Multidisciplinary System Design Optimization (MSDO)

Multiobjective Optimization (II)

Lecture 15

Dr. Anas Alfaris

MOO 2 Lecture Outline

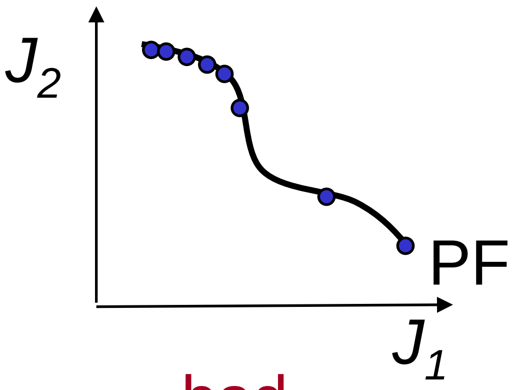
- Direct Pareto Front (PF) Calculation
  - Normal Boundary Intersection (NBI)
  - Adaptive Weighted Sum (AWS)
- Multiobjective Heuristic Programming
- Utility Function Optimization
- n-KKT
- Applications

Direct Pareto Front Calculation

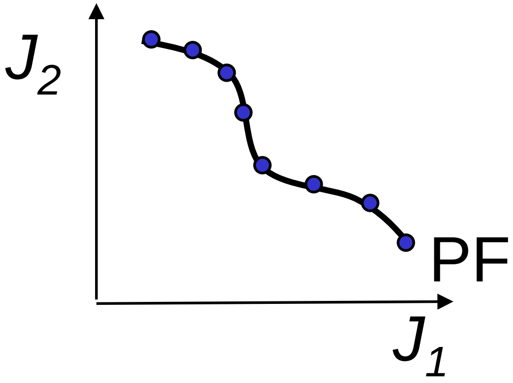
SOO: find **x***
MOO: find PF

- It must have the ability to capture all Pareto points
- Scaling mismatch between objective manageable
- An even distribution of the input parameters (weights) should result in an even distribution of solutions

⇒ A good method is
Normal-Boundary-Intersection (NBI)



bad

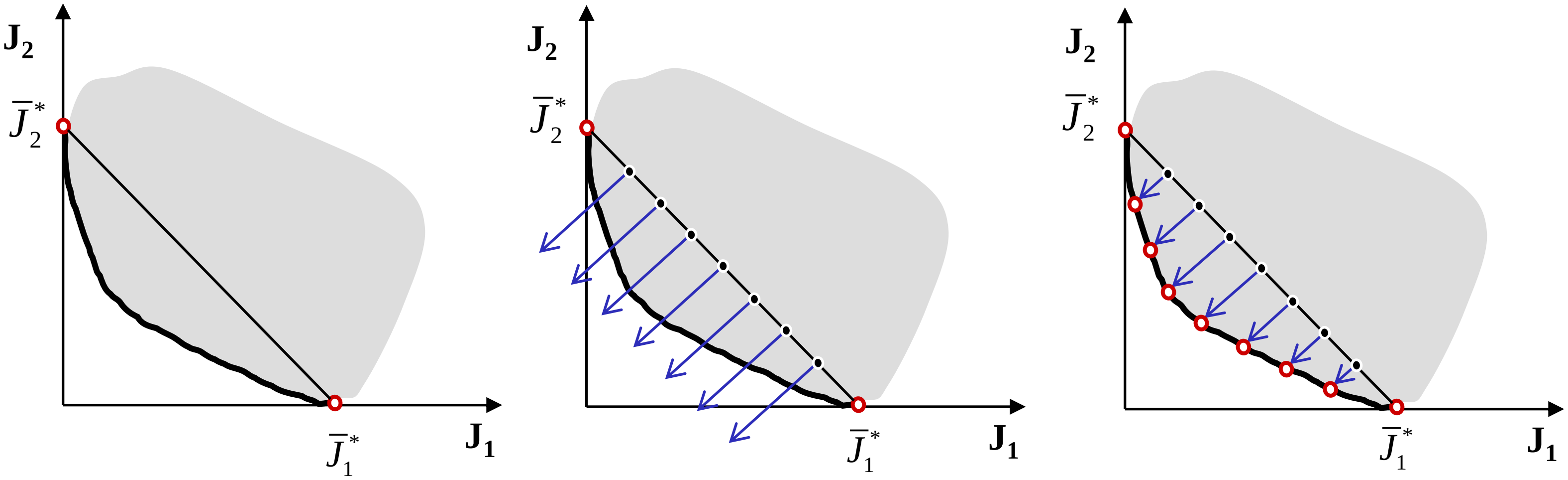


good

Normal Boundary Intersection (1)

Goal: Generate Pareto points that are well-distributed

- Carry out single objective optimization: $J_i^* = J_i \ \mathbf{x}^{\mathbf{i}*} \quad \forall i = 1, 2, \ldots, z$
- Find utopia point: $\mathbf{J}^{\mathbf{u}} = \left[J_1 \ \mathbf{x}^{\mathbf{1}*} \quad J_2 \ \mathbf{x}^{\mathbf{2}*} \quad \cdots \quad J_z \ \mathbf{x}^{\mathbf{z}*} \right]^T$
- U – Utopia Line between anchor points, NU – normal to Utopia line
- Move NU from $\bar{J}_1^*$ to $\bar{J}_2^*$ in even increments
- Carry out a series of optimizations
- Find Pareto point for each NU setting



Normal Boundary Intersection (2)

- Yields remarkably even distribution of Pareto points
- Applies for z>2, U-line becomes a Utopia-hyperplane.
- If boundary is sufficiently concave then the points found may not be Pareto Optimal. A Pareto filtering will be required.

Reference: Das I. and Dennis J, “Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems”, SIAM Journal on Optimization, Vol. 8, No.3,
1998, pp. 631-657

Adaptive Weighted Sum Method for Bi-objective Optimization

References:

Kim I.Y. and de Weck O.L., “Adaptive weighted-sum method for bi-objective optimization: Pareto front generation”, *Structural and Multidisciplinary Optimization*, 29 (2), 149-158, February 2005

Kim I.Y. and de Weck, O., “Adaptive weighted sum method for multi-objective optimization: a new method for Pareto front generation”, *Structural and Multidisciplinary Optimization*, 31 (2), 105-116, February 2006

AWS MOO - Motivation

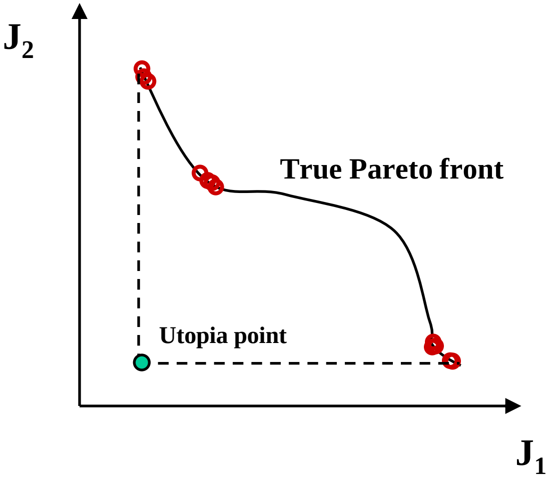
Drawbacks of Weighted Sum Method

(1) An even distribution of the weights among objective functions does not always result in an even distribution of solutions on the Pareto front.

In real applications, solutions quite often appear only in some parts of the Pareto front, while no solutions are obtained in other parts.

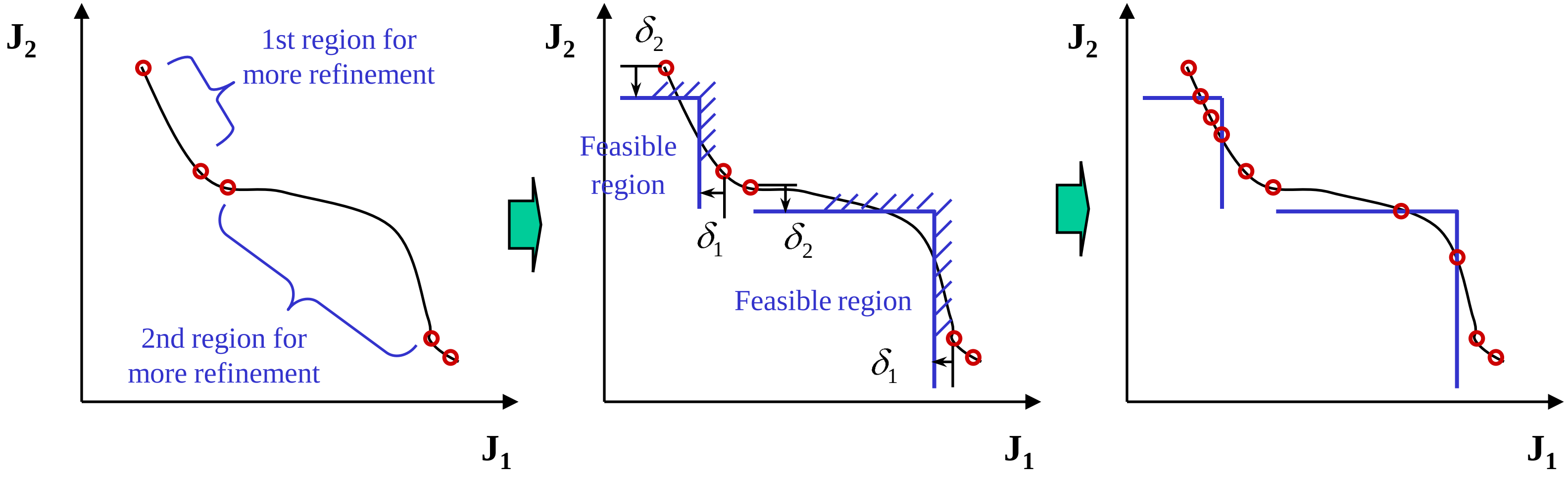
(2) The weighted sum approach cannot find solutions on non-convex parts of the Pareto front although they are non-dominated optimum solutions (Pareto optimal solutions). This is due to the fact that the weighted sum method is often implemented as a convex combination of objectives. Increasing the number of weights by reducing step size does not solve this problem.

Overall Procedure

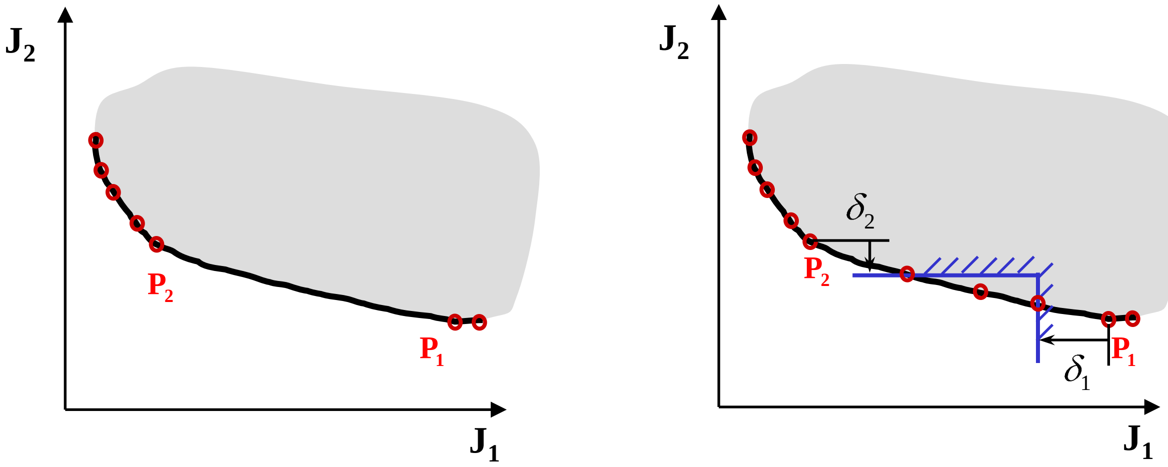


Weighted sum approach

Adaptive weighted sum approach



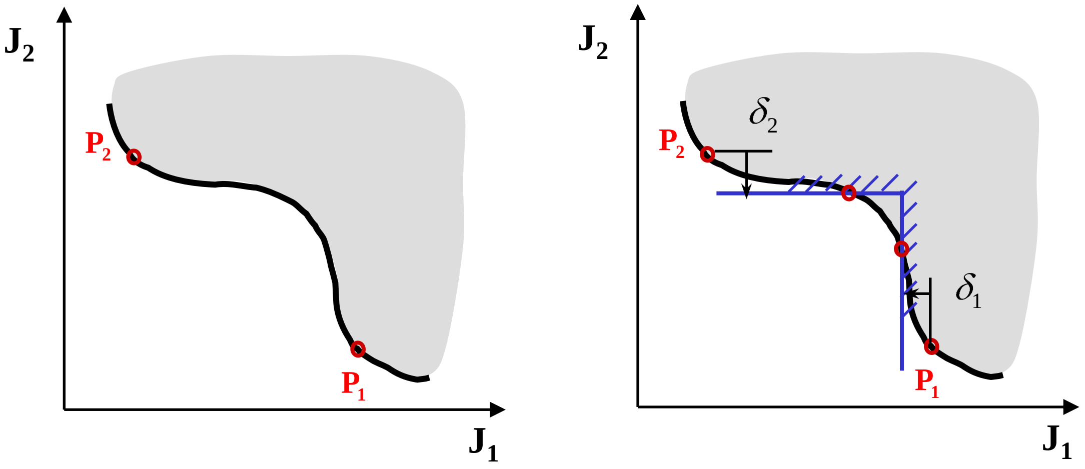
Convex Region with Non-Constant Curvature



Usual weighted sum method produces non-uniformly distributed solutions.

AWS focuses more on unexplored regions.

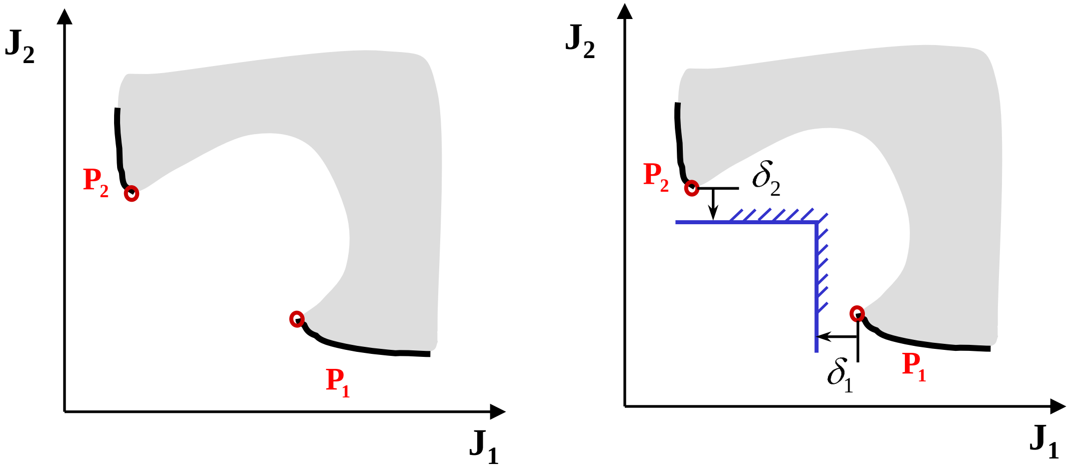
Non-convex Region: Non-Dominated Solutions



Usual weighted sum method cannot find Pareto optimal solutions in non-convex regions.

AWS determines Pareto optimal solutions in non-convex regions.

Non-Convex Region: Dominated Solutions



NBI erroneously determines dominated solutions as Pareto optimal solutions, so a Pareto filter is needed.

AWS neglects dominated solutions in non-convex regions.

Procedure (1)

[Step 1] Normalize the objective functions in the objective space

$$\bar{J}_i = \frac{J_i - J_i^U}{J_i^N - J_i^U}.$$

$\mathbf{J}^U = [J_1(\mathbf{x}^{1*}),\ J_2(\mathbf{x}^{2*})]$: Utopia point

$\mathbf{J}^N = [J_1^N,\ J_2^N]$ where $J_i^N = \max[J_i(\mathbf{x}^{1*})\ \ J_i(\mathbf{x}^{2*})]$: Nadir point

$\mathbf{x}^{i*}$: Optimal solution vector for the single objective optimization

[Step 2] Perform multiobjective optimization using the usual weighted sum approach

$\Delta\alpha = \dfrac{1}{n_{\text{initial}}}$: Uniform step size of the weighting factor

Procedures (2)

[Step 3] Delete nearly overlapping solutions on the Pareto front.

[Step 4] Determine the number of further refinements in each of the regions.

The longer the segment is, relative to the average length of all segments, the more it needs to be refined. The refinement is determined based on the relative length of the segment:

$$n_i = round\left(C \frac{l_i}{l_{avg}} \right) \qquad \text{for the } i\text{th segment}$$

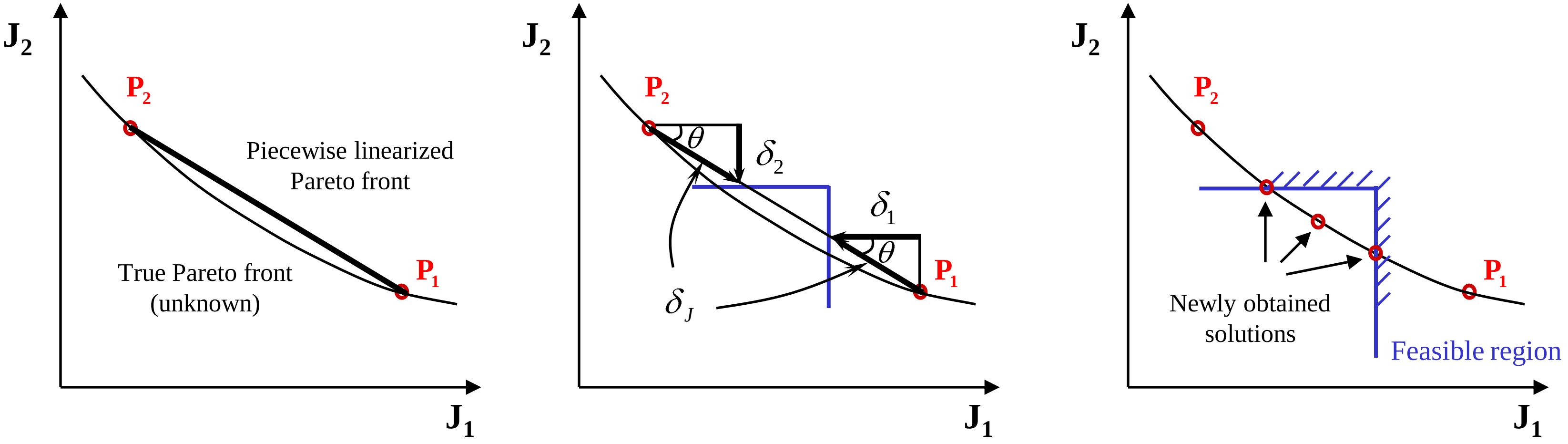
[Step 5] If n_i is less than or equal to one, no further refinement is conducted in the segment. For other segments whose number of further refinements is greater than one, go to the following step.

Procedures (3)

[Step 6] Determine the offset distances from the two end points of each segment

First, a piecewise linearized secant line is made by connecting the end points,

$$\theta = \tan^{-1}\left(-\frac{P_1^2 - P_2^2}{P_1^1 - P_2^1}\right) \qquad \delta_1 = \delta_J \cos\theta\,, \quad \delta_2 = \delta_J \sin\theta$$



Procedures (4)

[Step 7] Impose additional inequality constraints and conduct suboptimization with the weighted sum method in each of the feasible regions.

$$
\begin{aligned}
\min \quad & \alpha \bar{J}_1(x) + (1-\alpha)\bar{J}_2(x) \\
\text{s.t.} \quad & \bar{J}_1(x) \le P_1^x - \delta_1 \\
& \bar{J}_2(x) \le P_2^y - \delta_2 \\
& h(x) = 0 \\
& g(x) \le 0 \\
& \alpha \in [0,1]
\end{aligned}
$$

The uniform step size of the weighting factor for each feasible region is determined by the number of refinements (Step 4): $\Delta\alpha_i = \dfrac{1}{n_i}$

[Step 8] Convergence Check

Compute the length of the segments between all the neighboring solutions.
If all segment lengths are less than a prescribed maximum length, terminate the optimization procedure. If there are segments whose lengths are greater than the maximum length, go to Step 4.

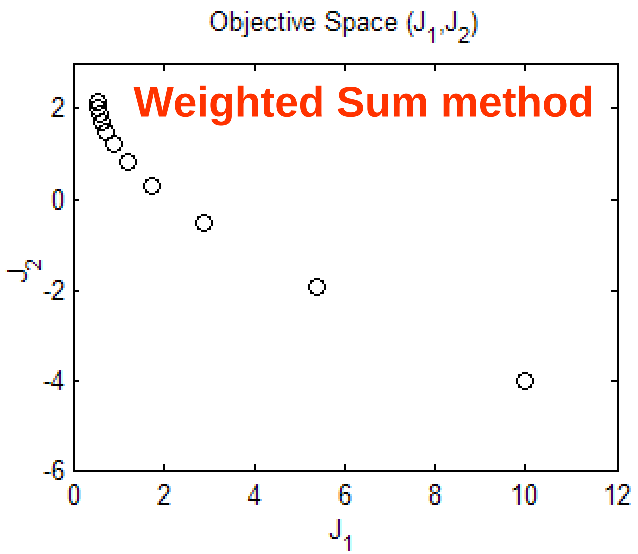
Example 1: Convex Pareto front

$$\text{minimize} \quad \begin{bmatrix} J_1 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \\ J_2 = 3x_1 + 2x_2 - \dfrac{x_3}{3} + 0.01(x_4 - x_5)^3 \end{bmatrix}$$

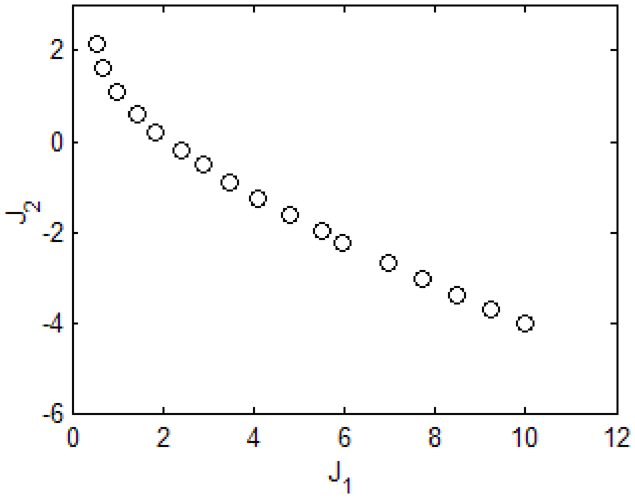
$$\begin{aligned} \text{subject to} \quad & x_1 + 2x_2 - x_3 - 0.5x_4 + x_5 = 2, \\ & 4x_1 - 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0, \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \le 10 \end{aligned}$$

Das, I., and Dennis, J. E., “Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems,” SIAM Journal on Optimization, Vol. 8, No. 3, 1998, pp. 631-657.

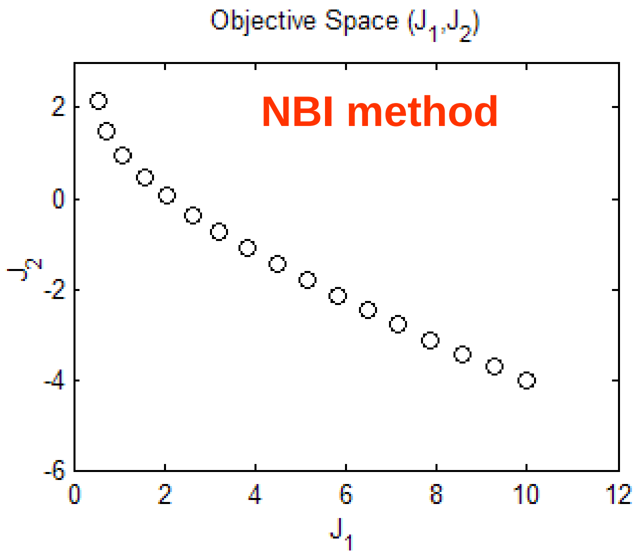
Example 1: (Convex Pareto front) - Results



AWS (Adaptive weighted Sum) method)



$\delta_J = 0.1$



	WS	NBI	AWS
No. of solutions	17	17	17
CPU time (sec)	1.71	2.43	3.83
Length variance ($\times 10^{-4}$)	266	0.23	2.3

Example 2: Non-convex Pareto front

$$\text{maximize} \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$$

$$J_1 = 3\ 1 - x_1\ {}^{2} e^{-x_1^2-(x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2-x_2^2}$$

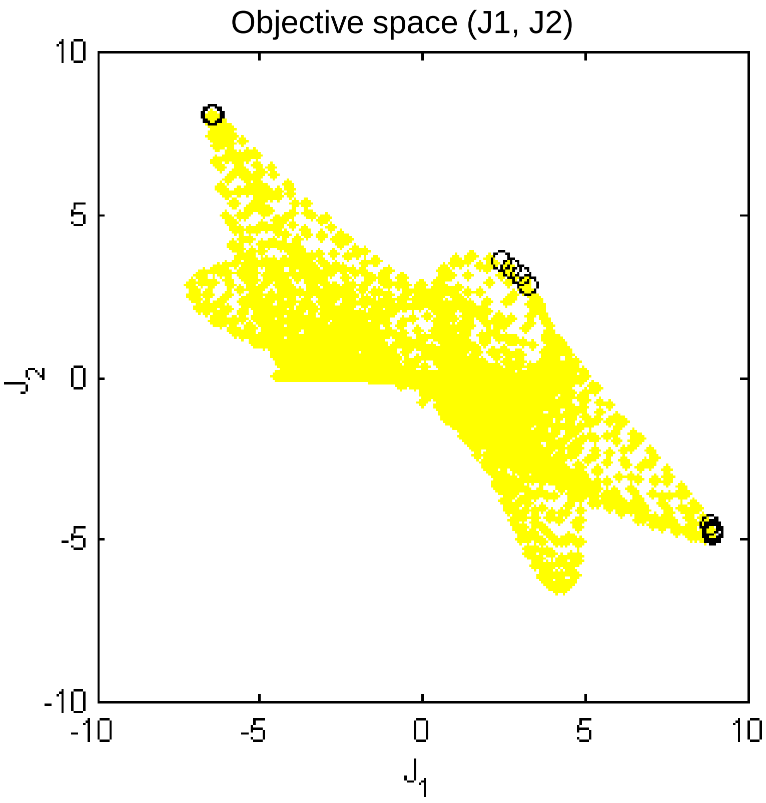
$$- 3e^{-(x_1+2)^2-x_2^2} + 0.5\ \ 2x_1 + x_2$$

$$J_2 = 3\ 1 + x_2\ {}^{2} e^{-x_2^2-(-x_1+1)^2} - 10\left(-\frac{x_2}{5} + x_2^3 + x_1^5\right) e^{-x_2^2-x_1^2}$$

$$- 3e^{-(2-x_2)^2-x_1^2}$$

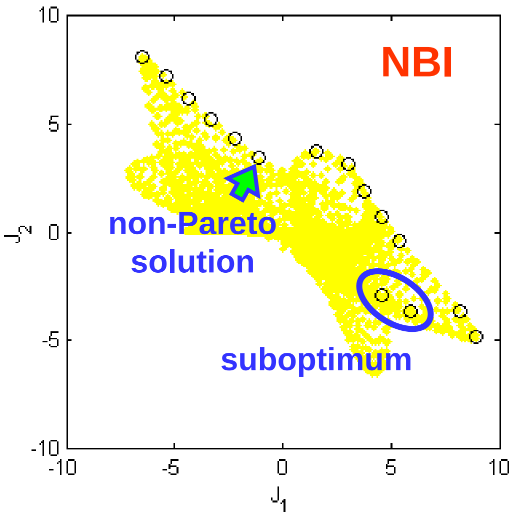
$$\text{subject to}\ \ -3 \le x_i \le 3\ ,\quad i = 1,2$$

Weighted Sum method

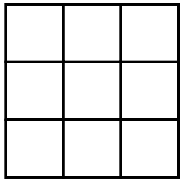


Example 2: Non-convex Pareto front - NBI

Case 1

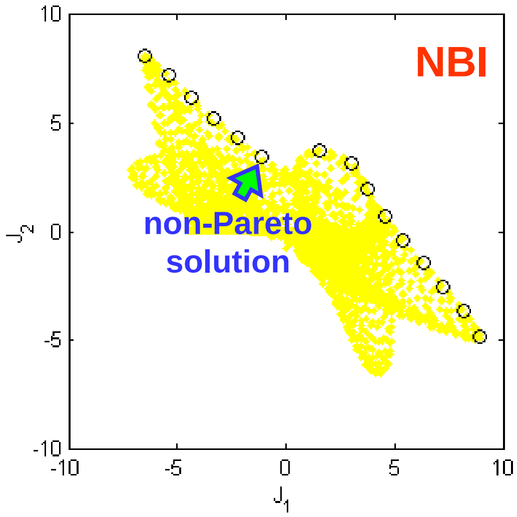


Initial starting point trials

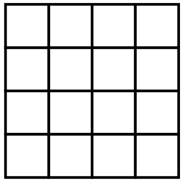


$\Delta x_1 = \Delta x_2 = 2.0$

Case 2

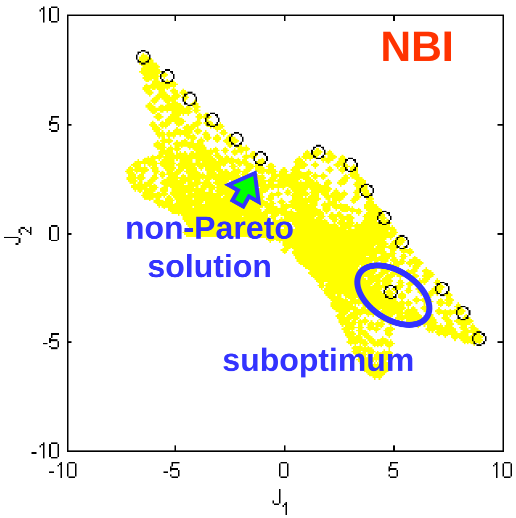


Initial starting point trials

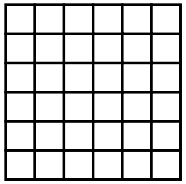


$\Delta x_1 = \Delta x_2 = 1.5$

Case 3

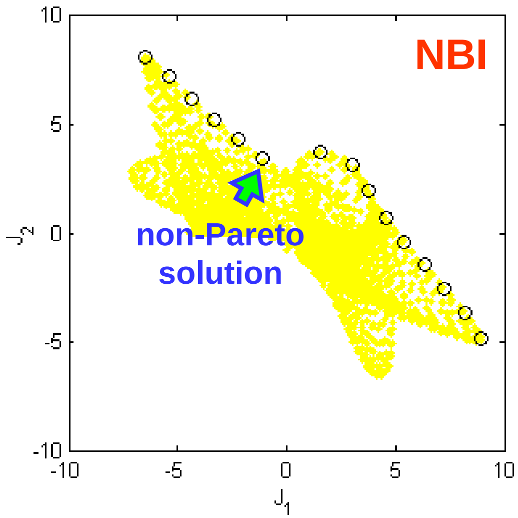


Initial starting point trials

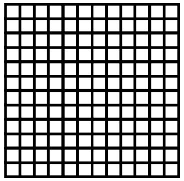


$\Delta x_1 = \Delta x_2 = 1.0$

Case 4



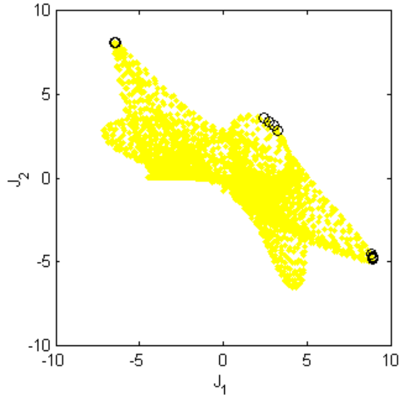
Initial starting point trials



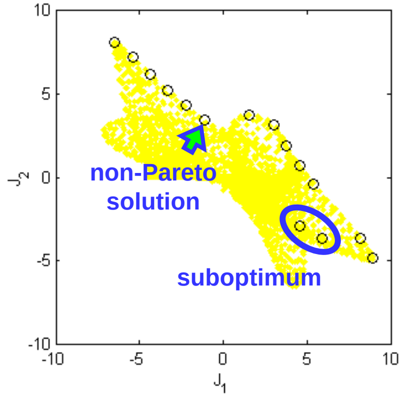
$\Delta x_1 = \Delta x_2 = 0.5$

Example 2: Non-convex Pareto front

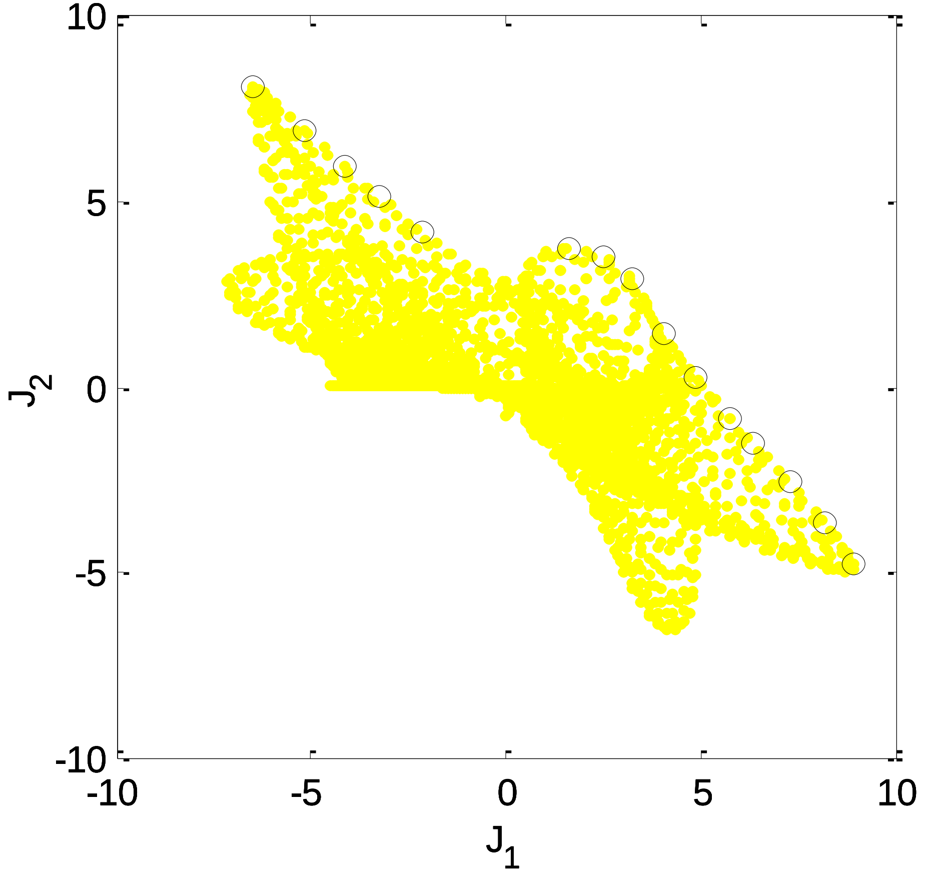
WS



NBI



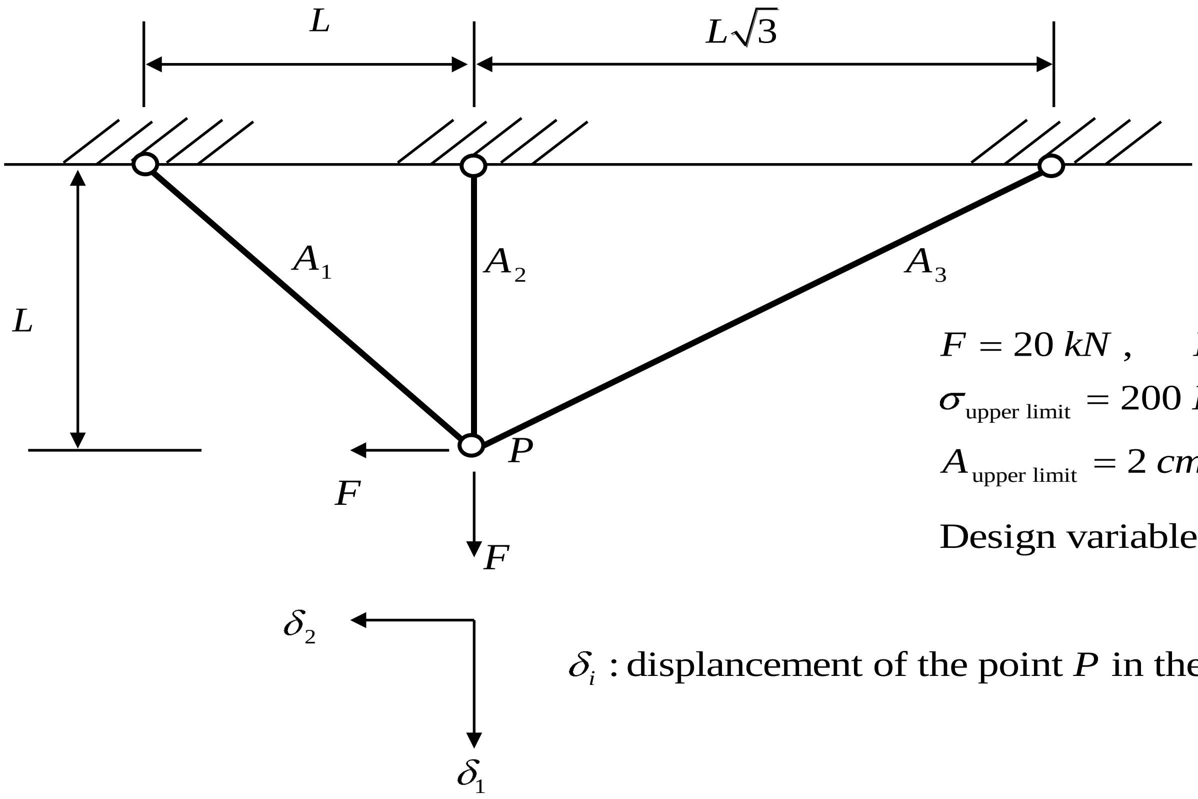
AWS $\delta_J = 0.1$



Example 2: Non-convex Pareto front

	WS	**NBI**				**AWS**			
Initial starting point case		Case 1	Case 2	Case 3	Case 4	Case 1	Case 2	Case 3	Case 4
No. of solutions	15	15	15	15	15	15	15	15	15
CPU time (sec)	0.4	17.8	24.5	52.9	165.6	28.1	44.0	87.6	289.2
Length variance (×10^{-4})	632	11	3.6	8.8	3.6	4.3	4.3	4.3	4.3
No. of suboptimum solutions	0	2	0	1	0	0	0	0	0
No. of non-Pareto solutions	0	1	1	1	1	0	0	0	0

Example 3: Static Truss Problem



$F = 20\ kN\ , \quad L = 100\ cm\ , \quad E = 200\ GPa$

$\sigma_{\text{upper limit}} = 200\ MPa\ , \quad \sigma_{\text{lower limit}} = -200\ MPa$

$A_{\text{upper limit}} = 2\ cm^2\ , \quad A_{\text{lower limit}} = 0.1\ cm^2$

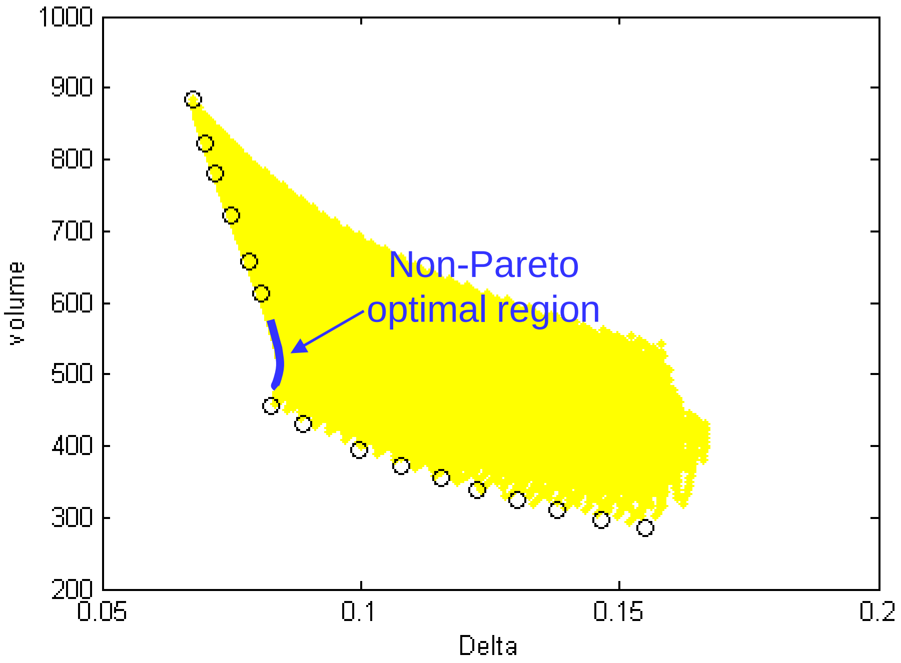
Design variables: $A_i\ , \quad i = 1, 2, 3$

δ_i : displancement of the point P in the ith direction

Koski, J., “Defectiveness of weighting method in multicriterion optimization of structures,” Communications in Applied Numerical Methods, Vol. 1, 1985, pp. 333-337

Example 3: Static Truss Problem

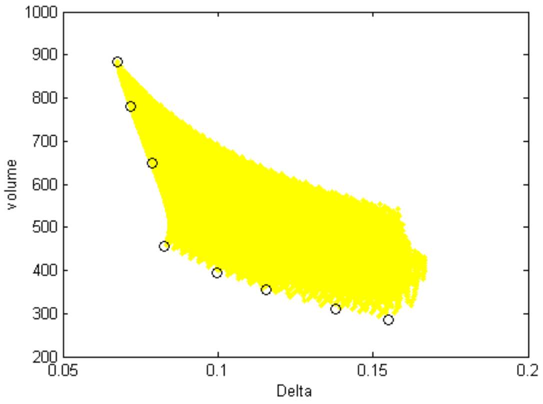
AWS



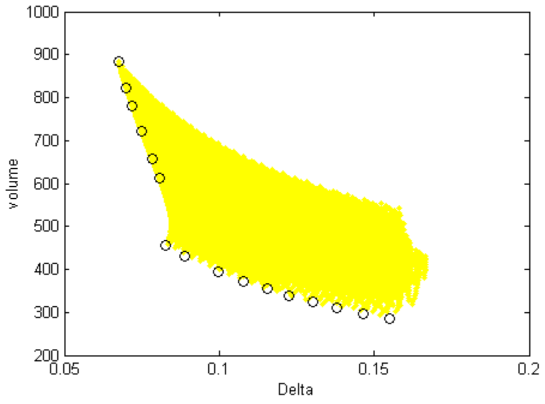
Non-Pareto optimal region

Example 3: Static Truss Problem

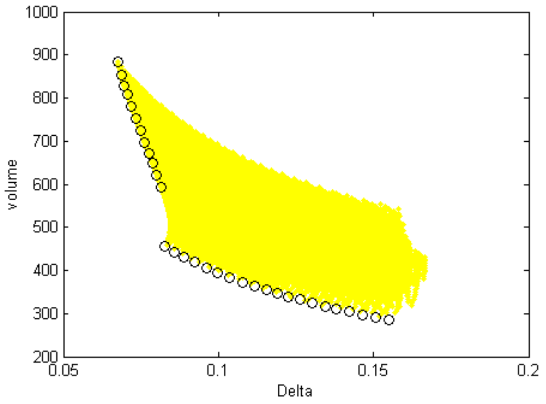
Solutions for different offset distances



$\delta_J = 0.2$



$\delta_J = 0.1$

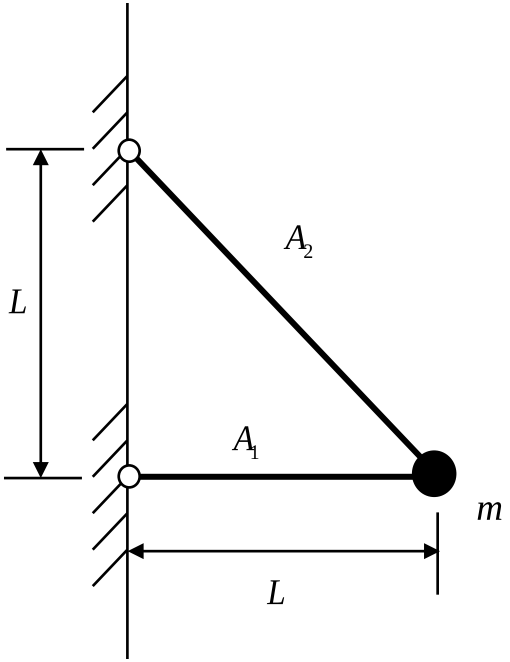


$\delta_J = 0.05$

Example 4: Dynamic Truss Problem

$$\text{mimimize} \begin{bmatrix} Volume(A_i) \\ \omega_1^2(A_i) \end{bmatrix}$$

$$\text{subject to } A_{\text{lower limit}} \leq A_i \leq A_{\text{lower limit}} \quad , \quad i = 1,2$$



$$\rho = 7850\ kg/m^3 \qquad L = 1\,m$$

$$E = 200\ GPa$$

$$A_{\text{lower limit}} = 1.0\ cm^2$$

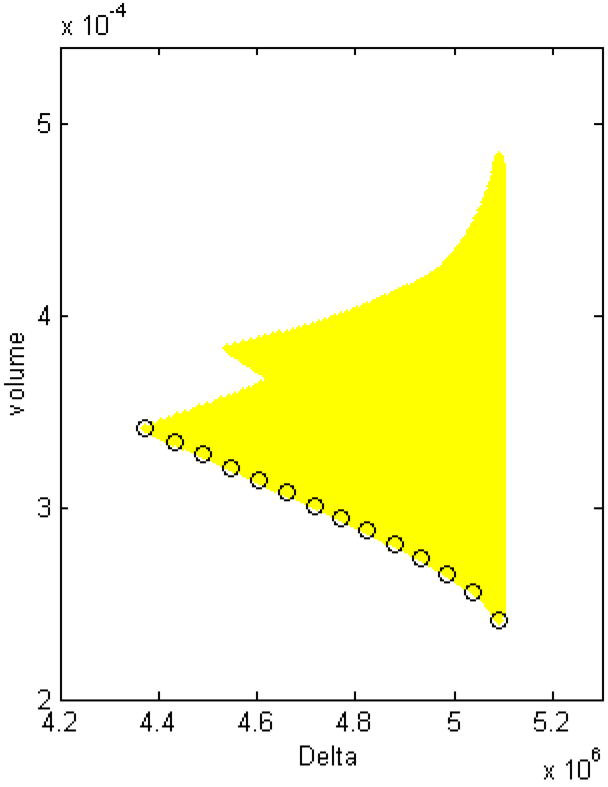
$$A_{\text{upper limit}} = 2.0\ cm^2$$

$$\Rightarrow m = \frac{\rho L}{2}(A_1 + \sqrt{2}A_2)$$

Koski, J., “Defectiveness of weighting method in multicriterion optimization of structures,” Communications in Applied Numerical Methods, Vol. 1, 1985, pp. 333-337

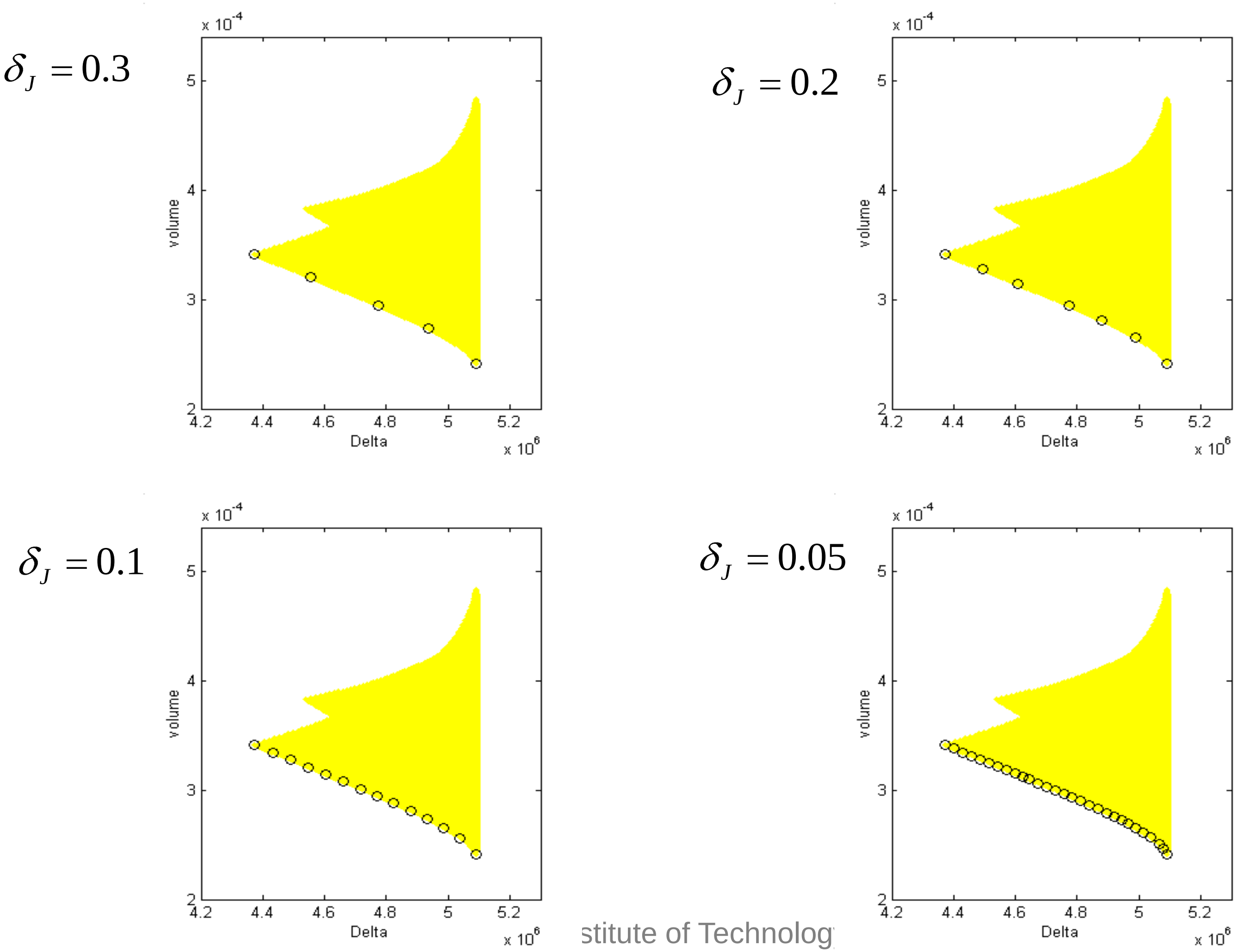
Example 4: Dynamic Truss Problem

AWS



Example 4: Dynamic Truss Problem

Solutions for different offset distances



Conclusions

The adaptive weighted sum method effectively approximates the Pareto front by gradually increasing the number of solutions on the front.

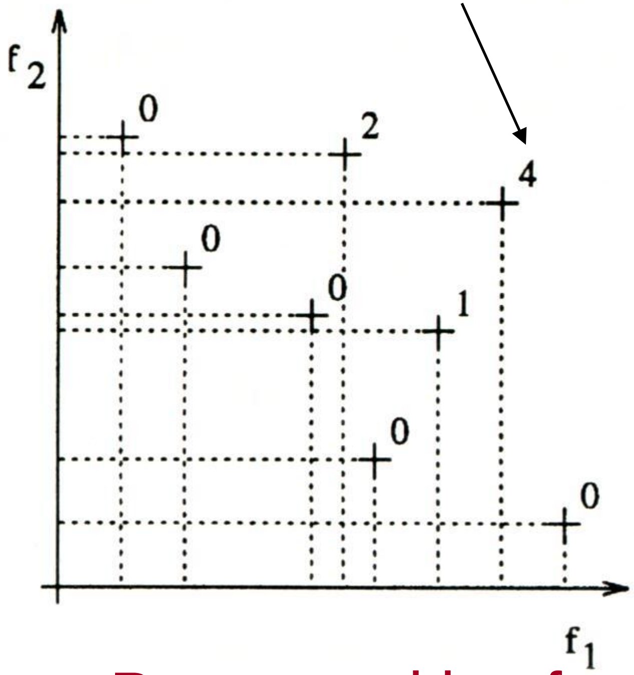
(1) AWS produces well-distributed solutions.
(2) AWS finds Pareto solutions on non-convex regions.
(3) AWS automatically neglects non-Pareto optimal solutions.
(4) AWS is potentially more robust in finding optimal solutions than other methods where equality constraints are applied.

AWS has been extended to multiple objectives (z>2), however, needed to introduce equality constraints for z>2.

Multiobjective Heuristics

Pareto Fitness - Ranking

This number comes from the D-matrix



Pareto ranking for
a minimization problem.

Recall: Multiobjective GA

- Pareto ranking scheme
- Allows ranking of population without assigning preferences or weights to individual objectives
- Successive ranking and removal scheme
- Deciding on fitness of dominated solutions is more difficult.

Example Multiobjective GA

Minimization

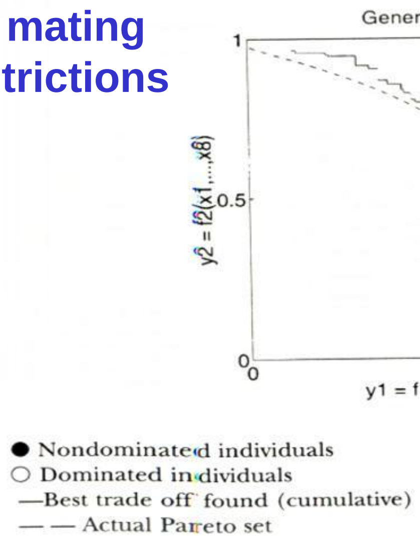
Objective 1

$$f_1 \quad x_1,\ldots,x_n \quad = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right]$$

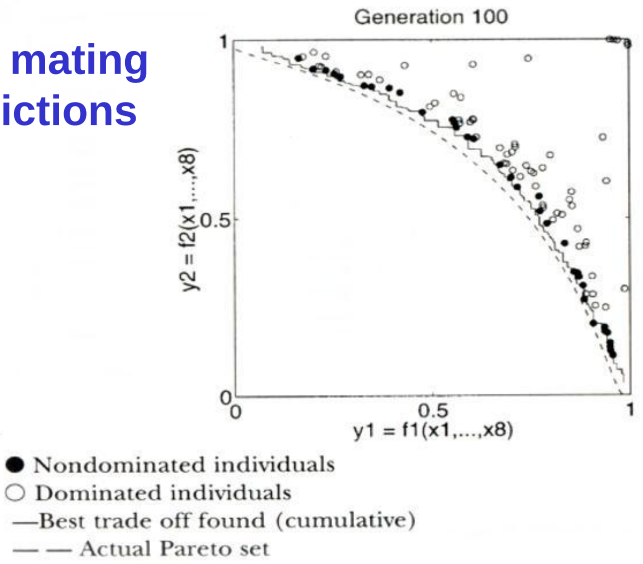
Objective 2

$$f_1 \quad x_1,\ldots,x_n \quad = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right]$$

No mating restrictions



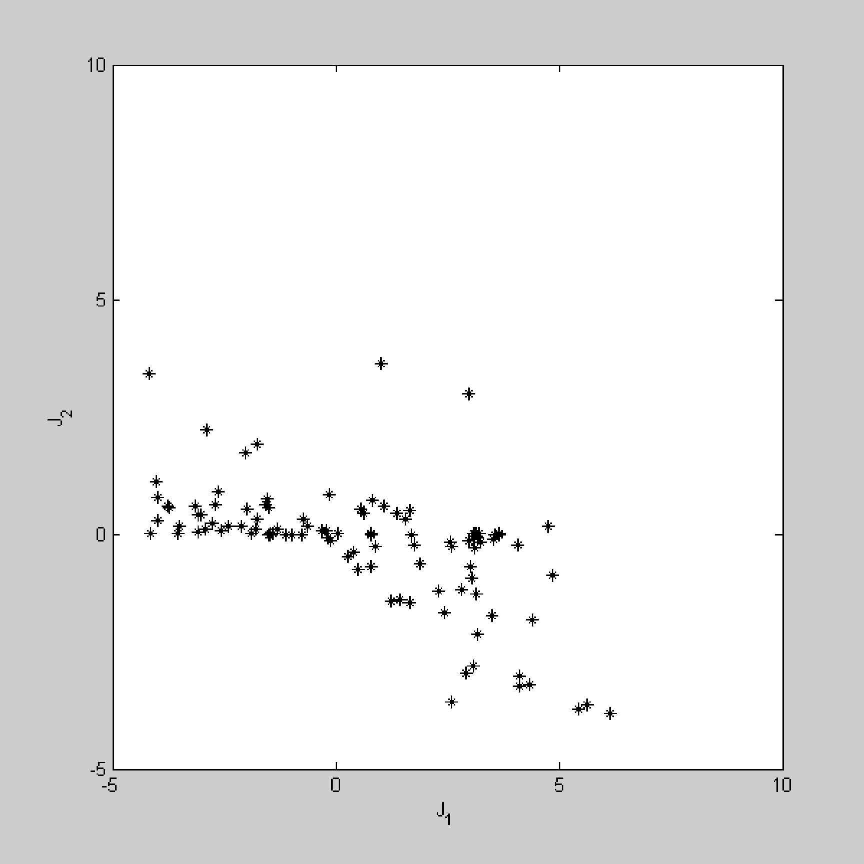
With mating restrictions



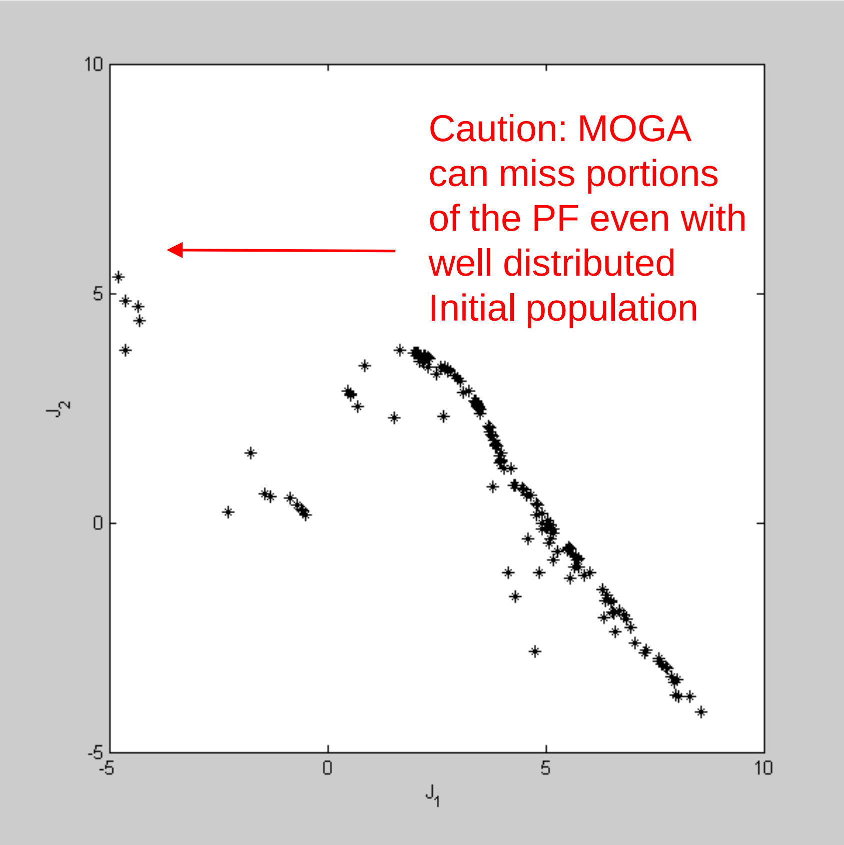
Double Peaks Example: MO-GA

Multiobjective Genetic Algorithm (MOGA)

Generation 1



Generation 10



Caution: MOGA can miss portions of the PF even with well distributed Initial population

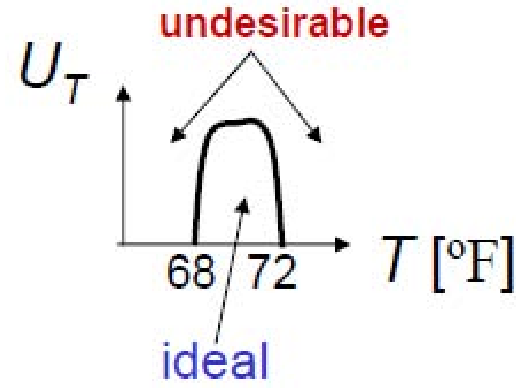
Utility Function Approach

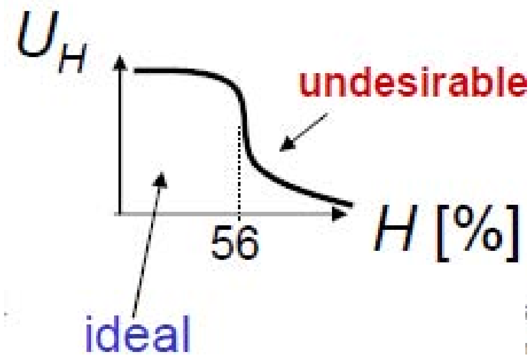
Want: - temperature in ideal range 68-72 ºF

- humidity above 56% is undesirable

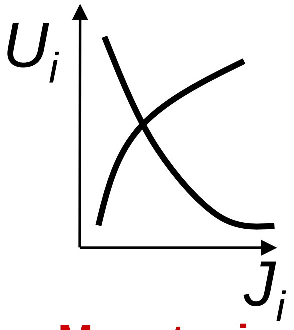
Assume: $T = \mathbf{c}^1\mathbf{x}$ temperature

$H = \mathbf{c}^2\mathbf{x}$ humidity

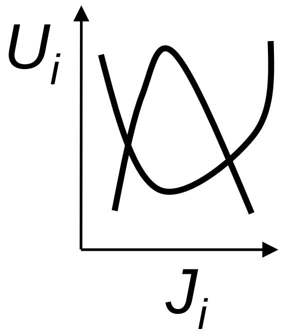




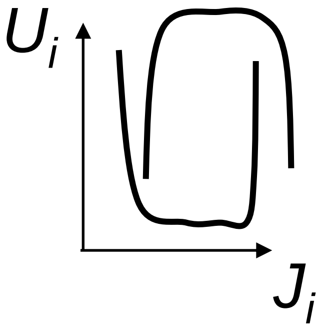
Utility Function Approach



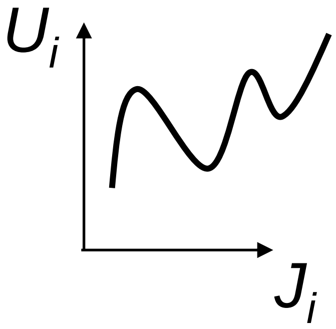
Monotonic increasing decreasing



Strictly Concave Convex



Concave Convex



Non-monotonic

H. Cook:

Smaller-is-better (SIB)
Larger-is-better (LIB)

Nominal-is -better (NIB)

Range -is-better (RIB)

-

A. Messac (Physical Programming):

Class 1S
Class 2S

Class 3S

Class 4S

Aggregated Utility

The total utility becomes the weighted sum of partial utilities:
… sometimes called multi-attribute utility analysis (MAUA)

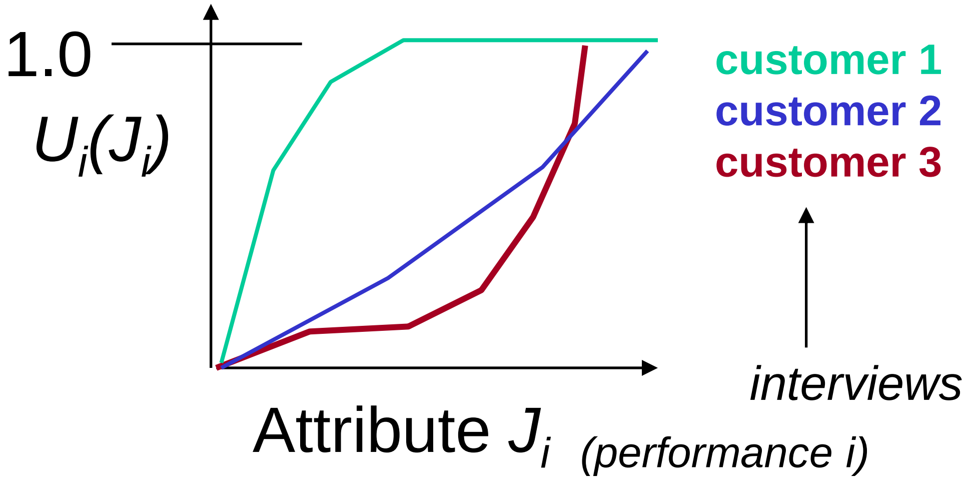
E.g. two utilities combined: $U\quad J_1, J_2 \quad = Kk_1k_2U(J_1)U(J_2)+k_1U(J_1)+k_2U(J_2)$

For 2 objectives: $K=(1-k_1-k_2)/k_1k_2$

Combine single utilities into overall utility function:

ki's determined during interviews
K is dependent scaling factor

Steps: MAUA
1. Identify Critical Objectives/Attrib.
2. Develop Interview Questionnaire
3. Administer Questionnaire
4. Develop Agg. Utility Function
5. Analyze Results



Caution: "Utility" is a surrogate for "value", but while "value" has units of [$], utility is unitless.

Notes about Utility Maximization

- Utility maximization is very common well accepted in some communities of practice
- Usually U is a non-linear combination of objectives *J*
- Physical meaning of aggregate objective is lost (no units)
- Need to obtain a mathematical representation for $U(J_i)$ for all *I* to include all components of utility
- Utility function can vary drastically depending on decision maker …e.g. in U.S. Govt change every 3-4 years

n-KKT

constrained case

If **x*** is non-inferior (=Pareto optimal) it satisfies the following KKT conditions:

a.) **x*** is feasible, i.e. $\mathbf{x}^* \in S$ and $S = \varnothing$

b.) all objective functions J_i and constraints g_j are differentiable[3]

c.) At **x*** the constraints are satisfied $g_j\left(\mathbf{x}^*\right) \leq 0 \quad \forall \;\; j = 1, 2, \ldots, m$

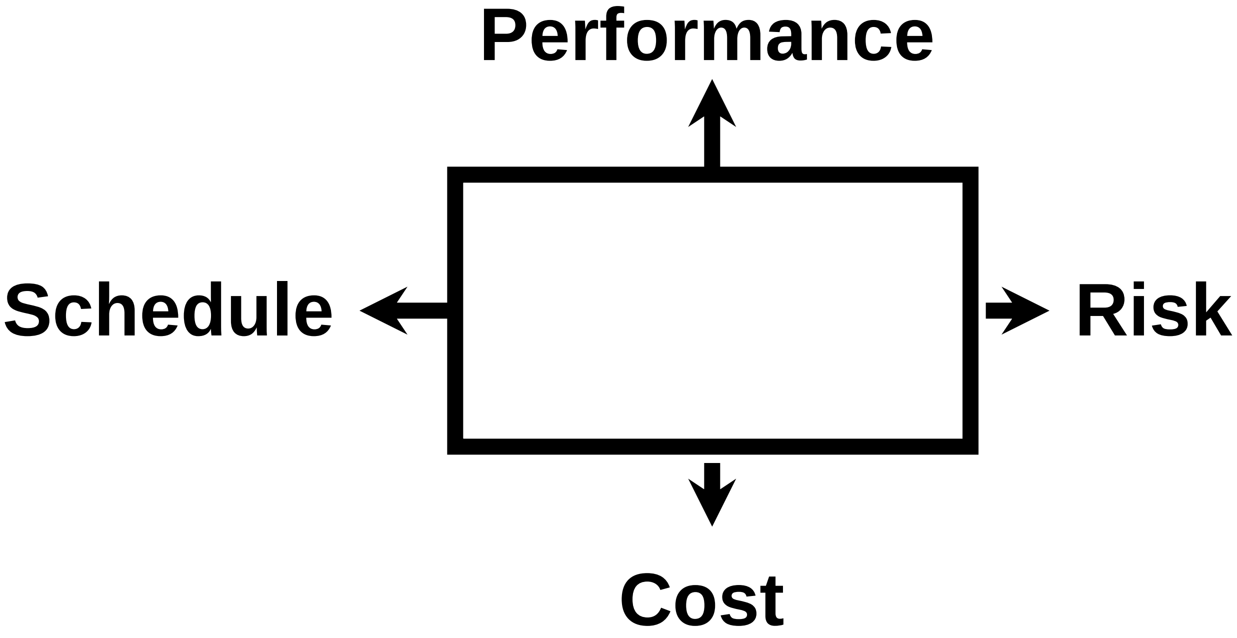
and $\lambda_j g_j\left(\mathbf{x}^*\right) = 0$ whereby $\lambda_j \geq 0 \; \forall \;\; j = 1, .., m$

d.) There exist $\mu_i \geq 0 \; \forall \;\; i = 1, .., n$ with strict inequality holding for at least one i such that

the condition $\sum_{i-1}^{n} \mu_i \nabla J_i(x^*) + \sum_{i-1}^{m} \lambda_j \nabla g_j(x^*) = 0$ is true.

These are the KKT conditions with n objectives

Four Basic Tensions (Trade-offs) in Product/System Development



Ref: Maier and Rechtin,
“The Art of Systems Architecting”, 2000

One of the main jobs of the system designer (together with the system architect) is to identify the principle tensions and resolve them

MO in iSIGHT

Screenshot of weighted sum optimization in iSIGHT software (Engineous) removed due to copyright restrictions.

Traditional iSIGHT is set up to do weighted-sum optimization

Note: Weights and Scale Factors in Parameters Table

Newer versions Have MOGA capability.

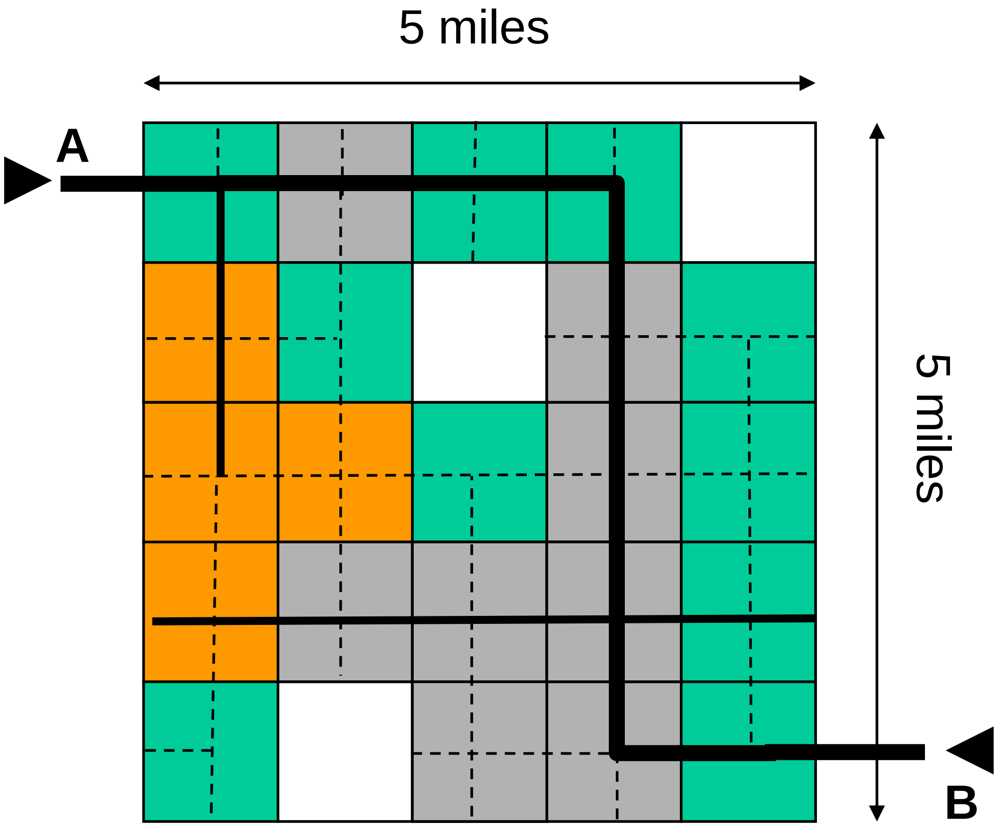
Multiobjective Optimization Game

Task: Find an optimal layout for a new city, which comprises 5x5 sqm and 50’000 inhabitants that will satisfy multiple disparate stakeholders.

Stakeholder groups:

a) Local Greenpeace Chapter
b) Chamber of Commerce
c) City Council (Government)
d) Resident’s Association
e) State Highway Commission

What layout should be chosen ?



Decision Space (I)

Each 1x1 sqm square can be on of four zones:

0	Vacant Zone
1	Commercial Zone (shops, restaurants, industry)
2	Recreational Zone (parks, lakes, forest)
3	Residential Zone (private homes, apartments)

All zones (except vacant) must be connected to each other via one of the following roads

3	━━━━━━	Highway (2 min/mile)
2	──────	Avenue (4 min/mile)
1	-----------	Street (6 min/mile)
0		Back Road (10 min/mile)

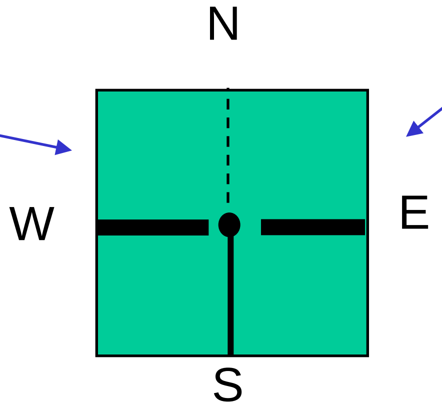
Decision Space (II)

The zoning is captured via the Z-matrix

$$Z=\begin{pmatrix} 2 & 1 & 2 & 2 & 0 \\ 3 & 2 & 0 & 1 & 2 \\ 3 & 3 & 2 & 1 & 2 \\ 3 & 1 & 1 & 1 & 2 \\ 2 & 0 & 1 & 1 & 2 \end{pmatrix}$$

The roads are captured via the “NEWS” R-matrix

$$R=\begin{array}{c|cccc} \text{Zone} & N & E & W & S \\ \hline 1 & 1 & 3 & 3 & 2 \\ 2 & 1 & 3 & 3 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 25 & 1 & 3 & 3 & 0 \end{array}$$



e.g. Zone 1

Both matrices Z and R uniquely define a city !

Objective Space

Objective Vector

T_c: daily average commuting time [min]
R_t: yearly city tax revenue [$/year]
C_i: initial infrastructure investment [$]
D_r: residential housing density [persons/sqm]
Q_a: average residential air pollution [ppm]

Weights Vector Λ

E.g. want to weigh short commute the highest

$$\Lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \end{bmatrix} = \begin{bmatrix} 500 & 125 & 125 & 100 & 150 \end{bmatrix}$$

Sum of weights must be 1000

Constraints

Fixed: City Population: 16,000
Hwy Connectors: 1W, 25E

Constraints:

(1) minimum 1 residential zone

(2) Hwy - must be connected somehow from upper left zone (1,1) to lower right zone (5,5)

Lecture Summary

- Two fundamental approaches to MOO
  - Scalarization of multiple objectives to a single combined objective (e.g. utility approach)
  - Pareto Approach with a-posteriori selection of solutions
- Methods for computing Pareto Front
  - Weighted Sum Approach
  - Design Space Exploration + Pareto Filter
  - Normal Boundary Intersection (NBI)
  - Adaptive Weighted Sum (AWS)
  - Multiobjective Heuristic Algorithms (e.g. MOGA)
- Resolving tradeoffs is essential in system design

References

[1] Pareto, V., *Manuale di Economia Politica*, Societa Editrice Libraria, Milano, Italy, Translated into English by Schwier, A. S. 1971: *Manual of Political Economy*, Macmillan, New York, 1906.

[2] Stadler, W., “A Survey of Multicriteria Optimization, or the Vector Maximum Problem,” *Journal of Optimization Theory and Applications*, Vol. 29, 1979, pp. 1-52.

[3] Stadler, W., “Applications of Multicriteria Optimization in Engineering and the Sciences (A Survey),” *Multiple Criteria Decision Making – Past Decade and Future Trends*, edited by M. Zeleny, JAI Press, Greenwich, Connecticut, 1984.

[4] Zadeh, L., “Optimality and Non-Scalar-Valued Performance Criteria,” *IEEE Transactions on Automatic Control*, AC-8, 59, 1963.

[5] Koski, J., “Multicriteria truss optimization,” *Multicriteria Optimization in Engineering and in the Sciences*, edited by Stadler, New York, Plenum Press, 1988.

[6] Marglin, S. *Public Investment Criteria*, MIT Press, Cambridge, Massachusetts, 1967.

[7] Steuer, R. E., *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley & Sons, New York, 1986.

[8] Lin, J., “Multiple objective problems: Pareto-optimal solutions by method of proper equality constraints,” *IEEE Transactions on Automatic Control*, Vol. 21, 1976, pp. 641-650.

References (cont.)

[9] Suppapitnarm, A. et al., "Design by multiobjective optimization using simulated annealing," *International conference on engineering design ICED 99*, Munich, Germany, 1999.

[10] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[11] Fonseca, C., and Fleming, P., "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, Vol. 3, 1995, pp. 1-18.

[12] Tamaki, H., Kita, H., and Kobayashi, S., "Multiobjective optimization by genetic algorithms: a review," *1996 IEEE International Conference on Evolutionary Computation*, ICEC '96, Nagoya, Japan, 1996.

[13] Messac, A., and Mattson, C. A., "Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming," *Optimization and Engineering*, Kluwer Publishers, Vol. 3, 2002, pp. 431-450.

[14] Mattson, C. A., and Messac, A., "Concept Selection Using s-Pareto Frontiers," *AIAA Journal*. Vol. 41, 2003, pp. 1190-1204.

[15] Das, I., and Dennis, J. E., "Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems," *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 631-657.

[16] Das, I., and Dennis, J. E., "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optimization*, Vol. 14, 1997, pp. 63-69.

[17] Koski, J., "Defectiveness of weighting method in multicriterion optimization of structures," *Communications in Applied Numerical Methods*, Vol. 1, 1985, pp. 333-337.

ESD.77 / 16.888 Multidisciplinary System Design Optimization
Spring 2010