ECEN 615
Methods of Electric Power Systems Analysis

Lecture 22: LP, OPF, Electricity Markets

Prof. Tom Overbye

Dept. of Electrical and Computer Engineering

Texas A&M University

overbye@tamu.edu

Announcements

- Read Chapter 8 and Appendices 3B and 3E of Chapter 3
- Homeworks 6 and 7 are assigned today, with Homework 6 due on Nov 12 and Homework 7 by Nov 24
- The second exam will be in class on Nov 17
  - Distance learners will be able to take the exam from Nov 16 to Nov 18
- Associated with Homework 7 will be student presentations; these will be about 15 minutes during class on Nov 19 or Nov 24
  - Other times can be arranged for the distance learners

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)

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

LP Standard Form

The standard form of the LP problem is

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

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

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

Maximum problems can be treated as minimizing the negative

where $\quad \mathbf{x}$ = 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!

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 $\boldsymbol{\lambda}$ 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

$$\text{Minimize} \quad x_1 - x_2 \text{ s.t. } x_1 + x_2 \geq 8$$

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

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

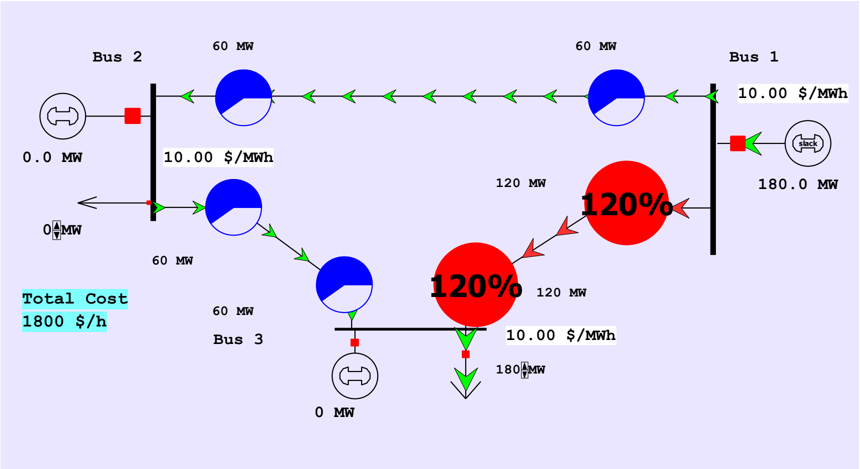
- LP OPF was introduced in
  - B. Stott, E. Hobson, "Power System Security Control Calculations using Linear Programming," (Parts 1 and 2) *IEEE Trans. Power App and Syst.*, Sept/Oct 1978
  - O. Alsac, J. Bright, M. Prais, B. Stott, "Further Developments in LP-based Optimal Power Flow," *IEEE Trans. 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

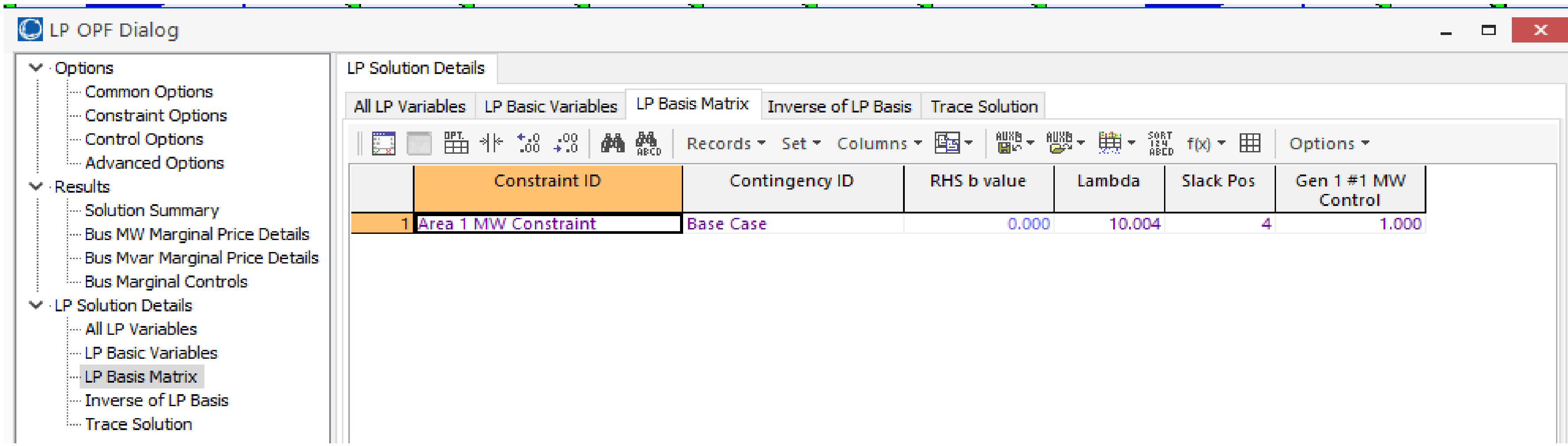
LP OPF Introductory Example

- 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

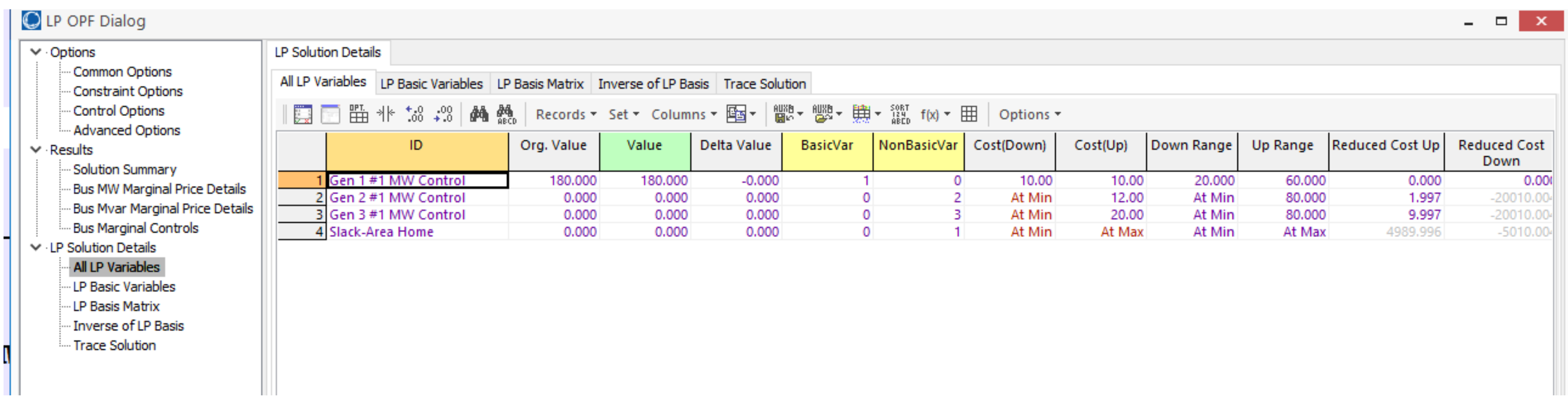


- The LP results are available from various pages on the dialog

LP OPF Introductory Example, cont



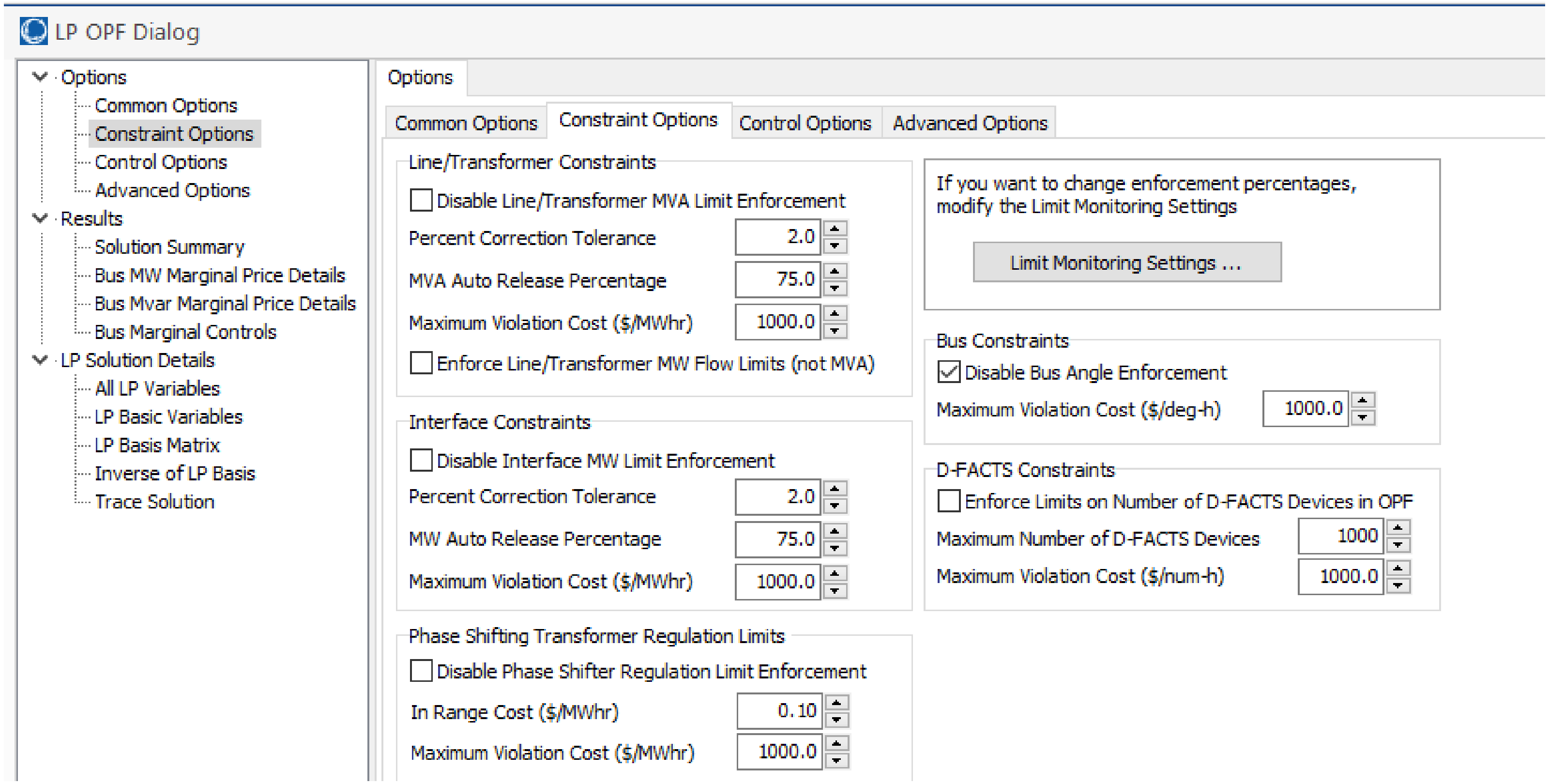
	Constraint ID	Contingency ID	RHS b value	Lambda	Slack Pos	Gen 1 #1 MW Control
1	Area 1 MW Constraint	Base Case	0.000	10.004	4	1.000



	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.

LP OPF Dialog

- Options
  - Common Options
  - Constraint Options
  - Control Options
  - Advanced Options
- Results
  - Solution Summary
  - Bus MW Marginal Price Details
  - Bus Mvar Marginal Price Details
  - Bus Marginal Controls
- LP Solution Details
  - All LP Variables
  - LP Basic Variables
  - LP Basis Matrix
  - Inverse of LP Basis
  - Trace Solution

LP Solution Details

All LP Variables | LP Basic Variables | LP Basis Matrix | Inverse of LP Basis | Trace Solution

Records ▾ Set ▾ Columns ▾ f(x) ▾ Options ▾

	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

LP OPF Dialog

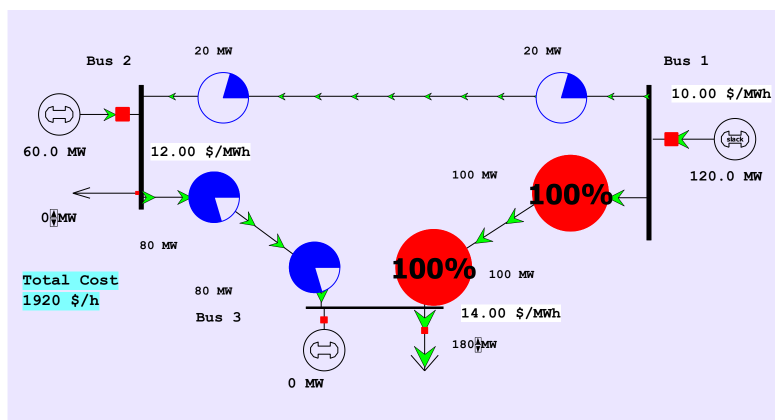
- Options
  - Common Options
  - Constraint Options
  - Control Options
  - Advanced Options
- Results
  - Solution Summary
  - Bus MW Marginal Price Details
  - Bus Mvar Marginal Price Details
  - Bus Marginal Controls
- LP Solution Details
  - All LP Variables
  - LP Basic Variables
  - LP Basis Matrix
  - Inverse of LP Basis
  - Trace Solution

LP Solution Details

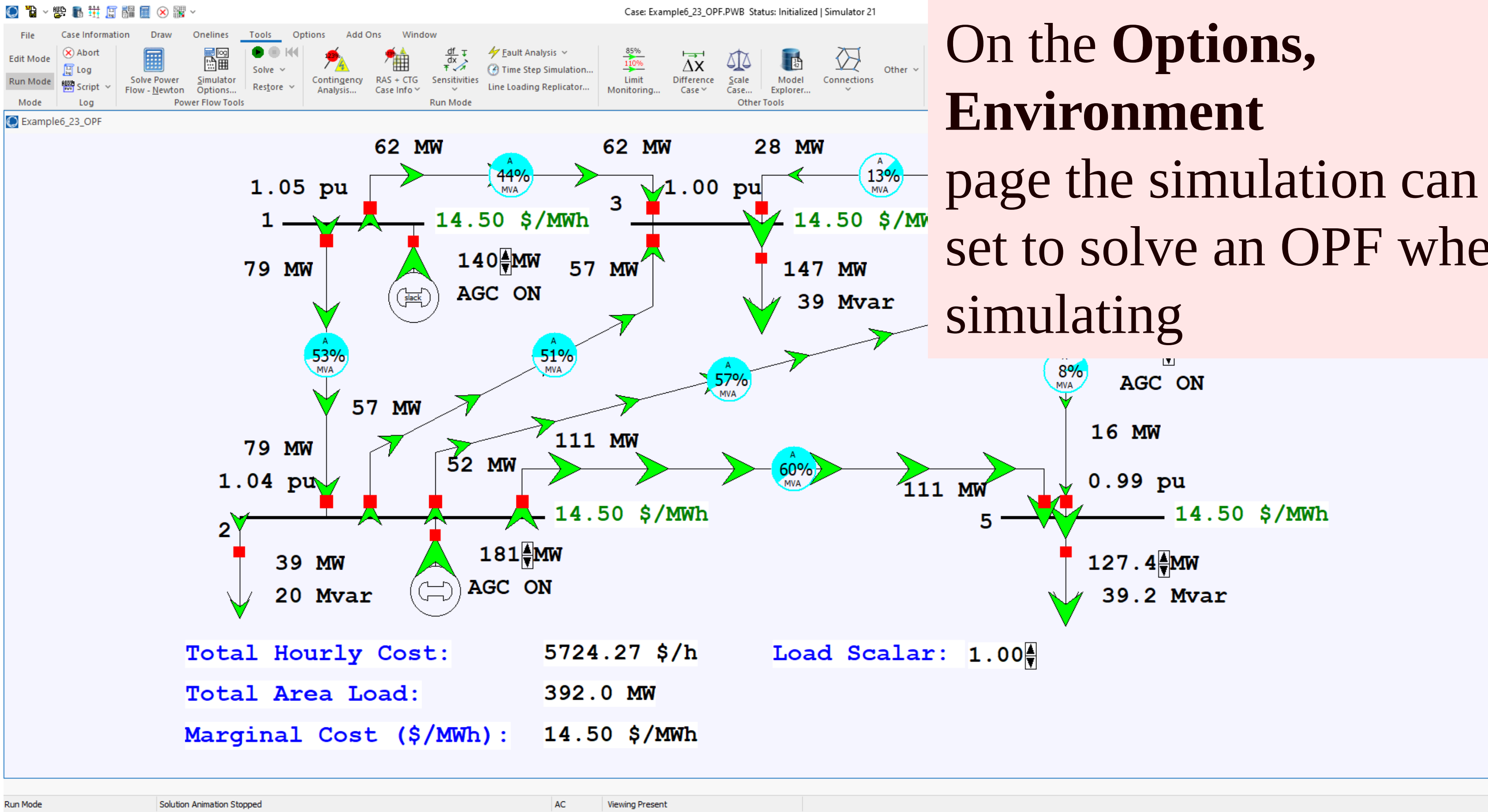
All LP Variables | LP Basic Variables | LP Basis Matrix | Inverse of LP Basis | Trace Solution

Records ▾ Set ▾ Columns ▾ f(x) ▾ Options ▾

	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



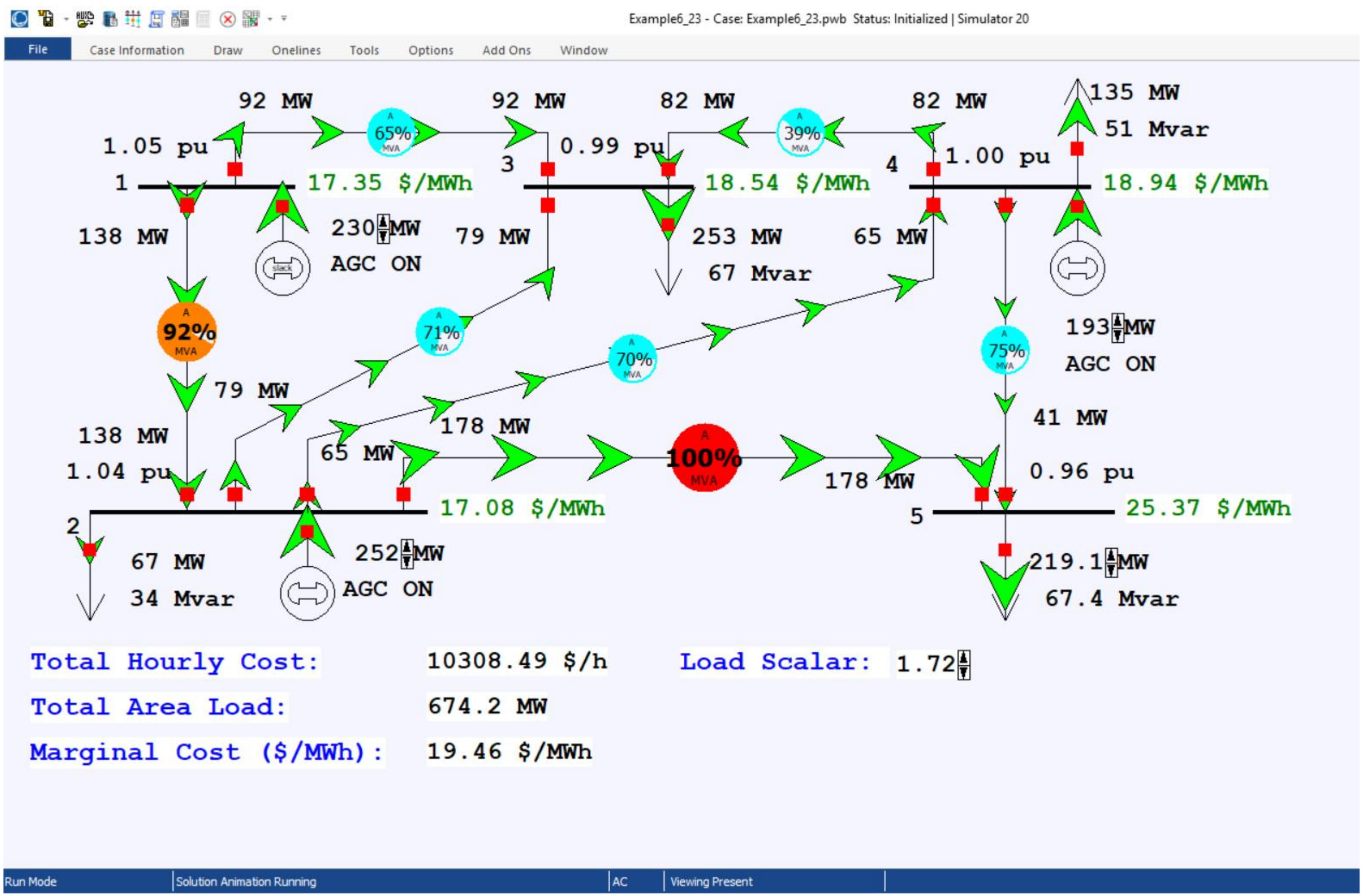
On the **Options, Environment** page the simulation can be set to solve an OPF when simulating

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

Locational Marginal Costs (LMPs)

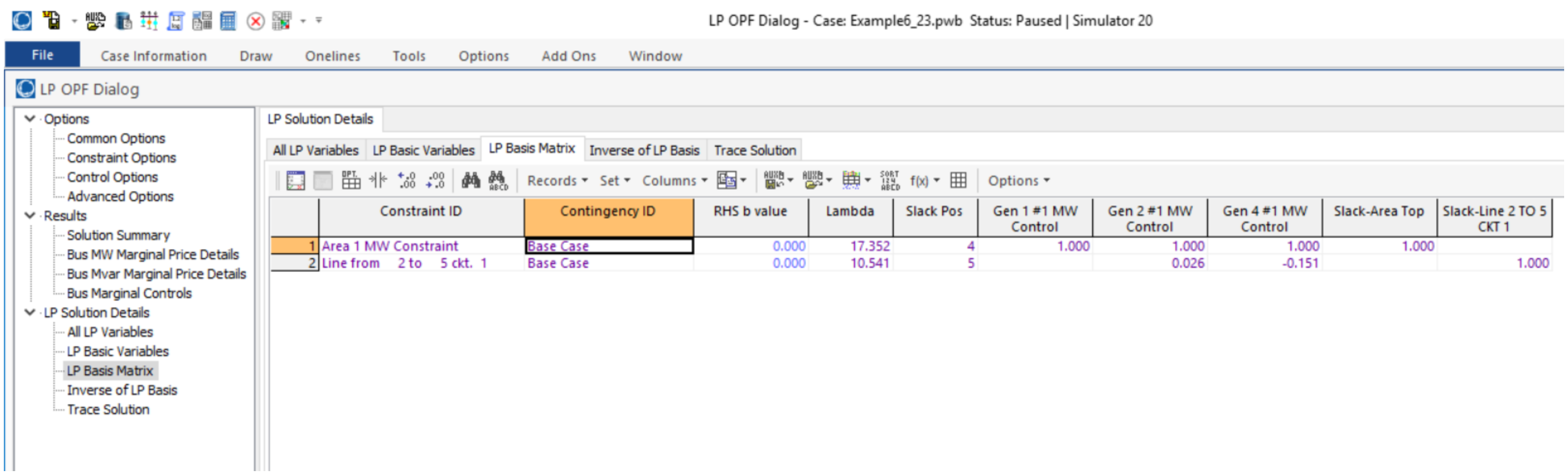
- In an OPF solution, the bus LMPs tell the marginal cost of supplying electricity to that bus
- The term "congestion" is used to indicate when there are elements (such as transmission lines or transformers) that are at their limits; that is, the constraint is binding
- Without losses and without congestion, 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)



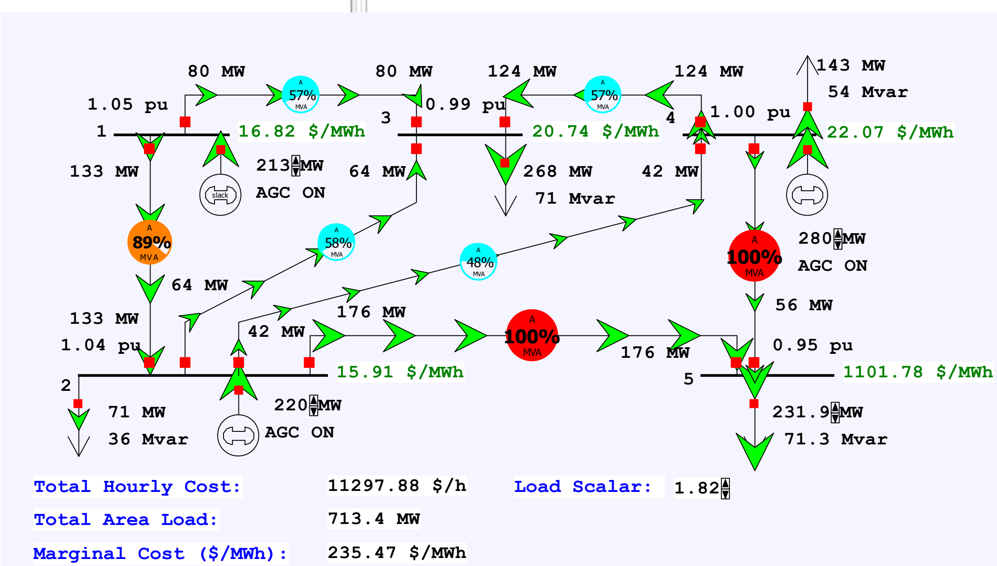
	Constraint ID	Contingency ID	RHS b value	Lambda	Slack Pos	Gen 1 #1 MW Control	Gen 2 #1 MW Control	Gen 4 #1 MW Control	Slack-Area Top	Slack-Line 2 TO 5 CKT 1
1	Area 1 MW Constraint	Base Case	0.000	17.352	4	1.000	1.000	1.000	1.000	
2	Line from 2 to 5 ckt. 1	Base Case	0.000	10.541	5		0.026	-0.151		1.000

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

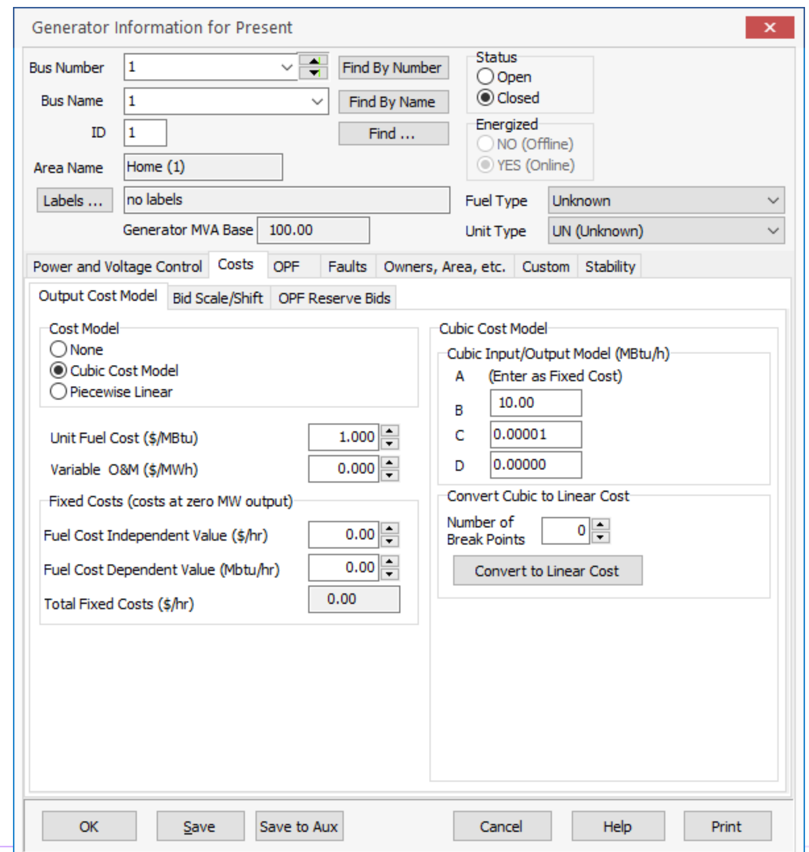
- This situation is infeasible, at least with available controls. There is a solution because the OPF is allowing one of the constraints to violate (at high cost)

						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

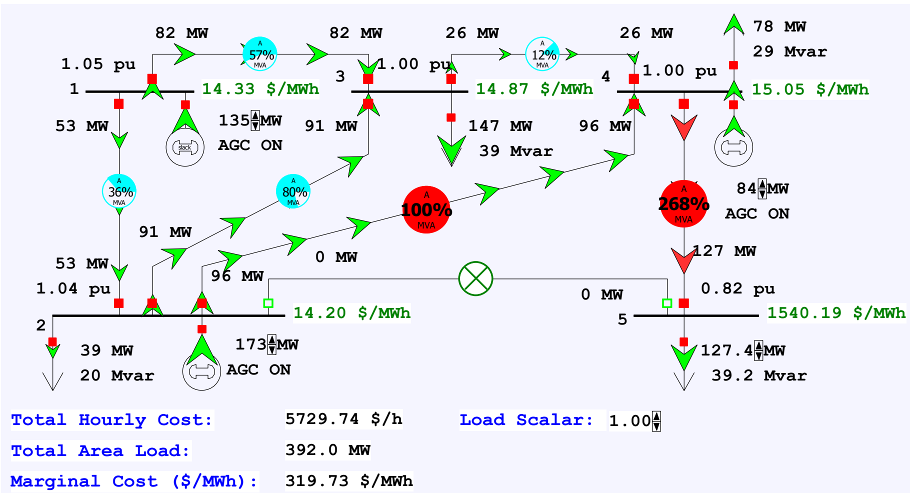
- Security constrained optimal power flow (SCOPF) is similar to OPF except it also includes contingency constraints
  - Again the goal is to minimize some objective function, usually the current system cost, subject to a variety of equality and inequality constraints
  - This adds significantly more 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.

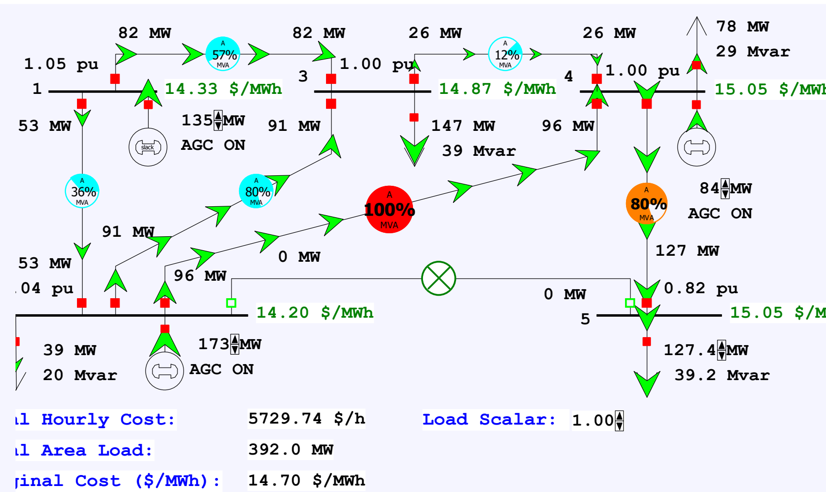
- With the inclusion of contingencies, there needs to be a distinction between what control actions must be done pre-contingent, and which ones can be done post-contingent
  - The advantage of post-contingent control actions is they would only need to be done in the unlikely event the contingency actually 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

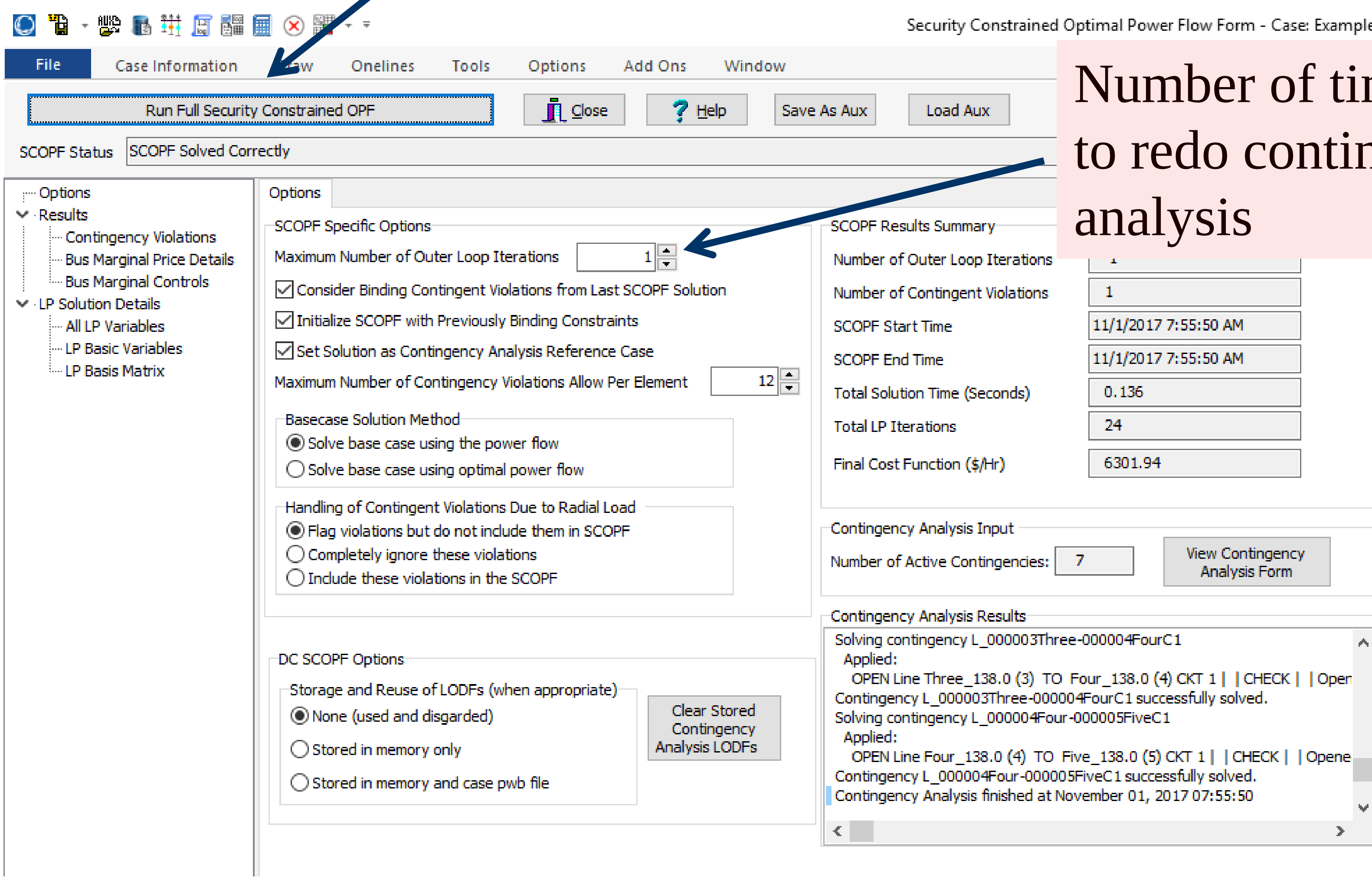


Modified with line 4-5 limit of 200 MVA with 2-5 out

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

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

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

Both $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are Lagrange Multipliers

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

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

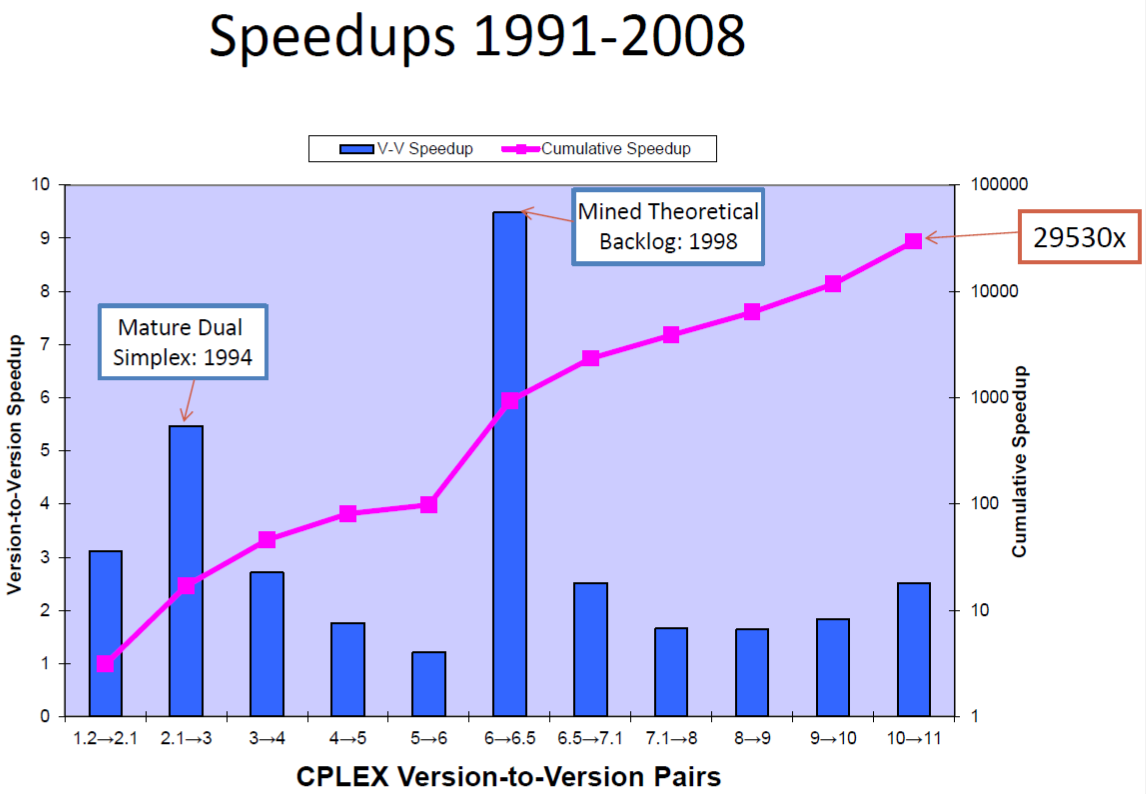
Mixed-Integer Programming

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

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

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

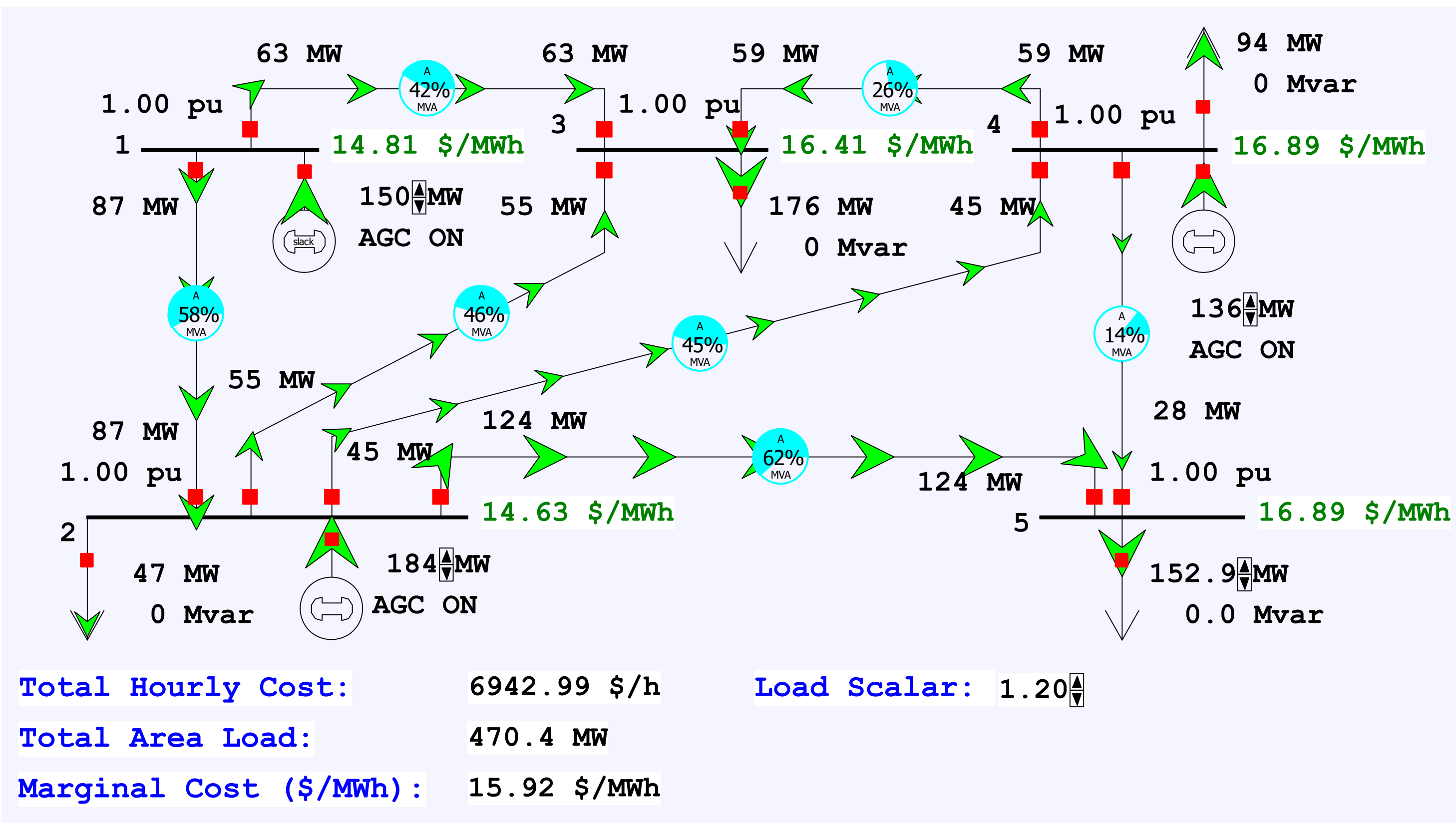
Linear programming can be solved by quadratic programming, which can be solved by second-order cone programming, which can be solved by semidefinite programming.

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**

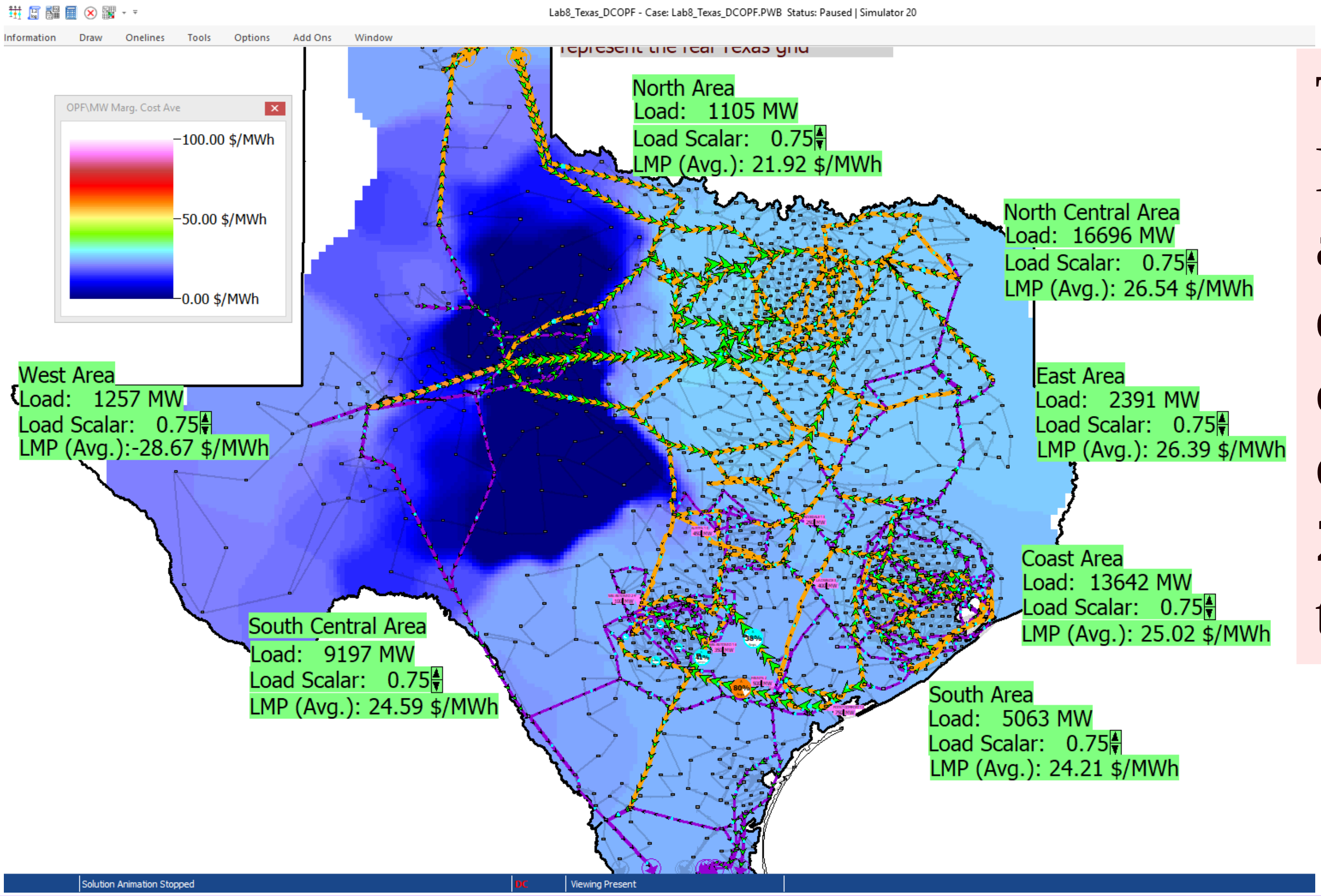
Example 6_13 DC SCOPF Results: Load Scalar at 1.20

- 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



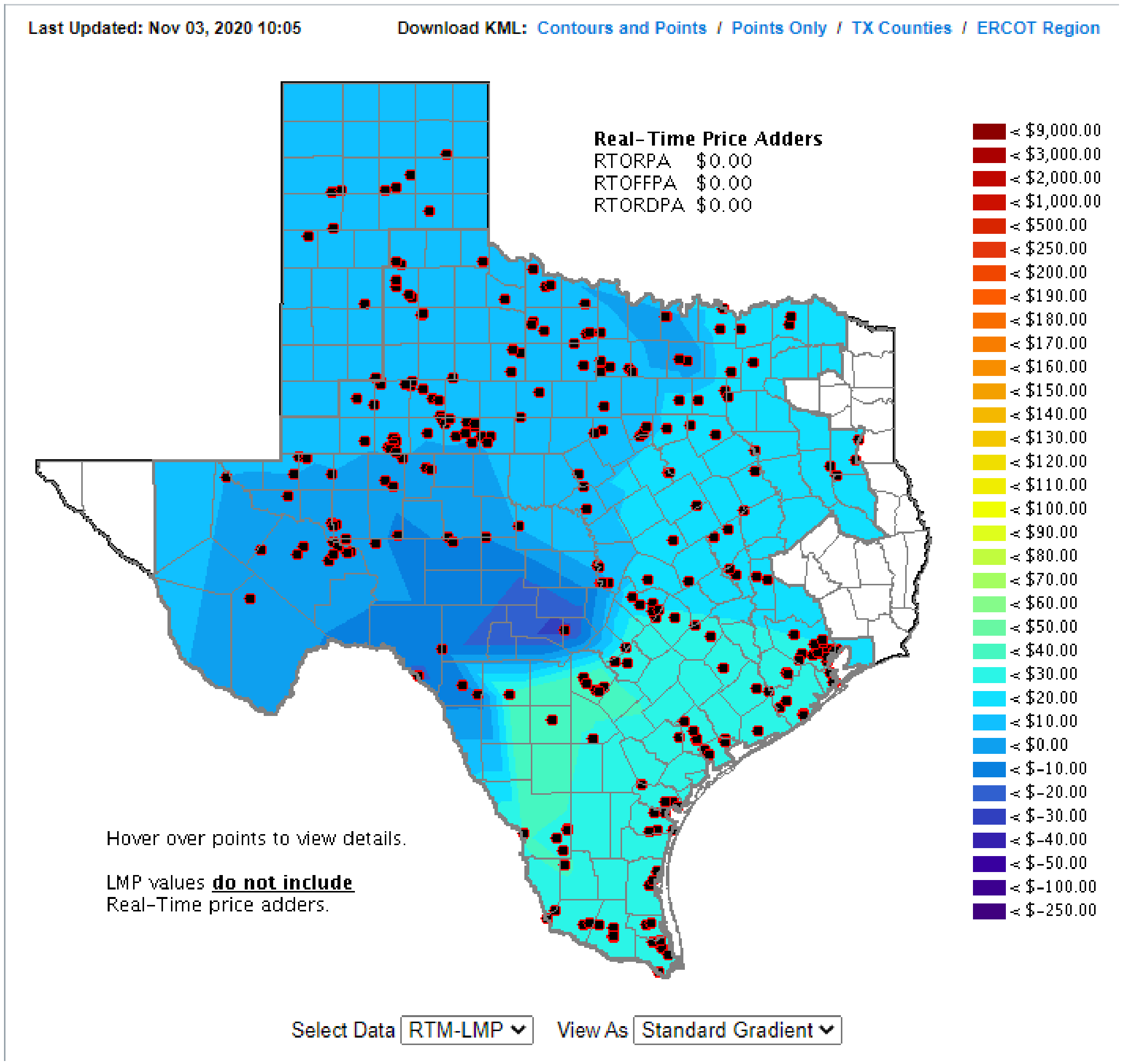
2000 Bus Texas Synthetic DC OPF Example

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



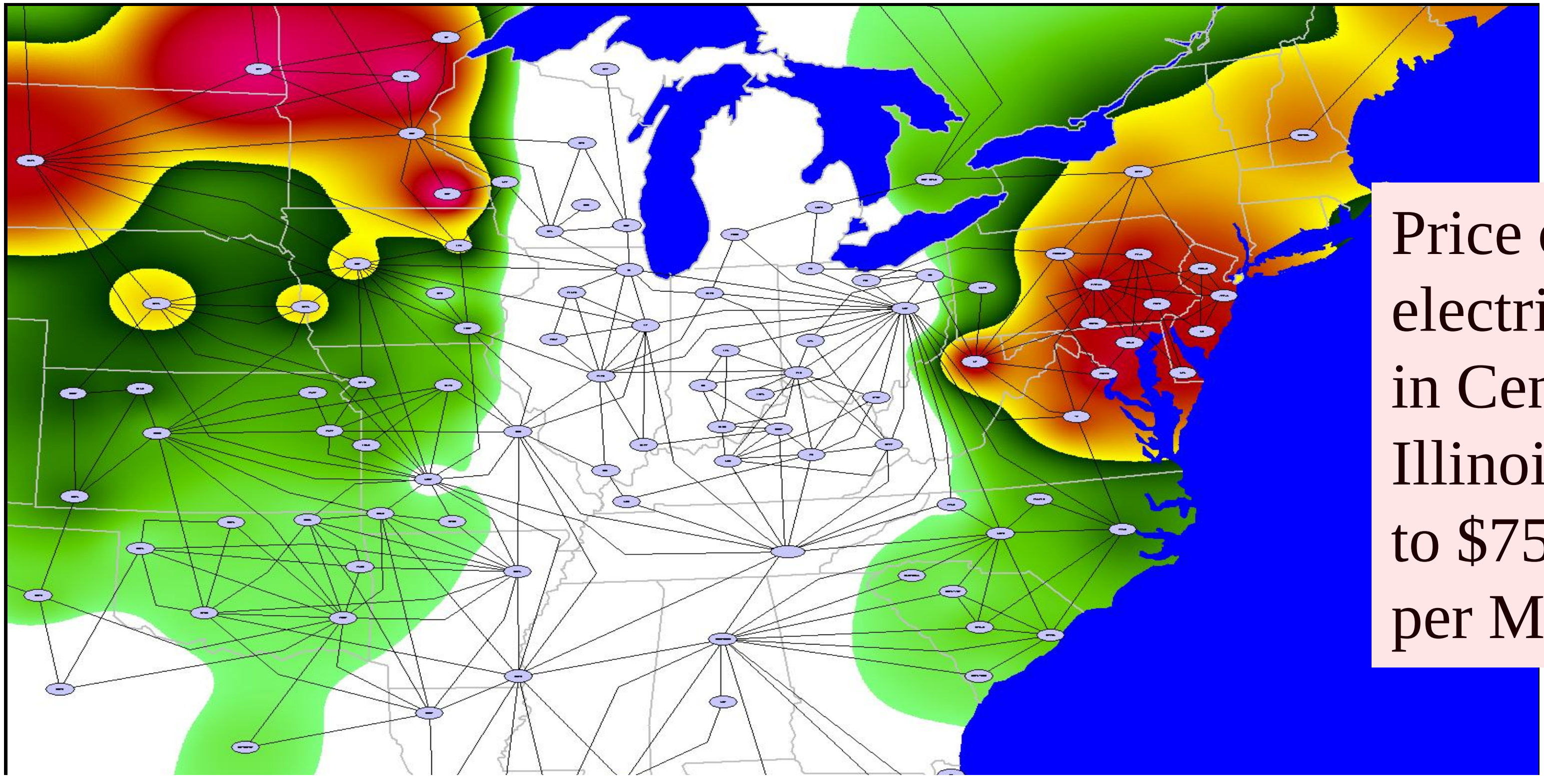
The quite low LMPs are actually due to a constraint on a single 230/115 kV transformer

Actual ERCOT LMPs on Nov 3, 2020 at 10:05 am



Source: www.ercot.com/content/cdr/contours/rtmLmp.html

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