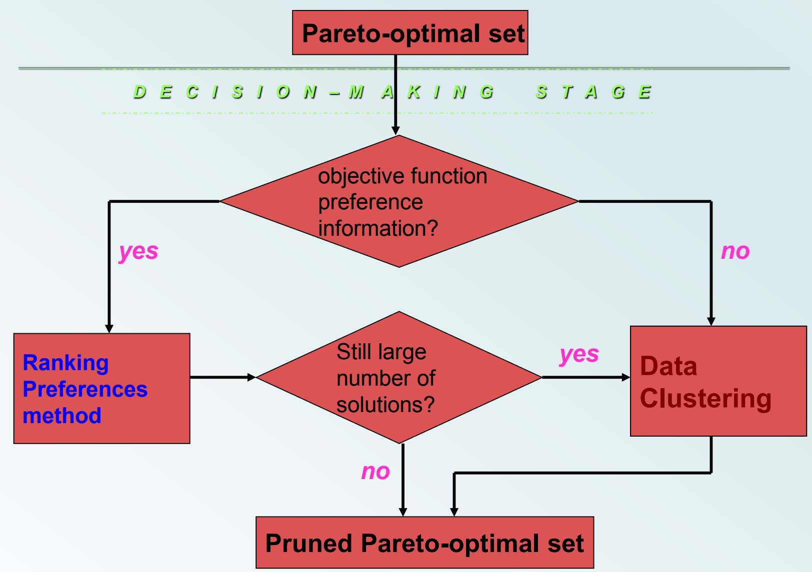
Multi-Objective Optimization Design:
Pruning Methods

Prof. David W. Coit & Assoc. Prof. Naruemon Wattanapongsakorn

King Mongut's University Technology of Thonburi (KMUTT)

Term 1/2016

Methods to prune the Pareto-optimal set

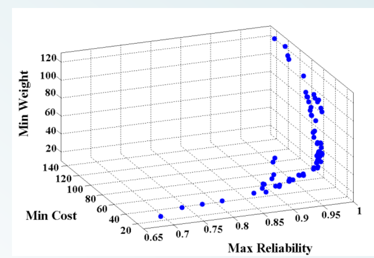


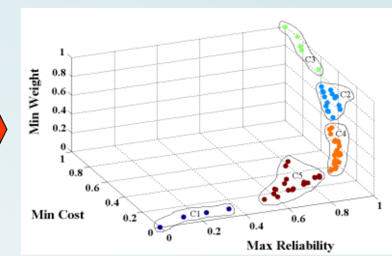
Data Clustering

- Clustering is a **data mining** technique, which in multi-dimensional situations can be very insightful
- Clustering methods are applied in many domains such as in artificial intelligence, pattern recognition, **decision-making**, etc.
- Cluster analysis is used to find groups in data, and such groups are called **clusters**.
- The clusters are formed in such a way that **objects in the same group are similar to each other**, **whereas objects in different groups are as dissimilar as possible.**

Multiobjective RAP Clustering

Pareto set of MO-RAP





Pruning by using data clustering

1. Obtain the entire Pareto-optimal set or sub-set
2. Apply the k-means algorithm to form clusters
3. Run the k-means algorithm for different values of k
4. Choose k with the highest average *silhouette width*
   - Rousseeuw, P.J. (1987). Journal of Computational and Applied Mathematics, 20, 53-65.
   - Kaufman, L. and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.
5. Per cluster, select a representative solution
6. Choose a solution for system implementation

k-means Algorithm

- **k-means algorithm is probably the most widely applied nonhierarchical clustering technique (Kaufman *et al.*, 1990).**
- **The k-means algorithm is well known for its efficiency in clustering data sets.**
- **The grouping is done by calculating the centroid for each group, and assigning each observation to the group with the closest centroid.**
- **The objective function that the k-means algorithm optimizes is:**

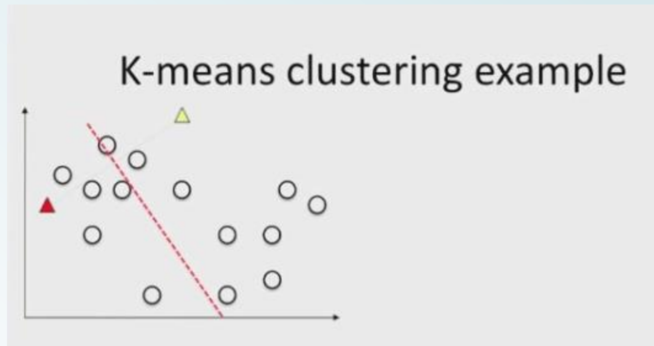
$$KM(X,C) = \sum_{i=1}^{n} \min_{j \in \{1,..,k\}} \| \mathbf{x}_i - \mathbf{c}_j \|^2$$

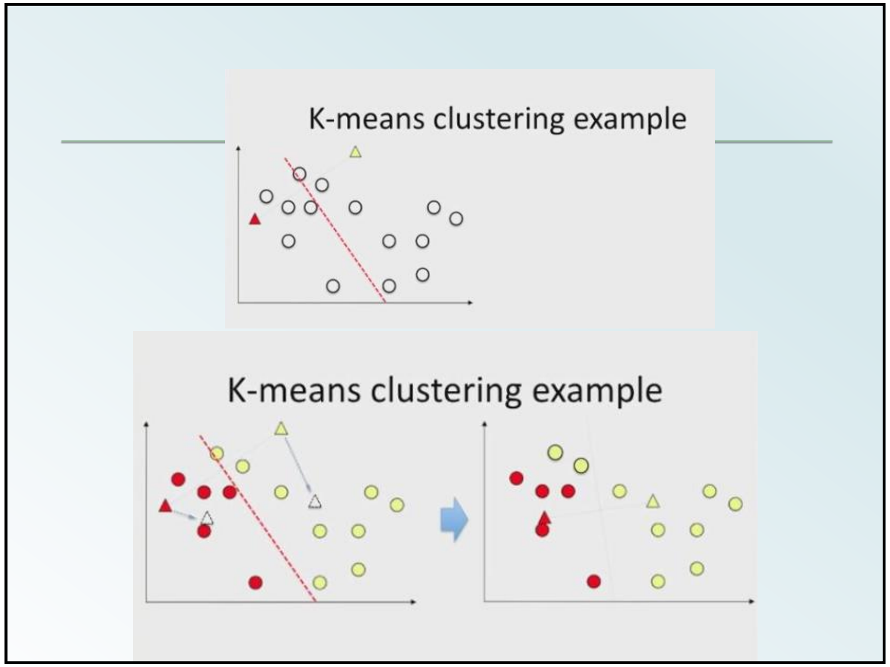
$\mathbf{x}_i$ = i^{th} data vector X = set of data vectors

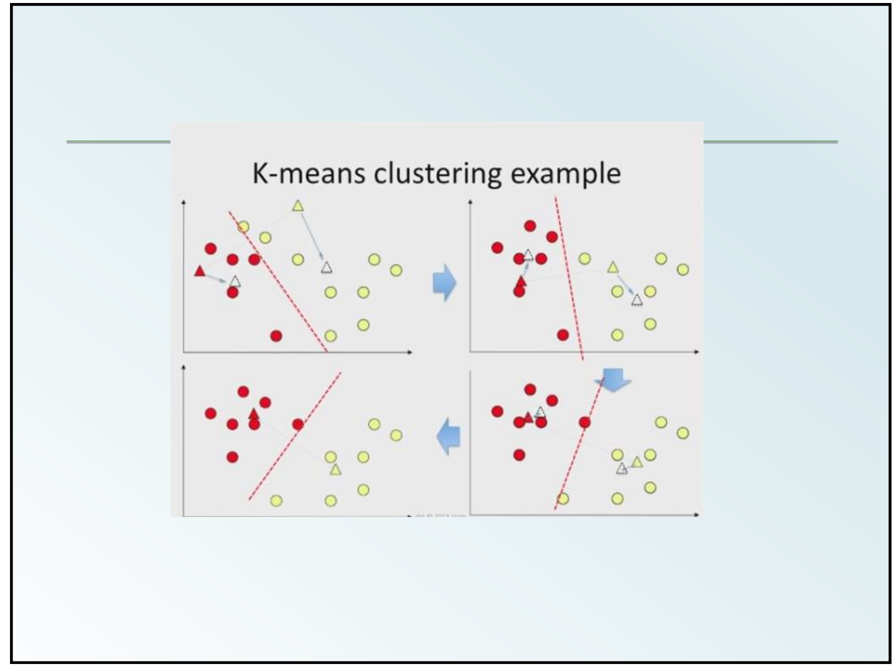
$\mathbf{c}_j$ = j^{th} cluster centroid C = set of centroids

The *K-Means* Clustering Method

- **Given *k*, the *k-means* algorithm is implemented in four steps:**
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change




K-means clustering example
K-means clustering example


K-means clustering example

Example: Multi-objective RAP

The objectives are to **determine the optimal design configuration that** will maximize system reliability, minimize the total cost, and minimize the system weight, for a series-parallel system.

system reliability	system cost	system weight

$$\max\left[\prod_{i=1}^{s} R_i(\mathbf{x}_i)\right],\ \min\left[\sum_{i=1}^{s}\sum_{j=1}^{m_i} c_{ij}x_{ij}\right],\ \min\left[\sum_{i=1}^{s}\sum_{j=1}^{m_i} w_{ij}x_{ij}\right]$$

Subject to:

$$1 \le \sum_{j=1}^{m_i} x_{ij} \le n_{\max,i} \quad \text{for} \quad \forall i = 1,2,..,s$$

$$x_{ij} \in \{0,1,2,..\}$$

Where:

s = number of subsystems

$\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,m_i})$

X_{ij} = quantity of the j^{th} available component used in subsystem i

m_i = total number of available components for subsystem i

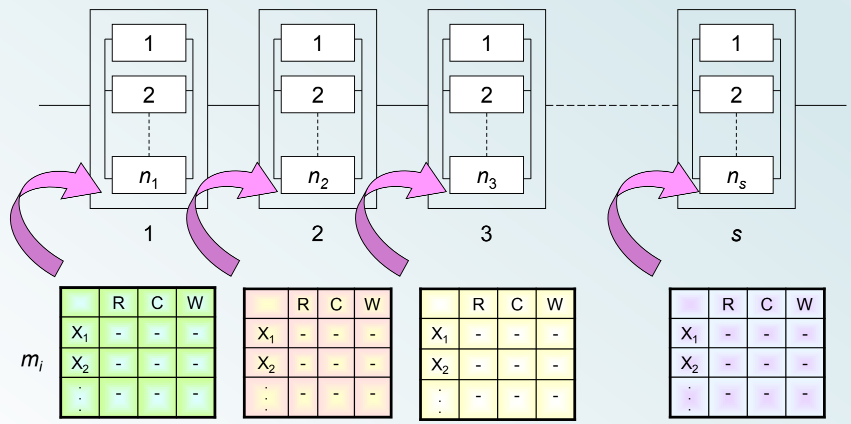
$n_{max,i}$ = user defined maximum number of components in parallel used in subsystem i

$R_i(\mathbf{x}_i)$ = reliability of subsystem i = $\left[1 - \prod_{j=1}^{m_i}(1 - r_{ij})^{x_{ij}}\right]$

c_{ij}, w_{ij}, r_{ij} = cost, weight and reliability for the j^{th} available component for subsystem i

Redundancy Allocation Problem (RAP)

- **System design optimization problem :** a total of ***s*** subsystems connected in series. For each subsystem, there are n_i functionally equivalent components arranged in parallel.



Example: Multiobjective RAP (cont'd…)

- MO-RAP was formulated and solved using the fast elitist non-dominated sorting genetic algorithm (NSGA-II).

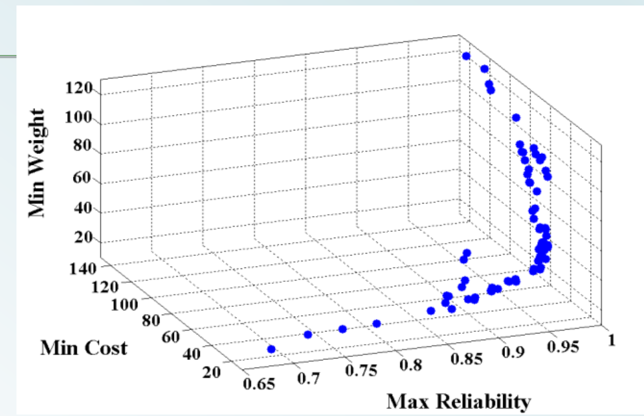
Example Problem:
- 3 subsystems
- 5, 4 and 5 component choices per subsystem
- **75 solutions were found in the Pareto set**

Component choices for each subsystem

Design Alternative *j*	Subsystem *i*								
	1			2			3		
	R	***C***	***W***	***R***	***C***	***W***	***R***	***C***	***W***
1	0.94	9	9	0.97	12	5	0.96	10	6
2	0.91	6	6	0.86	3	7	0.89	6	8
3	0.89	6	4	0.70	2	3	0.72	4	2
4	0.75	3	7	0.66	2	4	0.71	3	4
5	0.72	2	8				0.67	2	4

Solving the redundancy allocation problem has been shown to be NP-hard by Chern (1992).

Example: Multi-objective RAP (cont'd…)



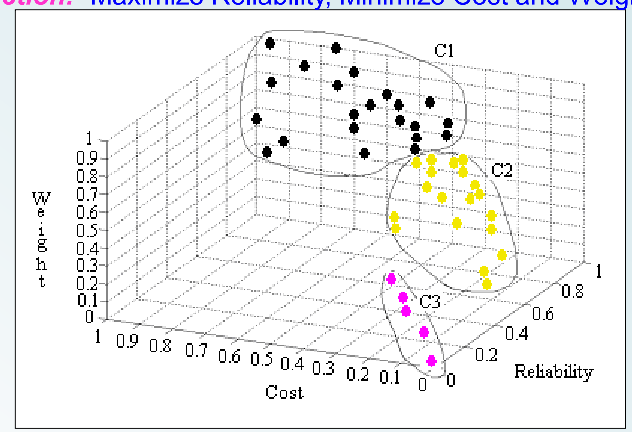
Pareto set of MO-RAP

RAP Example - Pruned results using data clustering

- 46 solutions in the Pareto-optimal set
- *k*-means algorithm to partition the data set
- Silhouette plots to validate the partition
- 3 as an "appropriate" number of clusters
  - **Cluster 1 --- 22 solutions**
  - **Cluster 2 --- 19 solutions**
  - **Cluster 3 --- 5 solutions**

Clustered Pareto-optimal set

Obj. function: Maximize Reliability, Minimize Cost and Weight



3 dimensional plot from normalized data

How to choose a solution from each cluster

- Identify the solution that is closest to its respective cluster centroid

	# of solutions	Representative solutions	Reliability	Cost	Weight
Cluster 1	22	#39	0.9978541	22	34
Cluster 2	19	#91	0.984265	15	25
Cluster 3	5	#87	0.819216	11	24

- The DM has now a small set of solutions

Pareto Set Pruning Conclusions

- Pruning solutions creates a pragmatically sized set that reflects DM's preferences
  - Pruning causes ~ 90% Reduction in the Pareto optimal set
- Pruning the solutions facilitates the decision making process
  - It focuses on solutions that are attractive to the decision maker
  - Filters out unwanted solutions
  - Reduces the size of the possible nondominated solutions that can be used for implementation