ME/IE542 Advanced Computational & Statistical Methods for Product and Process Design

Dr. Wei Chen
Department of Mechanical & Industrial Engineering
UIC

Spring 2003

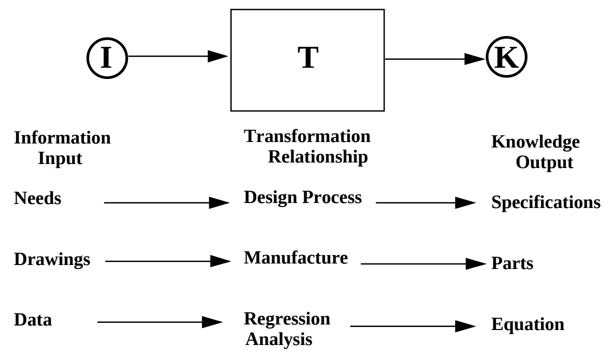
Course Goals

- Learn research opportunities in engineering design, in a broad context of product design, manufacturing process development, and designing for life cycle issues.
- Learn the mathematical fundamentals of engineering design
- Learn advanced and emerging methods of modeling and solving engineering design problems using computer-based techniques
- Apply the course material to graduate research

What is Design (Designing)?

Designing is an intellectual, cognitive activity that involves both the human and the computer / technology ....

Designing is the process of converting information that characterizes the needs and requirements for a product into knowledge about a product (Mistree). K= [T] I



Characteristics of Design

- Uncertainty and estimation cannot be avoided (we cannot afford to build the system and test it).
- We always have an *option* space from which the decision will be selected.
- Modeling is needed to get better estimation of the *outcome*.
- Designs are good or bad, not right or wrong.
- Need *value* to judge which option is better.

Major Design Phases

- **Clarification of the task –** collection of information about the requirements to be embodied in the solution and also about the constraints.
- **Conceptual design –** establishment of function structures, the search for suitable solution principles and their combination into concept variants.
- **Embodiment design –** engineering a solution principle by determining the general arrangements and preliminary shapes and materials of all components Embodiment design is sometimes called preliminary design.
- **Detail design –** all the details of the final design are specified and manufacturing drawings and documentation are produced.

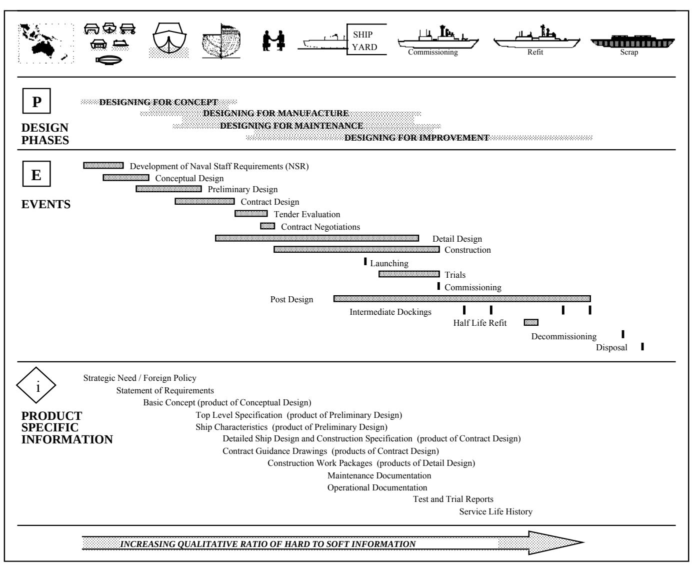
Design versus Analysis

- **Analysis** – A process of partitioning or decomposing any system into its subsystems and component parts to determine their separate and collective nature, proportion, functions, relationships. etc.
- **Synthesis** – A process of integrating a collection of subsystems so as to create a system with emergent properties.
- **Evaluation** – A process of assessing the degree to which a solution satisfied that goals that were originally stated.
- **Design involves all three of the above.**

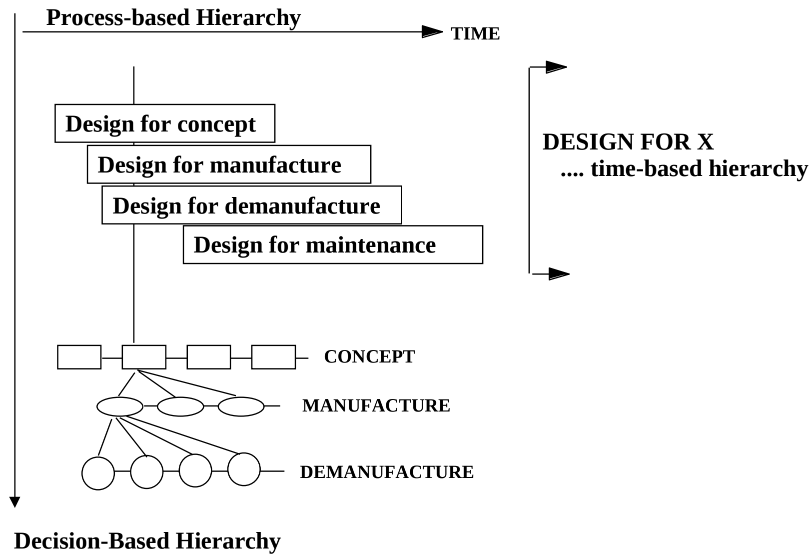
Design Types

- **Original Design -** an original solution principle is determined for a desired system and used to create the design of a product.
- **Adaptive Design -** an existing design is adapted to different conditions or tasks; thus, the solution principle remains the same but the product will be sufficiently different so that it can meet the changed tasks that have been specified.
- **Variant Design -** the size and/or arrangement of parts or subsystems of the chosen system are varied. The desired tasks and solution principle are not changed.

Design Processes

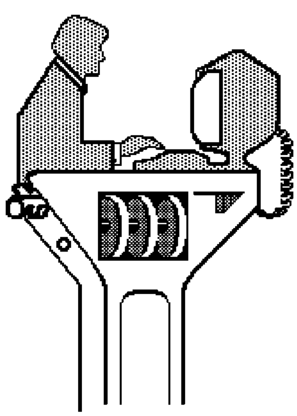


Design and Decision Hierarchies



Human-Computer Relationships

The Human-Computer Cyborg



OBSERVATIONS

- **The Computer extends human *ability*.**
- **The Human augments computer *capability*.**

COMPUTER-CENTERED DESIGN

- **Computer-Aided Design**
- **Computer-Aided Engineering**

HUMAN-CENTERED DESIGN

- **Decision-Based Design ...**

Design Theory and Approach

- **Is design** computer-based? CAD-based? information-based? Decision-based? Negotiation-based?...
- **Different schools of thinking**
  - German (Europe)
    - Pahl and Beitz - *systematic approach*
  - Japan
    - Yoshikawa - *Knowledge processing methods*
    - Taguchi - *Robust design*
  - USA
    - Suh - *Axiomatic design*
- **Emerging methods**
  - Decision-based design (Hazelrigg, Mistree).
- **Descriptive vs. Prescriptive**

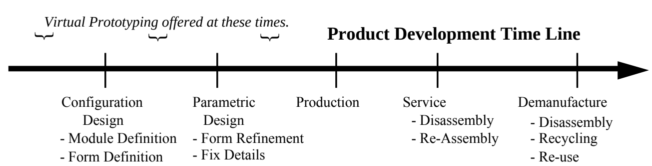
Hot Topics in Design Theory Research

- **Function-based concept generation methods -** *representation, grammar, etc.*
- **Product Architecture - Portfolio - Modular Design - Open Engineering Systems**
- **Decision analysis**
- **Design for X (DFX) - Environmentally Benign Design**
- **…**

Paradigm Shifts

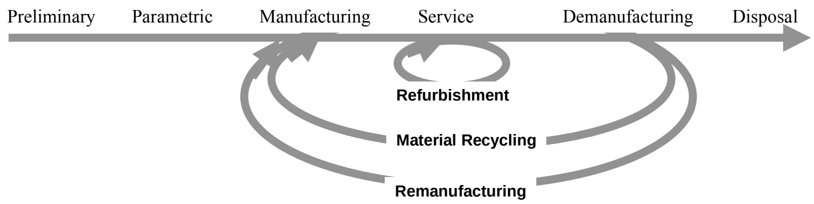
- From optimizing solutions to satisficing and robust solutions at early design stages.
- From design-for-manufacture to design-for-demanufacture and design-for-reuse.
- From CAD/CAM to Virtual Design Studios.
- From design from scratch to design using available assets.
- From mass production to mass customization.
- From single designer to distributed collaborative multidisciplinary design.

Virtual Design Studio and Collaborative Design



Research Thrusts	Product Design	Service Process	Demanufacture Process
Human-Computer Interaction	• Form definition • Fastener selection • Componentarrangement	• Dis/Re-assembly assessment • Dis/Re-assembly sequence planning and motion	
Decision Support	• Selection and Compromise Decision Support Problems	• Disassembly & Re-assembly simulation	• Disassembly process simulation • Demanufacture process simulation • Facilities modeling
Product Representation and Model Generation	• Virtual prototype generation on demand. • Level of detail match between design stage and prototype		
Collaborative Prototyping Environment	• Synchronous and asynchronous collaboration • Human-Computer collaborative interaction strategies • Central vs. Distributed representation		

Environmentally Benign Design



Demanufacturing - disassembly, inspection, cleaning, grinding, melting, etc.

Remanufacturing - refurbishment for reuse

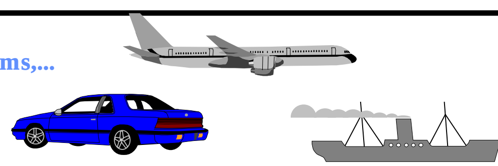
Material Recycling - reuse material after destroying part geometry

Product Recycling - reuse component or module without destroying part geometry

- Primary Reuse - reuse for original purpose
- Secondary Reuse - reuse for a different, lower quality use

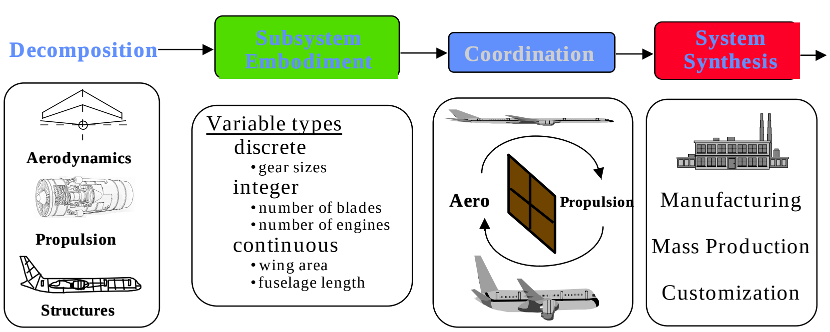
Challenges

In the design of complex systems,...



...what are some challenges?

Computation, Uncertainty, Mixed-Variable, Optimization



Intellectual Questions

Primary Intellectual Question:

How can product realization teams *accomplish more with fewer resources* in a globally competitive environment?

Questions to be answered in this class

- How to effectively search design option space ? (Optimization for continuos and mixed-variable problems)
- How to create design model or metamodel based on physical or computer experiments? (design of experiments, response surface methodology)
- How to make decisions when there are tradeoffs between multiple attributes? (multiobjective optimization, utility optimization, group decision making)
- How to make decisions under uncertainty ? (modeling, propagation, risk analysis, utility theory, robust design)

Design as Decision Making

- A decision is an irrevocable allocation of resources.
- A decision is an action taken from among a set of options.
- There is rich literature on decision analysis (decision science).

Characteristics of Decisions

- To make a decision one must have
  - Options (choices, alternative)
  - Expectations (distribution of performance outcome)
  - Values (metric by which options will be judged)
- Almost all decisions are made under conditions of uncertainty.
- Decisions are good or bad, not right or wrong
  - value can differ from person to person
  - perceptions of uncertainty can differ from person to person

Important Lexicon

- Design variable, design parameter..
- Attribute, criteria, objective, goal, Specifications, requirements ….
- Utility, Value, aggregated preference function, objective function…
- Information, Knowledge, design
- Uncertainty, risk, imprecision, ambiguity

Exampels of Lexicons (1)

- Attribute
  - performance parameters, synonyms for propreties and characteristics
  - Example: gas mileage, horsepower of a car
- Criteria
  - measure of effectiveness, basis for design evaluation
  - Criteria are emerging as a form of attributes or objectives.
  - Example: gas mileage, production cost
- Objective
  - something to be pursued to its fullest
  - An objective indicates the direction of change desired
  - Example: A car manufacture wants to maximize gas mileage or minimize production cost or minimize its level of air pollution.

Exampels of Lexicons (2)

- Specification
  - statements of the product requirements in terms of performance, time available, money to be spent, etc.
- Goal
  - Objective with indicated targets
- Utility
  - quantified value
  - preferences under conditions of risk

Mathematical Fundamentals of Engineering Design

- Set Theory - Design is a collection of options
- Measurement System - How to compare different designs

Godel-Bernays Axiomatic Set Theory

- Definition 1: A class is a collection **A** such that one can determine whether or not any object $x \in$ **A**.
- Definition 2: A set is a class **A** such that there exists another class **B** with **A** $\in$ **B**.

Bernays, P. *Axiomatic Set Theory*, North-Holland, New York, 1968.

Implications of Set Definitions

- Given a set of solutions, we can never deduce the universe of all possibilities which the set is from.
- Given a collection of designs with satisfy a requirement, we can never compute the universe of all possible designs which satisfy the requirement.
- Set structure is needed among the alternatives before any evaluation can be made.
- Given any arbitrary object, one can determine whether the object is in the set or not.
- Any two given objects in the collection can be differentiated.

Computations on Set

- Operations on Sets
  - Intersection and Union
- Functions on Sets
  - A function *f* on a set takes elements of the set and maps them to another set.
  - A composition of functions is a concatenation (sequence) of the function evaluations.
  - *Injective* (one to one) and *Surjective* (onto)
  - *Bijective* (or *invertible*): if *f* is both injective and surjective.
- Related topics - Topology (a point set)

Measurement Theory

- To know which designs are better than others
- To order the values
- In design, many measures don't have units.
- Different scales
  - Ordinal Scale
  - Interval Scale
  - Ratio Scale
  - Composite Scale

Ordinal Scales

- The ordering of the goodness of options (no numerical values are used)
- Ordering is called *weak* when equivalence is hold.
- Says nothing about how far apart one option is from the other.

Constructing of Ordinal Scales
Pugh Charts

- Procedure
  - Identify a datum alternative
  - Rank every other alternative as better/same/worst/ for each criterion . -1 for worse, 0 for same, 1 for better.
  - Sum and ranks for an overall evaluation of each configuration.
- If a final results has an alternative ranked at +3, it means the alternative is better than the base point by a net of 3 criteria.
- The result should not be used as an overall rank, but rather as an indicator of what to do next in the selection process.
- More useful for improving conceptual designs.

Interval Scales

- A numerical scale reflecting *how much* better or worse one design is compared to the others.
- Constructed for criteria that have no identifiable units of measure
- Need to have reference designs (x_0 - base point and x_1 - metric alternative) at two ends of performance for a particular criteria
- Constructed for a single criterion but not a design
- Often used when only qualitative information is available (safety, reliability, simplicity).

Constructing Interval Scales

- The lottery method (scale [0 1])
  - For an alternative x_i, on a scale of 0 to 1, what is your belief *f* that you are indifferent between (1) receiving the criteria performance provided by x_i or (2) receiving the criteria performance provided by x_1 (metric alternative) with certainty *f* and receiving the performance provided by x_0 (base point) with certainty 1-*f*?
- Assign ratings based on linguistic descriptions (scale [1 10])

Ratio Scale

- Often used for physically meaningful (quantitative) criterion (cost, power, speed)
- Normalized based on best possible and worst possible design options.

Normalization and Merit Function

- Normalized Rating
  - larger indicates preference $R = (r - r_{min})/(r_{max} - r_{min})$
  - smaller indicates preference $R = 1 - (r - r_{min})/(r_{max} - r_{min})$
- Merit Function
  - $M = \sum W_i R_i$

Specifying a Design Option

- Configuration - design layout and other configurational decisions
- Parameterization - identify the numerical quantities that specify each element in the system
- System specification consists of a set of values of system parameters together with its configuration.
- Describe product vs. describe process
- Include all phases of system life cycle
- Consider both physical and nonphysical options

The Phases in a System Life Cycle

- Predesign - research and development
- Design
- Test and design validation
- Manufacture/deployment
- Distribution/Sales
- Use
- Maintenance/repair/refurbishment
  Disposal

Nonphysical Options

- Nonphysical options are systems options that are not restricted by laws of nature
  - speed limit
  - interval between overhaul
  - method of attaching the driven gear to its shaft
- Nonphysical options interface with physical options - it is important to identify them.

The Concept of System

- *System* - a collection of entities that perform a specified set of tasks.
- System performs *functions*, or process, which results in an *output*.
- *Block diagram* - representation of system elements (inputs, outputs, system function).
- Different viewpoints results in different representations.

Hierarchical Levels of Modeling

- A Model separates the system from its environment.
- Anything "crosses" this boundary is a *link* between the system and its environment.
- A system is made up of components that are systems with their own functions and input/output.
- Every system is analyzed at a particular level of complexity that corresponds to the interests of individual who studies the system.

Model

- A model is an abstract description of the real world giving an approximate representation of more complex functions of physical systems.
- Two abroad categories
  - physical (iconic) model
  - symbolic model (by means of tools developed by human for abstraction)
    - drawings, verbalization, logic, and mathematics.
  - Symbolic models are more abstract and can be created at low cost.

Informal Models

- A designer’s interpretation of a description of the customers’ needs, engineering requirements, manufacturing and other product requirements, along with the designer’s interpretation of the conceived solutions.
  - Descriptions could be linguistic, graphical, acoustic, tactile, or aesthetic.
    - Vehicles must have a “comfortable” ride.
    - Food production plants must produce the “right” taste.
  - Unmeasureable concepts such as “configurations”, “physical principles”.
- Informal models must be converted into formal models to allow analytical (computational) treatment.

Elements of Formal Models

- *Design Space* - the *set* of considered possible design options, described using *design variables*, over which we have a direct choice.
  – Design variables must be controllable by the designer.
  – Design variables are not necessarily parameterized.
- *Design parameters* - quantities that are given specific value in any particular model statement.
- *Constant parameters* - quantifies fixed by underlying phenomenon.
- *Performance variables* - measureable quantities to express informal objectives
- *Mathematical relations* - equalities and inequalities that relate the above.

Analysis and Design Models

- *Analysis Models* - developed based on the principles of engineering science (once assumption is made, the model is unique).
- *Design Models* - constructed from the analysis models for specific prediction tasks and are problem dependent.

A Basic Modeling Approach (Otto & Wood)

- *Step 1 : Identify a Flow*.
  - Identify a material, energy, or information flow associated with each effect of the product concept
- *Step 2: Identify a Balance Relationship*
  - identify a balance relationship of the flow
- *Step 3: Identify the boundaries*
  - How is it loaded? How does it interact as a system with its environment (including human interface)? What are the inputs and outputs across the boundaries
- *Step 4: Derive an Equation (set of equations)*
  - Derive the analytical relationships
- *Step 5: Use the resulting model to explore design options (optimization)*

Optimization Models for Design

- The selection of a set of *(design) variables* to describe the design alternatives.
- The selection of an *objective* expressed in terms of the design variables, which we seek to minimize or maximize
- The determination of the a set of *constraints* expressed in terms of design variables which we seek to satisfy

Saw Mill Operation

A company owns two saw mills and two forests. See table for the capacity of each mill in logs/day and the distances between forests and mills. Each forest can yield up to 200 logs/day for the duration of the project and the cost of the transport the logs is estimated at 15 cents/km/log. At least 300 logs are needed each day. Formulate the problem to minimize the cost of transportation of logs each day.

	Distance (km)		
Mill	Forest 1	Forest 2	Mill Capacity/day
A	24.0	20.5	240 logs
B	17.2	18.0	300 logs

Minimum Weight Tubular Column Design

Design a minimum weight tubular column of length *l* supporting a load *P*. The column is fixed at the base and free at the top.

Models in Decision Making

- Accuracy
  - Will the system, as designed, work?
- Resolution
  - Which of two system alternatives is the better?
- Completeness
  - Do I properly understand the system?

Sources of Error in Symbolic Models

- What is a good model?
- Sources of error in symbolic models
  - abstraction error (error of model structure)
  - algorithmic error
  - Data error
  - computational error (computer only deal with a finite set of rational numbers rather than the full number set).
- Monte Carlo model cannot conduct a perfectly random and complete sampling.

Why Optimize?

- Optimization is decision making.
- Formal optimization methods enable us to consider infinities of options.
- There is usually a lot to be gained through formal optimization.

Classifications of Optimization Problems

- **By model**: single variable or multiple variable, constrained or unconstrained, linear or nonlinear problem
- **By solution method**: Linear programming, Sequential quadratic programming, integer programming, golden section, steepest descent, geometry programming, dynamic programming, etc.
- **By derivatives**: gradient-based method, nongradient based (random, simplex method, Genetic Algorithms) etc.
- **By nature of information**: deterministic vs. probabilistic models

Mathematical Concepts for NLP

- Local & global optimums
- Convex & concave sets
- Convex, concave, nonconvex (or nonconcave) functions
- <u>Theorem:</u> Suppose the feasible region *S* for NLP is a *convex* set. If f(x) is *convex/concave* on *S*, then any local *minimum/maximum* is an (global) optimal solution to this NLP.

Optimality Conditions

- **Necessary:** Conditions must be hold at any optimum for a problem (If x* is the optimal solution, then the condition must be satisfied).
- **Sufficient:** Conditions which, if satisfied at a point, guarantee that the point is an optimum (If the condition is satisfied, then x* must be the optimal solution).

Multiple Variable Unconstrained NLP-Optimality Condition

- *First-order necessary condition*
  - If x* is a local optimum, then $\nabla f(x^*) = 0$. (x* called a stationary point).
- *Second-order necessary condition*
  - If x* is a local minimum/maximum, then $\nabla f(x^*) = 0$ and Hessian matrix H(x*) is negative/positive semidefinite.
- *Sufficient Condition*
  - if (i) $\nabla f(x^*) = 0$ and
  - (ii) H(x*) is positive/negative semidefinite, then x* is the local minimum/maximum.

Kuhn-Tucker First-order Necessary Conditions for Constrained Problems

Max $f(x)$

s.t. $h_k(x) = 0$

$g_i(x) \le 0$

If x* is the optimum and a regular point, then

$$L(x,\lambda) = f(x) - \sum \lambda_k h_k(x) - \sum u_i g_i(x)$$

$$\frac{\partial L}{\partial x} = 0 \qquad \nabla f(x^*) - \sum \lambda_k \nabla h_k(x^*) - \sum u_i \nabla g_i(x^*) = 0$$

and $u_i g_i(x^*) = 0$;

$u_i \ge 0$; (no restriction of the sign of λ)

$h_k(x^*) = b_k$;

$g_i(x^*) \le b_i$;

Interpretation of Lagrange Multipliers (K-T condition)

- Economic Interpretation - shadow price
  - The rate of change of the optimal value *f* w.r.t. to b_i of *an active constraint* is given by the optimal value of λ_i (or u_i)

$$\frac{\partial f^o}{\partial b_i} = \lambda_i^o$$

- Geometrical Interpretation
  - Function gradient is a linear combination of the *active* constraints' gradients at the optimal point.

$$\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla g_i(x^*)$$

k - # of active constraints

Special Issues

- Minimization instead of maximization

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

- NLP in which variables must be nonnegative
  - $x_i \geq 0$.
- Constraint qualification - regularity condition
  - K-T condition may fail at an irregular point, though the point may be optimal!

Notes on the use of K-T conditions

- K-T conditions can be used to check or find *candidate* optimal points.
  - Need to check all equality and inequality constraints for *feasibility*.
  - Calculate all Lagrange multipliers and ensure those for inequality constraints are *nonnegative*.
- K-T condition may fail at an irregular point, though the point could be optimal.
- Whether it is truly maximum or minimum depends on H_L (convexity of f , g, and h all together).

Second-Order Conditions for Constrained Optimization

- Sufficient condition for convex/concave problems:
  - If f(x) is a convex/concave objective function defined on a convex feasible region (constraint set), then Kuhn-Tucker conditions are necessary as well as sufficient for a Global minimum/maximum.
- Necessary condition for general problems
  - Let there be nonzero feasible directions ($d \neq 0$) satisfying $\nabla h(x^*)^T d = 0$ and $\nabla g(x^*)^T d = 0$ (for active constraints), then if x* is a local minimum/maximum point, it must be true that $Q \geq 0$/ $Q \leq 0$ where $Q = d^T \nabla^2 L(x^*) d$.

-continued

- Second order sufficient conditions for general constrained problems
  - Let x* satisfy the first-order k-T necessary conditions. Define nonzero directions ($d \neq 0$) as solutions of the linear systems $\nabla h(x^*)^T d = 0$ and $\nabla g(x^*)^T d = 0$ (for active constraints). Also let $\nabla g(x^*)^T d \leq 0$ for those constraints with $u_i = 0$, if Q>0/Q<0, then x* is an isolated local minimum/maximum point.

Numerical Search Methods

- Zeroth order - look everywhere
- First order - go uphill
- Second order - aim toward the top of the hill.

General Procedure of Optimization Algorithms

- Optimization seeks to find a perturbation to an existing solution which will lead to an improvement.
- Most optimization algorithms apply a two-step process

  $\mathbf{X}^{k+1} = \mathbf{X}^k + \delta^k \mathbf{d}^k$ (k - kth iteration)

  – $\mathbf{d}^k$ is the **search direction** that will improve the current solution.

  – δ^k represents the **step size** in the search direction.
- The users always need to choose an initial design X^0 to start.
- The process will iterate (k=1, 2.. N) until convergence.

Measures for a Good Algorithm

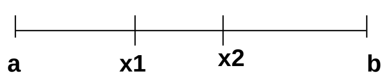
- **Robustness –**must be reliable and be able to converge to the solution point starting from any given starting point.
- **Generality –** should not impose restrictions on the model's constraints and objective functions.
- **Accuracy –** ability to converge to precise mathematical optimum point is important, though it may not be required in practice.
- **Ease of use –** by both experienced and inexperienced users. Should not have problem dependent tuning parameters.
- **Efficiency –**1) a faster rate of convergence requiring fewer iterations, and 2) least number of calculations within one iteration.

Methods for Single Variable Problem (unconstrained)

- Polynomial approximation or point-estimation method.
- Region elimination method or line search method:
  - Bracketing algorithm
  - Interval halfing
  - Golden Section
- Methods Requiring Derivatives
  - Newton-Raphson Method
  - Bisection method
  - Secant method
  - Cubic search

Golden Section Algorithm (MAX Problem)

```
Initialize:
x1 = a + (b-a)*0.382
x2 = a + (b-a)*0.618
f1 = ƒ(x1)
f2 = ƒ(x2)
Loop:
if f1 < f2 eliminate [a, x1], new interval (x1, b]
    a = x1; x1 = x2; f1 = f2
    x2 = a + (b-a)*0.618
    f2 = ƒ(x2)
else  (f1 ≥ f2)
eliminate [x2, b], new interval [a, x2]
    a=a; b = x2; x2 = x1; f2 = f1
    x1 = a + (b-a)*0.382
    f1 = ƒ(x1)
endif
```



Note: a, b, x1, x2, f1, f2 are updated in each loop.

Search requires only one new point at each loop.

Stop when interval is small enough

Methods for Multivariable Unconstrained Problems

- Without the use of derivatives
  - Simplex Method
  - Hooke and Jeeves method
  - Conjugate direction method
- Use first order derivatives
  - Steepest ascent (descent) method
  - Conjugate gradient method
  - Quasi-Newton method
- Use second order derivatives
  - Newton method

Algorithm of "Steepest Ascent"

- Basic Steps in each iteration k
  - calculate the gradient
  - set the search direction as $d^k = \nabla f(x^k)$
  - Search the best step size δ^k in the direction of d^k which *maximizes* the function $f(\delta^k)$.
- For minimization, the algorithm is called "steepest descent"
  - $d^k = -\nabla f(x^k)$
  - δ^k is sought to *minimize* the function.

Features of "Steepest Ascent"

- Successive directions are orthogonal to each other.
- The method is locally convergent by assuring $f(X^{k+1}) > f(X^k)$.
- Rate of converge is slow (one reason is because only the first-order information about the function is used). The rate is related to the condition number of Hessian at the optimum.
- Information calculated at the previous iterations is not used, which is inefficient.
- f is substantially increased in the first several steps.
- The direction of $\nabla f(x^k)$ is good in a local sense, but not in a global sense.

Fletcher-Reeves Conjugate Gradient Method (for MAX)

1. Set iteration counter, k=1
2. Calculate the gradient, $\nabla f(x^k)$
3. If k=1 then $d^k = \nabla f(x^k)$

   Else $\beta = \dfrac{\left|\nabla f(x^k)\right|^2}{\left|\nabla f(x^{k-1})\right|^2}$ $\quad d^k = \nabla f(x^k) + \beta d^{k-1}$
4. Search the best δ in the direction of d^k
5. Set k=k+1
6. Repeat from step 2 until convergence to the optimum.

Features of the Conjugate Gradient Method

Successive directions are not orthogonal to each other. Directions tend to cut diagonally through the orthogonal steepest ascent directions.

Easy to Program

Converges in N or fewer iterations for quadratic problems

Simple addition to the steepest ascent method but has dramatically better performance

Newton's Method (For Minimization)

One dimension Newton's method: $x_{k+1} = x_k - \frac{y'(x_k)}{y''(x_k)}$

How to extend it two multidimensional problem?

Go back to Taylor Expansion:

$$y(x) = y(x_k) + \nabla y(x_k)^T (x - x_k) + 0.5(x - x_k)^T H(x_k)(x - x_k)$$

$$y(x_{k+1}) = y(x_k) + \nabla y(x_k)^T (x_{k+1} - x_k) + 0.5(x_{k+1} - x_k)^T H(x_k)(x_{k+1} - x_k)$$

$$\frac{y(x_{k+1}) - y(x_k)}{(x_{k+1} - x_k)} = \nabla y(x_k)^T + H(x_k)(x_{k+1} - x_k) = 0$$

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla y(x_k) \qquad d = -H(x_k)^{-1} \nabla y(x_k)$$

Features of Newton Method

Requires function values, first and second derivatives. Preferred when **H** is easily calculated.

Good properties (fast convergence) if started near solution. Converges in one iteration for truly quadratic problems.

However, needs modifications if started far away from solution.

To overcome this, several modifications are often made.

- One of them is to add a search parameter α in front of the Hessian. (similarly like in steepest descent). This is often referred to as the modified Newton's method.
- Other modification focus on enhancing the properties of the second and first order gradient combination.
- Quasi-Newton methods build up curvature information by observing the behavior of the objective functions and its first order gradient. This info is used to generate an approximation of the Hessian.

Methods for Multivariable Constrained Problems

- Exterior penalty function methods
- Interior (barrier) function methods
- Gradient projection methods
- Generalized reduced gradient methods (GRG)
- Sequential linear programming method (SLP)
- Sequential quadratic programming method (SQP)
- Modified Feasible Direction Method

Penalty Methods

Penalty methods use a mathematical function that will increase the objective for any given constrained violation.

General transformation of constrained problem into an unconstrained Problem

T(x, R) = f(x) + P(R, g(x), h(x))

where

f(x) is the original function of the constrained problem.

R is a penalty scalar

P is a penalty function that imposes penalty for feasibility

T(x,R) is a transformed objective function.

Here is a typical penalty function:

$$P(R, g(x), h(x)) = R\left\{\sum_{i=1}^{p}[h_i(x)]^2 + \sum_{i=1}^{m}[g_i^+(x)]^2\right\}$$

where $g_i^+(x) = \max(0, g_i(x))$

Two Classes of Sequential Methods

Two major classes exist in sequential methods:

1) First class involves a sequence of infeasible points and feasibility is obtained only at the optimum. These are referred to as **penalty function** or **exterior-point penalty function** methods.
2) Second class is characterized by the property of preserving feasibility at all times. These are referred to as **barrier function** methods or **interior-point penalty function** methods.

General Barrier Function

$T(x, R) = f(x) + B(R^{(k)}, g(x))$

where k approaches infinity, $R^{(k)}$ approaches 0.

Typical Barrier functions B are the inverse or logarithmic.

$$B = (1/R)\sum_{i=1}^{m}[g_i^{-1}(x)]^2 \qquad B = (-1/R)\sum_{i=1}^{m}[\log(-g_i(x))]$$

The barrier function methods are applicable only to the **inequality constrained problems**.

Pros and Cons of the Penalty and Barrier Function Methods

Penalty function method:

- Applicable to equalities as well as inequalities.
- Starting point can be arbitrary.
- The method iterates through the infeasible region where the function is undefined.
- If the iterative process terminates prematurely, the final design may not be feasible and hence not usable.

Barrier Function Method:

- Applicable to inequalities constrained problem only.
- Starting point must be feasible.
- The method always iterates through feasible region. If it terminates prematurely, the final design is feasible and hence usable.

Sequential Linear Programming (SLP)

Algorithm:

1) **Linearize** the objective, f(x), and constraint functions, h(x) & g(x) at x^0.
2) **Solve the linear approximation** using the simplex or other algorithms.
3) **Iterate** until convergence to an optimum.

Linearization:

e.g., $$g(\underline{X}) = g(\underline{X}^0) + (X_1 - X_1^0)\left(\frac{\partial g}{\partial X_1}\right)_0 + (X_2 - X_2^0)\left(\frac{\partial g}{\partial X_2}\right)_0 + ... + (X_n - X_n^0)\left(\frac{\partial g}{\partial X_n}\right)_0$$

Move Limits:
Because the problem is nonlinear, the linearization is only good in the small region about x^0. Therefore, the change vector, Δx , i.e, $x - x^0$, must be restricted to say 20% of x^0.

Move limits are always determined by trial and error, depending on the degree of nonlinearality.

Features of SLP

Easy to program.

Move limits must be reduced as the optimization progresses to ensure solution for those cases where there are fewer active constraints than design variables at the optimum (under constrained).

Better suited for problems that have predominantly linear constraints.

Feature a quadratic rate of convergence when the optimum is at the vertex of the feasible region.

Exhibit slow convergence for problems with nonvertex optimum solutions.

SLP is not considered to be a good method by the theoreticians (the method is not convergent).

Experience has shown that SLP is a powerful and reliable tool if coded with care.

Sequential Quadratic Programming

- Employ second-order approximation.
- Use Newton’s method or Quasi-Newton methods to solve directly the K-T optimality conditions.
- Use quadratic approximation of the lagrange function with a positive definite approximation of its Hessian and linear approximation of the constraints.
- Globally convergent under certain mild assumptions.

Feasible-Direction Method (Minimization)

Iterations: $\mathbf{X}^{k+1} = \mathbf{X}^k + \alpha^k \mathbf{d}^k$

Emphasis: determining a search direction which will rapidly reduce the objective function while maintaining a feasible direction.

The method introduced attempts to **"follow"**, without being precisely tangent to, the **boundaries**.

- A direction which reduces the objective function is called a **usable direction**

$$\nabla f(x^k) \bullet d < 0$$

- A direction which does not cause violation of the active constraints upon move is a **feasible direction**

$$\nabla g_i(x^k) \bullet d < 0$$

- Direction satisfying both conditions is a **useable feasible direction**.

Determination of the Search Direction

- For **convex** constraints, to prevent violation, we try to stay away from the active constraint boundary using **push-off**:

$$\nabla g_i(x^k) \bullet d^k + \theta \leq 0$$

- In practice, we can afford larger push off if the gains in the objective function is large (small value of $\nabla f_i(x^k) \bullet d^k$)

$$\nabla g_i(x^k) \bullet d^k - [\nabla f_i(x^k) \bullet d^k]\theta \leq 0$$

- At the same time minimizing $\nabla f_i(x^k) \bullet d^k$ means maximizing β.

$$\nabla f_i(x^k) \bullet d^k + \beta \leq 0$$

Search direction, continued

The problem becomes to solving the following sub-problem

Max β

s.t. $\nabla f_i(x^k) \bullet d^k + \beta \le 0$ **Usable**

$\nabla g_i(x^k) \bullet d^k + \theta_i \beta \le 0$ **Feasible**, i belongs to J

$d \bullet d \le 1$ **Bounds** on d

Where J = Set of Active Constraints

The sub-problem is easily solved by a method similar to linear programming.

...Continued

- If the design is infeasible, the sub-problem becomes

Max $-\nabla f_i(x^k) \bullet d^k + \phi\beta$

s.t. $\nabla g_i(x^k) \bullet d^k + \theta_i \beta \le 0$

$d \bullet d \le 1$

Φ is a weighting factor determining the relative importance of the objective and the constraints.

This drives the design toward the feasible region with minimal increase of f(x).

Features of GRG

- Reduced gradient method is closely related to simplex LP method because variables are split into basic and nonbasic groups.
- Reduced gradient optimality criterion is equivalent to Lagrangian optimality criterion
- Good for solving problems with a number of equality constraints.
- Relies on Newton's method to return to the feasible region in the search. A lot of efforts are spent on performing Newton iterations.
- Requires modifications when dealing with highly nonlinear functions.
- Maybe unusable because of the inability to satisfy the constraints precisely.

Goal Programming (GP)

- One of the multiobjective mathematical programming techniques is **Goal Programming** (GP)
- The term "goal programming" is used by its developers to indicate the search for an "optimal" program (i.e., a set of policies to be implemented) for a **mathematical model that is composed solely of goals**.
- This **does not represent a limitation**, on the contrary, any mathematical programming model (e.g., linear programming) may find an equivalent representation in GP.
- Further, not only does GP provide an alternative representation, it often provides a representation that is **more effective in capturing the nature** of real world problems.

Difference between Objectives and Goals

In Goal Programming a distinction is made between an objective and a goal:

- **Objective:** In mathematical programming, an objective is a function that we seek to optimize, via changes in the problem variables.

  The most common forms of objectives are those in which we seek to maximize or minimize. For example,

  Minimize Z = A(X)

- **Goal:** It is an objective with a "right hand side".

  This right hand side (G) is the target value or aspiration level associated with the goal. For example,

  A(X) = G

Deviation Variables

- The deviation variable is defined as:

$$d = G_i - A_i(\underline{X})$$

- The deviation variable d can be negative or positive.
- A deviation variable represents the distance (deviation) between the aspiration level and the actual attainment of the goal.

Positive and Negative Deviations

- The deviation Varible d is replaced by two variables

$$d = d_i^- - d_i^+ \qquad d_i^- \bullet d_i^+ = 0 \qquad d_i^-, d_i^+ \geq 0$$

- The preceding ensures that the deviation variables never take on negative values. The product constraint ensures that one of the deviation variables will always be zero.

- The goal formulation becomes:

$$A_i(X) + d_i^- - d_i^+ = G_i$$

Values of deviation variables

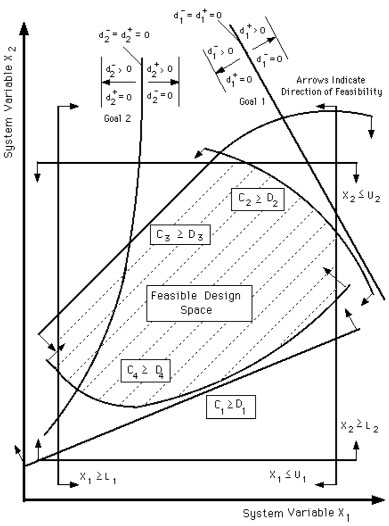
Note that a goal is always expressed as an equality.

When considering this equality, the following will be true:

$$\text{If } A_i \leq G_i \text{ then } d_i^- > 0 \text{ and } d_i^+ = 0$$
$$\text{If } A_i \geq G_i \text{ then } d_i^- = 0 \text{ and } d_i^+ > 0$$
$$\text{If } A_i = G_i \text{ then } d_i^- = 0 \text{ and } d_i^+ = 0$$

Frequently used Terms in Context

System variables,
deviation variables,
bounds,
constraints, and
goals
in a two-dimensional space



Prioritizing goals

Goals are not equally important to a decision maker.

How do we represent our preferences?

Two approaches:

1) **Assign weights** and calculate the sum of the deviation variables multiplied by their individual weights. In case the sum of the weights equals 1, then we speak of an archimedean formulation.
2) **Rank order** goal deviations in priority levels, often referred to as a preemptive formulation. The preemptive formulation does not exclude the assignment of weights.

Lexicographic Minimum

Given an order array $f = (f_1, f_2, \ldots f_n)$ of nonnegative elements f_k's, the solution given by $f^{(1)}$ is preferred to $f^{(2)}$ if

$$f_k^{(1)} < f_k^{(2)}$$

and all higher-order elements (I.e., $f_1, \ldots f_{k-1}$) are equal. If no other solution is preferred to f, then f is the lexicographic minimum.

Mixed-Integer Optimization

- Most optimization **algorithms** deal with **continuous variables**.
- In practice, **a large number** of optimization models have continuous and integer variables, which are called **mixed-integer** problems.
- When the constraints and objectives are all linear, they are called **MILP (Mixed-Integer Linear Programming)** problems which attracted a lot of attention in the field of **Operations Research** (allocation, scheduling,network, etc.)
- **Integers can be converted to Booleans**. This provides some nice properties which can be exploited, but typically, more variables are needed to characterize the same integer problem.
- **Mixed-Variable Problems** refer to those models have continuous, discrete or integer variables.

Complexity Issues in MIP

- The major difficult is due to the **combinatorial nature** of the integer variables (any choice of the integer variables results in a NLP on the continuous variables which can be solved for its best solution).

  - For instance, consider 100 Boolean variables (0, 1), we could have 2^{100} possible combinations.

- Often heuristics (rules of thumb) are used to limit the choices. In Operations Research, you may find the term **heuristic programming** occasionally.

- **Knowledge based systems** are a well known example of coding heuristics and have been connected to optimization algorithms.

- Another approach is based on **randomness and probability**, including monto-carlo simulation, simulated annealing, and genetic algorithms.

Outline of MILP Algorithms

Branch and Bound Methods

A binary tree is employed for the representation of the 0-1 combinations, the feasible region is partitioned into subdomains systematically.

Cutting Plane Methods

The feasible region is not divided into subdomains but instead new constraints, denoted as cuts, are generated and added which reduce the feasible region until a 0-1 optimal solution is obtained.

Decomposition Methods

The mathematical structure of the models is exploited via variable partitioning, duality, and relaxation methods.

Logic-Based Methods

Disjunctive constraints or symbolic inference techniques are utilized which can be expressed in terms of binary variables.

General Branch and Bound

- The main objective is to perform an enumeration of the alternatives **without examining all combinations of integer variables**.
- A key element is the representation of alternatives via a **binary tree**.
- **Branch:** Levels and nodes in the binary tree create subproblems (Pi)
- **Bound:** The solution of the relaxed problem (RP) provides a lower bound on the solution of problem (P).
  - In the relaxed problem, the whole or the part of the integer variables are freed as continuous variables.
  - If the solution of the RP is integer, the search could stop at this node because no candidate subproblems under this node will be better than the solution of this RP.

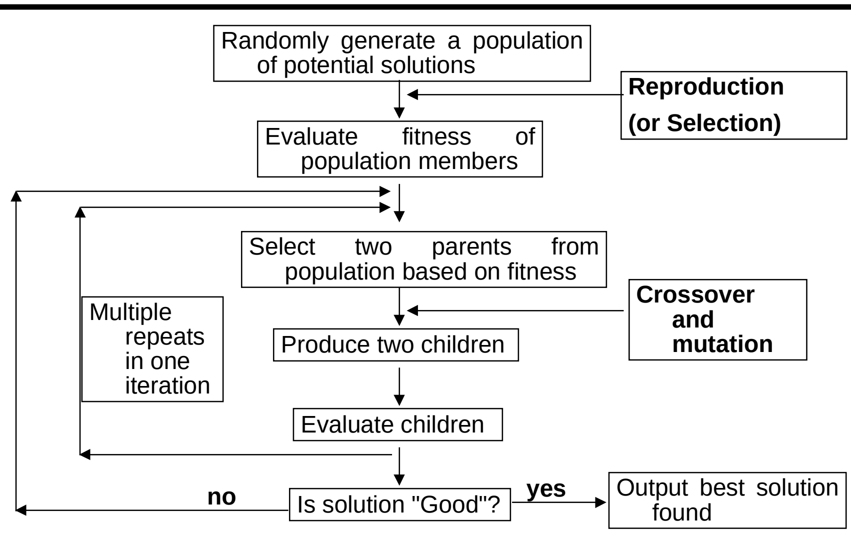
Stochastic Search Algorithms

- Monte Carlo methods employ **random number generation** as part of the iterative search for the solution of a (combinatorial) optimization problem.
- "Classical" Monte Carlo methods are nothing more than codes which randomly generate solutions and **keep track of the best**.
- More **"enhanced" methods** of growing importance are:
  - simulated annealing
  - genetic algorithms
- The methods in this category could be extended for the **nonconvex continuous problems** or problems with **disjoint design spaces**. They are sometimes called **non-gradient methods**. In general, these methods
  - Require a higher investment of **computational resource**.
  - Offer a greater potential for locating **global optimum**.

Historical Perspective of GA

- **Philosophical basis**: derived from biology, in Darwin's theory of "survival of the fittest".
- Genetic search belongs to the category of **stochastic search methods** – **random choice** is used as a tool to guide a highly exploitative search.
- Early ideas due to **Rechenberg** (60's).
- Contemporary format as proposed by **John Holland** (1975).
- Contributions of **Davis, DeJong, Goldberg and Grefenstette** useful in obtaining a better understanding of approach (1980s).

GA Flowchart



Language of Genetic Search

• **Genotype**: Total genetic prescription for the construction and operation of some organism. Several **chromosomes** comprise the genotype.

• A design variable is represented by **one chromosomal string**. A combination of such strings represents the complete design.

• Several such strings would model a **population pool.**

A Binary String Representation of the variable space

0 1 1 0	1 0 1	1 1	1 0 1 1
x1	**x2**	**x3**	**x4**

GA Basic Operators

- **Reproduction:** Individual strings with good objective function values f(x) are copied to form a new population. The bias towards strings with better performance can be achieved by:

  Assign each design i, a probability of selection as ,

  $$P_i = \frac{F_i}{\sum_N F_i}$$

  where F is the fitness function F=1/f(x)

- **Crossover:** is the mechanism by which new information is brought into the population pool - akin to exchange of genetic information between mating parents. **Crossover is used to promote the mixing of good genes within a population.**
- **Mutations:** In a random switching of bits in a string at randomly selected sites (very few!) to prevent premature loss of genetic information. **Mutation is used to promote genetic variations.**

Different Schemes of Crossover

- **One point crossover:** a point is determined in the chromosomes of the parents and the genes on each side of that point are recombined into two new combinations (children).
- **Multipoint crossover:** Multiple points are determined for recombining chromosome substrings.
- **Uniform crossover:** A flip of a coin determines whether a gene value of, say, parent 1 is given to child 1 or 2.

GA Adjustable Parameters

- Type of string representation (binary or other base)
- Discretization level (length of string, accuracy vs. efficiency)
- Population size
- Reproduction mechasim
- Percentage to which corssover occurs
- Customize crossover mechanism
- Percentage to which mutation occurs
- Customize mutation mechanism
- Maximum number of generations
- Number of process iterations/method of selecting first generation

Probabilities of Crossover and Mutation

Typical choices **P_c 0.6 – 0.8**

P_m 0.003 – 0.01

- Low P_c hinders the exploration of the search space.
- Excessively high P_c may result oscillatory behavior.
- High P_m is disruptive although some activity is desirable to guard against premature loss of information – especially true of small population sizes.

Population Size

- **Initial population** must be uniformly distributed over the range of variations in design variables.
- **Small population** sizes are likely to exhibit premature convergence.
- **Large population** sizes may prove to be computationally burdensome in those applications where function evaluations are expensive.

GA vs. Traditional Methods

- GAs work with a coding of the **entire set of design variables** and not the variables themselves.
- GAs do not optimize the design by advancing it form point to point. Instead, they **work from a population of designs**, advancing several designs in each cycle of evolution.
- Only function information is required– **no gradients**.
- Evolution and adaptation are implemented by **nondeterministic** transition rules. The method doesn't guarantee locally convergent.
- No. of function evaluations are directly linked to the population size and can be **prohibitively expensive** if GA is carried through **to convergence**.
- **Implicit parallelism** embedded in approach which yields significant computational advantage.

Simulated Annealing

- Basic idea is rooted in thermo-dynamics and the cooling (or annealing) of a liquid to a crystalline solid state:

  - If the liquid (e.g., molten steel) is cooled slowly and allowed to spend a long time near the freezing temperature, a perfect Crystal will form which has the <u>lowest state of energy</u>.

  - If the liquid is not allowed to spend a long time reaching an equilibrium near its freezing point and/or it is cooled too quickly, then the material will form a Crystal with many defects and a higher internal energy state (think of quenching of steel; the material may crack).

- Metropolis applied the same idea to simulate atoms in an equilibrium.
- Kirkpartrick was the first to use such a simulation in application to the combinatorial optimization problems.

Simulated Annealing – Metropolis Algorithm

- At each iteration, an "atom" is randomly displaced a small amount.
- The energy is calculated for each atom and the difference with the energy at its orginial location is caluated.
- If $\Delta E \leq 0$ then the new location is accepted.
- Otherwise, a random number R is generated and compared with P, the boltzmann probability factor $P = e^{(-\Delta E/KT)}$, where

  k = Boltzmann constant

  T= Temperature, and

  ΔE = the energy different between the two atom states
- If R > P, new location (high energy state) is accepted in the hope that the new location may eventually lead to a better location.
- If R < P, old location is retained and the algorithm generates a new location.

Simulated Annealing for Combinatorial Optimization

How to convert metropolis algorithm to general combinatorial optimization algorithm?

- State of system defined by values of the system design variables – assume the **state is denoted as x_1**.
- Compute the analogous **"energy" $E(x_1)$** corresponding to "state" x_1 – this can be the **objective function** with the constrained appended as a penalty.
- At some initial high temperature, state x_1 is varied at random to obtain new state x_2 for which $E(x_2)$ is computed.
  - If $E(x_2) < E(x_1)$, x_2 is accepted as the new state of the system.
  - If $E(x_2) > E(x_1)$, x_2 is accepted only by probabilistic decisions.
- As the optimization proceeds, the **temperature is lowered** and less solutions with high energy states (or objective functions) are allowed to remain.
- Hopefully, this **slow freezing** gives you the best solution.

Features of Simulated Annealing

- Enough equilibrium at a given temperature translates into allowing a **fixed number of random variations at a given temperature**.
- Temperature is lowered by an amount specified in the **“annealing schedule”** and **random variations** are repeated. Slower temperature reduction schedule and higher number of function evaluations imply higher computational costs.
- **Easy to accommodate discrete and integer variables**. No gradient calculations are required.
- **Ability to accept poorer design**, albeit with lower probability provides a mechanism to **escape from locally optimal design points**.

Other Remarks

- Genetic algorithms (and simulated annealing) are gaining ground as practical tools.
- They are relatively simple and elegant because of their analogy with Nature.
- Public domain and commercial codes exist.
  - However:
- They have a number of **control parameters** which affect the efficiency of solution.
- They typically perform a **large number of fitness function evaluations** (although less than an exhaustive search) and this may be unacceptable.
- They are **unconstrained minimization** algorithms, thus a penalty function or a lexicographic approach is required for constrained minimization problems.
- For a mixed integer problem, you may consider a hybrid GA which includes a regular optimization algorithm for the real variables.

Summary of Optimization Methods

- **Enumerative Schemes**: design spaces are too large to search.
- **Random Search (Monto-Carlo):** for large problems, cannot be expected to perform much better than enumerative schemes.
- **Nonlinear Programming**: efficient methods for a restrictive class of problems. Requirements of continuity and unimodality limit the scope of applicability.
- **Probabilistic Search Algorithms** such as simulated annealing and GA seek to bridge the robustness gap.

Terminology

- Probability
  - measure and evaluate uncertainty
  - related to set theory
  - related to experiments & historical data
- Sample space
  - set of possible outcomes formed by a set of events
  - discrete space (finite number of outcome)
  - continuos space (noncountable space)
- Event
  - a subset of sample space

Terminology (continued)

- Objective and subjective probability
  - Objective: tossing coin, based on the notion that the future will be just like the past.
  - Subjective: what will be the gas price in ten years? Derive from intuition.
- Mutually exclusive vs. totally unrelated events

Use of Bayes' Formula

- P(B) - prior probability - before the test
- Conduct a test for which event A occurs
- What is the posterior probability of B after the test - P(B|A)?
- Our state our information after the test is likely to be much better than the sate of information before the test.

Design of Experiment Techniques

- Empirical Modeling of Functional Relationships
  - when the design model is not available
  - when the design model (simulation program) is too complicated
- Data sampling for statistical analysis
- Identify factor significance
- Factors - input variables to be studied - design variables
- Response - output variable that is of interest - design performance
- Factor effects - change in the response when factor moves from the lower level to the upper level. The larger the effects, the more significant of the factor.

Experiment Designs

- Full Factorial
- Fractional factorial
- Central Composite Designs
- Latin Hyper Cube
- Orthogonal Array

Design of Fractional Factorial Experiments

- Generator and Defining Relation
  - defining the confounding structure of the design
- Word
- Resolution
  - the length of the shortest word in the defining relation
  - RIII, m.e. do not confound with m.e.; m.e. confound with 2 factor interaction
  - RIV, m.e. do not confound with 2 factor interaction, etc.
- Construct $2^{(k-p)}$
  - Generate a full factorial for (k-p) factors
  - Add columns for remaining p factors based on the generators

Rules in Designing Fractional Factorial Experiments

- Maximize resolution
  - the effects of same order are of equal importance
  - higher order effects are less important than lower order effects.
- Minimize Aberration - # of words with shortest length.

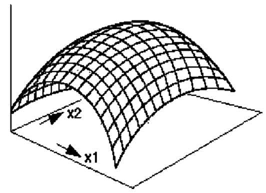
Response Surface Modeling Techniques

- Linear Regression
- Neural Networks
- Kriging Method
- MARS (Multivariate Adaptive Regression Splines)

In each of these categories, some methods belong to interpolation approaches, the others are associated with minimizing least square errors.

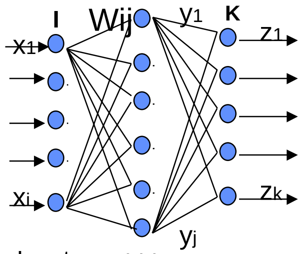
RSM and ANN

Response Surface Methodology (RSM)



$$y = \beta_0 + \sum_i \beta_i x_i + \sum_i \beta_{ii} x_i^2 + \sum \beta_{ij} x_i x_j$$

Artificial Neutral Networks (ANN)



Transformation: e.g.,

$$\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

Design of Experiment Techniques

- Empirical Modeling of Functional Relationships
  - when the design model is not available
  - when the design model (simulation program) is too complicated
- Data sampling for statistical analysis
- Identify factor significance
- Factors - input variables to be studied - design variables
- Response - output variable that is of interest - design performance
- Factor effects - change in the response when factor moves from the lower level to the upper level. The larger the effects, the more significant of the factor.

Experiment Designs

- Full Factorial
- Fractional factorial
- Central Composite Designs
- Latin Hyper Cube
- Orthogonal Array

Design of Fractional Factorial Experiments

- Generator and Defining Relation
  - defining the confounding structure of the design
- Word
- Resolution
  - the length of the shortest word in the defining relation
  - RIII, m.e. do not confound with m.e.; m.e. confound with 2 factor interaction
  - RIV, m.e. do not confound with 2 factor interaction, etc.
- Construct $2^{(k-p)}$
  - Generate a full factorial for (k-p) factors
  - Add columns for remaining p factors based on the generators

Rules in Designing Fractional Factorial Experiments

- Maximize resolution
  - the effects of same order are of equal importance
  - higher order effects are less important than lower order effects.
- Minimize Aberration - # of words with shortest length.

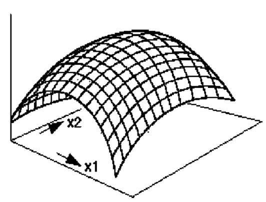
Response Surface Modeling Techniques

- Linear Regression
- Neural Networks
- Kriging Method
- MARS (Multivariate Adaptive Regression Splines)

In each of these categories, some methods belong to interpolation approaches, the others are associated with minimizing least square errors.

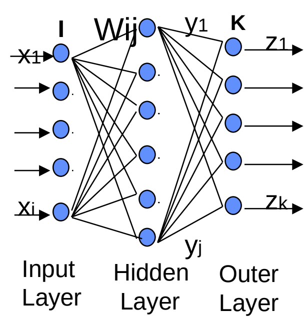
RSM and ANN

Response Surface Methodology (RSM)



$$y = \beta_0 + \sum_i \beta_i x_i + \sum_i \beta_{ii} x_i^2 + \sum \beta_{ij} x_i x_j$$

Artificial Neutral Networks (ANN)



Transformation: e.g.,

$$\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

DOEs Included in iSIGHT

L_i = # levels for factor i

- Full Factorial (L_1 x L_2 x L_3 …x L_i)
- Parameter Study (1 +L_1 + L_2 + L_3 …+L_i)
  - Study the sensitivity to each factor independent of other factors
  - Each factor is studied at its specified levels while others are fixed at their baseline.
- Orthogonal Arrays (fractional subset of full factorial)
- Central Composite
- Latin Hypercubes (user specifies the total number of samples n)
- User Defined

Central Composite Design

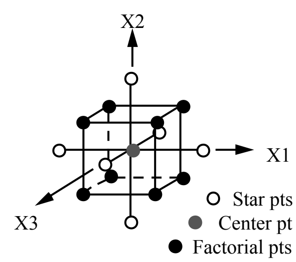
CCD consists of

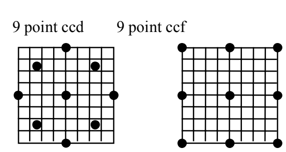
- Cube portion - A (fractional) factorial design
- Star points - $\pm\alpha$ from the origin
- Center point

Features

- Efficient in capturing nonlinearity and fitting 2nd-order model
- $\alpha > 1$ in general
- $\alpha = 1$, Face-centered CCD or called CCF
- $\alpha < 1$ inscribed central composite (CCI)

Run #	X1	X2	X3
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1
9	$-\alpha$	0	0
10	$+\alpha$	0	0
11	0	$-\alpha$	0
12	0	$+\alpha$	0
13	0	0	$-\alpha$
14	0	0	$+\alpha$
15	0	0	0



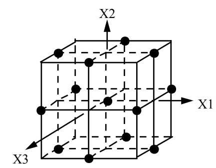


Box-Behnken Designs

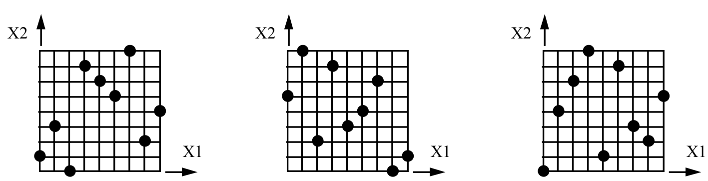
- Formed by combining 2^k factorial and incomplete block designs
- Efficient 3-level designs for fitting second-order model
- Good alternative to CCD (reduced from 5-level or 3-level)

- Poor prediction capability at corners
- Designs for 4, 5, and 6 factors require 25, 41, and 49 points.

Run#	X1	X2	X3
1	-1	-1	0
2	-1	1	0
3	1	-1	0
4	1	1	0
5	0	-1	-1
6	0	-1	1
7	0	1	-1
8	0	1	1
9	-1	0	-1
10	1	0	-1
11	-1	0	1
12	1	0	1
13	0	0	0



Latin Hypercubes Sampling (LHS)



9 points for 2 factors

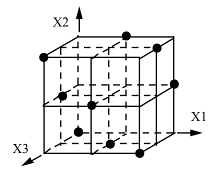
- *n* samples for *k* factors, *n* divisions on each factor direction
- Stratified sampling - each of the factor is sampled at *n* levels. Evenly distributed when projected to a single dimension.
- Randomly distributed therefore the design is not unique.

Orthogonal Arrays (Owen 1992)

- Any combination of *r* (strength) number of columns only appears once in the design as a row.
- Balanced sparse designs for any projection into *r* factors (*r* dimension projection)
- Balancing may lead to ill-conditioning of the correlation matrix when using Kriging method.

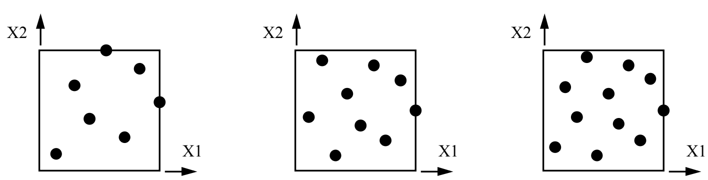
- Don’t project well into single dimensions as Latin hypercube designs.
- Number of samples is fixed at certain levels.

Run #	X1	X2	X3
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2



Strength r =2
Level l =3

Hammersley Sequence (Diwekar 1995)



2 factors with 7, 9, and 11 points

- Low-discrepancy experimental design for placing *n* points in a k-dimensional hypercube.
- Discrepancy *D* is a quantitative measure for the deviation of the sequence from uniform distribution. If the number of sample points is N and B is a subspace of the hypercube

$$D_N(B) = \sup \left| \frac{\text{point number in B}}{\text{N}} - \text{the probability of B in The Hypercube} \right|$$

- Providing better uniformity properties over the k-dimensional space than Latin hypercubes.

Issues in Optimal Design

- Criteria used in classical DOE theory (physical experiments) are not fully applicable for computer experiments.
- In computer experiments, the goal is to fill the space as evenly as possible with the minimum number of samples (called space-filling or stratified experiments).
- Achieving the optimal experiment criteria is often computationally extensive because of the highly nonlinearity and local optimality (Many research papers on this topic. Genetic algorithms and simulated annealing algorithms are applied).

D-Optimality Criterion in Classical DOE theory

For Linear Regression $y = X\beta + \varepsilon$

ε is a random variable with mean 0, variance σ^2

The least-squares estimate of β is $\hat{\beta} = (X'X)^{-1} X'y$

The covariance matrix of $\hat{\beta}$ is $(X'X)^{-1}\sigma^2$

The variance of the predicted $\hat{y}$ is $v(x) = f(x)'(X'X)^{-1} f(x)\sigma^2$

The most popular D-optimality criterion is

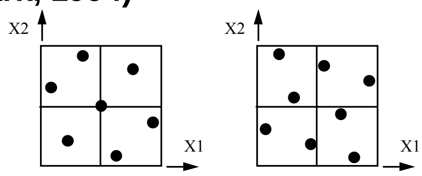
MAX $|X'X|$

To achieve: Low variances, Low correlation, and Low maximum $v(x)$ over the region

which indicates: satisfactory distribution of information throughout the whole region, fitted value as close as possible to the true value.

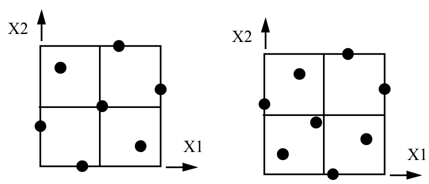
Optimal LHS

Minimization of IMSE (Integrated Mean Square Error) (Park, 1994)



- Good projection and well spread out
- Do not extend all the way to the edge

Maximization of minimum distance (Park, 1994)



- Good projection and well spread out
- Extend all the way to the edge

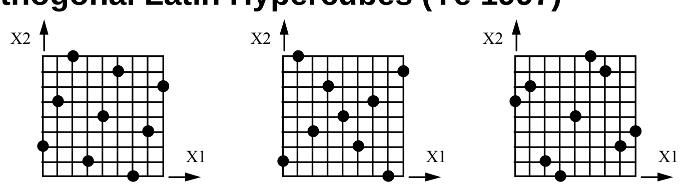
Optimal LHS (continued)

Maximum Entropy Design (Morris and Mitchell 1995)

Minimize Negative Entropy $E\left[-\log|\mathbf{R}|\right]$

Where $R\left(\mathbf{x}_i,\mathbf{x}_j\right)=\sigma^2\exp\left\{-\sum_{k=1}^{d}\theta_k\left(x_{i,k}-x_{j,k}\right)^2\right\}$ and $\theta_k \geq 0$

Orthogonal Latin Hypercubes (Ye 1997)



Orthogonality guarantees that the quadratic and interaction effects are uncorrelated with the estimates of linear effects.

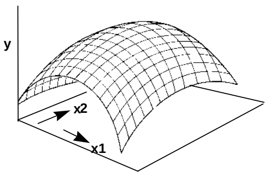
Easily constructed by purely algebraic means without the aid of computers.

Polynomial Regression

- General form of a PR Model

$$\hat{y} = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_i \sum_j \beta_{ij} x_i x_j$$

- Where:
  - β_i are parameters used to fit the model
  - Model is usually fit using least squares regression



A Second Order PR Model

- Remarks:
  - PR will not interpolate the data.
  - Difficult to take sufficient sample data to estimate all the coefficients in higher-order polynomials, particularly in large dimensions.

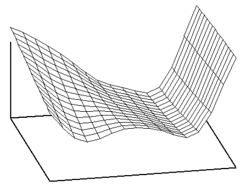
Background on Kriging Method

- Rooted in geostatistics - predicting true ore grade distributions based on sampled ore grades (in 2-D)
- Named after a South African mining engineer, D.G. Krige (1950s)
- Extended to k-dimension prediction
- Synonym of data mining
- Treat prediction function as a realization of a stochastic process with mean zero, variance σ^2, and non-zero covariance.

Kriging Model

- General form of Kriging model:

$$\hat{y} = \beta + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y_0} - \mathbf{1}^T\beta)$$



where: $\beta = (\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}^T)^{-1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y_0}$

  - **R** is correlation between each sample point

$$R(\mathbf{x}_0{}^{\mathbf{i}}, \mathbf{x}_0{}^{\mathbf{j}}) = \sigma^2 \exp(-\sum_{k=1}^{d} \theta_k \left(x_{0k}^{\ i} - x_{0k}^{\ j}\right)^2)$$

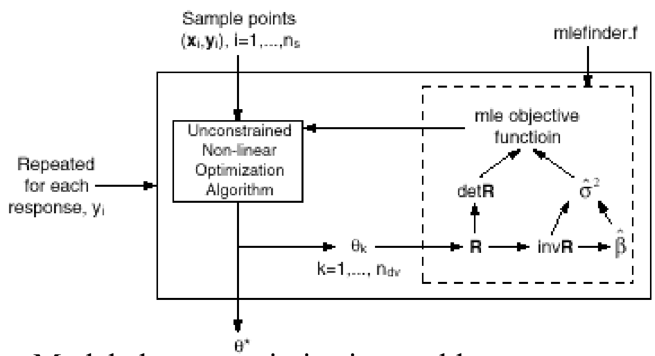
  - $\mathbf{r}^T(\mathbf{x})$ is correlation between new point and each sample point

$$\mathbf{r}(\mathbf{x}) = \left[R(\mathbf{x_0}^1, \mathbf{x}), \ldots, R(\mathbf{x_0}^n, \mathbf{x})\right]$$

  - θ_k are parameters used to fit kriging model

Remarks
  - Kriging model interpolates the sampled data
  - Efficient for highly nonlinear behavior
  - Optimization required to find θ_k

Building Kriging Models



Modeled as an optimization problem:

Find: θ_k $k = 1\ldots n_{dv}$, the number factors

Maximize maximum likelihood estimates $-\dfrac{[n_s \ln(\hat{\sigma}^2) + \ln|R|]}{2}$

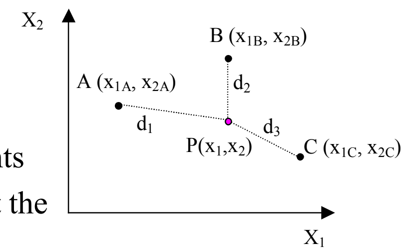
Radial Basis Function

- General form of RBF Model

$$\hat{y} = \sum_i a_i \left\| \mathbf{x} - \mathbf{x}_{0i} \right\|$$

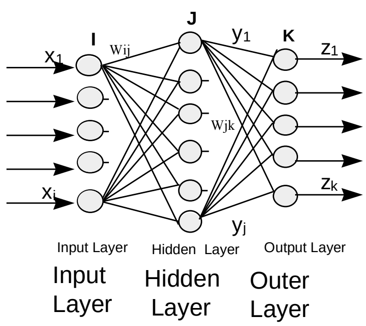
where:

- $\mathbf{x}_{0i}$ are the sample data points
- a_i are parameters used to fit the model
- ||•|| is the Euclidean norm(distance)



- Remarks:
  - As commonly applied, the method is an interpolating approximation.
  - a_i is found by replacing f(**x**) with known sample data and solving resulting linear system of equations.

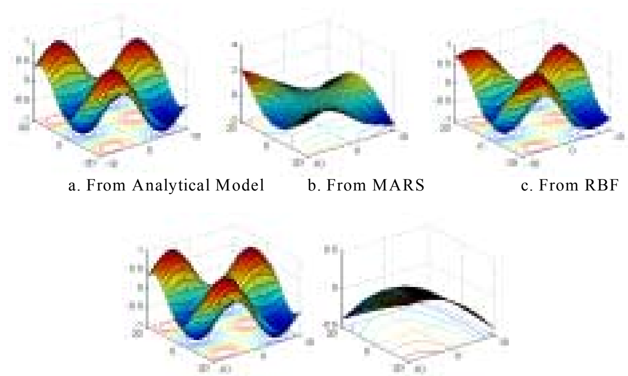
Neural Networks



$$\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

- Weights w_{ij} are identified by minimizing the least square error of the output .
- Nodes and layers can be used to transform highly nonlinear behavior.
- Networks can model any types of mathematical behaviors.
- A lot of try-and-errors are involved in selecting the number of nodes and layers.

Example



a. From Analytical Model b. From MARS c. From RBF

d. From KG f. From PR

Figure 12 Grid Plots for Problem 7

KG is extremely accurate for modeling the waving behavior for this particular case, while the RBF is the second best.

Confirmation Metrics

- **On the basis of confirmation samples**
  - ***R Square*** **(R^2)**

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} = 1 - \frac{MSE}{Variance}$$

  - ***Relative Average Absolute Error*** **(RAAE)**

$$RAAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n * STD}$$

  - ***Relative Maximum Absolute Error*** **(RMAE)**

$$RMAE = \frac{\max(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|)}{STD}$$

Robust Design

The fundamental principle of Robust Design is to improve quality by minimizing the effect of the causes of variation without eliminating the causes **(a concept for quality engineering).**

Two important aspects of Taguchi's Method for Robust Design:

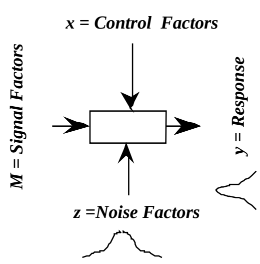
- Efficient experimentation to find dependable information about the design parameters.
  ---> Orthogonal Arrays

- Measurement of the quality during design and development
  ---> Signal to noise ratio

Our focus is to introduce the nonlinear programming methods for robust design and the extension of this method for sensitivity analysis and managing uncertainties.

P - Diagram

P – Diagram



- ***Signal Factors*** *(M)*: Parameters, which are set by a designer to express the intended value for the response of a product.

- ***Noise Factors*** *(x)*: Parameters, which cannot be controlled by a designer. The noise factors cause the response y to deviate from the specified target.

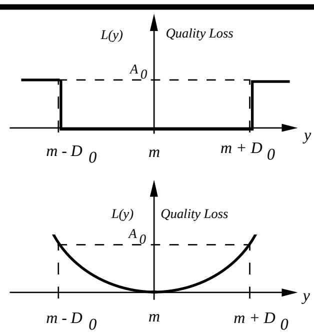
- ***Control Factors*** *(z)*: Parameters, which can be specified freely by a designer.

Types of Noise Factors

- **External Noise Factors –** sources of variability that come from outside the product.
- **Unit-to-Unit Noise –** unit-to-unit variability caused by manufacturing processes and materials.
- **Deterioration noise –** internal noise factor.

In order to benefit from the robust design procedure, the **noises must be evaluated either analytically** or experimentally.

Quality Loss Function



(a) Step Function

If $m–D_0 < y < m+D_0$, $L(y)=0$

Else, $L(y) = A_0$

(b) Quadratic Loss Function

$L(y) = k (y - m)^2$

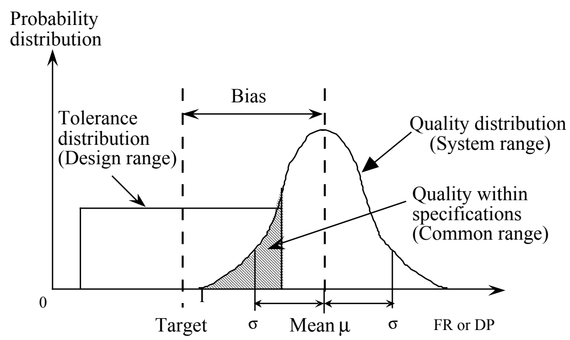
where

y	**is the quality characteristic of a product/process,**
m	**is the target value for y, and**
k	**is a constant, the quality loss coefficient.**

Objectives in Robust Design

Robust Design

- Bring the "mean" on target
- Minimize the dispersion (variation)–make the "bell shape" thinner.



Concepts of Mean and Variance

- **Mean – average value of all the observations in the sample; measures the central tendency of a distribution.**

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

- **Variance (σ^2) – measurement of the degree of dispersion of a distribution.**

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n-1}$$

Note: σ is named as the standard deviation.

Maximization of Signal-to-noise Ratio

- Nominal-the-best

$$\eta = 10\log_{10}\frac{\mu^2}{\sigma^2}$$

- Small-the-better

$$\eta = -10\log_{10}[\frac{1}{n}\sum_{i=1}^{n} y_i^2]$$

- Large-the-better

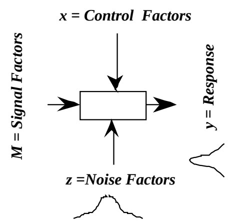
$$\eta = -10\log_{10}[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{y_i^2}]$$

The Selection of Adjustment and Tuning Parameters

- Adjustment (scaling) factors- those that have a significant effect on μ, but no effect on η.
- Tuning factors - those that have no effect on μ, but significant effect on η.
- Neural factors - those that have on effect on μ and no effect on η.

Robust Design: Type I

Reduce the influence on the system performance caused by the noise factors



Given

- **Response Surface model f(x, z)**
- **Mean and deviations of noise factors (μ_z, σ_z)**

Find

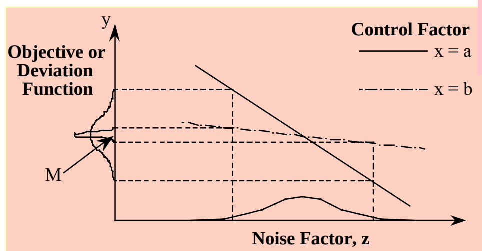
- **System Variables–Control factors x**

Satisfy

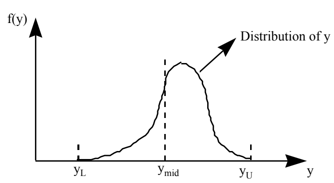
- **System constraints $g^*(x, z) \leq 0$**

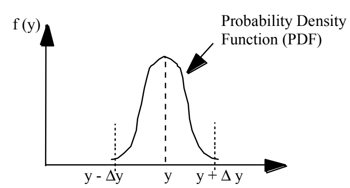
Min or Max

- **System objectives**
  - **Bring the mean on target μ_y**
  - **Minimize the variance σ_y**



The Variation of Design Performance





<u>Normal distribution</u> is often assumed based on the central limit theorem. The probability density function of a normal distribution is given as:

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

$\Delta y = k\sigma$, k is often taken as 3.

k (number of standard deviations)	1	2	3	4
% of performance conforming to distribution	84.13	97.73	99.87	99.997

Estimations of Mean and Variance

Approach I – Statistical Analysis based on Simulations

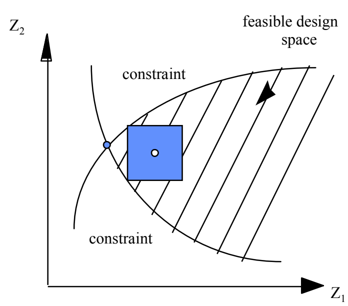
Choose sufficient # of points according to variations of noise factors; simulate the performance y; calculate mean and variance.

Approach II – Taylor Expansion

$\mu_y = f(x, \mu_z)$.

$$\sigma_y = \sqrt{\left(\frac{\partial y}{\partial z_1}\right)^2 \sigma_{z_1}^2 + \ldots\ldots\ldots + \left(\frac{\partial y}{\partial z_k}\right)^2 \sigma_{z_k}^2} \quad \text{or} \quad \Delta y = \sum_{i=1}^{k} \left| \frac{\partial f}{\partial z_i} \Delta z_i \right|$$

Sensitivity of the Constraint



- nominal optimum
- robust optimum

Worst Case Scenario:

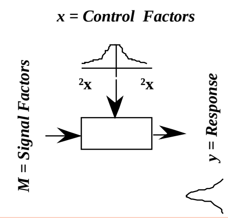
assuming that all variations of system performance may occur simultaneously in the **worst possible combinations** of design parameters and design variables.

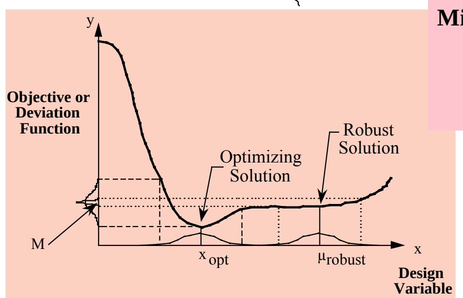
$g_j(x, z) \leq 0$ is changed to:

$$g_j(x, \mu_z) + \sum_{i=1}^{k} \left| \frac{\partial g_j}{\partial z_i} \Delta z_i \right|_{x, \mu_Z} \leq 0$$

Robust Design – Type II

Reduce the influence on the system performance caused by the control factors





Given
- **Response Surface model f(x)**

Find
- **Range of System variables–Control factors x, Δx**

Satisfy
- **System constraints g*(x) ≤ 0**

Min or Max
- **System objectives**
  - **Bring the mean on target μ_y**
  - **Minimize the variance σ_y**

$$\mu_{\hat{y}} = f(\mathbf{x})$$

$$\sigma_y = \sqrt{\left(\frac{\partial y}{\partial x_1}\right)^2 \sigma_{x1}^2 + \ldots\ldots\ldots + \left(\frac{\partial y}{\partial x_l}\right)^2 \sigma_{xl}^2}$$

Flexible Designs using Type II Robust Design

Conventional Optimization

Robust Design Solution

Departure from the traditional post optimal sensitivity analysis to reducing design "noise" upfront using robust design techniques.

Feasibility Assessment under Uncertainty (1)

How to formulate and evaluate g(x, p) ≥ 0 ?*

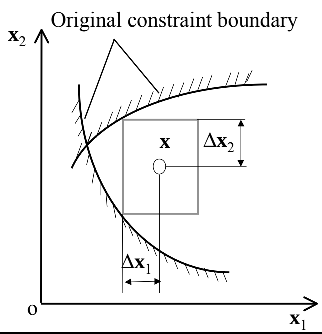
Methods do not require probabilistic analysis

Worst Case Analysis *(Parkinson et al. 1993)*

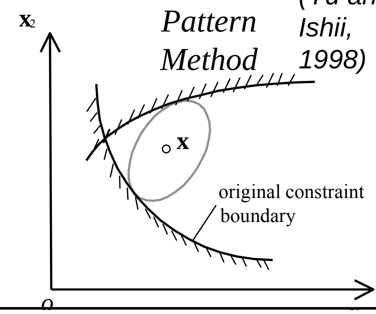
$$|\Delta g(\mathbf{X},\mathbf{P})| = \sum_{i=1}^{n}\left|\frac{\partial g}{\partial \mathbf{x}_i}\Delta \mathbf{x}_i\right| + \sum_{i=1}^{m}\left|\frac{\partial g}{\partial \mathbf{p}_i}\Delta \mathbf{p}_i\right|$$

$$\mu_g - |\Delta g(\mathbf{X},\mathbf{P})| \geq 0$$

Corner Space Method *(Sundaresan, et al., 1993)*



Variation Pattern Method *(Yu and Ishii, 1998)*



Feasibility Assessment under Uncertainty (2)

Methods require probabilistic analysis

Moment Matching Formation *(Parkinson et al. 1993)*

$$P[g(\mathbf{x},\mathbf{p}) \geq 0] = \Phi(\frac{\mu_g}{\sigma_g})$$

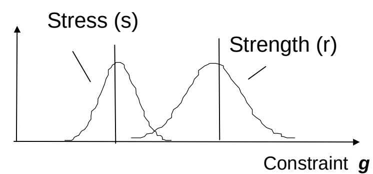
$$\mu_g - k\sigma_g \geq 0 \quad k = \Phi^{-1}(P_0)$$

Probabilistic Feasibility Formation (Eggert and Mayne, 1993)

$$P[g(\mathbf{x},\mathbf{p}) \geq 0] \geq P_0$$

$$P[g(\mathbf{x},\mathbf{p}) \geq 0] = \int_{g(\mathbf{x},\mathbf{p})\geq 0} f_{\mathbf{xp}}(\mathbf{x},\mathbf{p})d\mathbf{x}d\mathbf{p}$$

Similar to Reliability Assessment

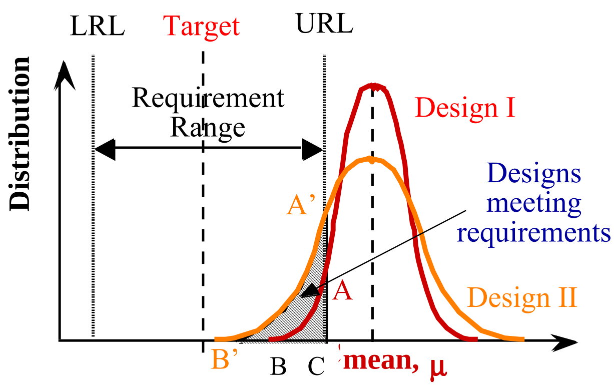


$P\{r - s \geq 0\} \geq P_0$

or $P\{g(\mathbf{x},\mathbf{p}) \leq c\} \geq P_0$

Limit State

Design Metric for Satisfying A Range of Specifications



$$C_{dl} = \frac{\bar{x} - LSL}{3\sigma} \qquad C_{du} = \frac{USL - \bar{x}}{3\sigma}$$

$$C_{dk} = \min\{C_{dl}, C_{du}\}$$

Remarks of Robust Design

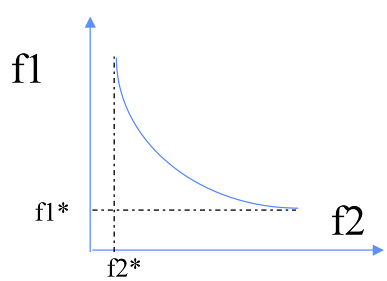
- Variations of performance y could be caused by both the variations of noise factors and control factors.
- Robust design principle could be applied to not only reducing the noise in a manufacturing process, but also reducing the noise in a design process.
- Type I robust design could be applied to handling uncertainty by modeling it as noise.
- Type II robust design could be used to develop flexible design solutions.
- Robust design needs to be treated as a multiobjective optimization problem. The ways for making tradeoffs between the two aspects of robust design are worth future research investigations.

Notions of Multiobjective Optimum Solution

- **An ideal solution:** or called superior solution or utopia point, is the one that optimizes each objective function simultaneously. If this solution is feasible, then there would be no conflict among objectives.
- **Nondominated solution:** or called noninferior solution, efficient solution or pareto-optimal solution, is the solution when there exists no other feasible solution that will yield an improvement in one objective without causing a degradation in at least one other objective.
- **Satisfying solutions:** A satisficing solution of Simon is a reduced subset of the feasible set which exceeds all of the aspiration levels of each attribute. Satisficing solutions need not be nondominated.

Pareto and Efficient Solutions

- **Pareto Solution - A point over the design space, if there is no other feasible point, which gives a lower minimum objective function for at least one of the objectives.**
- **Efficient Solution - Image of a Pareto solution in objective space**



The Compromise Programming (CP) Approach

- Basic idea in CP is the identification of an ideal solution (utopia point) as the one where each attribute under consideration achieves its optimum value.
- It has been mathematically proven that CP is superior to the WS method in locating the efficient solutions.

The Weighted Tchebycheff Approach in CP

General CP Problem

$$\text{minimize} \quad \|f(x) - u\|$$
$$\text{s. t.} \quad x \in X$$

where
$\| \cdot \|$ - metric of choice,
X - the design space formed by the design constraints and the range of design variables,
u - utopia or ideal point.

Weighted Tchebycheff CP(∞, ω) approach

min-max problem

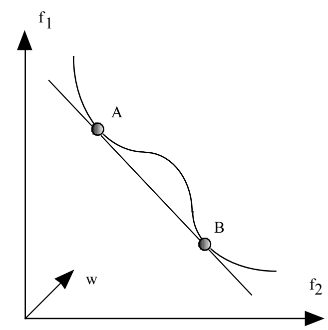
$$\min_{x \in X} \max_{i=1,2.} \{w_i(f_i(x) - u_i),$$

β problem

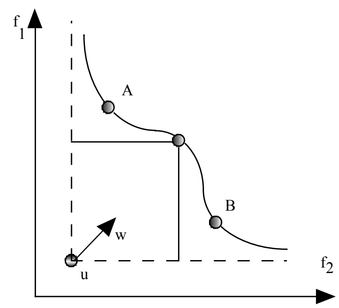
$$\text{minimize } \beta$$
$$\text{s. t.} \quad w_i(f_i(x) - u_i) \le \beta, \quad i = 1,\ 2$$
$$x \in X,$$

where
β - new positive variable.

Compromise Programming (CP) vs. Weight-Sum (WS) Method



Generating Efficient Solutions by the WS Method



Generating Efficient Solutions by the Weighted Tchebycheff Method

Conditions for Solution to be Necessary for Pareto Optimality

- For a multicriteria optimization problem with a continuous and smooth attainable region boundary near a Pareto optimal point c^o , the optimization with respect to the objective function will have solution c^o only if the attainable region boundary and the objective function constant value surfaces are tangent at c^o.

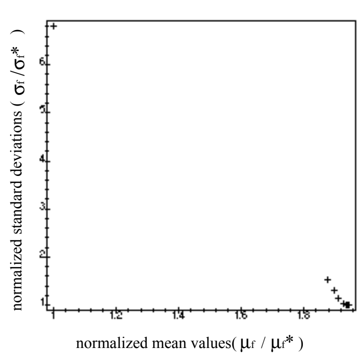
Non-convex Attainable Sets

- The weighted sum objective can only locate those efficient solutions that sit on the convex portion of the efficient frontier.
- The Compromise Programming approach can locate the complete set of efficient solutions.

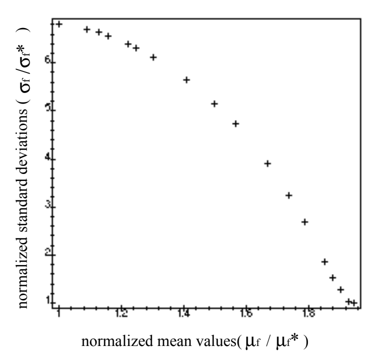
Example 1- A Mathematical Problem

Min $F(x) = (x1-4.0)^3+(x1-3.0)^4 + (x2-5.0)^2 + 10$

s.t. $g(x) = -x1 - x2 + 6.45 \leq 0,\ 1 \leq x1 \leq 10,\ 1 \leq x2 \leq 10$



Efficient solutions cannot be located in the nonconvex portion using the WS method



Complete set of efficient solutions using the CP