An Illustrated Guide to Finding Solutions

Philipp M. Christophel

joint work with Jeff Day, Baris Kacar and others

Operations Research R&D –
Advanced Analytics Div., SAS Institute Inc.

EWO Seminar, 9-17-2013

sas THE POWER TO KNOW®

Agenda

1. Introduction
2. Modeling and Data, Data, Data
3. The Magic of Mixed Integer Programming
4. Homebrew Solutions
5. Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

Agenda

1 Introduction

2 Modeling and Data, Data, Data

3 The Magic of Mixed Integer Programming

4 Homebrew Solutions

5 Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

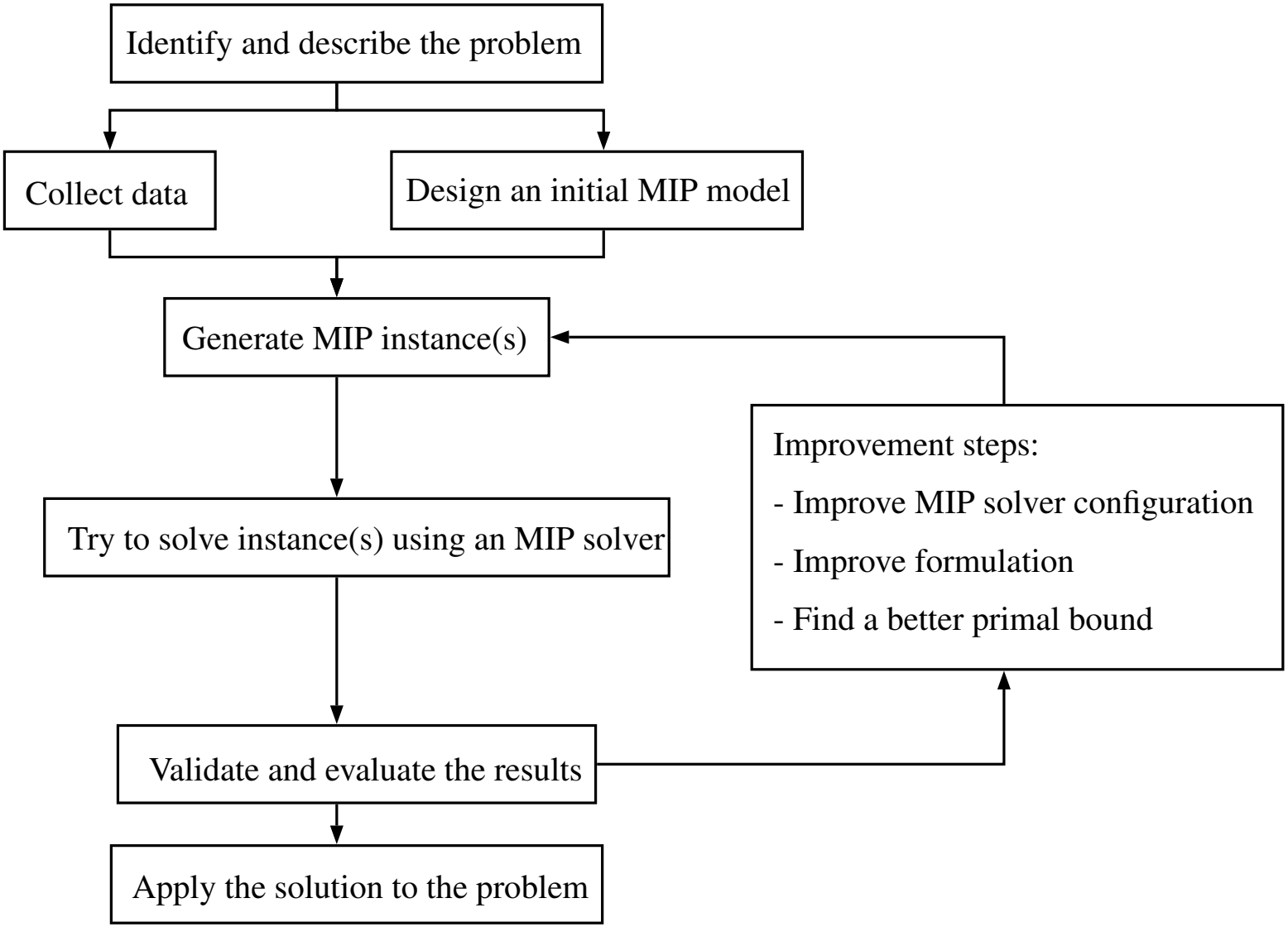
Mixed Integer (Linear) Programming

$$\begin{aligned} \min \quad & cx \\ & Ax \leq b \\ & x_i \in \mathbb{Z} \quad \forall i \in I \end{aligned}$$

- A lot of problems can be modeled
- High-quality, reliable software available
- The process of modeling gives you insight!

Copyright ©2011, SAS Institute Inc. All rights reserved.

From Business to Algebra and Back Again



Copyright ©2011, SAS Institute Inc. All rights reserved.

What You Should Have In Your Toolbox

- Data processing software
  - Read and write data
  - Reporting
  - Visualization tools
  - Forecasting
- An algebraic modeling language
- Optimization solvers
  - Linear programming (LP)
  - Quadratic Programming (QP)
  - Mixed integer programming (MIP)
  - Constraint Programming (CP)
  - Non-linear programming (NLP)
  - Specialized (e.g. network algorithms)
- Simulation software

- Base SAS
- SAS/GRAPH
- SAS Forecast Server
- SAS/OR
  - PROC OPTMODEL
- SAS Simulation Studio

Copyright ©2011, SAS Institute Inc. All rights reserved.

What You Should Have In Your Toolbox

- Data processing software
  - Read and write data
  - Reporting
  - Visualization tools
  - Forecasting
- An algebraic modeling language
- Optimization solvers
  - Linear programming (LP)
  - Quadratic Programming (QP)
  - Mixed integer programming (MIP)
  - Constraint Programming (CP)
  - Non-linear programming (NLP)
  - Specialized (e.g. network algorithms)
- Simulation software

- Base SAS
- SAS/GRAPH
- SAS Forecast Server
- SAS/OR
  - PROC OPTMODEL
- SAS Simulation Studio

Copyright ©2011, SAS Institute Inc. All rights reserved.

What else does SAS do?

- SAS = Analytics

  "The IDC research showed analytics leader SAS with a 36.2 percent share of the 2012 global advanced analytics [market ...]. In fact, all of the other named advanced analytics providers' market share combined totaled just 24.9 percent."

- SAS Solutions
  - » SAS Marketing Optimization, SAS Pack Optimization, SAS Revenue Optimization ...

- SAS is customer-driven
  - » SAS R&D Advanced Analytics and Optimization Services

Copyright ©2011, SAS Institute Inc. All rights reserved.

Agenda

1 Introduction

2 Modeling and Data, Data, Data

3 The Magic of Mixed Integer Programming

4 Homebrew Solutions

5 Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

The Model Matters

- Lots of literature on modeling available
  - Williams: "Model Building in Mathematical Programming" (OPTMODEL code available)
  - Pochet and Wolsey: "Production Planning By Mixed Integer Programming"
- What you should do
  - Model the decisions that matter
  - Research model improvements
  - If you can't model it: simplify, relax, approximate
  - Know, clean and use your data
  - Try alternative formulations
- ... and WHAT NOT!
  - Don't use big and small numbers
  - Don't use fractions
  - Don't expect a model to scale

Copyright ©2011, SAS Institute Inc. All rights reserved.

But MY problem is... nonlinear!

- Many problems can be linearized and still give useful information
- Sometimes the non-linearity is just in part of the problem

Example

A consumer goods company wants to solve a multi-product batch mixing problem involving nonlinear chemical processes. The SAS NLP solver is used to compute the objective value for a number of batches. Then the MILP solver is used to solve a set partitioning problem deciding which end-product to make from which batch.

Copyright ©2011, SAS Institute Inc. All rights reserved.

But MY problem is... stochastic!

- Stochastic programming and robust optimization still use MILP/LP solvers
- Use forecast data for your model

Example

A bank wants to optimize their money trucks that restock ATMs. The goal is to minimize "cash out" events. The demand for each ATM is forecast using SAS Forcasting Server, then an MILP is solved with SAS/OR PROC OPTMODEL.

Copyright ©2011, SAS Institute Inc. All rights reserved.

But MY problem is... too big!

- Too many variables: Change the scope, e.g. fewer time periods, solve each product separately, ...
- Too many constraints: Ignore constraints that do not influence feasibility

Example

A company wants to decide on a product launch schedule. The prices realized depend 100% on binary product launch decisions. But the rules of this dependency are very complicated. Drop or approximate the complicated constraints. Find a solution for the binary decisions using the SAS/OR MILP solver, then compute the true prices for all real world constraints using PROC OPTMODEL.

Copyright ©2011, SAS Institute Inc. All rights reserved.

Agenda

1 Introduction

2 Modeling and Data, Data, Data

3 The Magic of Mixed Integer Programming

4 Homebrew Solutions

5 Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

Just solve it!

```
NOTE: The MILP presolver value AUTOMATIC is applied.
NOTE: The MILP presolver removed 66 variables and 178 constraints.
NOTE: The MILP presolver removed 264 constraint coefficients.
NOTE: The MILP presolver modified 0 constraint coefficients.
NOTE: The presolved problem has 2081910 variables, 92533 constraints, and
      8271684 constraint coefficients.
NOTE: The MILP solver is called.
          Node  Active    Sols    BestInteger      BestBound       Gap    Time
             0       1       1    117.0000000    117.0000000    0.00%      94
             0       0       1    117.0000000    117.0000000    0.00%      94
NOTE: Optimal.
NOTE: Objective = 117.
NOTE: PROCEDURE OPTMILP used (Total process time):
      real time           1:56.26
      user cpu time       1:49.44
```

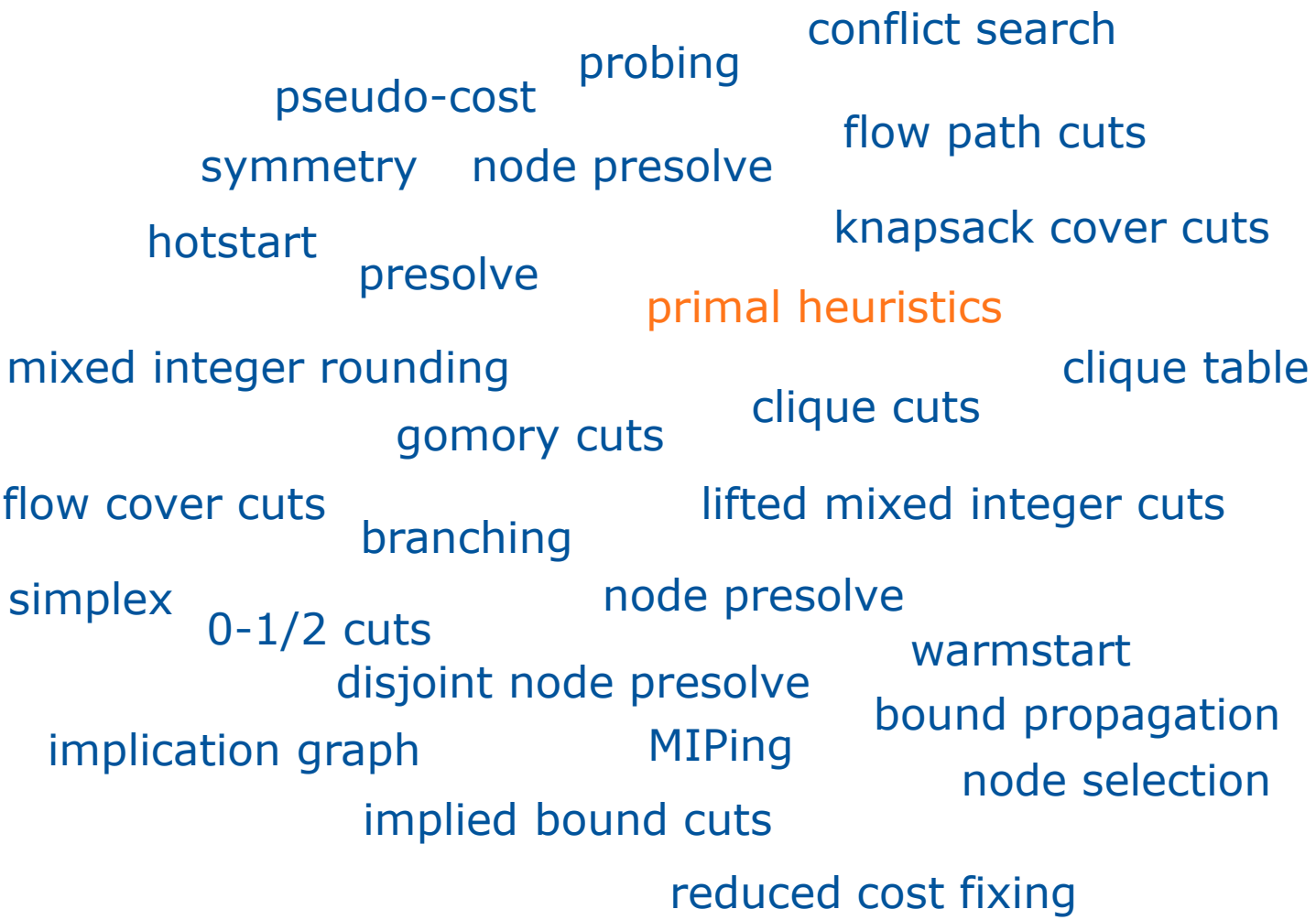
Copyright ©2011, SAS Institute Inc. All rights reserved.

Why does it work?

- MILP solvers use methods based on 50+ years of research
- MILP solvers are tailored to specific types of problems that come up a lot
- MILP solvers have problem specific techniques that are turned on and off automatically
- MILP solvers detect common substructures and try to exploit them
- MILP solvers "borrow" techniques from other fields like constraint programming and meta-heuristics

Copyright ©2011, SAS Institute Inc. All rights reserved.

Techniques in the SAS MILP solver



Copyright ©2011, SAS Institute Inc. All rights reserved.

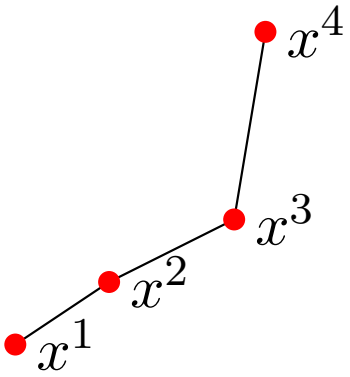
Types of Primal Heuristics

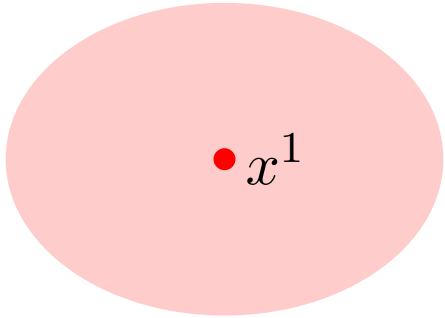
$$cx < c\bar{x} \qquad (1)$$
$$Ax \leq b \qquad (2)$$
$$x_i \in \mathbb{Z} \quad \forall i \in I \qquad (3)$$

- Starting heuristics
  - input satisfies: (1) and (2) but not (3)
- Improvement heuristics
  - input satisfies: (2) and (3) but not (1)
- Repair heuristics
  - input satisfies: (3) and (1) but not (2)
- Special heuristics
  - No requirements

Copyright ©2011, SAS Institute Inc. All rights reserved.

Underlying Principles of Primal Heuristics





Line Search

- Rounding
- Diving
- Feasibility Pump
- Pivoting
- ...

Neighborhood Search

- MIPing
- Genetic Algorithms
- ...

Copyright ©2011, SAS Institute Inc. All rights reserved.

Solver Components That Support Heuristics

- Dual and primal simplex implementations focused on warm and hot start
- Implication graph and clique table to propagate bound changes

Example

Formulation

$$x_1 + x_2 + x_3 \leq 1$$
$$y \leq 5x_1$$
$$x \in \{0, 1\}$$
$$y \in \mathbb{R}$$

Solver stores

$$x_1 = 1 \rightarrow \quad x_2 = 0 \wedge x_3 = 0$$
$$x_2 = 1 \rightarrow \quad x_1 = 0 \wedge x_3 = 0$$
$$x_3 = 1 \rightarrow \quad x_1 = 0 \wedge x_2 = 0$$
$$x_1 = 0 \rightarrow \quad y \leq 0$$

Copyright ©2011, SAS Institute Inc. All rights reserved.

A Heuristics Framework

- The framework manages the heuristics: i.e it stores, controls and reports
- All heuristics are categorized by
  - Type: starting, improvement, repair and special
  - Speed: very fast, fast, moderate, slow
- The framework usually manipulates categories of heuristics
  - Independent from which heuristics are actually used
- Two solution pools store solutions
  - Improvement pool
  - Repair pool
- The framework is controlled by the heuristics strategy

Copyright ©2011, SAS Institute Inc. All rights reserved.

Why does it (sometimes) not work?

- MILP solvers by default try to solve problems to optimality
  - » Heuristics sometimes help, but might also be a waste of time
  - » For the SAS MILP solver, turning off all heuristics means only slightly worse performance in terms of solving to optimality
  - » But without heuristics we can't find any solution to a lot of hard problems!
  - » Play with the options!
- Very complex problems with many different (overlapping) structures
- Problem structures that MILP solver developers have never seen before
- Very few solutions or almost infeasible problems
- Bad modeling

Copyright ©2011, SAS Institute Inc. All rights reserved.

Agenda

1 Introduction

2 Modeling and Data, Data, Data

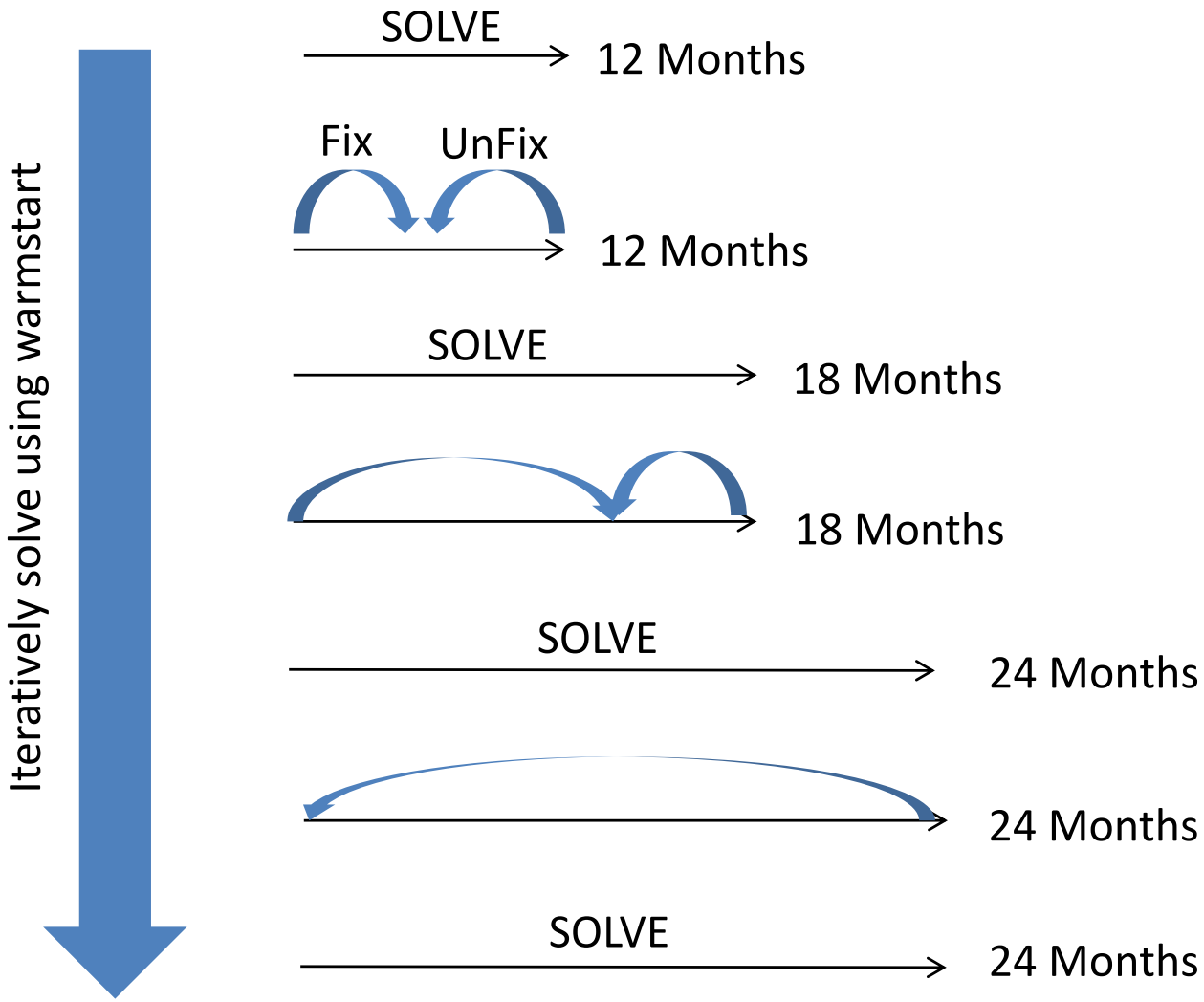
3 The Magic of Mixed Integer Programming

4 Homebrew Solutions

5 Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

A Fairly General Heuristic Idea (Relax-and-fix)



Copyright ©2011, SAS Institute Inc. All rights reserved.

Practical Example (1 of 2)

```
/* Simplify formulation.  */
drop HP;
for{<l,s> in MOVES} Move[l,s].lb = 0;

/* Add dummy variables for lower bounds on moves.  */
var lbMove{<l,s> in MOVES} >= 0 <= moveLB[l,s];

/* Create objective to minimize infeasibility from bounds.  */
min FeasLB = sum{<l,s> in MOVES} (moveLB[l,s] - lbMove[l,s]);

/* Fix variables for now */
fix Move;
fix lbMove;

/* Initialize PERIODS set */
[...]
```

Copyright ©2011, SAS Institute Inc. All rights reserved.

Practical Example (2 of 2)

```
/* Solve simplified problem by truck type.  */
for{l in TRUCK, <a,b> in PERIODS} do;
   for{<(l),s> in MOVES: s in perSegs[a,b]} do;
      unfix Move[l,s];
      unfix lbMove[l,s];
   end;

   solve obj FeasLB with milp / primalin relobjgap=0.01;

   for{<(l),s> in MOVES: s in perSegs[a,b]} do;
      fix Move[l,s];
      fix lbMove[l,s];
   end;
end;
```

Copyright ©2011, SAS Institute Inc. All rights reserved.

Agenda

1 Introduction

2 Modeling and Data, Data, Data

3 The Magic of Mixed Integer Programming

4 Homebrew Solutions

5 Conclusions

Copyright ©2011, SAS Institute Inc. All rights reserved.

Conclusions

- Finding solutions to optimization problem using mixed integer programming requires
  - » Understanding the problem and data
  - » A good model
  - » "Luck" or a good problem specific heuristic
- Good problem specific heuristics do not have to be very complex, they just have to work
- SAS can provide tools, industry specific software and consulting to solve your real world problems

Copyright ©2011, SAS Institute Inc. All rights reserved.

Thank you for your attention.

Philipp.Christophel@sas.com