MVE165/MMG630, Applied Optimization
Lecture 12
Combinatorial optimization models, theory and algorithms

Ann-Brith Strömberg

2011–04–15

Overview

- TSP and routing problems
- Branch–and–bound for structured problems
- Convexity; local and global optima
- Heuristics: constructive heuristics, local search methods, metaheuristics, approximation algorithms.

The assignment model (Ch. 13.5)

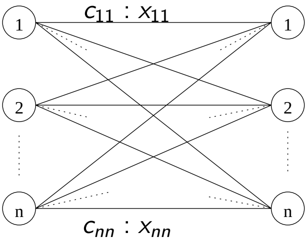
Assign each task to one resource, and each resource to one task

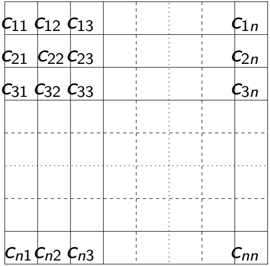
- Linear cost c_{ij} for assigning task i to resource j, $i, j \in \{1, \ldots, n\}$
- Variables: $x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to resource } j \\ 0, & \text{otherwise} \end{cases}$

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

The assignment model

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

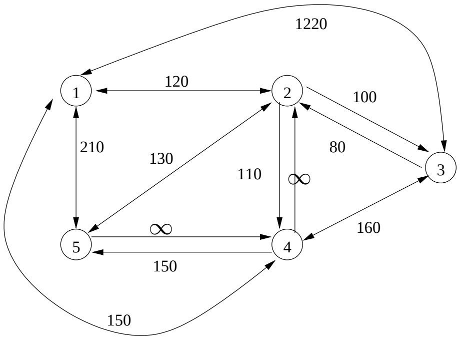




- This integer linear model has integral extreme points, since it can be formulated as a network flow problem
- Can be efficiently solved using, e.g., the network simplex algorithm
- More efficient special purpose (primal–dual–graph-based) algorithms exist

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)

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

- 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 (instead of fractional variables)
- Branching $\Leftrightarrow$ partitioning of the solution space
- Draw an example

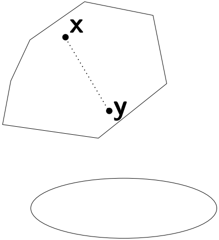
Recall: convex sets

- A set S is convex if, for any elements $\mathbf{x}, \mathbf{y} \in S$ it holds that

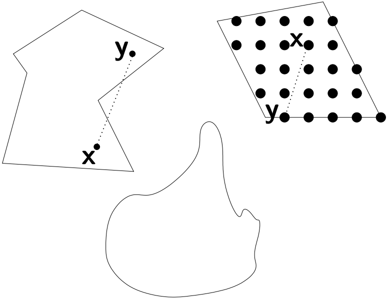
$$\alpha \mathbf{x} + (1-\alpha)\mathbf{y} \in S \text{ for all } 0 \leq \alpha \leq 1$$

- Examples:

Convex sets



Non-convex sets



$\Rightarrow$ Integrality requirements $\Rightarrow$ nonconvex feasible set

Local vs. global optima

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

- Global optimum:
  A solution $\mathbf{x}^* \in X$ such that $\mathbf{c}^{\mathrm{T}}\mathbf{x}^* \leq \mathbf{c}^{\mathrm{T}}\mathbf{x}$ for all $\mathbf{x} \in X$
- ε-neighbourhood of $\bar{\mathbf{x}}$: $N_\varepsilon(\bar{\mathbf{x}}) = \left\{\mathbf{x} \in X \,\middle|\, |\mathbf{x} - \bar{\mathbf{x}}| \leq \varepsilon\right\}$
- The distance measure $|\mathbf{x} - \bar{\mathbf{x}}|$ may be "freely" defined as, e.g., # arcs differing (Hamming distance), Euclidean, Manhattan, 2-interchange, ...
- Local optimum:
  A solution $\bar{\mathbf{x}} \in X$ such that $\mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}} \leq \mathbf{c}^{\mathrm{T}}\mathbf{x}$ for all $\mathbf{x} \in N_\varepsilon(\bar{\mathbf{x}})$

Heuristic algorithms

- Optimization problems with high complexity may be too time consuming to solve to optimality
- Heuristic algorithms can be utilized
- But: Only local optimality can then be guaranteed

Heuristics I: Constructive heuristics (Ch. 16.3)

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}} \mathbf{x}$$

- Start by an "empty set" and "add" elements according to some (simple) rule.
- Sometimes no guarantee that a feasible solution will be found
- No measure of how close to a global optimum a solution is
- Special rules for structured problems
- E.g. the greedy algorithm is a constructive heuristic (finds optimal solution to the minimum spanning tree)
- For TSP: nearest neighbour, cheapest insertion, farthest insertion, etc.
- EXAMPLE!

Heuristics II: Local search (Ch. 16.4)

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}} \mathbf{x}$$

- Start from a feasible solution, which is iteratively improved by limited modifications
- Finds a local optimum
- No measure of how close to a global optimum a solution is
- Specialized for structured problems, but also general (Ch. 16.2)
- For TSP: e.g. 2-interchange, 3-interchange,
- Example!

Local search heuristic algorithm (Ch. 16.4)

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

0. Initialization: Choose a feasible solution $\mathbf{x}^0 \in X$. Let $k = 0$.
1. Find all feasible points in an ε-neighbourhood $N_\varepsilon(\mathbf{x}^k)$ of $\mathbf{x}^k$
2. If $\mathbf{c}^{\mathrm{T}}\mathbf{x} \geq \mathbf{c}^{\mathrm{T}}\mathbf{x}^k$ for all $\mathbf{x} \in X \cap N_\varepsilon(\mathbf{x}^k) \Rightarrow$ Stop. $\mathbf{x}^k$ is a local optimum (w.r.t. ε)
3. Choose $\mathbf{x}^{k+1} \in X \cap N_\varepsilon(\mathbf{x}^k)$ such that $\mathbf{c}^{\mathrm{T}}\mathbf{x}^{k+1} < \mathbf{c}^{\mathrm{T}}\mathbf{x}^k$
4. Let $k := k + 1$ and go to step 1

Heuristics III: Approximation algorithms (Ch. 16.6)

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

- Performance guarantee: $(\bar{z} - z^*)/z^* \leq \alpha$ for some $0 < \alpha \leq 1$
- Specialized algorithms for structured problems
- EXAMPLE!

Spanning tree heuristic for TSP

- Consider a TSP on an undirected graph $G = (N, E, \mathbf{c})$
- Assume
  - G complete $\Leftrightarrow$ edges between all pairs of nodes
  - Δ-inequality: $c_{ij} \leq c_{ik} + c_{kj}$ for all $i, j, k \in N$

1. Find a minimum spanning tree $T \subset E$ on G
2. Create a multigraph G' using *two copies* of each edge in T
3. Find an Eulerian walk of G' and and embedded TSP-tour

- Guarantee: $(\bar{z} - z^*)/z^* \leq 1$
- Not worse than twice the optimal tour!
- EXAMPLE!

Proof

- Let $c(\text{TSP}) = z^*$ and $c(\text{tour}) = \bar{z}$
- A spanning tree is a relaxation of a TSP:
  All soubtour elimination constraints are fulfilled, but not the node valence (2 edges incident to each node)

$\Rightarrow$ $c(\text{MST}) \leq c(\text{TSP})$

- Two copies of each edge $\Rightarrow c(\text{tour}) \leq 2c(\text{MST}) \leq 2c(\text{TSP})$

$\Rightarrow$ $\frac{c(\text{tour}) - c(\text{TSP})}{c(\text{TSP})} \leq 1$

Heuristics IV: Metaheuristics (Ch. 16.5)

Consider a minimization problem:

$$\min_{\mathbf{x} \in X} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

- Intends to be more efficient than just plain local search methods
- Includes tabu search, simulated annealing
- EXAMPLE!

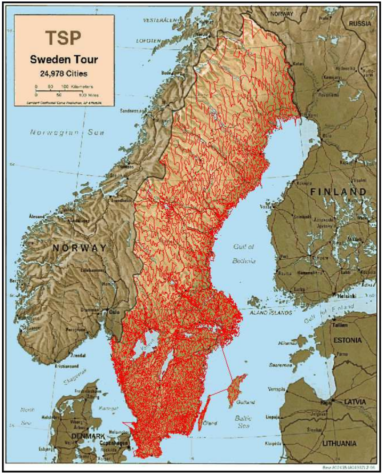
More about heuristics

- Start using a constructive heuristic $\Rightarrow$ feasible solution
- The choice of definition of a neighbourhood is model specific (e.g. Euclidean distance, number of arcs differing,)
- Apply a local search algorithm
- Finds a *local* optimal solution
- *No guarantee* to find global optimal solutions
- Extensions (e.g. tabu search): Temporarily allow worse solutions to "move away" from a local optimum (Ch. 16.5)
- Larger neighbourhoods yield better local optima, but takes more computation time to explore

The historical development of TSP solution

- Optimal solutions to TSP's of different sizes found

year	$n =$
1954	49
1962	33
1977	120
1987	532
1987	666
1987	2392
1994	7397
1998	13509
2001	15112
2004	24978
2005/06	85900



The worlds largest TSP solved "so far" (2004) ...

- A TSP of 24 978 cities and villages (red houses) in Sweden
- Optimal tour: $\approx$ 72 500 km (855597 TSP LIB units)
- The tour of length 855 597 was found in March 2003 (Lin-Kernighan's TSP heuristic)
- It was proven in May 2004 that no shorter tour exists
- A variety of heuristics, B&B, and cut generation algorithms
- The final stages that improved the lower bound from 855 595 up to 855 597 required $\approx$ 8 years of computation time (running in parallel on a network of Linux workstations)

  "Without knowledge of the 855 597 tour we would not have made the decision to carry out this final computation"
- `www.tsp.gatech.edu`
- New record in 2005/06: 85 900 cities in a VLSI application `www.tsp.gatech.edu/pla85900`