Optimization Fundamentals 1

Presented by:

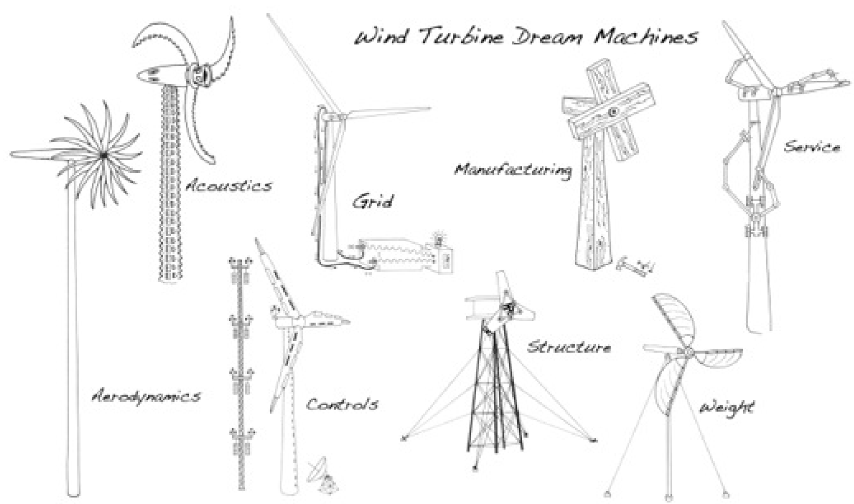
Katherine Dykes, Senior Engineer

National Renewable Energy Laboratory

MCEN4228-5228, Optimization with Application to Wind Plant Design

CU Boulder, CO, USA

1/19/2016



Overview

- Why Optimization?
- Basic Elements of Optimization
- A Simple Linear Optimization Problem
- A Simple Unconstrained Non-linear Optimization Problem

Why Optimization?

Why not optimization?

- Optimization is hard – complex systems, lots of uncertainty, lots of trade-offs, lots of sensitivity to the problem formulations and models
- Practical design is siloed anyway – the aerodynamicist designs the blade with limited input from the structural engineers and rest of the component designers; manufacturing is done often by a separate organization who will tweak a “finished” design from the OEM
- Current design approaches are “good enough” – been designing things this way for decades and it works just fine

Why Optimization?

- Yes, optimization is hard, but…
- Designing a system requires many different trade-offs…
  - The cost of a wind turbine and its performance or reliability
  - The power production from a wind plant and the infrastructure costs
- And optimization allows the designer to take into account all these complex system trade-offs (across disciplines, organizations) to find the “best” design that at the same time meets all the system design requirements
- Non-intuitive “better” designs can often be identified by optimization but not by traditional design processes based on heuristics and experience

Basic Elements of Optimization

Basic Elements of Optimization Overview

- Optimization structure includes:
  - Objective Function – what do we care about the most?
  - Design variables – what choices do we have about our system design? What can we manipulate?
  - Parameters – what can’t we change about our system? What is fixed in our design a priori?
  - Constraints – what other system requirements do we have that must be met?

Basic Elements of Optimization Overview

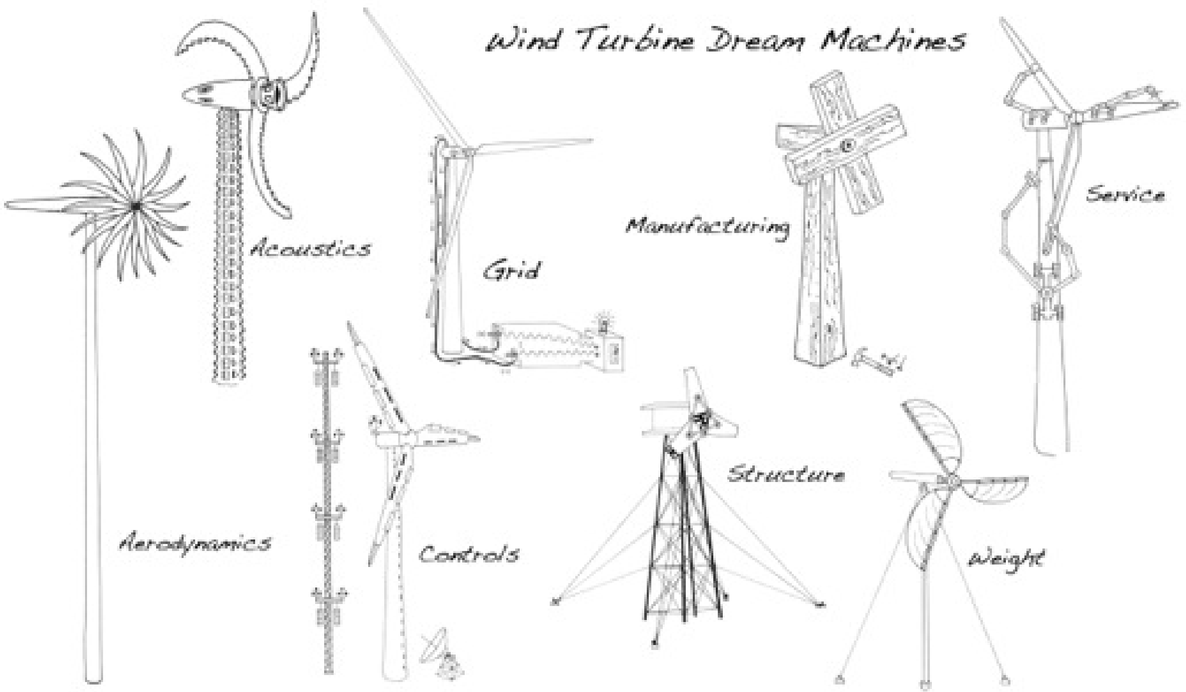
- Generally expressed mathematically:

$$\begin{aligned}&minimize\ O(p,v)\\&with\ respect\ to\ v\\&subject\ to\ \ C_1(p,v)=0, C_2(p,v)\le 0\end{aligned}$$

  - Where O is the objective function
  - C_1 and C_2 are equality and inequality constraints respectively
  - p are parameters and v are design variables

Objective Function

- Objective functions define what it means to be the best design
- The choice of objective function critically affects the outcome; the wrong objective or an objective defined too narrowly can result in bad system designs



Objective Function

- What if I have multiple system objectives?
  - Example: generator design for a wind turbine
    - Mass
    - Cost
    - Efficiency (performance)
    - Air-gap radius (size)
  - Example: layout design for a wind turbine
    - Gross energy
    - Loss minimization
    - Cost of infrastructure

Objective Function

- Approaches to multiple objectives:

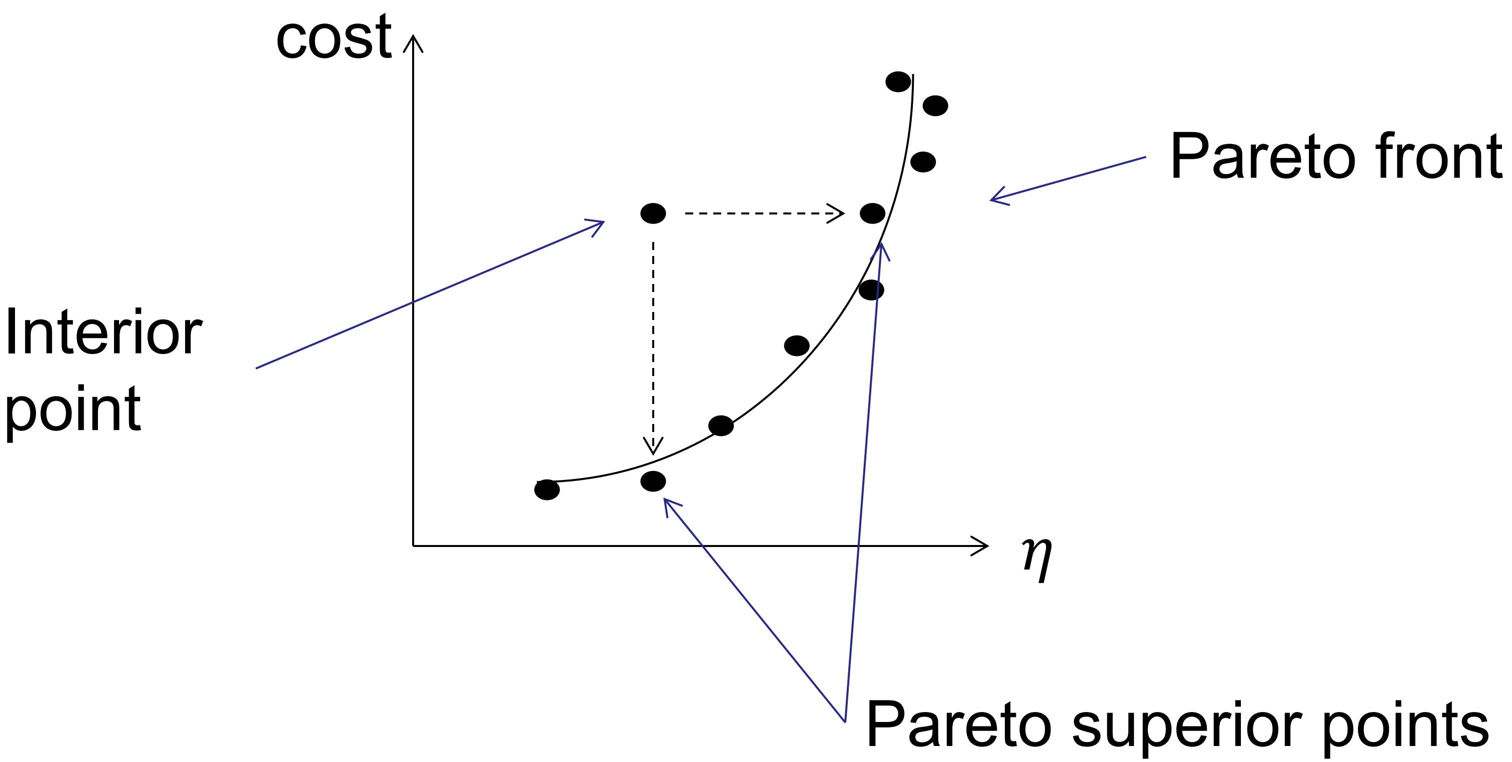
1. Select global objective and turn other objectives into constraints
   - Generator cost as global subject to maximum allowable mass, air gap radius and minimum allowable efficiency

2. Role objectives up into higher level objective:
   - Weighted combination of objectives, or
   - Higher level objective: $COE = \frac{F * CAPEX + OPEX}{AEP}$

Objective Function

- Approaches to multiple objectives

3. Take a multi-objective approach (explore the trade-space)

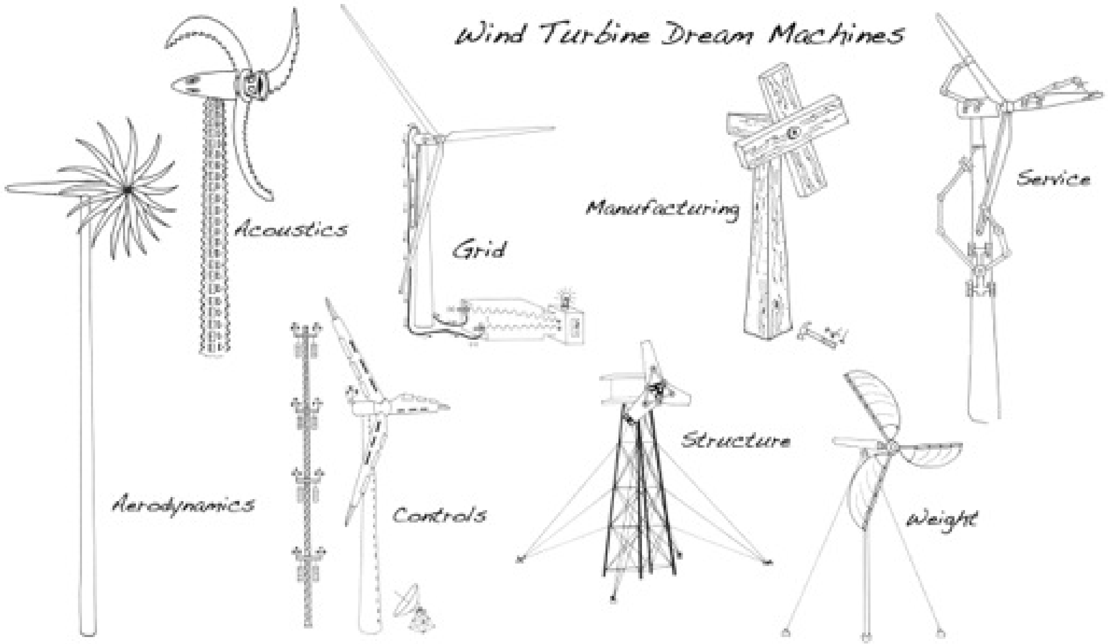


Objective Function

- Complexity of objective functions can vary drastically
  - Single Equation: $f(x) = 3x^3 + 6x^2 - 9$
  - Single Discipline Model:
    - Aerodynamic design of a wind turbine blade using blade-element momentum theory
    - Structural design of a wind turbine blade using a finite-element code
    - Layout of a wind plant for energy production with pre-specified turbine cp, ct curves

Objective Function

- Most real-world design problems involve at least 2 up to many disciplines (multi-disciplinary optimization):
  - Turbine design:



  - Overall wind plant system design:

$$COE = \frac{F * CAPEX + OPEX}{AEP}$$

Design Variables

- Design / decision variables
  - Key elements of system design that are allowed to change
  - Must be independent of each other (else decomposition is necessary)
    - What a design variable can be depends a lot on model fidelity – i.e. rotor diameter may be a D.V. for a simple model but could be an intermediate/derived variable for a higher fidelity model where there blade length and hub radius are variables
  - Design variables may be continuous (turbine location in a plant) or discrete/integer variables (type of turbines in a plant)
    - Types of variables will determine type of optimization problem

Constraints

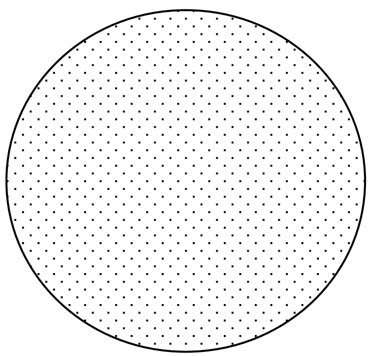
- Some optimizations may be *unconstrained* (without constraints) but most are *constrained*
  - Presence of constraints determines optimization type
- There are both equality and inequality constraints – usually try to use inequality constraints if possible
- Simplest constraints are the *bounds* (allowable ranges) for the design variables:

$$lb \leq v \leq ub$$

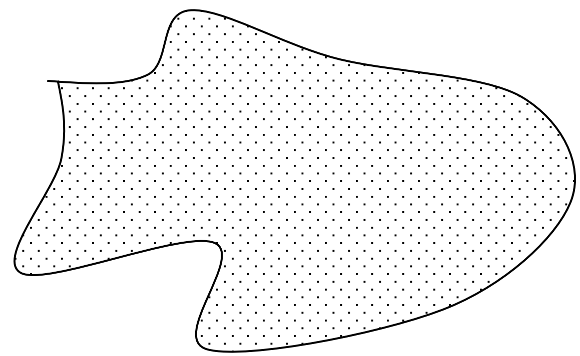
  - Example: all turbine locations must be within the boundary of the wind farm
- Other system constraints can be described as functions
  - Example: the minimum distance between 2 turbines must be X

Constraints

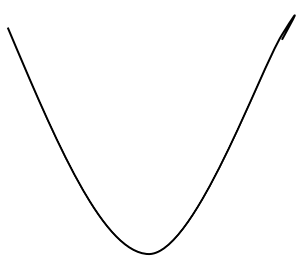
- Constraints define the “feasible” design space – the set of solutions in terms of combinations of design variables that are allowed for the system design
  - The objective function selects the optimum out of this space



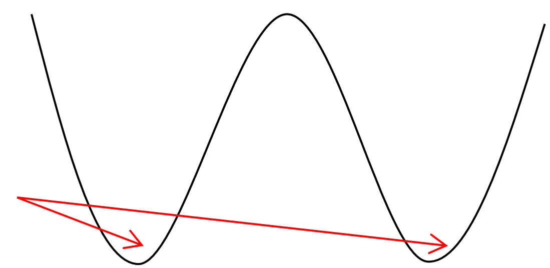
Convex design space



Nonconvex design space



Local minima



Types of Optimization

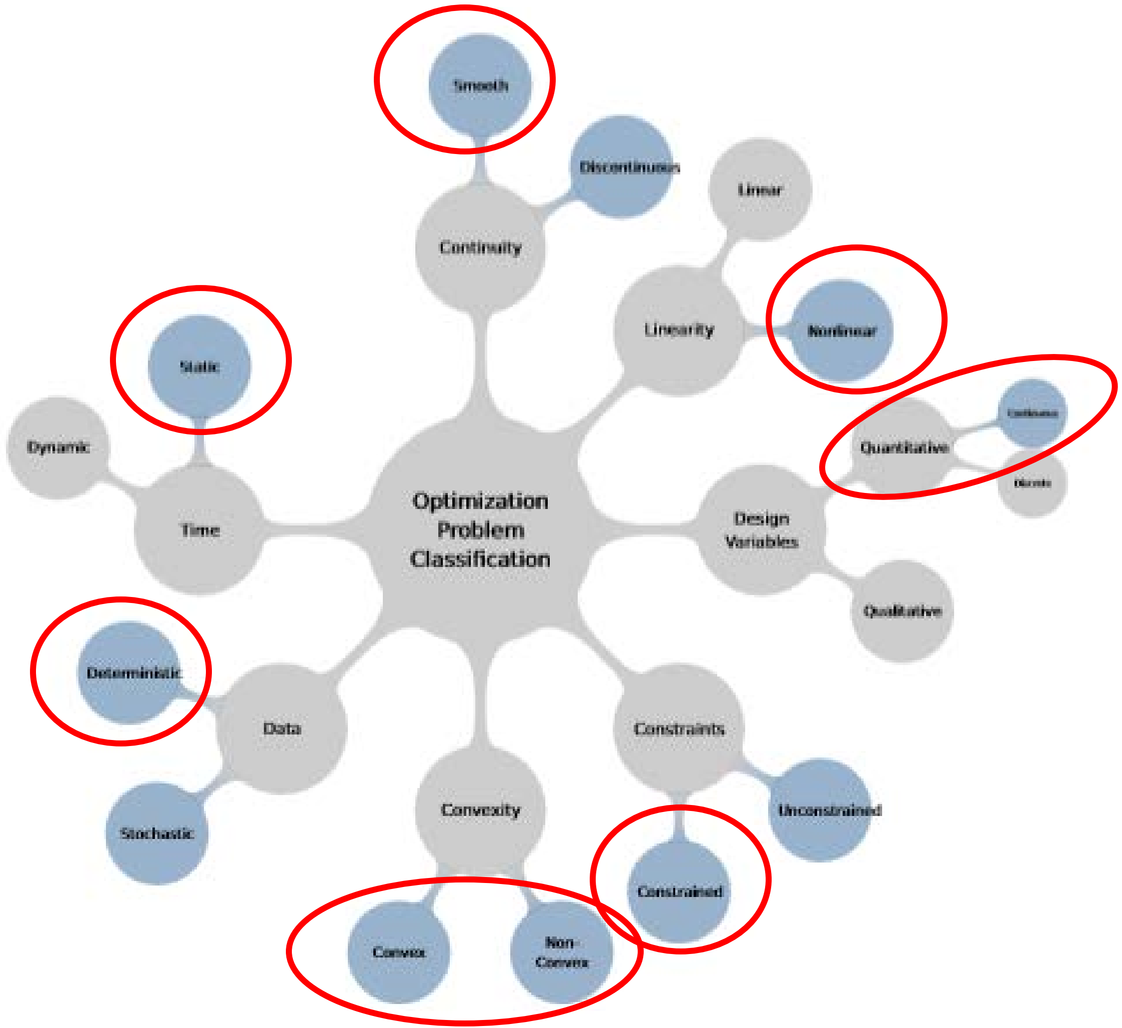
- Objective functions and constraints can both be linear or non-linear

- These attributes (linear/non-linear, constrained/unconstrained, continuous/integer/discrete) determine the type of optimization

- There are additional classes of methods for optimizations that involve:
  - Uncertainty in design variables and/or models (deterministic vs. static)
  - A progression of decisions over time versus a single point of decision (static vs. dynamic)

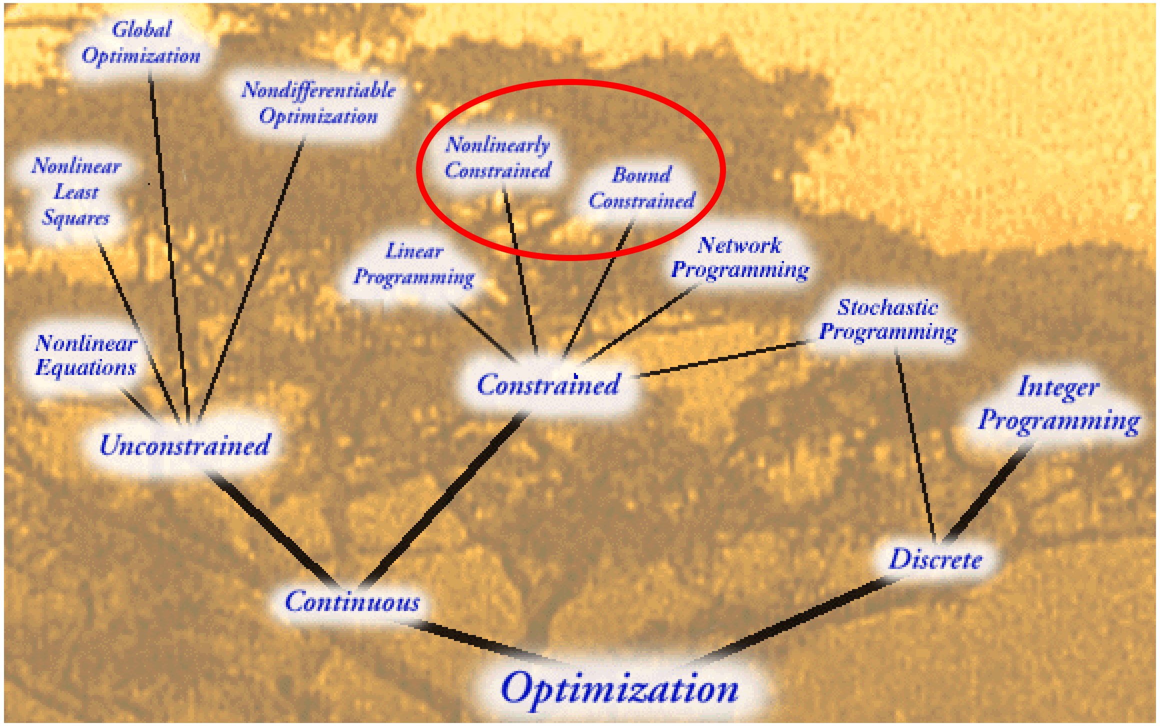
Types of Optimization

- One classification of types of optimization (Dr. Andrew Ning, BYU)…



Types of Optimization

- … and another along with associated methods (Dr. Cameron Thraen, OSU)



Optimization Methods

- Optimization type influences optimization method:
  - Convex optimization using gradients – easiest to solve, will be able to “prove” global optimum is found (we will start here!)
  - Non-convex optimization (discrete variables, bumpy landscape/design space):
    - Use gradient techniques with multiple starts (multiple initial conditions for design variables), or
    - Use gradient-free methods, or
    - Mix gradient-based and gradient-free methods

A Simple Linear Optimization Problem

A Simple Linear Optimization Problem

- An example from economics / business operations: maximize the revenue for a business that sells three different products (each requiring different amounts of resources and time):

Resource	Desk	Table	Chair	Total Resource available
Lumber	8 ft.	6 ft.	1 ft.	48 ft
Finishing time	4 hrs	2 hrs	1.5 hrs	20 hrs
Carpentry time	2 hrs	1.5 hrs	0.5 hrs	9 hrs

- Each selling for a different price: desk for $60, a table for $30, and a chair for $20
- And only 5 tables can be sold at most in a time period

Problem Set Up

- What are our design variables?
- What is our objective function?
- What are our constraints?
- What are the parameters?
- Now put the model into the form:

$$
\begin{aligned}
&\mathit{maximize}\ O(p, v)\\
&\mathit{with\ respect\ to}\ v\\
&\mathit{subject\ to}\ \ C_1(p, v) \leq 0
\end{aligned}
$$

Problem Set Up

- The resulting form of the optimization problem should look like this:

$$
\begin{aligned}
&Max\ revenue = 60v_1 + 30v_2 + 20v_3 \\
&s.t. \\
&\qquad 8v_1 + 6v_2 + v_3 \leq 48\ (lumber) \\
&\qquad 4v_1 + 2v_2 + 1.5v_3 \leq 20\ (finishing) \\
&\qquad 2v_1 + 1.5v_2 + 0.5v_3 \leq 9\ (carpentry) \\
&\qquad v_2 \leq 5\ (table\ demand) \\
&\qquad v_1, v_2, v_3 \geq 0\ (bounds)
\end{aligned}
$$

- What does this look like?

Solving the Problem

- A linear optimization problem can be solved by linear programming
- The *Simplex Algorithm* is a popular approach for linear programming that essentially solves a system of equations using *elementary row operations*
  - We introduce a “slack” variable (new d.v.) for each constraint and set it equal to the constraint value
  - We set an initial solution where the design variables are all zero
  - This is our “basic feasible solution” – it is feasible but not optimal (i.e. we can choose to produce nothing)

Solving the Problem - Simplex

- With slack variables the set-up becomes:

$$
\begin{aligned}
&Max\ revenue = 60v_1 + 30v_2 + 20v_3 \\
&s.t. \\
&\qquad 8v_1 + 6v_2 + v_3 + s_1 \leq 48\ (lumber) \\
&\qquad 4v_1 + 2v_2 + 1.5v_3 + s_2 \leq 20\ (finishing) \\
&\qquad 2v_1 + 1.5v_2 + 0.5v_3 + s_3 \leq 9\ (carpentry) \\
&\qquad v_2 + s_4 \leq 5\ (table\ demand) \\
&\qquad v_1, v_2, v_3 \geq 0\ (bounds)
\end{aligned}
$$

- There are 4 slack variables for each of the four constraints

Solving the Problem - Simplex

- Step 1: Is our basic feasible solution optimal?

$$revenue = 60v_1 + 30v_2 + 20v_3$$

- No, if increasing a non-basic variable (the design variables) increases the objective function then the current solution is not optimal

- v_1 has the largest impact on profit, therefore look at increasing v_1

Solving the Problem - Simplex

- Step 2: Increase a non-basic variable as much as possible without violating any constraints

- Use "ratio test", given v_2, v_3 are 0, solve for the maximum value of v_1 to use up each constraint:

$$\begin{aligned} 8v_1 + s_1 &\leq 48 \\ 4v_1 + s_2 &\leq 20 \\ 2v_1 + s_3 &\leq 9 \\ s_4 &\leq 5 \end{aligned}$$

- Maximum increase in v_1 is minimum from solving each of the above equations with the slacks set to 0 (in this case 4 from the last equation $v_1 = \frac{9}{2} = 4.5$ → 4

Solving the Problem - Simplex

- Step 3: “Pivot” to turn the entering variable into a basic variable (i.e. use it all up to the maximum level allowed by the “binding constraint”)
- Use elementary row operations on our problem to have a coefficient of 1 for v_1 in the constraint row 3 (this is the pivot row) and a coefficient of 0 in all other rows
- This turns v_1 into a basic variable and the slack variable in row 3 into a non-basic variable

Solving the Problem - Simplex

ERO 0	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		15	-5			30		240	
1				-1	1		-4		16	
2			-1	0.5		1	-2		4	
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 1	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		15	-5			30		240	
1				-1	1		-4		16	
2			-2	1		2	-4		8	multiplied row 2 by 2
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 2	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	added 5 times row 2 to row 0
1				-1	1		-4		16	
2			-2	1		2	-4		8	
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 3	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	
1			-2		1	2	-8		24	added row 2 to ro1
2			-2	1		2	-4		8	
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 4	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	
1			-2		1	2	-8		24	
2			-2	1		2	-4		8	
3		1	1.25			-0.5	1.5		2.5	add -1/4 times row 2 to row 3
4			1					1	5	

Solving the Problem - Simplex

- Repeat steps 1-3 until an optimal solution is found

- Step 1: is the current solution optimal?

$$revenue = 240 - 15v_2 + 5v_3 - 30s_3$$

- No – increase v_3 will increase revenue

- Step 2: Use ratio test on current state of constraint rows

$$\begin{aligned} -v_3 + s_1 &\le 16 \\ 0.5v_3 + s_2 &\le 4 \\ v_1 + 0.25v_3 &\le 4.5 \\ s_4 &\le 5 \end{aligned}$$

- Limited by row 2 where $v_3 = \frac{4}{0.5} = 8$, so step 3 pivot on row 2

Solving the Problem - Simplex

ERO 0	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		15	-5			30		240	
1				-1	1		-4		16	
2			-1	0.5		1	-2		4	
3		1	0.75	0.25			0.5		4	
4			1					1	5	

ERO 1	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		15	-5			30		240	
1				-1	1		-4		16	
2			-2	1		2	-4		8	multiplied row 2 by 2
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 2	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	added 5 times row 2 to row 0
1				-1	1		-4		16	
2			-2	1		2	-4		8	
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 3	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	
1			-2		1	2	-8		24	added row 2 to ro1
2			-2	1		2	-4		8	
3		1	0.75	0.25			0.5		4.5	
4			1					1	5	

ERO 4	profit	v1	v2	v3	s1	s2	s3	s4	rhs	ero description
0	1		5			10	10		280	
1			-2		1	2	-8		24	
2			-2	1		2	-4		8	
3		1	1.25			-0.5	1.5		2.5	add -1/4 times row 2 to row 3
4			1					1	5	

Solving the Problem - Simplex

- Check step 1, is the current solution optimal?

$$revenue = 280 - 5v_2 - 10s_2 - 10s_3$$

- Yes! Increase the value of v_2 will not increase the objective function any further; this is our *optimal basic feasible solution*

- Final solution:
  - Revenue = 280
  - $v_1 = 4, v_2 = 0, v_3 = 2$ or 4 desks, no tables and 2 chairs
  - $s_1 = 24, s_2 = 0, s_3 = 0, s_4 = 0$ or there are 24 units of lumber left over after the optimal solution is solved but no remaining time and there are 5 units of table demand left

Slack Variables & Constraints

- Slack variables tell us something about the solution, in our final solutions, s_2, s_3 are part of our non-basic variable set; these constraints are the “binding constraints”

- “Binding constraints” are those that bind the optimum so that you can not do any better
  - if the overall constraint in this case on finishing time or carpentry time where increased, we could increase our profit

- “Non-binding constraints” are those that don’t affect the overall optimum – you could increase them and it wouldn’t have the results
  - The constraints on lumber and table number did not affect the results; there are still some left over after the optimization is solved

Slack Variables & Constraints

- How much impact would relaxing a constraint have on the optimum?

- Return to the ratio tests:
  - the first binding constraint on v_1 was the carpentry constraint $2v_1 + s_3 \leq 8$
  - An increase of 1 on the carpentry time limit of 8 would allow for an extra 0.5 desks to be made which would increase profit by 0.5*60 or $30
  - An increase of 1 on the finishing time limit of 20 would result in an increase in the constraint for pivot 2 for the finishing constraint $0.5v_3 + s_2 \leq 5$. This would allow for 2 more chairs at an increased revenue of 2*20 or $40

- $30 and $40 are the "shadow prices" of the constraints – how much more could be made by a relaxation of the constraint by one unit
  - In other words, the "sensitivity" of the solution to those constraints

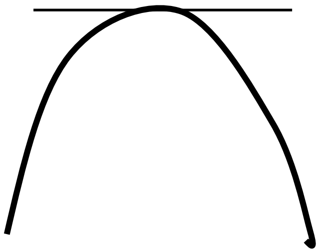
Simple Linear Optimization Problem

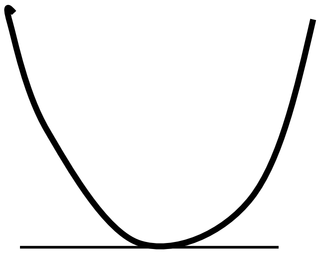
- Recap:
  - Linear optimization problems (where objective functions and constraints are linear) are generally the easiest class of optimization problems to solve
  - If all the variables are continuous as well, then using simple linear programming methods (i.e. the Simplex method) is possible
  - The problems still highlight various important aspects of all optimization problems:
    - Problem set-up according to standard structure
    - Feasible design/solution space
    - Iterative approach to improving objective function through sequential increase/decrease of design variables
    - Role of the constraint in “binding” / helping to determine the optimum solution
    - The role of constraint relaxation in terms of influencing optimum and optimum sensitivity to different binding constraints

A Simple Unconstrained Non-linear Optimization Problem

Unconstrained Optimization

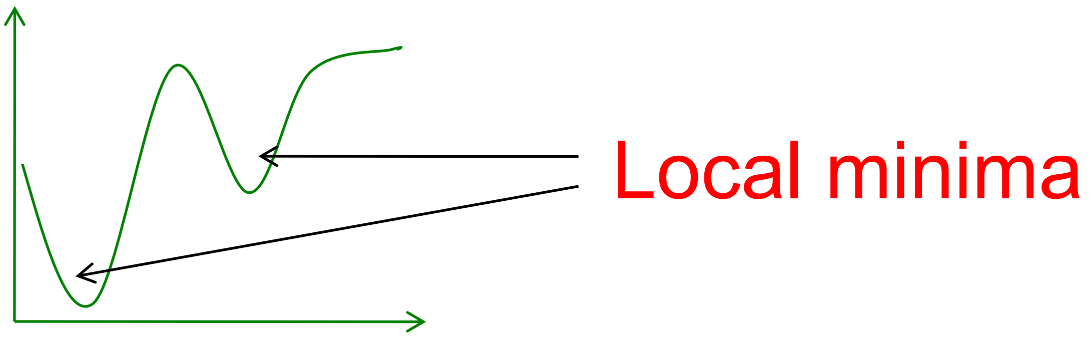
- Ignoring constraints, focus on minimization/maximization of a non-linear function f(x)
- Recall basic calculus
- First-order conditions
  - First derivative = 0
- Second-order conditions
  - 2^{nd} derivative > 0 => Minimum
  - 2^{nd} derivative < 0 => Maximum





Unconstrained Optimization

- Converting a maximization problem to a minimization problem and vice versa is a useful tool:
  - Max f(x) = Min [-f(x)]
  - Solving the complementary problem is the “dual” and provides useful information
- Complexities arise even in the unconstrained world for nonlinear functions



Unconstrained Optimization (formal)

- Unconstrained local minimum x* defined s.t.:
  - $f(x^*) <= f(x)$ *for all x with* $|x-x^*| <= e$
- Write the Taylor series for f (x) about x* as
  - $f(x^* + a) = f(x^*) + a(df(x)/dx) + 0.5a^2\, d^2f(x)/dx^2 + \ldots$ *[at x = x*]*
  - For small a, first three terms dominate.
- We see from the Taylor series that for x to be a local minimum then:
  - $df(x)/dx = 0$ (first derivate zero)
  - $d^2f(x)/dx^2 >= 0$ (second derivative > 0)
- These are called the first- and second-order necessary conditions for a local minimum.

Example: Unconstrained Optimization

- Find the local minima, maxima and the global minimum, maximum (if finite) for the following equations:

- Find first derivative, use Matlab or other method to find the roots of:

$$f(x) = x^5 + 4x^4 - 67x^3 - 22x^2 + 444x - 360$$

- Find second derivative at each root point, if positive minima, else it is a maxima

Unconstrained Optimization (formal)

- In the vector case, where x is a real vector, the Taylor series about x* is:

$$f(x^* + \alpha s) = f(x^*) + \alpha s^T \nabla f(x^*) + \frac{1}{2}\alpha s^T \nabla^2 f(x^*)\, s + \cdots$$

- In this case, the first order condition now depends on the gradient of f(x):

$$\nabla f(x^*) = 0$$

- And the second order condition now depends on the Hessian:

$$\nabla^2 f(x^*) \geq 0.$$

  - A symmetric square matrix >= 0 is positive semi-definite or > 0 is positive definite
  - Thus, Hessian positive semi-definite for minima

Unconstrained Optimization (formal)

- In the scalar example, one variable allowed easy identification of roots and minima/maxima
- Determining $\nabla f\left(x^{*}\right)=0$ for a vector optimization problem often involves a system of non-linear equations
- Therefore need to use a method to solve system of non-linear equations, such as Newton's method

Optimization: Newton's Method

- Process for Newton's method:
  - Make an initial guess x0
  - Expand g(x) around x0 using Taylor series:

$$g\left(x^0 + \Delta x\right) = g\left(x^0\right) + \nabla g\left(x^0\right)\Delta x + \cdots$$

$$\nabla g\left(x^0\right) = \left.\begin{pmatrix} \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n(x)}{\partial x_1} & \cdots & \frac{\partial g_n(x)}{\partial x_n} \end{pmatrix}\right|$$

  - Want to find delta-x such that:

$$g\left(x^0 + \Delta x\right) \approx g\left(x^0\right) + \nabla g\left(x^0\right)\Delta x = 0$$

Optimization: Newton's Method

- Process for Newton's method (continued):
  - Solve for delta-x:

$$\Delta x = -\left[\nabla g\left(x^0\right)\right]^{-1} g\left(x^0\right)$$

  - Update for new estimate of x:

$$x^1 = x^0 + \Delta x$$

  - Iterate process beginning with x1
  - Repeat until process converges (i.e. the magnitude of delta-x smaller than a threshold value / tolerance)
  - *A* solution will be found, not necessarily *THE* solution

Example II: Unconstrained Optimization

- Find the local minima, maxima and the global minimum, maximum (if finite) for the following equation:
  - F(x) = (x1 + 3)^2 + x2^2 – x1*x2 + cos (x1) + 5x2 + 6
- Find first order condition (gradient) equation:
  - $\nabla f\left(x^*\right)=0 \quad = \quad \begin{pmatrix} \dfrac{\partial f(x)}{\partial x_1} \\ \dfrac{\partial f(x)}{\partial x_2} \end{pmatrix}$
- Similarly, find Hessian matrix
  - $\nabla^2 f\left(x^*\right) \geq 0. \quad = \quad \begin{pmatrix} \dfrac{\partial^2 f(x)}{\partial x_1^2} & \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_2^2} \end{pmatrix}$
- Use Matlab to find minimum/maximum

Recap

Recap

- Optimization is a complex discipline involving a wide variety of problems
- Type of problem will determine the possible types of methods for solving the problem
- The two most simple optimization problems are:
  - Unconstrained optimization of a non-linear convex problem
  - Constrained optimization of a linear problem
- References:
  - Winston & Venkataramanan, *Introduction to Mathematical Programming, 4th Edition*
  - Course text on Optimization