ECEN 615
Methods of Electric Power Systems Analysis

Lecture 21: Economic Dispatch, Optimal Power Flow, Linear Programming

Prof. Tom Overbye

Dept. of Electrical and Computer Engineering

Texas A&M University

overbye@tamu.edu

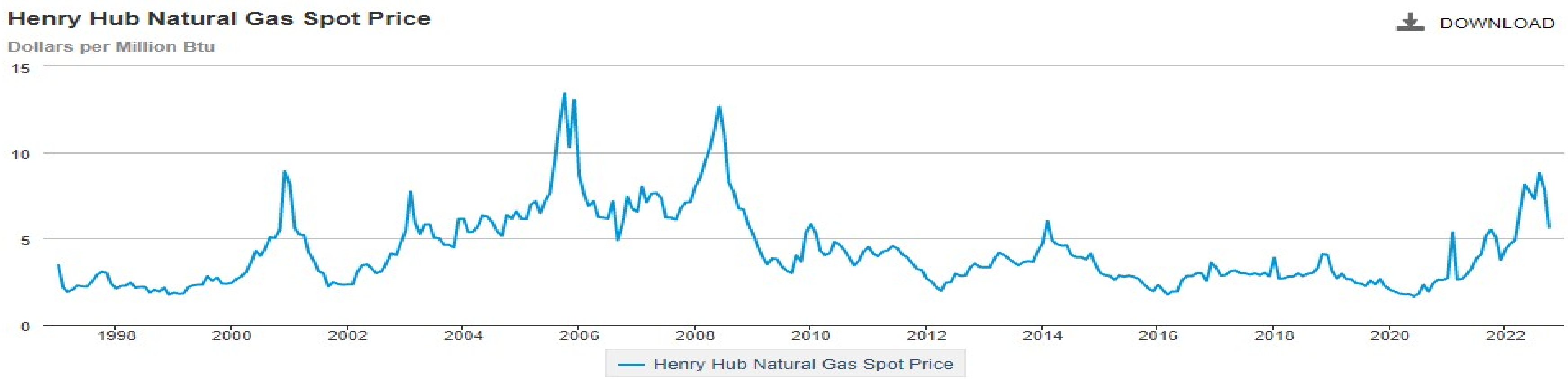
Announcements

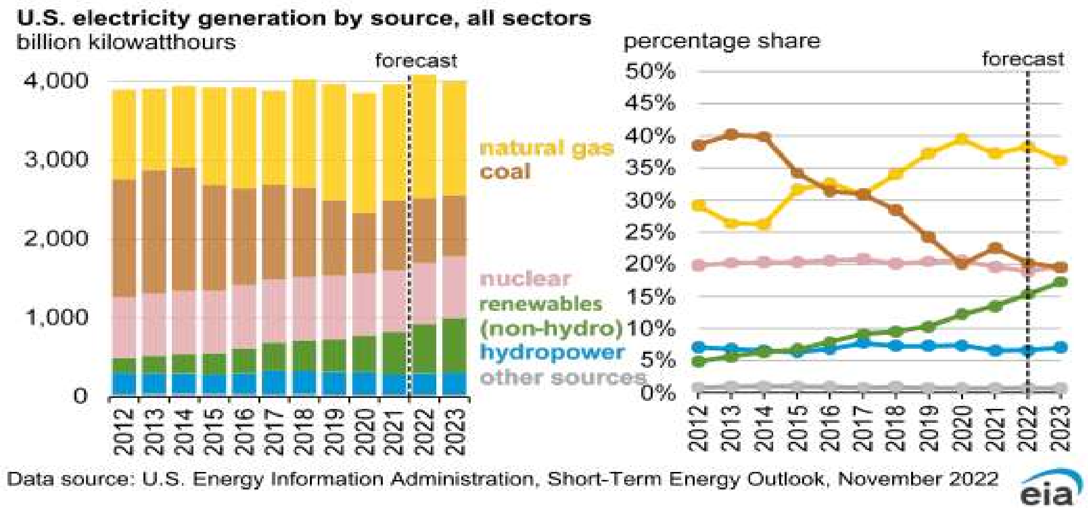
- Read Chapter 8
- Read the Chapter 3 appendices (3A covers optimization with constraints, 3B covers linear programming, 3D covers dynamic programming, and 3E convex optimization
- An excellent book on optimization is Linear and Nonlinear Programming by Luenberger and Ye (the 5^{th} edition came out in 2021)
- Homework 6 is now due on Thursday Nov 17 but it counts as two regular homeworks.

Power System Economic Dispatch

- Generators can have vastly different incremental operational costs
  - Some are essentially free or low cost (wind, solar, hydro, nuclear)
  - Because of the large amount of natural gas generation, electricity prices are very dependent on natural gas prices
- Economic dispatch is concerned with determining the best dispatch for generators without changing their commitment
- Unit commitment focuses on optimization over several days. It is discussed in Chapter 4 of the book, but will just be briefly covered here

Variation in Natural Gas Prices and Generation Sources





Source: www.eia.gov/dnav/ng/hist/rngwhhdm.htm

Power System Economic Dispatch

- Economic dispatch is formulated as a constrained minimization
  - The cost function is often total generation cost in an area
  - Single equality constraint is the real power balance equation
- Solved by setting up the Lagrangian (with P_D the load and P_L the losses, which are a function the generation)

$$\mathrm{L}(\mathbf{P}_{\mathrm{G}},\lambda) = \sum_{i=1}^{m} C_i(P_{Gi}) + \lambda(P_D + P_L(\mathbf{P}_G) - \sum_{i=1}^{m} P_{Gi})$$

- A necessary condition for a minimum is that the gradient is zero. Without losses this occurs when all generators are dispatched at the same marginal cost (except when they hit a limit)

Power System Economic Dispatch

$$\mathrm{L}(\mathbf{P}_{\mathrm{G}},\lambda) = \sum_{i=1}^{m} C_i(P_{Gi}) + \lambda(P_D + P_L(P_G) - \sum_{i=1}^{m} P_{Gi})$$

$$\frac{\partial \mathrm{L}(\mathbf{P}_{\mathrm{G}},\lambda)}{\partial P_{Gi}} = \frac{dC_i(P_{Gi})}{dP_{Gi}} - \lambda(1 - \frac{\partial P_L(P_G)}{\partial P_{Gi}}) = 0$$

$$P_D + P_L(P_G) - \sum_{i=1}^{m} P_{Gi} = 0$$

- If losses are neglected then there is a single marginal cost (lambda); if losses are included then each bus could have a different marginal cost

Economic Dispatch Penalty Factors

Solving each equation for λ we get

$$\frac{dC_i(P_{Gi})}{dP_{Gi}} - \lambda(1 - \frac{\partial P_L(P_G)}{\partial P_{Gi}} = 0$$

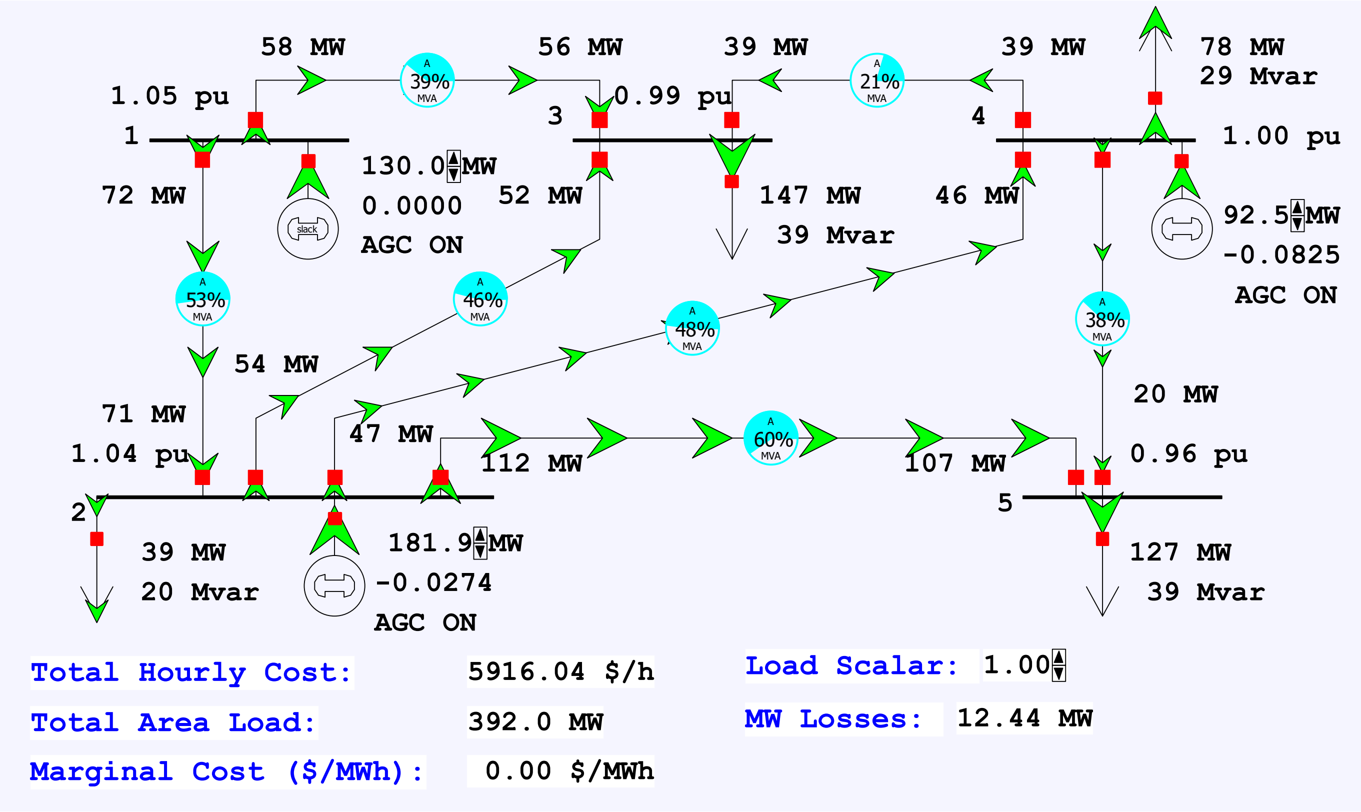
$$\lambda = \frac{1}{\left(1 - \frac{\partial P_L(P_G)}{\partial P_{Gi}}\right)} \frac{dC_i(P_{Gi})}{dP_{Gi}}$$

Define the penalty factor L_i for the i^{th} generator

$$L_i = \frac{1}{\left(1 - \frac{\partial P_L(P_G)}{\partial P_{Gi}}\right)}$$

The penalty factor at the slack bus is always unity!

Economic Dispatch Example



Case is GOS_Example6_22; use **Power Flow Solution Options, Advanced Options** to set Penalty Factors

Optimal Power Flow (OPF)

- OPF functionally combines the power flow with economic dispatch
- SCOPF adds in contingency analysis
- Goal of OPF and SCOPF is to minimize a cost function, such as operating cost, taking into account realistic equality and inequality constraints
- Equality constraints
  - bus real and reactive power balance
  - generator voltage setpoints
  - area MW interchange

OPF, cont.

- Inequality constraints
  - transmission line/transformer/interface flow limits
  - generator MW limits
  - generator reactive power capability curves
  - bus voltage magnitudes (not yet implemented in Simulator OPF)
- Available Controls
  - generator MW outputs
  - transformer taps and phase angles
  - reactive power controls

Two Example OPF Solution Methods

- Non-linear approach using Newton's method
  - handles marginal losses well, but is relatively slow and has problems determining binding constraints
  - Generation costs (and other costs) represented by quadratic or cubic functions
- Linear Programming
  - fast and efficient in determining binding constraints, but can have difficulty with marginal losses.
  - used in PowerWorld Simulator
  - generation costs (and other costs) represented by piecewise linear functions
- Both can be implemented using an ac or dc power flow

OPF and SCOPF Current Status

- OPF (really SCOPF) is currently an area of active research, with ARPA-E having an SCOPF competition (see gocompetition.energy.gov)
- A 2016 National Academies Press report, titled “Analytic Research Foundations for the Next-Generation Electric Grid,” recommended improved AC OPF models
  - I would recommend reading this report; it provides good background on power systems include OPF
  - It is available for free at www.nap.edu/catalog/21919/analytic-research-foundations-for-the-next-generation-electric-grid

OPF and SCOPF History

- A nice OPF history from Dec 2012 is provided by the below link, and briefly summarized here
- Prior to digital computers economic dispatch was solved by hand and the power flow with network analyzers
- Digital power flow developed in late 50's to early 60's
- First OPF formulations in the 1960's
  - J. Carpienterm, "Contribution e l'étude do Dispatching Economique," Bulletin Society Francaise Electriciens, 1962
  - H.W. Dommel, W.F. Tinney, "Optimal power flow solutions," *IEEE Trans. Power App. and Sys*tems, Oct. 1968
    - "Only a small extension of the power flow program is required"

www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf (by M Cain, R. O'Neill, A. Castillo)

OPF and SCOPF History

- A linear programming (LP) approach was presented by Stott and Hobson in 1978
  - B. Stott, E. Hobson, "Power System Security Control Calculations using Linear Programming," (Parts 1 and 2) *IEEE Trans. Power App and Syst.*, Sept/Oct 1978
- Optimal Power Flow By Newton's Method
  - D.I. Sun, B. Ashley, B. Brewer, B.A. Hughes, and W.F. Tinney, "Optimal Power Flow by Newton Approach", *IEEE Trans. Power App and Syst.*, October 1984
- Follow-up LP OPF paper in 1990
  - O. Alsac, J. Bright, M. Prais, B. Stott, "Further Developments in LP-based Optimal Power Flow," *IEEE Trans. Power Systems*, August 1990

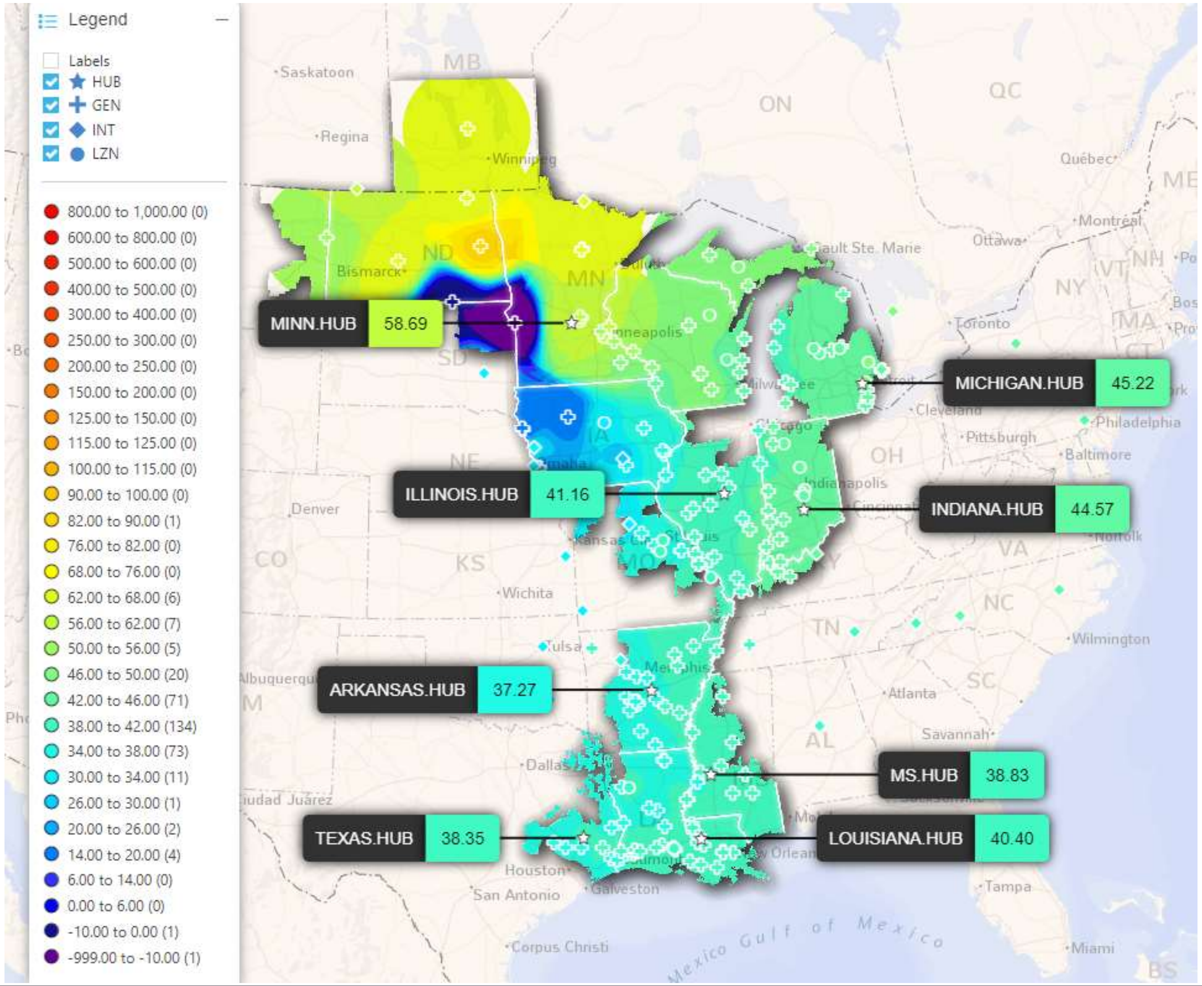
OPF and SCOPF History

- Critique of OPF Algorithms
  - W.F. Tinney, J.M. Bright, K.D. Demaree, B.A. Hughes, "Some Deficiencies in Optimal Power Flow," *IEEE Trans. Power Systems*, May 1988
- Hundreds of other papers on OPF
- Comparison of ac and dc optimal power flow methods
  - T.J. Overbye, X. Cheng, Y. San, "A Comparison of the AC and DC Power Flow Models for LMP Calculations," Proc. 37th Hawaii International Conf. on System Sciences, 2004

Key SCOPF Application: Locational Marginal Prices (LMPs)

- The locational marginal price (LMP) tells the cost of providing electricity to a given location (bus) in the system
- Concept introduced by Schweppe in 1985
  - F.C. Schweppe, M. Caramanis, R. Tabors, "Evaluation of Spot Price Based Electricity Rates," *IEEE Trans. Power App and Syst*., July 1985
- LMPs are a direct result of an SCOPF, and are widely used in many electricity markets worldwide
  –

Example MISO LMP Contour, 11/11/2022

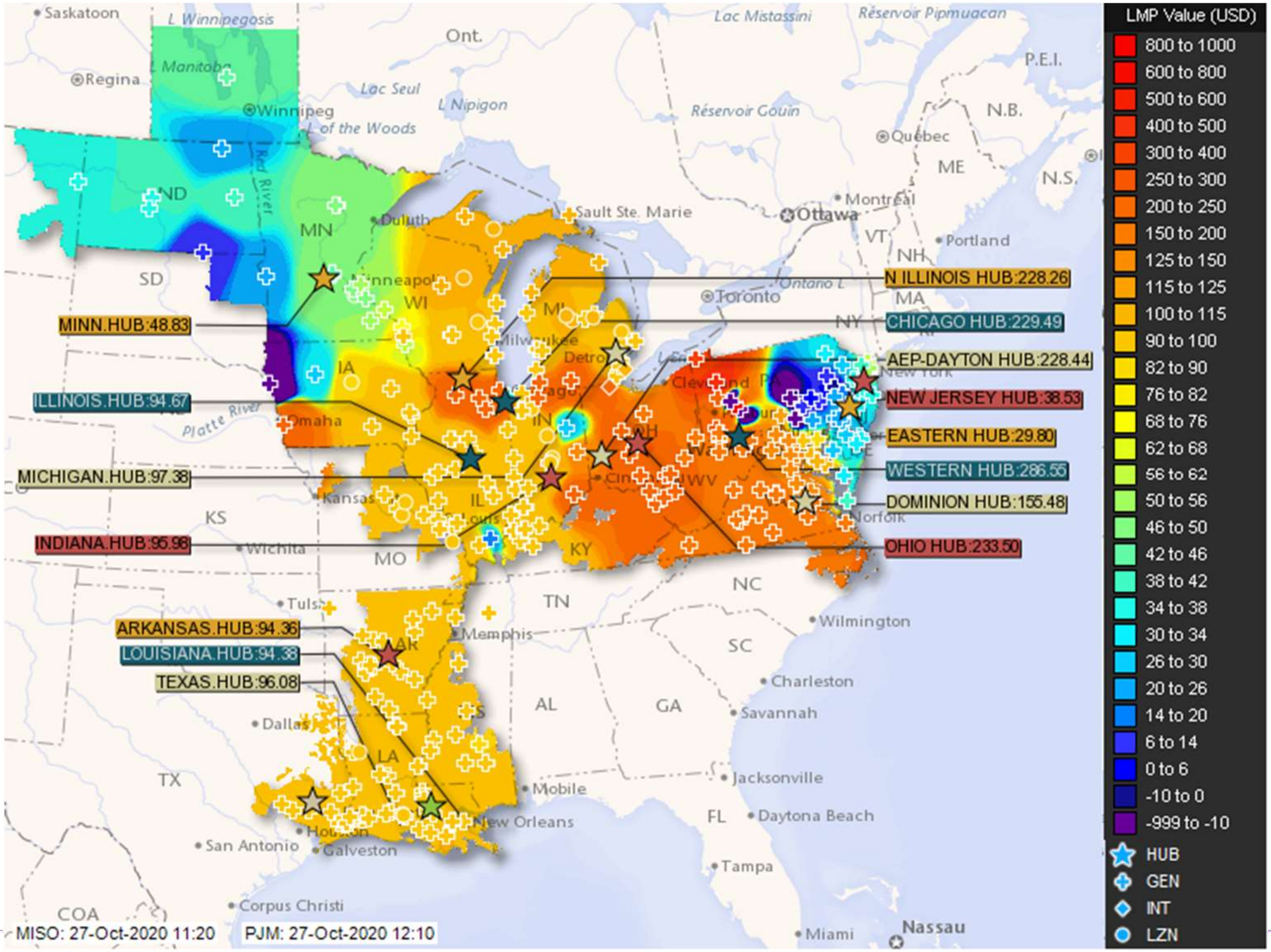


LMPs are now widely visualized using color contours; the first use of LMP color contours was presented in [1]

[1] T.J. Overbye, R.P. Klump, J.D. Weber, "A Virtual Environment for Interactive Visualization of Power System Economic and Security Information," IEEE PES 1999 Summer Meeting, Edmonton, AB, Canada, July 1999

Image: api.misoenergy.org/MISORTWD/lmpcontourmap.html

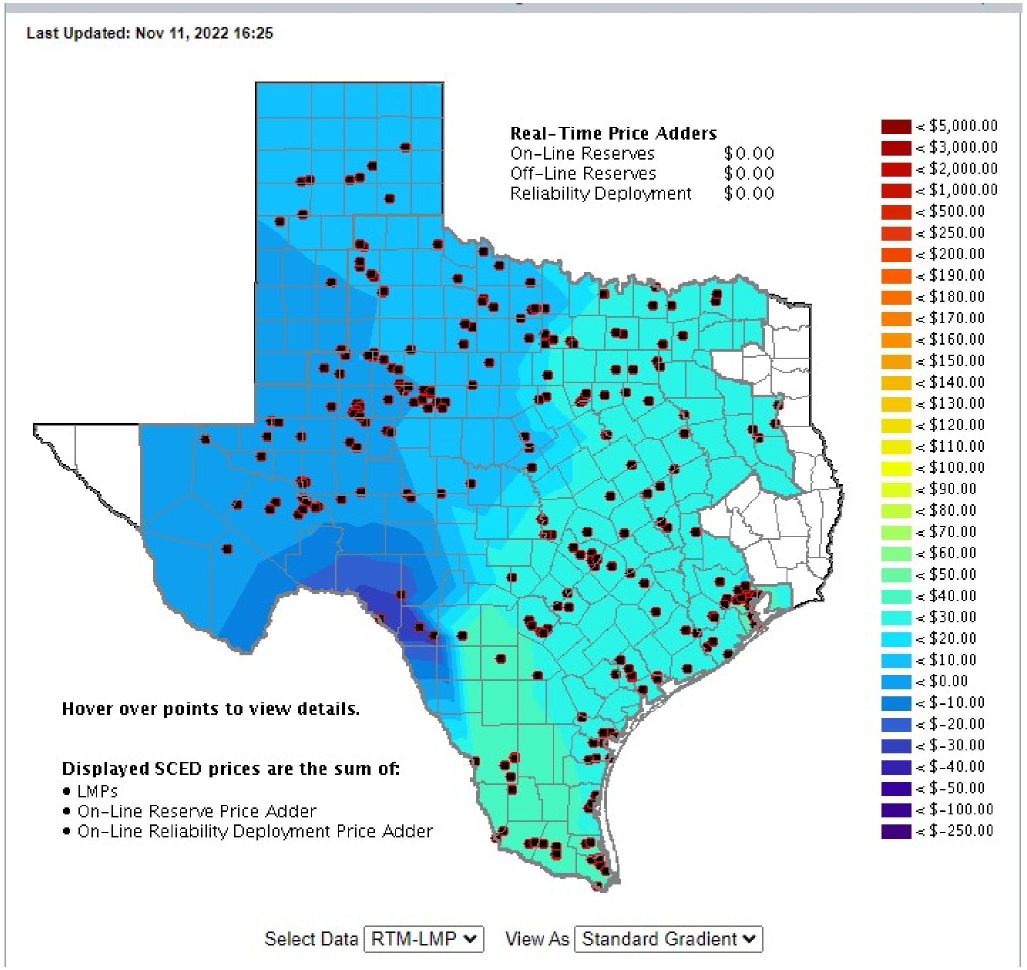
Example LMP Contour: 10/27/2020



Note the wide range in LMPs including some negative values!

This is just the real-time market; most electricity is not traded here.

ERCOT LMPs, Nov 11, 2022 at 4:25 pm



OPF Problem Formulation

- The OPF is usually formulated as a minimization with equality and inequality constraints

$$\text{Minimize } F(\mathbf{x},\mathbf{u})$$

$$\mathbf{g}(\mathbf{x},\mathbf{u}) = \mathbf{0}$$

$$\mathbf{h}_{\min} \le \mathbf{h}(\mathbf{x},\mathbf{u}) \le \mathbf{h}_{\max}$$

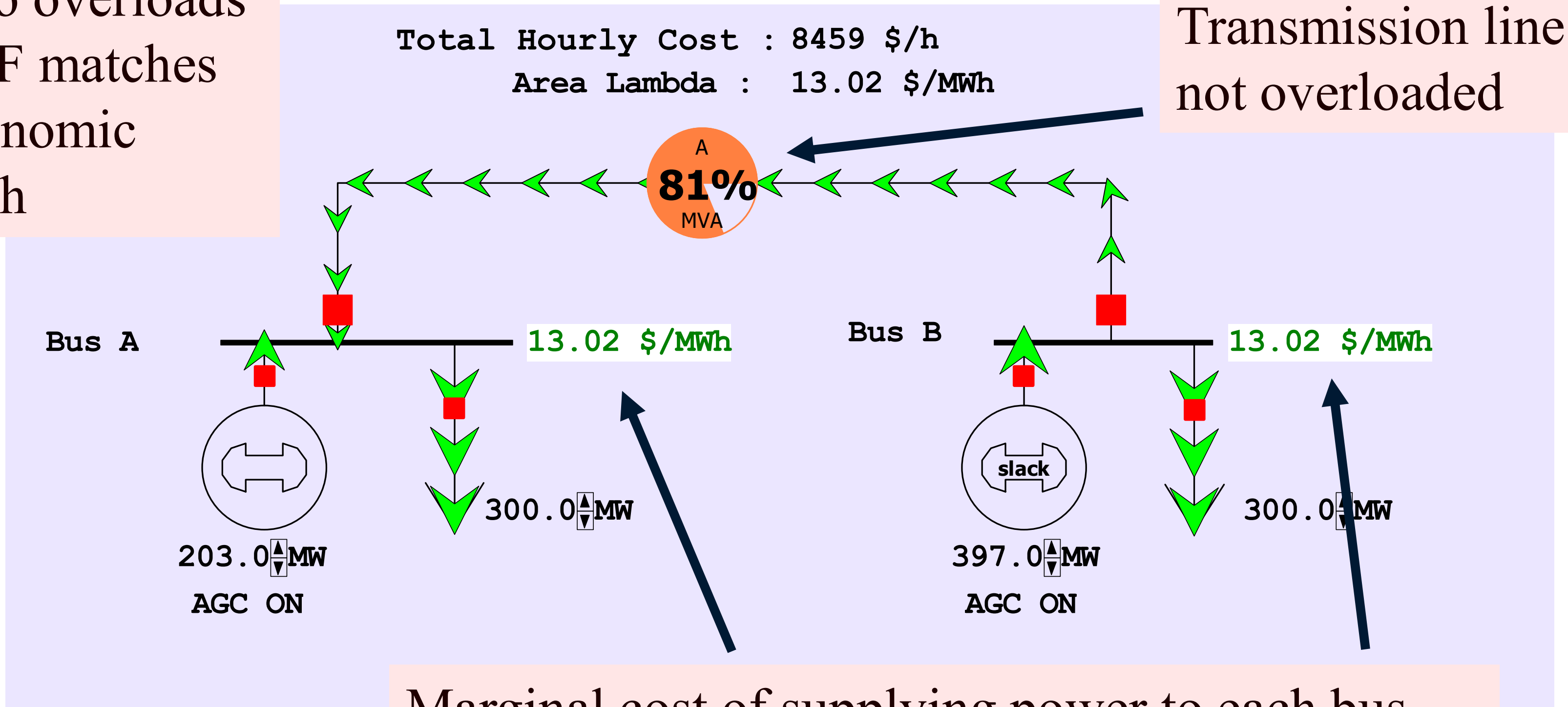
$$\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}$$

where **x** is a vector of dependent variables (such as the bus voltage magnitudes and angles), **u** is a vector of the control variables, F(**x**,**u**) is the scalar objective function, **g** is a set of equality constraints (e.g., the power balance equations) and **h** is a set of inequality constraints (such as line flows)

Two Bus with Unconstrained Line

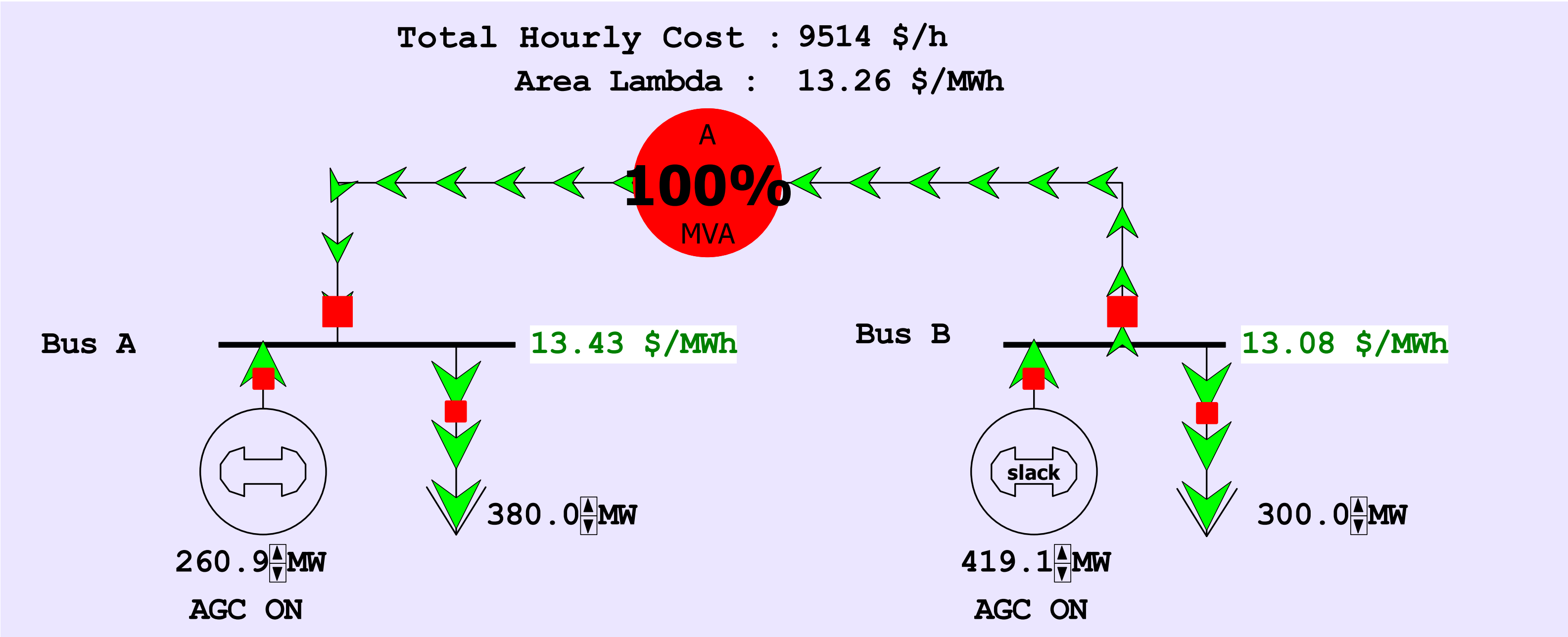
With no overloads the OPF matches the economic dispatch

Transmission line is not overloaded



Marginal cost of supplying power to each bus (locational marginal costs)

Two Bus with Constrained Line

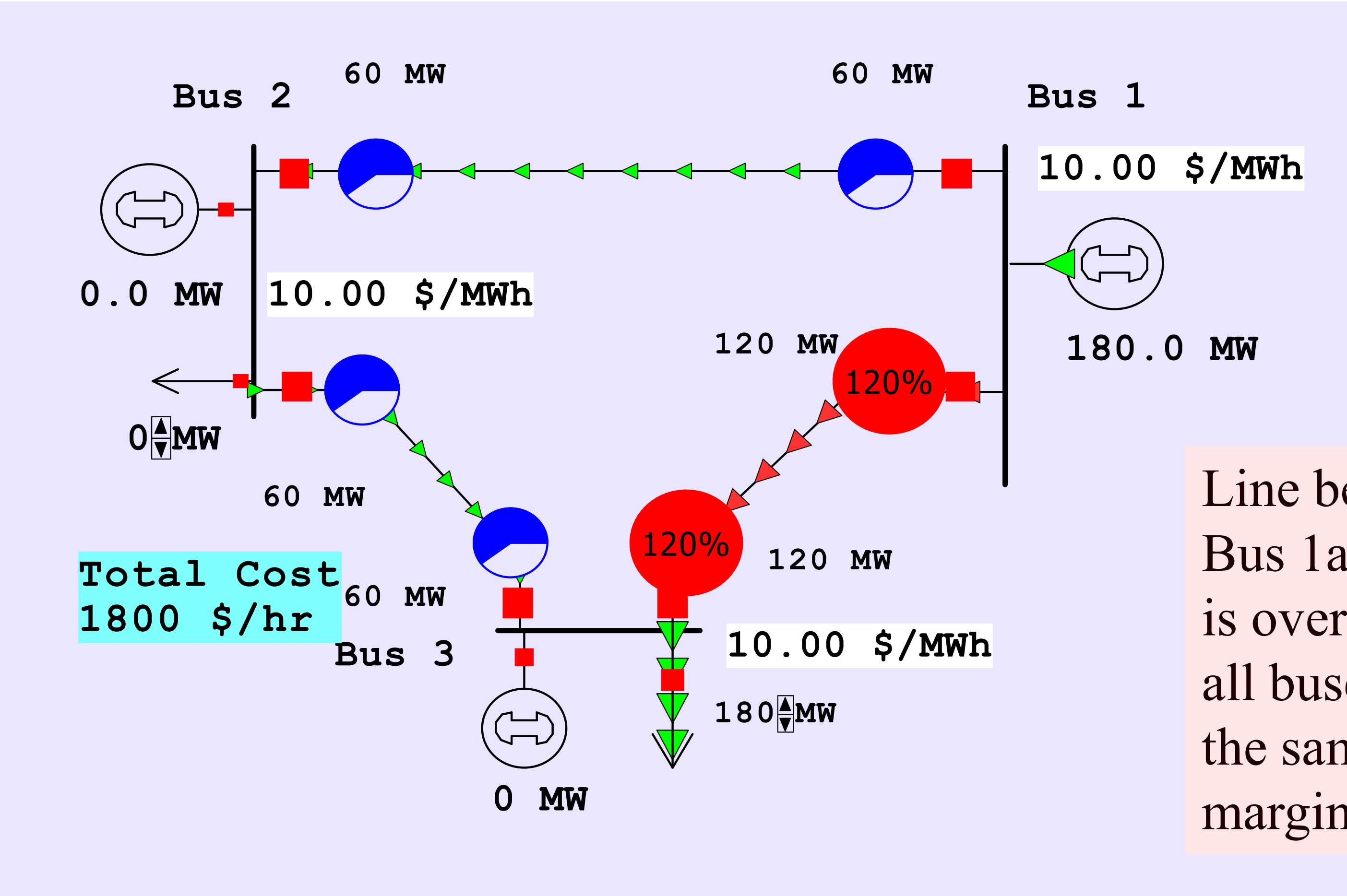


With the line loaded to its limit, additional load at Bus A must be supplied locally, causing the marginal costs to diverge.

Three Bus (B3) Example

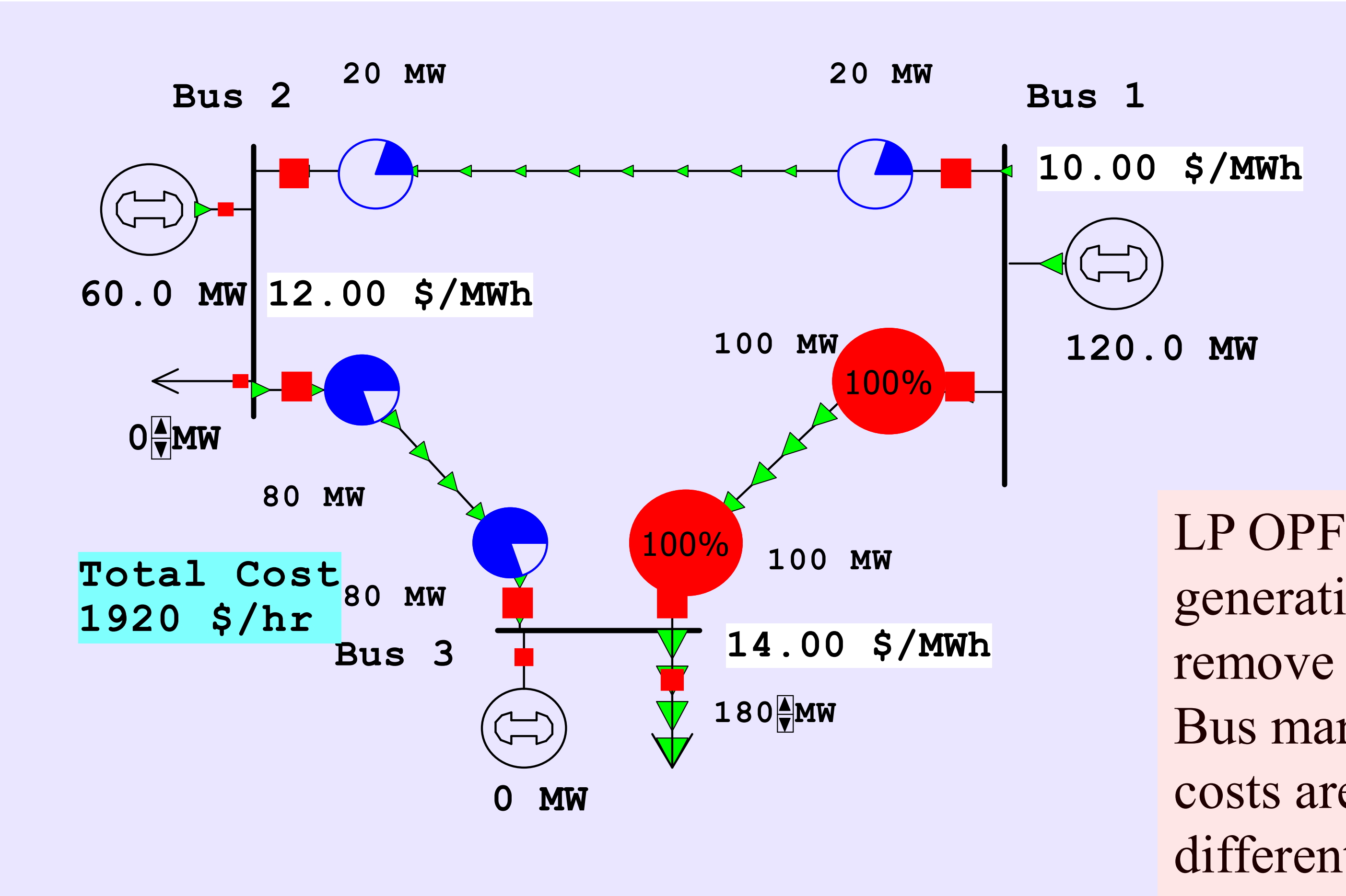
- Consider a three bus case (Bus 1 is system slack), with all buses connected through 0.1 pu reactance lines, each with a 100 MVA limit
- Let the generator marginal costs be
  - Bus 1: 10 $ / MWhr; Range = 0 to 400 MW
  - Bus 2: 12 $ / MWhr; Range = 0 to 400 MW
  - Bus 3: 20 $ / MWhr; Range = 0 to 400 MW
- Assume a single 180 MW load at bus 2

B3 with Line Limits NOT Enforced



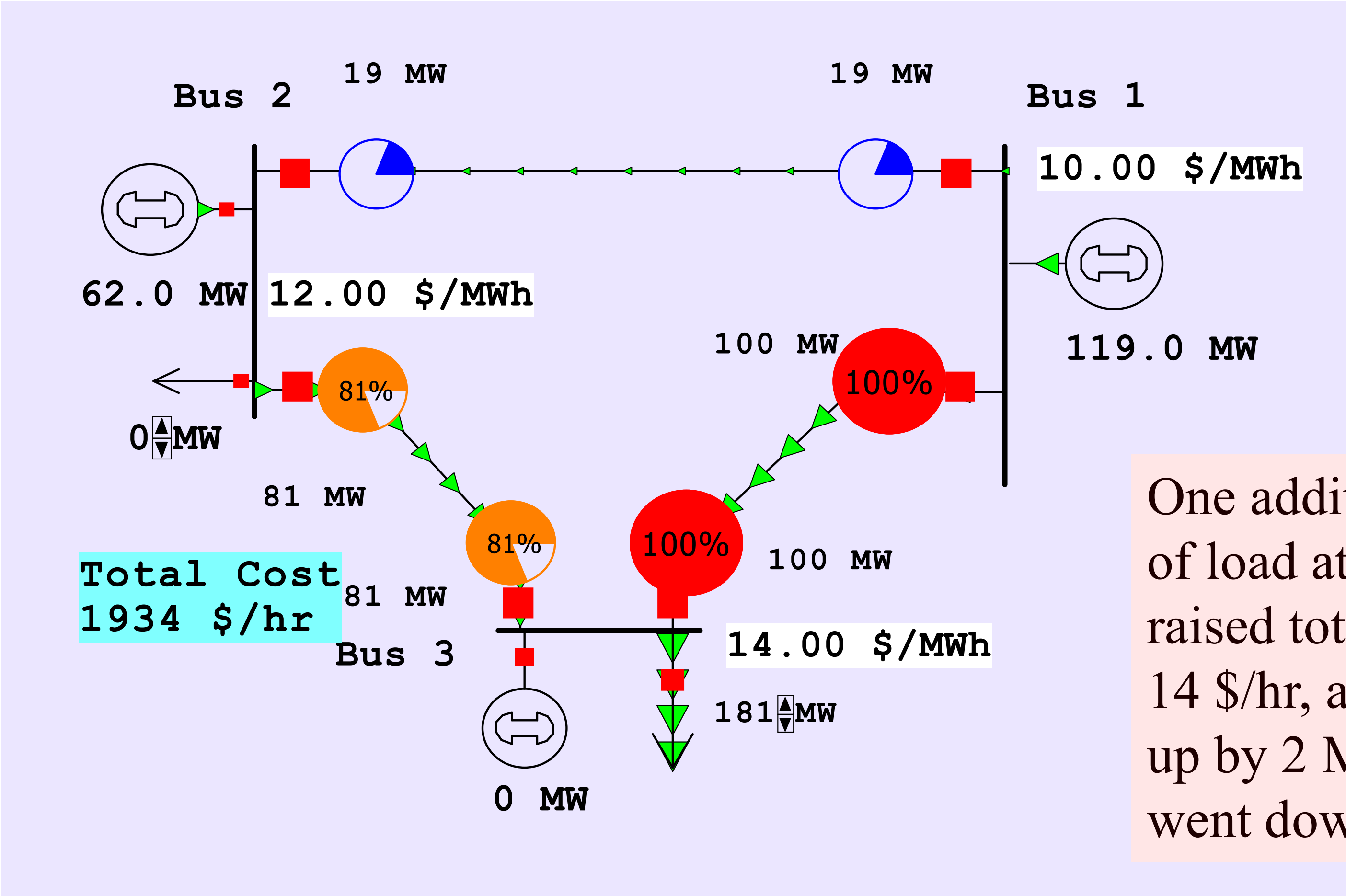
Line between Bus 1and Bus 3 is overloaded; all buses have the same marginal cost

B3 with Line Limits Enforced



LP OPF changes generation to remove violation. Bus marginal costs are now different.

Verify Bus 3 Marginal Cost



One additional MW of load at bus 3 raised total cost by 14 $/hr, as G2 went up by 2 MW and G1 went down by 1MW

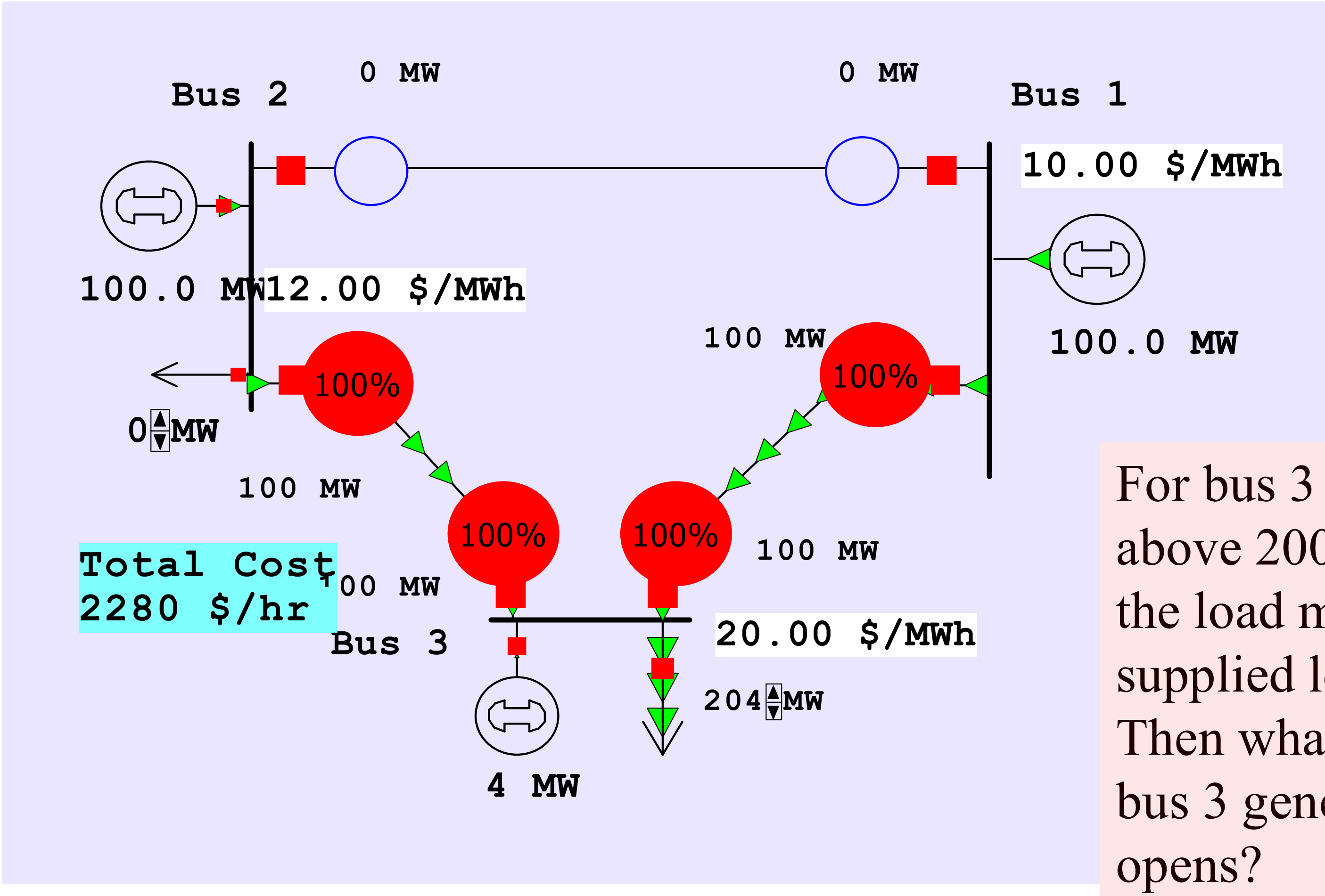
Why is bus 3 LMP = $14 /MWh

- All lines have equal impedance. Power flow in a simple network distributes inversely to impedance of path.
  - For bus 1 to supply 1 MW to bus 3, 2/3 MW would take direct path from 1 to 3, while 1/3 MW would "loop around" from 1 to 2 to 3.
  - Likewise, for bus 2 to supply 1 MW to bus 3, 2/3MW would go from 2 to 3, while 1/3 MW would go from 2 to 1to 3.

Why is bus 3 LMP \$ 14 / MWh, cont'd

- With the line from 1 to 3 limited, no additional power flows are allowed on it.
- To supply 1 more MW to bus 3 we need
  - $\Delta P_{G1} + \Delta P_{G2} = 1$ MW
  - $2/3\ \Delta P_{G1} + 1/3\ \Delta P_{G2} = 0$; (no more flow on 1-3)
- Solving requires we up P_{G2} by 2 MW and drop P_{G1} by 1 MW -- a net increase of \$24 – \$10 = \$14.

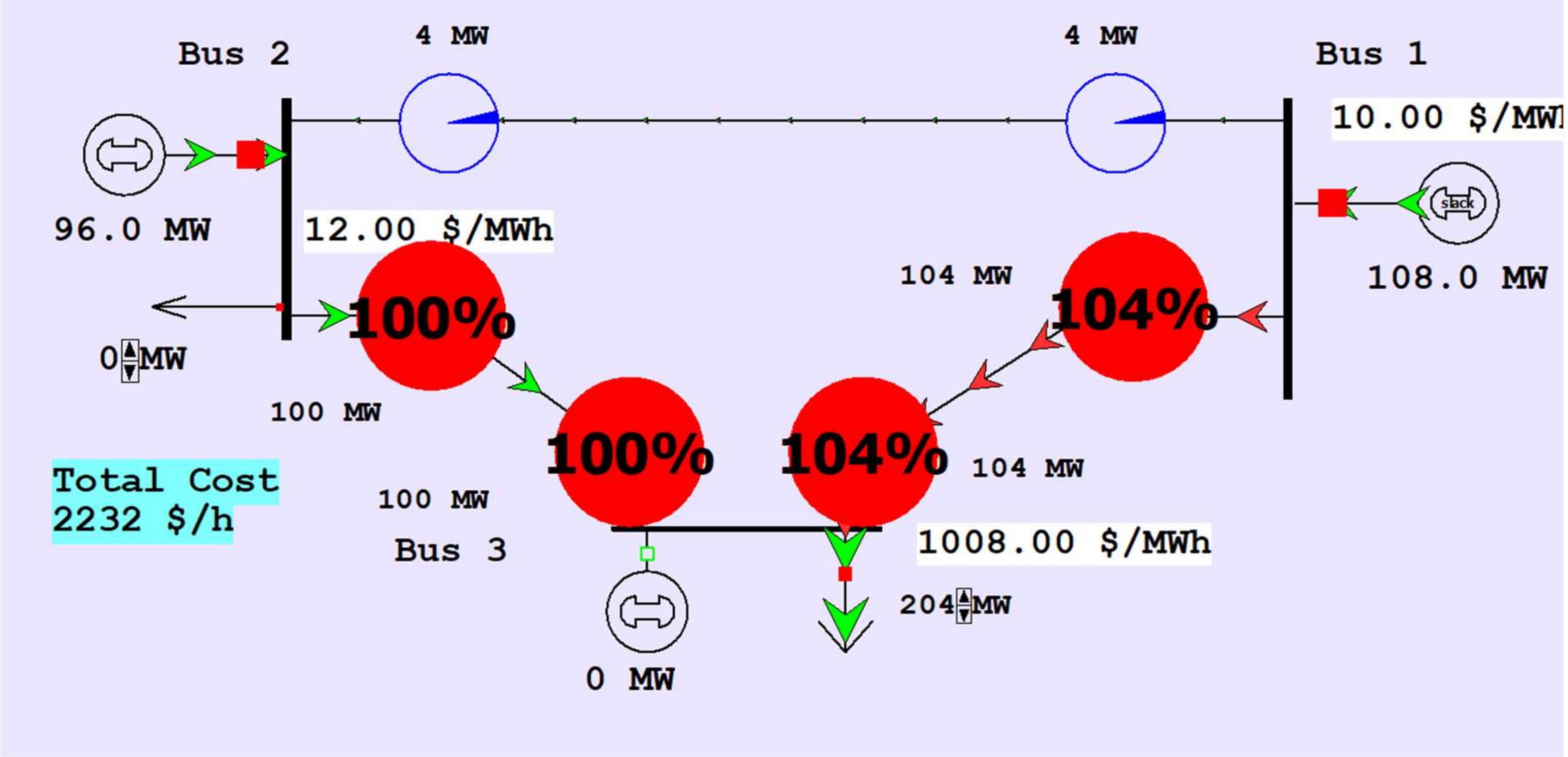
Both lines into Bus 3 Congested



For bus 3 loads above 200 MW, the load must be supplied locally. Then what if the bus 3 generator opens?

Both lines into Bus 3 Congested

An infeasible example can be created by opening the generator at Bus 3 with the Bus 3 load above 200 MW. There is no way to serve the load without overloading a transmission line.



LP OPF Solution Method

- There are different OPF solution techniques. One common approach uses linear programming (LP)
- The LP approach iterates between
  - solving a full ac or dc power flow solution
    - enforces real/reactive power balance at each bus
    - enforces generator reactive limits
    - system controls are assumed fixed
    - takes into account non-linearities
  - solving a primal LP
    - changes system controls to enforce linearized constraints while minimizing cost

Quick Coverage of Linear Programming

- LP is probably the most widely used mathematical programming technique
- It is used to solve linear, constrained minimization (or maximization) problems in which the objective function and the constraints can be written as linear functions

Example Problem 1

- Assume that you operate a lumber mill which makes both construction-grade and finish-grade boards from the logs it receives. Suppose it takes 2 hours to rough-saw and 3 hours to plane each 1000 board feet of construction-grade boards. Finish-grade boards take 2 hours to rough-saw and 5 hours to plane for each 1000 board feet. Assume that the saw is available 8 hours per day, while the plane is available 15 hours per day. If the profit per 1000 board feet is $100 for construction-grade and $120 for finish-grade, how many board feet of each should you make per day to maximize your profit?

Problem 1 Setup

Let x_1=amount of cg, x_2= amount of fg

Maximize $\quad 100x_1 + 120x_2$

s.t. $\quad 2x_1 + 2x_2 \le 8$

$$3x_1 + 5x_2 \le 15$$

$$x_1, x_2 \ge 0$$

Notice that all of the equations are linear, but they are inequality, as opposed to equality, constraints; we are seeking to determine the values of x_1 and x_2

Example Problem 2 (Nutritionist Problem)

- A nutritionist is planning a meal with 2 foods: A and B. Each ounce of A costs \$ 0.20, and has 2 units of fat, 1 of carbohydrate, and 4 of protein. Each ounce of B costs \$0.25, and has 3 units of fat, 3 of carbohydrate, and 3 of protein. Provide the least cost meal which has no more than 20 units of fat, but with at least 12 units of carbohydrates and 24 units of protein.

Problem 2 Setup

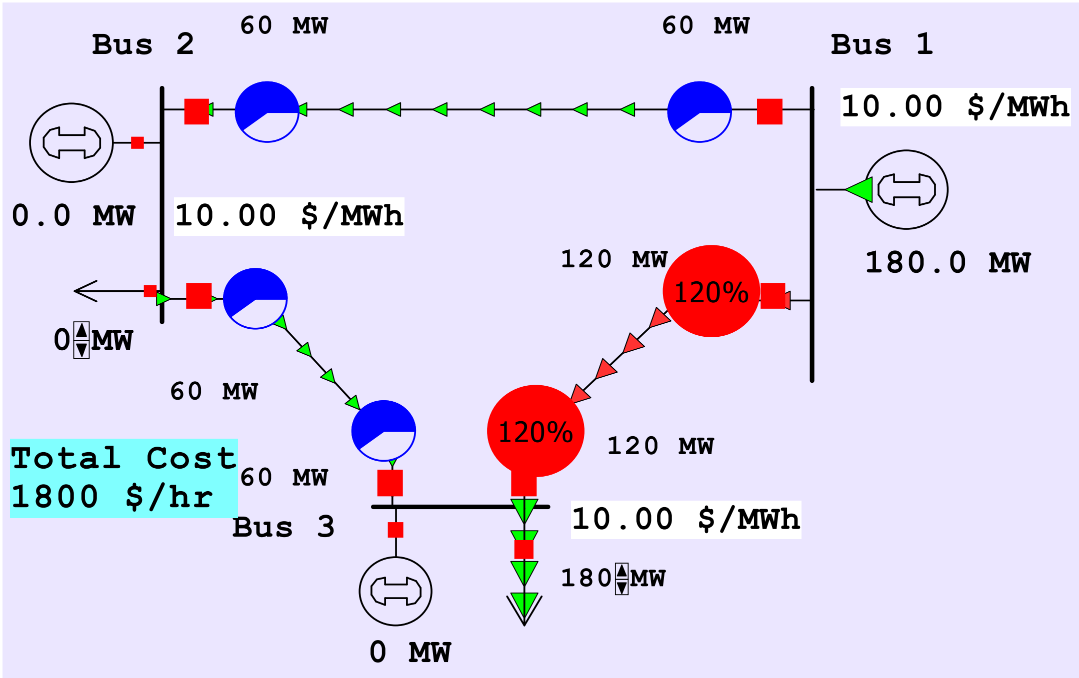
Let x_1=ounces of A, x_2= ounces of B

$$
\begin{aligned}
\text{Minimize}\quad & 0.20x_1 + 0.25x_2 \\
\text{s.t.}\quad & 2x_1 + 3x_2 \le 20 \\
& x_1 + 3x_2 \ge 12 \\
& 4x_1 + 3x_2 \ge 24 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

Again all of the equations are linear, but they are inequality, as opposed to equality, constraints; we are again seeking to determine the values of x_1 and x_2; notice there are also more constraints than solution variables

Three Bus Case Formulation

- For the earlier three bus system given the initial condition of an overloaded transmission line, minimize the cost of generation such that the change in generation is zero, and the flow on the line between buses 1 and 3 is not violating its limit
- Can be setup considering the change in generation, (ΔP_{G1}, ΔP_{G2}, ΔP_{G3})



Three Bus Case Problem Setup

Let $x_1 = \Delta P_{G1}$, $x_2 = \Delta P_{G2}$, $x_3 = \Delta P_{G3}$

Minimize $\quad 10x_1 + 12x_2 + 20x_3$

s.t. $\quad \frac{2}{3}x_1 + \frac{1}{3}x_2 \le -20$ Line flow constraint

$x_1 + x_2 + x_3 = 0$ Power balance constraint

enforcing limits on $x_1,\ x_2,\ x_3$

LP Standard Form

The standard form of the LP problem is

Minimize $\mathbf{c}\,\mathbf{x}$

s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$

$\mathbf{x} \geq \mathbf{0}$

Maximum problems can be treated as minimizing the negative

where $\mathbf{x}$ = n-dimensional column vector

$\mathbf{c}$ = n-dimensional row vector

$\mathbf{b}$ = m-dimensional column vector

$\mathbf{A}$ = $m \times n$ matrix

For the LP problem usually n>> m

The previous examples were not in this form!

Replacing Inequality Constraints with Equality Constraints

- The LP standard form does not allow inequality constraints
- Inequality constraints can be replaced with equality constraints through the introduction of slack variables, each of which must be greater than or equal to zero

$$\ldots \leq b_i \rightarrow \ldots + y_i = b_i \quad \text{with } y_i \geq 0$$

$$\ldots \geq b_i \rightarrow \ldots - y_i = b_i \quad \text{with } y_i \geq 0$$

- Slack variables have no cost associated with them; they merely tell how far a constraint is from being binding, which will occur when its slack variable is zero

Lumber Mill Example with Slack Variables

- Let the slack variables be x_3 and x_4, so

Minimize the negative

$$\begin{aligned} \text{Minimize} \quad & -(100x_1 + 120x_2) \\ \text{s.t.} \quad & 2x_1 + 2x_2 + x_3 = 8 \\ & 3x_1 + 5x_2 + x_4 = 15 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

LP Definitions

This is a key LP concept!

A vector $\mathbf{x}$ is said to be basic if

1. $\mathbf{Ax} = \mathbf{b}$
2. At most m components of $\mathbf{x}$ are non-zero; these are called the basic variables; the rest are non basic variables; if there are less than m non-zeros then $\mathbf{x}$ is called degenerate

$\mathbf{A}_{\mathrm{B}}$ is called the basis matrix

Define $\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\mathrm{B}} \\ \mathbf{x}_{\mathrm{N}} \end{bmatrix}$ (with $\mathbf{x}_{\mathrm{B}}$ basic) and $\mathbf{A} = \begin{bmatrix} \mathbf{A}_B & \mathbf{A}_N \end{bmatrix}$

With $\begin{bmatrix} \mathbf{A}_B & \mathbf{A}_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathrm{B}} \\ \mathbf{x}_{\mathrm{N}} \end{bmatrix} = \mathbf{b}$ so $\mathbf{x}_{\mathrm{B}} = \mathbf{A}_B^{-1}\left(\mathbf{b} - \mathbf{A}_N \mathbf{x}_{\mathrm{N}}\right)$

Fundamental LP Theorem

- Given an LP in standard form with **A** of rank m then
  - If there is a feasible solution, there is a basic feasible solution
  - If there is an optimal, feasible solution, then there is an optimal, basic feasible solution
- Note, there could be a LARGE number of basic, feasible solutions
  - Simplex algorithm determines the optimal, basic feasible solution usually very quickly

LP Graphical Interpretation

- The LP constraints define a polyhedron in the solution space
  - This is a polytope if the polyhedron is bounded and nonempty
  - The basic, feasible solutions are vertices of this polyhedron
  - With the linear cost function the solution will be at one of vertices

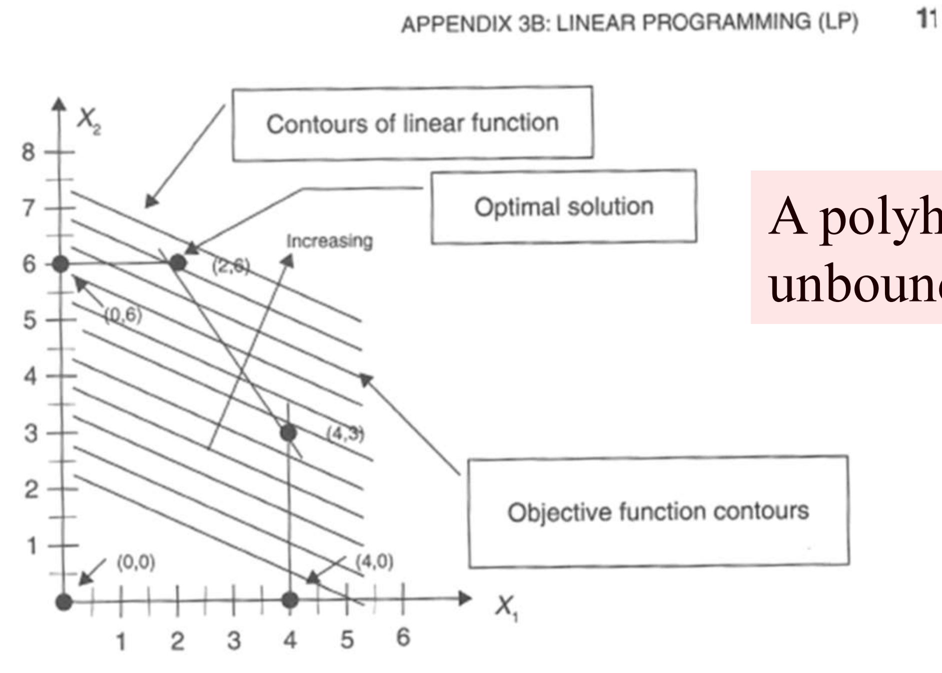


FIGURE 3.26 x_1, x_2 plane with cost contours and the optimal solution shown.

A polyhedron can be unbounded

Image: Figure 3.26 from course text

Simplex Algorithm

- The key is to move intelligently from one basic feasible solution (i.e., a vertex) to another, with the goal of continually decreasing the cost function
- The algorithm does this by determining the "best" variable to bring into the basis; this requires that another variable exit the basis, while always retaining a basic, feasible solution
- This is called pivoting

Determination of Variable to Enter the Basis

- To determine which non-basic variable should enter the basis (i.e., one which currently 0), look at how the cost function changes w.r.t. to a change in a non-basic variable (i.e., one that is currently zero)

$$\text{Define } \mathbf{z} = \mathbf{c}\,\mathbf{x} = [\mathbf{c}_B \quad \mathbf{c}_N]\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$$

$$\text{With } \mathbf{x}_\mathrm{B} = \mathbf{A}_B^{-1}\left(\mathbf{b} - \mathbf{A}_N \mathbf{x}_\mathrm{N}\right)$$

$$\text{Then } \mathbf{z} = \mathbf{c}_B \mathbf{A}_B^{-1}\mathbf{b} + \left(\mathbf{c}_N - \mathbf{c}_B \mathbf{A}_B^{-1}\mathbf{A}_N\right)\mathbf{x}_\mathrm{N}$$

Elements of $\mathbf{x}_\mathrm{n}$ are all zero, but we are looking to change one to decrease the cost

Determination of Variable to Enter the Basis, cont.

- Define the reduced (or relative) cost coefficients as

$$\mathbf{r} = \mathbf{c}_N - \mathbf{c}_B \mathbf{A}_B^{-1} \mathbf{A}_N$$

r is an n-m dimensional row vector

- Elements of this vector tell how the cost function will change for a change in a currently non-basic variable
- The variable to enter the basis is usually the one with the most negative relative cost
- If all the relative costs are nonnegative then we are at an optimal solution

Determination of Variable to Exit Basis

- The new variable entering the basis, say a position j, causes the values of all the other basic variables to change. In order to retain a basic, feasible solution, we need to insure no basic variables become negative. The change in the basic variables is given by

$$\tilde{\mathbf{x}}_B = \mathbf{x}_B - \mathbf{A}_B^{-1}\mathbf{a}_j\,\varepsilon$$

where ε is the value of the variable entering the basis, and $\mathbf{a}_j$ is its associated column in $\mathbf{A}$

Determination of Variable to Exit Basis, cont.

We find the largest value ε such

$$\tilde{\mathbf{x}}_B = \mathbf{x}_B - \mathbf{A}_B^{-1}\mathbf{a}_j\varepsilon \geq \mathbf{0}$$

If no such ε exists then the problem is unbounded; otherwise at least one component of $\tilde{\mathbf{x}}_B$ equals zero. The associated variable exits the basis.

Canonical Form

- The Simplex Method works by having the problem in what is known as canonical form
- Canonical form is defined as having the m basic variables with the property that each appears in only one equation, its coefficient in that equation is unity, and none of the other basic variables appear in the same equation
- Sometime canonical form is readily apparent

$$\begin{array}{ll} \text{Minimize} & -(100x_1 + 120x_2) \\ \text{s.t.} & 2x_1 + 2x_2 + x_3 = 8 \\ & 3x_1 + 5x_2 + x_4 = 15 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{array}$$

Note that with x_3 and x_4 as basic variables $\mathbf{A}_B$ is the identity matrix

Canonical Form

- Other times canonical form is achieved by initially adding artificial variables to get an initial solution
- Example of the nutrition problem in canonical form with slack and artificial variables (denoted as **y**) used to get an initial basic feasible solution

Let x_1=ounces of A, x_2= ounces of B

Minimize $y_1+y_2+y_3$

s.t.

$$2x_1 + 3x_2 + x_3 + y_1 = 20$$
$$x_1 + 3x_2 - x_4 + y_2 = 12$$
$$4x_1 + 3x_2 - x_5 + y_3 = 24$$
$$x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3 \ge 0$$

Note that with y_1, y_2, and y_3 as basic variables $\mathbf{A}_B$ is the identity matrix

LP Tableau

- With the system in canonical form, the Simplex solution process can be illustrated by forming what is known as the LP tableau
  - Initially this corresponds to the **A** matrix, with a column appended to include the **b** vector, and a row added to give the relative cost coefficients; the last element is the negative of the cost function value
  - Define the tableau as **Y**, with elements $\mathbf{Y}_{ij}$
  - In canonical form the last column of the tableau gives the values of the basic variables
- During the solution the tableau is updated by pivoting

LP Tableau for the Nutrition Problem with Artificial Variables

- When in canonical form the relative costs vector is

$$\mathbf{r} = \mathbf{c}_N - \mathbf{c}_B \mathbf{A}_B^{-1} \mathbf{A}_N = \mathbf{c}_B \mathbf{A}_N$$

$$\mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T - \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 & 0 & 0 \\ 1 & 3 & 0 & -1 & 0 \\ 4 & 3 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -7 \\ -9 \\ -1 \\ 1 \\ 1 \end{bmatrix}^T$$

- The initial tableau for the artificial problem is then

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
2	3	1	0	0	1	0	0	20
1	3	0	–1	0	0	1	0	12
4	3	0	0	–1	0	0	1	24
–7	–9	–1	1	1	0	0	0	–56

Note the last column gives the values of the basic variables

LP Tableau Pivoting

- Pivoting is used to move from one basic feasible solution to another
  - Select the pivot column (i.e., the variable coming into the basis, say q) as the one with the most negative relative cost
  - Select the pivot row (i.e., the variable going out of the basis) as the one with the smallest ratio of x_i/Y_{iq} for $Y_{iq} > 0$; define this as row p (x_i is given in the last column)

That is, we find the largest value ε such

$$\tilde{\mathbf{x}}_B = \mathbf{x}_B - \mathbf{A}_B^{-1}\mathbf{a}_q\varepsilon \geq \mathbf{0}$$

If no such ε exists then the problem is unbounded;

otherwise at least one component of $\tilde{\mathbf{x}}_B$ equals zero.

The associated variable exits the basis.

LP Tableau Pivoting for Nutrition Problem

- Starting at

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
2	3	1	0	0	1	0	0	20
1	3	0	–1	0	0	1	0	12
4	3	0	0	–1	0	0	1	24
–7	–9	–1	1	1	0	0	0	–56

- Pivot on column q=2; for row get minimum of {20/3, 12/3, 24/3), which is row p=2

LP Tableau Pivoting

- Pivoting on element Y_{pq} is done by
  - First dividing row p by Y_{pq} to change the pivot element to unity.
  - Then subtracting from the k^{th} row Y_{kq}/Y_{pq} times the p^{th} row for all rows with Y_{kq} <> 0

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
2	3	1	0	0	1	0	0	20
1	**3**	0	–1	0	0	1	0	12
4	3	0	0	–1	0	0	1	24
–7	–9	–1	1	1	0	0	0	–56

I'm only showing fractions with two ROD digits

	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
	1	0	1	1	0	1	–1	0	8
Pivoting gives	0.33	1	0	–0.33	0	0	0.33	0	4
	3	0	0	1	–1	0	–1	1	12
	–4	0	–1	–2	1	0	3	0	–20

LP Tableau Pivoting, Example, cont.

- Next pivot on column 1, row 3

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
1	0	1	1	0	1	–1	0	8
0.33	1	0	–0.33	0	0	0.33	0	4
3	0	0	1	–1	0	–1	1	12
–4	0	–1	–2	1	0	3	0	–20

- Which gives

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
0	0	1	0.67	0.33	1	–0.67	–0.33	4
0	1	0	–0.44	0.11	0	0.44	–0.11	2.67
1	0	0	0.33	–0.33	0	–0.33	0.33	4.0
0	0	–1	–0.67	–0.33	0	1.67	1.33	–4

LP Tableau Pivoting, Example, cont.

- Next pivot on column 3, row 1

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
0	0	**1**	0.67	0.33	1	–0.67	–0.33	4
0	1	0	–0.44	0.11	0	0.44	–0.11	2.67
1	0	0	0.33	–0.33	0	–0.33	0.33	4
0	0	–1	–0.67	–0.33	0	1.67	1.33	–4

- Which gives

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	
0	0	1	0.67	0.33	1	–0.67	–0.33	4
0	1	0	–0.44	0.11	0	0.44	–0.11	2.67
1	0	0	0.33	–0.33	0	–0.33	0.33	4
0	0	0	0	0	1	1	1	0

Since there are no negative relative costs we are done (with getting a starting solution)

LP Tableau Full Problem

- The tableau from the end of the artificial problem is used as the starting point for the actual solution
  - Remove the artificial variables
  - Update the relative costs with the costs from the original problem and update the bottom right-hand corner value

$$\mathbf{c} = [0.2 \quad 0.25 \quad 0 \quad 0 \quad 0]$$

$$\mathbf{r} = \mathbf{c}_N - \mathbf{c}_B \mathbf{A}_B^{-1} \mathbf{A}_N = \mathbf{c}_B \mathbf{A}_N$$

$$\mathbf{r} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}^T - [0 \quad 0.25 \quad 0.2] \begin{bmatrix} 0.67 & 0.33 \\ -0.44 & 0.11 \\ 0.33 & -0.33 \end{bmatrix} = \begin{bmatrix} 0.04 \\ 0.04 \end{bmatrix}^T$$

- Since none of the relative costs are negative we are done with x_1=4, x_2=2.7 and x_3=4

Marginal Costs of Constraint Enforcement in LP

If we would like to determine how the cost function will change for changes in **b**, assuming the set of basic variables does not change then we need to calculate

$$\frac{\partial z}{\partial \mathbf{b}} = \frac{\partial(\mathbf{c}_B \mathbf{x}_B)}{\partial \mathbf{b}} = \frac{\partial(\mathbf{c}_B \mathbf{A}_B^{-1}\mathbf{b})}{\partial \mathbf{b}} = \mathbf{c}_B \mathbf{A}_B^{-1} = \boldsymbol{\lambda}$$

So the values of $\boldsymbol{\lambda}$ tell the marginal cost of enforcing each constraint.

The marginal costs will be used to determine the OPF locational marginal costs (LMPs)

Nutrition Problem Marginal Costs

- In this problem we had basic variables 1, 2, 3; nonbasic variables of 4 and 5

$$\mathbf{x}_B = \mathbf{A}_B^{-1}\left(\mathbf{b} - \mathbf{A}_N\mathbf{x}_N\right) = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 3 & 0 \\ 4 & 3 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 20 \\ 12 \\ 24 \end{bmatrix} = \begin{bmatrix} 4 \\ 2.67 \\ 4 \end{bmatrix}$$

$$\boldsymbol{\lambda} = \mathbf{c}_B\mathbf{A}_B^{-1} = \begin{bmatrix} 0.2 & 0.25 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 \\ 1 & 3 & 0 \\ 4 & 3 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 \\ 0.044 \\ 0.039 \end{bmatrix}$$

There is no marginal cost with the first constraint since it is not binding; values tell how cost changes if the **b** values were changed

Lumber Mill Example Solution

Minimize $-(100x_1 + 120x_2)$

s.t. $2x_1 + 2x_2 + x_3 = 8$

$3x_1 + 5x_2 + x_4 = 15$

$x_1, x_2, x_3, x_4 \geq 0$

An initial basic feasible solution is $x_1 = 0, x_2 = 0, x_3 = 8, x_4 = 15$

The solution is $x_1 = 2.5, x_2 = 1.5, x_3 = 0, x_4 = 0$

Then $\boldsymbol{\lambda} = \begin{bmatrix} 100 & 120 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 3 & 5 \end{bmatrix}^{-1} = \begin{bmatrix} 35 \\ 10 \end{bmatrix}$

Economic interpretation of λ is the profit is increased by 35 for every hour we up the first constraint (the saw) and by 10 for every hour we up the second constraint (plane)

Complications

- Often variables are not limited to being ≥ 0
  - Variables with just a single limit can be handled by substitution; for example if x ≥ 5 then x-5=z ≥ 0
  - Bounded variables, high ≥ x ≥ 0 can be handled with a slack variable so x + y = high, and x,y ≥ 0
- Unbounded conditions need to be detected (i.e., unable to pivot); also the solution set could be null

Minimize $x_1 - x_2$ s.t. $x_1 + x_2 \geq 8$

$\rightarrow x_1 + x_2 - y_1 = 8 \rightarrow x_2 = 8$ is a basic feasible solution

x_1	x_2	y_1	
1	1	−1	8
2	0	−1	8

Complications

- Degenerate Solutions
  - Occur when there are less than m basic variables > 0
  - When this occurs the variable entering the basis could also have a value of zero; it is possible to cycle, anti-cycling techniques could be used
- Nonlinear cost functions
  - Nonlinear cost functions could be approximated by assuming a piecewise linear cost function
- Integer variables
  - Sometimes some variables must be integers; known as integer programming; we'll discuss after some power examples

LP Optimal Power Flow

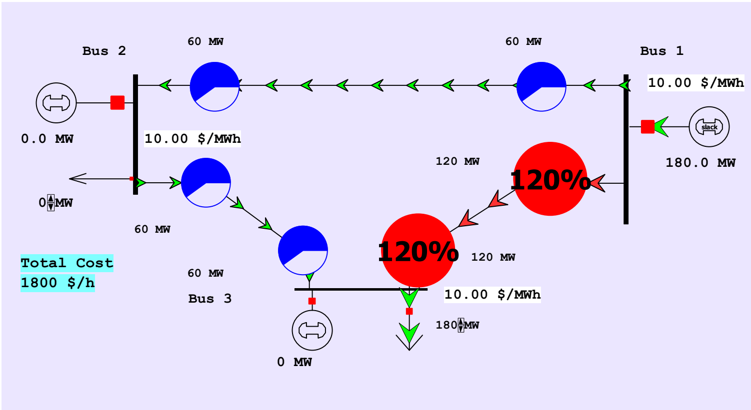
- LP OPF was introduced in
  - B. Stott, E. Hobson, "Power System Security Control Calculations using Linear Programming," (Parts 1 and 2) *IEEE Trans. Power App and Syst.*, Sept/Oct 1978
  - O. Alsac, J. Bright, M. Prais, B. Stott, "Further Developments in LP-based Optimal Power Flow," *IEEE Trans. Power Systems*, August 1990
- It is a widely used technique, particularly for real power optimization; it is the technique used in PowerWorld

LP Optimal Power Flow

- Idea is to iterate between solving the power flow, and solving an LP with just a selected number of constraints enforced
- The power flow (which could be ac or dc) enforces the standard power flow constraints
- The LP equality constraints include enforcing area interchange, while the inequality constraints include enforcing line limits; controls include changes in generator outputs
- LP results are transferred to the power flow, which is then resolved

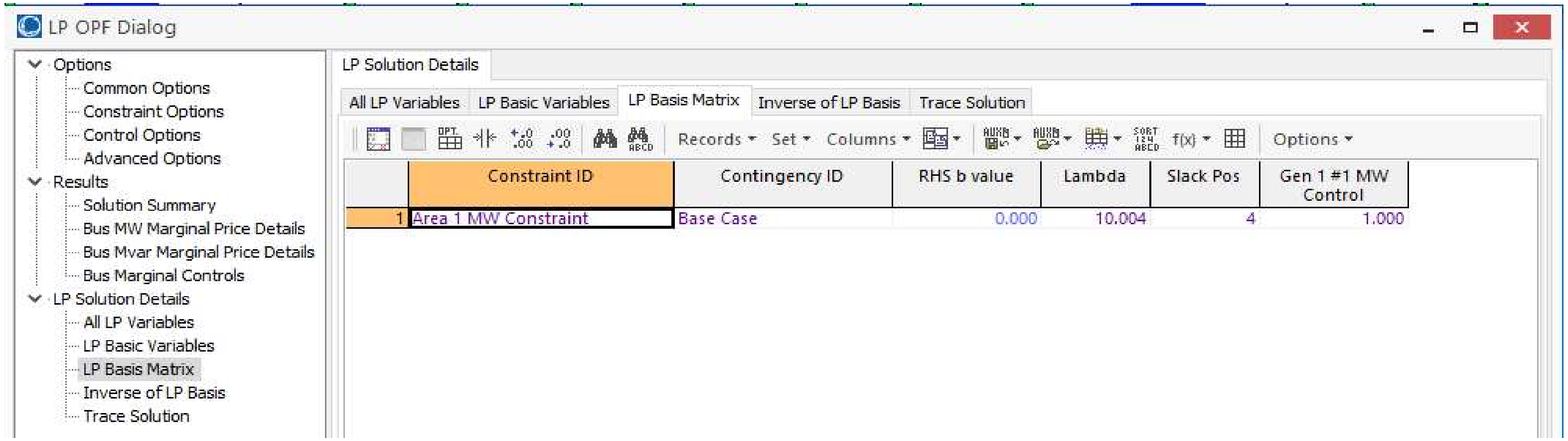
LP OPF Introductory Example

- In PowerWorld load the **B3LP** case and then display the LP OPF Dialog (select **Add-Ons, OPF Case Info, OPF Options and Results)**
- Use **Solve LP OPF** to solve the OPF, initially with no line limits enforced; this is similar to economic dispatch with a single power balance equality constraint

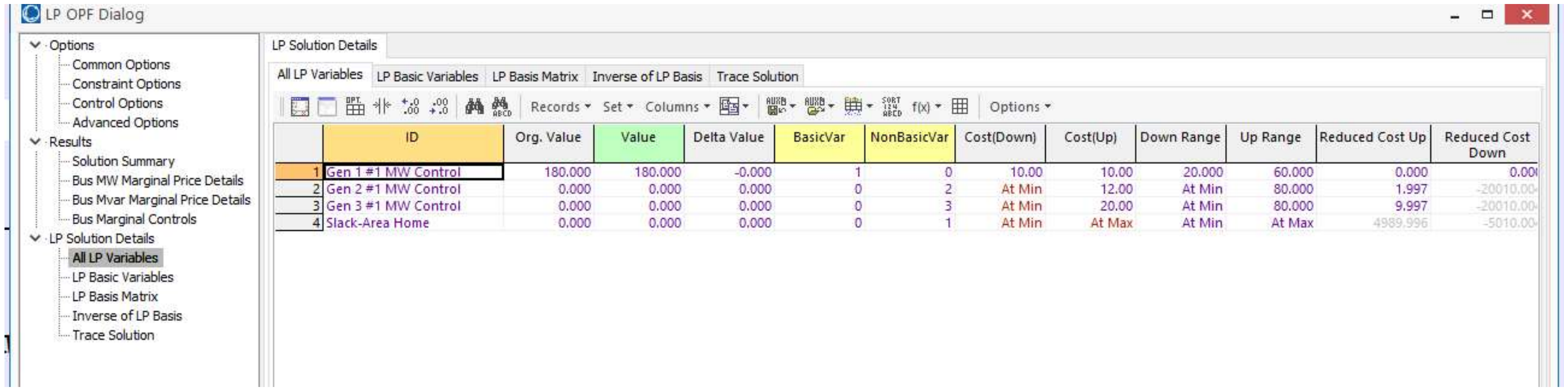


- The LP results are available from various pages on the dialog

LP OPF Introductory Example, cont



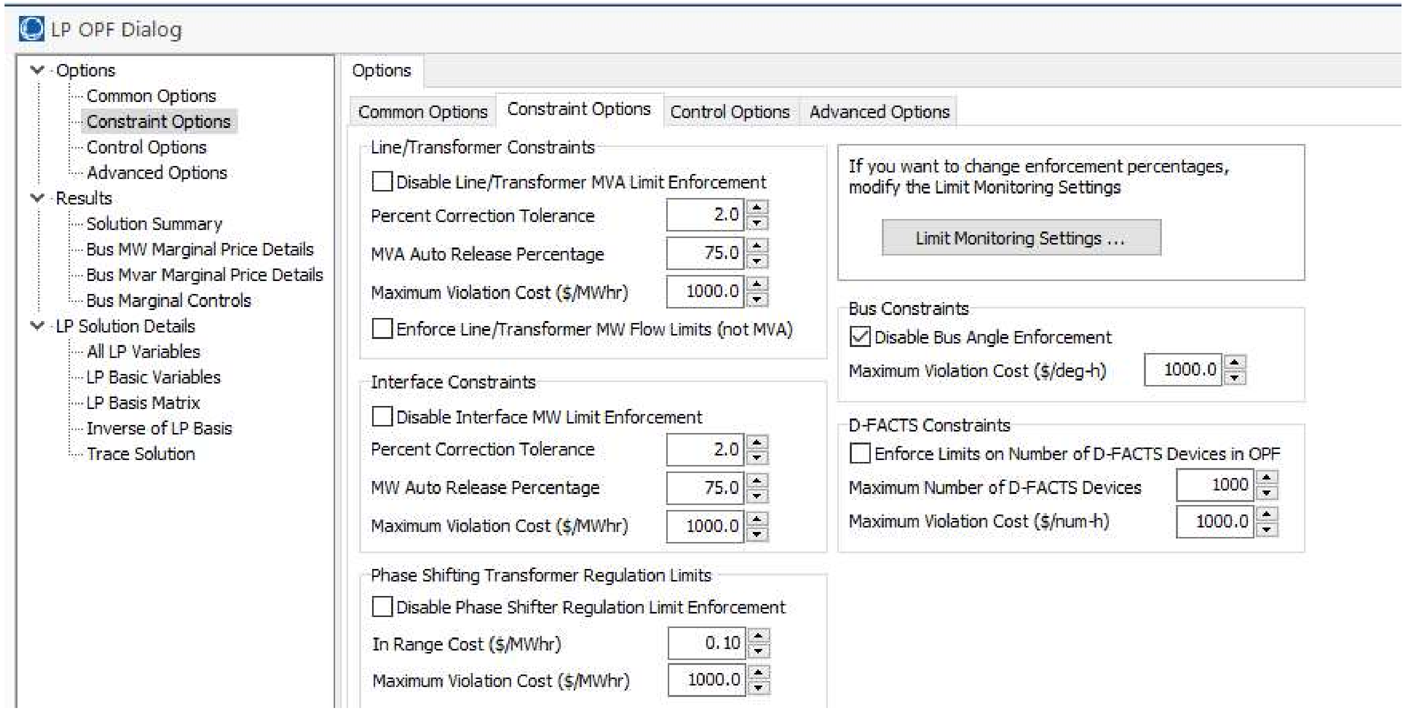
	Constraint ID	Contingency ID	RHS b value	Lambda	Slack Pos	Gen 1 #1 MW Control
1	Area 1 MW Constraint	Base Case	0.000	10.004	4	1.000



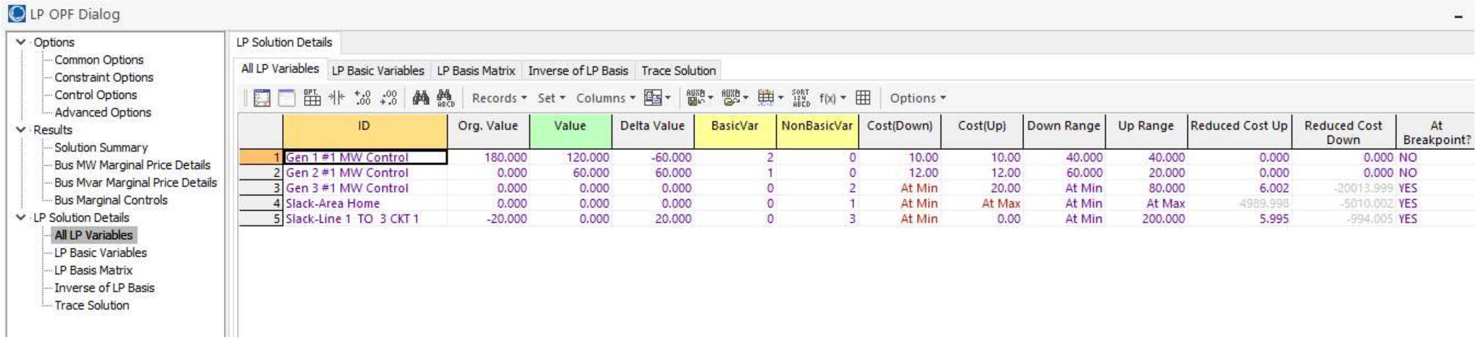
	ID	Org. Value	Value	Delta Value	BasicVar	NonBasicVar	Cost(Down)	Cost(Up)	Down Range	Up Range	Reduced Cost Up	Reduced Cost Down
1	Gen 1 #1 MW Control	180.000	180.000	-0.000	1	0	10.00	10.00	20.000	60.000	0.000	0.00
2	Gen 2 #1 MW Control	0.000	0.000	0.000	0	2	At Min	12.00	At Min	80.000	1.997	-20010.00
3	Gen 3 #1 MW Control	0.000	0.000	0.000	0	3	At Min	20.00	At Min	80.000	9.997	-20010.00
4	Slack-Area Home	0.000	0.000	0.000	0	1	At Min	At Max	At Min	At Max	4989.996	-5010.00

LP OPF Introductory Example, cont

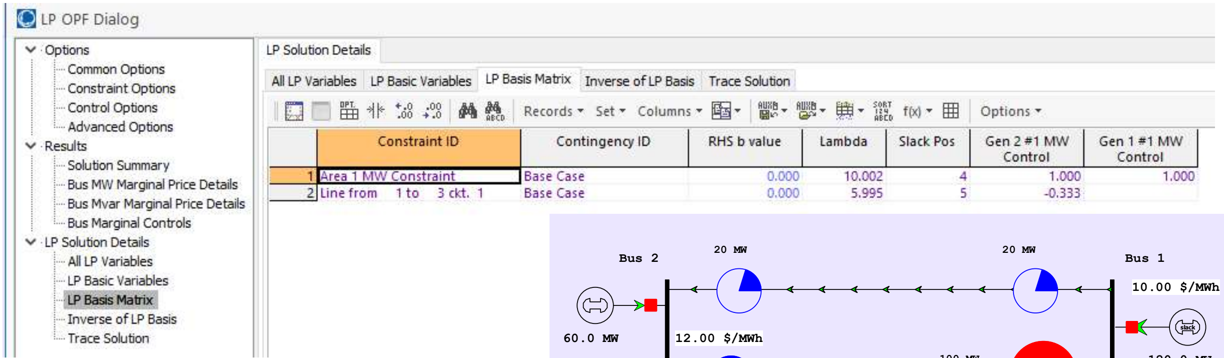
- On use **Options, Constraint Options** to enable the enforcement of the Line/Transformer MVA limits



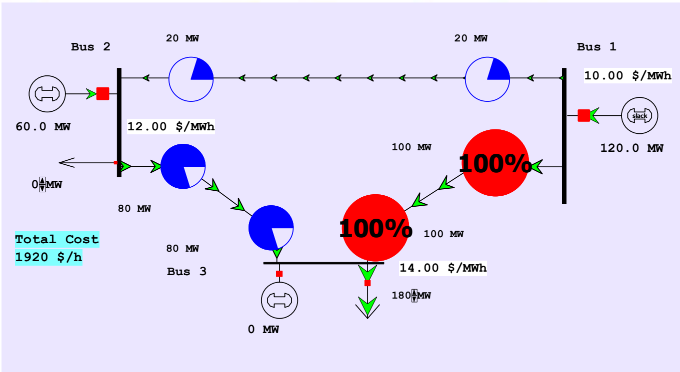
LP OPF Introductory Example, cont.



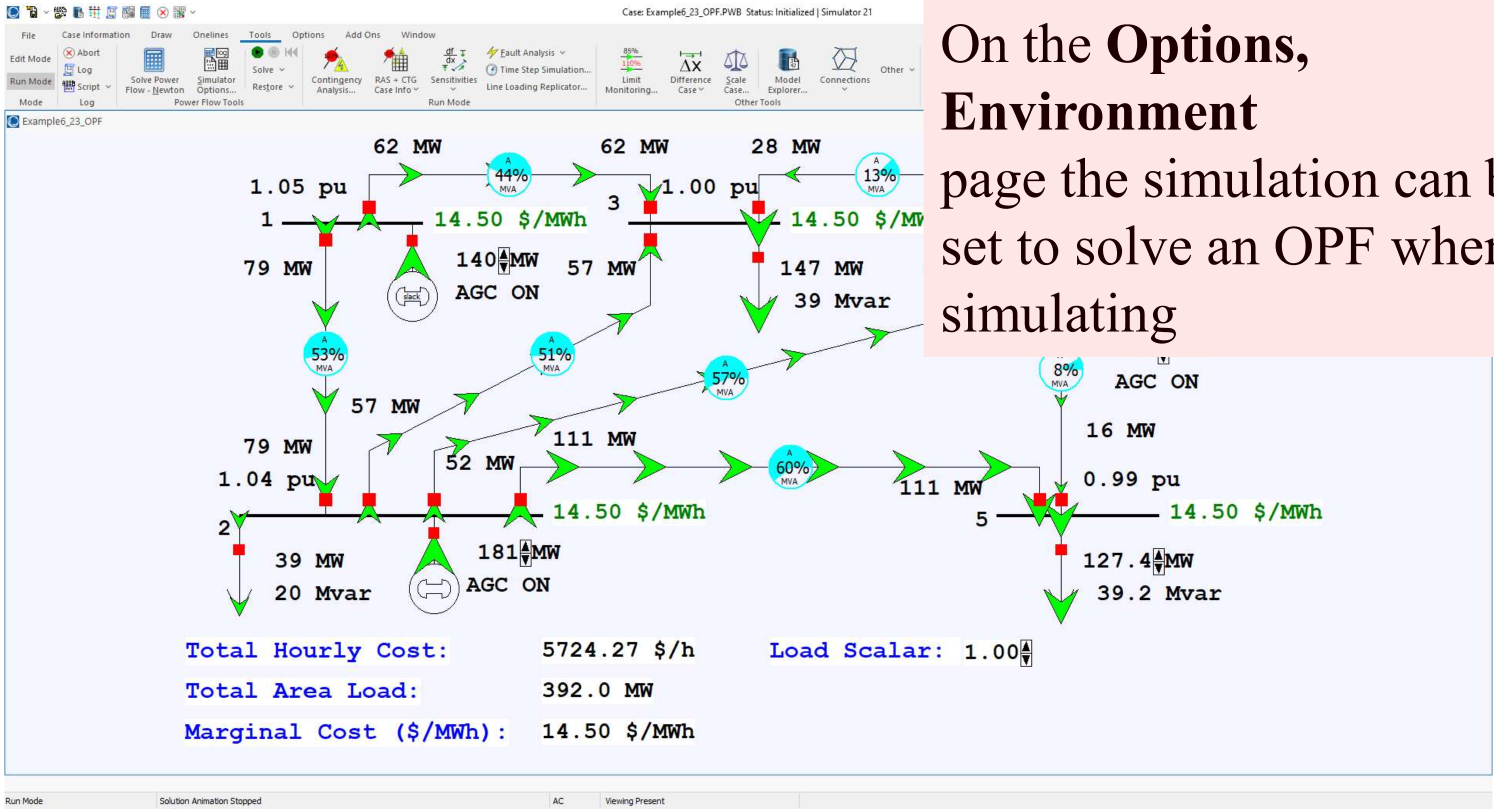
	ID	Org. Value	Value	Delta Value	BasicVar	NonBasicVar	Cost(Down)	Cost(Up)	Down Range	Up Range	Reduced Cost Up	Reduced Cost Down	At Breakpoint?
1	Gen 1 #1 MW Control	180.000	120.000	-60.000	2	0	10.00	10.00	40.000	40.000	0.000	0.000	NO
2	Gen 2 #1 MW Control	0.000	60.000	60.000	1	0	12.00	12.00	60.000	20.000	0.000	0.000	NO
3	Gen 3 #1 MW Control	0.000	0.000	0.000	0	2	At Min	20.00	At Min	80.000	6.002	-20013.999	YES
4	Slack-Area Home	0.000	0.000	0.000	0	1	At Min	At Max	At Min	At Max	4989.998	-5010.002	YES
5	Slack-Line 1 TO 3 CKT 1	-20.000	0.000	20.000	0	3	At Min	0.00	At Min	200.000	5.995	-994.005	YES



	Constraint ID	Contingency ID	RHS b value	Lambda	Slack Pos	Gen 2 #1 MW Control	Gen 1 #1 MW Control
1	Area 1 MW Constraint	Base Case	0.000	10.002	4	1.000	1.000
2	Line from 1 to 3 ckt. 1	Base Case	0.000	5.995	5	-0.333	



Example 6_23 Optimal Power Flow



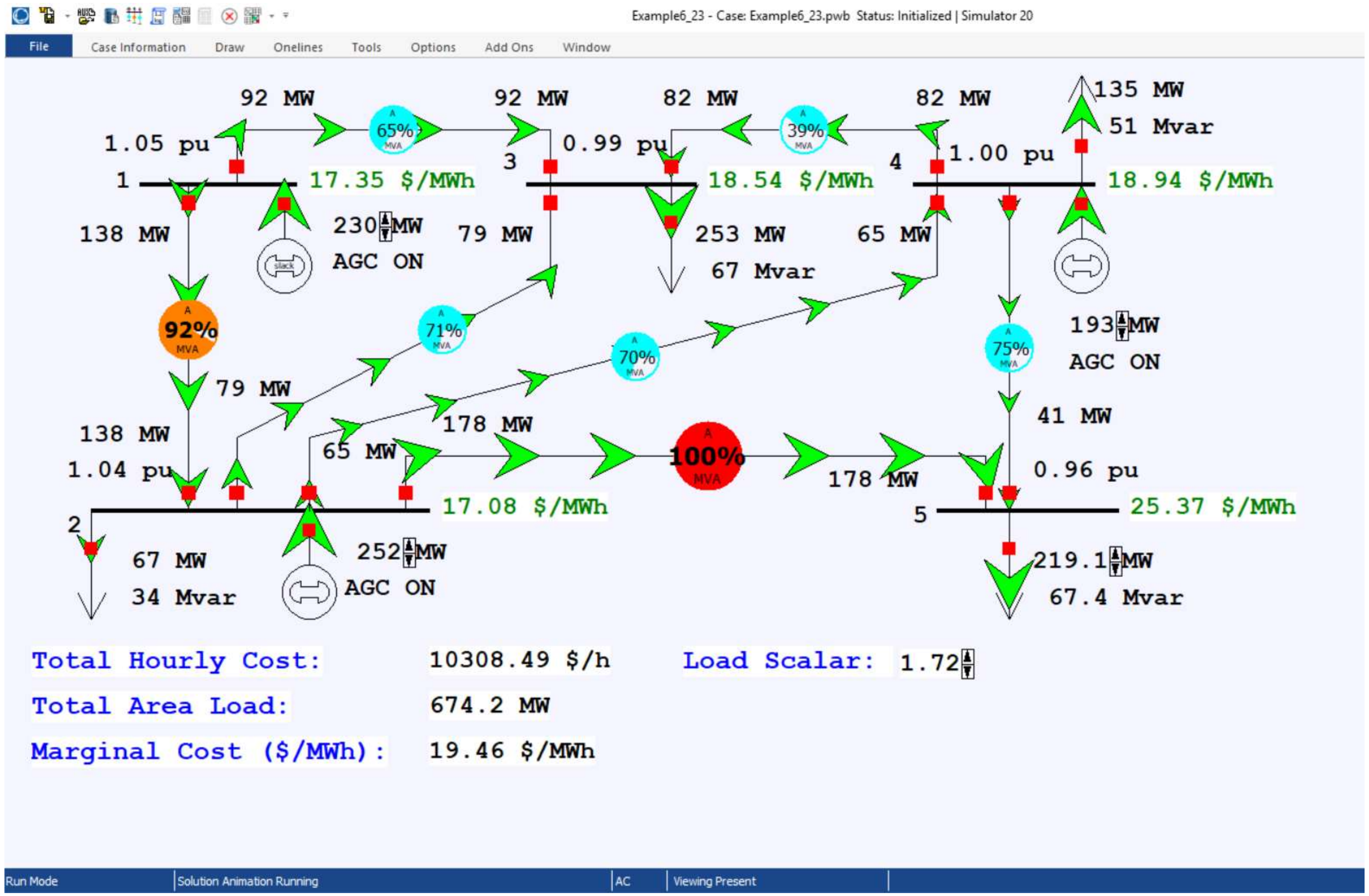
On the **Options, Environment** page the simulation can be set to solve an OPF when simulating

Open the case **Example6_23_OPF.** In this example the load is gradually increased

Locational Marginal Costs (LMPs)

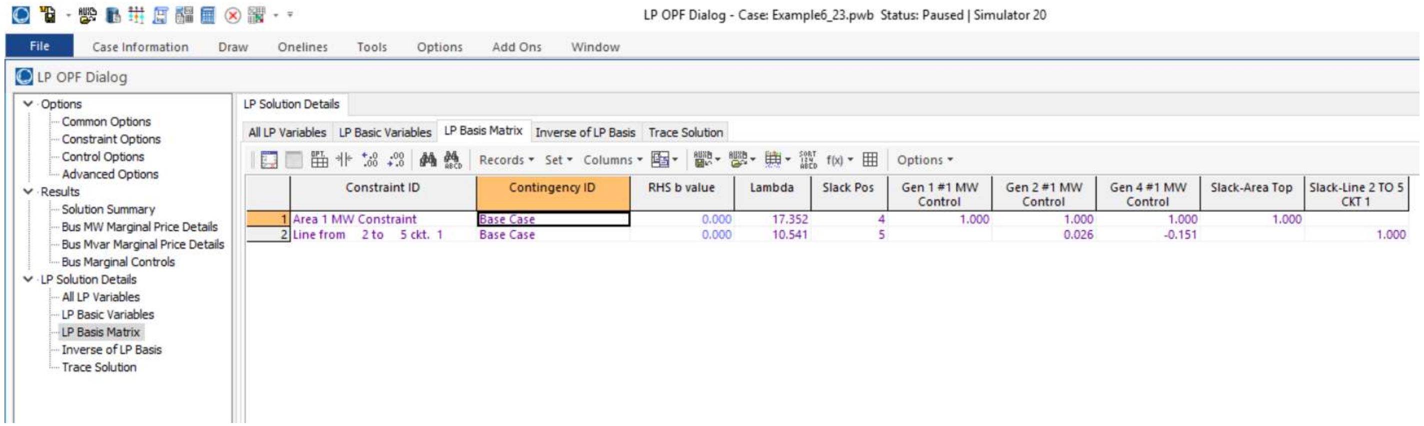
- In an OPF solution, the bus LMPs tell the marginal cost of supplying electricity to that bus
- The term "congestion" is used to indicate when there are elements (such as transmission lines or transformers) that are at their limits; that is, the constraint is binding
- Without losses and without congestion, all the LMPs would be the same
- Congestion or losses causes unequal LMPs
- LMPs are often shown using color contours; a challenge is to select the right color range!

Example 6_23 Optimal Power Flow with Load Scale = 1.72



Example 6_23 Optimal Power Flow with Load Scale = 1.72

- LP Sensitivity Matrix (**A** Matrix)



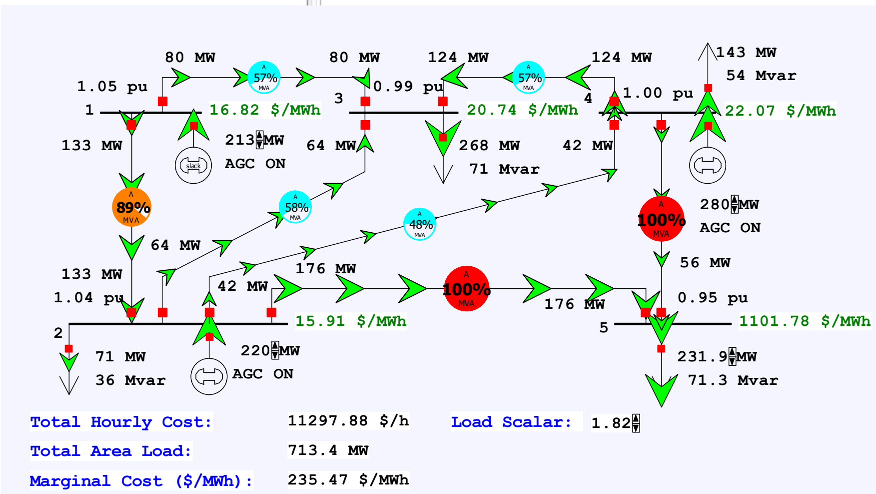
	Constraint ID	Contingency ID	RHS b value	Lambda	Slack Pos	Gen 1 #1 MW Control	Gen 2 #1 MW Control	Gen 4 #1 MW Control	Slack-Area Top	Slack-Line 2 TO 5 CKT 1
1	Area 1 MW Constraint	Base Case	0.000	17.352	4	1.000	1.000	1.000	1.000	
2	Line from 2 to 5 ckt. 1	Base Case	0.000	10.541	5		0.026	-0.151		1.000

The first row is the power balance constraint, while the second row is the line flow constraint. The matrix only has the line flows that are being enforced.

Example 6_23 Optimal Power Flow with Load Scale = 1.82

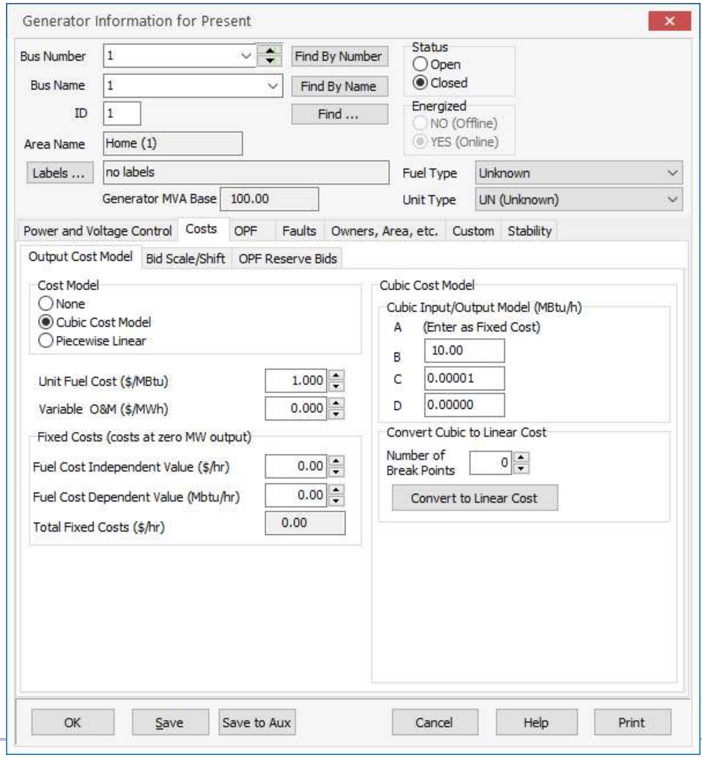
- This situation is infeasible, at least with available controls. There is a solution because the OPF is allowing one of the constraints to violate (at high cost)

						Control	Control	Control		CKT 1
1	Area 1 MW Constraint	Base Case	0.000	16.824	4	1.000	1.000	1.000	1.000	
2	Line from 2 to 5 ckt. 1	Base Case	0.000	993.664	5		0.026	-0.146		1.000
3	Line from 4 to 5 ckt. 1	Base Case	-0.002	1000.000	6		-0.024	0.140		



Generator Cost Curve Modeling

- LP algorithms require linear cost curves, with piecewise linear curves used to approximate a nonlinear cost function
- Two common ways of entering cost information are
  - Quadratic function
  - Piecewise linear curve
- The PowerWorld OPF supports both types



Security Constrained OPF

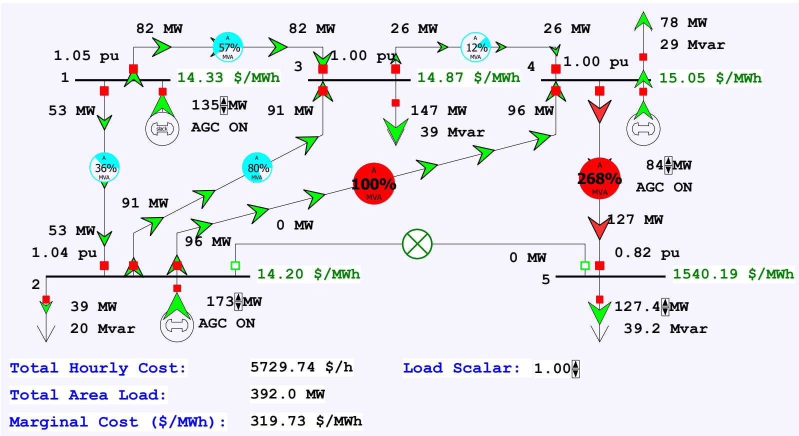
- Security constrained optimal power flow (SCOPF) is similar to OPF except it also includes contingency constraints
  - Again the goal is to minimize some objective function, usually the current system cost, subject to a variety of equality and inequality constraints
  - This adds significantly more computation, but is required to simulate how the system is actually operated (with N-1 reliability)
- A common solution is to alternate between solving a power flow and contingency analysis, and an LP

Security Constrained OPF, cont.

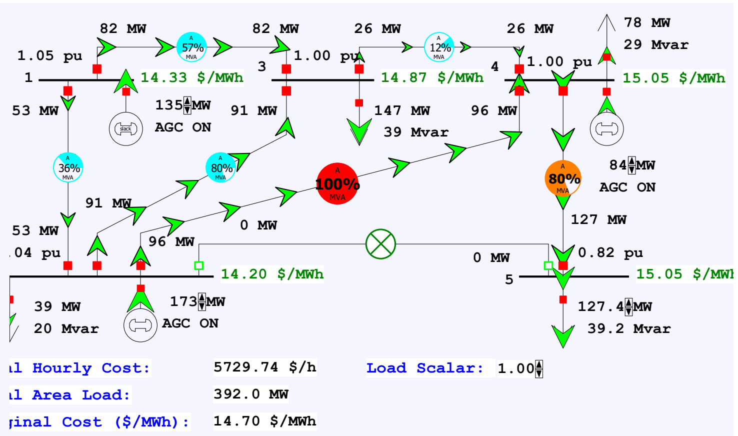
- With the inclusion of contingencies, there needs to be a distinction between what control actions must be done pre-contingent, and which ones can be done post-contingent
  - The advantage of post-contingent control actions is they would only need to be done in the unlikely event the contingency actually occurs
- Pre-contingent control actions are usually done for line overloads, while post-contingent control actions are done for most reactive power control and generator outage re-dispatch

SCOPF Example

- We’ll again consider Example 6_23, except now it has been enhanced to include contingencies and we’ve also greatly increased the capacity on the line between buses 4 and 5; named Bus5_SCOPF_DC



Original with line 4-5 limit of 60 MW with 2-5 out

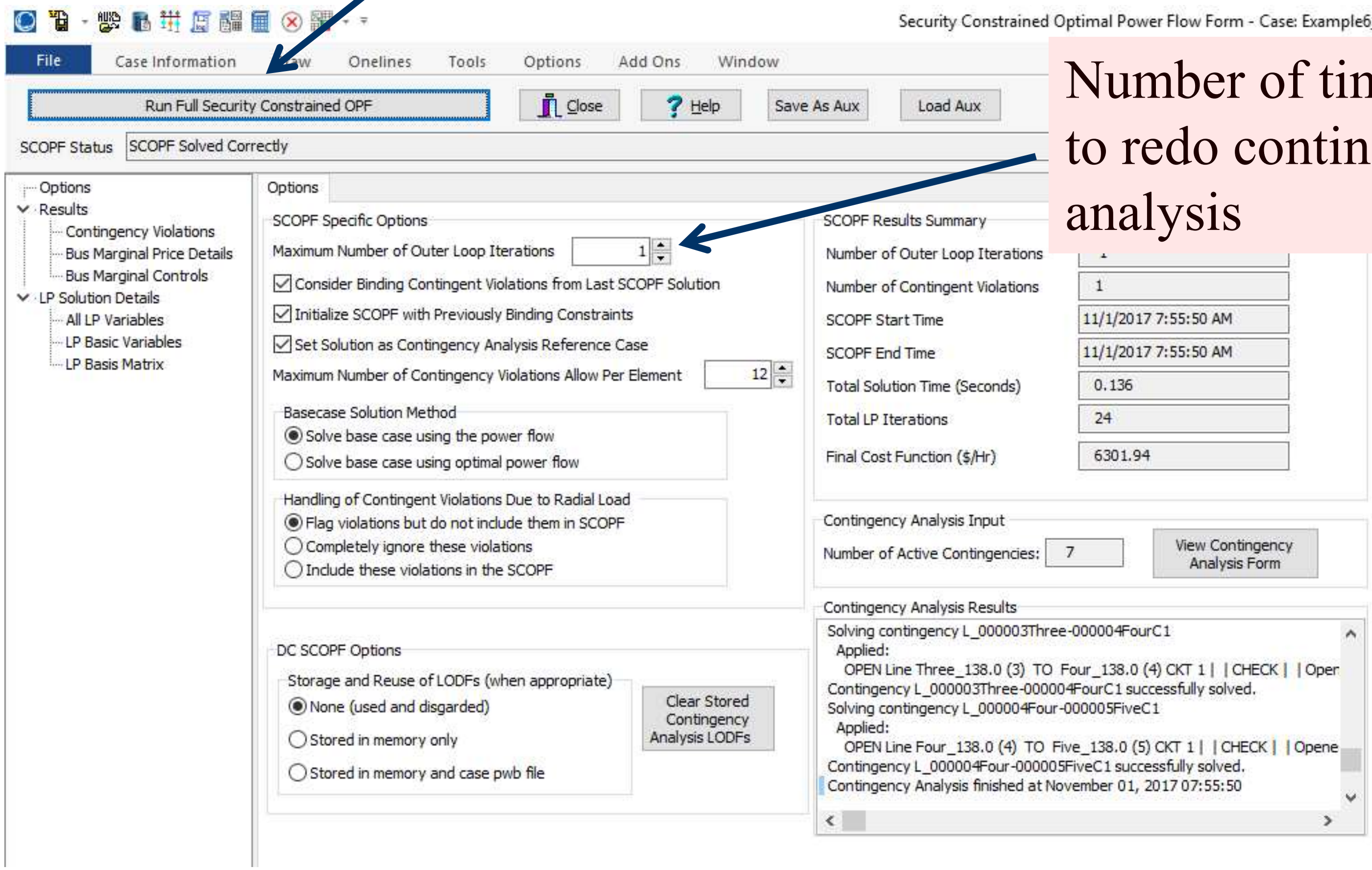


Modified with line 4-5 limit of 200 MVA with 2-5 out

PowerWorld SCOPF Application

Just click the button to solve

Number of times to redo contingency analysis



LP OPF and SCOPF Issues

- The LP approach is widely used for the OPF and SCOPF, particularly when implementing a dc power flow approach
- A key issue is determining the number of binding constraints to enforce in the LP tableau
  - Enforcing too many is time-consuming, enforcing too few results in excessive iterations
- The LP approach is limited by the degree of linearity in the power system
  - Real power constraints are fairly linear, reactive power constraints much less so

OPF Solution by Newton's Method

- An alternative to using the LP approach is to use Newton's method, in which all the equations are solved simultaneously
- Key paper in area is
  - D.I. Sun, B. Ashley, B. Brewer, B.A. Hughes, and W.F. Tinney, "Optimal Power Flow by Newton Approach", *IEEE Trans. Power App and Syst.*, October 1984
- Problem is

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{s.t.} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ &\qquad\quad \mathbf{h}(\mathbf{x}) \le \mathbf{0} \end{aligned}$$

For simplicity **x** represents all the variables and we can use **h** to impose limits on individual variables

OPF Solution by Newton's Method

- During the solution the inequality constraints are either binding (=0) or nonbinding (<0)
  - The nonbinding constraints do not impact the final solution
- We'll modify the problem to split the **h** vector into the binding constraints, $\mathbf{h}_1$ and the nonbinding constraints, $\mathbf{h}_2$

$$
\begin{aligned}
&\text{Minimize } f(\mathbf{x}) \\
&\text{s.t.} \qquad \mathbf{g}(\mathbf{x}) = \mathbf{0} \\
&\qquad\qquad \mathbf{h}_1(\mathbf{x}) = \mathbf{0} \\
&\qquad\qquad \mathbf{h}_2(\mathbf{x}) < \mathbf{0}
\end{aligned}
$$

OPF Solution by Newton's Method

- To solve first define the Lagrangian

$$L(\mathbf{x},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) = f(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}_1(\mathbf{x})$$

Let $\mathbf{z} = \begin{bmatrix} \mathbf{x} & \boldsymbol{\mu} & \boldsymbol{\lambda} \end{bmatrix}$

- A necessary condition for a minimum is that the gradient is zero

$$\nabla L(\mathbf{z}) = \mathbf{0} = \begin{bmatrix} \dfrac{\partial L(\mathbf{z})}{\partial z_1} \\ \dfrac{\partial L(\mathbf{z})}{\partial z_2} \\ \vdots \end{bmatrix}$$

Both μ and λ are Lagrange Multipliers

OPF Solution by Newton's Method

- Solve using Newton's method. To do this we need to define the Hessian matrix

$$\nabla^2 L(\mathbf{z}) = \mathbf{H}(\mathbf{z}) = \left[\frac{\partial^2 L(\mathbf{z})}{\partial z_i \partial z}\right] = \begin{bmatrix} \frac{\partial^2 L(\mathbf{z})}{\partial x_i \partial x_j} & \frac{\partial^2 L(\mathbf{z})}{\partial x_i \partial \mu_j} & \frac{\partial^2 L(\mathbf{z})}{\partial x_i \partial \lambda_j} \\ \frac{\partial^2 L(\mathbf{z})}{\partial \mu_i \partial x_j} & \mathbf{0} & \mathbf{0} \\ \frac{\partial^2 L(\mathbf{z})}{\partial \lambda \partial x_{ji}} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

- Because this is a second order method, as opposed to a first order linearization, it can better handle system nonlinearities

OPF Solution by Newton's Method

- Solution is then via the standard Newton's method. That is

Set iteration counter k=0, set k_{max}

Set convergence tolerance ε

Guess $\mathbf{z}^{(k)}$

While $\left(\left\|\nabla L(\mathbf{z})\right\| \geq \varepsilon\right)$ and $\left(k < k_{max}\right)$

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \left[\mathbf{H}(\mathbf{z})\right]^{-1} \nabla L(\mathbf{z})$$

$k = k + 1$

End While

No iteration is needed for a quadratic function with linear constraints

Example

- Solve

Minimize $x_1^2 + x_2^2$ such that $3x_1 + x_2 - 2 \geq 0$

Solve initially assuming the constraint is binding

$$L(\mathbf{x},\lambda) = x_1^2 + x_2^2 + \lambda(3x_1 + x_2 - 2)$$

$$\nabla L(\mathbf{x},\lambda) = \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \frac{\partial L}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} 2x_1 + 3\lambda \\ 2x_2 + \lambda \\ 3x_1 + x_2 - 2 \end{bmatrix}$$

No iteration is needed so any "guess" is fine. Pick (1,1,0)

$$\nabla^2 L(\mathbf{x},\lambda) = \mathbf{H}(\mathbf{x},\lambda) = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 3 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.4 \end{bmatrix}$$

Because λ is positive the constraint is binding

Newton OPF Comments

- The Newton OPF has the advantage of being better able to handle system nonlinearities
- There is still the issue of having to deal with determining which constraints are binding
- The Newton OPF needs to implement second order derivatives plus all the complexities of the power flow solution
  - The power flow starts off simple, but can rapidly get complex when dealing with actual systems
- There is still the issue of handling integer variables

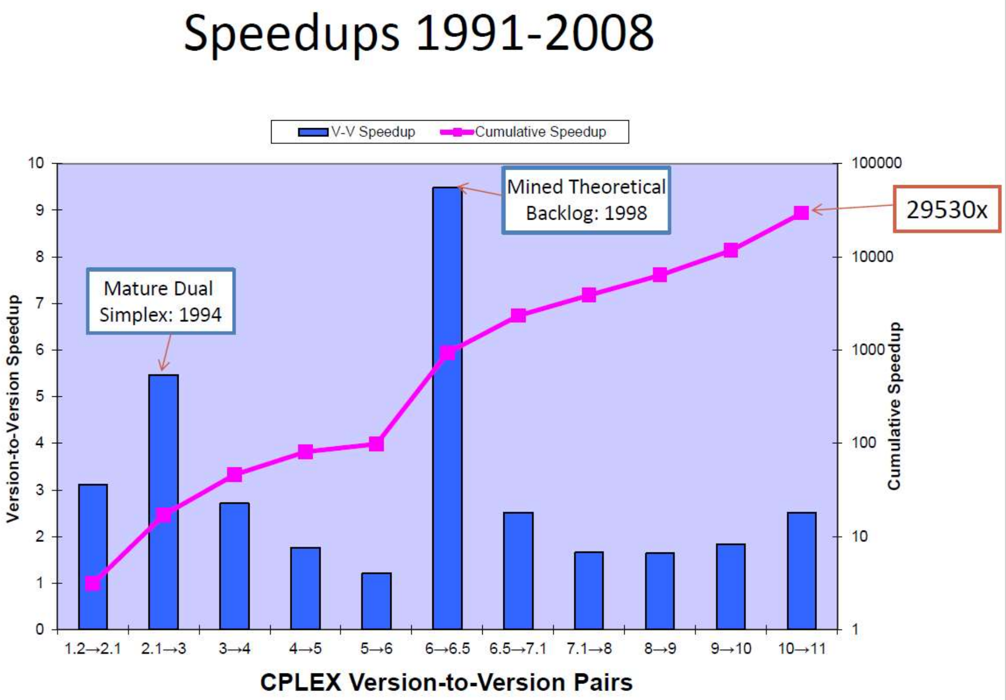
Mixed-Integer Programming

- A mixed-integer program (MIP) is an optimization problem of the form

$$
\begin{array}{ll}
\text{Minimize} & \mathbf{c}\,\mathbf{x} \\
\text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0} \\
\text{where} & \mathbf{x} = \text{n-dimensional column vector} \\
 & \mathbf{c} = \text{n-dimensional row vector} \\
 & \mathbf{b} = \text{m-dimensional column vector} \\
 & \mathbf{A} = \text{m}\times\text{n matrix} \\
 & \text{some or all } x_j \text{ integer}
\end{array}
$$

Mixed-Integer Programming

- The advances in the algorithms have been substantial



Speedups from 2009 to 2015 were about a factor of 30

Notes are partially based on a presentation at Feb 2015 US National Academies Analytic Foundations of the Next Generation Grid by Robert Bixby from Gurobi Optimization titled "Advances in Mixed-Integer Programming and the Impact on Managing Electrical Power Grids"

Mixed-Integer Programming

- Suppose you were given the following choices?
  - Solve a MIP with today's solution technology on a 1991 machine
  - Solve a MIP with a 1991 solution on a machine from today?
- The answer is to choose option 1, by a factor of approximately 300
- This leads to the current debate of whether the OPF (and SCOPF) should be solved using generic solvers or more customized code (which could also have quite good solvers!)

Notes are partially based on a presentation at Feb 2015 US National Academies Analytic Foundations of the Next Generation Grid by Robert Bixby from Gurobi Optimization titled "Advances in Mixed-Integer Programming and the Impact on Managing Electrical Power Grids"

More General Solvers Overview

- OPF is currently an area of active research
- Many formulations and solution methods exist…
  - As do many ***tools*** for highly complex, large-scale computing!

- While many options exist, some may work better for certain problems or with certain programs you already use

- Consider experimenting with a new language/solver!

Gurobi and CPLEX

- Gurobi and CPLEX are two well-known commercial optimization solvers/packages for linear programming (LP), quadratic programming (QP), quadratically constrained programming (QCP), and the mixed integer (MI) counterparts of LP/QP/QCP
- Gurobi and CPLEX are accessible through object-oriented interfaces (C++, Java, Python, C), matrix-oriented interfaces (MATLAB) and other modeling languages (AMPL, GAMS)

Solver Comparison

Algorithm Type ----------------- *Solver*	LP/MILP linear/mixed integer linear program	QP/MIQP quadratic/mixed integer quadratic program	SOCP second order cone program	SDP semidefinite program
*CPLEX**	x	x	x	
GLPK	x			
*Gurobi**	x	x	x	
IPOPT		x		
*Mosek**	x	x	x	x
SDPT3/SeDuMi			x	x

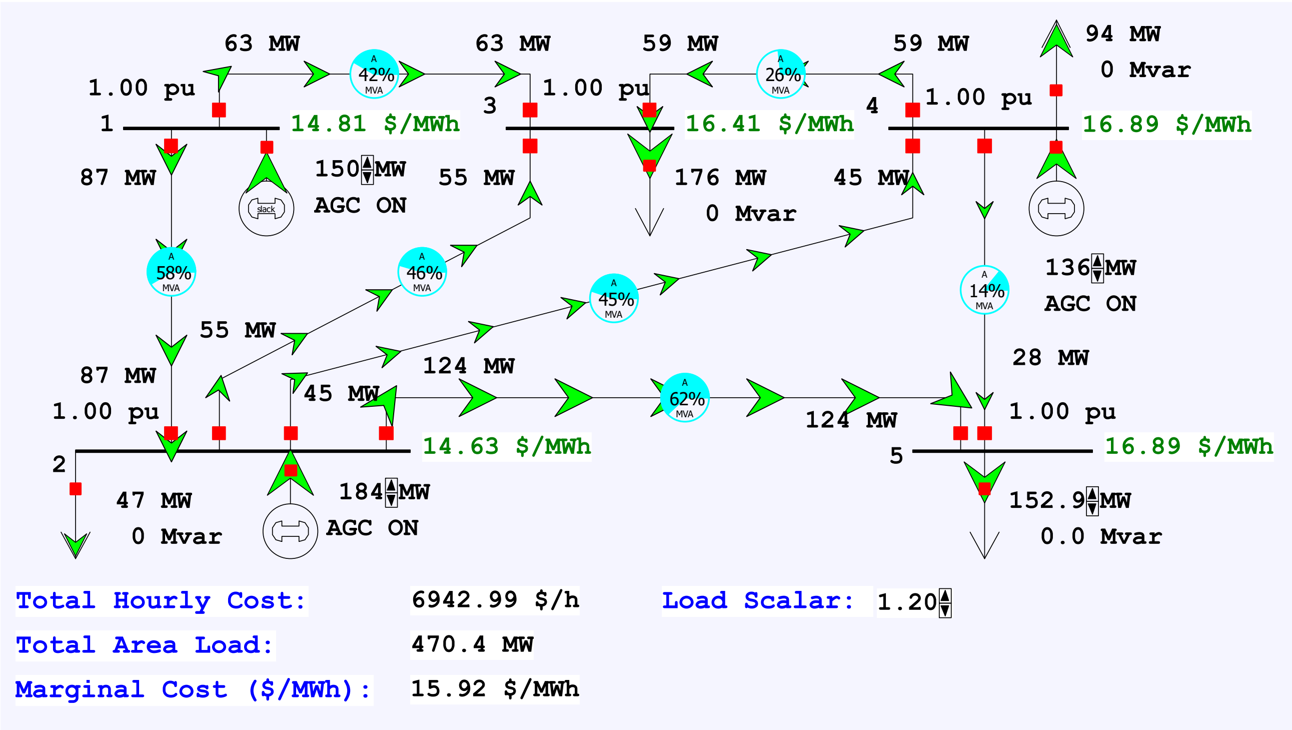
Linear programming can be solved by quadratic programming, which can be solved by second-order cone programming, which can be solved by semidefinite programming.

DC OPF and SCOPF

- Solving a full ac OPF or SCOPF on a large system is difficult, so most electricity markets actually use the more approximate, but much simpler DCOPF, in which a dc power flow is used
- PowerWorld includes this option in the **Options, Power Flow Solution, DC Options**

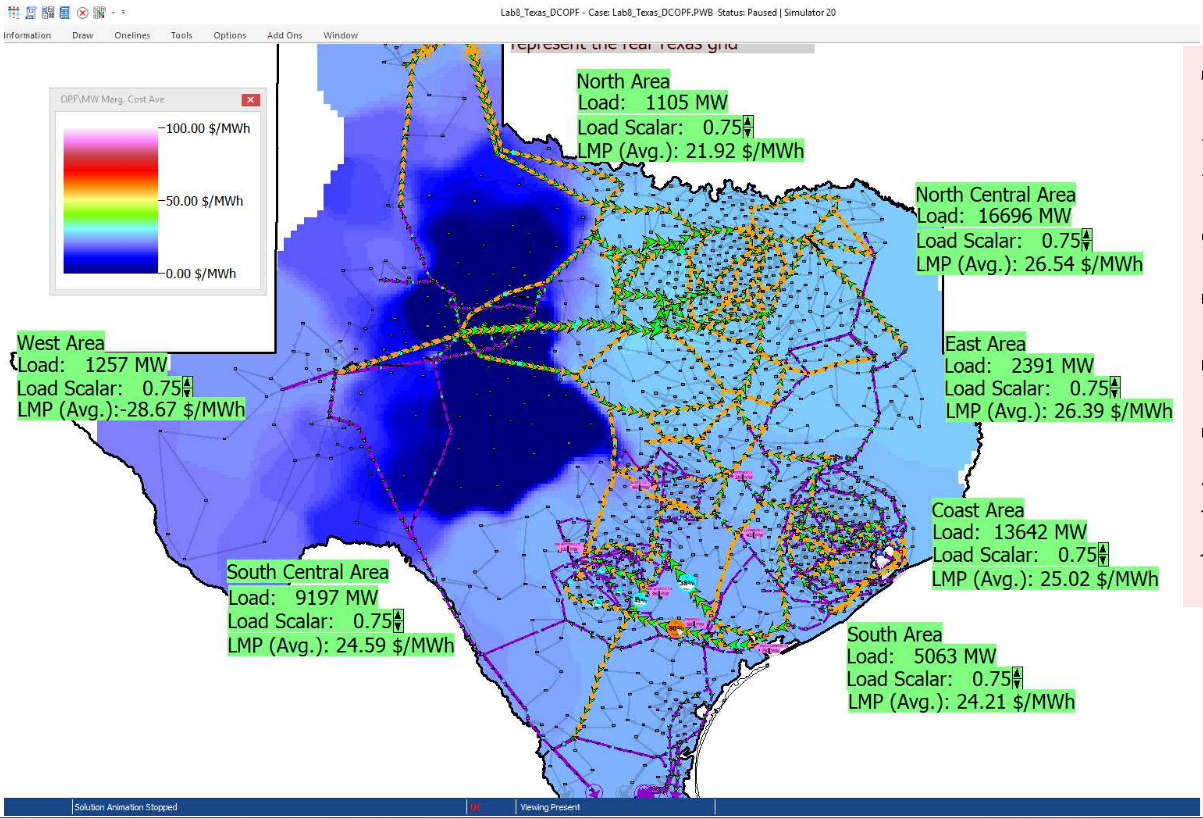
Example 6_13 DC SCOPF Results: Load Scalar at 1.20

- Now there is not an unenforceable constraint on the line between 4-5 (for the line 2-5 contingency) because the reactive losses are ignored



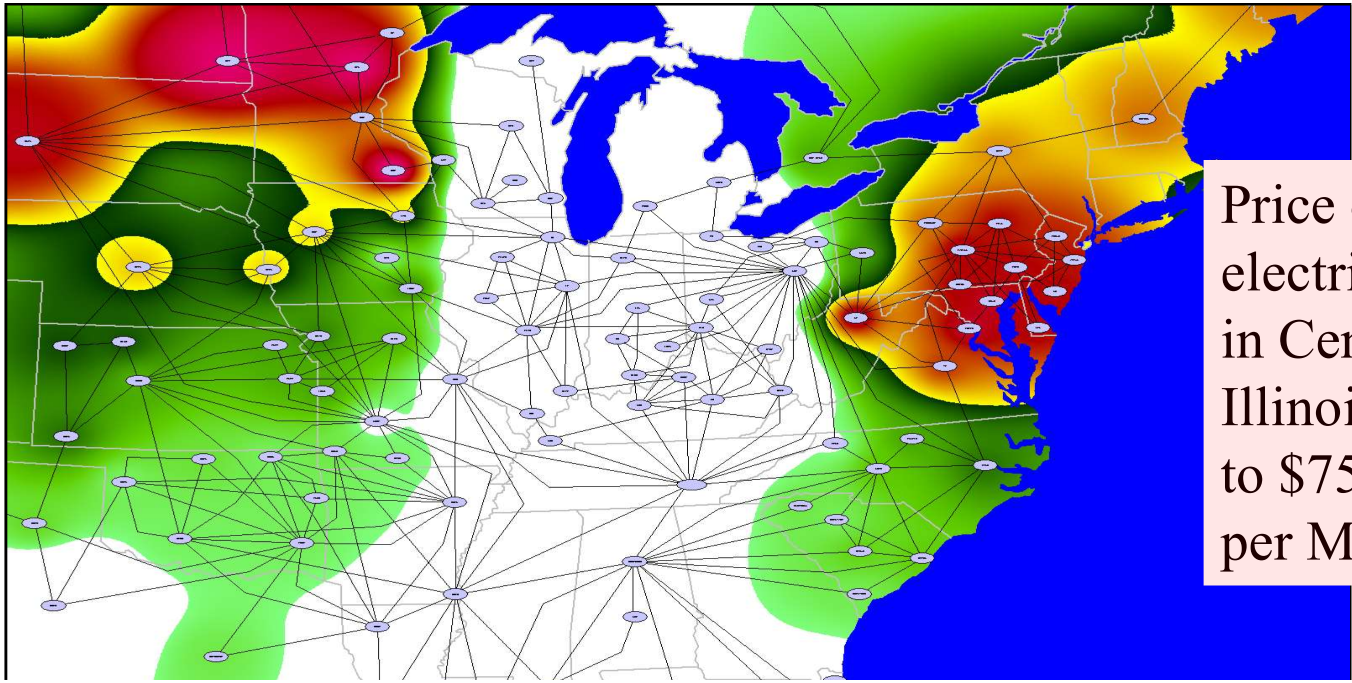
2000 Bus Texas Synthetic DC OPF Example

- This system does a DC OPF solution, with the ability to change the load in the areas



The quite low LMPs are actually due to a constraint on a single 230/115 kV transformer

June 1998 Heat Storm: Two Constraints Caused a Price Spike



Price of electricity in Central Illinois went to $7500 per MWh!

Colored areas could NOT sell into Midwest because of constraints on a line in Northern Wisconsin and on a Transformer in Ohio