Electricity Market Modeling with Mosel and the Native User Interface

Tom Halliburton
Energy Modeling Consultants Ltd
tom.halliburton@attglobal.net

for

Dash Optimization Users Group Meeting
17th May, 2007

Summary

- Model features
  - Electricity market model (Cournot equilibrium)
  - Hydro and thermal generation
  - Variability of inflows
- Software components
  - Spreadsheet interface
  - Visual Basic data manipulation
  - Mosel modeling language
  - Native user interface data exchange
  - C++

New Zealand Power System

- Approximately 70% hydro, with seasonal storage
- Thermal generation - coal, gas, oil
- Five large generation companies
- Electricity market - pool with nodal pricing
- Independent grid owner
- Separate local lines companies
- Electricity retailers, most owned by generators
- Price volatility, low inflow periods, transmission constraints, market power, illiquid derivatives

Model Objectives

- Developed for integrated generation - retail company
- Mid term planning
  - month ahead to 10 years ahead
  - weekly time steps
- Value at risk
- Fuel consumption forecasting
- Hydro reservoir management
- Marketing strategies, pricing
- New generation project analysis
- Policy responses

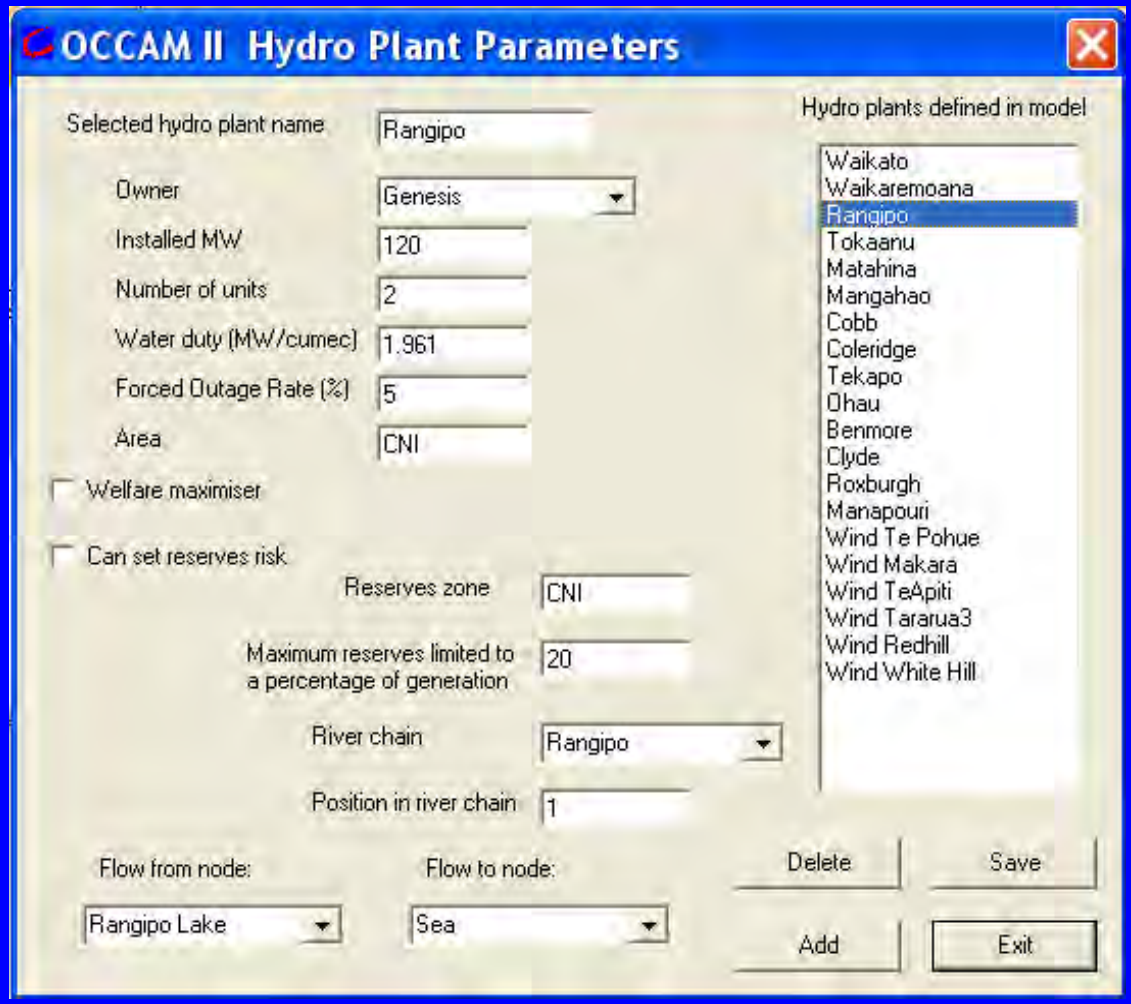
Algorithm Overview

- Solve a stochastic LP of the generation system
- Pass information to the Cournot market simulation for the first week
  - plant capacity, costs of generation, demand, hedge levels
- Solve for Cournot equilibrium for one time period for peak, shoulder, off-peak, etc
- Pass results back to Mosel
  - quantity of generation, prices, demand
- Roll LP forward one week
  - update constraints, bounds
- Solve stochastic LP again for 2 - 3 years ahead

User Interface

- Excel spreadsheet
- Users are expert analysts
- Excel provides all the editing and display features, so user interface development is minimized
- Add menu bar item to access model controls
- Data input directly or via VB forms
- Solved spreadsheet contains most results
- Further details in text files, retrieved by VB

Typical Data Input Form



Visual Basic

- Compiled dll, controlled from Excel menu bar
- Reads data from spreadsheet
- Displays forms for more complex editing
- Writes text files for Mosel
- Shells a process to run Mosel
- Waits until process finishes
- Writes results back to spreadsheet from LP and market simulation phases
- Enables access to more detailed results

Mosel

- Reads text data files
- Includes a function call to the market simulation
  - looks like any other function in Mosel
  - Mosel compiler checks arguments of function call
- LP objective is to minimize total system variable costs over a period of 2 or more years
- Detailed fuel supply modeling
- Stochastic aspect from uncertainty of inflows to hydro stations and reservoirs

Stochastic LP

- Dash stochastic module not used
  - Similar model written previously
- Inflow scenarios form a tree, with first period forming the trunk
- Each week more scenario branches appear
- Arrays structured with rows as scenarios, columns as time steps
- Use **`If exists`** syntax to test whether a scenario exists for each time period when creating constraints
- Branching has to be limited as problem can quickly become very large

Native User Interface

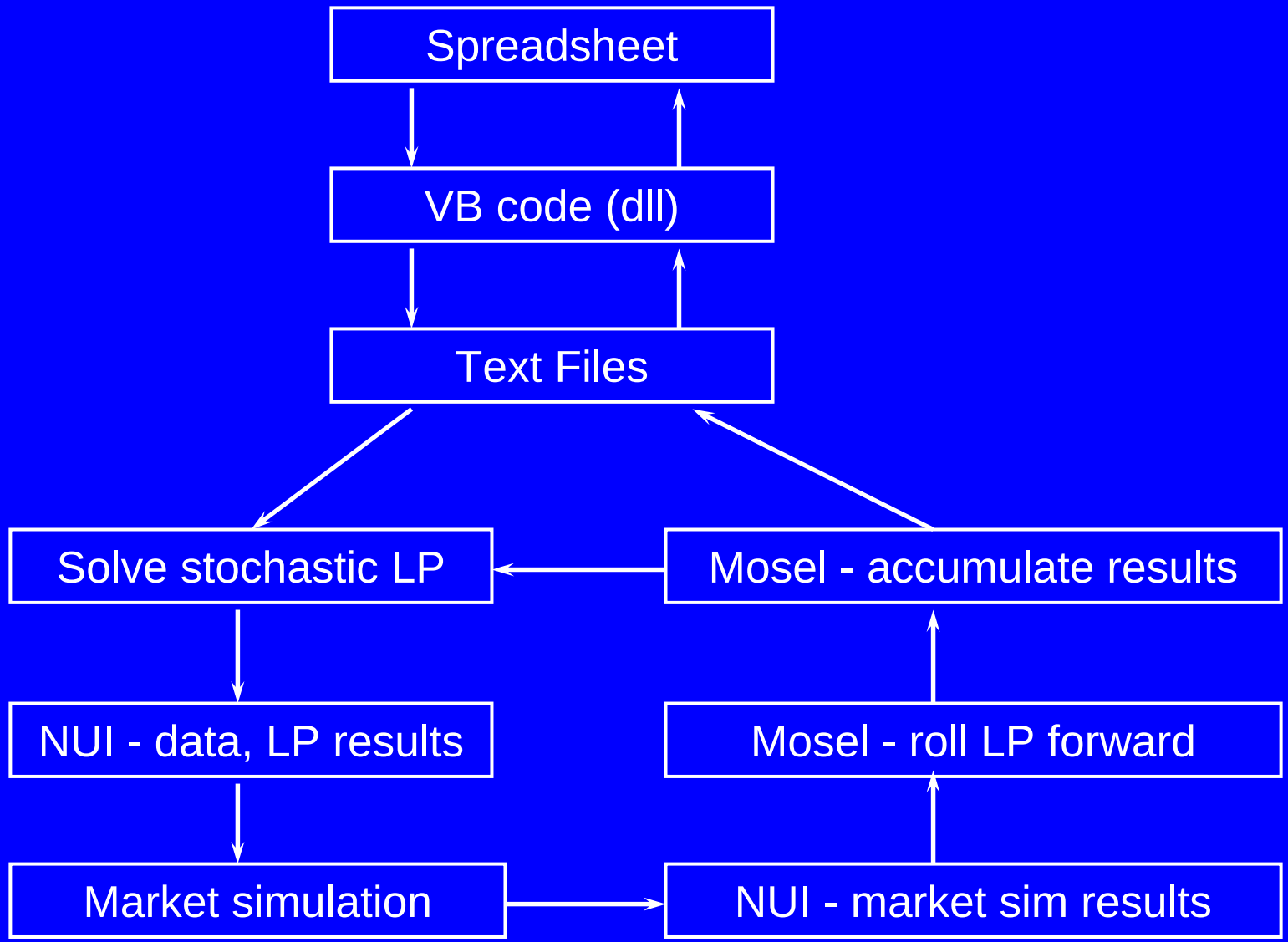
- Create a dso file
- Consists of compiled MS C++, using headers and NUI example code as a template
- Pass in data from Mosel
  - some processing to get this information into arrays etc for C code to work with
- Then write usual C++ code
- Similar processing to put data into other NUI arguments to send back to Mosel
- Debugging:
  - Mosel using IVE features
  - dso file using MS Visual Studio

C Code

- Uses data passed from Mosel by NUI
- Developed in parallel with Mosel code using bugging text files as input
- Solves a set of non linear equations
  - Lagrangian describing market equilibrium conditions
- Uses a non linear solution algorithm to find the Cournot equilibrium
- Function evaluation for the non linear solver contains a small LP

Mosel Again

- After solving for Cournot equilibrium roll the stochastic LP model forward one time period
  - update constraint coefficients
  - reload problem and basis
  - update bounds
  - minimize
  - write basis
  - global solve if required for unit commitment
  - save LP results to arrays
  - call market simulation function
  - save market simulation results to arrays
- Finally write text files of results, log files


Spreadsheet
VB code (dll)
Text Files
Solve stochastic LP
Mosel - accumulate results
NUI - data, LP results
Mosel - roll LP forward
Market simulation
NUI - market sim results

Software Component Issues

- VB development quick, easy to debug, some object oriented structuring possible
- Mosel very easy, IVE wonderful, has got better
- Mosel code easy to maintain, even a long period after development
- NUI initially intimidating but was quickly implemented:
  - start with example code supplied
  - add debugging outputs
  - write code test code for each data type to be used (integers, floating point arrays, strings)
  - fast data transfer
  - trouble free in operation
  - optional text output of data passed from Mosel in final version for debugging & maintenance

Software Issues continued

- Small LP embedded in C code required a large amount of effort - highlighted the benefits of Mosel
- Independent development of C dso market simulation code essential to meet schedule
- Microsoft C++ more rigid syntax than ANSI C in some situations - Dash support resolved some issues
- Instructions for setting up MS Visual Studio and debug settings from Dash support