Outline

1. Models

2. An Overview of Software for MIP

3. ZIBOpt

Modeling

- Min cost flow
- Shortest path
- Max flow
- Assignment and Bipartite Matching
- Transportation
- Multicommodities
Models
An Overview of Software for MIP
ZIBOpt

Modeling

Set Covering

\[
\begin{align*}
\text{min } & \sum_{j=1}^{n} c_j x_j \\
\text{s.t. } & \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \forall i \\
x_j & \in \{0, 1\}
\end{align*}
\]

Set Partitioning

\[
\begin{align*}
\text{min } & \sum_{j=1}^{n} c_j x_j \\
\text{s.t. } & \sum_{j=1}^{n} a_{ij} x_j = 1 \quad \forall i \\
x_j & \in \{0, 1\}
\end{align*}
\]

Set Packing

\[
\begin{align*}
\text{max } & \sum_{j=1}^{n} c_j x_j \\
\text{s.t. } & \sum_{j=1}^{n} a_{ij} x_j \leq 1 \quad \forall i \\
x_j & \in \{0, 1\}
\end{align*}
\]

Traveling Salesman Problem

Figure 3.1 Locations of the 42 cities.

Figure 3.2 Solution of the initial LP relaxation.

Figure 3.3 LP solution after three subtour constraints.
Figure 3.4 LP solution satisfying all subtour constraints.

Figure 3.7 What is wrong with this vector?

Figure 3.8 A violated comb.

Figure 3.9 An optimal tour through 42 cities.
minimize $c^T x$ subject to

$0 \leq x_e \leq 1$ for all edges $e$,

$\sum(x_e : v \text{ is an end of } e) = 2$ for all cities $v$,

$\sum(x_e : e \text{ has one end in } S \text{ and one end not in } S) \geq 2$

for all nonempty proper subsets $S$ of cities,

$\sum_{i=0}^{3}(\sum(x_e : e \text{ has one end in } S_i \text{ and one end not in } S_i) \geq 10$,

for any comb

24,978 Cities

solved by LK-heuristic and proved optimal by branch and cut

10 months of computation on a cluster of 96 dual processor Intel Xeon 2.8 GHz workstations

http://www.tsp.gatech.edu

MIP for Scheduling

24,978 Cities

Formulation for $Qm|pj = 1|\sum h_j(C_j)$ and relation with transportation problems

Formulation of $1|prec|\sum w_j C_j$ and $Rm||\sum C_j$ as weighted bipartite matching and assignment problems.

Formulation of $1|prec|\sum w_j C_j$ and how to deal with disjunctive constraints

http://www.tsp.gatech.edu
Outline

1. Models

2. An Overview of Software for MIP

3. ZIBOpt

How to solve MIP programs

- Use a mathematical workbench like MATLAB, MATHEMATICA, MAPLE, R.

- Use a modeling language to convert the theoretical model to a computer usable representation and employ an out-of-the-box general solver to find solutions.

- Use a framework that already has many general algorithms available and only implement problem specific parts, e.g., separators or upper bounding.

- Develop everything yourself, maybe making use of libraries that provide high-performance implementations of specific algorithms.

Thorsten Koch
“Rapid Mathematical Programming”
Technische Universität, Berlin, Dissertation, 2004

Advantages: easy if familiar with the workbench

Disadvantages: restricted, not state-of-the-art

Advantages: flexible on modeling side, easy to use, immediate results, easy to test different models, possible to switch between different state-of-the-art solvers

Disadvantages: algorithmical restrictions in the solution process, no upper bounding possible
How to solve MIP programs

- Use a framework that already has many general algorithms available and only implement problem specific parts, e.g., separators or upper bounding.

**Advantages:** allow to implement sophisticated solvers, high performance bricks are available, flexible

**Disadvantages:** view imposed by designers, vendor specific hence no transferability,

---

Develop everything yourself, maybe making use of libraries that provide high-performance implementations of specific algorithms.

**Advantages:** specific implementations and max flexibility

**Disadvantages:** for extremely large problems, bounding procedures are more crucial than branching

---

Modeling Languages

Thorsten Koch
“Rapid Mathematical Programming”
Technische Universität, Berlin, Dissertation, 2004

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>Solver</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMMS</td>
<td><a href="http://www.aimms.com">www.aimms.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>AMPL</td>
<td><a href="http://www.ampl.com">www.ampl.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>GAMS</td>
<td><a href="http://www.gams.com">www.gams.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>LINGO</td>
<td><a href="http://www.lindo.com">www.lindo.com</a></td>
<td>fixed</td>
<td>commercial</td>
</tr>
<tr>
<td>LPL</td>
<td><a href="http://www.virtual-optima.com">www.virtual-optima.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>MISTOPT</td>
<td><a href="http://www.potomac.edu/MISTOPT">http://www.potomac.edu/MISTOPT</a></td>
<td>open</td>
<td>mixed</td>
</tr>
<tr>
<td>Molpro</td>
<td><a href="http://www.dashoptimization.com">www.dashoptimization.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>MPEC</td>
<td><a href="http://www.matheonsoftware.com">www.matheonsoftware.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>ONL</td>
<td><a href="http://www.onl.com">www.onl.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>OEPL</td>
<td><a href="http://www.onl.com">www.onl.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>GAMS-MP</td>
<td><a href="http://www.gams.com">www.gams.com</a></td>
<td>open</td>
<td>commercial</td>
</tr>
<tr>
<td>ZIMPL</td>
<td><a href="http://www.zib.de/Optimization/Software/Soplex">www.zib.de/Optimization/Software/Soplex</a></td>
<td>open</td>
<td>fixed</td>
</tr>
</tbody>
</table>

“Software Survey: Linear Programming” by Robert Fourer
http://www.lionhrtpub.com/orms/orms-6-05/frsurvey.html

LP-Solvers

Choose from:

- CPLEX http://www.ilog.com/products/cplex
- XPRESS-MP http://www.dashoptimization.com
- Soplex http://www.zib.de/Optimization/Software/Soplex
- COIN CLP http://www.coin-or.org
- GLPK http://www.gnu.org/software/glpk
- LP-SOLVE http://lpsolve.sourceforge.net/

http://www.lionhrtpub.com/orms/orms-6-05/frsurvey.html
ZIBOpt

- **Zimpl** is a little algebraic **Modeling language** to translate the mathematical model of a problem into a linear or (mixed-) integer mathematical program expressed in .lp or .mps file format which can be read and (hopefully) solved by a LP or MIP solver.

- **Scip** is an **IP-Solver**. It solves Integer Programs and Constraint Programs: the problem is successively divided into smaller subproblems (branching) that are solved recursively. Integer Programming uses LP relaxations and cutting planes to provide strong dual bounds, while Constraint Programming can handle arbitrary (non-linear) constraints and uses propagation to tighten domains of variables.

- **SoPlex** is an **LP-Solver**. It implements the revised simplex algorithm. It features primal and dual solving routines for linear programs and is implemented as a C++ class library that can be used with other programs (like SCIP). It can solve standalone linear programs given in MPS or LP-Format.

**Modeling Cycle**