Transportation Logistics

Part IV: TSP history and milestones

Timeline - instances solved to optimality

Year	Research Team	Size
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 cities
1971	M. Held and R.M. Karp	64 rnd pts
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 citie
1977	M. Grötschel	120 cities
1980	H. Crowder and M.W. Padberg	318 cities
1987	M. Padberg and G. Rinaldi	532 cities
1987	M. Grötschel and O. Holland	666 cities
1987	M. Padberg and G. Rinaldi	2,392 cities
1994	D. Applegate, R. Bixby, V. Chvátal, W. Cook	7,397 cities
1998	D. Applegate, R. Bixby, V. Chvátal, W. Cook	13,509 cities
2001	D. Applegate, R. Bixby, V. Chvátal, W. Cook	15,112 cities
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, K. Helsgaun	24,978 cities

Source: `http://www.tsp.gatech.edu/index.html`

The largest TSP instance solved to optimality

The largest TSP instance solved to optimality

Size: 85,900 cities

The largest TSP instance solved to optimality

Size: 85,900 cities

Research Team: D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K. Helsgaun

The largest TSP instance solved to optimality

Size: 85,900 cities

Research Team: D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K. Helsgaun

Time needed: 136 CPU years (on a 2.4 GHz AMD Opteron 250)

The largest TSP instance solved to optimality

Size: 85,900 cities

Research Team: D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K. Helsgaun

Time needed: 136 CPU years (on a 2.4 GHz AMD Opteron 250)

Location: Georgia Tech (Atlanta, US)

The largest TSP instance solved to optimality

Size: 85,900 cities

Research Team: D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K. Helsgaun

Time needed: 136 CPU years (on a 2.4 GHz AMD Opteron 250)

Location: Georgia Tech (Atlanta, US)

Year: 2006

The largest TSP instance solved to optimality

Size: 85,900 cities

Research Team: D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K. Helsgaun

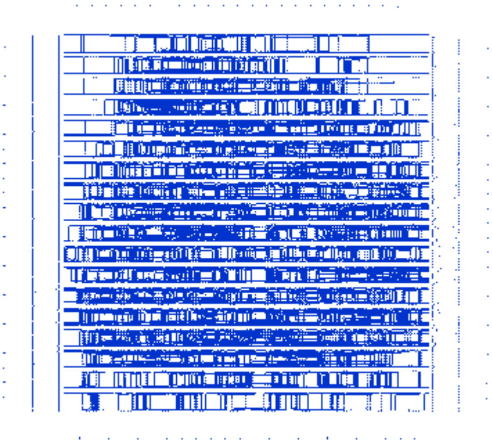
Time needed: 136 CPU years (on a 2.4 GHz AMD Opteron 250)

Location: Georgia Tech (Atlanta, US)

Year: 2006

Solver: CONCORDE

The largest TSP instance solved to optimality



Source: http://www.tsp.gatech.edu/index.html

Branch and cut

Branch and cut algorithms

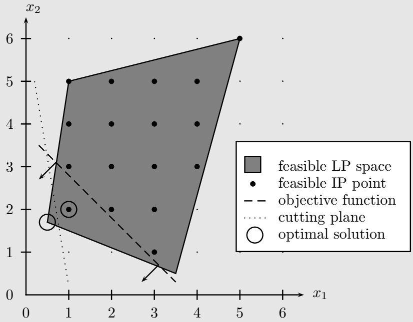
combine the branch and bound and the cutting plane approach

Branch and cut

Branch and cut algorithms

combine the branch and bound and the cutting plane approach

Cutting planes (cuts)



Branch and cut

The idea:

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \to \min \tag{1}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} = 1 \qquad \forall j \in V, \tag{2}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} = 1 \qquad \forall i \in V, \tag{3}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset V, |S| \geq 2, \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{5}$$

Branch and cut

The idea:

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \to \min \tag{1}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} = 1 \qquad \forall j \in V, \tag{2}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} = 1 \qquad \forall i \in V, \tag{3}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset V, |S| \geq 2, \tag{4}$$

$$(x_{ij} \in \{0,1\}) \qquad \forall (i,j) \in A. \tag{5}$$

relax binary requirements : $x_{ij} > 0$

Branch and cut

The idea:

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \rightarrow \min \tag{1}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} = 1 \qquad \forall j \in V, \tag{2}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} = 1 \qquad \forall i \in V, \tag{3}$$

$$(\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset V, |S| \geq 2,) \tag{4}$$

$$(x_{ij} \in \{0,1\}) \qquad \forall (i,j) \in A. \tag{5}$$

relax binary requirements : $x_{ij} > 0$
add subtour eliminations constraints (and other **valid inequalities**) in terms of cuts.

Branch and cut

At each node in the branch and bound tree

- **Repeat** until not additional cuts can be found.
    - solve the LP with all cuts generated so far
    - run a **separation algorithm** to identify violated cuts
    - add violated cuts to LP

Branch and cut

At each node in the branch and bound tree

- **Repeat** until not additional cuts can be found.
  - solve the LP with all cuts generated so far
  - run a **separation algorithm** to identify violated cuts
  - add violated cuts to LP

Separation algorithm

An exact method or a heuristic to find violated cuts in the solution of the current LP relaxation.

References

TSP code (Chapter 16 (by A. Lodi and A. Punnen) of the book: G. Gutin, A. Punnen (Eds) (2002) The Traveling Salesman Problems and its Variations. Kluwer Academic Publishers, 2002)
http://www.or.deis.unibo.it/research_pages/tspsoft.html

Largest TSP instances solved to optimality:
http://www.tsp.gatech.edu/index.html