Short Course on

Evolutionary Computing Methodologies for Single and Multi-Objective Optimization

Kalyanmoy Deb
Deva Raj Chair Professor
Indian Institute of Technology Kanpur
PIN 208016, India
Email: deb@iitk.ac.in
URL: http://www.iitk.ac.in/kangal/deb.htm
Also Helsinki School of Economics, Finland
(Kalyanmoy.Deb@hse.fi)

Aim of the Course

- Many problem-solving activities in practice involve an optimization task
- Applied optimization tasks are of different types
  - Non-linearity, multiple objectives, hard constraints, uncertainties, multi-level optimization, decision making, expensive computations
- Require a robust and flexible optimization procedure
- Evolutionary optimization is promising and is the main focus of this course

Course Structure

- Monday, 17/08/09:
  - 08:00-09:45 (L1), 10:00-12:00 (L2)
- Tuesday, 18/08/09:
  - 08:00-09:30 (L3), 09:45-11:15 (L4), 11:15-12:00 (Lab)
- Wednesday, 19/08/09:
  - 08:00-09:45 (L5), 10:00-12:00 (L6)
- Thursday, 20/08/09:
  - 08:00-09:30 (L7), 09:45-11:15 (L8), 11:15-12:00 (Lab)
- Friday, 21/08/09:
  - 08:00-09:45 (L9), 10:00-12:00 (L10)

Course Structure (cont.)

- 17 August, 2009:
  - Optimization and its scope in practice
  - Classical optimization methods and practical difficulties
  - Introduction to *evolutionary optimization algorithms (EAs)*
    - Binary-coded genetic algorithm (GA) and case studies
- 18 August, 2009:
  - Constraint Handling, theoretical underpinnings
  - Real-parameter EAs
    - SBX, PCX, DE, PSO, ES, etc.

Course Structure (cont.)

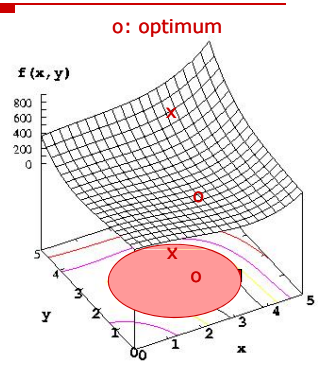
- 19 August, 2009:
  - EAs for more complex optimization tasks
    - Multi-modal EAs, meta-model assisted EAs, mixed integer EAs
  - Evol. Multi-objective optimization (EMO) – Introduction
  - Constraint handling in EMO, applications of EMO
- 20 August, 2009:
  - Advanced EMO Techniques
    - Partial frontier, many objectives, uncertainty handling etc.
  - Some fundamental issues in EMO
- 21 August, 2009:
  - Decision making in EMO
  - Bilevel optimization, dynamic optimization, omni-optimizer
  - Conclusions

Lecture 1:
Optimization and Its Need in Practice

- What is optimization?
- Scope of optimization in practice
- Steps in an optimization task
  - Objectives, constraints and variables
- Fundamental optimality conditions

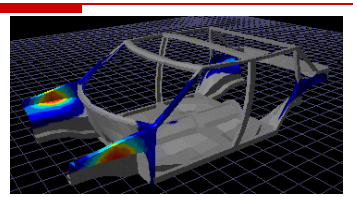
What is Optimization?

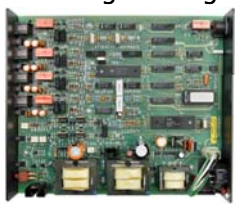
- A task of searching for a set of *decision variables* which would minimize or maximize one or more *objective* functions subject to satisfying *constraints*
- Decision variables: *(x,y)*
- Objective function: *f(x,y)*
- Constraint function: Equality or inequality type



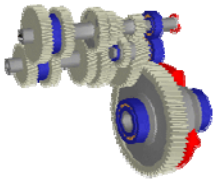
Scope of Optimization in Practice

- *Optimal design & manufacturing* for desired goals
- Major application in engineering



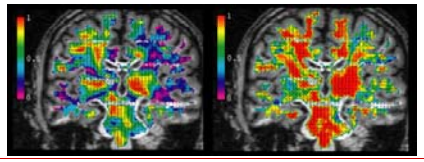






Scope of Optimization (cont.)

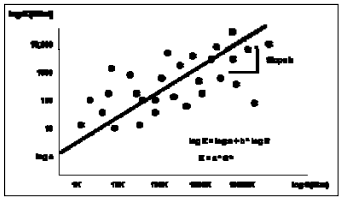
- *Inverse Problems*
  - Output known, find input
  - Often with a goal: minimize distortion, maintain physics, occum's razor (simplest) etc.
  - Tomography, reconstruction, 3D from 2D images
  - Lead to multiple solutions



Reverse current in brain models (Johnson, 2006)

Scope of Optimization (cont.)

- *Parameter optimization* for optimal performance

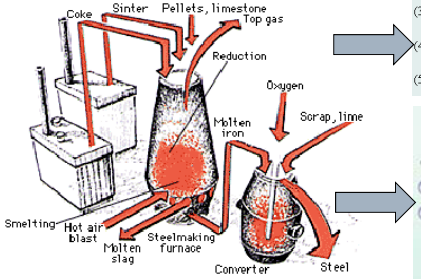


- Scientific experiments, computer experiments



Scope of Optimization (cont.)

- *Modeling* (system, process)



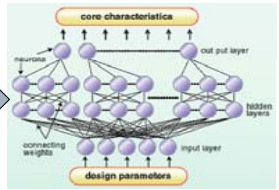
$$(1)\ \frac{\partial P}{\partial t} = -\gamma P + \beta \lambda I_A - \kappa P S_A$$

$$(2)\ \frac{\partial I_A}{\partial t} = r I_A - g_A I_A + \kappa P S_A - \lambda I_A + m b_{B\to A} I_B - m b_{A\to B} I_A$$

$$(3)\ \frac{\partial S_A}{\partial t} = r S_A - g_A S_A - \kappa P S_A + m b_{B\to A} S_B - m b_{A\to B} S_A$$

$$(4)\ \frac{\partial I_B}{\partial t} = -g_B I_B - m b_{B\to A} I_B + m b_{A\to B} I_A$$

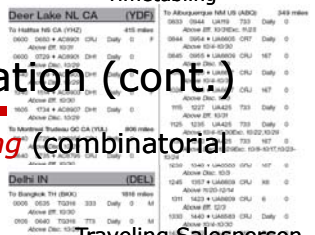
$$(5)\ \frac{\partial S_B}{\partial t} = -g_B S_B - m b_{B\to A} S_B + m b_{A\to B} S_A$$



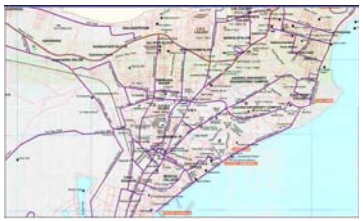
Scope of Optimization (cont.)

- *Scheduling and planning* (combinatorial optimization)

Timetabling



Routing & Scheduling



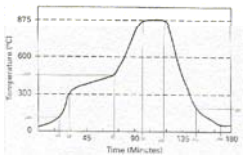
Traveling Salesperson problem

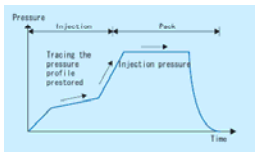


Scope of Optimization (cont.)

- *Optimal control*
- Time-variant profiles are to be found
  - How to lower load?
  - How to control temp, pressure?









Scope of Optimization (cont.)

- *Forecasting and prediction*





Scope of Optimization (cont.)

- *Data mining* (classification, clustering, pattern recognition)





Scope of Optimization (cont.)

- *Machine learning*
  - Designing intelligent systems

Developed at KanGAL

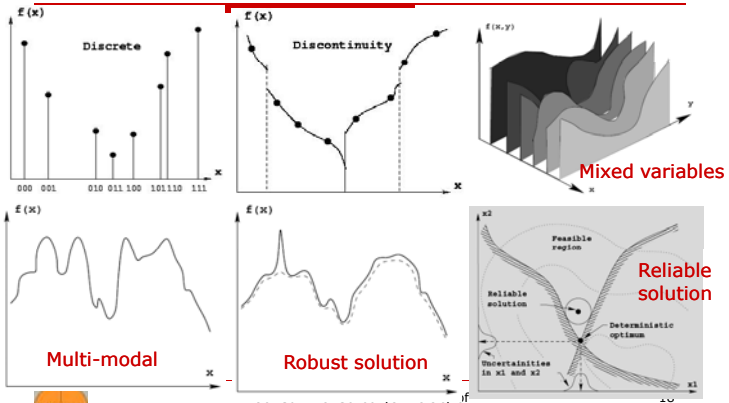




Properties of Practical Optimization Problems

- Non-differentiable functions and constraints
- Discontinuous search space
- Discrete search space
- Mixed variables (discrete, continuous, permutation)
- Large dimension (variables, constraints, objectives)
- Non-linear constraints
- Multi-modalities, multiple levels
- Multi-objectivity
- Uncertainties in variables
- Computationally expensive problems
- Multi-disciplinary optimization

Different Problem Complexities: A Pictorial View

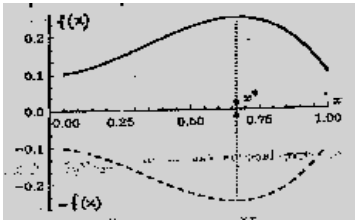


Design Variables, Constraints, and Objectives

- List any and every parameter related to the problem
- Specify the type of each parameter (binary, discrete, continuous)
- Choose a few of them as design variables
- First thumb rule: Use as few variables as possible
  - Usually from the experience of the user
  - From minimum variability consideration
  - From sensitivity analysis
- Constraints are of two types:
- - Inequality: $g(x) \geq 0$ or $g(x) \leq 0$
- - Equality: $h(x) = 0$

Further Comments

- Second thumb rule: Avoid complex equality constraints, use simple equality constraints to eliminate a variable
- $\delta(x) = 5$ can, be relaxed as $\delta \geq 4$ and $\delta \leq 6$
- Objective function: Minimize or maximize
- Duality principle helps unification
- Variable bounds: $x^L \leq x \leq x^U$



Steps in An Optimization Task

- Need for optimization
- Problem formulation
  - Identify problem parameters
  - Choose design variables from parameters
  - Formulate Constraints
  - Formulate objective function
  - Set up variable bounds
  - Requires 50-60% of the effort
- Choose an optimization algorithm
- Obtain solution(s)
- Reformulation and rerun, if desired

Optimization Problem

- Decision variables: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
- Constraints restrict some solutions to be feasible

$$\begin{aligned} &\text{Min. } f(\boldsymbol{x}) \\ &\text{s.t. } g_j(\boldsymbol{x}) \geq 0 \quad j = 1,2,\ldots,J \\ &\quad\; h_k(\boldsymbol{x}) = 0 \quad k = 1,2,\ldots,K \\ &\quad\; x_i^L \leq x_i \leq x_i^U \quad i = 1,2,\ldots,n \end{aligned}$$

- Equality and inequality constraints
- Minimum of $f(\boldsymbol{x})$ need not be constrained minimum
- Constraints can be non-linear

Fundamental Optimality Conditions for Unconstrained Minimum Points

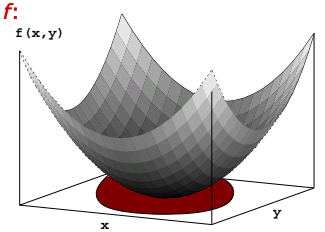
- Single variable: $df/dx = 0$, $d^2f/dx^2 > 0$
  - General rule: Identify first non-zero derivative at point x, say its order is n. If n is odd, the sign of that derivative (+ve or –ve) determines minimum or maximum, respectively. If n is even, it is a saddle point.
- Multiple variables: *grad f = 0*, Hessian matrix $H(f)$ is +ve definite for minimum
  - A matrix is +ve definite, if all eigenvalues are positive

Constrained Optimization

- Karush-Kuhn-Tucker (KKT) conditions for optimality
  - First-order necessary conditions
  - Convex search space, convex f: KKT point is minimum

$$\nabla f(x) - \sum_{j=1}^{J} u_j \nabla g_j(x) - \sum_{k=1}^{K} v_k \nabla h_k(x) = 0$$

$$\begin{aligned} &g_j(\boldsymbol{x}) \geq 0 \quad && j = 1,2,\ldots,J \\ &h_k(\boldsymbol{x}) = 0 \quad && k = 1,2,\ldots,K \\ &u_j g_j(\boldsymbol{x}) = 0 \quad && j = 1,2,\ldots,J \\ &u_j \geq 0 \quad && j = 1,2,\ldots,J \end{aligned}$$



- Involve derivatives and solve for roots
- Difficult for classical methods

No Free Lunch (NFL) Theorem

- In the context of *optimization*
  - Wolpert and McCardy (IEEE TEC, 1997)
  - Algorithms A1 and A2
  - All possible problems **F**
  - Performances P1 and P2 using A1 and A2 for a fixed number of evaluations
  - P1 = P2
- NFL breaks down for a class of problems or algorithms
- Research effort: Find the best algorithm for a class of problems
  - Unimodal, multi-modal, quadratic etc.



Summary

- Optimization is widespread in practice and an essential component in many practical problems
  - Better solution (design)
  - Better control
  - Search for innovative solutions
  - Understand systems better
- Theoretically sound, but often theory not easy to use in practice
- NFL theorem requires one to know alternative methods
  - *Strong need for a course like this one!*



An Introduction to Classical Optimization

- Single variable numerical optimization
  - Direct search methods (Bounding phase and Golden section search)
  - Gradient search methods (Newton's search)
  - Point estimation method
- Multi-variable numerical optimization
  - Direct search methods (simplex method, Hooke-Jeeves Pattern search)
  - Gradient search methods (Steepest descent, Fletcher-Reeves and DFP method)



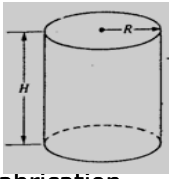
Constrained Optimization by Classical Means

- Constrained optimization
  - Penalty function approach
  - Linearized methods using LP solvers
  - Specific equality constraint handling methods
- Structured optimization
  - Geometric Programming
  - Dynamic programming
  - Integer programming



An Example

- Design of a Beer Can
- Minimize cost of manufacturing (directly related to surface area)
- Can must hold at least a specified amount (say, 330 ml)
- Handling, shipping, aesthetic, and fabrication limitation
- Diameter of can not be more than 8 cm and less than 3.5 cm
- Height of can not be more than 18 cm and less than 8 cm

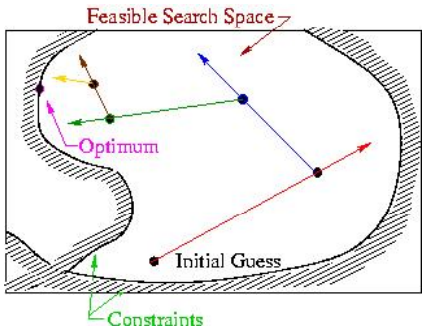


Beer Can Example (cont.)

- Problem parameters: Can thickness, can diameter, can height, can shape (circular or square), can material, fabrication process, refrigerated temperature, reuse, etc.
- Choose variables: diameter (D) and height (H)
- Circular from an available AL sheet
- Constraint: $g(D,H)$: $\pi D^2H/4 \geq 330$ (ml)
- Objective function: $f(D,H) = \pi DH + \pi D^2/2$ (cm^2)
- Variable bound: $3.5 \leq D \leq 8$, $8 \leq H \leq 18$ (cm)
- Optimization problem has 2 variables and 1 inequality constraint, and variable bounds

Classical Optimization Methods

- Point-by-point method
- Direct methods
  - No use of any gradient or other information
- Gradient-based methods
  - Explicit use of first and second-order gradients



Uni-Directional Search

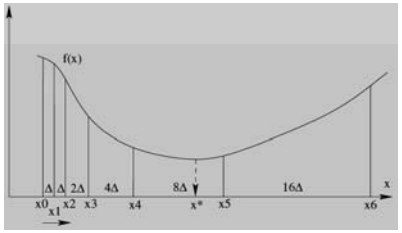
- A multi-variable problem, but the search is restricted along a direction
  - Single-variable search
  - $\mathbf{x}(a) = \mathbf{x}^{(t)} + a^*\mathbf{s}^{(t)}$, a is the only variable
- Repeatedly used in multi-variable optimization
- Need a quick single-variable search method

Single-Variable Optimization

- Only one variable
- Two-step method for direct search
  - Rough bounding: Bounding phase method
  - Accurate bounded search: Golden section search
- Assumption: Existence of only one optimum

Bounding Phase Method:
Coarse Bounding of Optimum

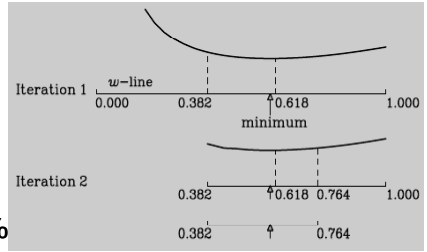
- Start with $x^{(0)}$ and Δ
- Determine sign of Δ, k=1
- $x^{(k)} = x^{(k-1)}+2^k \Delta$
- Compare $f(x^{(k)})$ and $f(x^{(k-1)})$
- Continue till no more improvement
- Exponentially approach minimum



Bound: $(x^{(k-2)}, x^{(k)})$

Golden Section Search:
Finding Minimum Precisely

- Search within bounds obtained by rough bounding
- Golden number (0.618)
- Every iteration eliminates 38.2% of search space
- Most efficient on a serial processor



What strategy is best for a parallel computer?

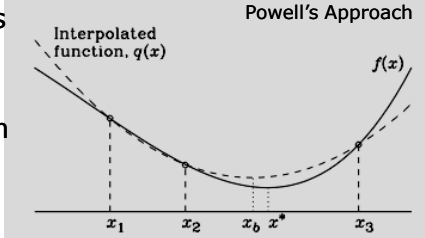
Gradient Based Method:
Single-variable search

- Each gradient takes two evaluations
- Central difference method:

  $df/dx \approx (f(x+\delta x)- f(x-\delta x))/(2*\delta x)$
- $\delta x = 0.01|x|$, if $|x|<0.01$, else 0.0001
- $(\delta x)^2$ accuracy
- Newton's search requires the second derivative -> 3 function evaluations
  - Better accuracy with more evaluations

Point Estimation Method

- Take three points and compute an approximate model
- Find the optimum of the model
- Develop an algorithm
- A primitive version of meta-modeling technique



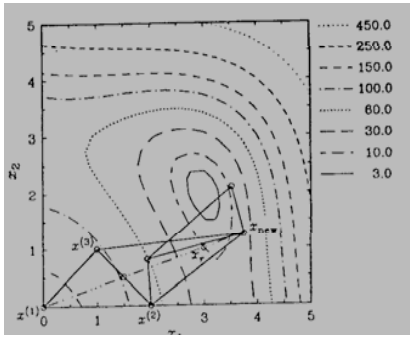
Extension to multiple variables computationally expensive!

Multi-Variable Optimization

- Direct search methods do not use any gradients
  - Nelder and Meade's simplex search method
  - Hooke-Jeeves Pattern search methods
  - Powell's conjugate direction method
- Gradient based methods
  - Steepest descent, Newton's, Quasi-Newton's methods

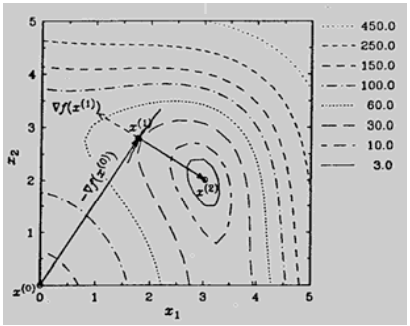
Simplex Search Method

- Simplex search method
- Conjugate direction method
- Randomized methods
- Hill-climbing methods
- Trust region methods



Gradient-Based Methods

- Steepest-descent method
  - Reduces the function value maximally
  - For convex problems, probably best
- Conjugate gradient method
- Quasi-Newton method
  - BFGS, DFP



Gradient-Based Methods (cont.)

- Gradient computation is expensive
- Gradient vector takes *2n* soln. evals.
- Hessian matrix takes $2n^2 + 1$ solution evaluations
- Quasi-Newton methods are good compromise for time and accuracy
  - Attempt to simulate Hessian matrix at $\boldsymbol{x}^*$ using gradient vectors at successive iterations
  - DFP, BFGS adopted by MATLAB etc.

Constrained Optimization

- Decision variables: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
- Constraints restrict some solutions to be feasible

$$\begin{aligned} &\text{Min. } f(\boldsymbol{x}) \\ &\text{s.t. } g_j(\boldsymbol{x}) \geq 0 \quad j = 1,2,\ldots,J \\ &\quad\; h_k(\boldsymbol{x}) = 0 \quad k = 1,2,\ldots,K \\ &\quad\; x_i^L \leq x_i \leq x_i^U \quad i = 1,2,\ldots,n \end{aligned}$$

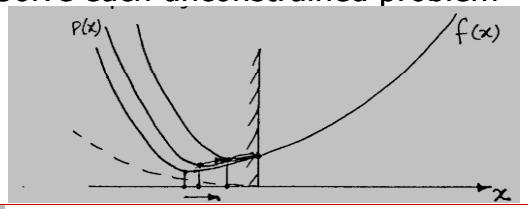
- Equality and inequality constraints
- Minimum of $f(\boldsymbol{x})$ need not be constrained minimum
- Constraints can be non-linear

Constraint Handling Algorithms

- Penalty function method

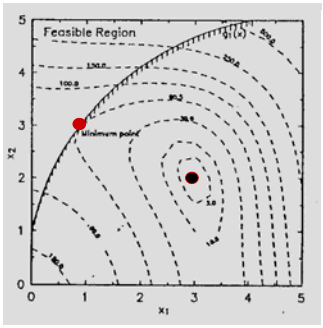
$$P(x, R) = f(x) + \Omega(R, g(x), h(x))$$

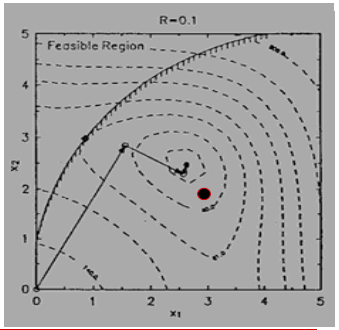
- Start with a small R, then increase
- Solve each unconstrained problem



An Example Problem

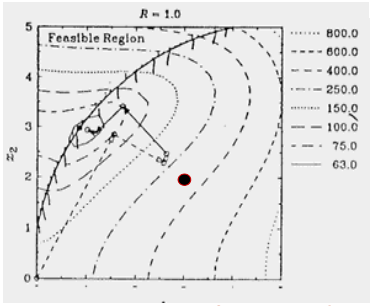
R=0 R=0.1



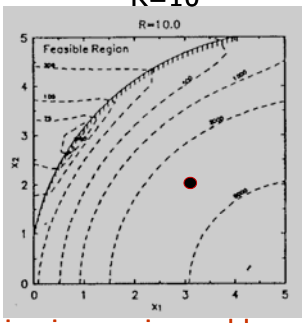


Penalty Function Approach (cont.)

R=1



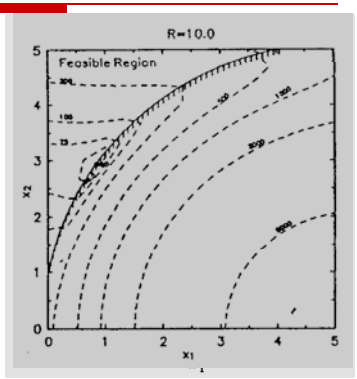
R=10



Distortion in objective function is a major problem

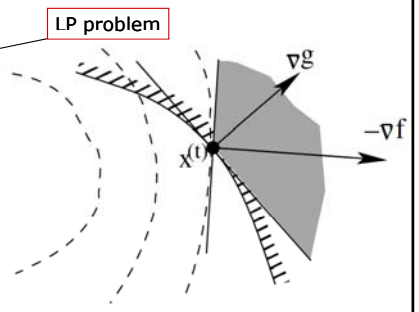
Augmented Lagrangian Method

- Penalty update is different
  - Expressions are complex
- Not much distortion in the penalized function
- In addition, the resulting solution is a KKT point
- Lagrange multiplier are by-products



Other Specialized Algorithms

- Feasible Direction Method
  - Direction both feasible and descent
- Cutting plane method
- Generalized reduced gradient method
  - Move along constraint boundaries



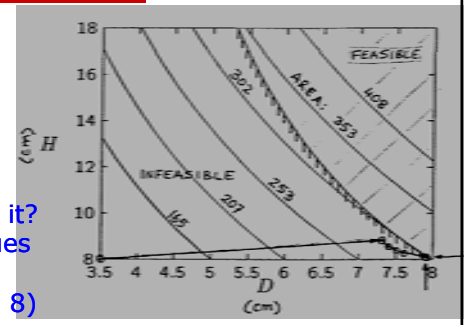
Beer Can Example

- Problem parameters: Can thickness, can diameter, can height, can shape (circular or square), can material, fabrication process, refrigerated temperature, reuse, etc.
- Choose variables: diameter (D) and height (H)
- Circular from an available AL sheet
- Constraint: $g(D,H)$: $\pi D^2H/4 \geq 400$
- Objective function: $f(D,H) = \pi DH + 2\pi D^2/4$
- Variable bound: $3.5 \leq D \leq 8$, $8 \leq H \leq 18$
- Optimization problem has 2 variables and 1 inequality constraint, and variable bounds

Beer Can Optimization

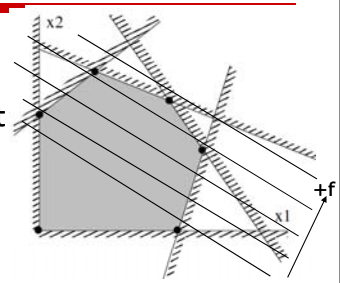
- Final solution is D = 7.956 and H = 8.048 cm,
- Volume 400 ml
- Surface area 300.6 cm^2
- Can we fabricate it?
- Use discrete values

D: (3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8)
H: in steps of 0.5 cm



Structured Algorithms: *Linear Programming*

- Optimization problems having linear objective and constraint functions
- Optimum is a corner point or multiple points along an edge
  - Search restricted among corner points (Simplex Search)
  - Search inside feasible region (Interior point method)
    - Karmarkar's algorithm



For small sized problems, use 'simplex search'

Integer Programming Methods

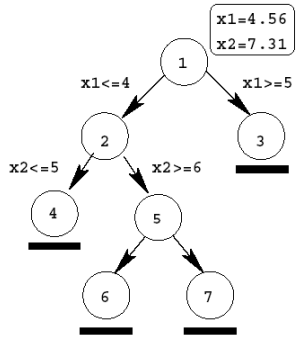
- Many design variables take integer values only
- Example: Number of teeth in a gear, pipe diameter may come in discrete dimensions, etc.
  - Remember beer can problem
- Search space becomes discrete and discontinuous
- Gradient methods can not be applied directly (Example: Material)
- Direct search methods also fail
  - Points dictated by the algorithm may not be available

Integer Programming (cont.)

- One simple-minded strategy:
  - Assume design variables can take any real value
  - Round-off to nearest integer solutions
- Example: Two integer variables (23.76, 4.12)
- Compare (23, 4), (23, 5), (24, 4), and (24, 5)
- For 10 integer variables compare 210 or 1,024 designs
- Even then true optimum is guaranteed

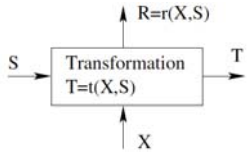
Branch-and-Bound Method

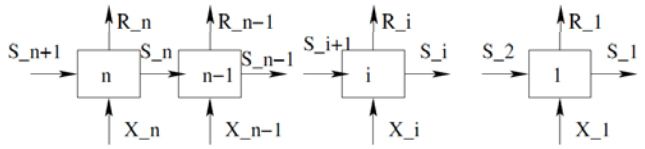
- Extremely popular
- Treat problem without integer restrictions
- Branch into two problems with an additional constraint each
- Bound:
  - Infeasible solution
  - Integer solution
  - Best function value is worse than an integer solution



Dynamic Programming

- Applicable to a multi-stage decision process
- Optimize each stage with a parameter S
- Min Σ R_i(S_i+1,X_i)
- Constraints: S_i=t_i(X_i,S_i+1)





Initial Value Problem

Backward pass

- S_n+1 is given, find all X_i
- Optimization of subsystems
  - Start from subsystem 1 (parametric optimization with S_2)
    - Know x_1*, R_1* (=f_1*), S_1* as a function of S_2
  - Move to subsystem 2 (parametric with S_3): Min R_2(X_2,S_3)+f_1*(s_2)
    - Know X_2*,R_2*,S_2* as a function of S_3
  - Carry on till the end (i=n)
- Here, S_n+1 is given, make a forward pass to determine all S_i*, X_i* and R_i*

Duality Theory in Optimization

- By using constraints, a primal problem has an equivalent dual problem
- Dual problem is always concave
  - Comparatively easier to solve
- Theoretical results:
  - Convex problems and in some special cases:
    - Optimal primal and dual function values are the same
  - Generic cases:
    - Optimal dual function value underestimates optimal primal function value
- Beneficial to use duality theory in many problems

Summary

- Foundation based on mathematics
  - Must-know for everyone interested in optimization
- Structured algorithms
  - If applicable, no competitor, use them
- Unfortunately, practice is full of exceptions
- One needs to be open-minded, but first application must be a classical one
- Many theories of optimization can be utilized in designing efficient methods

