MVE165/MMG631
Linear and integer optimization with applications
Lecture 7
Discrete optimization: theory and algorithms

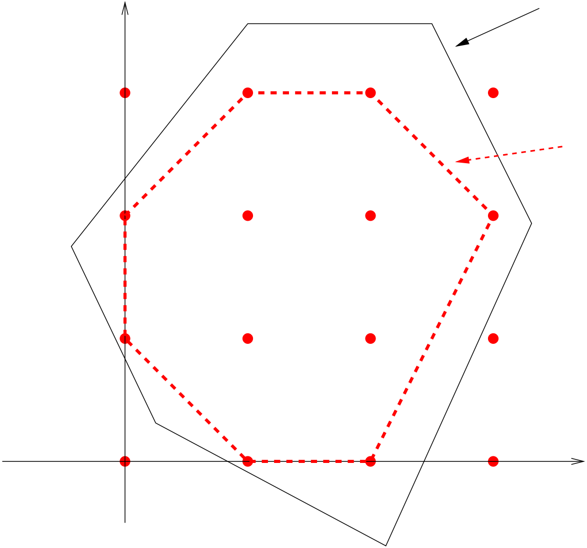
Ann-Brith Strömberg

2014–04–08

Overview

- Relaxations: cutting planes and Lagrangean duals
- TSP and routing problems
- Branch–and–bound for structured problems

Good and ideal formulations (Ch. 14.3)



$\mathbf{Ax} \leq \mathbf{b}$

Ideal since all extreme points are integral

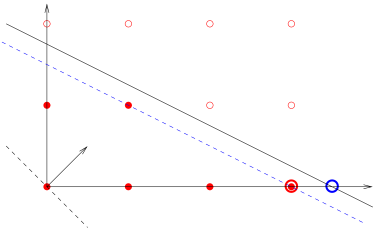
The linear program has integer extreme points

Cutting planes: A very small example

- Consider the following ILP:

$$\min\{-x_1 - x_2 : 2x_1 + 4x_2 \leq 7, x_1, x_2 \geq 0 \text{ and integer}\}$$

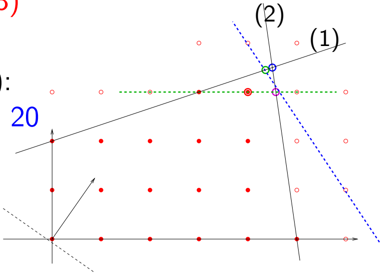
- ILP optimal solution: $z = -3$, $\mathbf{x} = (3, 0)$
- LP (continuous relaxation) optimum: $z = -3.5$, $\mathbf{x} = (3.5, 0)$
- Generate a simple cut:
  "Divide the constraint" by 2
  and round the RHS down
  $x_1 + 2x_2 \leq 3.5 \Rightarrow x_1 + 2x_2 \leq 3$
- Adding this cut to the continuous relaxation yields the optimal ILP solution



- Consider the ILP

$$
\begin{array}{llll}
\max & 7x_1 + 10x_2 & & \\
\text{subject to} & -x_1 + 3x_2 & \leq 6 & (1) \\
 & 7x_1 + x_2 & \leq 35 & (2) \\
 & x_1, x_2 & \geq 0, \text{ integer} &
\end{array}
$$

- LP optimum: $z = 66.5$, $\mathbf{x} = (4.5, 3.5)$
- ILP optimum: $z = 58$, $\mathbf{x} = (4, 3)$
- Generate a VI by "adding" the two constraints (1) and (2): $6x_1 + 4x_2 \leq 41 \Rightarrow 3x_1 + 2x_2 \leq 20$ $\Rightarrow \mathbf{x} = (4.36, 3.45)$
- Generate a VI by "$7 \cdot (1) + (2)$": $22x_2 \leq 77 \Rightarrow x_2 \leq 3$ $\Rightarrow \mathbf{x} = (4.57, 3)$



Cutting plane algorithms (iterativley better approximations of the convex hull) (Ch. 14.5)

- Choose a suitable mathematical formulation of the problem

1. Solve the linear programming (LP) relaxation
2. If the solution is integer, Stop. An optimal solution is found
3. Add one or several *valid inequalities* that cut off the fractional solution *but none of the integer solutions*
4. Resolve the new problem and go to step 2.

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

About cutting plane algorithms

- Problem: It may be necessary to generate VERY MANY cuts
- Each cut should also pass through at least one integer point $\Rightarrow$ faster convergence
- Methods for generating valid inequalities
  - Chvatal-Gomory cuts (combine constraints, make beneficial roundings of LHS and RHS)
  - Gomory's method: generate cuts from an optimal simplex basis (Ch. 14.5.1)
- Pure cutting plane algorithms are usually less efficient than branch–&–bound
- In commercial solvers (e.g. CPLEX), cuts are used to help (presolve) the branch–&–bound algorithm
- For problems with specific structures (e.g. TSP and set covering) problem specific classes of cuts are used

Lagrangian relaxation ($\Rightarrow$ optimistic estimates of z^*) (Ch. 17.1–17.2)

- Consider a minimization integer linear program (ILP):

$$
\text{[ILP]} \qquad z^* = \begin{array}{rrcll}
\min & \mathbf{c}^{\mathrm{T}}\mathbf{x} & & & \\
\text{subject to} & \mathbf{A}\mathbf{x} & \leq & \mathbf{b} & \qquad (1) \\
 & \mathbf{D}\mathbf{x} & \leq & \mathbf{d} & \qquad (2) \\
 & \mathbf{x} & \geq & \mathbf{0} \text{ and integer} &
\end{array}
$$

- Assume that the constraints (1) are complicating (subtour eliminating constraints for TSP, e.g.)
- Define the set $X = \{\mathbf{x} \in Z_+^n \,|\, \mathbf{D}\mathbf{x} \leq \mathbf{d}\}$
- Remove the constraints (1) and add them—with penalty parameters $\mathbf{v}$—to the objective function

$$
h(\mathbf{v}) = \min_{\mathbf{x} \in X} \left\{ \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{v}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \right\} \qquad (3)
$$

Weak duality of Lagrangian relaxations

Theorem: For any $\mathbf{v} \geq \mathbf{0}$ it holds that $h(\mathbf{v}) \leq z^*$.

Proof: Let $\bar{\mathbf{x}}$ be feasible in [ILP] $\Rightarrow \bar{\mathbf{x}} \in X$ and $\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}$. It then holds that

$$h(\mathbf{v}) = \min_{\mathbf{x} \in X} \left\{ \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{v}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \right\} \leq \mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}} + \mathbf{v}^{\mathrm{T}}(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}) \leq \mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}}.$$

Since an optimal solution $\mathbf{x}^*$ to [ILP] is also feasible, it holds that

$$h(\mathbf{v}) \leq \mathbf{c}^{\mathrm{T}}\mathbf{x}^* = z^*.$$

□

$\Rightarrow$ $h(\mathbf{v})$ is a *lower bound* on the optimal value z^* for any $\mathbf{v} \geq \mathbf{0}$

- The best lower bound is given by

$$h^* = \max_{\mathbf{v} \geq \mathbf{0}} h(\mathbf{v}) = \max_{\mathbf{v} \geq \mathbf{0}} \left\{ \min_{\mathbf{x} \in X} \left\{ \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{v}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \right\} \right\}$$

Tractable Lagrangian relaxations

- Special algorithms for minimizing the Lagrangian dual function h exist (e.g., subgradient optimization, Ch. 17.3)
- h is always concave but typically nondifferentiable
- For each value of $\mathbf{v}$ chosen, a *subproblem* (3) must be solved
- For general ILP's: typically a non-zero duality gap $h^* < z^*$
- The Lagrangian relaxation bound is never worse that the linear programming relaxation bound, i.e. $z^{\mathrm{LP}} \leq h^* \leq z^*$
- If the set X has the integrality property (i.e., X^{LP} has integral extreme points) then $h^* = z^{\mathrm{LP}}$
- Choose the constraints ($\mathbf{Ax} \leq \mathbf{b}$) to dualize such that the relaxed problem (3) is computationally tractable but still does not possess the integrality property

An ILP Example

[HOMEWORK]

Find optimistic and pessimistic bounds for the following ILP example using the branch–&–bound algorithm, a cutting plane algorithm, and Lagrangean relaxation.

$$\begin{array}{lrcl} \max & 5x_1 + 4x_2 & & \\ \text{s.t.} & x_1 + x_2 & \leq & 5 \\ & 10x_1 + 6x_2 & \leq & 45 \\ & x_1, x_2 & \geq & 0 \text{ and integer} \end{array}$$

The linear programming optimal solution is given by $z = 23.75$, $x_1 = 3.75$ and $x_2 = 1.25$

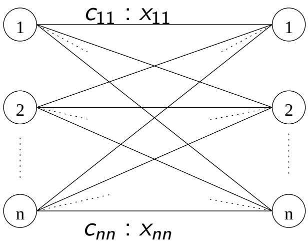
Assign each task to one resource, and each resource to one task

- Linear cost c_{ij} for assigning task i to resource j, $i, j \in \{1, \dots, n\}$
- Variables: $x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to resource } j \\ 0, & \text{otherwise} \end{cases}$

$$
\begin{array}{lrcll}
\text{min} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} & & & \\
\text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} & = & 1, & i = 1, \dots, n \\
& \displaystyle\sum_{i=1}^{n} x_{ij} & = & 1, & j = 1, \dots, n \\
& x_{ij} & \geq & 0, & i, j = 1, \dots, n
\end{array}
$$

The assignment model

- Choose *one* element from each row and each column

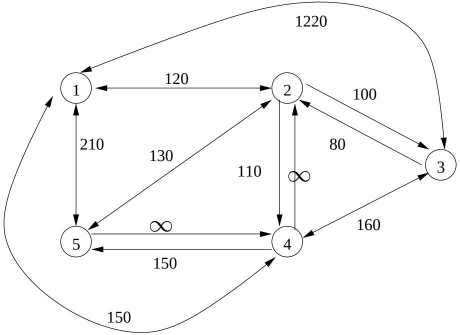


c_{11}	c_{12}	c_{13}						c_{1n}
c_{21}	c_{22}	c_{23}						c_{2n}
c_{31}	c_{32}	c_{33}						c_{3n}
c_{n1}	c_{n2}	c_{n3}						c_{nn}

- This integer linear model has integral extreme points, since it can be formulated as a network flow problem (Ch. 8) which has a *unimodular* constraint matrix (Def. 8.1)
- Can be efficiently solved using, e.g., the network simplex algorithm
- More efficient special purpose (primal–dual–graph-based) algorithms exist

The travelling salesperson problem (TSP, Ch. 13.10)

- Given n cities and connections between all cities (distances on each connection)
- Find shortest tour that passes through all the cities



- Complexity: NP-hard due to the combinatorial explosion

An ILP formulation of the TSP problem

- Let the distance from city i to city j be d_{ij}
- Introduce binary variables x_{ij} for each connection
- Let $V = \{1, \ldots, n\}$ denote the set of nodes (cities)

$$
\begin{aligned}
\min \quad & \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}, & & & \\
\text{s.t.} \quad & \sum_{j \in V} x_{ij} & = & \; 1, \quad i \in V, & (1) \\
& \sum_{i \in V} x_{ij} & = & \; 1, \quad j \in V, & (2) \\
& \sum_{i \in U, j \in V \setminus U} x_{ij} & \geq & \; 1, \quad \forall U \subset V : 2 \leq |U| \leq |V| - 2, & (3) \\
& x_{ij} & & \text{binary} \quad i, j \in V & (4)
\end{aligned}
$$

- Cf. the assignment problem
- Enter and leave each city exactly once $\Leftrightarrow$ (1) and (2)
- Constraints (3): *subtour elimination*

Solution methods for the TSP Problem

- Tailored branch–&–bound (Ch. 15)
- Heuristics
  - Constructive heuristics (Ch. 16.3)
  - Local search heuristics (Ch. 16.4)
  - Approximation algorithms (Ch. 16.6)
  - Metaheuristics (Ch. 16.5)
- ...
- Common difficulty for *all* solution methods for the TSP: *Combinatorial explosion*: # possible tours $\approx n!$

$\Rightarrow$ Very many subtour elimination constraints (3)

Branch–and–bound algorithm for TSP (Ch. 15.4.2)

- Relaxing just the binary constraints (4) in TSP does not yield a tractable problem, since the number of subtour elinimating constraints (3) is very large $\Rightarrow$ An LP with *very many* constraints
- Relaxing the subtour eliminating constraints (3) yields an assignment problem, which can be solved in polynomial time
- Solutions to a relaxed problem typically contains a number of sub-tours
- Branch on these sub-tours (rather than on fractional variables)
- Branching $\Leftrightarrow$ partitioning of the solution space
- DRAW AN EXAMPLE