



Under the Hood

Controlling Matrix Size and Other Helpful Tips

Andrew Cogswell, Senior Analyst

September 29th, 2005

Topics

- Review tips from last year
- Controlling matrix size
 - Variable time steps (eliminate decision variables)
 - Mixed, summary outputs (reduce accounting rows)
- Efficiency considerations
 - Summary outputs (reduce build times)
- Order of operations (reduce errors)
- Audience Q & A



Review: Tips from last year's UG

Append new types Areas section (now automatic)

Complex yield table arithmetic (_SUM, _MAI, _CAI)

Model log file (warnings!!)

Configure available memory in Woodstock

Matrix bloaters (constraints on inventory outputs)

Controlling Matrix Size

What makes a matrix large?

- Decision variables (actions)
 - Future choices are the real culprits
- Constraints (inventory constraints are the meatiest!!!)
- Non-zeros (output/yield coefficients)
- All of the above
- Length of projection (number of planning periods)
 - Desired projection length may produce too large an LP representation

If it ain't broke, don't fix it!!

Controlling Matrix Size

Solution: eliminate choices

How do you do it??

- Exclude action choices in certain periods
 - Optimize section keyword
 - Problem with narrow operability windows
 - Problems with constraint declarations
- Nasty to code

Controlling Matrix Size

Solution: eliminate choices

How do you do it??

- Compress time (future feature?)
 - Control section keyword
 - Eliminate choices (ages) beginning in a specific period
 - Keep first & last age in range, eliminate every Xth choice
 - *Take age, divide by X, if remainder = 0 keep it*
 - *Future types only!*

Controlling Matrix Size

Mixed summary outputs

- Outputs that reference both action- and inventory-based outputs
- May inflate LP matrix if used in Optimize section
 - Addition of accounting rows
- Avoid using mixed, summary outputs in objective function and constraints
 - Declare objective function using separate outputs for each output type



Efficiency considerations

Summary outputs

- Outputs that reference previously-defined outputs
 - Woodstock must check each mask in previous output declarations
 - Depending on degree of nesting, can impact model build times
- Coefficients tab (Contains a new attribute...nodes!)
 - Look for outputs with a large node count
 - Build a basic with a general development mask



Order of Operations

In a planning period, Woodstock performs the following activities in the following order:

1. Calculate operable inventories (what can be cut)
2. Calculate action outputs (what is cut)
3. Growth (ages all development types)
4. Calculate standing inventories (what is leftover)



Order of operations

When using standing inventory outputs to calculate fixed costs based on forest conditions

- Bear in mind that the output is calculated *AFTER* growth
- May need to add 1 to age/age-range in output declaration to issue costs in a timely manner



Questions?