RiverWare Optimization

RiverWare User Group Meeting
February 6 - 7, 2007

Tim Magee and Patrick Lynn

Reimplementation of Optimization

- Goals of reimplementation
  - Replace the Constraint editor with the Rules editor
    - Editable, shared code, one interface for users
  - Code had outgrown the original controller
  - Reduce the gap between Rules and Optimization
    - Remove the artificial differences
    - Lay the groundwork for combining them
- Replicated the results of the old optimization
- This year: Add an open source solver

  → Make it easier to do a little optimization

Overview

- Describe the Optimization framework
- Limitations of the old controller
- New optimization controller
- Demonstration – TVA's 6-hour model
- Future Development
  - 2007
  - Combining Optimization and Rules
- Discussion – Are there parts of your model that seem like an optimization problem?

Similarity Between Optimization and Rule Based Simulation

- Prioritized policy
  - From extreme conditions to normal operations
- Gradually remove the degrees of freedom from the solution

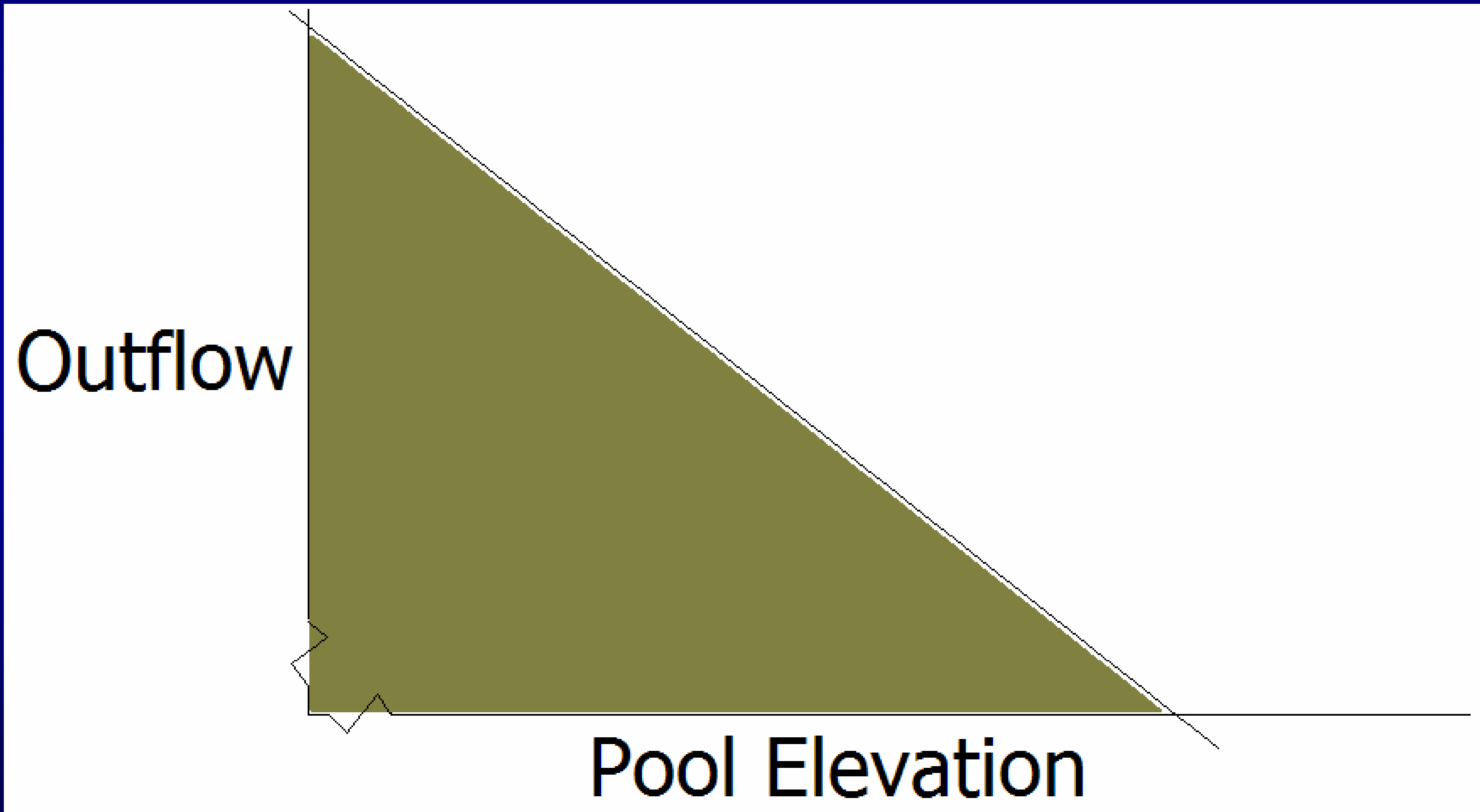
Main Differences Between Optimization and Simulation

- Best solution vs. evaluating inputs or following rules
- Solve all time steps simultaneously vs. stepping through time steps
- Degrees of freedom
  - Equations and Unknowns vs. If-Then
- Approximation vs. Exact calculation
  - Nonlinear functions

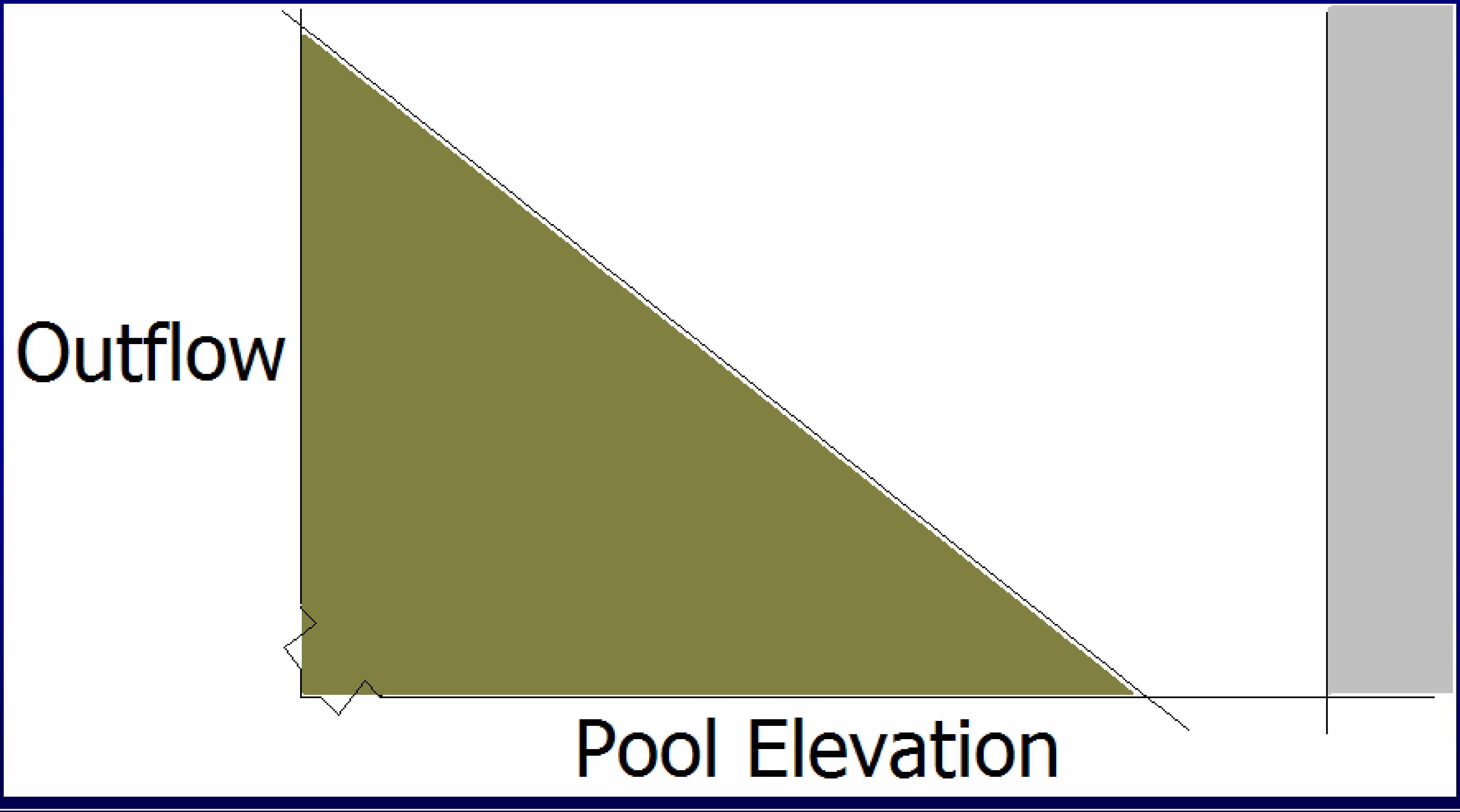
Goal Programming Example

1. Pool Elevation ≤ Elev Guide 1
2. Outflow ≤ Flow Guide 1
3. Pool Elevation ≤ Elev Guide 2
4. Outflow ≤ Flow Guide 2
5. Outflow ≤ Flow Guide 3

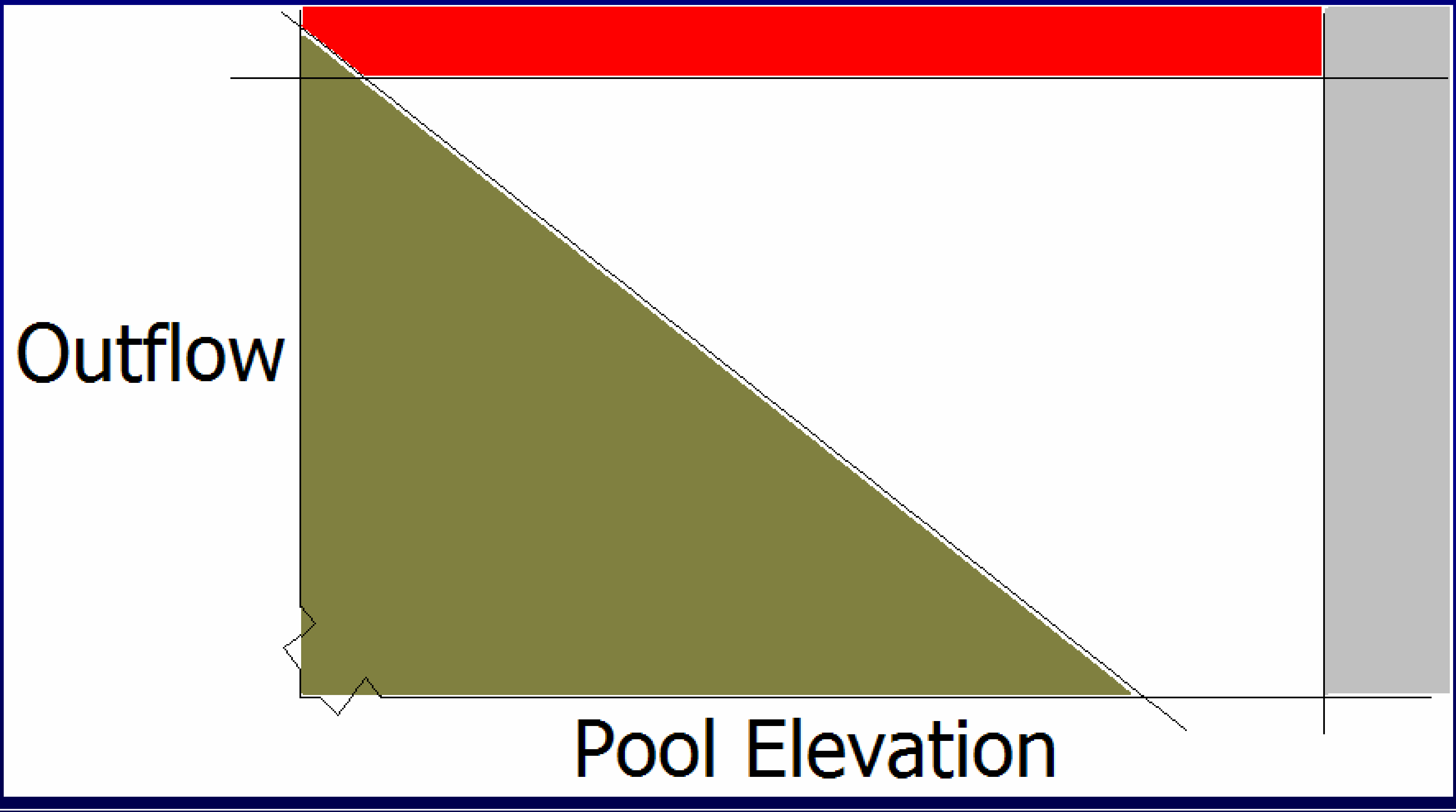
Goal Programming Example
Mass Balance with Minimal Inflow



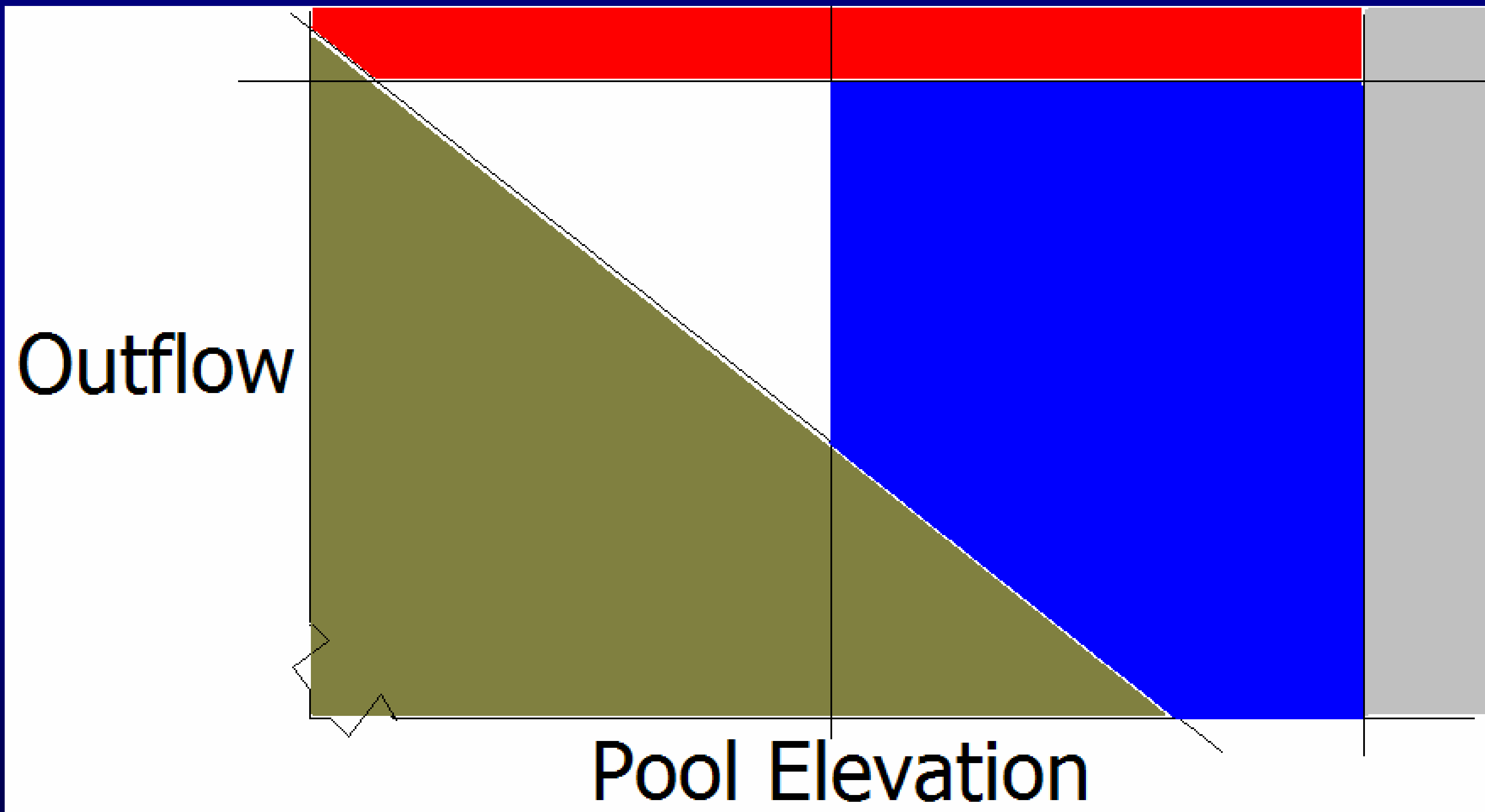
Goal Programming Example
Pool Elevation ≤ Elev Guide 1



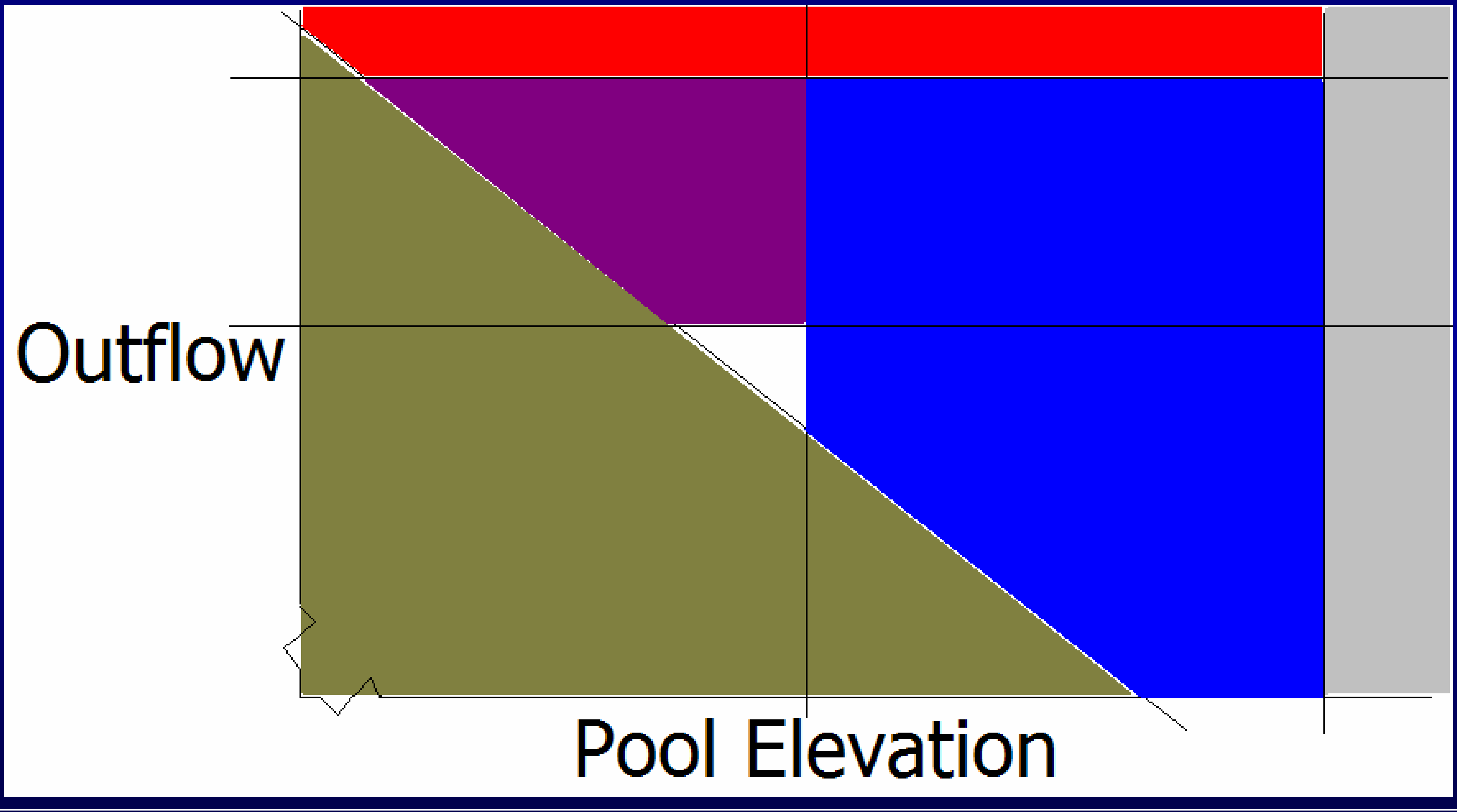
Goal Programming Example Outflow ≤ Flow Guide 1



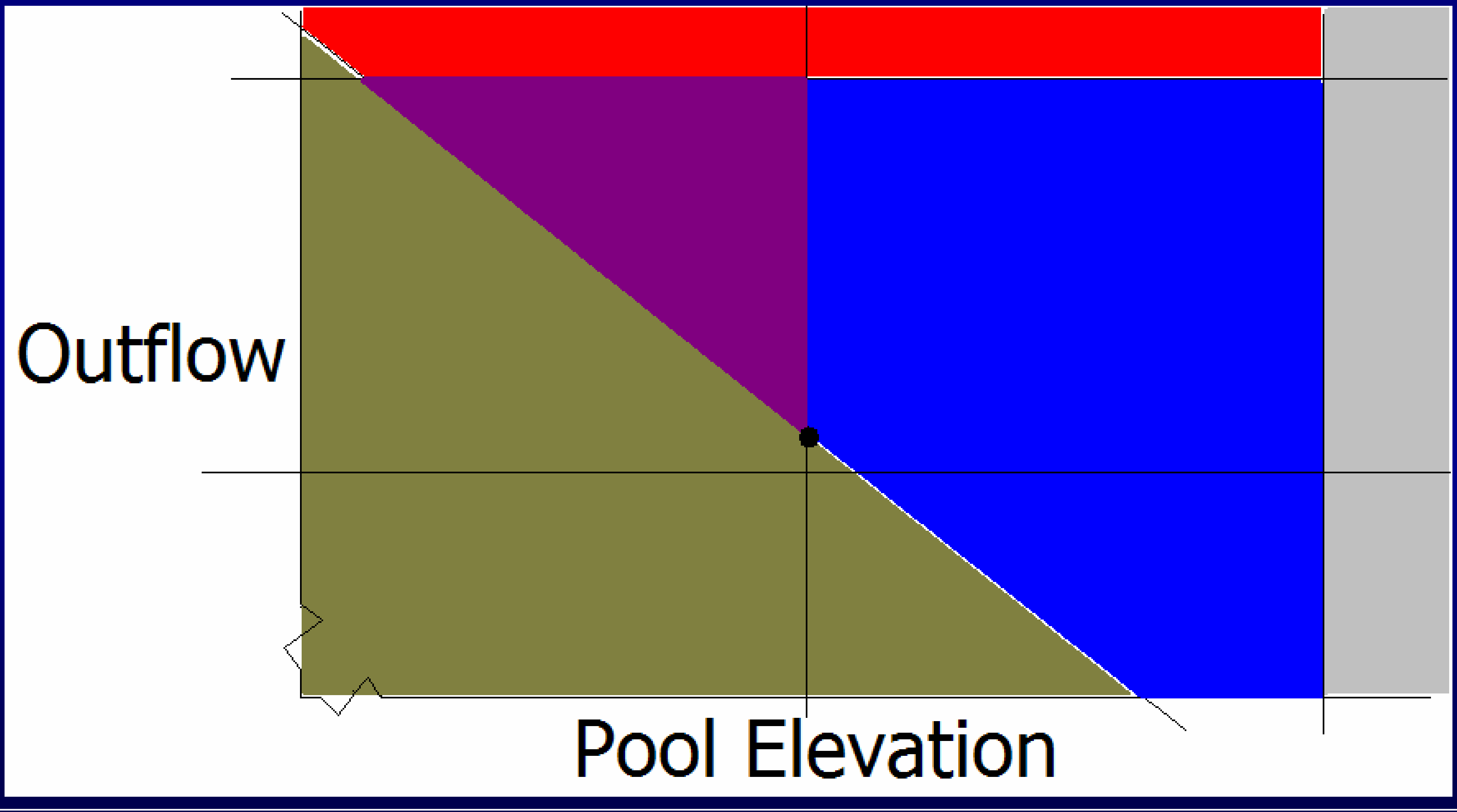
Goal Programming Example
Pool Elevation ≤ Elev Guide 2



Goal Programming Example
Outflow ≤ Flow Guide 2



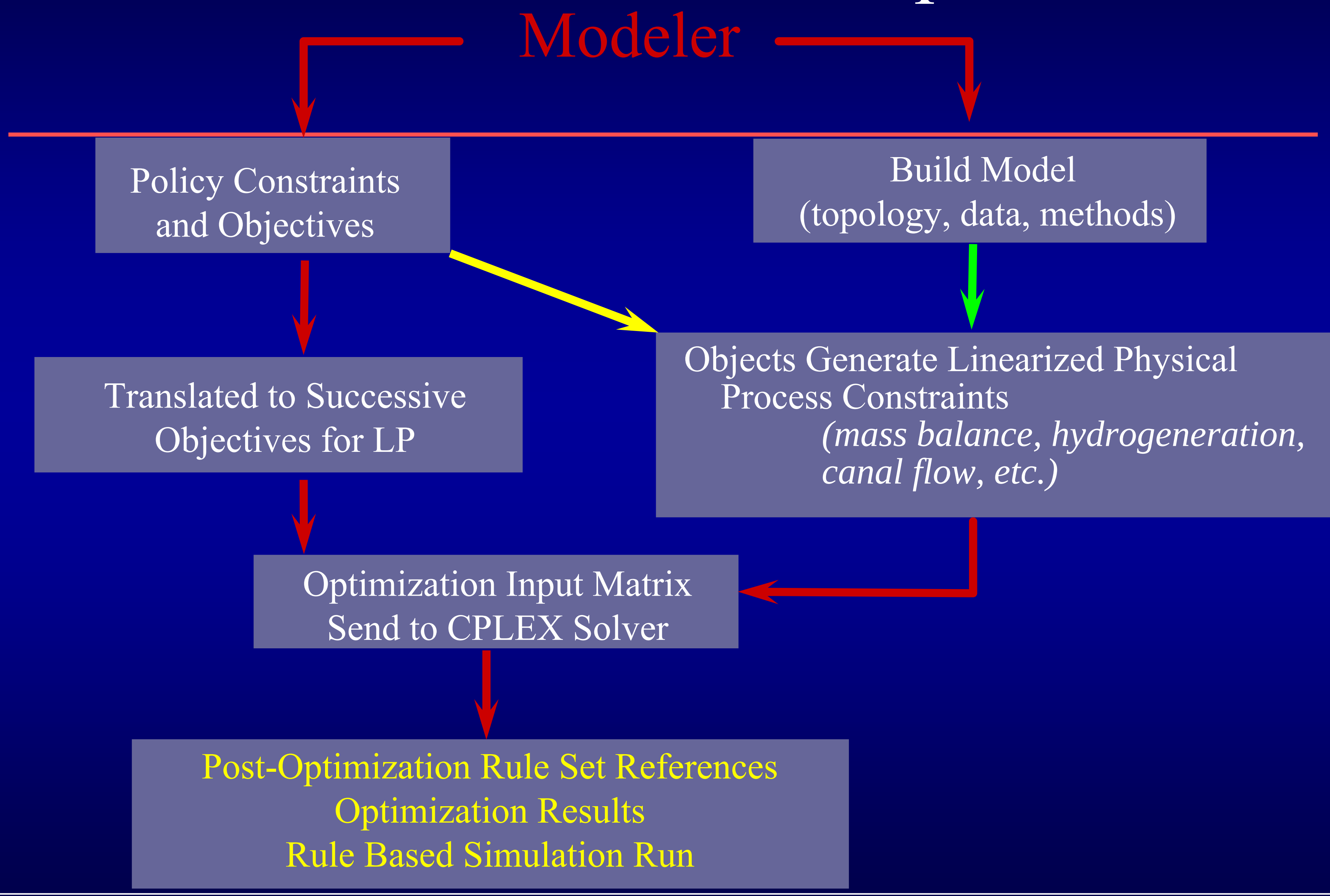
Goal Programming Example
Outflow ≤ Flow Guide 3



Components of a Preemptive Goal Program

- Variables with bounds
- Hard Constraints
- Prioritized policies
  - Objective functions and/or
  - Soft constraints

Automatic Generation of Optimization



Variables

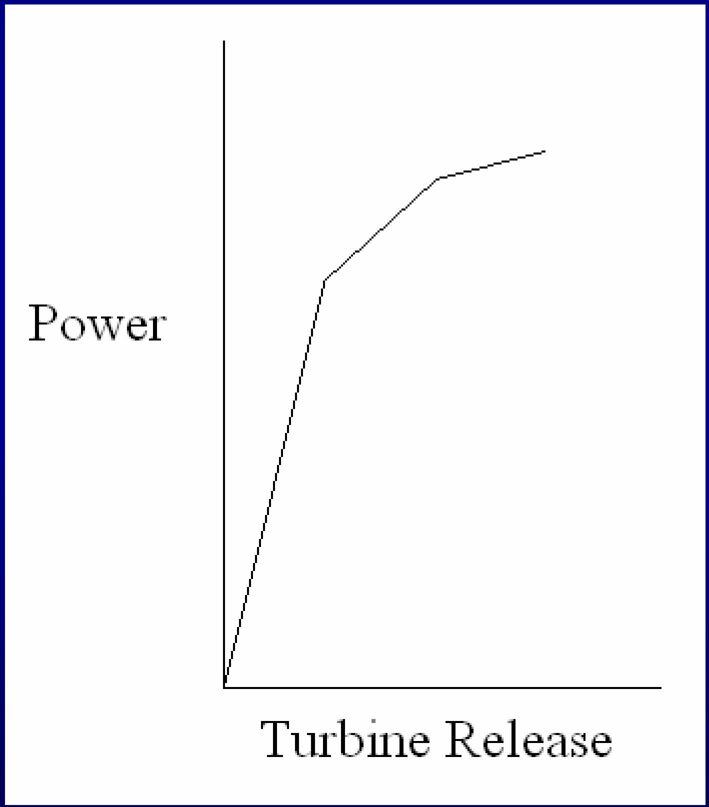
- Subset of the slots are decision variables
  - Could include more slots in the future
  - Only time steps without values
  - Only those referenced directly or indirectly by the policy
- Internal variables to assist with linearization of nonlinear functions
- Internal variables to convert soft constraints to objectives

Physical Model (Optimization)

- Mass Balance
- Links
- Lags on Reaches, Aggregate Reaches
- Canal Flow (linearization)
- Profile Storage (Headwater, Backwater and Intermediate Backwater Elevations)
- Power Plant (including Pumping)
  - Turbine Capacity, varying efficiency
- Could include more objects and processes

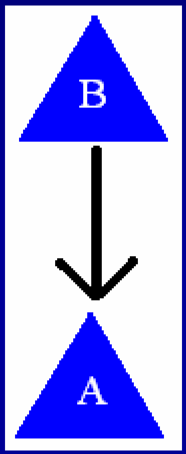
Linearization

- Many non-linear functions, for example:
  - Elevation = f(storage)
  - Power = f(turbine release, operating head)
- Optimization uses substitution, linear approximation and piecewise linear approximation



Replacement

- Some equations can be used to eliminate both a variable and a constraint
- Example
  - Reservoir A.Inflow = Reservoir B.Outflow
  - Could replace Reservoir A.Inflow with Reservoir B.Outflow throughout the model and eliminate the constraint



- In general, any equation can be used for replacement if the bounds on the variables are consistent

  → Smaller problem, usually more efficient to solve

Goal Programming

- Simultaneously solve all objects and time periods
- Prioritized sequence of objectives and soft constraints
  - Highest priority: Move towards normal region
    - Flood control, minimum flows, etc.
  - Lowest priority: In the normal region
    - e.g. Optimizing hydropower

Goal Programming continued

- "Freeze" each objective at the optimal value
  - Equivalent to writing a constraint
- Use remaining flexibility for other objectives.
- Objectives
  - Minimize or Maximize function
  - Derived Objectives: Minimize constraint violations
    - Summation – minimize total deviations
    - MiniMax – minimize the largest violation
    - Repeated MiniMax

Thermal Object

- “Economic” Object would be a better name
- Links to other objects to calculate system totals
- Block value of hydropower
- Incorporates outside power (Min, Max & Total)
- Pumped storage
- Policy referencing the thermal object can drive the whole system

Block Value of Hydropower

- For each time period:
  - Blocks of 50 MW of power
  - Value of generating each block decreases as blocks increase.
  - Value reflects the expected savings from turning off other sources of power, reducing purchases, or increasing sales.
- Objective function credits each block generated.

End Conditions: Two Options for Each Reservoir

1. Constrain final pool elevation or storage:
   - Pool Elevation >= Desired Value, or
   - Pool Elevation = Desired Value

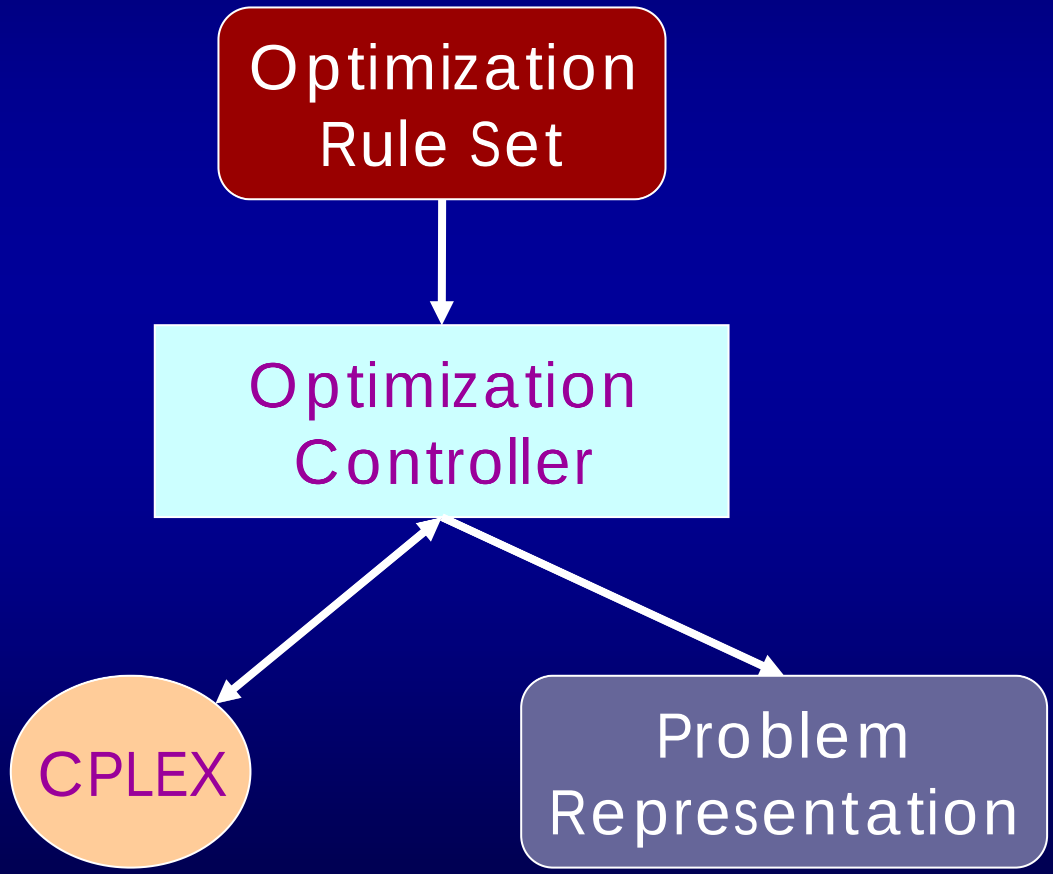
OR

2. Add cumulative value of storage to the objective
   - Piecewise linear function

Limitations of the Old Controller

- Constraint "Editor"
  - Actually was create or delete
  - Incredibly unforgiving of mistakes
- Aggregate Series
  - 3 Columns: Sim, Opt In, and Opt Out
  - Disconnected the optimization from the rest of RiverWare
- Post-Optimization Simulation
  - Placed inputs on all outflows
- Brittle: enhancements had outgrown the original controller

New Controller: RPL-based Optimization



New Controller: RPL-based Optimization

- Output values not cleared
- Run iterates through optimization rules (not time steps)
- Optimization rule execution:
  - Adds a constraint to the problem, or
  - Solves the problem, or
  - Freezes the problem
- Run result: final problem solution values (accessible from RPL)

New RPL Statements

- ADD CONSTRAINT <boolean expr>

  ADD CONSTRAINT Fish Lake.Outflow[Jan 2007] ≤ 100 [cfs]

- Expression linearized, not evaluated:
  - Lookup values
  - Replace non-decision variables
  - Add physical constraints
  - Result: $a_1x_1 + a_2x_2 + \ldots + a_nx_n < b$

New RPL Statements (continued)

- MAXIMIZE <numeric expr>

  MAXIMIZE Fish Lake.Power[Jan 2007] +
  Fish Lake.Power[Feb 2007]

- MINIMIZE <numeric expr>

  MINIMIZE TotalSpill()

New RPL Statements (continued)

- SUMMATION
  ADD CONSTRAINT <boolean expr>
  END SUMMATION

```
SUMMATION
   ADD CONSTRAINT Fish Lake.Outflow[Jan 2007] ≤ 100 [cfs]
   ADD CONSTRAINT Ice Lake.Outflow[Feb 2007] ≤ 200 [cfs]
END SUMMATION
```

New RPL Statements (continued)

- REPEATED MAXIMIN

      ADD CONSTRAINT <boolean expr>

  END MAXIMIN
- SINGLE MAXIMIN

      ADD CONSTRAINT <boolean expr>

  END MAXIMIN
- FREEZE

New RPL Statements (continued)

- IF (<boolean expression>
  <statements>
  END IF
- IF (<boolean expression>
  <statements>
  ELSE
  <statements>
  END IF

Other RPL enhancements

- Save Policy with Model
- Disable RPL Statements or List Items
- New Operators: SUM, AVE
- NUMERIC OptValue(SLOT, DATETIME)

A Typical Use of RPL-based Optimization

Simulation → **Optimization** → **Rulebased Simulation**

Compute consequences of input values

Solve a series of LP problems

Compute consequences of input values and selected optimal values

The Post-Optimization rule set

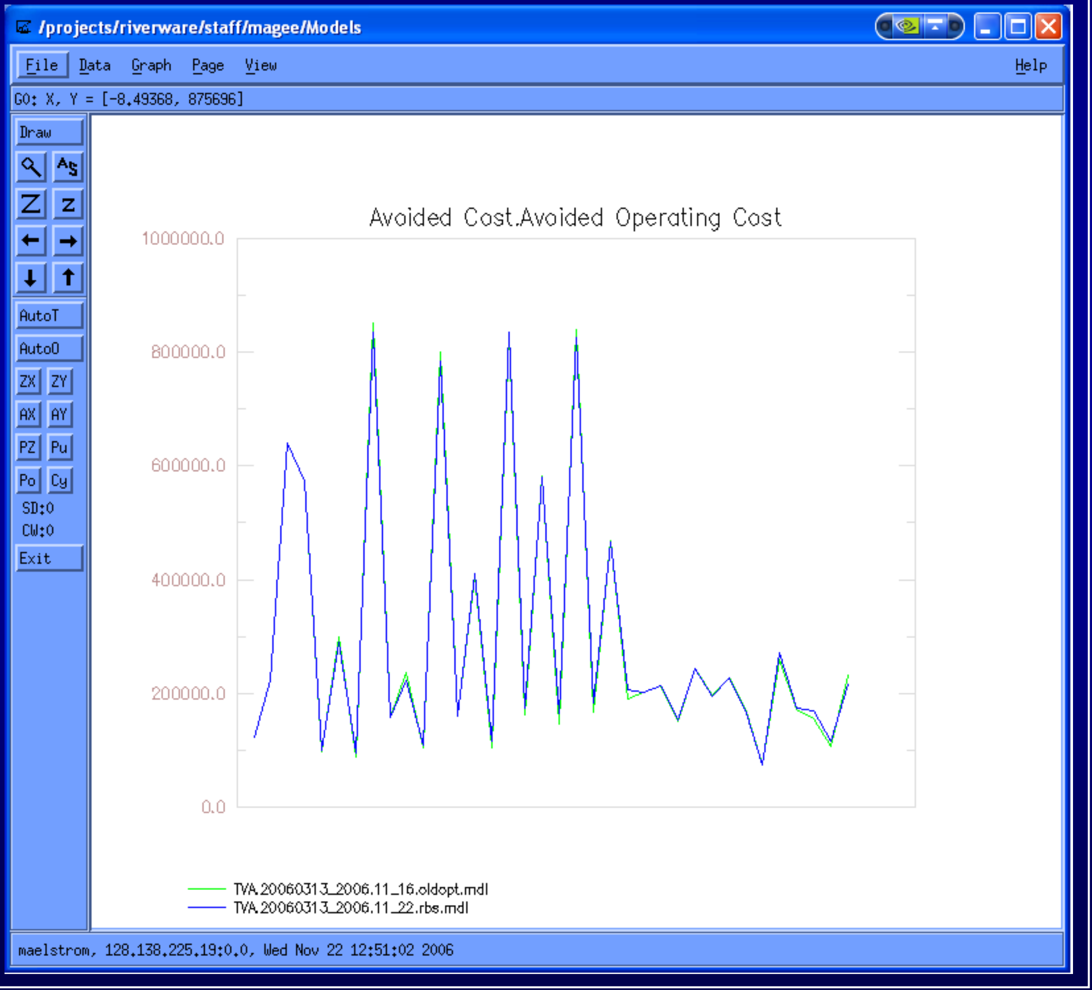
- Automatically created by switching from RPL-based optimization controller
- Sets reservoir Outflow values to the value computed by optimization
- Will drive a simulation based on inputs followed by optimal solution values
- Corrects for approximation errors in solution

Future Work

- Porting TVA Models - Current
- Performance Improvement - Next
- Integer Programming for Power - FY 2007
- Alternative Solver – 2007
- Potential Enhancements
  - Run Analysis
  - GUI
  - Many others

Port of TVA 6-Hour Model

- Model was used to debug the new code
  - Matched the optimal objective function within 0.25%
  - Mildly different timing
  - Soon to be a regression test

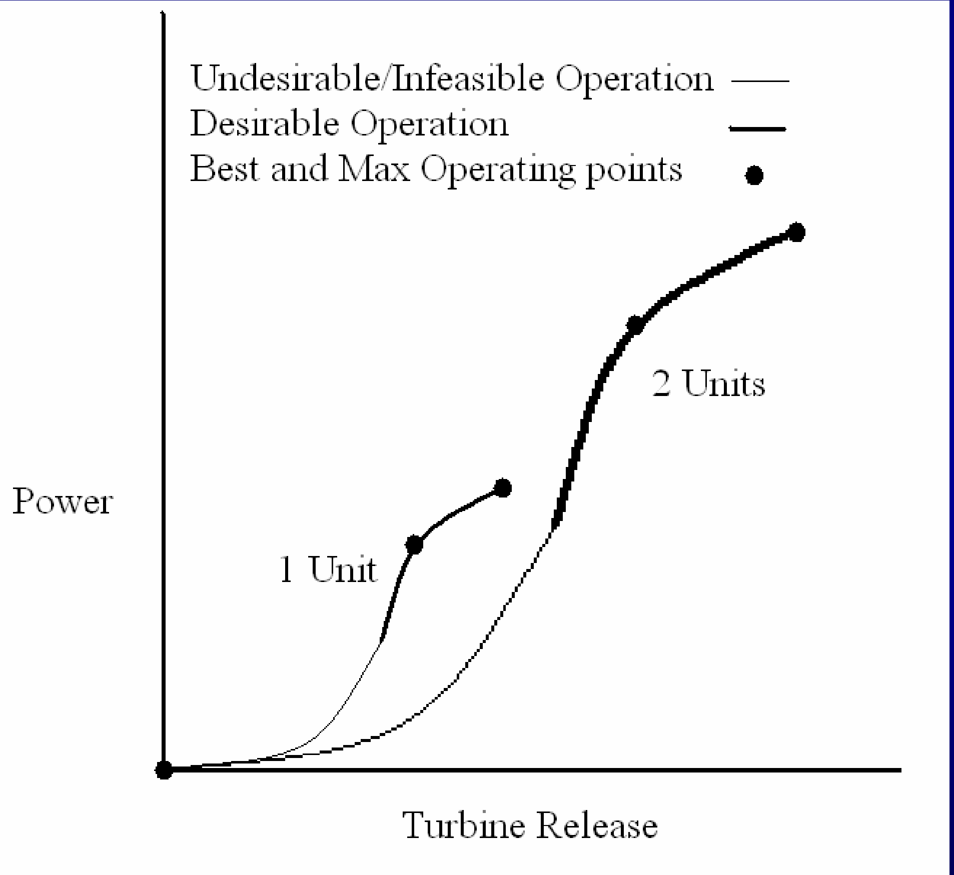


Immediate Tasks

- Finish port of TVA's Hourly Model
  - Policy written
  - Filling in additional data
- Performance
  - Current run time is approximately 2 hours
    - Zero effort on this so far
  - Old optimization was approximately 5 minutes
- Side by side testing

Integer Programming for Power

- Currently, piecewise linear approximation
- Leads to manual post-processing
  - Time consuming
  - Can violate constraints
  - Suboptimal



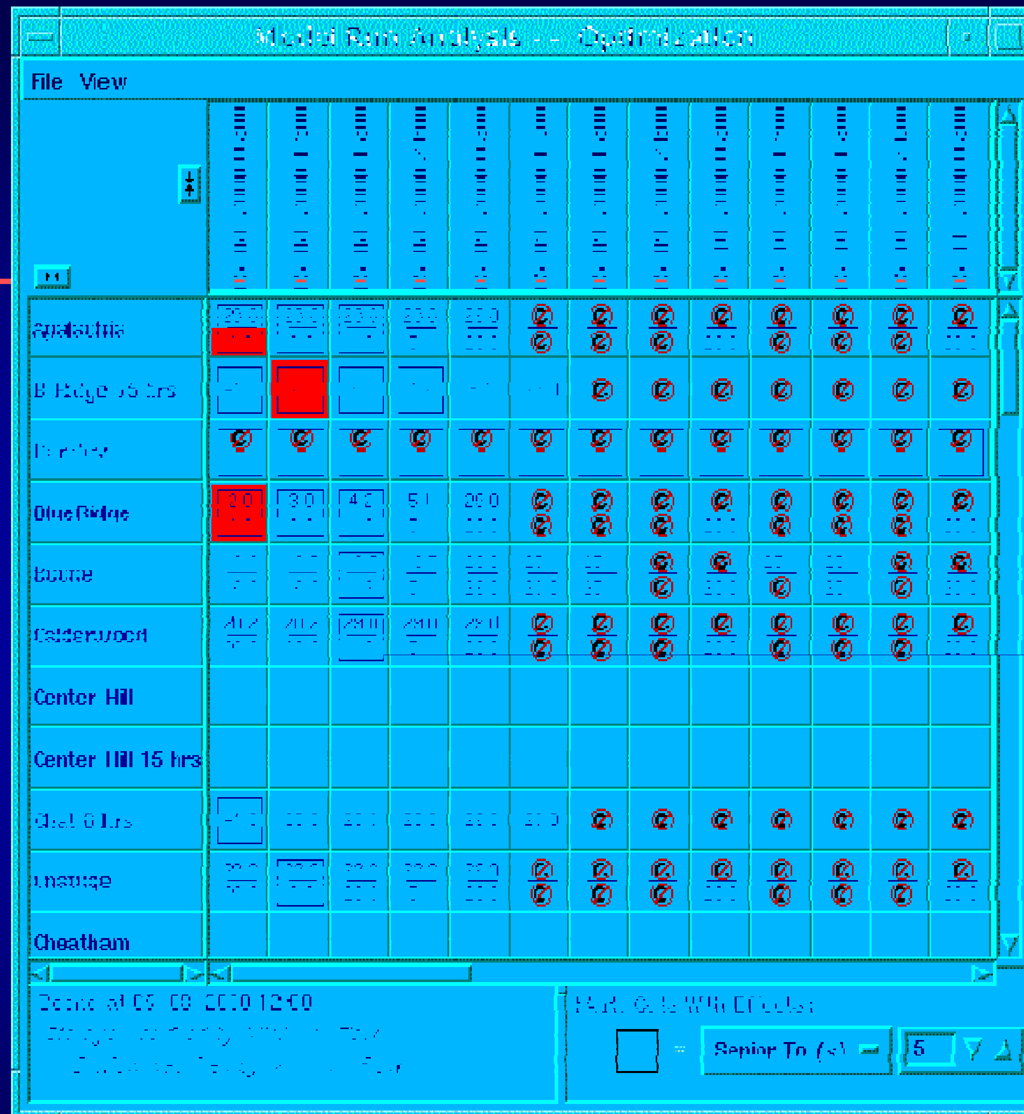
Operation of 1 and 2 Units

Alternative Solver Update

- CPLEX is great, but expensive
- Researched commercial and open source alternatives
- Selected COIN-OR Project
  - Open source, C++ solvers, and more
  - COIN Linear Programming (CLP)
  - COIN Branch and Cut (CBC) Integer Programming
  - CLP and CBC based on commercial IBM OSL
    - Continued support from IBM staff
  - Support of the operations research community, INFORMS
  - Critical Mass in 2006.
    - First Workshop
    - Versions 1.0+
- Will add CLP and CBC to RiverWare in 2007

Optimization Analysis

- What drove the solution?
- Hints
- In the old optimization
- Needs to be reimplemented
- Galaxy



Additional GUI

- Physical Constraints
- Optimization "Problem"
  - Optimal Solution
  - Only solution access right now is via a RPL predefined function

Combining Optimization and Rules

- Design that opened this possibility
  - Both in RPL
  - Treat the slots the same way
  - Optimization only affects the workspace through rules
- Partially there already
  - If-then logic in optimization
    - Conditional on pre-run conditions AND/OR
    - Conditional on the last optimization solution
  - Rules allowed before and after the run

Rules Before and After the Run

- Pre-optimization Rules
  - To enforce consequences of input values
- Optimization
- Post-optimization Rules
  - Pre-optimization rules
  - Additional high priority rules
    - To selectively override optimization results
  - Return part or all of the optimization solution
  - Additional low priority rules
    - To set slots or time steps that were not optimized

Future: “Hypothetical” Optimization

- Function similar to Hypothetical Simulation
  - Subset of objects
  - Subset of time steps
  - Creates optimization problem instead of cloning
  - Returns values to a rule
  - The calling rule may set current slots or take other action based on the optimization results

Future: "Iterative" Optimization

- Similar to Iterative MRM
  - Start with the existing framework
    - Sim/Rules
    - Optimization
    - Rules
  - Add iteration to the sequence as a whole
  - Could include an MRM RPL Set

Discussion

- Are there parts of your model that seem like an optimization problem?