state evolution model (oracle MOEA/D)
  - Update as many neighbouring subproblems as it can (oracle MOEA/D)

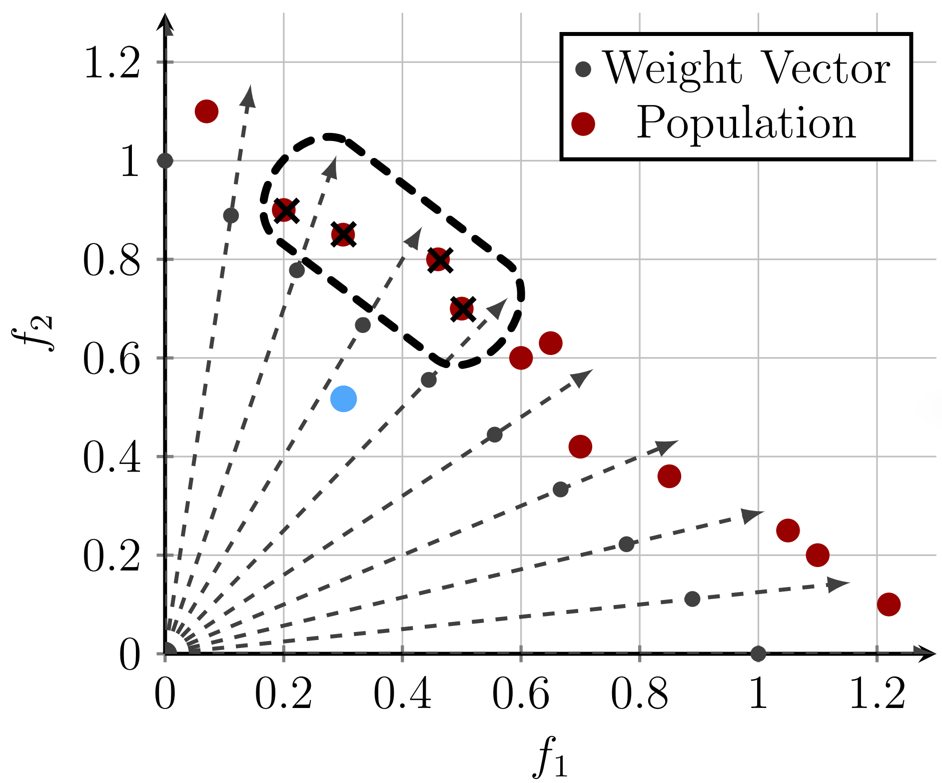


- The replacement strategy of the oracle MOEA/D is too greedy
- Offspring is only allowed to replace a limited number of parents [38]
  - Pros: Good for diversity
  - Cons: convergence may be slow

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Replacement

- Replacement: update the parent population
  - Steady-state evolution model (oracle MOEA/D)
  - Update as many neighbouring subproblems as it can (oracle MOEA/D)

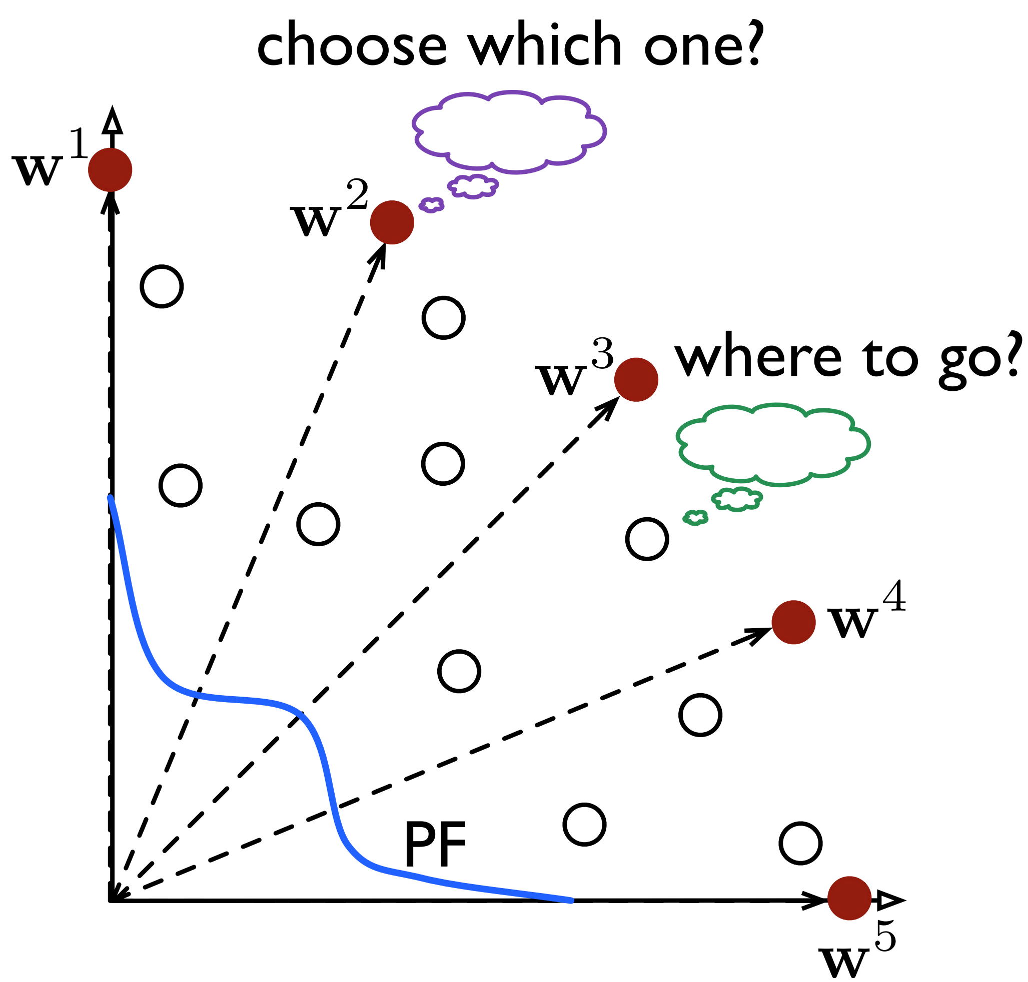


- The replacement strategy of the oracle MOEA/D is too greedy
- Offspring is only allowed to replace a limited number of parents [38]
  - Pros: Good for diversity
  - Cons: convergence may be slow

[38] H. Li and **Q. Zhang**, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans. Evol. Comput., 13(2): 284-302, 2009.

Replacement (cont.)

- Matching-based selection [41, 42]
  - Subproblems and solutions are two sets of agents
  - Subproblems 'prefer' convergence, solutions 'prefer' diversity

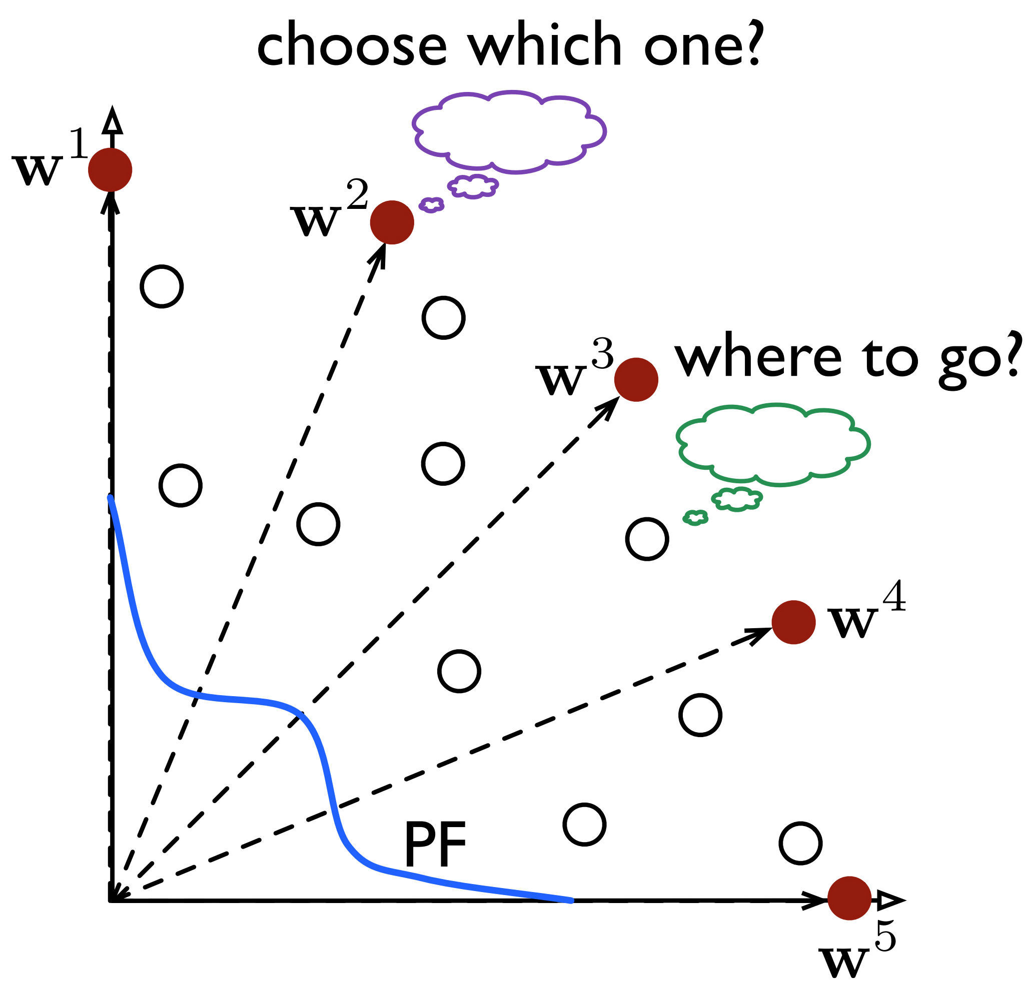


[41] **K. Li**, **Q. Zhang**, et al., "Stable Matching Based Selection in Evolutionary Multiobjective Optimization", IEEE Trans. Evol. Comput., 18(6): 909–923, 2014.

[42] M. Wu, **K. Li**, et al., "Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization", IEEE Trans. Evol. Comput., 21(4): 554–568, 2017.

Replacement (cont.)

- Matching-based selection [41, 42]
  - Subproblems and solutions are two sets of agents
  - Subproblems ‘prefer’ convergence, solutions ‘prefer’ diversity



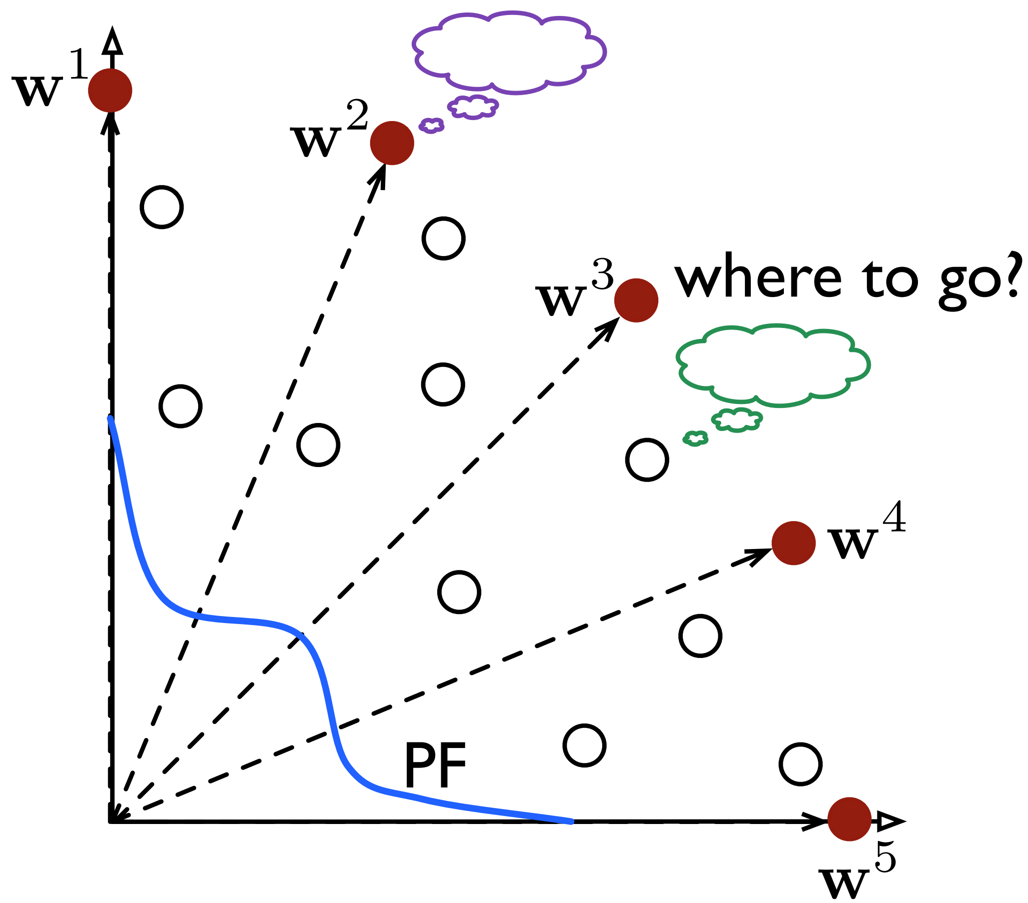
selection — matching

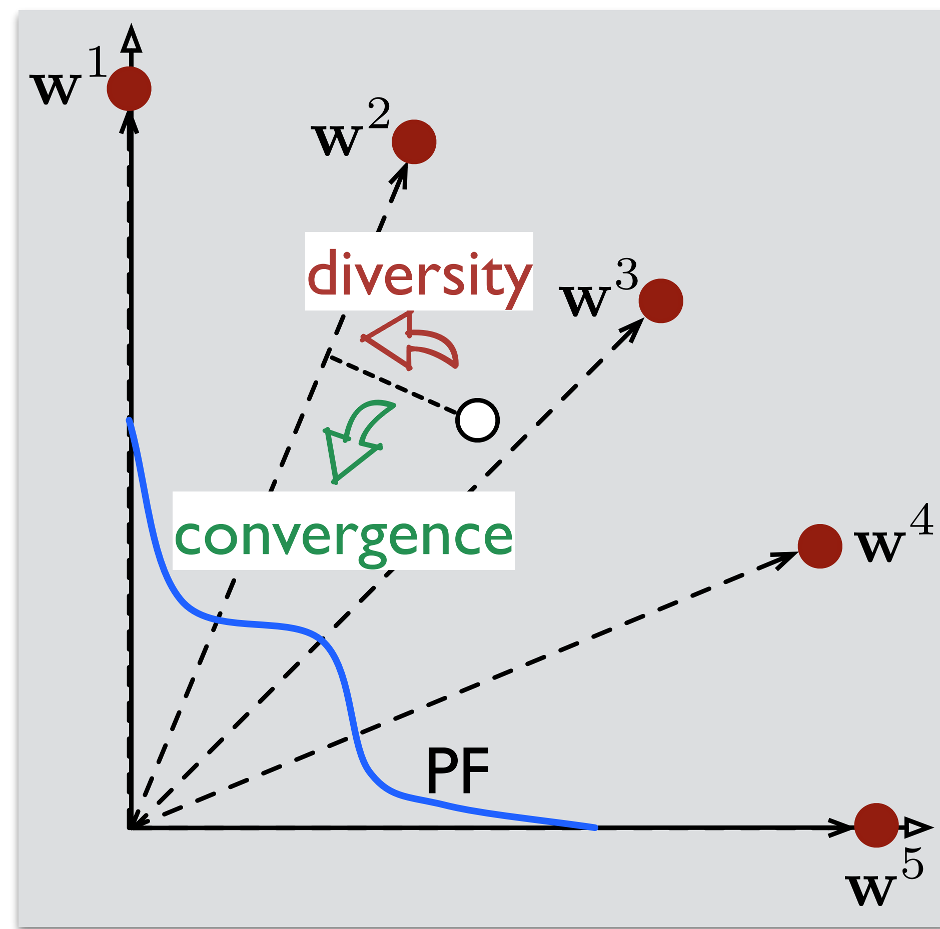
[41] **K. Li**, **Q. Zhang**, et al., “Stable Matching Based Selection in Evolutionary Multiobjective Optimization”, IEEE Trans. Evol. Comput., 18(6): 909–923, 2014.

[42] M. Wu, **K. Li**, et al., “Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization”, IEEE Trans. Evol. Comput., 21(4): 554–568, 2017.

Replacement (cont.)

- Matching-based selection [41, 42]
  - Subproblems and solutions are two sets of agents
  - Subproblems 'prefer' <u>convergence</u>, solutions 'prefer' <u>diversity</u>



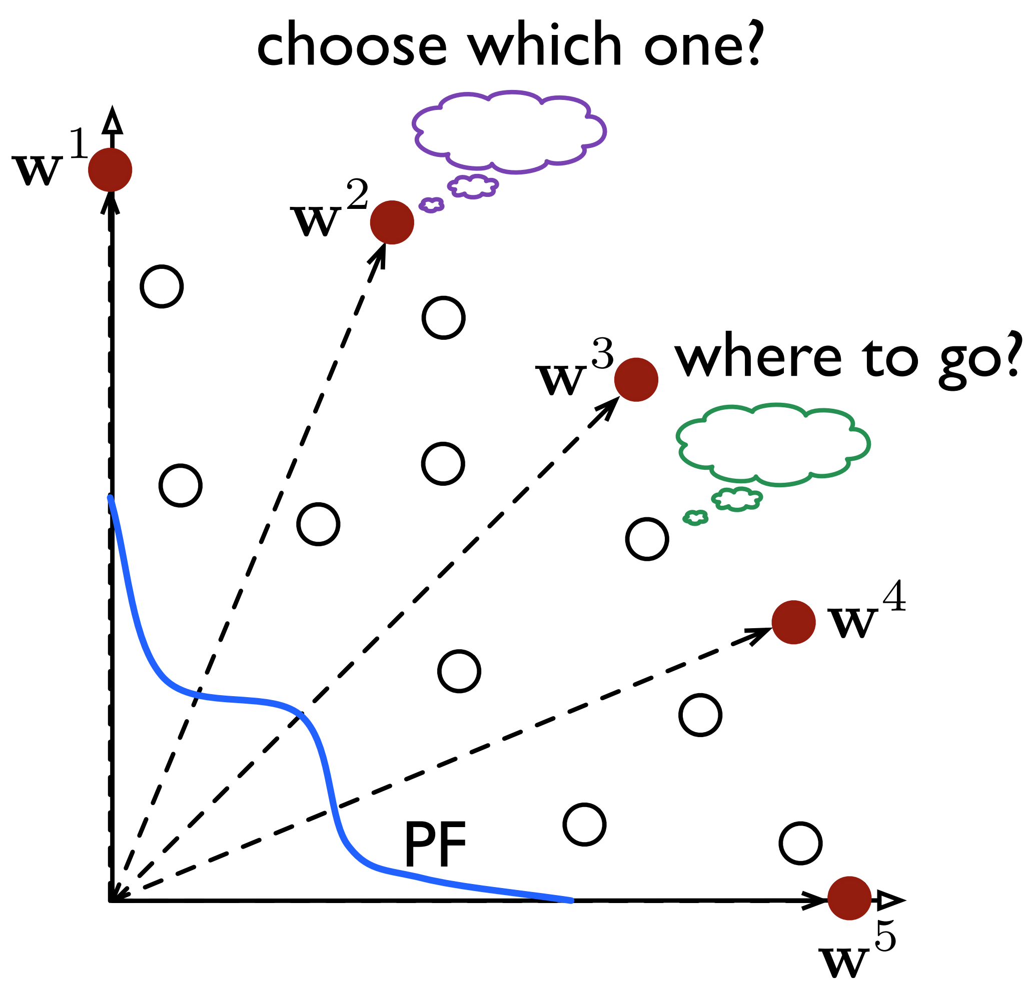


selection — matching

[41] **K. Li**, **Q. Zhang**, et al., "Stable Matching Based Selection in Evolutionary Multiobjective Optimization", IEEE Trans. Evol. Comput., 18(6): 909–923, 2014.
[42] M. Wu, **K. Li**, et al., "Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization", IEEE Trans. Evol. Comput., 21(4): 554–568, 2017.

Replacement (cont.)

- Matching-based selection [41, 42]
  - Subproblems and solutions are two sets of agents
  - Subproblems 'prefer' convergence, solutions 'prefer' diversity



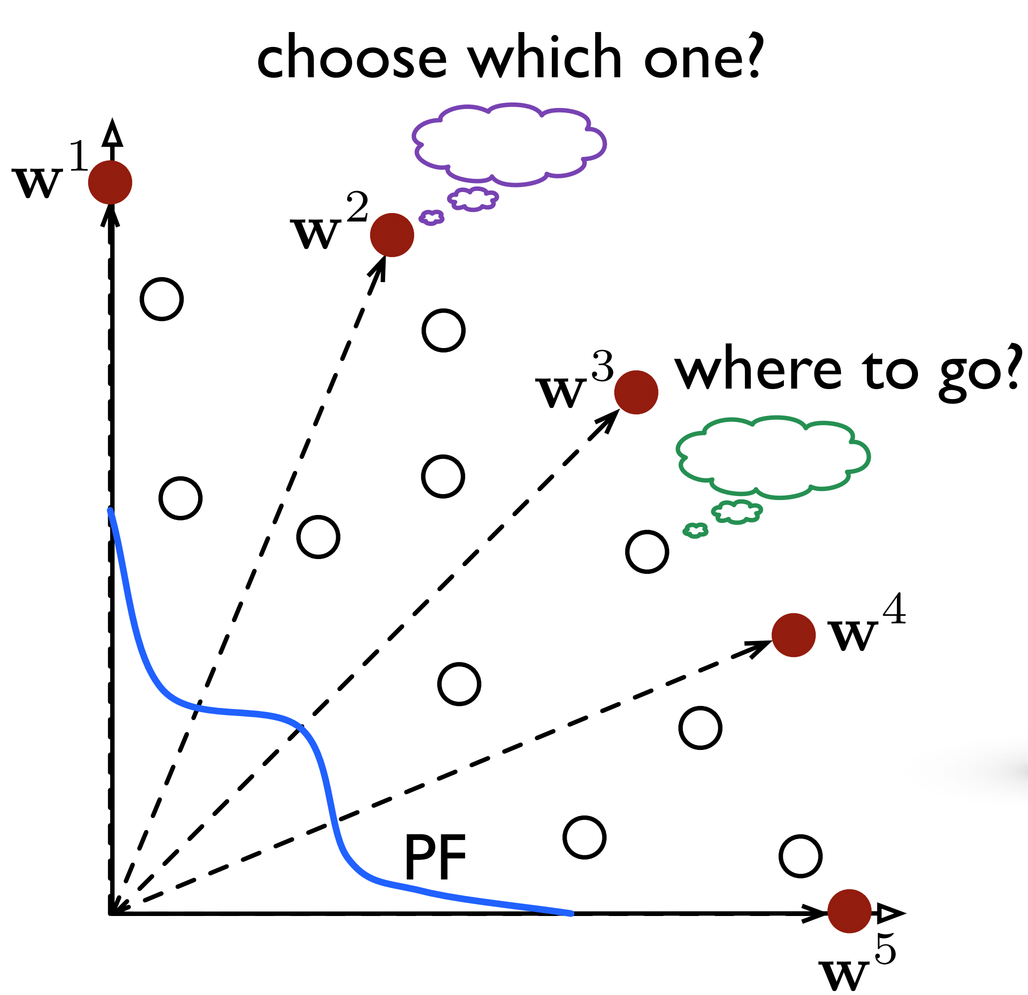
selection — matching

[41] **K. Li**, **Q. Zhang**, et al., "Stable Matching Based Selection in Evolutionary Multiobjective Optimization", IEEE Trans. Evol. Comput., 18(6): 909–923, 2014.

[42] M. Wu, **K. Li**, et al., "Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization", IEEE Trans. Evol. Comput., 21(4): 554–568, 2017.

Replacement (cont.)

- Matching-based selection [41, 42]
  - Subproblems and solutions are two sets of agents
  - Subproblems 'prefer' convergence, solutions 'prefer' diversity



selection — matching

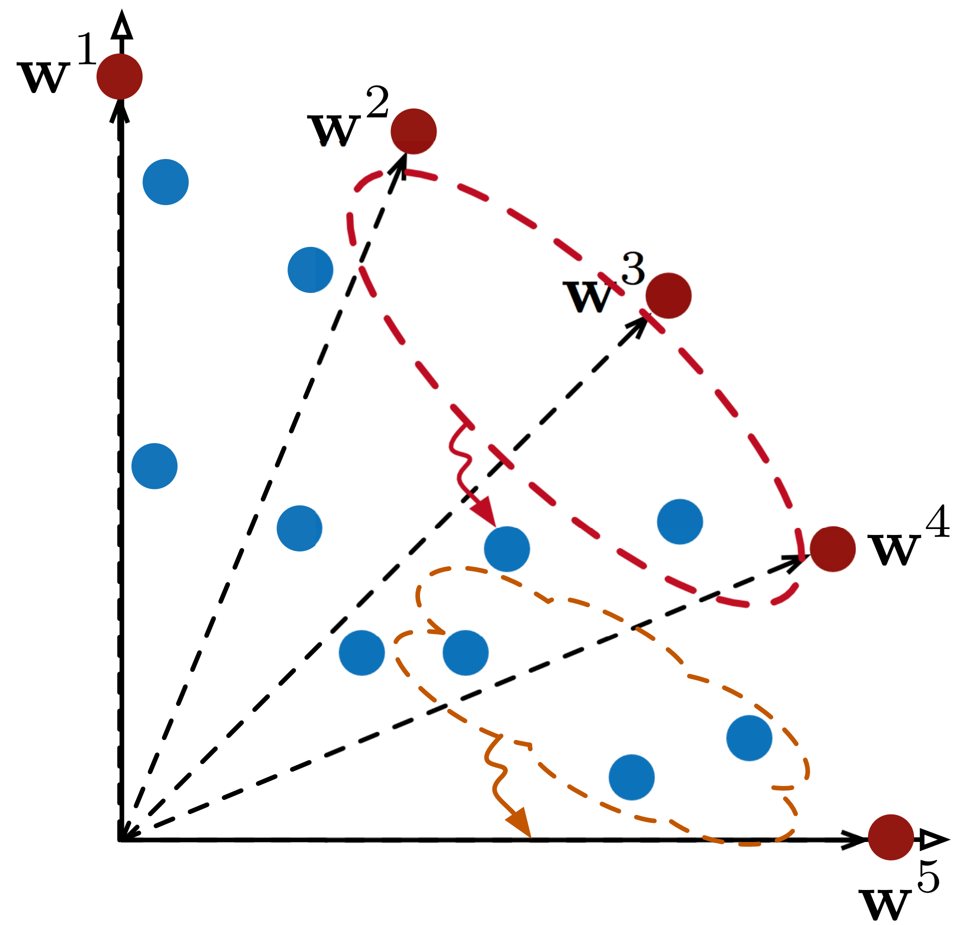
- A unified perspective to look at selection
- A generational evolution model for MOEA/D
  - ✓ What is convergence?
    - ➡ Aggregation function, …
  - ✓ What is diversity?
    - ➡ Perpendicular distance, angle …
  - ✓ Mechanism to match
    - ➡ Stable matching, …

[41] **K. Li**, **Q. Zhang**, et al., "Stable Matching Based Selection in Evolutionary Multiobjective Optimization", IEEE Trans. Evol. Comput., 18(6): 909–923, 2014.

[42] M. Wu, **K. Li**, et al., "Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization", IEEE Trans. Evol. Comput., 21(4): 554–568, 2017.

Replacement (cont.)

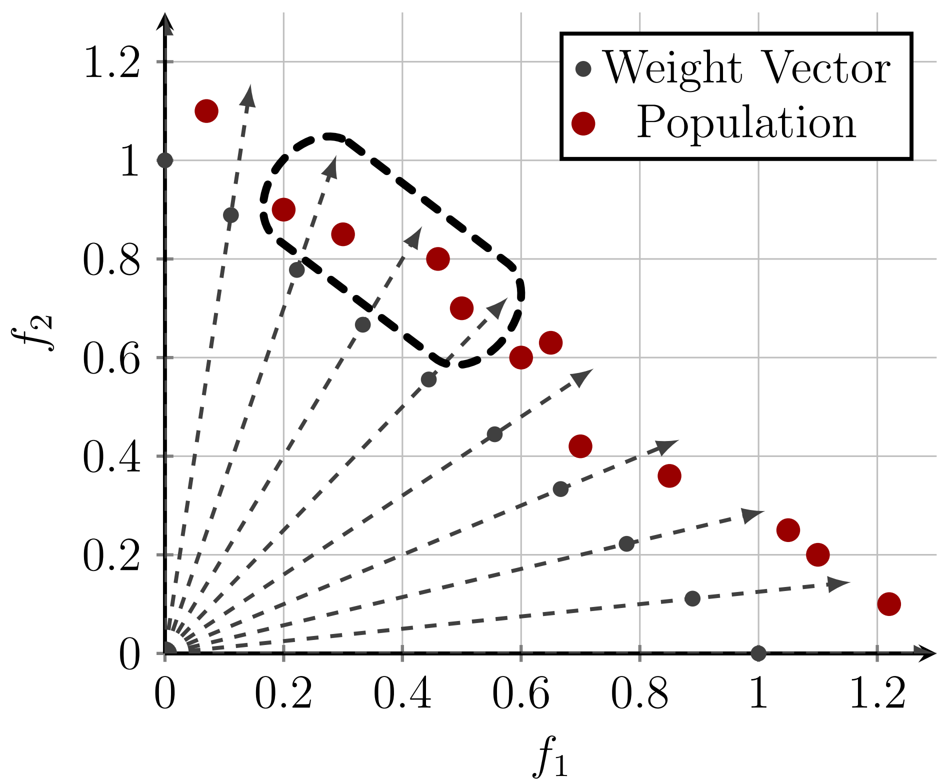
- Matching-based selection (extension) [43]
  - Identify the inter-relationship between subproblems and solutions
    - Find the related subproblems to each solution (e.g. fitness)
    - Find the related solutions for each subproblem (e.g. closeness)
  - Selection mechanism: each subproblem chooses its favourite solution



[43] **K. Li**, **Q. Zhang**, et al., "Interrelationship-based selection for decomposition multiobjective optimization", IEEE Trans. Cybern. 45(10): 2076–2088, 2015.

Replacement (cont.)

- Matching-based selection (extension):
  - Global replacement [44]
    - If the newly generated offspring is way beyond the current neighbourhood …
    - Find the 'best agent' (i.e. subproblem) for the newly generated offspring
    - Compete with solutions associated with this 'best agent'
  - MOEA/D-DU [45]
    - Update the newly generated offspring's 'nearest' subproblems

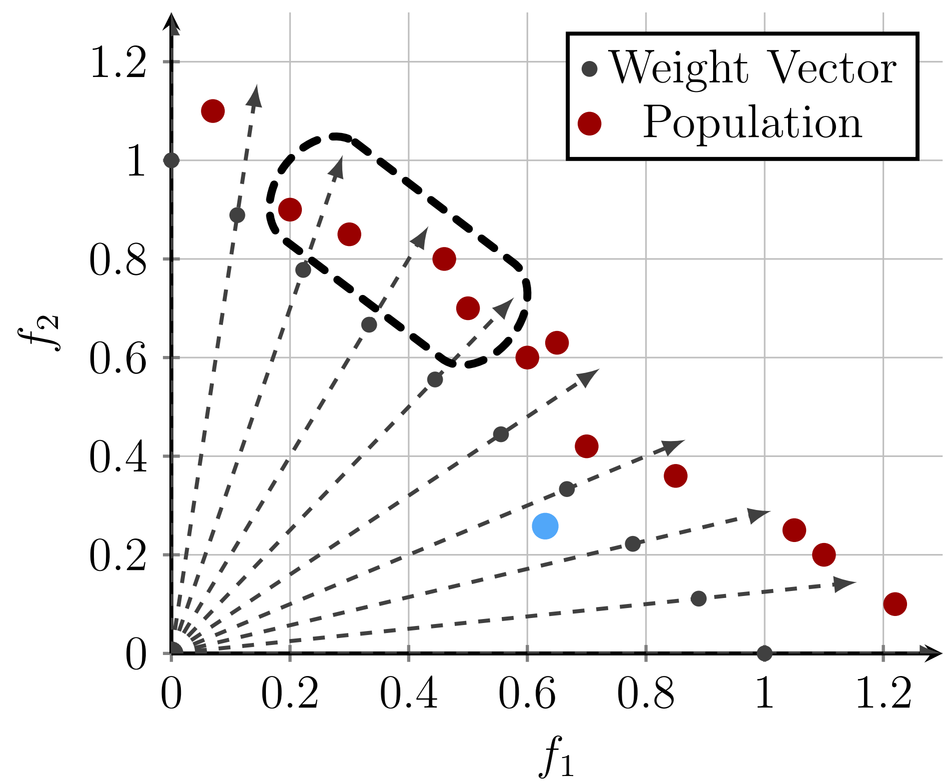


[44] Z. Wang, **Q. Zhang**, et al., "Adaptive Replacement Strategies for MOEA/D", IEEE Trans. Cybern., 46(2): 474-486, 2016.

[45] Y. Yuan, et al., "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers", IEEE Trans. Evol. Comput., 20(2): 180-198, 2016.

Replacement (cont.)

- Matching-based selection (extension):
  - Global replacement [44]
    - If the newly generated offspring is way beyond the current neighbourhood …
    - Find the 'best agent' (i.e. subproblem) for the newly generated offspring
    - Compete with solutions associated with this 'best agent'
  - MOEA/D-DU [45]
    - Update the newly generated offspring's 'nearest' subproblems

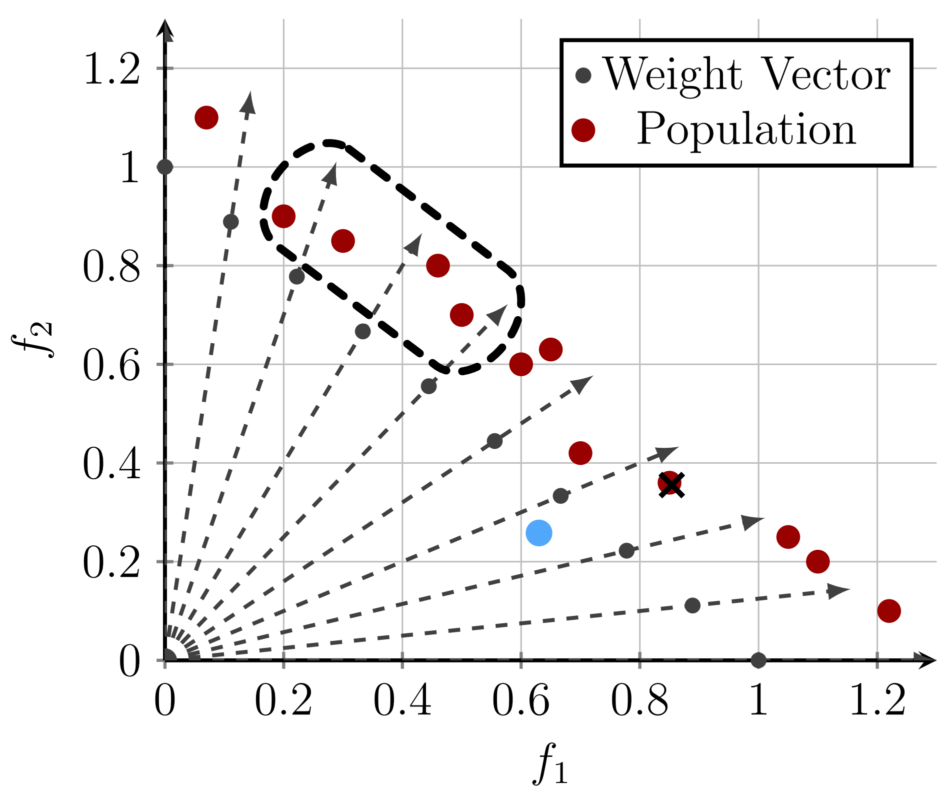


[44] Z. Wang, **Q. Zhang**, et al., "Adaptive Replacement Strategies for MOEA/D", IEEE Trans. Cybern., 46(2): 474-486, 2016.

[45] Y. Yuan, et al., "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers", IEEE Trans. Evol. Comput., 20(2): 180-198, 2016.

Replacement (cont.)

- Matching-based selection (extension):
  - Global replacement [44]
    - If the newly generated offspring is way beyond the current neighbourhood …
    - Find the 'best agent' (i.e. subproblem) for the newly generated offspring
    - Compete with solutions associated with this 'best agent'
  - MOEA/D-DU [45]
    - Update the newly generated offspring's 'nearest' subproblems



[44] Z. Wang, **Q. Zhang**, et al., "Adaptive Replacement Strategies for MOEA/D", IEEE Trans. Cybern., 46(2): 474-486, 2016.

[45] Y. Yuan, et al., "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers", IEEE Trans. Evol. Comput., 20(2): 180-198, 2016.

Outline

- **Part I: Basics**
  - Basic Concepts
  - Simple MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
      - Mating selection
      - Replacement
  - Resources
  - Future Directions

Resources

- IEEE CIS task force on decomposition-based techniques in EC

coda-group.github.io

CODA Group
- Home
- Members
- Openings
- IEEE DTEC
- News

Research
- Publications
- Projects
- Reading Group

Miscellaneous
- Readings
- Links

Visitor Stats.

ShinyStat™
Online 2
Tot p.views 39996

Exeter

IEEE Computational Intelligence Society

IEEE CIS Task Force on Decomposition-based Techniques in Evolutionary Computation

Objectives

As the name suggests, the basic idea of the decomposition-based technique is to transform the original complex problem into simplified subproblem(s) so as to facilitate the optimization. Decomposition-based techniques have been widely used for solving both single- and multi-objective optimization problems. More specifically, in single-objective optimization, especially for the large-scale scenarios, which consider a tremendous amount of decision variables, the decomposition-based technique contains three aspects: 1) analyzing and understanding the fitness landscape and modularity structure of the underlying problem; 2) decomposing the original complex problem into several loosely coupled or independent subproblems based on the learnt characteristics; 3) using a meta-heuristic to solve these subproblems in a sequential or concurrent manner. As for multi-objective optimization, the decomposition means to decompose the original multi-objective optimization problem into a number of single-objective optimization sub-problems (or simple multi-objective optimization problems) and then uses a meta-heuristic to optimize these sub-problems simultaneously and collaboratively. In this big data era, the decomposition-based techniques used for both single- and multi-objective optimization can be sythesized to address the challenges posed by the *curse of dimensionality*, i.e., many objectives and large scale variables.

The key objective of this task force it to generalize the decomposition-based idea and to promote its related research, including its development, education and understanding of its sub topic areas.

The main objectives of the task force can be summarized as follows:

- create an active and healthy community to promote theme areas of decomposition-based techniques
- make student, researchers, end-users, developers, and consultants aware of the state-of-the-art
- promote the use of decomposition-based methodologies/techniques and tools
- organize conferences/workshop with IEEE CIS Technical Co-Sponsorship
- organize tutorials, workshops and special sessions
- launch edited volumes, books, and special issues in journals

Anticipated Interests

This task force will focus on all aspects, including theory, practice and applications, of the decomposition-based technique in evolutionary computation for solving both single-, multi- and many-objective optimization problems.

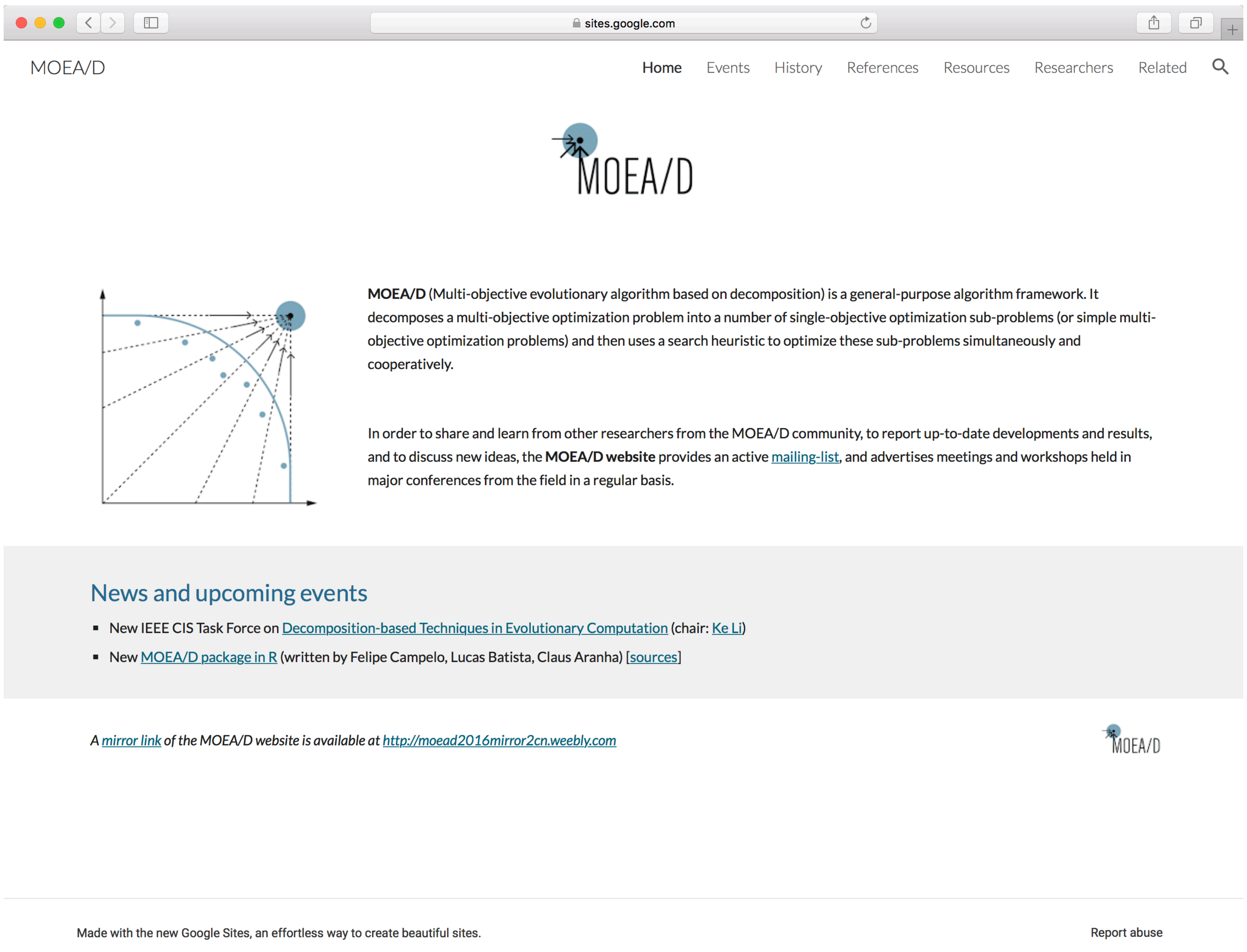
Topics of interest including but are not limited to the following:

- Design of novel weight vector generation methods
- Development of new decomposition methods
- Design of novel computational resource allocation strategies

GECCO

Resources (cont.)

- Website of MOEA/D: https://sites.google.com/view/moead/home



Resources (cont.)

- Two survey papers

440 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 21, NO. 3, JUNE 2017

A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition

Anupam Trivedi, *Member, IEEE*, Dipti Srinivasan, *Senior Member, IEEE*, Krishnendu Sanyal, and Abhiroop Ghosh

***Abstract*—Decomposition is a well-known strategy in traditional multiobjective optimization. However, the decomposition strategy was not widely employed in evolutionary multiobjective optimization until Zhang and Li proposed**

where Ω is the search space and x is the decision variable vector. $F : \Omega \rightarrow \Re^m$, where m is the number of objective functions, and $\Re^m$ is the objective space.

A Survey of Decomposition Methods for Multi-objective Optimization

Alejandro Santiago, Héctor Joaquín Fraire Huacuja, Bernabé Dorronsoro, Johnatan E. Pecero, Claudia Gómez Santillan, Juan Javier González Barbosa and José Carlos Soto Monterrubio

Events

- Workshop on decomposition techniques in evolutionary optimisation (DTEO)

Events

- Workshop on Computational Intelligence for Massive Optimisation (CIMO)

sites.google.com

CIMO 2018

CIMO 2018

1st International Workshop on Computational Intelligence for Massive Optimization (CIMO 2018)

July 12-13, 2018 — Nagano, Japan

@ contact

Overview

This is the first event of the **CIMO workshop series**. It will take place from **July 12** to **July 13, 2018** on the Engineering Campus of Shinshu University, in **Nagano (Japan)**. The CIMO workshop aims at bringing together researchers interested in developing integrated **computational intelligence** techniques into **advanced (evolutionary) optimization** paradigms for solving **massive optimization** problems. Boosted by the recent creation of the international associated laboratory **MODŌ**, this thematic workshop targets an international audience, with a particular emphasis in strengthening the long history and sustained scientific relations and collaborations between France and Japan. The program will consist of talks and posters about late-breaking research reflecting the state-of-the-art of evolutionary computation for massive optimization, and will provide many opportunities to interact with other attendees in order to encourage international cooperation and to favor the education of young researchers.

Notice that CIMO 2018 is purposely organized just before **GECCO 2018**, one of the main conference in evolutionary computation, that will also be held in Japan (Kyoto) at the same period (July 15-19). This makes it convenient for interested attendees to participate in both events. Kyoto can be reached from Nagano in less than 4 hours by train.

Outline

- **Part I: Basics**
  - Basic Concepts
  - Simple MOEA/D
  - A Simple Variant
- **Part II: Advanced Topics**
  - Current Developments
    - Decomposition methods
    - Search methods
    - Collaboration
      - Mating selection
      - Replacement
  - Resources
  - Future Directions

Future Directions

- Big optimisation
  - Many objectives
    - Is approximating the high-dimensional PF doable?
    - Problem reformulation (dimensionality reduction)
    - Visualisation
    - …
  - Many variables (large-scale)
    - Decomposition from decision space (divide-and-conquer): dependency structure analysis
    - What is the relationship between the decomposed variable and subproblem?
    - Sensitivity analysis for identifying important variables
    - …
  - Distributed and parallel computing platform
- EMO + MCDM: Human computer interaction perspective
  - Subproblem is another way to represent decision maker's preference
    - e.g. weighted scalarizing function, simplified MOP
  - How to help decision maker understand the solutions and inject appropriate preference information?
  - How to use preference information effectively?
  - …

Future Directions (cont.)

- How to make the collaboration more effective?
  - "*In case of two agents for one problem, collaboration is useful*" [34]
  - How about a multi-agent system and cooperative game?
- Automatic problem solving: meta-optimisation/learning perspective
  - Is the current MOEA/D the perfect algorithm structure?
  - Use artificial intelligence to design algorithm autonomously
  - Landscape analysis and problem feature engineering
  - Algorithm portfolio: choose the right algorithm structure for the right problem
  - …
- Data-driven optimisation
  - Build and maintain a surrogate for each subproblem
  - Subproblem has knowledge, e.g. solution history, knowledge can be shared among neighbourhood: transfer learning or multi-tasking?
  - …

[34] B. Huberman, et. al., "An Economics Approach to Hard Computational Problems", Science, 275(5296): 51-54, 1997.

Future Directions (cont.)

- Theoretical studies
  - Convergence analysis
  - Stopping condition
  - From an equilibrium perspective?
  - …
- Applications
  - Engineering, e.g. water, manufacturing, renewable energy, healthcare …
  - Search-based software engineering
  - …
- Any suggestions?
  - …

[22] B. Huberman, et. al., "An Economics Approach to Hard Computational Problems", Science, 275(5296): 51-54, 1997.


Thank you for your participation and any questions?