SAND2013-8276C

A Study of Various Model Formulations and Settings for a Real-World Large-Scale Acquisition Problem

Frank M. Muldoon
Sandia National Laboratories

10/06/2013

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Outline

- Introduction
- CPAT 

- Coupling with CPLEX
- Various Model Formulations
- CPLEX 12.5 vs. 12.3
- CPLEX Parameter Settings
- Final Thoughts

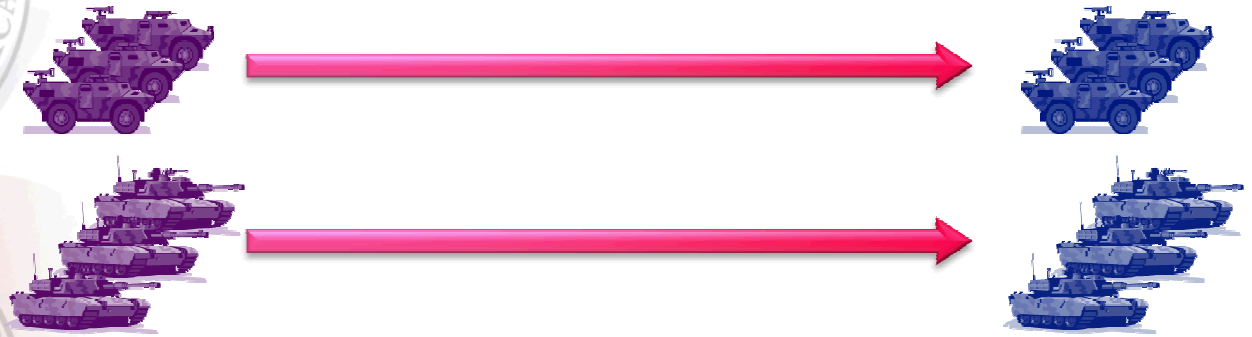
Introduction

- The U.S. Army currently has a large vehicle fleet that they wish to optimize over the next 25 to 30 years
- Questions they want answered include:
  - What is the highest performing fleet possible?
  - When should vehicles be upgraded or purchased?
  - Are current schedule requirements even possible?
  - Are budget restrictions feasible and how does this affect the fleet?

Capability Portfolio Analysis Tool (CPAT)

Introduction

- **CPAT optimizes the mixture of vehicles within the entire fleet through time.**

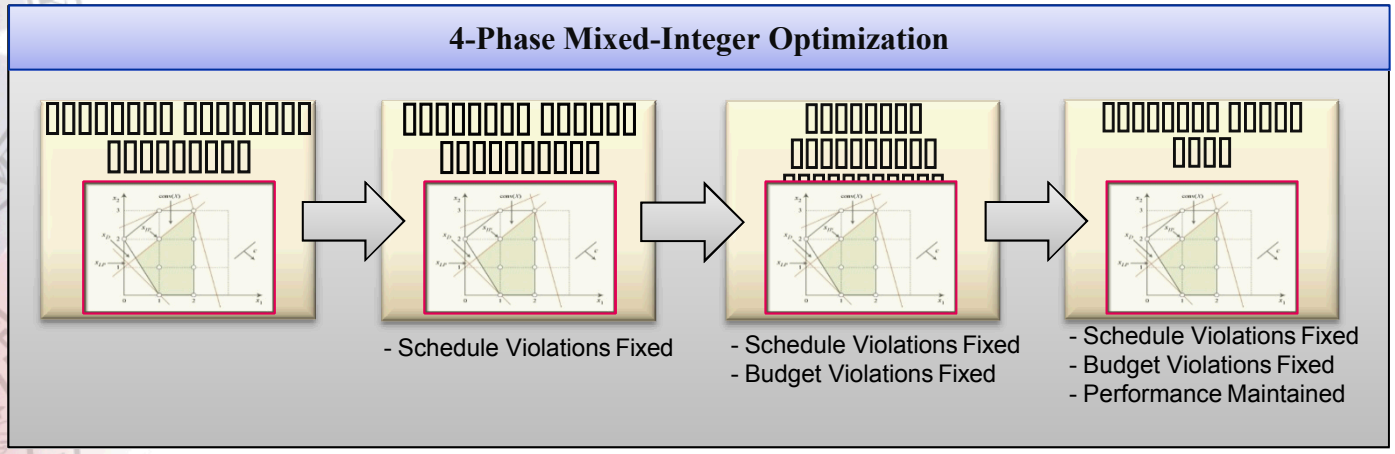


Current fleet Future fleet

- **It uses a multi-stage mixed-integer linear programming (MILP) approach to perform this optimization**
  - **Goal is to maximize the sum of the performance of all vehicles in the fleet over the study horizon**

CPAT and CPLEX

- **CPAT solves four optimization problems in the following sequence:**

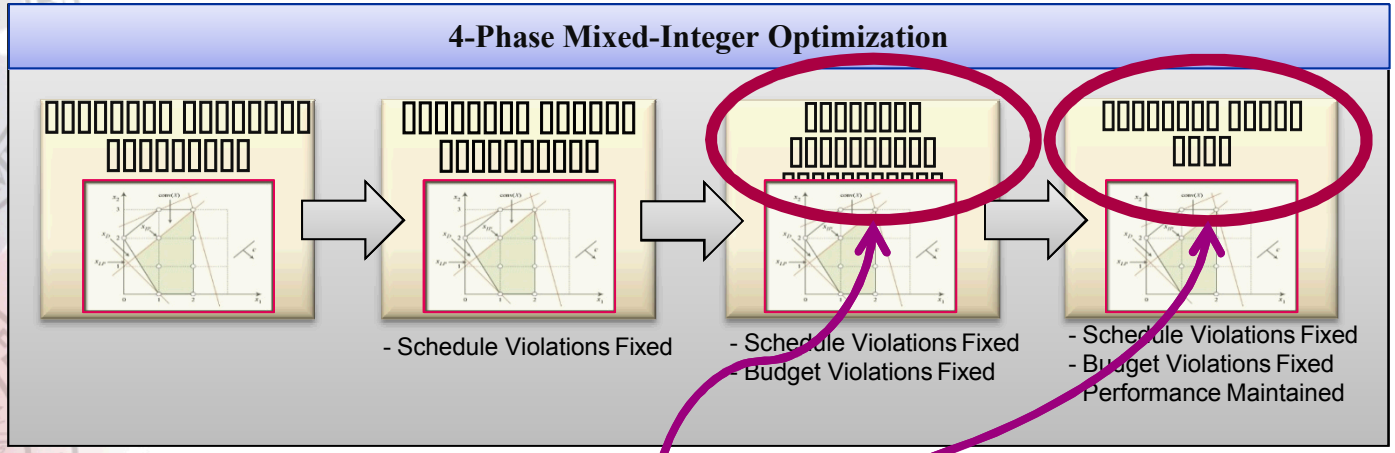


  - **Solutions are passed from one phase to the next**

- **CPAT uses CPLEX to solve these difficult MILPs**

CPAT and CPLEX

- **CPAT solves four optimization problems in the following sequence:**



We will primarily focus on the Performance Phase and Cost Phase

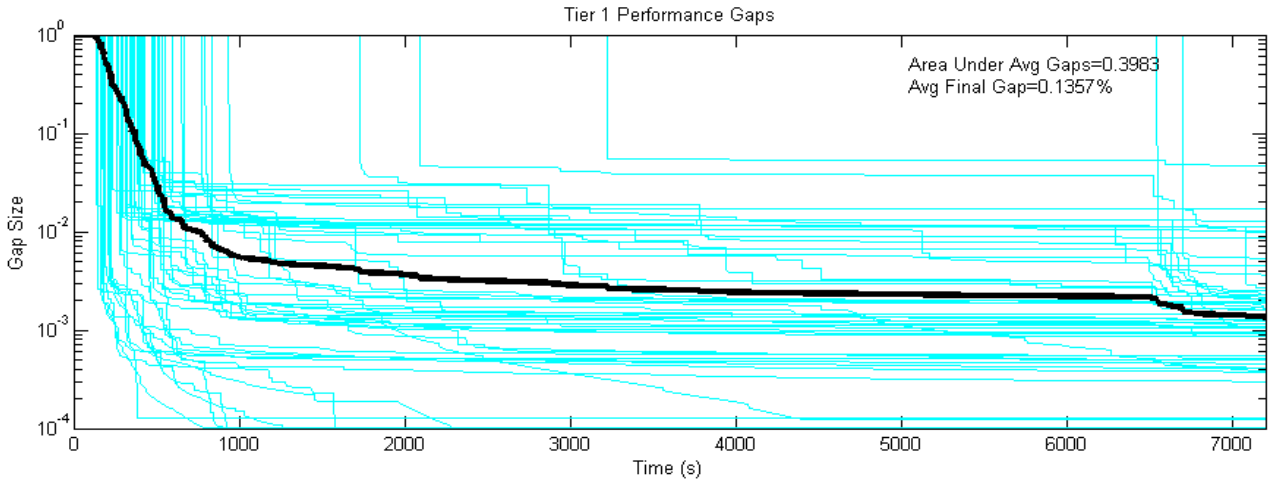
Goals

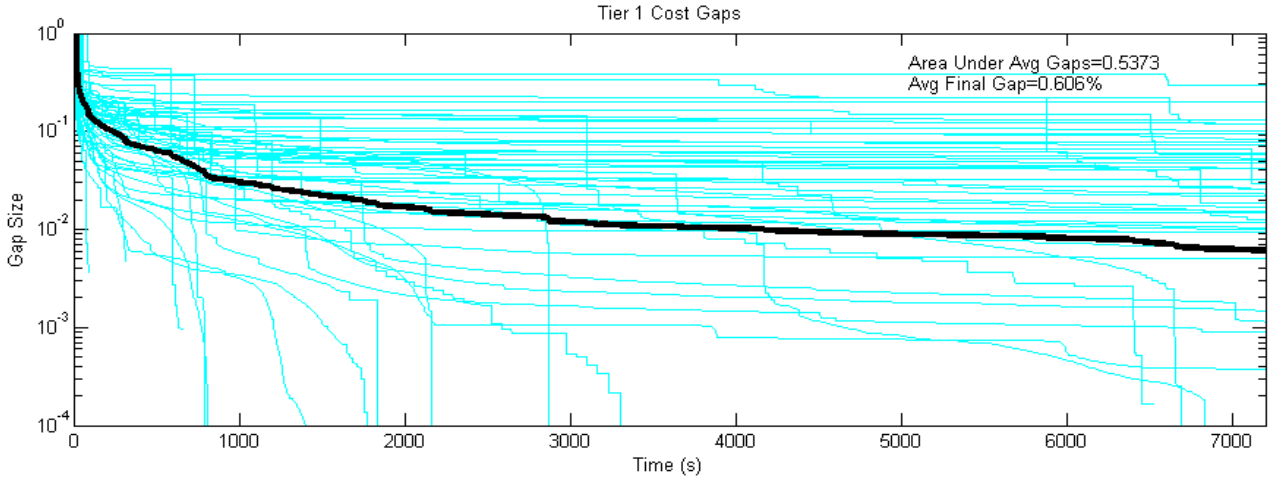
- The Performance Phase and Cost Phase are the most difficult to solve so we will primarily focus on these phases
- We wish to decrease the running time and decrease the "gap"
- The "gap" is defined as the percent difference between the value of the optimal continuous relaxation and the value of the current best integer solution
- The target gap for the Performance Phase is 0.01%
- The target gap for the Cost Phase is 1%

Formulation Tweaks Model Size

- We selected 64 variations on a baseline model that represent samplings of real-life situations
- Performance Phase had a time limit of two hours
- Cost Phase had a time limit of two hours
- Solved using CPLEX 12.3
- Model Size:
  - ≈60,000 constraints (pre-solve reduced to ≈ 20,000)
  - ≈ 12,000 variables
    - ≈ 5,000 integer (pre-solve reduced to ≈ 2,100)
    - ≈ 6,000 binaries (pre-solve reduced to ≈ 1,600)

Baseline

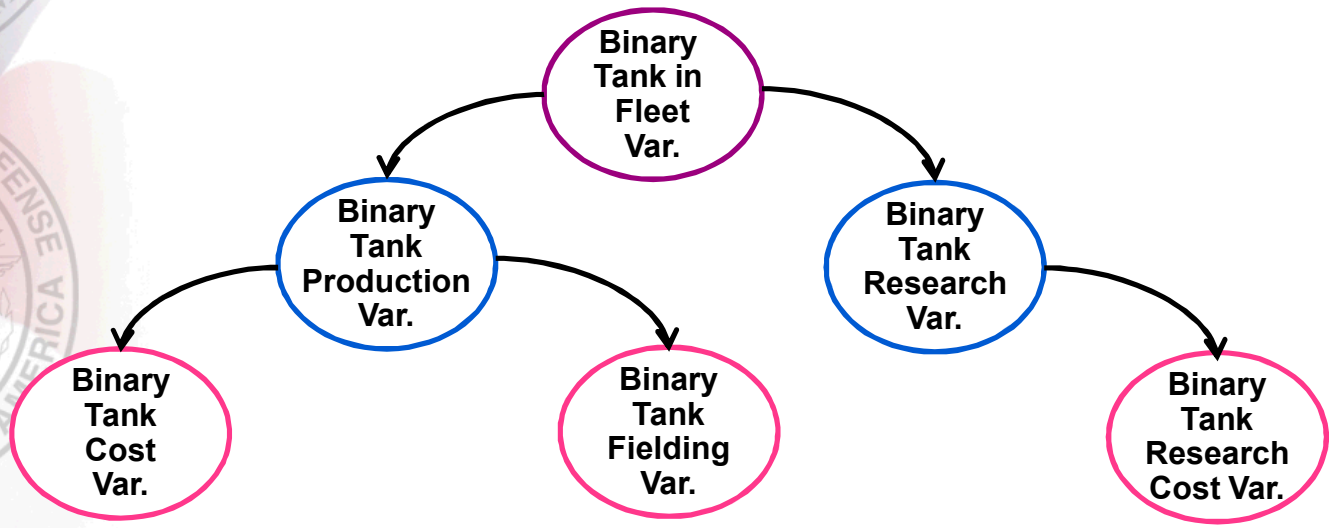




Sandia National Laboratories

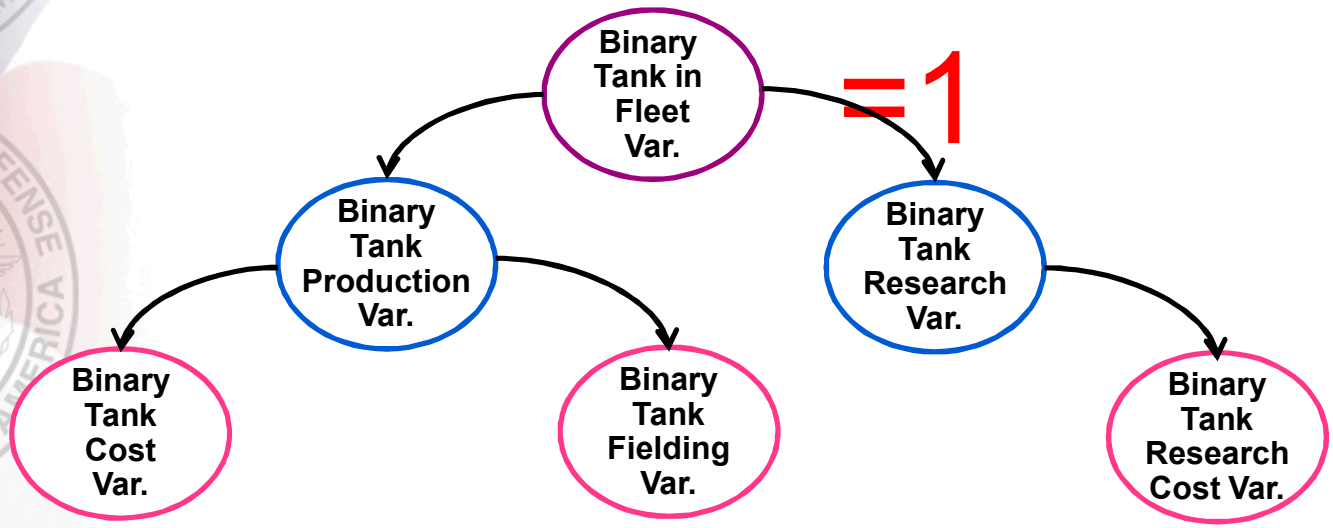
Binary Order Formulation Tweak

- First formulation tweak was to create a hierarchy of binary variables
  - Prior knowledge of the structure of the program allowed us to enforce relationships between binary variables



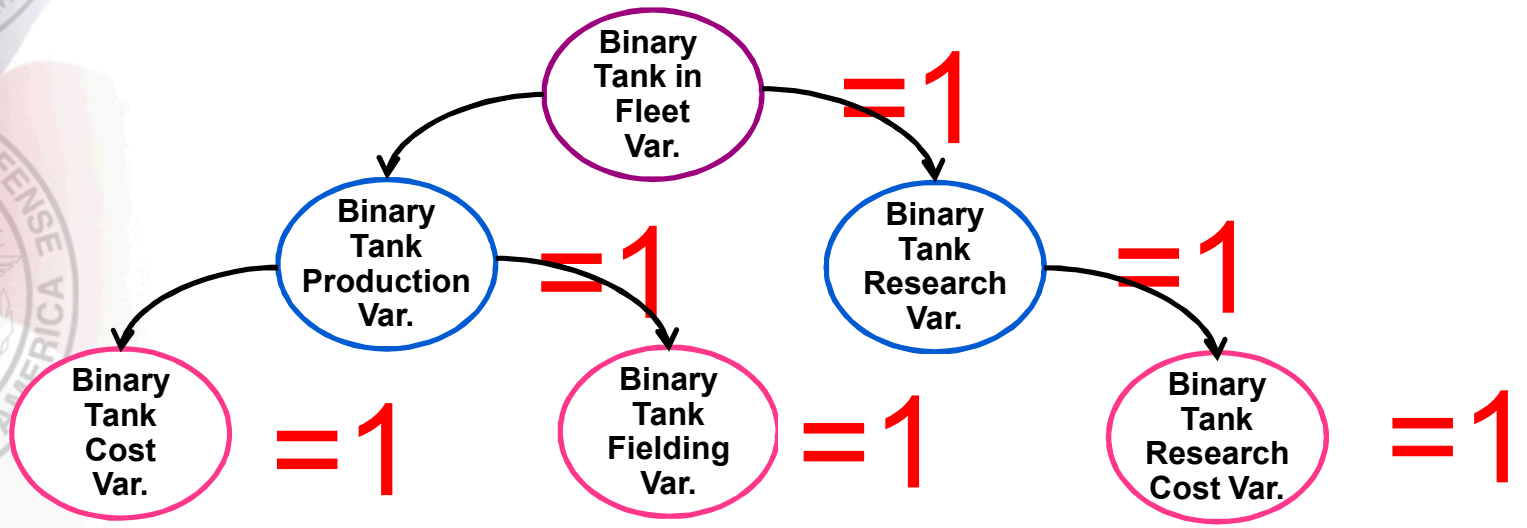
Binary Order Formulation Tweak

- If this Tank is going to appear in the fleet then the binary tank in fleet variable is set to **one**



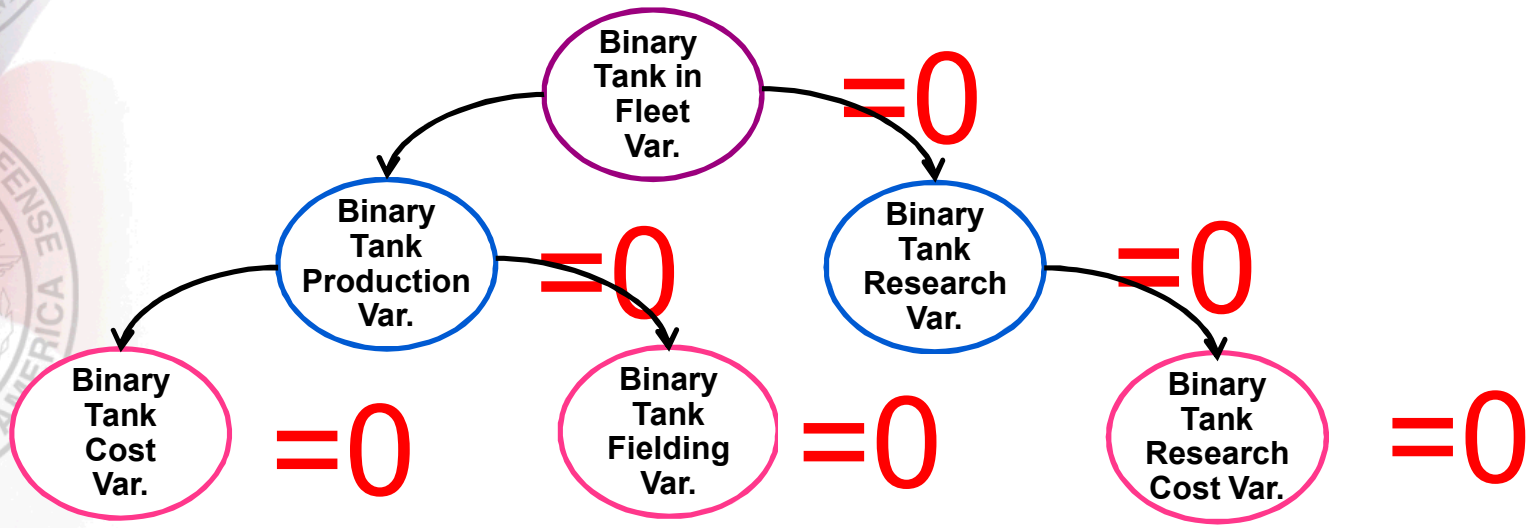
Binary Order Formulation Tweak

- If this Tank is going to appear in the fleet then the binary tank in fleet variable is set to one
- All other binary variables will be set to one

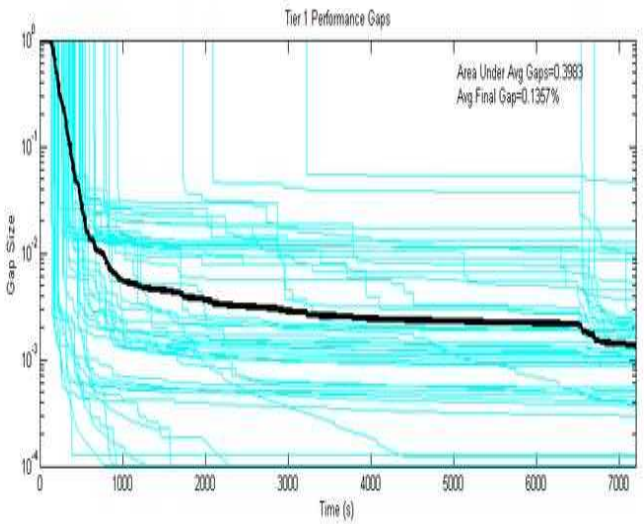


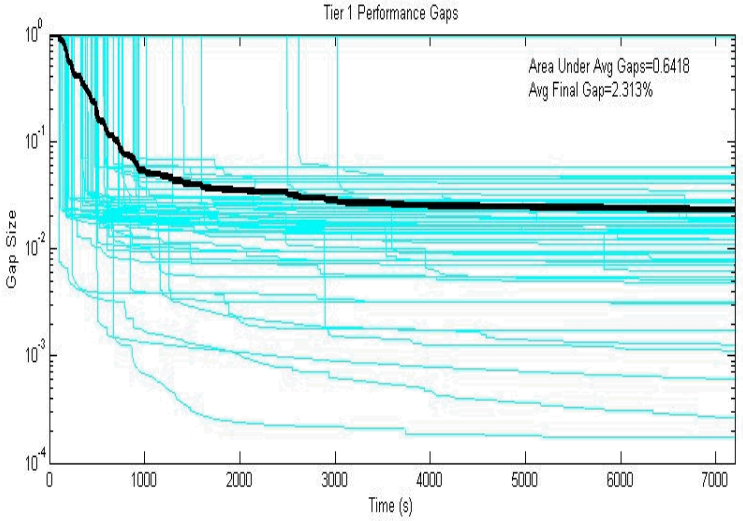
Binary Order Formulation Tweak

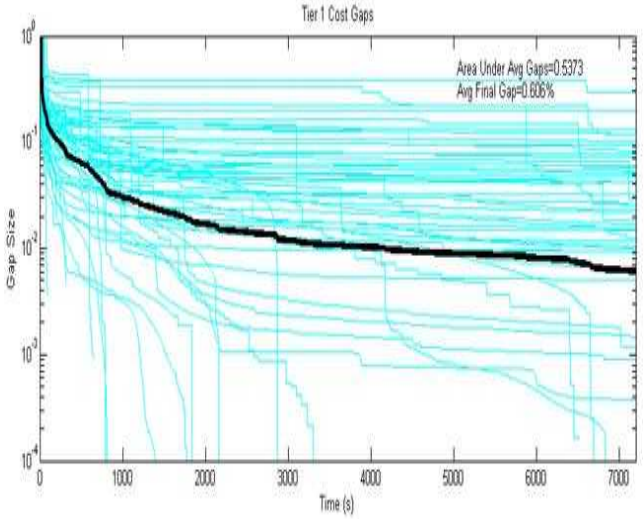
- Similar case if this Tank does not appear in the fleet
- All binary variables are set to zero

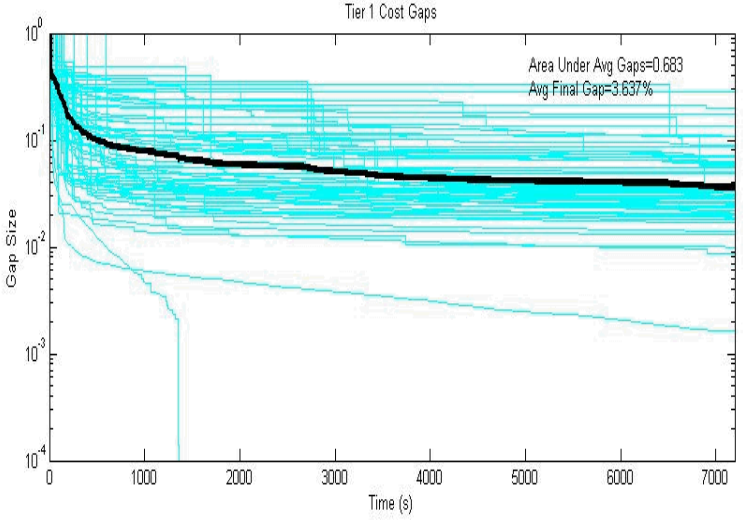


Baseline vs. Binary Tweak







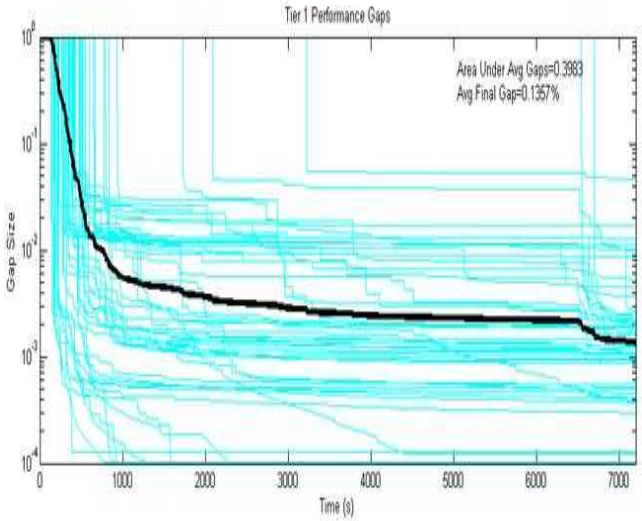


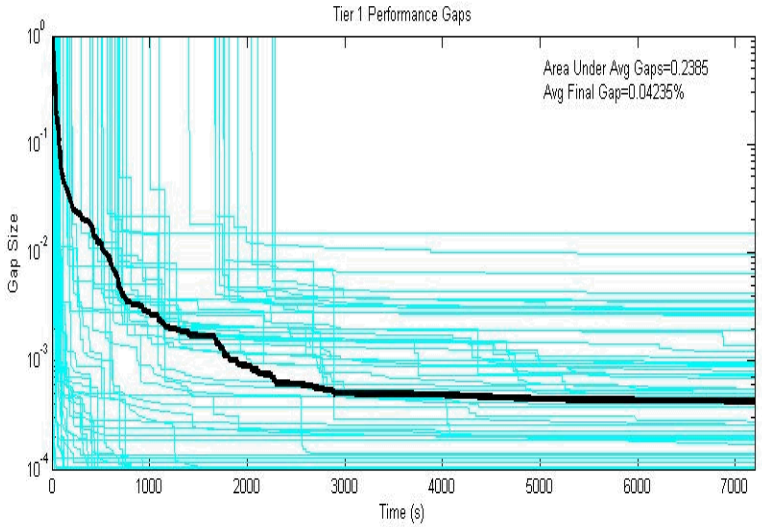
Binary Tweak had about 500 fewer binary variables but did much worse!

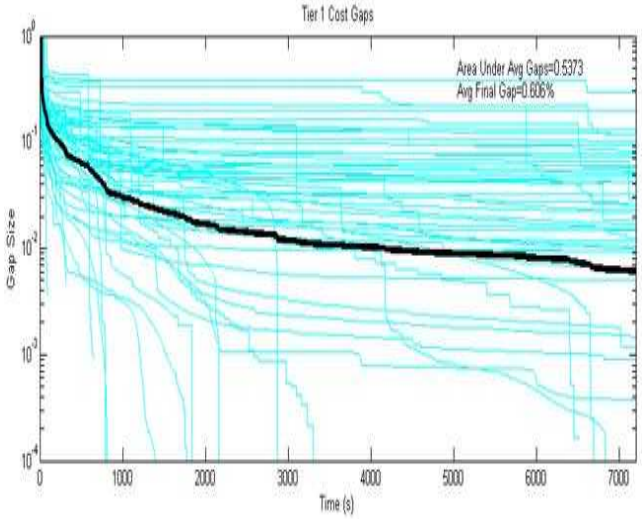
Batch Size Tweak

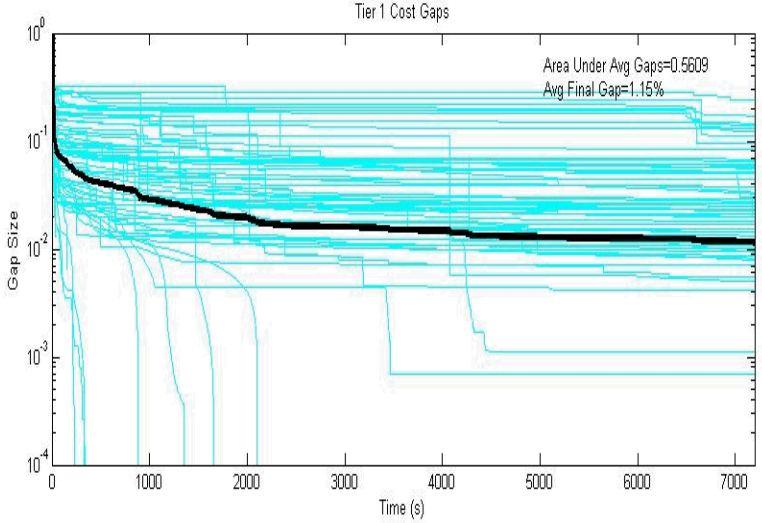
- Our model has many integer variables representing producing, upgrading, and fielding of individual vehicles
- In real-life, these decisions are not made for each individual vehicle but rather for groups of vehicles
- Example
  - Vehicles are fielded in brigade sets
    - 1 brigade = 25 vehicles
    - Fielding variables changed to brigade sizes which have fewer realizations
  - Vehicles are purchased in batch sizes
    - 1 batch = 15 vehicles
    - Purchase variables changed to batch sizes which have fewer realizations

Baseline vs. Group Sized Variables







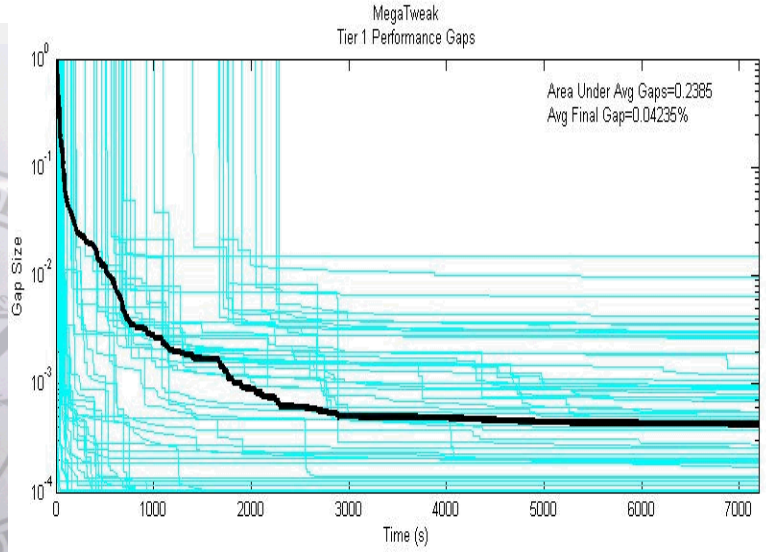


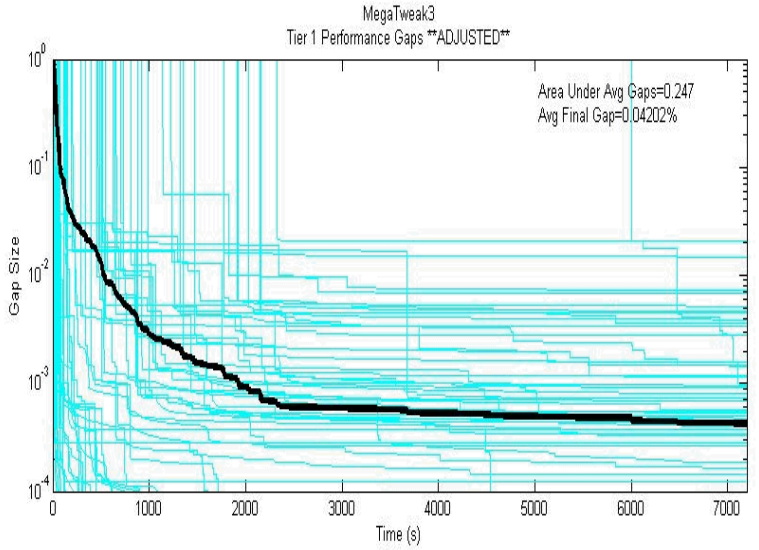
Group sized integer variables had far fewer realizations and helped!

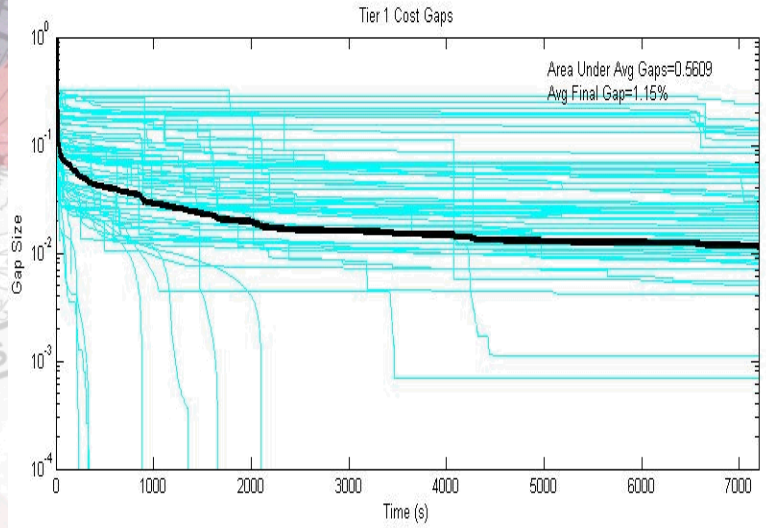
Batch Tweak and Rescaled Objective

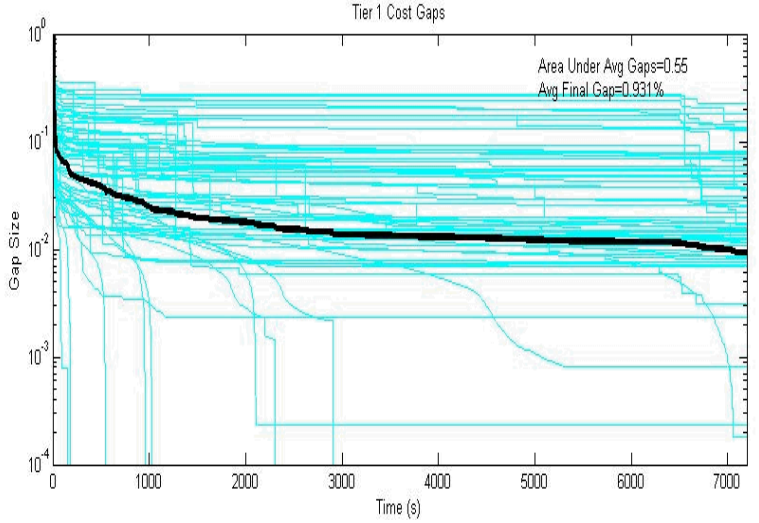
- The objective function value is much larger than any of the coefficients in the constraint matrix even though we have some Big M's in the formulation
- The objective function is rescaled to be more in line with the coefficients and smaller than the Big M's in the formulation
- The goal is to improve numerical stability in the model

Group Sized vs. Rescaled Objective









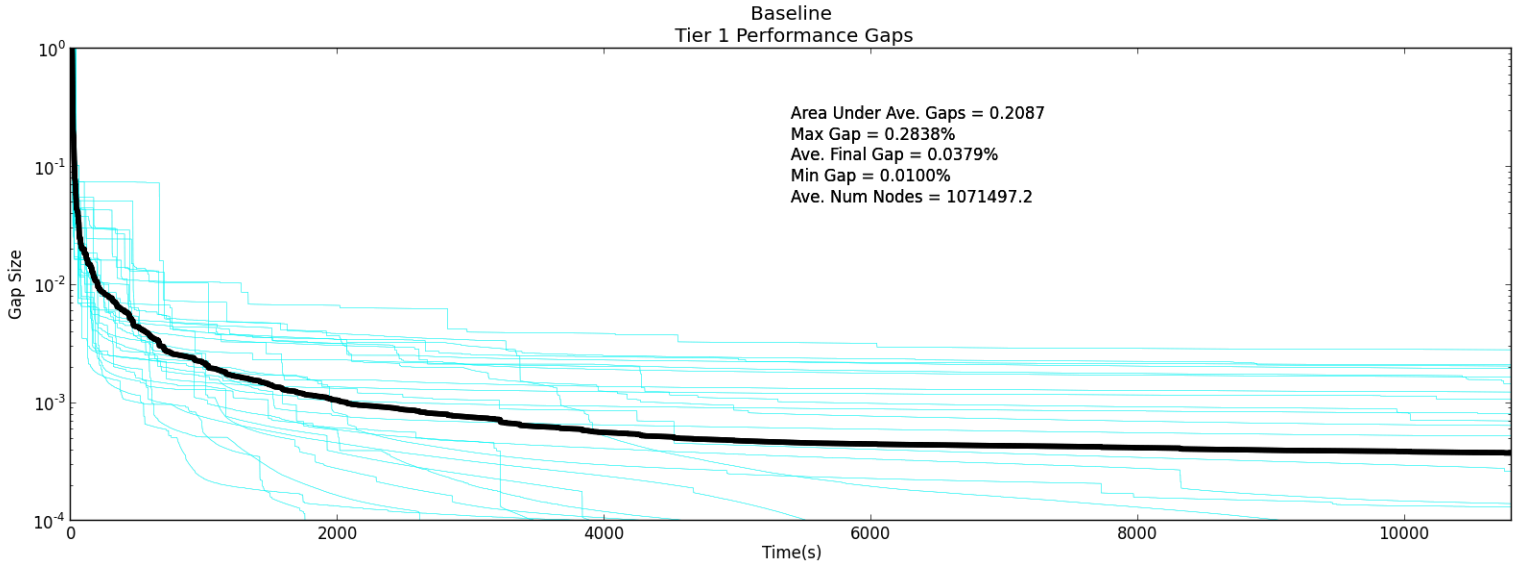
No difference for a rescaled objective function

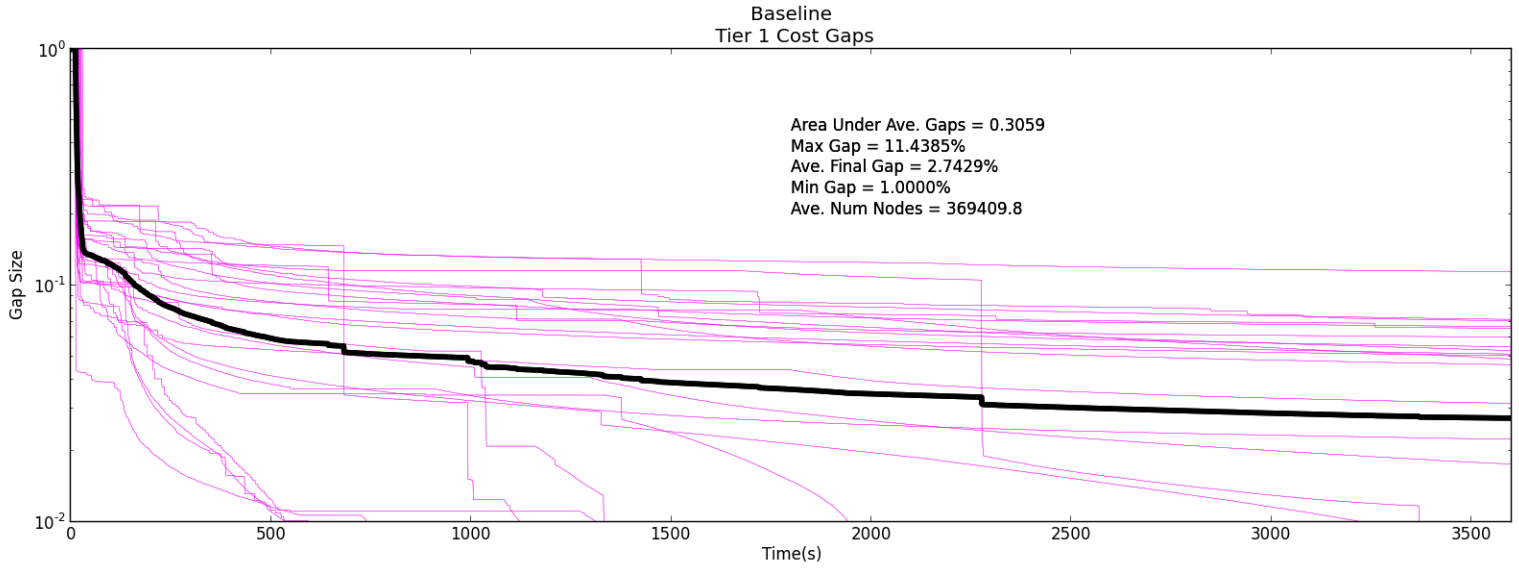
Sandia National Laboratories

CPLEX Parameter Tweaks Model Size

- We selected 12 models of various difficulty
  - Ran each model twice with aggressive parallelism enabled
- Performance Phase had a time limit of three hours Cost Phase had a time limit of one hour
- Solved using CPLEX 12.5
- Model Size:
  - ≈60,000 constraints (pre-solve reduced to ≈ 20,000)
  - ≈ 12,000 variables
    - ≈ 5,000 integer (pre-solve reduced to ≈ 2,100)
    - ≈ 6,000 binaries (pre-solve reduced to ≈ 1,600)

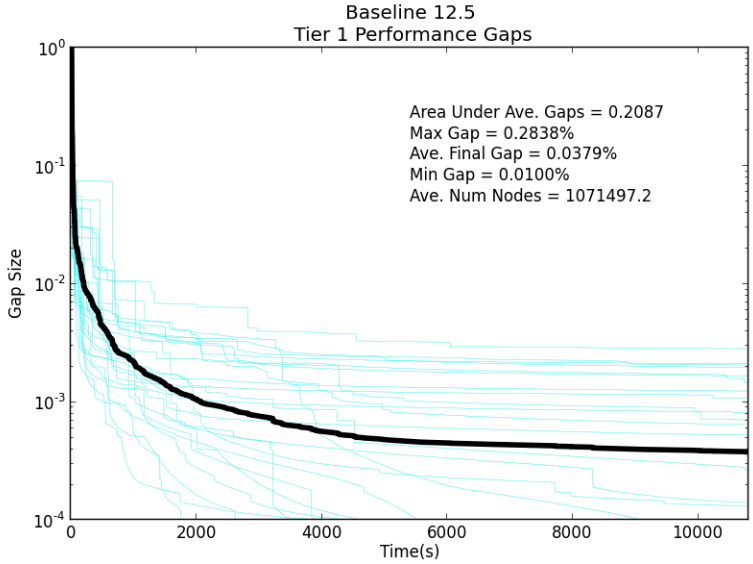
Baseline

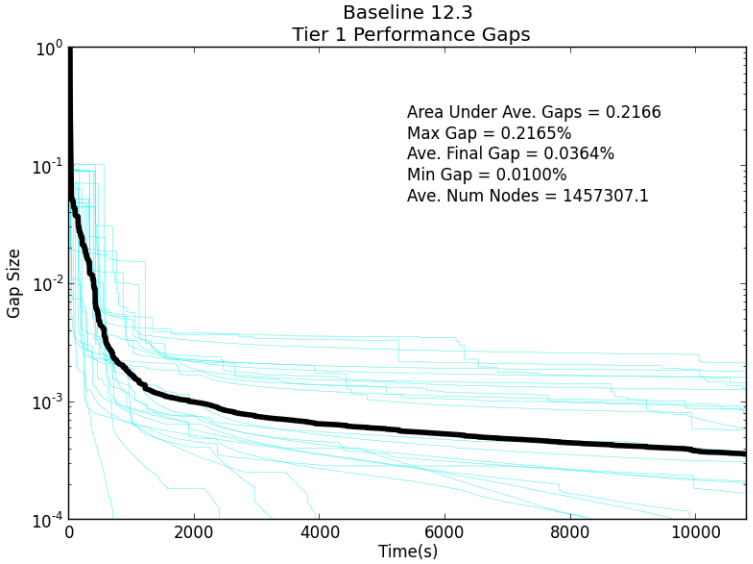


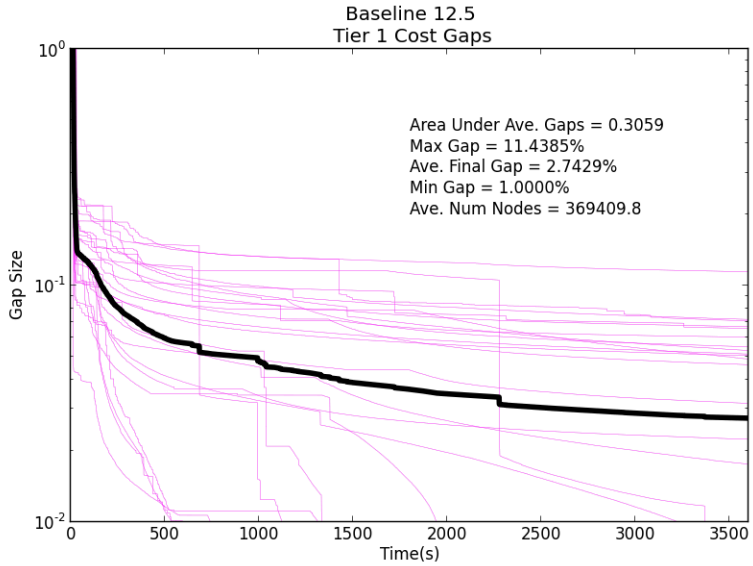


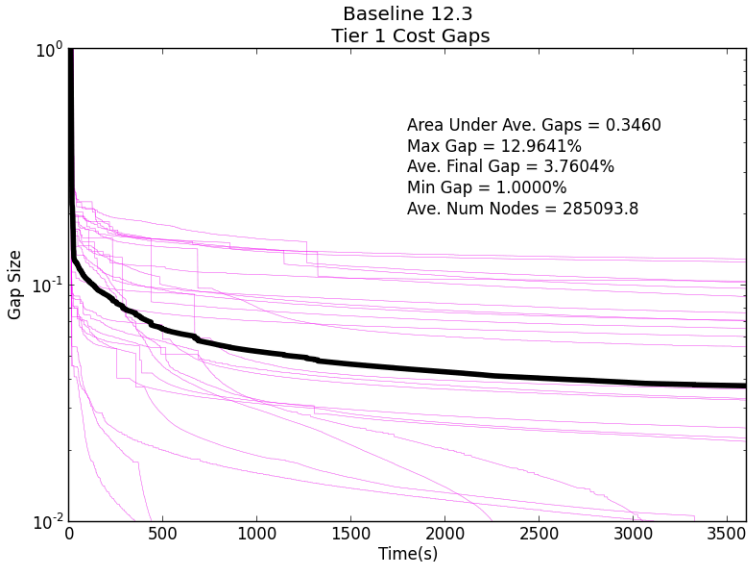
Sandia National Laboratories

Baseline CPLEX 12.5 vs. CPLEX 12.3









Different bends in the graphs

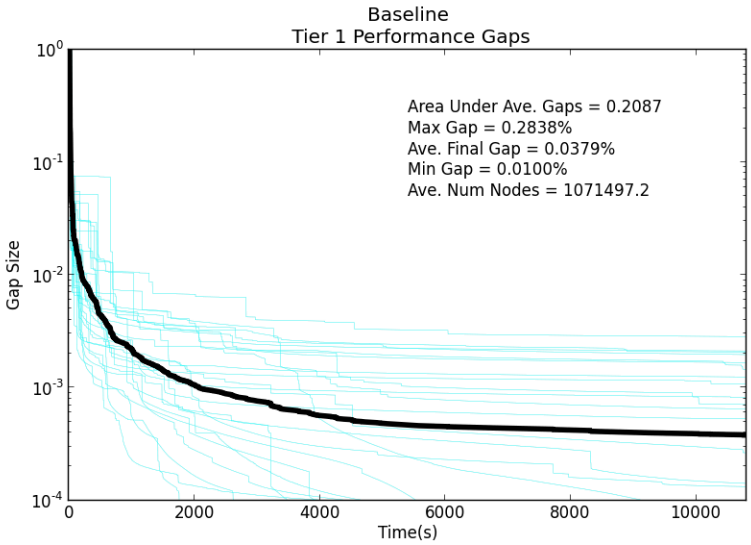
CPLEX 12.5 vs. CPLEX 12.3

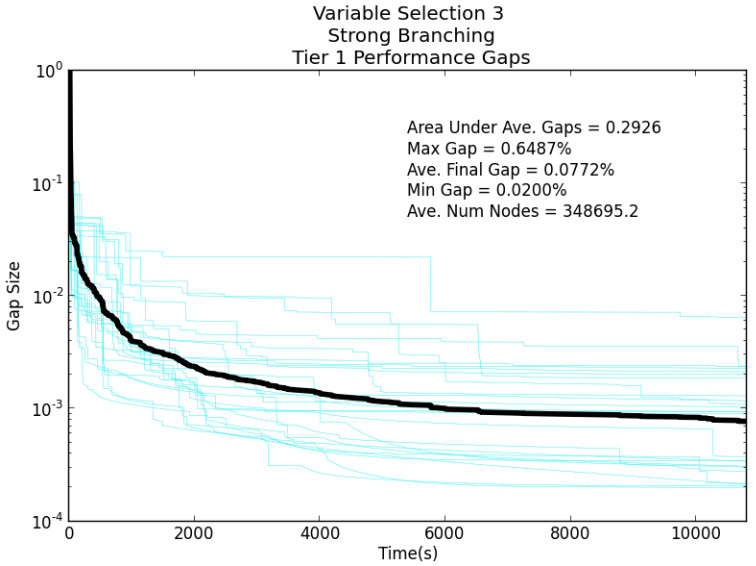
- Within the last two months, we upgraded CPAT to use CPLEX 12.5 instead of 12.3
- The gaps are fairly close with CPLEX 12.5 performing slightly better in the Performance Phase and a little better more in the Cost Phase
- Running the test set of 24 models in CPLEX 12.3 generally resulted in 4-6 runs having errors ($\approx 30\%$)
  - Ran out of memory (memory management issue)
  - Singular basis (numerical instability)
- CPLEX 12.5 has only had 2 errors!
  - Over 200 model instances have run (< 0.5%)!

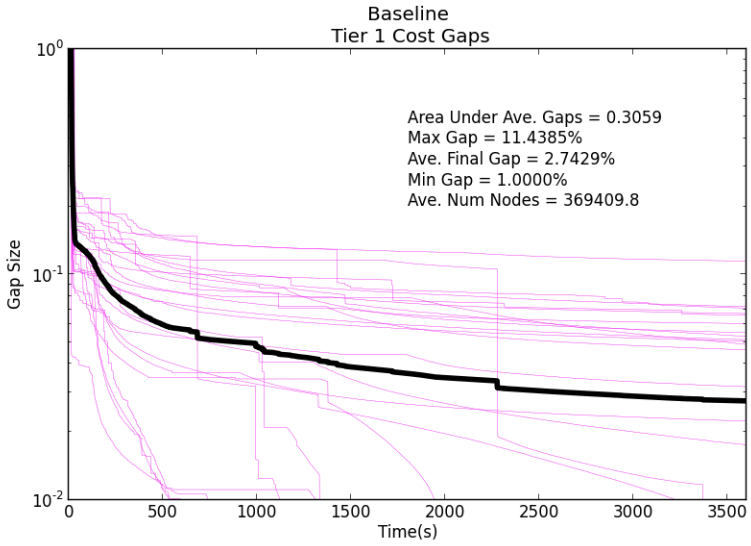
Strong Branching

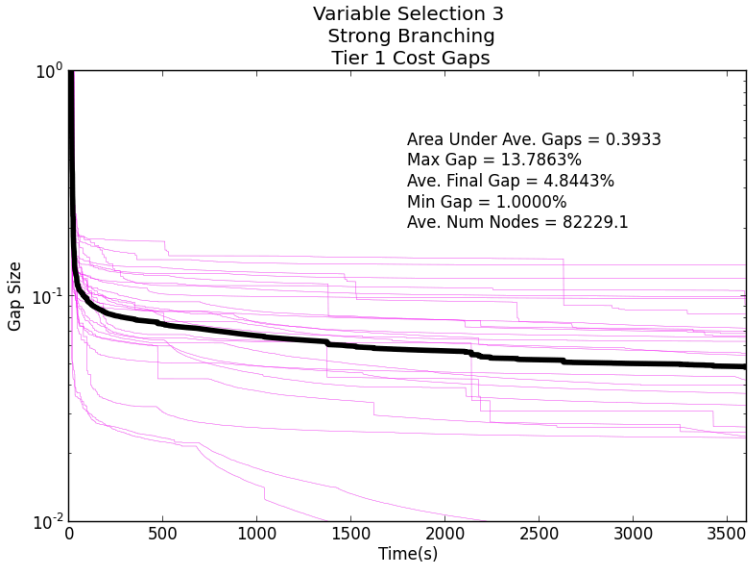
- CPLEX allows you choose a criteria for selecting the next branching variable at each node
- Strong branching (Variable Selection = 3) causes variable selection based on partially solving a number of sub-problems with tentative branches to see which branch is the most promising.
- CPLEX documentation says that, “this strategy can be effective on large, difficult MIP problems.”

Strong Branching







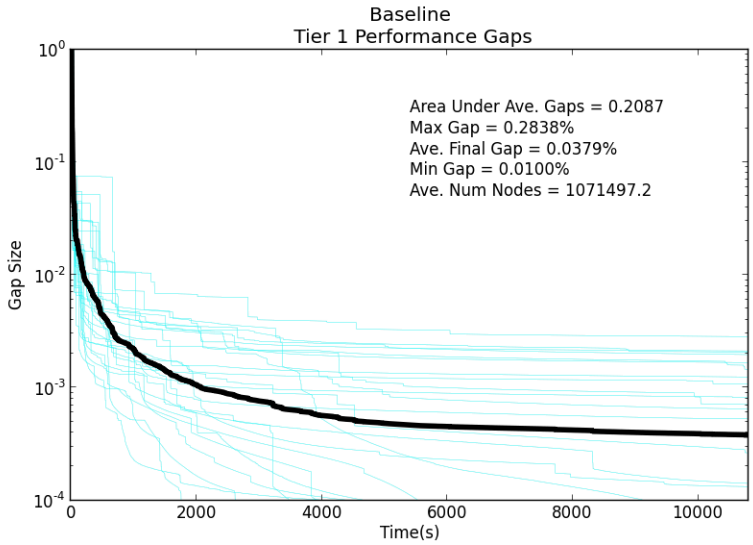


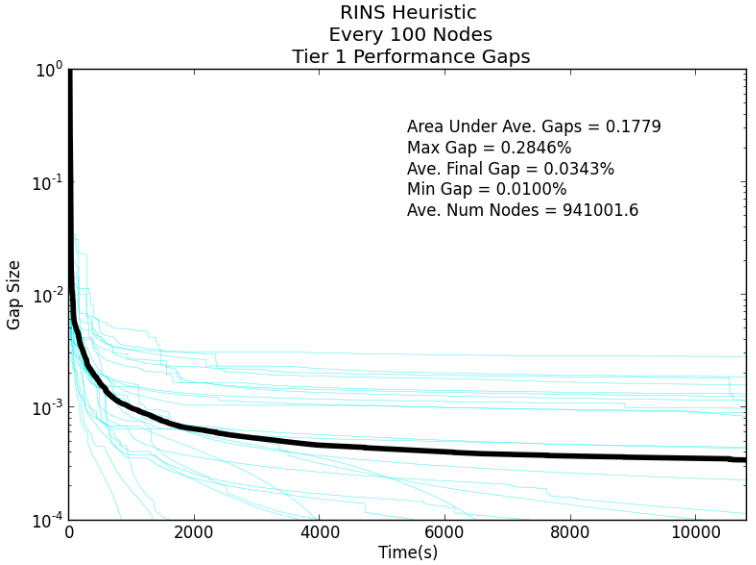
Strong branching did not help!

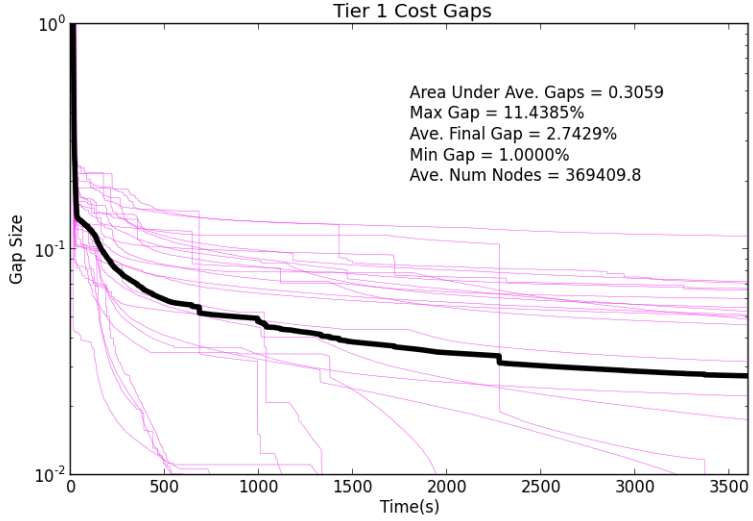
RINS Heuristic

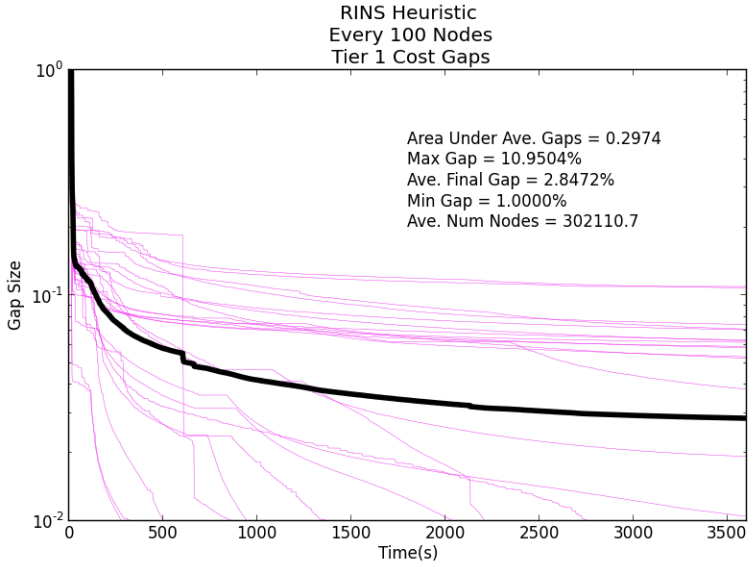
- Relaxation induced neighborhood search (RINS) heuristic attempts to improve the best IP solution found so far
- CPLEX automatically applies the RINS heuristic at intervals it chooses
- Helps find “high quality” solutions but may take time to do it
- Set CPLEX to use RINS Heuristic every 100 nodes in the branch and bound tree

RINS Heuristic







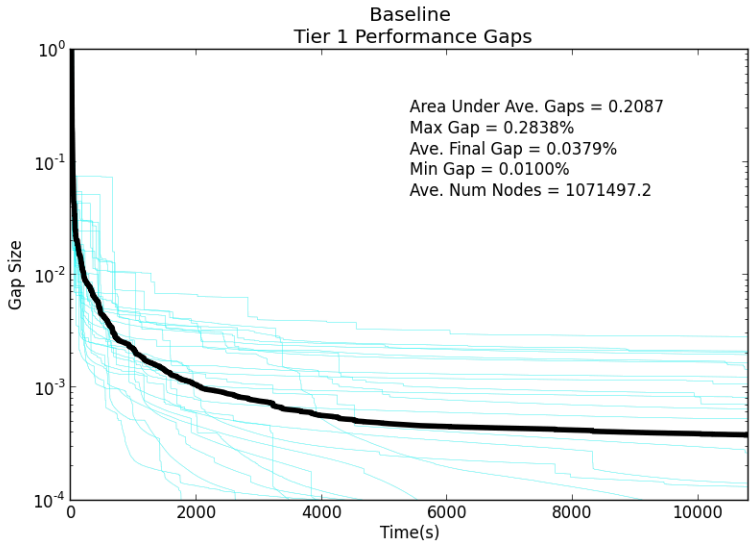


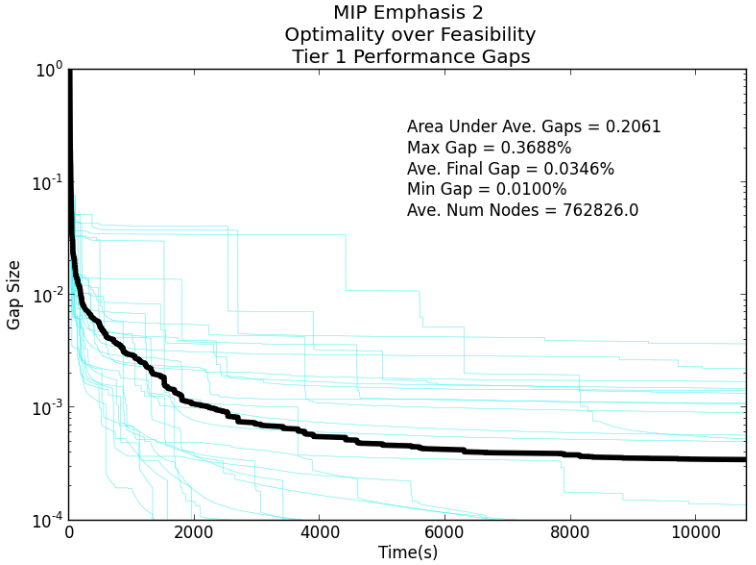
RINS Heuristic helps early on, but is not as helpful later

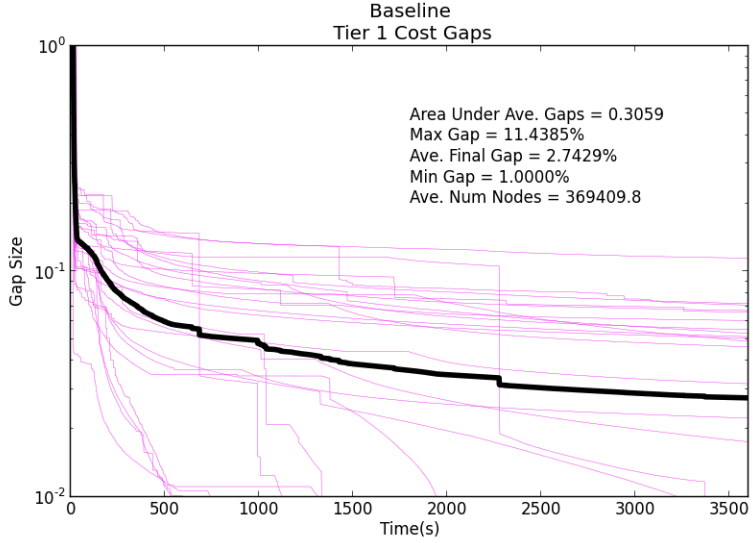
MIP Emphasis 2

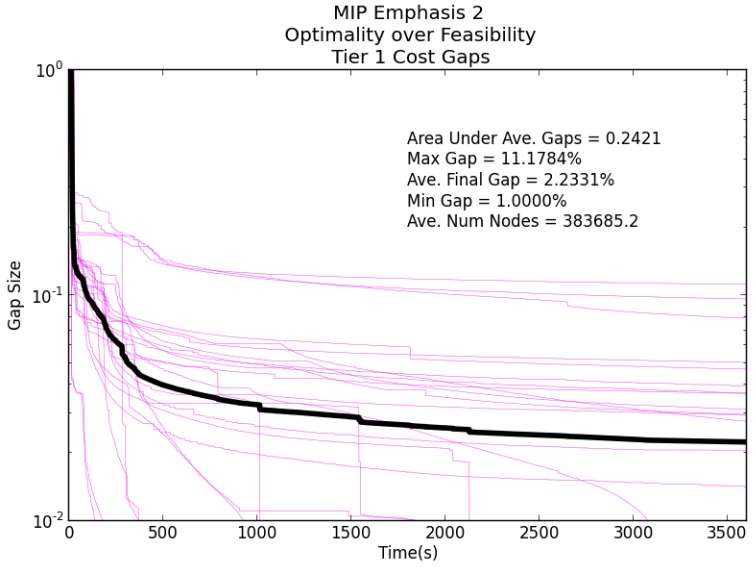
- "Emphasize Optimality over Feasibility"
- Documentation is vague as to what CPLEX does when this is set
  - Says, "less effort may be applied to finding feasible solutions early"
- What does CPLEX do? (my thoughts)
  - Does not actively try to find feasible solutions
    - Just stumbles upon them
  - Spends more time finding a better upper bound on the problem

MIP Emphasis 2







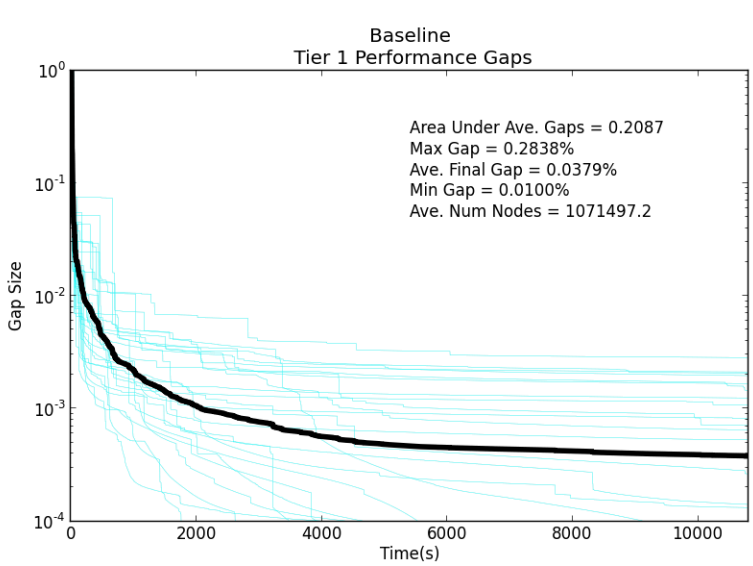


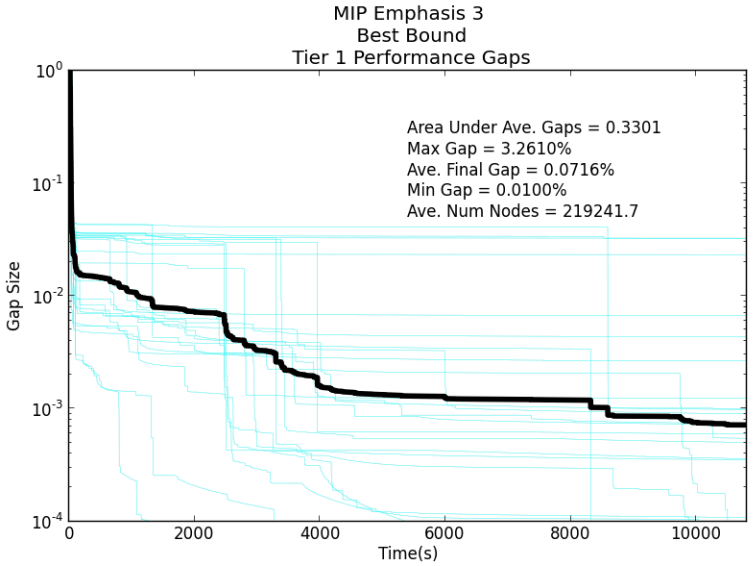
A slower bend in the curve

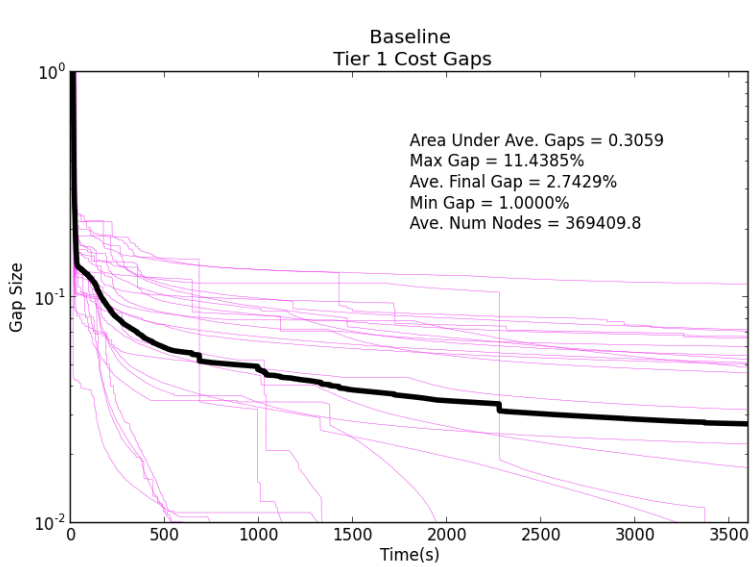
MIP Emphasis 3

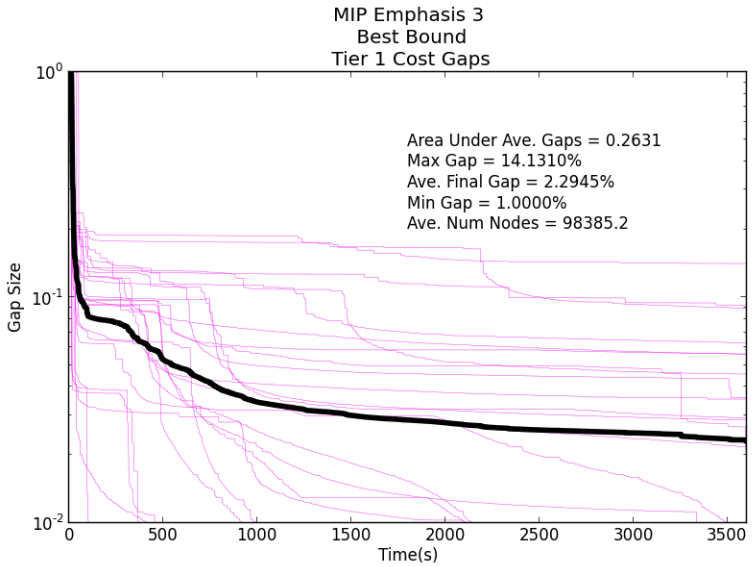
- "Emphasize Moving Best Bound"
- Documentation says a little more as to what CPLEX does when this set
  - Says, "even greater emphasis is places on proving optimality through moving the best bound value"
  - Says, "the detection of feasible solutions along the way becomes almost incidental"
- CPLEX just tries to move the LP value lower
- It does not even attempt to look for feasible solutions

MIP Emphasis 3







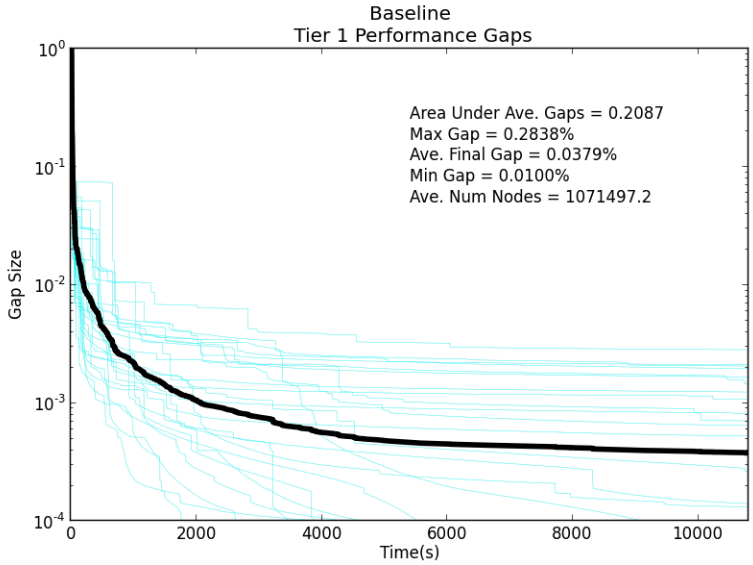


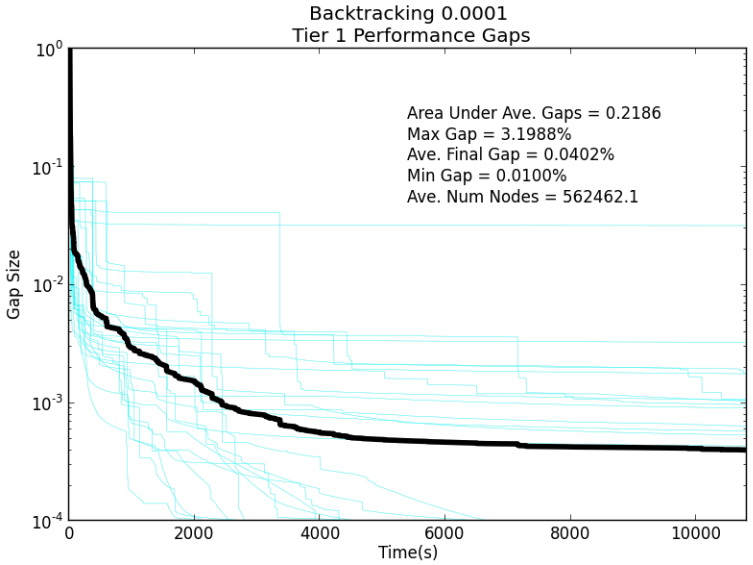
Not the best idea to only focus on lowering the upper bound

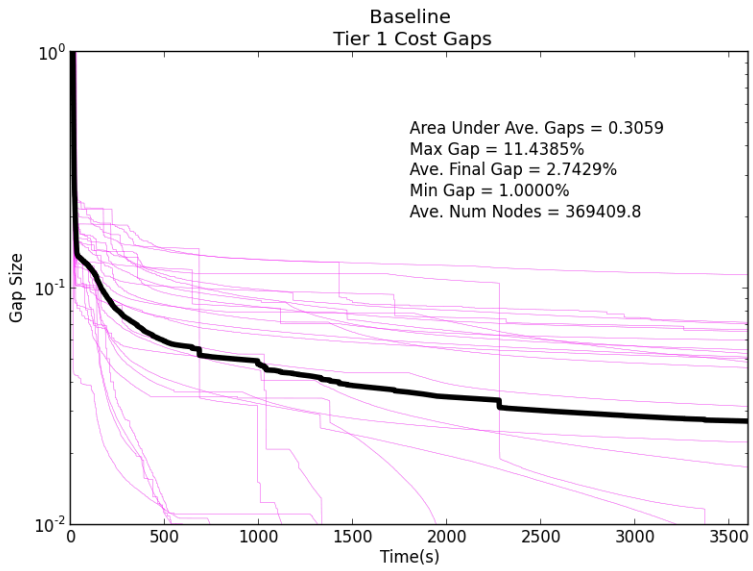
Backtracking 0.0001

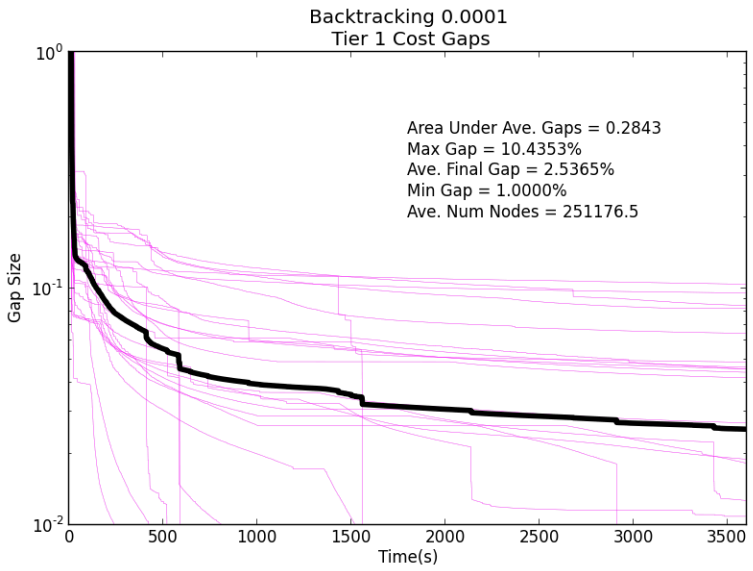
- Determines when CPLEX decides to backtrack back up the search tree
- CPLEX is allowed to backtrack once the difference between the best integer solution and the best bound at the current node is above this threshold
- A low backtracking value has CPLEX perform more of a breadth-first search
- A high backtracking value has CPLEX perform more of a depth-first search
- Default CPLEX value is 0.9999

Backtracking 0.0001



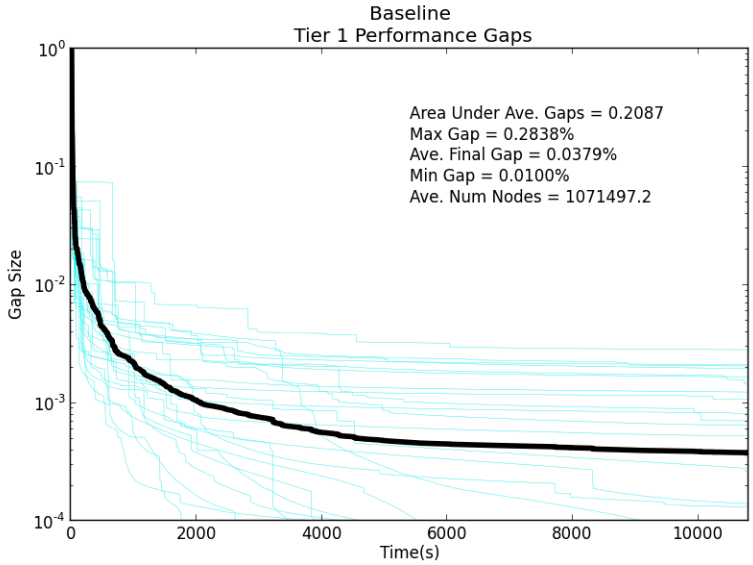


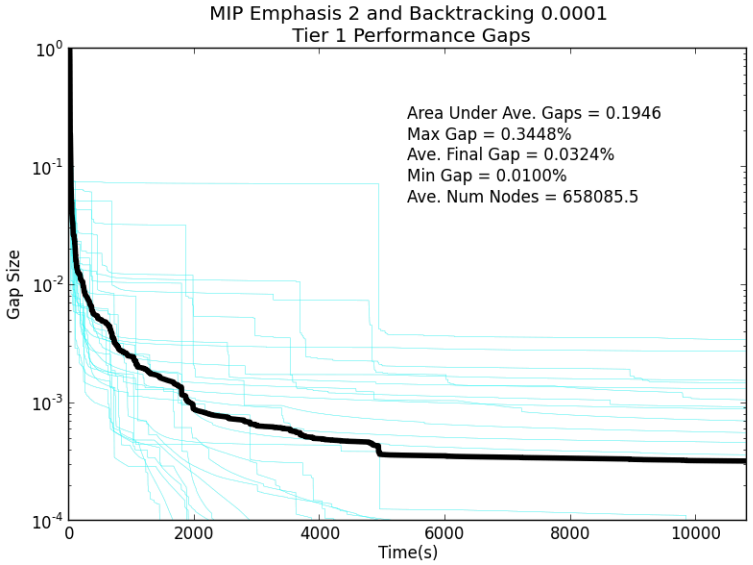


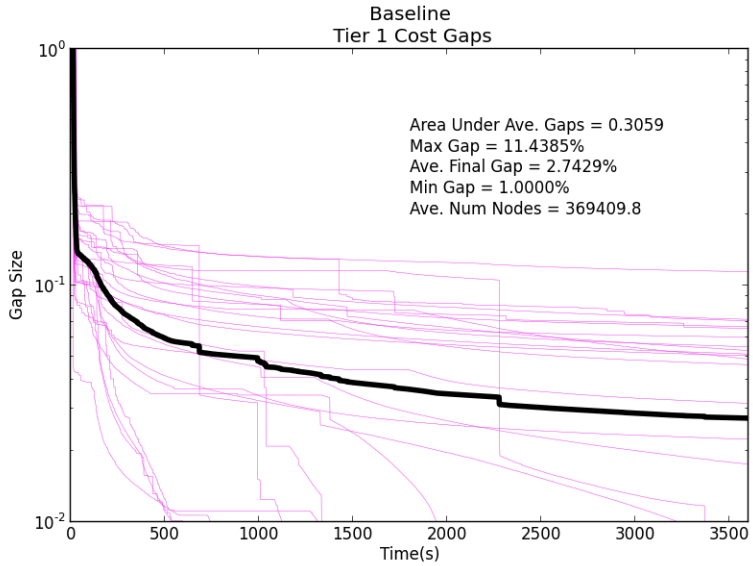


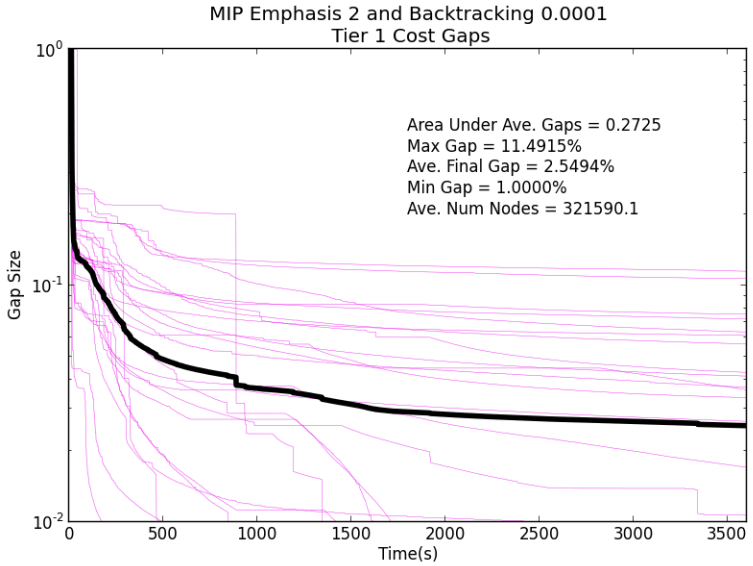
Backtracking helped make the easier problems easier

MIP Emphasis 2 & Backtracking 0.0001









Tried MIP Emphasis 2 and Backtracking 0.0001 together

Sandia National Laboratories

Final Thoughts

- We have been able to drastically improve the runtime and final gap of CPAT models via formulation changes and parameter settings even while the problem has gotten more computationally difficult
- This allows us to give quality results to the Army about their acquisition decisions for the next 25 – 30 years
- What we have learned
  - CPLEX 12.5 does a better job with memory management and numerical stability
  - Recommended settings do not always help in optimization
    - Some have the opposite effect (MIP Emph = 3)
  - Settings that give CPLEX more freedom seem to help (Backtracking = 0.0001)
  - Efficient formulations help more than parameter tweaks (using brigade/batch sized variables)
    - Clever formulations may or may not help in optimization (binary structure)