A Short Introduction to Integer Linear Programming

April 15, 2008

Birgit Grohe

Integer Linear Programming (ILP)

- A Small Example (see separate set of slides)
- Modelling with integer variables
- Algorithms for ILP
  - Branch & Bound
  - Cutting Plane
  - (Others: decomposition methods, heuristics etc.)
- Theoretical aspects on LP and ILP

Modelling with Integer Variables

Variables

In LP we use continuous variables: $x_{ij} \geq 0$.

In ILP we can also use *integer*, *binary* and *discrete* variables.

If both continuous and integer variables are used in a program, it is called a *Mixed Integer Programming (MIP)* problem.

Constraints

In an ILP (or MIP) it is possible to model linear constraints, but also e.g. if-then- and either-or-relations. This is done by introducing additional binary variables and additional constraints.

The Set Covering Problem

Given a number of items and a cost for each item. Given a number of subsets of items. Find a selection of items such that each subset contains at least one selected item and such that the total cost of the selected items is minimized.

Mathematical formulation

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax \geq \mathbf{1} \\ & x \text{ binary} \end{aligned}$$

where $c, \mathbf{1}, x$ are vectors, and A is matrix with $a_{ij} \in \{0, 1\}$.

Related problems: *Set Partitioning* and *Set Packing*.

Example: Installing Security Telephones

(Taha, example 9.1-2, modified) A company wants to install emergency telephones such that each street has access to at least one phone. It is logical to place the phones on crossings of streets. Each crossing has an installation cost. Find the cheapest selection of crossings to provide all streets with phones.

Taha, figure 9.1

Installing Security Telephones: Mathematical Model

Introduce binary variables for each crossing: $x_j = 1$ if a phone is build, $x_j = 0$ otherwise.

For each street, introduce a constraint saying that a phone should be placed at least one of its crossings. E.g. for street G: $x_1 + x_6 \geq 1$ (for full set of constraints, see Taha, example 9.1-2).

Objective function:
$min\, 2x_1 + 2x_2 + 3x_3 + 4x_4 + 3x_5 + 2x_6 + 2x_7 + x_8$

ILP-Solution: $x_1, x_2, x_5, x_7 = 1$ all other variables are 0. Objective value: 9.

LP-Solution: $x_j = 0.5$ for all variables. Objective value: 8.5.

More Modelling Examples

(Taha, example 9.1-3) Given three telephone companies A, B and C which charge a fixed start-up price of 16, 25 and 18, respectively. For each minute of call-time the A, B, C charge 0.25, 0.21 and 0.22.

If we want to phone 200 minutes, which company should we choose?

Variables for the number of minutes called by A, B and C: $x_i \geq 0$

Binary variables $y_i = 1$ if $x_i > 0$, $y_i = 0$ otherwise. (Pay start-up price only if calls are made with company i.)

ILP model see Taha page 375

More Modelling Examples (2)

(Taha, example 9.1-4) Suppose that we wish to process three jobs on one machine. Each job has a processing time p_j, a due date d_j and a penalty cost c_j if the due date is missed. How should the jobs be scheduled to minimize the total penalty cost?

Taha, table on page 360

Algorithms for Solving the ILP Problem

- Branch & Bound
- Cutting Plane
- Decomposition methods
- Heuristics
- ...

Small example on other handout

Branch & Bound (B&B)

Idea: Solve the LP-relaxation. If the solution is integer, then an optimal solution is found. Otherwise the objective function value is a *upper bound (UB)* on the ILP problem (maximization problem).

Choose a fractional variable and create two new LPs by *branching* on this variable.

Continue branching until either an integer solution is found. Maintain the best integer solution as a *lower bound (LB)*.

Or until UB<LB, then the current branch can be *pruned* (the optimal solution cannot be in this branch).

For an ILP with n binary variables, the complete B&B tree contains $O(2^n)$ nodes, i.e. up to $O(2^n)$ LPs have to be solved!

B&B: An Example

(Taha, example 9.2-1) Given the following ILP:

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 5 \\ & 10x_1 + 6x_2 \leq 45 \end{aligned}$$

LP-optimum is $z = 23.75$, $x_1 = 3.75$ and $x_2 = 1.25$.

Taha, figure 9.6, 9.7, 9.8 and 9.9

B&B: Design Decisions

Design of a B&B algorithm:

- Variable ordering (choice of the branching variable)
- Value ordering (which branch to examine first)
- Good upper and lower bounds (heuristics, use problem structure, etc)

Cutting Plane Algorithms

Idea: Solve the LP-relaxation. If the solution is integer, then an optimal solution is found.

Otherwise find a *cut*, i.e. a constraint that cuts off the fractional solution, *but none of the integer solutions*.

The cut is also required to pass through at least one integer point.

Add cuts to the current LP and resolve until an integer solution is found.

Remark: An inequality in higher dimensions defines a *hyper-plane*, therefore the name cutting *plane*.

Cutting Plane: A Very Small Example

Given the following ILP:

$$\max\{x_1 + x_2 : 2x_1 + 4x_2 \leq 7, x_i \text{ binary}\}$$

ILP solution: $z = 3$, $x = (3, 0)$.

Solution of the LP-relaxation: $z = 3.5$, $x = (3.5, 0)$.

Generating a simple cut: Divide the constraint by 2:

$$x_1 + 2x_2 \leq 3.5 \quad \rightarrow \quad x_1 + 2x_2 \leq 3$$

Adding this cut to the LP-relation gives the optimal ILP solution.

Cutting Plane: Another Example

(Taha, example 9.2-2) Given the following ILP:

$$\begin{aligned} \max \quad & 7x_1 + 10x_2 \\ \text{s.t.} \quad & -x_1 + 3x_2 \leq 6 \\ & 7x_1 + x_2 \leq 35 \end{aligned}$$

LP-optimum is $z = 66.5$, $x_1 = 4.5$ and $x_2 = 3.5$.

Taha, figure 9.10

Cutting Plane Algorithms

Problem: It may be necessary to generate MANY cuts.

General methods, e.g. Chvatal-Gomory cuts. Problem specific cuts more efficient, e.g. comb-inequalities for TSP.

Pure Cutting Plane algorithms are usually not as efficient as B&B. In commercial solvers (e.g. CPLEX), cuts are used sometimes to help the B&B algorithm. If the problem has a specific structure, e.g. TSP, Set Cover etc. then problem specific cuts are used.

The Travelling Salesperson Problem (TSP)

Given n cities and connections between all cities (distances on each connection). Find shortest round tour.

Taha, example 9.3-2 and 9.3-3

An ILP Formulation of the TSP Problem

Let the distances between the cities be d_{ij}. Introduce binary variables x_{ij} for each connection.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} & \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1 \; \forall i & (1)\\
& \sum_{i=1}^{n} x_{ij} = 1 \; \forall j & (2)\\
& \sum_{i\in U, j\in A\setminus U} \geq 1 \quad 2 \leq |U| \leq |V|-2 & (3)\\
& x_{ij} \text{ binary} &
\end{aligned}
$$

We have to enter and to leave each city exactly once, constraints (1) and (2). Constraints (3) are called *subtour elimination constraints*.

An ILP Formulation of the TSP Problem

Alternative formulation of (3):

$$\sum_{(ij)\in U} x_{ij} \leq |U|-1 \quad 2 \leq |U| \leq |V|-2$$

Solution methods for the TSP Problem

- B&B
- Cutting plane algorithms
- Heuristics
  - Nearest neighbor algorithm
  - 2-opt, 3-opt, etc.
  - Christofides heuristic
  - ...
- ...

General problems for all solution methods for the TSP: *Combinatorial explosion*, i.e. very many possible tours $O = (n!)$, very many subtour elimination constraints.

Literature

Taha: Operations Research, An Introduction (course book)

Kleinberg, Tardos: Algorithm Design

Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms

Nemhauser, Wolsey: Integer and Combinatorial Optimization

Cook, Cunningham, Pullyblank, Schrijver: Combinatorial Optimization

Garey, Johnson: Computers and Intractability

Remarks on TSP

There exist different versions of the TSP: Euclidean, metric, symmetric, etc.

The TSP is an extremely well-studied problem in Combinatorial Optimization, has been a competition problem in the past. Homepages only dealing with TSP.

Despite of the TSP being a very difficult problem, it is today possible to solve comparably large instances.