Tutorial: Multi-Objective Optimization and Recommendations

Part 1: Multi-Objective Optimization

David Wang
Principal Data Scientist
Morningstar, Inc.
11/29/2022
Time: 10:30 - 12:00 ET

Contents

Background

- Multi Objective Optimization Problems Everywhere

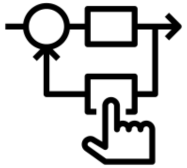
Consumer

Price ↓
Fuel consumption ↓
Comfort ↑
Performance ↑

Central Bank

Inflation ↓
Unemployment ↓
Trade deficit ↓

Engineer



Performance ↑
Cost ↓

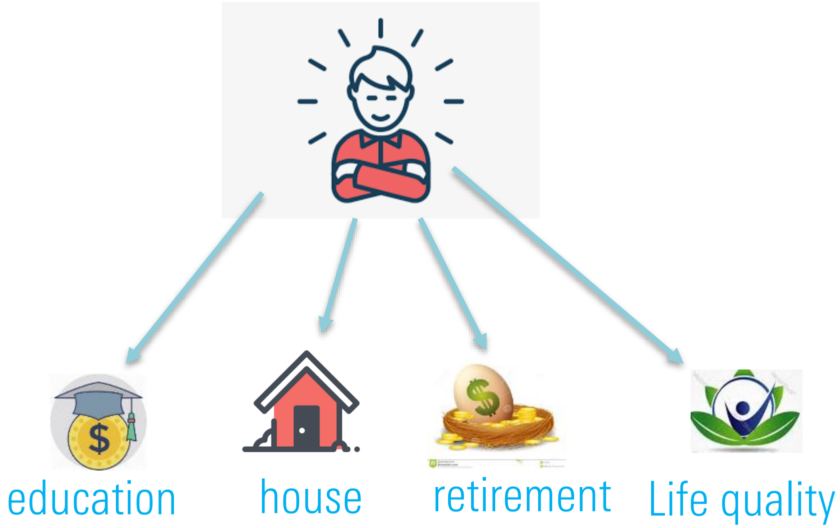
Communication

Data rate (bandwidth) ↑
Latency ↓
Energy consumption ↓

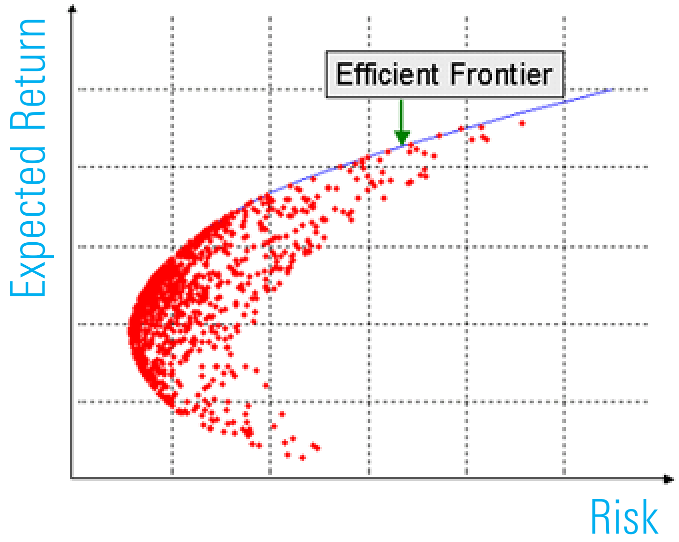
Background

- Multi Objective Optimization in Finance

Financial Planning



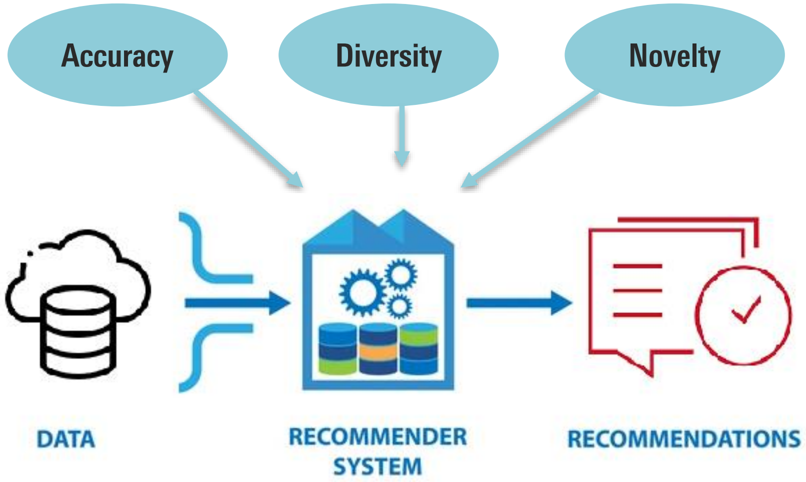
Portfolio Optimization



+ESG (Environmental, Social, and Governance)

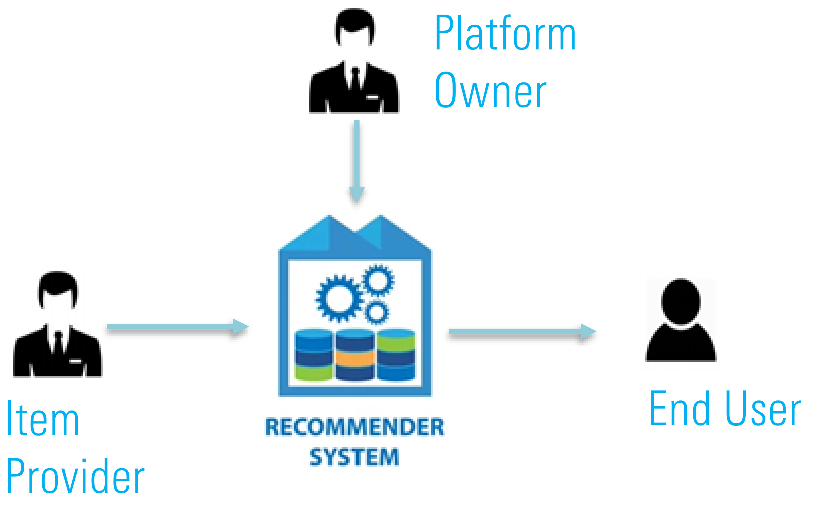
Background

- Multi Metrics in Recommendation Systems
  - Goal: Meets user's need
  - Objectives:
    - Maximize Accuracy
    - Maximize Diversity
    - Maximize Novelty
  - Challenge
    - Increase Diversity may decease Accuracy
    - Increase Novelty may decease Accuracy



Background

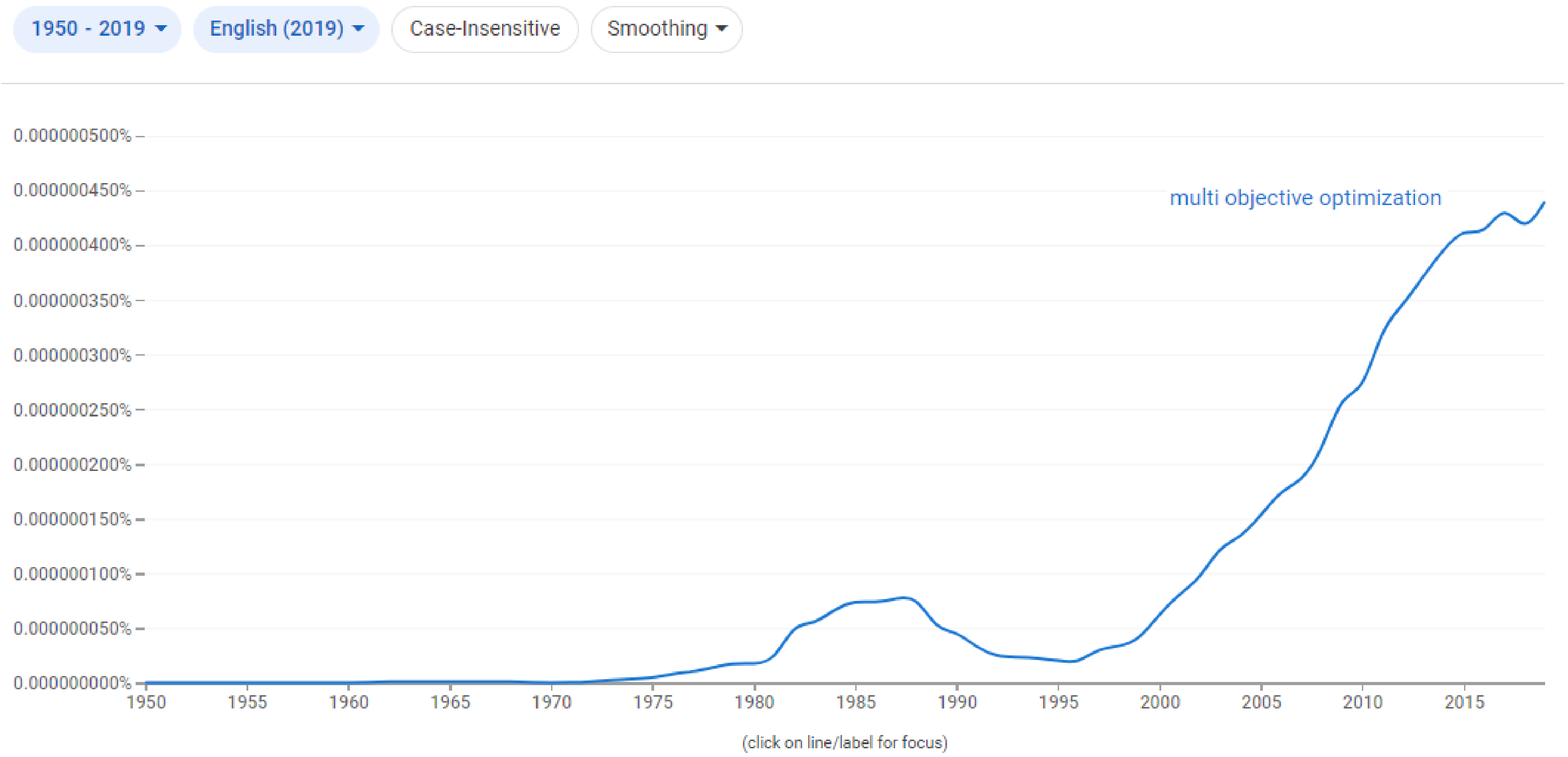
- Multi Stakeholder Recommendation Systems
  - Goal: Meet interests of all stakeholders
  - Objectives: Maximize Three Utilities
    - In respect of End User
    - In respect of Provider
    - In respect of Platform Owner
  - Challenge
    - Utilities regarding to three stakeholders may conflict each other



MORNINGSTAR®

Background

- MOO Research is Becoming Popular

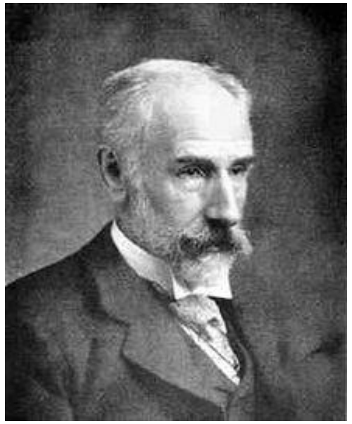


- Sources: https://books.google.com/ngrams/graph?content=multi+objective+optimization&year_start=1950&year_end=2019&corpus=26&smoothing=3

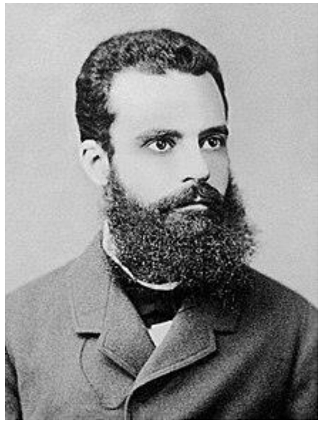
MORNINGSTAR®

History

- Francis Ysidro Edgeworth (1845-1926)
  - ***Mathematical Psychics**: **An Essay on the Application of Mathematics to the Moral Sciences***, published in 1881
  - *"It is required to find a point (x, y) such that, in whatever direction we take an infinitely small step, P and Π do not increase together, but that, while one increases, the other decreases"*



- Vilfredo Pareto (1848-1923)
  - ***Manual of Political Economy***, published in 1906
  - Formally defined '*Pareto optimal*'
  - *"The optimum allocation of the resources of a society is not attained so long as it is possible to make at least one individual better off in his own estimation while keeping others as well off as before in their own estimation."*



Contents

Multi Objective Optimization (MOO): Definition

- Multi Objective Optimization (MOO) Problem

$$\min_{\boldsymbol{x}}(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_M(\boldsymbol{x}))$$

Subject to:

$$g_j(\boldsymbol{x}) \geq 0, \qquad j = 1, 2, \dots, J$$
$$h_k(\boldsymbol{x}) = 0, \qquad k = 1,2, \dots, K$$
$$x_i^L \leq x_i \leq x_i^U, \qquad i = 1, 2, \dots, n$$

Decision variable: $\boldsymbol{x} \in \boldsymbol{R}^n$

Objective Functions: $f_i,\ i = 1,2, \dots, M, \quad (f_1, f_2, \dots, f_M) \in \boldsymbol{R}^M,$

Feasible Solutions: $S \subseteq \boldsymbol{R}^n$

$$S = \{x \mid x_i^L \leq x_i \leq x_i^U, g_j(x) \geq 0, h_k(x) = 0, j = 1, 2, \dots, J, k = 1,2, \dots, K, i = 1, \dots, n\}$$

Multi Objective Optimization (MOO): Example

- Example: Two Objectives (metrics) in Recommender Systems

Name	Symbol	Meaning
Decision Variable	x	A Top N recommendation list
Feasible Solution Set	S	All top N recommendations
First Objective	$f_1(x)$	1 – accuracy
Second Objective	$f_2(x)$	1 – diversity

- Find recommendation list that maximize accuracy and diversity

$$\min_{x \in S}(f_1, f_2) \quad \text{or} \quad \min_{x \in S} F(x), \text{ where } F(x) = (f_1, f_2)$$

Multi Objective Optimization (MOO): Special Features

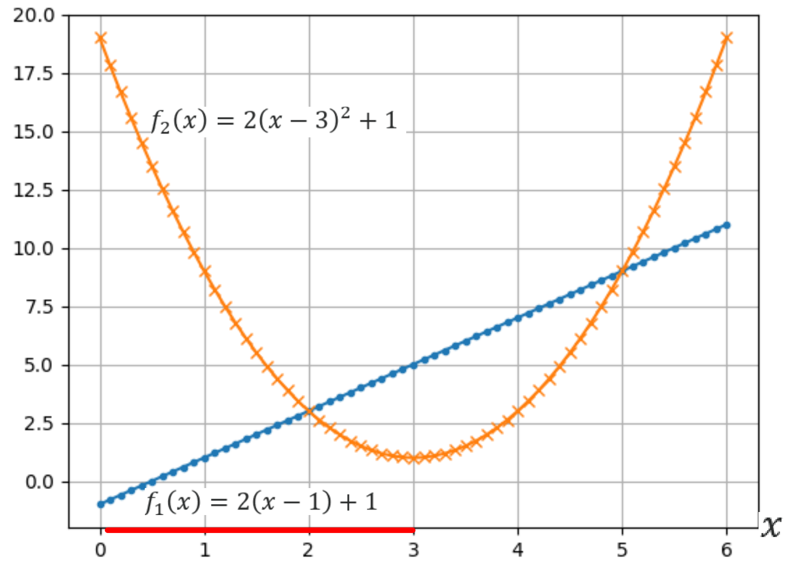
- Special Characteristic of MOO
  - Objectives may be conflict each other
  - Cannot determine which solution is better
  - Example:

$$\min_{x}(f_1, f_2)$$

Where $f_1(x) = 2(x-1)+1,$

$$f_2(x) = 2(x-3)^2 + 1$$

Subject $x \in [0, 6]$



Multi Objective Optimization (MOO): Dominance Relation

- Dominance Relation

A solution x is said to be Dominated by x^* if and only if

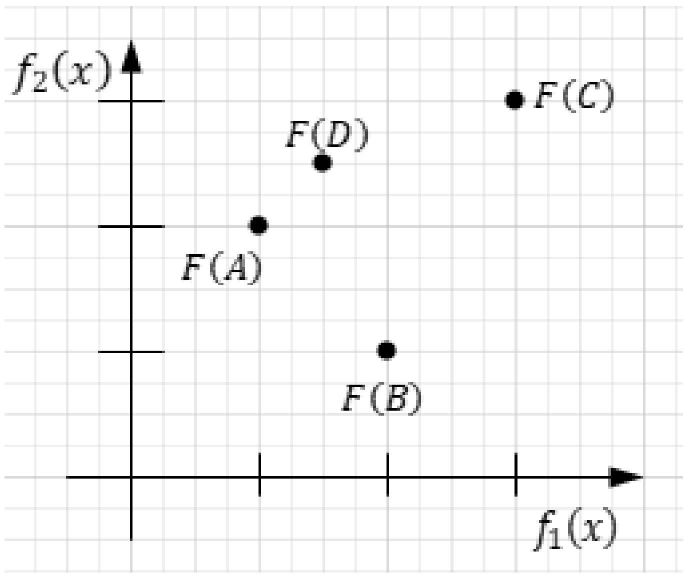
$$f_m(x^*) \leq f_m(x) \text{ for all } m = 1,2,\ldots,M$$

and there exists at least one m' such that:

$$f_{m'}(x^*) < f_{m'}(x)$$

A and B dominate C,
D is only dominated by A.
A and B: no dominance relationship
D and B: no dominance relationship

$$\min_x (f_1, f_2)$$

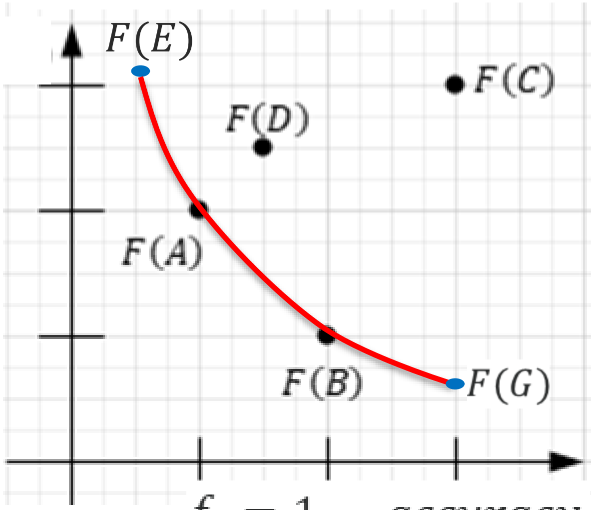


Multi Objective Optimization (MOO): Pareto Optimal

- Non-Dominated Solution (Pareto Optimal Solution)
  - Not dominated by any other solutions
  - Solution A, B, E and G are Pareto Optimal
- Pareto Optimal Set (in decision variable space):
  - All x such that $F(x)$ is on curve from $F(E)$ to $F(G)$
- Pareto Front (in objective space):
  - All $F(x)$ on curve from $F(E)$ to $F(G)$

$$\min_x(f_1, f_2)$$

$$f_2 = 1 - diversity$$



$$f_1 = 1 - accuracy$$

Multi Objective Optimization (MOO): Example

- Example:

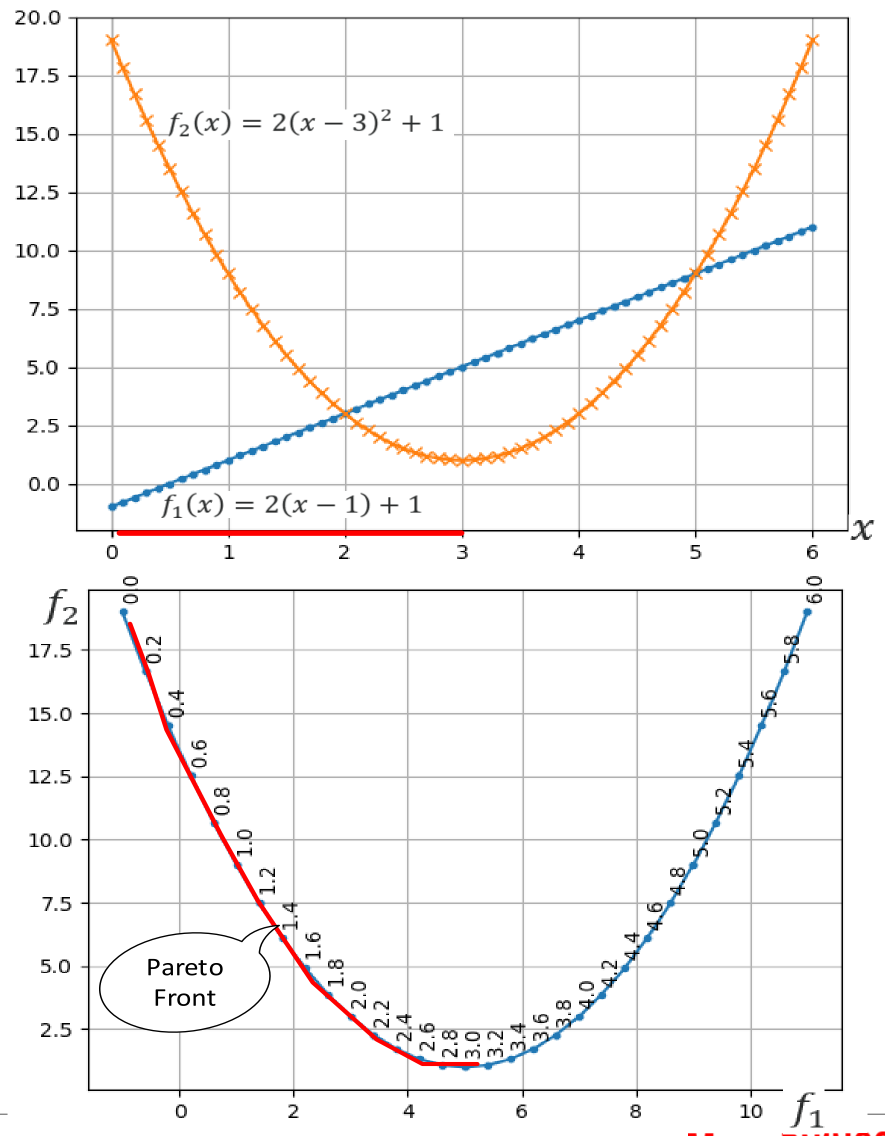
$$\min_{x}(f_1, f_2)$$

Where $f_1(x) = 2(x-1)+1,$

$$f_2(x) = 2(x-3)^2 + 1$$

Subject $x \in [0, 6]$

- Analysis
  - Feasible solutions: $S = [0,6]$
  - Pareto Set: $\{x \mid x \in [0,3]\}$
  - Pareto Front: $\{(f_1, f_2) | x \in [0, 3]\}$



Multi Objective Optimization (MOO): Goals

- Solving Multi Objective Optimization (MOO) Problem

$$\min_{x \in S} F(\boldsymbol{x})$$

where $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))$

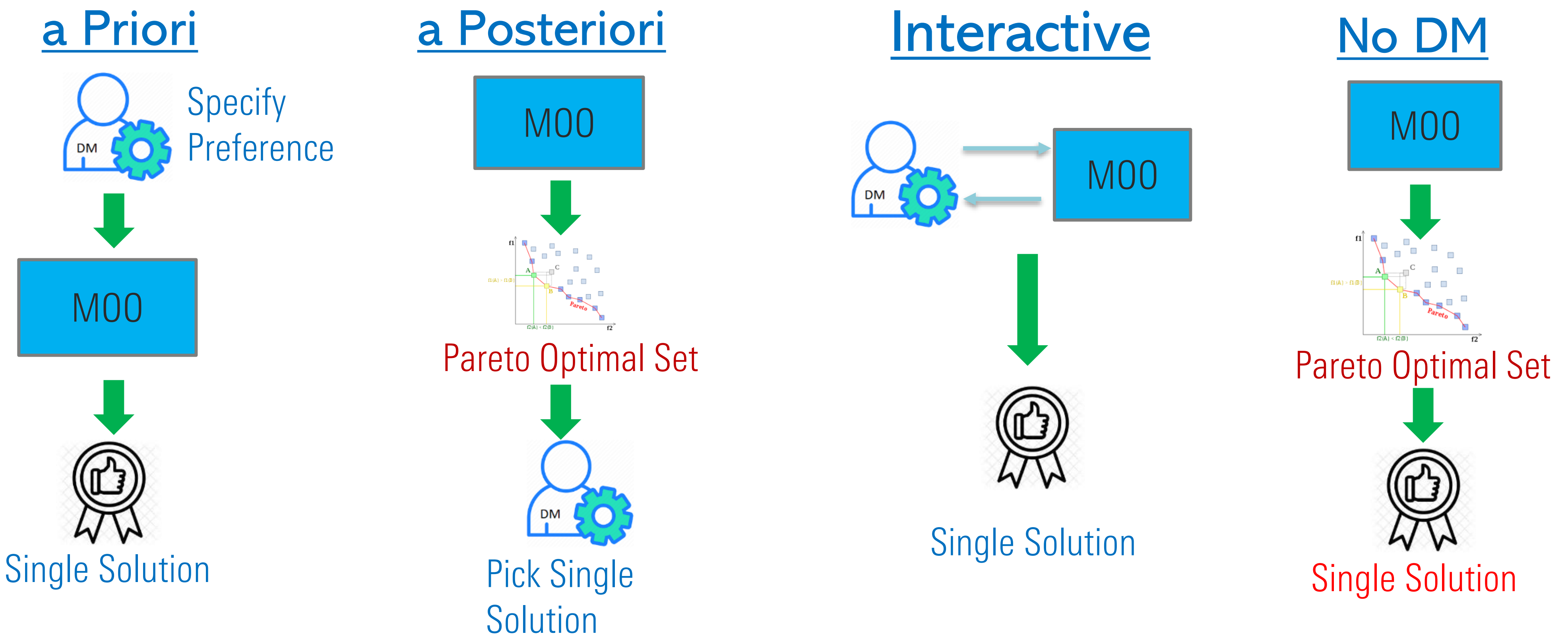
$\boldsymbol{x} \in S$

S is set of all feasible solutions

- Outputs
  - Find a Non-Dominated Solution
  - Find All Non-Dominated Solutions (Pareto Set)

Multi Objective Optimization (MOO): Decision Making Process

- MOO Decision Making Process



MORNINGSTAR®

Multi Objective Optimization (MOO): Decision Making Process

- MOO Decision Making Process

DM Method	DM Preference Stage	Pareto Set Needed	Use Case
A Priori	Before	No	DM know the preference of objective
A Posteriori	After	Yes	DM not clear about objective preference
Interactive	Middle	No	DM know the objective preference
No DM	Not available	Yes	DM is not available

Multi Objective Optimization (MOO): Algorithms Summary

- Scalarization Algorithms
  - Transform multi-objectives into a single objective
  - Solve it by single objective optimizer
  - Find one Pareto optimal solution in one run
  - Find Pareto Set in multiple run

- Multi Objective Evolutionary Algorithms (MOEA)
  - Follow natural evolution process such as gene evolution, a flock of birds seeking food and other resources, a cooling process of melted crystal, . . .
  - Find multiple Pareto optimal solutions in one run

Contents

- Background and History
- Multi Objective Optimization (MOO)
- MOO Algorithms
  - Scalarization Algorithms
  - Multi Objective Evolutionary Algorithms
  - Algorithm Evaluation Based on Pareto Set
- Selection of a Single Solution in Pareto Set
- MOO Libraries
- Summary & QA

Scalarization Algorithms

- Weighting Methods
- ϵ-Constraint Method
- Normal Boundary Intersection (NBI) & Normal Constraint (NC)
- Goal Programming
- Physical Programming
- Lexicographic Method

Scalarization Algorithms: Weighted Sum Methods

- Weighted Sum Method
  - A weight vector based on DM preference of each objectives:

$$\min_{x} \sum_{i=1}^{M} w_i f_i(x)$$

subject to $x \in S$

Where $\sum_{i=1}^{M} w_i = 1$ and $w_i > 0$

  - The condition of the weights guarantees Pareto optimal

MORNINGSTAR®

Scalarization Algorithms: Weighted Sum Methods

- Example: Two Objective Metrics Recommender Systems

$$\min_{x \in S}(f_1, f_2), \qquad f_1 = 1 - \text{accuracy},\ f_2 = 1 - \text{diversity}$$

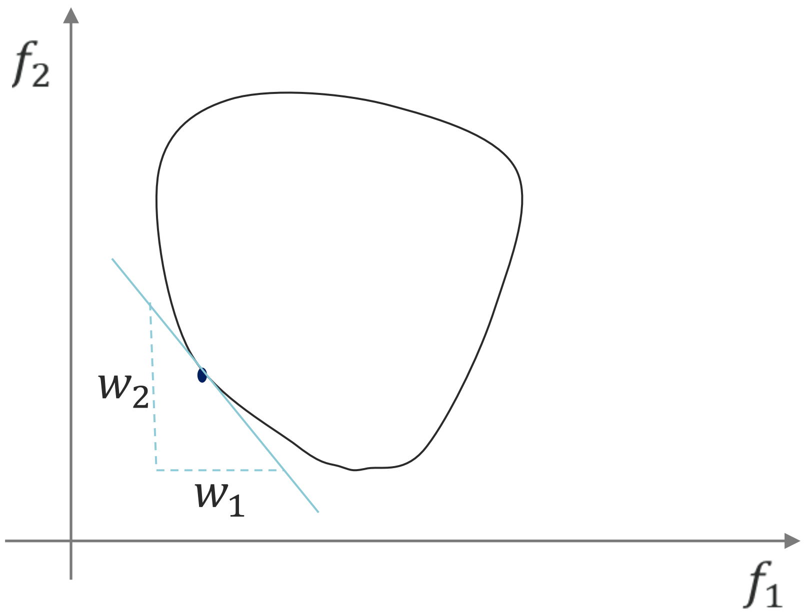
Solve: $$\min_{x \in S}(w_1 f_1 + w_2 f_2)$$

$$w_1 + w_2 = 1, w, w_2 > 0$$

- Each (w_1, w_2) gives one Pareto solution
- Question: Can we get all Pareto solutions in this way?

Scalarization Algorithms: Weighted Sum Methods

- Question: Can we get all Pareto solutions in this way?
  - Try all (w_1, w_2)?
- Answer:
  - Not guaranteed
- A Sufficient Condition[1]
  - S is convex in R^n
  - Each objective function $f_k(x)$ is convex



1. Yair Censor, *Pareto Optimality in Multiobjective Problems*, Applied Mathematics and Optimization 4, 41- 59

MORNINGSTAR®

Scalarization Algorithms: Weighted Sum Methods

- A nonconvex problem(https://commons.wikimedia.org/wiki/File:NonConvex.gif)

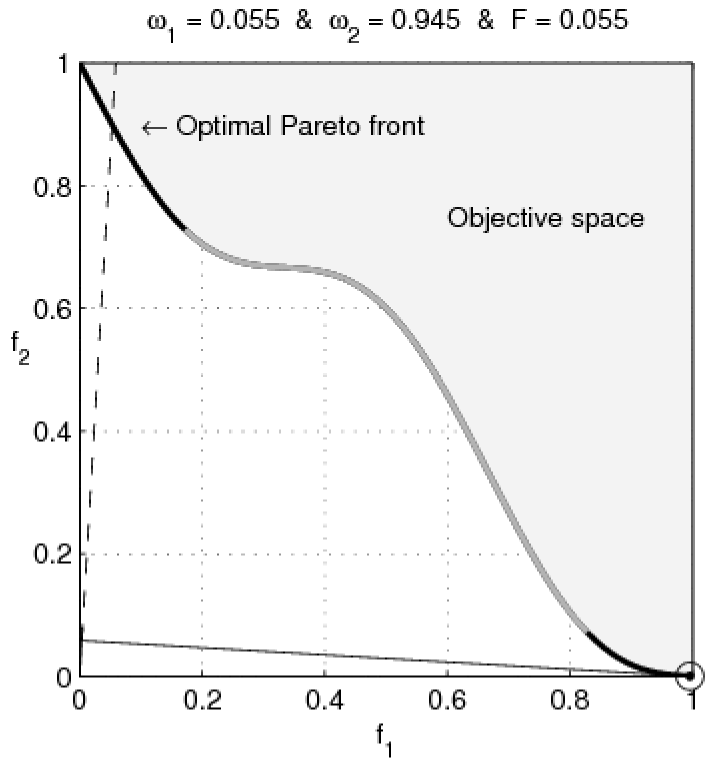
$$\min_{\boldsymbol{x}} F(x) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}))$$

Where $f_1(\boldsymbol{x}) = x_1$

$$f_2(\boldsymbol{x}) = 1 + x_2^2 - x_1 - 0.1\sin(3\pi x_1)$$

$0 \le x_1 \le 1$ and $-2 \le x_2 \le 2$

Here f_2 is not convex on S



MORNINGSTAR

Scalarization Algorithms: Other Weighting Methods

- Weighted Exponential Sum[1]

$$U = \sum_{i=1}^{M} w_i [f_i(x)]^p \ , f_i(x) > 0 \ \text{ for all } i = 1, 2, \dots, M$$

where $1 \leq p < \infty$, $\sum_{i=1}^{M} w_i = 1$ and $w_i > 0$

1) The condition of the weights guarantees the solution is Pareto optimal[1]
2) Bigger p increase the effectiveness of finding all Pareto solutions

1. P. L. Yu, *A Class of Solutions for Group Decision Problems*, Management Science, Vol. 19, No. 8, Application Series (Apr., 1973), pp. 936-946

Scalarization Algorithms: Other Weighting Methods

- Weighted Metric Methods[1]

$$U = \left[\sum_{i=1}^{M} w_i^p \, |f_i(x) - f_i^*|^p\right]^{\frac{1}{p}}$$

where $1 \leq p < \infty$, $\sum_{i=1}^{M} w_i = 1$ and $w_i > 0$

$f^* = (f_1^*, f_2^*, \ldots, f_M^*)$ is ideal point in objective space

1) f^*: Utopian point (min value of each objective), goal point (specified by DM) or 0
2) The condition of the weights guarantees the solution is Pareto optimal
3) Bigger p increase the effectiveness of finding all Pareto solutions

1. P. L. Yu and G. Leitmann, *Compromise Solutions, Domination Structures, and Salukvadze's Solution*, Journal of Optimization Theory and Applications: Vol. 13, No. 3, 1974

Scalarization Algorithms: Other Weighting Methods

- Weighted Chebyshev method[1]

$$U = \max_{i} \{w_i |f_i(x) - f_i^*|\}$$

where $\sum_{i=1}^{M} w_i = 1$ and $w_i > 0$

1) Taking $p \to \infty$ in $\left[\sum_{i=1}^{M} w_i^{\,p} |f_i(x) - f_i^*|^p\right]^{\frac{1}{p}}$

2) Can get complete Pareto set by changing weights without convex conditions[2]

3) May get non-Pareto solutions

1. Michael R. Lightner and Stephen W. Director, *Multiple Criterion Optimization for the Design of Electronic Circuits*, IEEE Transactions on CIrcuil-S and Systems, Vol. Cas-28, No. 3, March 1981
2. A. Messac and others. *Ability of Objective Functions to Generate Points on Nonconvex Pareto Frontiers*, AIAA JOURNAL, Vol. 38, No. 6, June 2000

Scalarization Algorithms: Other Weighting Methods

- Exponential Weighted Criterion[1]

$$U = \sum_{i=1}^{K} (e^{p w_i} - 1) e^{p f_i(x)} \ , p \geq 1$$

Can find Pareto set in non-convex problem

- Weighted Product Method[2]

$$U = \prod_{i=1}^{K} |f_i(x)|^{w_i}$$

Minimize impact of different magnitude of objective function

1. Timothy Ward Athan & Panos Y. Papalambros, *A Note on Weighted Criteria Methods for Compromise Solutions in Multi-Objective Optimization*, Engineering Optimization, 27:2, 155-176
2. E. N. Gerasimov and V. N. Repko, *Multicriterial Optimization*, 1979, Plenum Publishing Corporation

Scalarization Algorithms: Weighting Methods Summary

Method	Formula	Pro	Con
Weighted Sum	$\sum_{i=1}^{M} w_i f_i(x)$	Simple	Require convex condition for Pareto set
Weighted Exponential Sums	$\sum_{i=1}^{M} w_i [f_i(x)]^p$	Increase p to approximate Pareto set	Bigger p may give non-Pareto solution
Weighted Metric Methods	$\left[\sum_{i=1}^{M} w_i^p \lvert f_i(x) - f_i^* \rvert^p\right]^{\frac{1}{p}}$	Different choice of ideal points	Bigger p may give non-Pareto solution
Weighted Chebyshev method	$\max_i \{w_i \lvert f_i(x) - f_i^* \rvert\}$	Can find complete Pareto set	Some solution may not Pareto optimal
Exponential Weighted Criterion	$\sum_{i=1}^{K} (e^{p w_i} - 1) e^{p f_i(x)}$	Can find complete Pareto set	May lead computation overflow
Weighted Product Method	$\prod_{i=1}^{K} \lvert f_i(x) \rvert^{w_i}$	Deal with different magnitude of objectives	Rarely used

Scalarization Algorithms: ϵ-Constraint

- $\boldsymbol{\epsilon}$-Constraint Method[1]

$$\min f_l(x)$$

subject to $f_i(x) \leq \epsilon_i$, for all $\boldsymbol{i} \neq l$

ϵ_i is a known upper bound of f_i

1) Choose different ϵ_i may produce all Pareto solutions

2) No convex requirement

3) May not Pareto optimal

1. Haimes, Lasdon, Wismer, *On a Bicriterion Formulation of the Problems of Integrated*, System Identification and System Optimization, IEEE Transactions on Systems, Man, And Cybernetics, July 1971

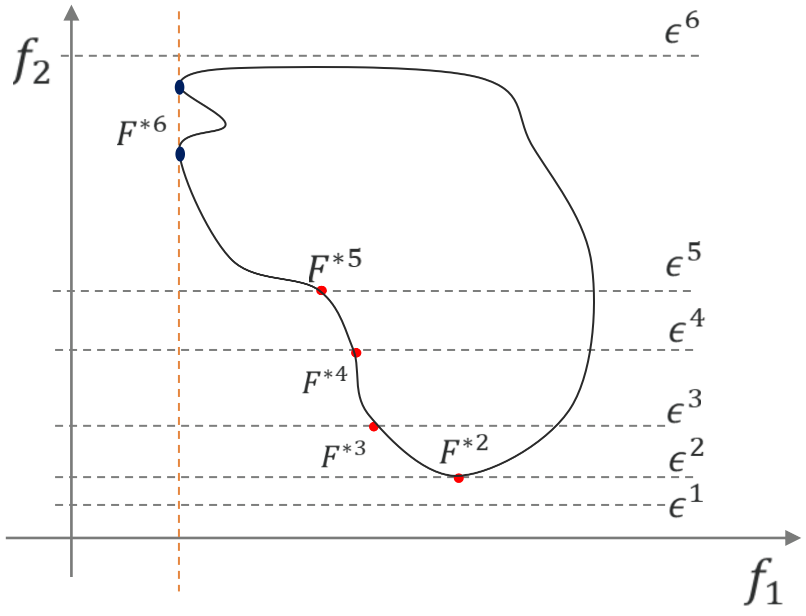
Scalarization Algorithms: ϵ-Constraint – a sufficient condition

- A sufficient condition of Pareto optimal[1]
  - If optimal solution x^* is unique
- Two objective example

solve

$$\min_{x \in S} f_1(x)$$
subject to $f_2 \leq \epsilon$

Constraint	Unique Solution	Pareto Optimal
ϵ^1	No solution	NA
$\epsilon^2, \epsilon^3, \epsilon^4, \epsilon^5$	Yes	Yes
ϵ^6	No	Not necessary



1. V. Chankong, Y. Haimes, Multiobjective Decision Making, Dover Publication, 1983

Scalarization Algorithms: NBI and NC

- Normal Boundary Intersection (NBI[1]) and Normal Constraint (NC[2])
  - Step 1: Find Anchor Points in objective space
  - Step 2: Define Utopia Line(hyperplane) connecting Anchor Points
  - Step 3: Set evenly distributed base points on Utopia Line
  - Step 4: Find Utopia line normal vector at each based point
  - Step 5: Optimize one objective in area above normal vectors at each base point

1. Das, Dennis, 1998: *Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems.* SIAM J. Optim. 8, 631–657
2. Messac, Ismail-Yahaya, Mattson, 2003: *The normalized normal constraint method for generating the Pareto frontier*, Struct Multidisc Optim 25, 86–98 (2003)

- Example of two objective MOO problem

$$\min_x F(x),\ F(x) = (f_1, f_2)$$

- Anchor Points: A_1 and A_2
  - A_1: x^{*1} only minimize f_1

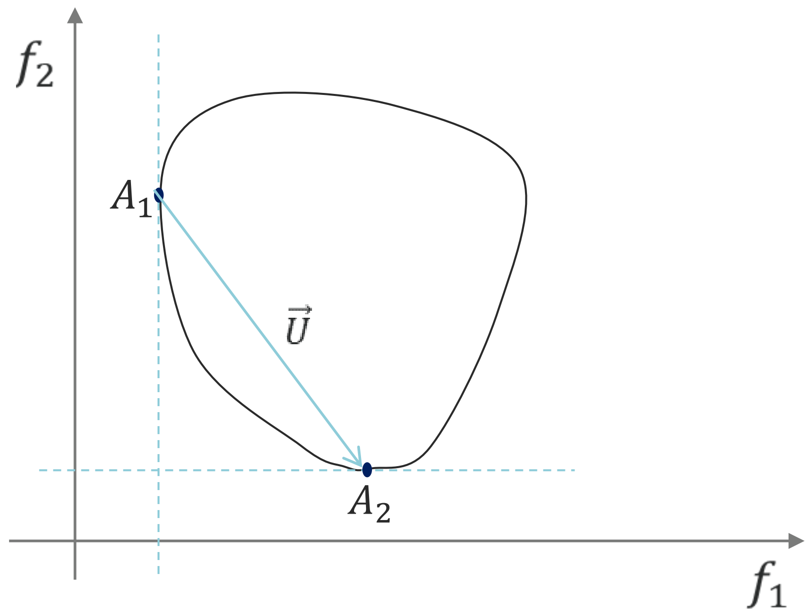
$$A_1 = \left(f_1^*(x^{*1}), f_2(x^{*1})\right)$$

  - A_2: x^{*2} only minimize f_2

$$A_2 = \left(f_1(x^{*2}), f_2^*(x^{*2})\right)$$

- Utopia Line Vector: $(\overline{A_1 A_2})$:

$$\vec{U} = A_2 - A_1$$



Scalarization Algorithms: NBI & NC Method

- Set evenly distributed base points (F_{pj}) on Utopia Line ($\vec{U}$)

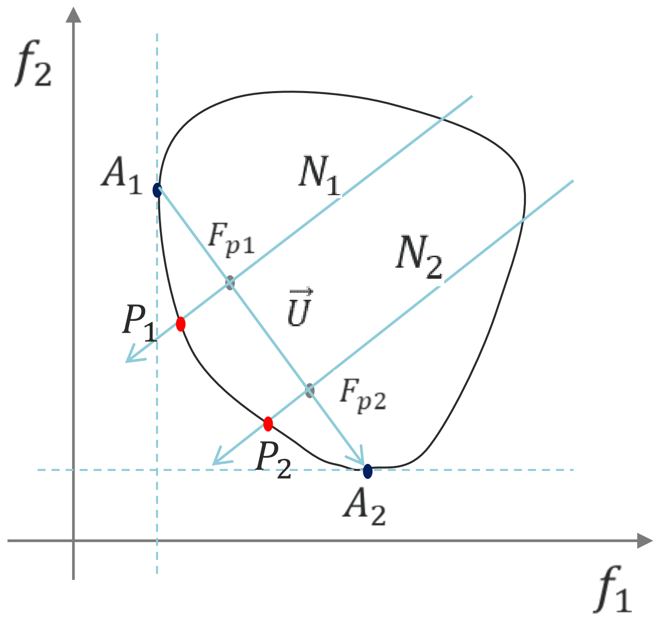
$$F_{pj} = \omega_{1j}A_1 + \omega_{2j}A_2$$

- Normal vector: $\vec{N}_1$, $\vec{N}_2$
- Optimize f_2 with new constraint

$$\min_x f_2(x)$$

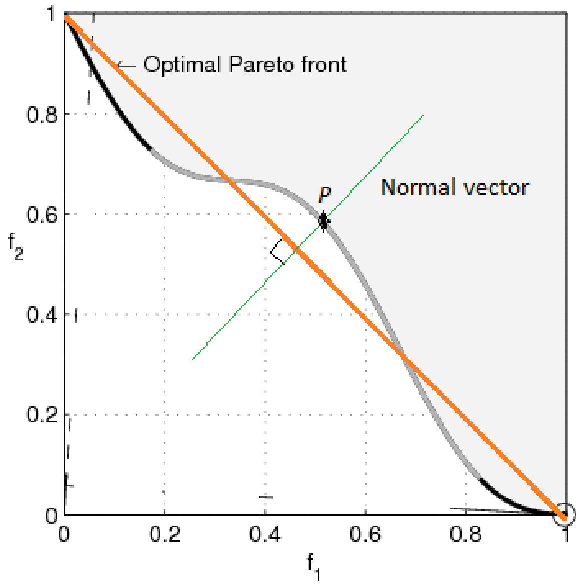
subject to $\vec{U} \cdot \left(F(x) - F_{pj}\right) \leq 0$,

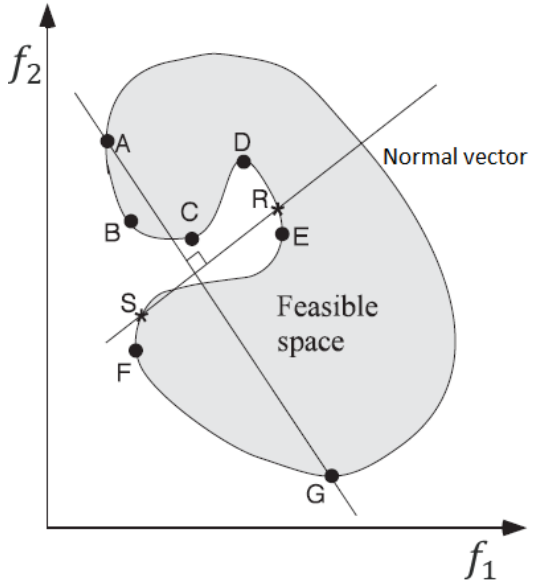
- Pareto Optimal: P_1 , P_2



Scalarization Algorithms: NBI & NC Method

- Works for non-convex case, may find Non-Pareto points: R





- Need to use filter to remove dominated points

Scalarization Algorithms: Other Methods

- Other Scalarization Methods

Method	Idea	Scalarization	Characteristic
Goal Programming[1]	Set up goal for each objective	$\min \sum_{i=1}^{M} \lvert d_i \rvert$	May not be Pareto optimal
Physical Programming[2]	Map goals and objective to utility functions $\bar{g}_i$	$\min \log_{10} \sum \bar{g}_i$	Pareto optimal Need detail knowledge of each objective
Lexicographic Method[3]	Order each objective by importance	Minimize each objective in order	The solution may not be feasible

1. A. Charnes and W.W. Cooper, *Goal programming and multipleobjective optimizations*, European Journal of Operational Research I (1977) 39-54
2. Achille Messac, *Physical Programming: Effective Optimization for Computational Design*, AIAA JOURNAL Vol. 34, No. 1, January 1996
3. Peter C. Fishburn, *Exceptional Paper—Lexicographic Orders, Utilities and Decision Rules: A Survey*. Management Science 20(11):1442-1471 (1974)

Scalarization Algorithms: Summary

- Major scalarization methods summary

Method	be a Pareto Solution	To Get All Pareto Solutions	Other
Weighting methods	$\sum_{i=1}^{M} w_i = 1$ and $w_i > 0$	Convex for weighed sum	Some methods may get non-Pareto solution
ϵ-Constraint	Solution is unique	Solution is unique	May not get solution
Normal Bonded Intersection & Constraint	Pareto front is Convex	Pareto front is Convex	May get non-Pareto solution in concave case
Goal Programming	No guarantee	Not available	
Physical Programming	Guaranteed	Guaranteed	
Lexicographic Method	Not available	Not available	

Contents

Multi Objective Evolutionary Algorithms

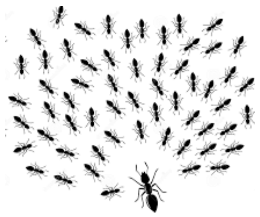
- Evolutionary Algorithms inspired by natural evolutionary process:

Genetic Algorithm (GA)

Particle Swam Optimization (PSO)

Simulated Annealing (SA)

Ant Colony Optimal (ACO)

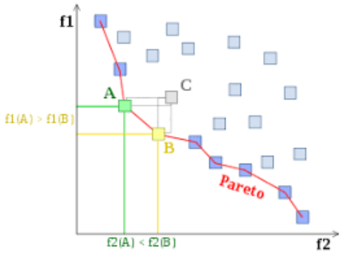


Multi Objective Evolutionary Algorithms

- Benefits:

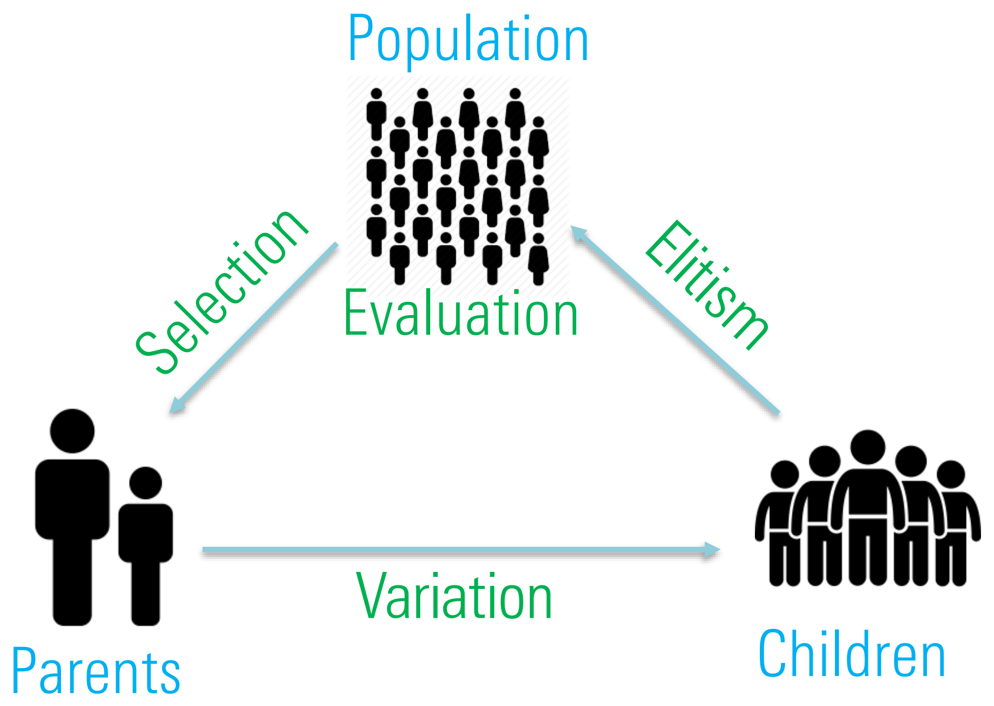
Objective can be any function

Parallel computing



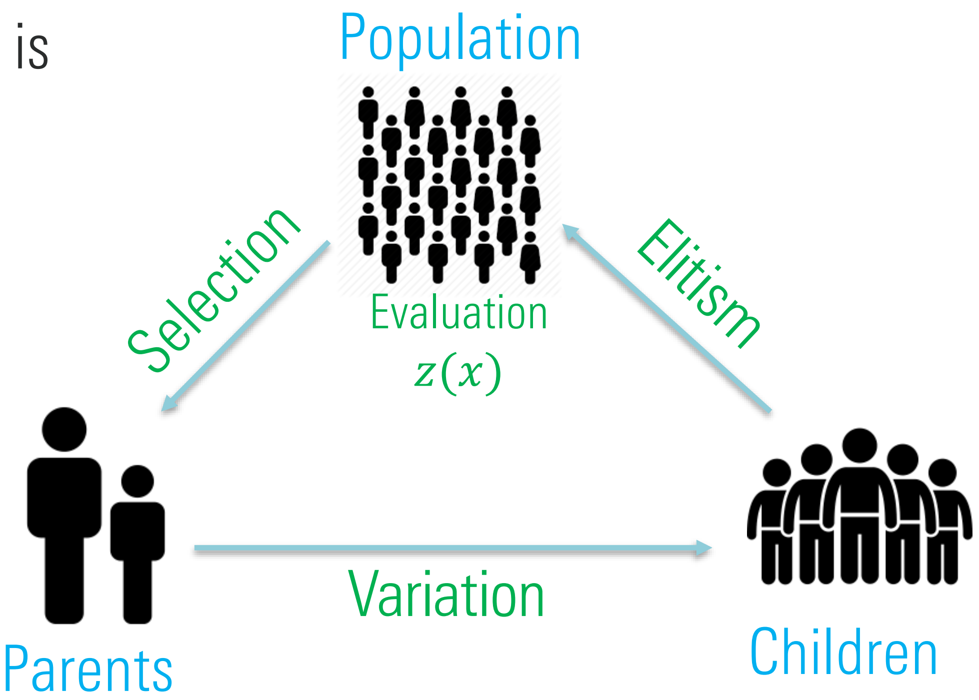
Get Pareto Set in one run

- Terminologies of Solutions:
  - Individual : a feasible solution x
  - Population: a set of individuals
  - Parents: selected from Population
  - Children: produced from Parents

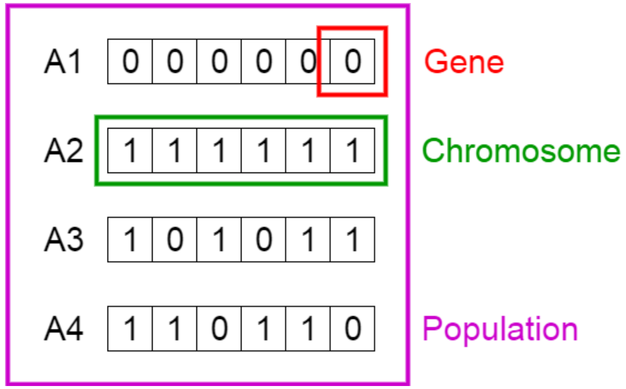


Evolutionary Algorithms : Basic Concepts

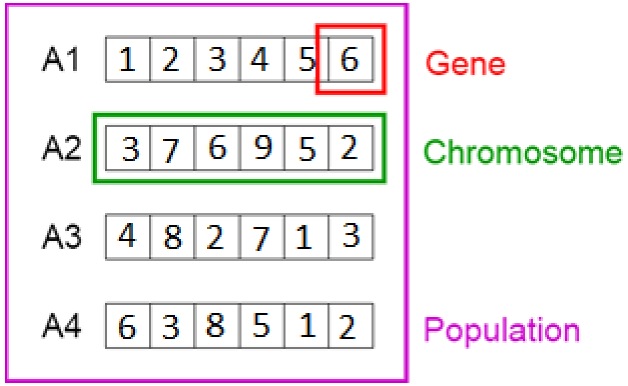
- Operators of Genetic Algorithm
  - Evaluation: measure how 'good' each solution is
    - assigning fitness value (or order): $z(x)$
  - Selection: find Parents
    - Random process
    - Tournament process
  - Variation: produce children
    - Crossover
    - Mutation
  - Elitism:
    - maintain 'better' solution in each iteration



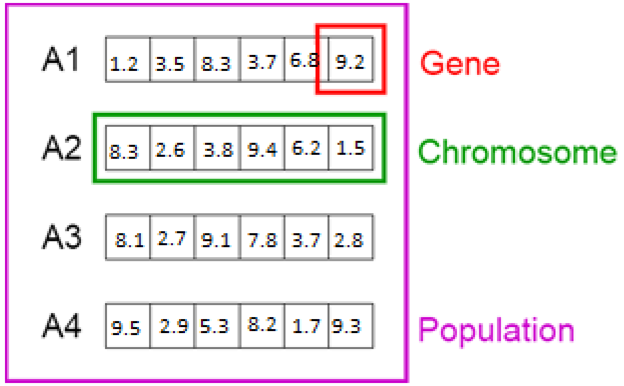
- Binary Encoding in Recommender System
  - Chromosome: a recommendation list (decision variable x)
  - Length of Chromosome = total number of available items
  - Each Gene position is corresponding an item
    - 1: item is in recommender list ,
    - 0: item is not in recommender list



- Permutation Encoding in Recommender System
  - Length of Chromosome = N in top N recommendation list
  - The value of each Gene: index of an item
    - 6: the 6th item
    - Total 9 available items
    - $N = 6$

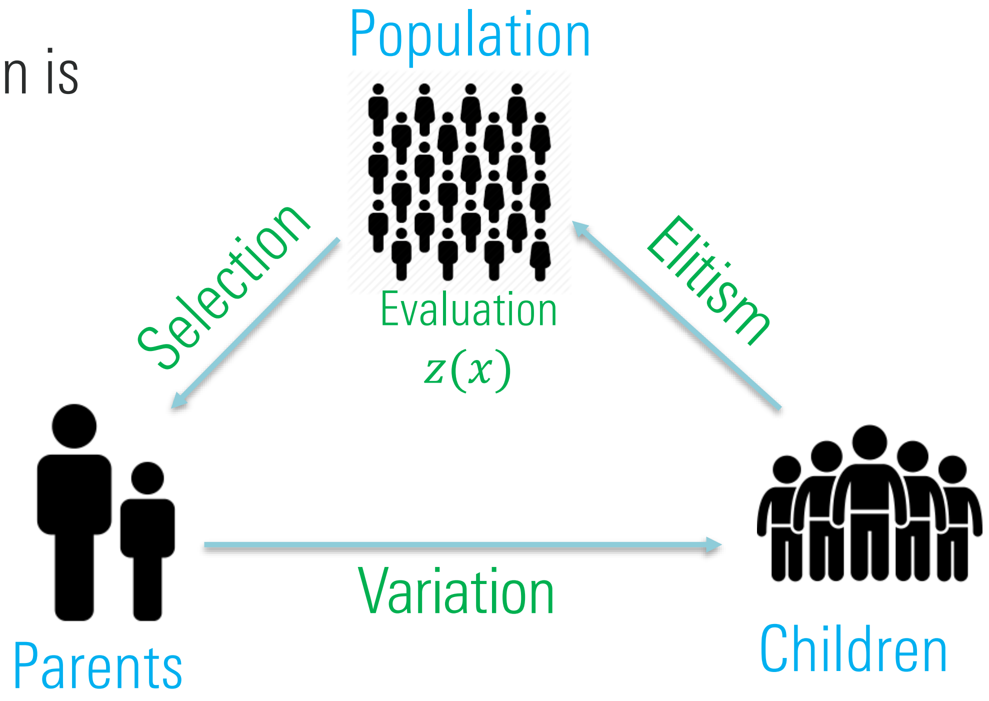


- Real Value Encoding for parameters
  - Chromosome: a set of parameters (such as weights)
  - Length of Chromosome = Number of parameters to be optimized
  - Value in the Gene: value of the parameter
    - 6 parameters here
    - Parameter value is real value in $[1, 10]$



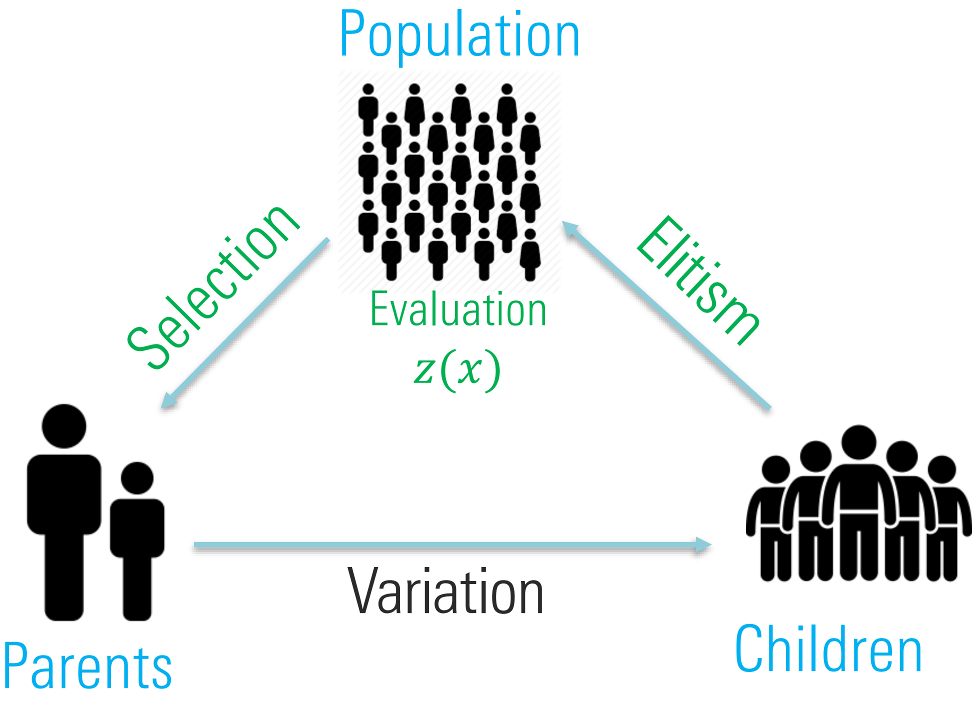
Summary: Single Objective Evolutionary Algorithms

- Operators
  - Evaluation: measure how ‘good’ each solution is
    - assigning fitness value (or order): $z(x)$
  - Selection: find Parents
    - Random process
    - Tournament process
  - Variation: produce children
    - Crossover
    - Mutation
  - Elitism:
    - maintain ‘better’ solution in each iteration



From Single Objective to Multi Objective Evolutionary Algorithms

- Solve a MOO problem
  - Find non-dominated solutions (Pareto optimal)
- Where are multi objective considered? Where are dominance relations applied?
  - Fitness value $z(x)$ evaluation
  - Parent selection
  - Elitism



Multi Objective Evolutionary Algorithms:

- Major MOO Genetic Algorithm Methods

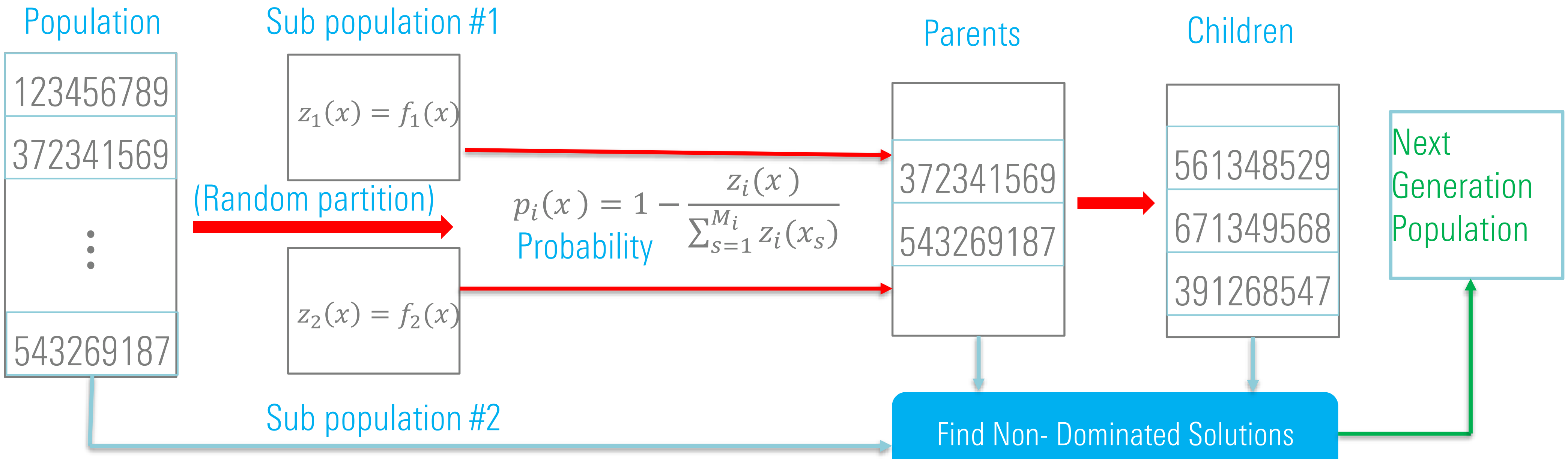
Method	Fitness Evaluation	Parent Selection	Elitism
VEGA (Schaffer,1985)	Single objective	Probability distribution	Dominance relation
MOGA (Fonseca & Fleming,1993)	Dominance relation	Probability distribution	NA
NSGA (Srinivas and Deb,1994)	Dominance relation	Probability distribution	NA
NSGA-II (Debb, etc., 2002)	Dominance relation	Probability distribution	Dominance relation
NPGA (Horn, etc. ,1994)	No Fitness	Tournament method (dominance relation)	NA
PAES (Knowles and Corne, 1999)	No Fitness	Local search (dominance relation)	Dominance relation

Classification of Multi Objective Genetic Algorithms

- *Dominance* relation in Elitism
  - VEGA (Schaffer,1985)
- *Dominance* relation in fitness function
  - MOGA (Fonseca & Fleming,1993),
  - NSGA (Srinivas and Deb,1994) ,
  - NSGA-II (Debb, etc., 2002)
- No fitness values, but *dominance* relation in selection/elitism process
  - NPGA (Horn, etc. ,1994),
  - PAES (Knowles and Corne, 1999)

Multi Objective Evolutionary Algorithms: VEGA

- Vector Evaluated Genetic Algorithm (VEGA[1])
  - First genetic algorithm applied to MOO: $\min_{x \in S}(f_1, f_2)$
  - Fitness value, $z_i(x)$, is based on objective function $f_i(x)$



1. Schaffer, 1985: *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms,* The First International Conference on Genetic Algorithms and their Applications (held in Pittsburgh), pp. 93–100

Multi Objective Evolutionary Algorithms: MOGA

- Multi-Objective Genetic Algorithm (MOGA[1])
  - Rank all solutions: $r(x) = 1 +$ (# of solutions that dominates x)
  - Fitness value:

$$z(x) = N - \sum_{k=1}^{r(x)-1} n_k - 0.5\left(n_{r(x)} - 1\right)$$

N =5, Total number of solutions

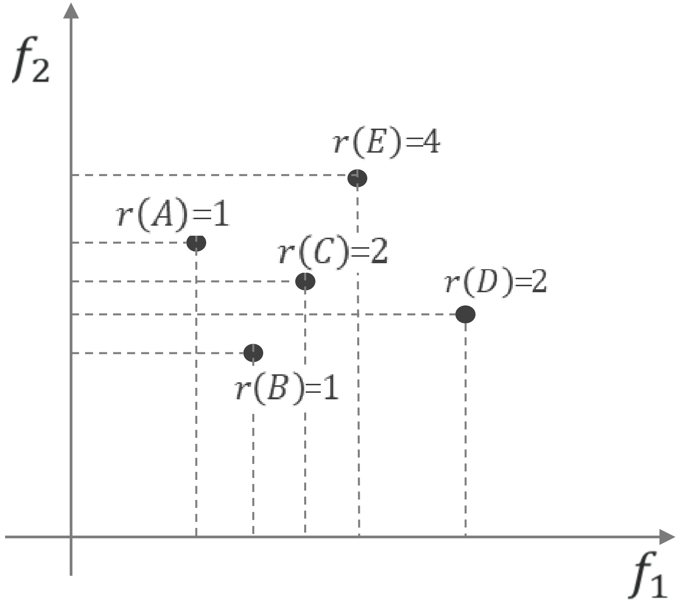
n_k =Number of solutions with $r(x) = k$:

$$n_1 = 2, \quad n_2 = 2, \quad n_3 = 0, \quad n_4 = 1$$

$z(E) = 5 - (2 + 2) - 0.5(1 - 1) = 1$

$z(C) = z(D) = 5 - (1) - 0.5(2 - 1) = 3.5$

$z(A) = z(B) = 5 - 0 - 0.5(2 - 1) = 4.5$



1. Fonseca and Fleming, 1993: *Multiobjective Genetic Algorithm*, IEEE colloquium on 'Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130), 28 May 1993. London, UK: IEE; 1993

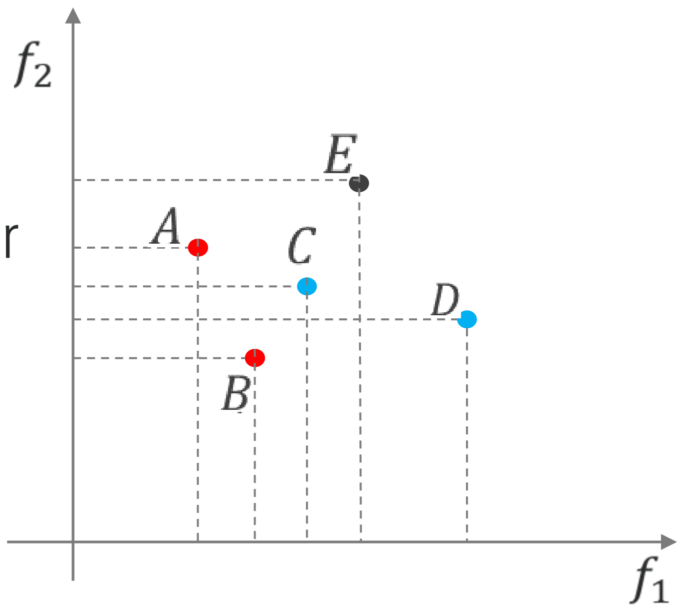
Multi Objective Evolutionary Algorithms: NSGA

- Nondominated Sorting Genetic Algorithm (NSGA[1])
  - Sorting by selecting non-dominated solutions without replacement each time:

    $P_0 = \{A, B\}$, $P_1 = \{C, D\}$, $P_2 = \{E\}$

    P_0 dominates P_1 dominates P_2
  - Assign same fitness value $z(x)$ for each x in P_i based on order

    P_0: $z(A) = z(B) = 10$
    P_1: $z(C) = z(D) = 8$
    P_2: $z(E) = 5$



1. Srinivas and Deb, 1994: *Multiobjective optimization using nondominated sorting in genetic algorithms*. Journal of Evolutionary Computing 1994;2(3):221–48.

Multi Objective Evolutionary Algorithms: NSGA-II

- Nondominated Sorting Genetic Algorithm II (NSGA-II[1])
  - Sorting by number of dominated solutions

    $P_1 = \{A, B\}$, 0 dominated solution

    $P_2 = \{C, D\}$, 1 dominated solution

    $P_3 = \{\}$, 2 dominated solutions

    $P_4 = \{E\}$, 3 dominated solutions

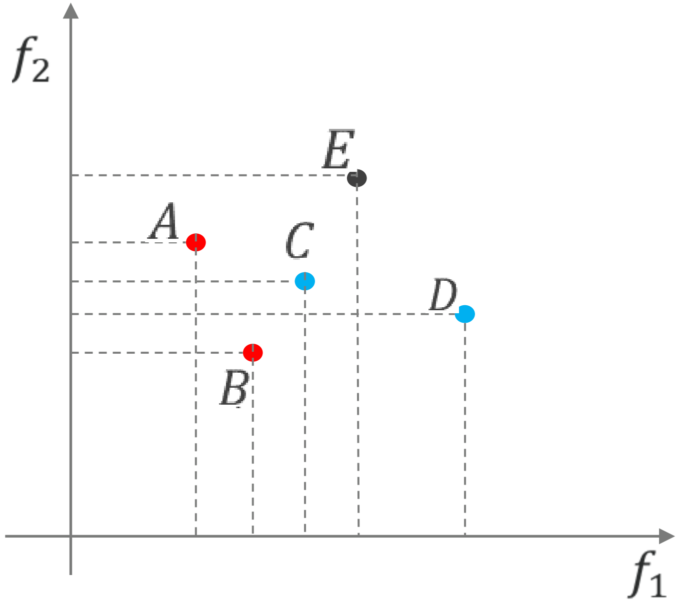
  - Assign same rank value n_x for each x in P_i based on order

    $P_1 : n_A = n_B = 1$

    $P_2 : n_C = n_D = 2$

    $P_4 : n_E = 3$

  - All solutions are ranked by pair $(n_x\ d_x)$, d_x is a crowding distance



1. Deb, Pratap, Agarwal, Meyarivan, 2002: *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Trans Evolutionary Computing 2002;6(2):182–97.

Multi Objective Evolutionary Algorithms: NPGA

- Niched Pareto Genetic Algorithm (NPGA[1])
  - Use tournament method for Parent selection

Selection: for randomly pair $x, y \in P$,

randomly choose $G = \{C\}$

choose a 'better one' comparing with G:

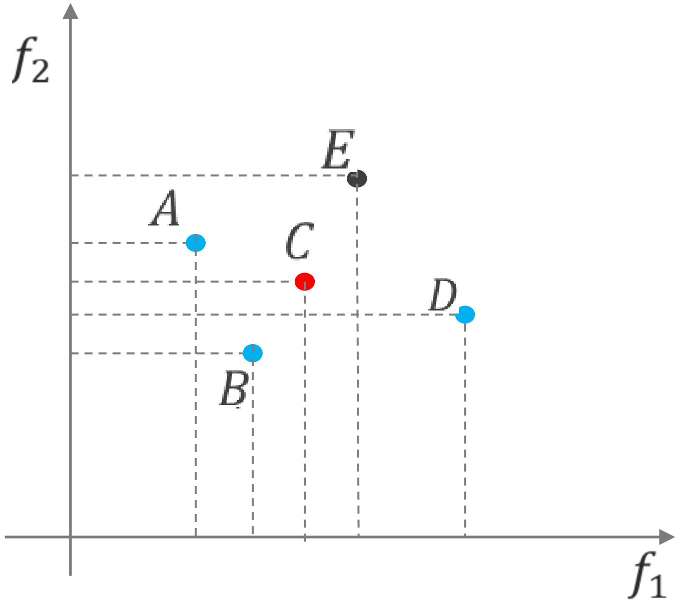
(A, E): E is dominated by G ⟶ choose A

(B, E): E is dominated by G ⟶ choose B

(B, D): neither B or D is dominated by G ⟶ choose D

(D is in less crowded area)

Parent = $\{A, B, D\}$



1. Horn, Nafpliotis and Goldberg A, 1994: *A niched Pareto genetic algorithm for multiobjective optimization*. Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, 27–29 June, Orlando, FL, USA, 1994.

Multi Objective Evolutionary Algorithms: PAES

- Pareto Archived Evolution Strategy (PAES[1])
  - Selection Operator

    Archived Solutions: G = {Non-Dominated Solutions discovered} ,
    initialize $G = \emptyset$

    Local Search:

    Random select A_0 as a Parent, $G = \{A_0\}$

    Mutate $A_0 \longrightarrow A_0'$, dominated by A_0, discard A_0'

    Mutate $A_0 \longrightarrow A_1$, dominates A_0 , be a new Parent

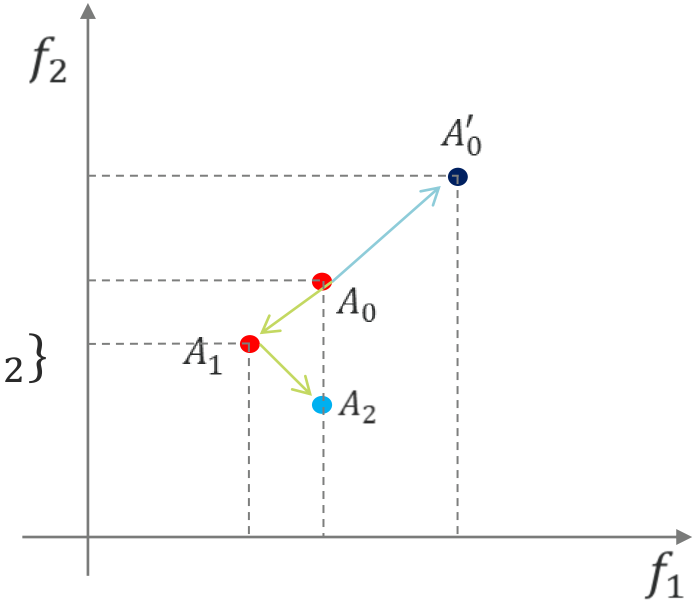
    Mutate $A_1 \longrightarrow A_2$, A_1 and A_2 are equally 'good'

    Update Archive G:

    Replace A_0 in G by a better A_2 (if A_2 dominates A_0) and $G = \{A_2\}$

    or

    Choose either A_1 or A_2 (based on crowdedness) to add to G if A_2 does not dominate G



1. Knowles and Corne, 1999: *The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization*. Proceedings of the 1999 congress on evolutionary computation, 6–9 July 1999. Washington, DC USA

Multi Objective Evolutionary Algorithms: Special Considerations

- Inherited weakness of EA
  - Converge towards local optima rather than the global optimum
- How to maintain solution diversity?
  - Niche Count[1]: $nc(x)$
    - More crowed around x $\longrightarrow$ bigger $nc(x)$
  - Fitness Sharing[1]: discount fitness value with niche count:

$$z'(x) = \frac{z(x)}{nc(x)}$$

1. David E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley Publishing Company, 1989

Multi Objective Evolutionary Algorithms: Other EAs

- MOEA methods based on other EAs
  - Particle Swam Optimization (PSO)

    James Kennedy and Russell C. Eberhart. *Particle swarm optimization*. Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, New Jersey, 1995

    M. Reyes-Sierra and C. Coello, *Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art*, International Journal of Computational Intelligence Research, Vol.2, No.3 (2006), pp. 287–308
  - Simulated Annealing (SA)

    Paolo Serafini, *Simulated Annealing for Multi Objective Optimization Problems*, Multiple Criteria Decision Making, Springer-Verlag New York, Inc. 1994
  - Ant Colony Optimal (ACO)

    Marco Dorigo, Gianni Di Car0, *Ant Colony Optimization: A New Meta-Heuristic*, Proceedings of the 1999 Congress of Evolutionary Computation - CEC99

Algorithm Evaluation Based on Pareto Set

- What is a good approximation of Pareto Set
- Hypervolume Indicator[1] $H(A)$
  - $A = \{a_1, a_2, a_3, a_4\}$ is the solution set in objective space
  - r is the reference, anti-utopia point (worst of each objective)
  - $H(A)$ is area covered by cubes defined by a_i and r:

    $$H(A) = \text{area}(\cup c(a_i, r))$$

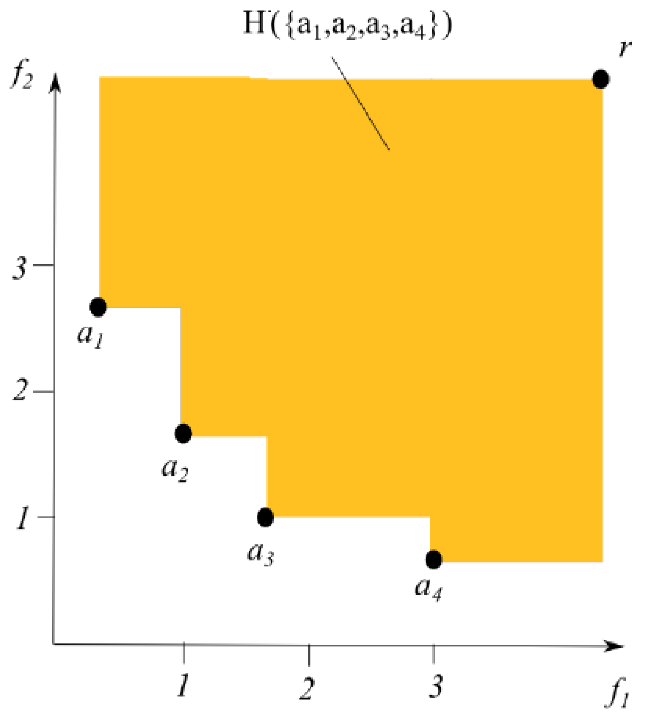


Image Source: ResarchGate

A bigger $H(A)$ represents a better solution set

1. Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele, The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration, EMO 2007, LNCS 4403, pp. 862–876, 2007.

Contents

MORNINGSTAR®

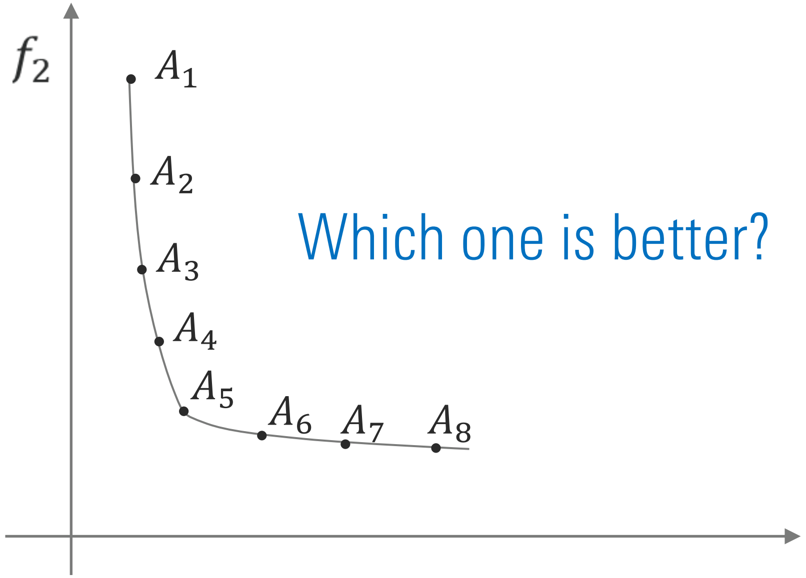
Selection of a Single Solution in Pareto Set: No DM Available

- When do we need to produce Pareto set?
  - Most MOEA method (A posterior, No DM available)
  - Scalarization without DM preference (A posterior, No DM available)

- Example: Recommender Systems balancing multi metrics
  - End user (DM) cannot choose from Pareto set
  - A single best recommendation list needs to be produced

Selection of a Single Solution in Pareto Set: Best Guess

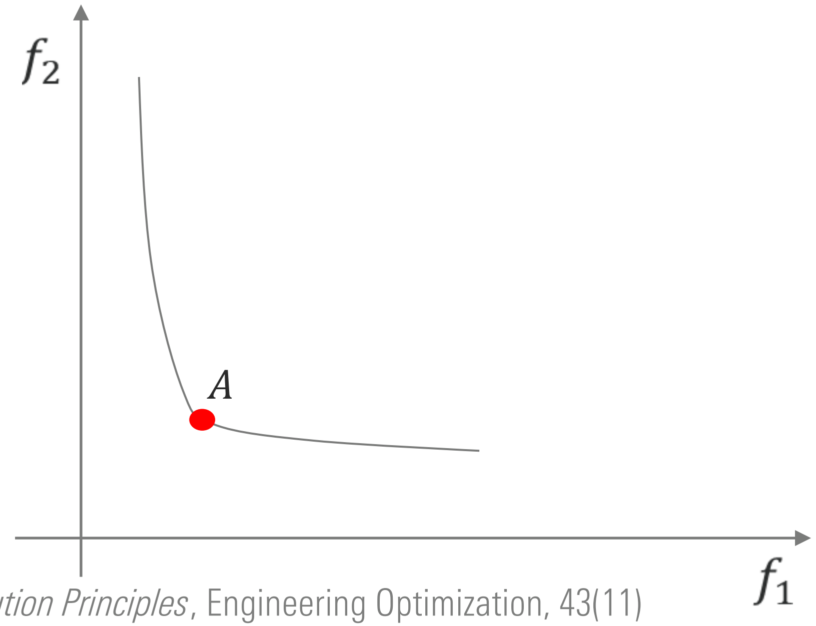
- No information from decision maker (DM)
  - All solutions in Pareto set are ‘equally good’
  - Need to make the ‘best’ guess
- Best guess method
  - Knee point method
  - Hypervolume Method
  - Multiple-criteria decision-making (MCDM) methods



Which one is better?

Selection of Single Solution in Pareto Set: Knee Point

- Knee Point
  - A special point (A) on Pareto Front
  - small improvement in either objective will cause a large deterioration in the other objective,
- Methods of Finding Knee Point
  - Angle Based Method[1]
  - Marginal Utility Method[1]
  - Hyperplane Normal Vector Method[2]



1. Deb and Gupta, *Understanding Knee Points in Bicriteria Problems and Their Implications as Preferred Solution Principles*, Engineering Optimization, 43(11)
2. Yu, Jin, Olhofer, A Method for a Posteriori Identification of Knee Points Based on Solution Density, 2018 IEEE Congress on Evolutionary Computation (CEC)

Selection of a Single Solution in Pareto Set: Angle Based Knee Point

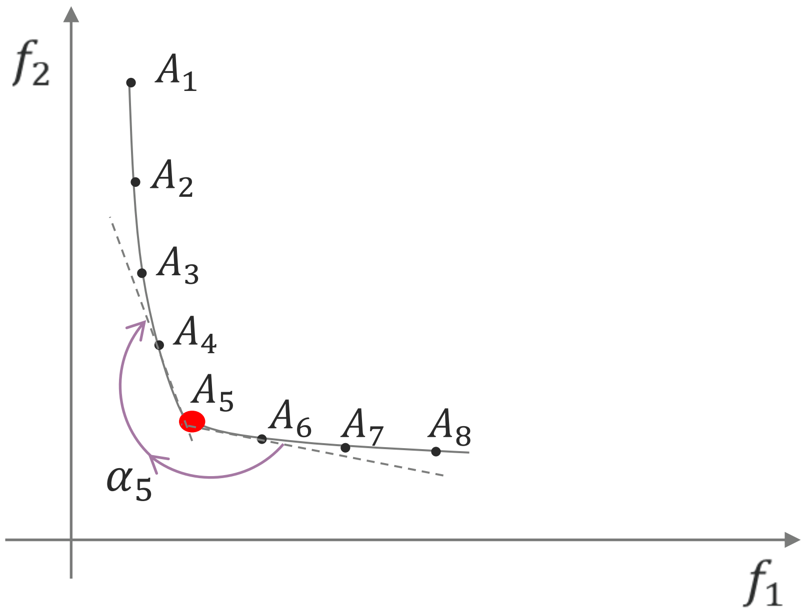
- Angle Based Knee Point[1]
  - Only work in two objectives MOO
  - Pareto front: $\{A_1, A_2, \dots, A_8\}$
  - Calculate reflex angle α for each point:

$$\alpha_i = \angle A_{i-1} A_i A_{i+1}$$

  - Find the point with $\max \alpha_i$

$$\alpha_5 = \max\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$$

A_5 is the Knee point



1. Deb and Gupta, *Understanding Knee Points in Bicriteria Problems and Their Implications as Preferred Solution Principles*, Engineering Optimization, 43(11)

MORNINGSTAR

Selection of a Single Solution in Pareto Set: Hypervolume Method

- Hypervolume Method[1]
  - Define reference point r: worst of each objective (Upper Bound Point)

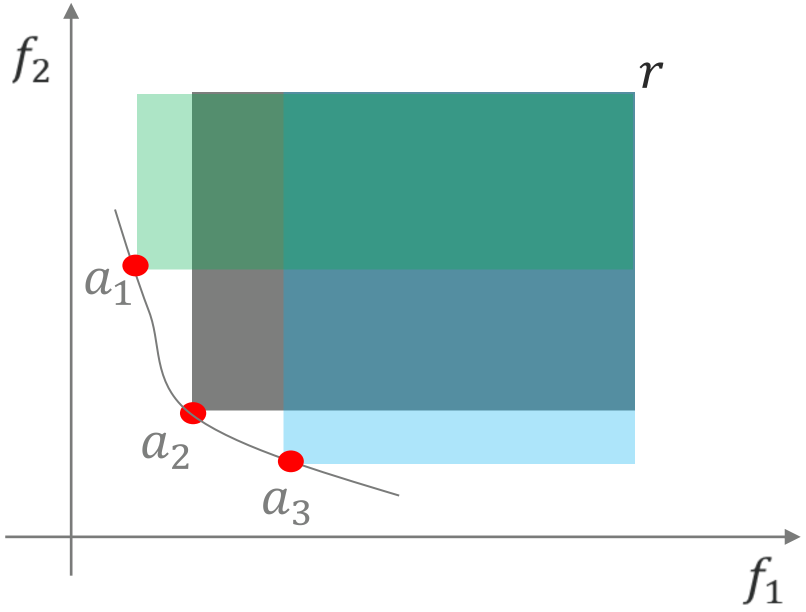
$$r = F^- = (f_1^-, f_2^-, \dots, f_M^-)$$

  - Evaluate Hypervolume for each solution $\{a_i\}$:

$$H(a_i), i = 1, 2, \dots, N$$

  - Best solution with max hypervolume

$$arg \max_i H(a_i)$$



1. Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele, The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration, EMO 2007, LNCS 4403, pp. 862–876, 2007.

Selection of a Single Solution in Pareto Set: MCDM Methods

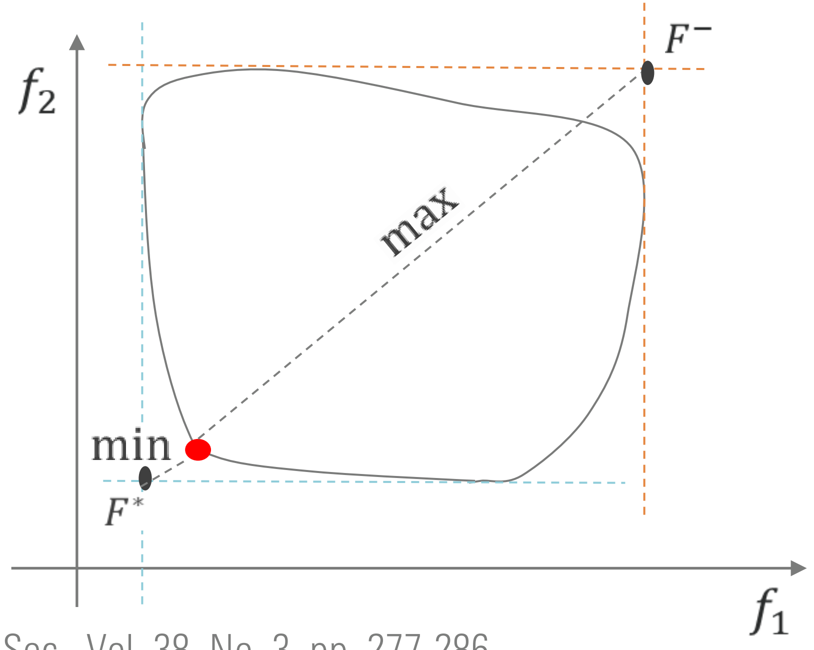
- Multiple-criteria decision-making (MCDM) methods[1]:
  - Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)
  - Preference Ranking Organization METHod for Enrichment of Evaluations (PROMETHEE)
  - Least misery method
  - ÉLimination et Choix Traduisant la REalité (ÉLECTRE)
  - …
- Most MCDM methods still need DM preference

1. https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis

Selection of a Single Solution in Pareto Set: TOPSIS

- Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS[1])
  - A Pareto solution point: that is most close to ideal point or most apart to anti-ideal point
  - Utopia point: $F^* = (f_1^*, f_2^*, \dots, f_M^*)$
  - Upper bound point: $F^- = (f_1^-, f_2^-, \dots, f_M^-)$
  - Find best Solution x in Pareto set:

$$\min_x ||F(x) - F^*||$$
$$\max_x ||F(x) - F^-||$$



1. Kwangsun Yoon, *A Reconciliation among Discrete Compromise Solutions*, J. opl Res. Soc., Vol. 38, No. 3, pp. 277-286, 1987.

MORNINGSTAR

Selection of a Single Solution in Pareto Set: PROMETHEE

- Preference Ranking Organization METHod for Enrichment of Evaluations (PROMETHEE), e.g., pairwise ranking,

$d_k(a,x) =$ $f_k(a) - f_k(x)$

Difference between each objective

$\pi_k(a,x)$ $\in [0,1]$

Min-max Normalization

$\pi\ (a,x) =$ $\sum_{k=1}^{M} \pi_k(a,x) w_k$

Weighted Sum of Difference

$\phi^+(a) =$ $\frac{1}{|P|-1} \sum_{x \in P} \pi(a,x)$

$\phi^-(a) =$ $\frac{1}{|P|-1} \sum_{x \in P} \pi(x,a)$

+, -

Preference Score

$\phi\ (a) =$ $\phi^+(a) - \phi^-(a)$

Net Preference Score

- Need weights for each objective

Contents

MORNINGSTAR®

MOO Libraries

Name	MOO Methods	Single Objective	Language	Open Source	Last update	Link
PyGMO	NSGA-II, MOEA, MH-AOC, NS-PSO	Yes	Python	Yes	2021	https://esa.github.io/pygmo2/index.html
pymoo	NSGA-II, NSGA-III, MOEA	Yes	Python	Yes	2022	https://pymoo.org/
Inspyred	PAES, NSGA-II	Yes	Python	Yes	2019	https://pythonhosted.org/inspyred/
Platypus	NSGA-II, MOEA, SPEA2, MOEA/D, PSO, PAES, PESA2	No	Python	Yes	2022	https://platypus.readthedocs.io/
MOEA Framework	NSGA-II, NSGA-III, PAES, PESA2, SPEA2, MOEA, MO-PSO	Yes	Java	Yes	2022	http://moeaframework.org/
MATLAB & Simulink	MOEA, NSGA, SPEA2	Yes	Matlab	No	2021	https://www.mathworks.com/matlabcentral/fileexchange
openGA	NSGA-III	Yes	C++	Yes	2020	https://github.com/Arash-codedev/openGA

MOO Libraries: pymoo demo (https://pymoo.org/getting_started/index.html)

- Define Problem:

$$\begin{aligned}
\min\ & f_1(x) = 100\,(x_1^2 + x_2^2) \\
\min\ & f_2(x) = (x_1 - 1)^2 + x_2^2 \\
\text{s.t.}\ & g_1(x) = 2\,(x_1 - 0.1)\,(x_1 - 0.9)\,/\,0.18 \leq 0 \\
& g_2(x) = -20\,(x_1 - 0.4)\,(x_1 - 0.6)\,/\,4.8 \leq 0 \\
& -2 \leq x_1 \leq 2 \\
& -2 \leq x_2 \leq 2 \\
& x \in \mathbb{R}
\end{aligned}$$

```python
import numpy as np
from pymoo.core.problem import ElementwiseProblem

class MyProblem(ElementwiseProblem):

    def __init__(self):
        super().__init__(n_var=2,
                         n_obj=2,
                         n_ieq_constr=2,
                         xl=np.array([-2,-2]),
                         xu=np.array([2,2]))

    def _evaluate(self, x, out, *args, **kwargs):
        f1 = 100 * (x[0]**2 + x[1]**2)
        f2 = (x[0]-1)**2 + x[1]**2

        g1 = 2*(x[0]-0.1) * (x[0]-0.9) / 0.18
        g2 = - 20*(x[0]-0.4) * (x[0]-0.6) / 4.8

        out["F"] = [f1, f2]
        out["G"] = [g1, g2]


problem = MyProblem()
```

- Initiate the MOO Algorithm

```python
from pymoo.algorithms.moo.nsga2 import NSGA2
from pymoo.operators.crossover.sbx import SBX
from pymoo.operators.mutation.pm import PM
from pymoo.operators.sampling.rnd import FloatRandomSampling

algorithm = NSGA2(
    pop_size=40,
    n_offsprings=10,
    sampling=FloatRandomSampling(),
    crossover=SBX(prob=0.9, eta=15),
    mutation=PM(eta=20),
    eliminate_duplicates=True
)
```

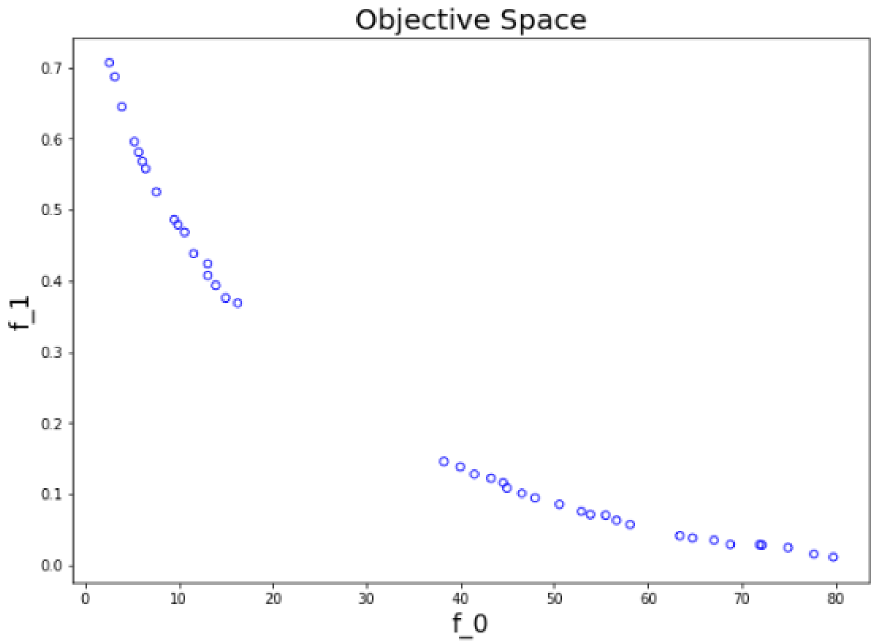
MOO Libraries: pymoo demo (https://pymoo.org/getting_started/index.html)

- Define Termination Condition:

```
from pymoo.termination import get_termination

termination = get_termination("n_gen", 40)
```

- Run Optimizer:

```
from pymoo.optimize import minimize

res = minimize(problem,
               algorithm,
               termination,
               seed=1,
               save_history=True,
               verbose=True)

# save results for later analysis
X = res.X  # Pareto Set
F = res.F  # Pareto Front
hist = res.history  # running history
```



MORNINGSTAR

MOO Libraries: pymoo demo (https://pymoo.org/getting_started/index.html)

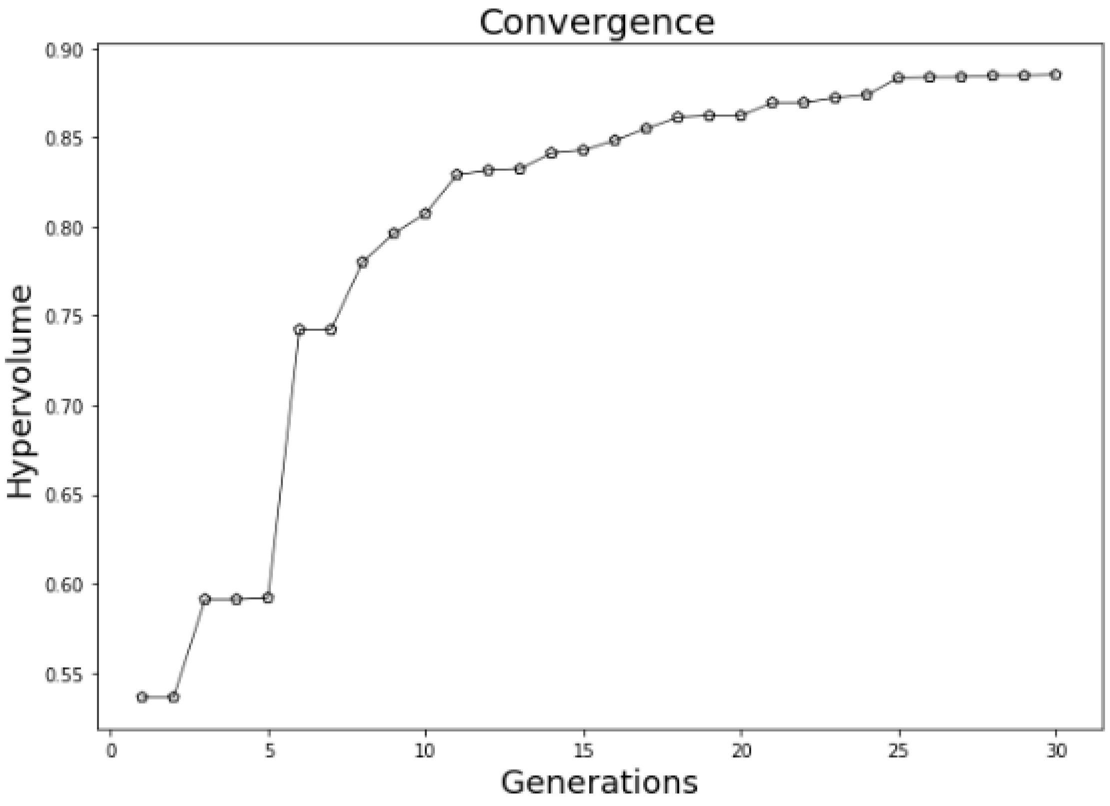
- Convergence: Collect history data

```python
n_evals = []             # corresponding number of function evaluations
hist_F = []              # the objective space values in each generation

for algo in hist:

    # store the number of function evaluations
    n_evals.append(algo.evaluator.n_eval)

    # retrieve the optimum from the algorithm
    opt = algo.opt

    # filter out only the feasible and append and objective space values
    feas = np.where(opt.get("feasible"))[0]
    hist_F.append(opt.get("F")[feas])
```

MOO Libraries: pymoo demo (https://pymoo.org/getting_started/index.html)

- Convergence: Check Hypervolume

```
# used for normalization of objective
approx_ideal = F.min(axis=0)
approx_nadir = F.max(axis=0)

ref_point = np.array([1.1, 1.1])

from pymoo.indicators.hv import Hypervolume

metric = Hypervolume(ref_point=ref_point ,
                     norm_ref_point=False,
                     zero_to_one=True,
                     ideal=approx_ideal,
                     nadir=approx_nadir)

hv = [metric.do(_F) for _F in hist_F]

plt.figure(figsize=(10, 7))
plt.plot(range(1, len(n_evals) + 1), hv,  color='black', lw=0.7, label="Avg. CV of Pop")
plt.scatter(range(1, len(n_evals) + 1), hv,  facecolor="none", edgecolor='black', marker="p")
plt.title("Convergence", fontsize=20)
plt.xlabel("Generations", fontsize=18)
plt.ylabel("Hypervolume", fontsize=18)
plt.show()
```



Summary, Q&A

- The solution of MOO problem must be Pareto Optimal
- A Pareto Set must be found if DM preference is not available
  - Multiple runs of Scalarization algorithms is needed to get Pareto set
  - A single run of Evolutionary Algorithms can get Pareto set
  - Evenly distributed Pareto set is needed
- One ‘best’ solution should be identified in Pareto set for DM in most cases