

DMP204
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 5
**Mixed Integer Programming
Models and Exercises**

Marco Chiarandini

Outline

1. Models
2. An Overview of Software for MIP
3. ZIBOpt

2

Outline

Models
An Overview of Software for MIP
ZIBOpt

1. Models
2. An Overview of Software for MIP
3. ZIBOpt

Modeling

Models
An Overview of Software for MIP
ZIBOpt

- Min cost flow
- Shortest path
- Max flow
- Assignment and Bipartite Matching
- Transportation
- Multicommodities

3

4

Set Covering

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \geq 1 \quad \forall i \\ & x_j \in \{0, 1\} \end{aligned}$$

Set Partitioning

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j = 1 \quad \forall i \\ & x_j \in \{0, 1\} \end{aligned}$$

Set Packing

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \leq 1 \quad \forall i \\ & x_j \in \{0, 1\} \end{aligned}$$

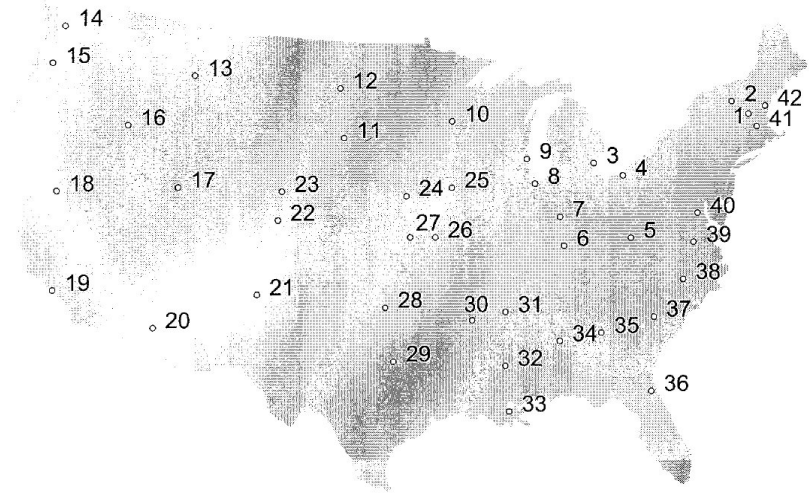


Figure 3.1 Locations of the 42 cities.

5

6

Traveling Salesman Problem

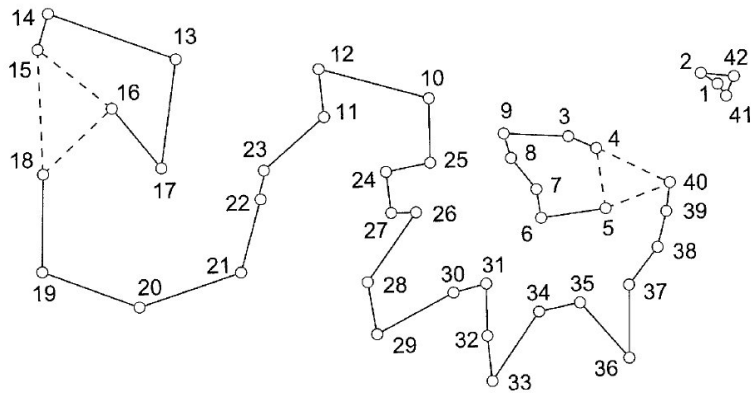


Figure 3.2 Solution of the initial LP relaxation.

Traveling Salesman Problem

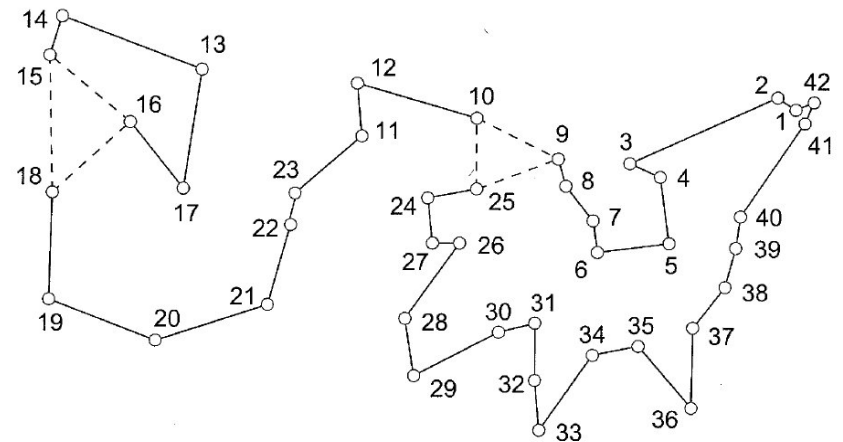


Figure 3.3 LP solution after three subtour constraints.

6

6

Traveling Salesman Problem

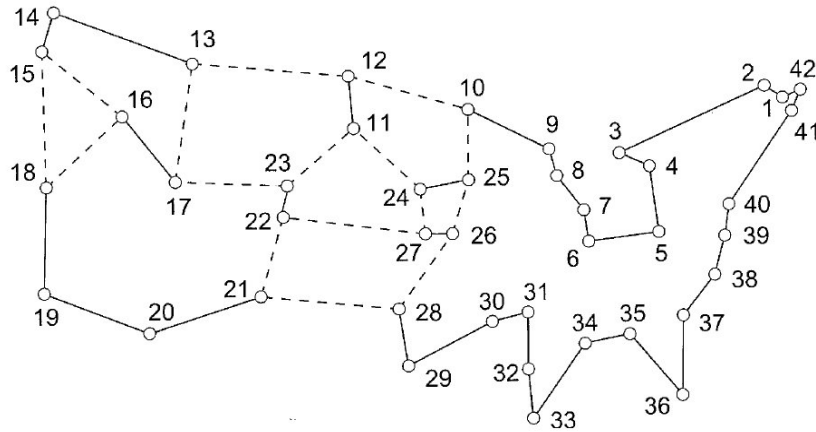


Figure 3.4 LP solution satisfying all subtour constraints.

Traveling Salesman Problem

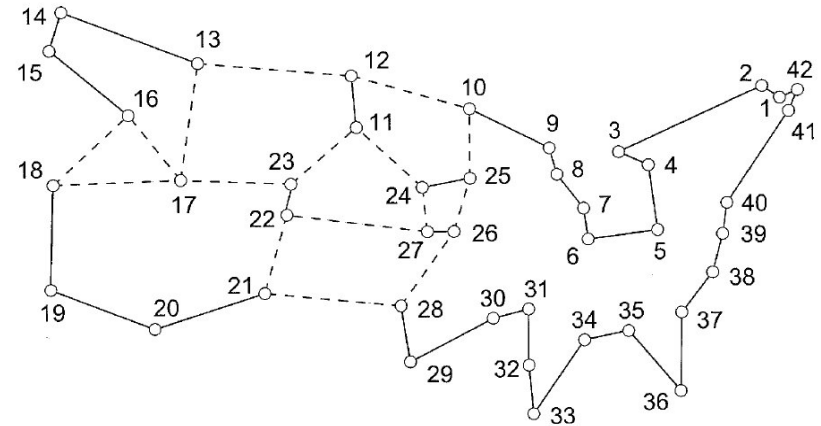


Figure 3.7 What is wrong with this vector?

6

6

Traveling Salesman Problem

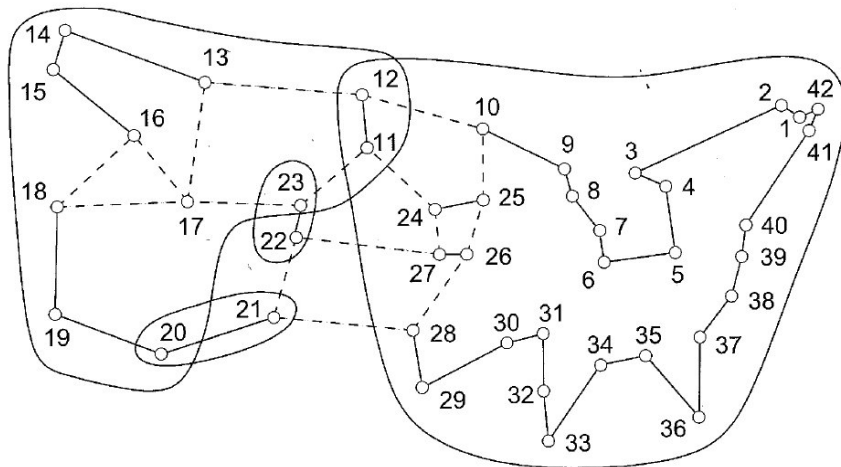


Figure 3.8 A violated comb.

Traveling Salesman Problem

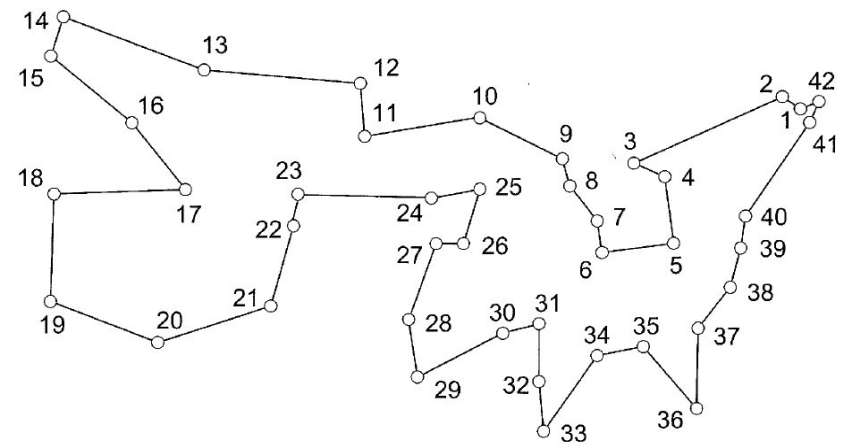


Figure 3.9 An optimal tour through 42 cities.

6

6

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,

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



24,978 Cities

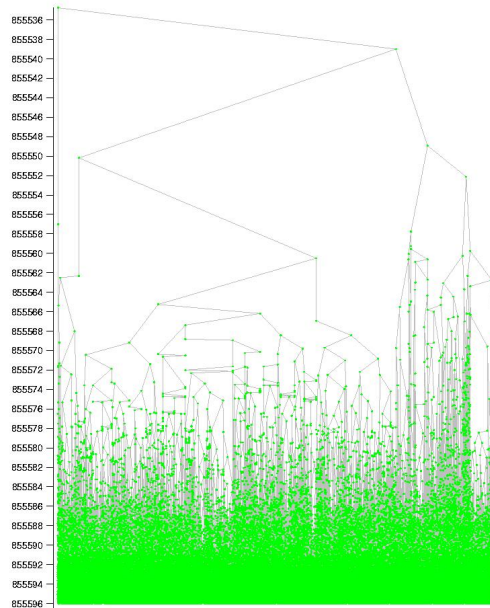
solved by LK-heuristic
and proved optimal
by branch and cut

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

<http://www.tsp.gatech.edu>

7

sw24978 Branching Tree - Run 5



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

- Formulation for $Qm|p_j = 1| \sum h_j(C_j)$ 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

1. Models
2. An Overview of Software for MIP
3. ZIBOpt

- 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

10

11

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

Advantages: easy if familiar with the workbench

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

- 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.

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

12

13

- 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

Name		URL	Solver	State
AIMMS	Advanced Integrated Multi-dimensional Modeling Software	www.aimms.com	open	commercial
AMPL	A Modeling Language for Mathematical Programming	www.ampl.com	open	commercial
GAMS	General Algebraic Modeling System	www.gams.com	open	commercial
LINGO	Lingo	www.lindo.com	fixed	commercial
LPL	(Linear Logical Literate) Programming Language	www.virtual-optima.com	open	commercial
MINOPT	Mixed Integer Non-linear Optimizer	titan.princeton.edu/MINOPT	open	mixed
MOSEL	Mosel	www.dashoptimization.com	fixed	commercial
MPL	Mathematical Programming Language	www.maximalsoftware.com	open	commercial
OMNI	Omni	www.haverly.com	open	commercial
OPL	Optimization Programming Language	www.ilog.com	fixed	commercial
GNU-MP	GNU Mathematical Programming Language	www.gnu.org/software/glpk	fixed	free
ZIMPL	Zuse Institute Mathematical Programming Language	www.zib.de/koch/zimpl	open	free

Thorsten Koch
"Rapid Mathematical Programming"
Technische Universität, Berlin, Dissertation, 2004

LP-Solvers

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/

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

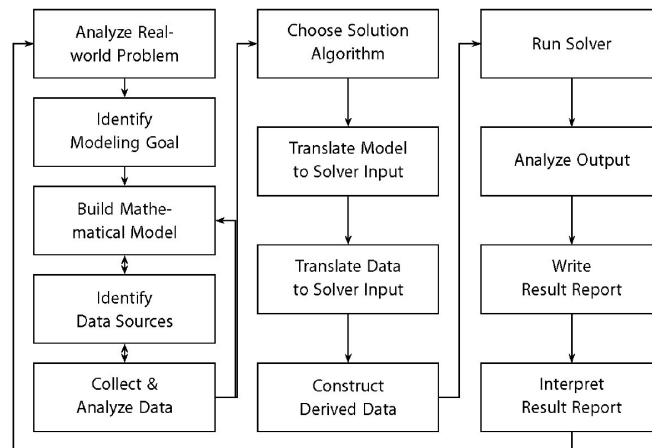
1. Models
2. An Overview of Software for MIP
3. ZIBOpt

- **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.

18

19

Modeling Cycle



*H. Schichl. "Models and the history of modeling".
In Kallrath, ed., Modeling Languages in Mathematical Optimization, Kluwer,
2004.*

20