TSP and Integer Programming

Daniel Espinoza

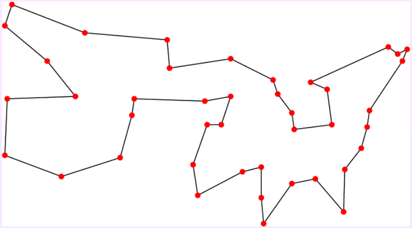
Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Industrial

August 6, 2007

Outline

- Most well known combinatorial optimization problem.
- Simply stated, yet $\mathcal{NP}$-complete.
- It has been (is) the birthplace of most techniques for MIP.
- One of the biggest success of IP.
- Vast research on this theme.
- Natural sub-problem of many practical problems.
- Based on the work of
  - David Applegate, Robert Bixby, Vašek Chvátal and William Cook.
  - “Finding Tours in the TSP”[ABCC95]
  - “Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems”[ABCC03]

UNIVERSIDAD DE CHILE FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS DEPARTAMENTO DE INGENIERIA INDUSTRIAL



Definition:

Given a finite set of *cities*, and travel costs between each pair of cities, find a tour that visits each city exactly once and goes back to the starting point.

More Precisely:

We will deal with problems where the costs are *symmetric*, i.e. the cost of travel from city x to city y is the same as to travel from city y to city x. Note also that the condition to visit *all cities* implies that the problem can be reduced to decide the order in which every city will be visited.

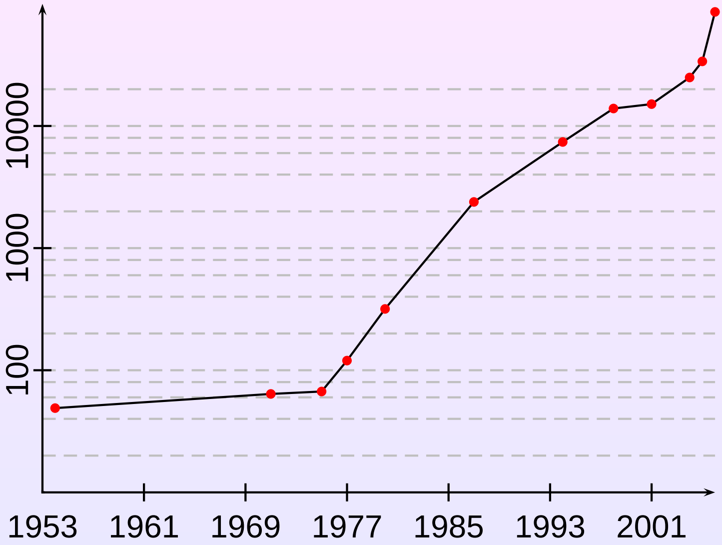
- First references date back from 1832. Practical guide for travelling salesman persons.
- Karl Menger, 1930, (Shortest Hamiltonian Path).
- J.B. Robinson, “On the Hamiltonian game (a traveling-salesman problem)”, 1949. First reference on its modern form.
- G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem”, 1954. Exact solution of a 49-city problem (state capitals of the USA), introduces cuts and B&B.
- M. Held and R.M. Karp, “A dynamic programming approach to sequencing problems”, 1962. introduction of heuristics based on dynamic programming.
- Lin and B.W. Kernighan, “An effective heuristic for the traveling-salesman problem”, 1973.

Record TSP over Time

Year	Authors	Cities
1954	Dantzig, Fulkerson, and Johnson	49
1971	Held and Karp	64
1975	Camerini, Fratta, and Maffioli	67
1977	Grötschel	120
1980	Crowder and Padberg	318
1987	Padberg and Rinaldi	532
1987	Grötschel and Holland	666
1987	Padberg and Rinaldi	2,392
1994	Applegate, Bixby, Chvátal, and Cook	7,397
1998	Applegate, Bixby, Chvátal, and Cook	13,509
2001	Applegate, Bixby, Chvátal, and Cook	15,112
2004	Applegate, Bixby, Chvátal, Cook, and Helsgaun	24,978
2005	Cook et. al.	33,810
2006	Cook et. al.	85,900

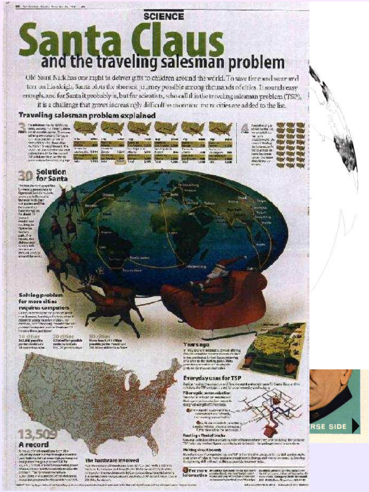
UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA INDUSTRIAL

Record TSP over Time



Some TSP Applications

- Vehicle Routing.
    - Scholar bus.
    - Emergency calls.
    - Express mail.
- Gene sequencing.
- Watching the skies (NASA).
- Chip production.
- World Tour.
- Santa's problem.


SCIENCE

Santa Claus and the traveling salesman problem

Old Saint Nick has one night to deliver gifts to children around the world. To save time and wear and tear on his sleigh, Santa plots the shortest journey possible among thousands of cities. It sounds easy enough, and for Santa it probably is, but for scientists, who call it the traveling salesman problem (TSP), it is a challenge that grows increasingly difficult as more and more cities are added to the list.

Traveling salesman problem explained

30 Solution for Santa

Solving problem for more cities requires computers

13,509 A record

Years ago

Everyday uses for TSP

The hardware involved

For more information

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

- Could we enumerate?
  - 10 cities: $\approx 10^{5.5}$ posibilities.
  - 100 cities: $\approx 10^{156}$ posibilities.
  - 1,000 cities: $\approx 10^{2,565}$ posibilities.
  - 33,810 cities: $\approx 10^{138,441}$ posibilities.
  - Universe age: $\approx 10^{18}$ seconds.
  - Atoms in the universe: $< 10^{100}$.
  - Limited capability.
- Held-Karp has guarantee $n^2 2^n$ in worst case.
- On euclidean instances, Christofides heuristic has guarantee $\frac{3}{2}$.
- On euclidean instances, Lin-Kernigan variantes achives in practice less than 1% GAP.
- Can we obtain better warranties?

Looking for good solutions

- Practical interest.
- Key ingridient in any branch and bound approach.
- A lot of research in the area.
- We will focus on looking for near-optimal solutions in a short time period.
- Compare other common heuristic approaches.

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA INDUSTRIAL

Heuristics and its limits

- If $\mathcal{P} \neq \mathcal{NP}$, there is no polynomial time heuristic for the TSP that guarantee $A(I)/OPT(I) \leq 2^n$ for all instances I [SG76].
- If $\mathcal{P} \neq \mathcal{NP}$, there exists $\varepsilon > 0$ such that no polynomial heuristic for the TSP has a guarantee $A(I)/OPT(I) \leq 1 + \varepsilon$ for all instances I with costs satisfying the triangular inequality [ALM$^+$92].
- There exists an algorithm A that, given an euclidian instance for the TSP, and a constant $\varepsilon > 0$, it runs on time $n^{\mathcal{O}(1/\varepsilon)}$ and gurantees $A(I)/OPT(I) < 1 + \varepsilon$ [Aro96].

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA INDUSTRIAL

Nearest Neighbour (NN)

- Initialization $k = 1$, randomly choose an initial city $i_k \in \{1, \ldots, n\}$.
- Loop while $k \neq n$, let $k = k + 1$, and choose $i_k \in \{1, \ldots, n\} \setminus \{i_1, \ldots, i_{k-1}\}$ minimizing $c(i_{k-1}, i_k)$.
- Finish return the tour $(i_1, \ldots, i_n)$.
- Notes:
  - Running time is $\mathcal{O}(n^2)$.
  - Worst-case guarantee $NN(I)/OPT(I) \leq \frac{1}{2}(\lfloor \log_2(n) \rfloor + 1)$.
  - Worst-known instances $NN(I)/OPT(I) \approx \Theta(\log(n))$.

Greedy heuristic (GR)

- Initialization Sort edges $e \in V \times V$ by cost in increasing order $e_1, \ldots, e_M$, and assign $m = k = 0$.
- Loop While $k \neq n$, let $k = k + 1$ y $m = m + 1$, while $\{e_1, \ldots, e_{k-1}\} \cup \{e_m\}$ can not be extended to a tour, let $m = m + 1$. Assign $e_k = e_m$.
- Finish return the tour described by $\{e_1, \ldots, e_n\}$.
- Notes:
  - Running time is $\mathcal{O}(n^2 \log(n))$.
  - Worst-case guarantee $NN(I)/OPT(I) \leq \frac{1}{2}(\lfloor \log_2(n) \rfloor + 1)$.
  - Worst-known instances $NN(I)/OPT(I) \approx \Theta(\log(n)/3\log(\log(n)))$.

UNIVERSIDAD DE CHILE FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS DEPARTAMENTO DE INGENIERIA INDUSTRIAL

Enumeration and Heuristics:

Christofides (CHR)

- Step 1 Build a minimum weight spanning tree T in $G = (V, E)$, where $V = \{1, \ldots, n\}$ and $E = V \times V$; note that $c(T) \leq c^*$.
- Step 2 Build a matching M ammong the odd degree nodes in T; note that $c(M) \leq \frac{1}{2} c^*$.
- Step 3 Note that $M \cup T$ is connected and eulerian, thus exist an ordering of the nodes that induces a tour with cost less than $c(M \cup T)$.
- Notes:
  - Ejecución es $\mathcal{O}(n^3)$.
  - Garantía $NN(I)/OPT(I) \leq \frac{3}{2}$.
  - Existen instancias tal que $NN(I)/OPT(I) \approx \Theta(\frac{3}{2})$.

Comparing Heuristics

Random euclidean instances (SEP GAP)

N	10^2	10^3	10^4	10^5	10^6
NN	25.6	26.0	24.3	23.6	23.3
GR	19.5	17.0	16.6	14.9	14.2
CHR	9.5	9.7	9.9	9.9	-

Random instances (SEP GAP)

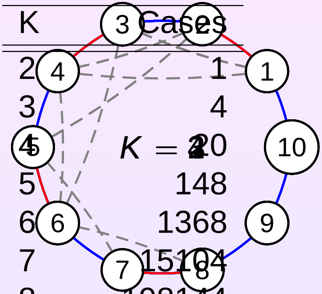
N	10^2	10^3	10^4	10^5	10^6
NN	130	240	360	-	-
GR	100	170	250	-	-

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA INDUSTRIAL

K-Opt heuristics

- Local improvement idea.
- Exchange 2 edges
- Exchange 3 edges
- how to reconnect?
- Exchange k edges.
- Lin-Kernighan uses 2-edge exchanges.
- Lin-Kernigham-Helsgun uses 5-edge exchanges.
- Don’t provide good bounds.

K	Cases
2	1
3	4
4	20
5	148
6	1368
7	15104
8	198144
9	2998656
10	51290496



K-opt limits:

- If $\mathcal{P} \neq \mathcal{NP}$, no heuristic that at every local-improvement iteration runs in polynomial time, satisfies $A(I)/OPT(i) \leq C$ for any constant C, even if we allow exponential-sized neighbourhood.
- Even if $\mathcal{P} = \mathcal{NP}$, no heuristic with polynomial size neighborhoods that do not depend on I can find the optimal solution of I.
- $2 - opt(I)/OPT(I) \geq \frac{1}{4}\sqrt{n}$ for instances where the triangular inequality holds for the cost matrix.

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

K-opt limits:

- $3-opt(I)/OPT(I) \geq \frac{1}{4}\sqrt[6]{n}$ for instances where the triangular inequality holds for the cost matrix.
- $k-opt(I)/OPT(I) \geq \frac{1}{4}\sqrt[2k]{n}$ for instances where the triangular inequality holds for the cost matrix.
- $k-opt(I)/OPT(I) \approx \mathcal{O}(\log(n))$ for euclidean instances.
- There are euclidean instances where $k-opt(I)/OPT(I) \approx \Theta(\log(n)/\log(\log(n)))$.

Comparing Heuristics

Random euclidean instances (SEP GAP)

N	10^2	10^3	10^4	10^5	10^6
GR	19.5	17.0	16.6	14.9	14.2
CHR	9.5	9.7	9.9	9.9	-
2-Opt	4.5	4.9	5.0	4.9	4.9
3-Opt	2.5	3.1	3.0	3.0	3.0

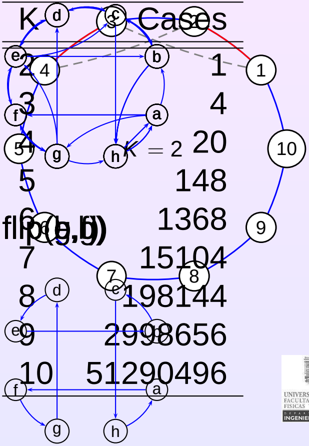
Random instances (SEP GAP)

N	10^2	10^3	10^4	10^5	10^6
GR	100	170	250	-	-
2-Opt	34	70	125	-	-
3-Opt	10	33	63	-	-

Can we do any better?

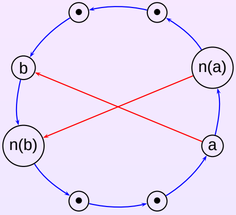
- Try larger k-opt values.
  - Takes really long.
  - Experiments suggest that gain is negligible.
  - How to program a 10-opt heuristic?
- Any k-opt move can be represented as a sequence of 2-opt moves.
- Explore promising partial moves.
- Basic idea behind Lin-Kernighan's heuristic.

K	Cases
2	1
3	4
4	20
5	148
6	1368
7	15104
8	198144
9	2998656
10	51290496



Lin-Kernighan Heuristic

- Basic Idea: improve one edge at a time.
  - Ask for $c(a, n(a)) - c(n(a), n(b)) > 0$
  - Such nodes called *promising*
- Basic Algorithm:
  1. $\Delta \leftarrow 0$
  2. while $\exists$ promising nodes.
  3. Choose b promising, $\Delta \leftarrow \Delta + c(a, n(a)) + c(b, n(b)) - c(a, b) - c(n(a), n(b))$.
  4. do $\mathit{flip}(n(a), b)$.
  5. If $\Delta > 0$ return current tour.



Lin-Kernighan Refinements

- How do we choose b?
  - Maximize $c(b,n(b)) - c(n(a),n(b))$.
  - Only consider k closest neighbors of $n(b)$.
- What if we do not succeed?
  - Allow backtracking.
  - More at lower levels.
  - Try also to replace $(p(a),a)$.
  - Sort promising nodes by $c(n(a),n(b))$.

- Basic Algorithm:
  1. $\Delta \leftarrow 0$
  2. while $\exists$ promising nodes.
  3. Choose b promising, $\Delta \leftarrow \Delta + c(a,n(a)) + c(b,n(b)) - c(a,b) - c(n(a),n(b))$.
  4. do $flip(n(a),b)$.
  5. If $\Delta > 0$ return current tour.

Lin-Kernighan Refinements

- How do we choose *a*?
  - Set all nodes as marked.
  - While there are marked nodes.
  - Call `lk_search(v,T)` for some marked node v.
    - if unsuccessful, unmark v.
    - if successful, mark all endpoints in the flip sequence.
- Can we do better?
  - While there is available time, generate a new initial tour T, call `lin_kernighan(T)`, keep best tour.
    - Called repeated Lin-Kernighan.
    - Best approach up to 1991.
  - Introduction of the *kick* concept.
    - Idea is to look harder close to *good* tours.
    - Called chained Lin-Kernighan.
    - Usual kick is the 4-bridge perturbation.

Basic Routines

- `flip(a,b)` - inverts the segment from a to b.
- `next(a)` - returns node after a in the tour.
- `prev(a)` - returns node before a in the tour.
- `sequence(a,b,c)` - returns 1 if b lies in the segment $a - c$ of the tour.

Instances

Name	Size	Target tour
pcb3038	3038	139070
usa13509	13509	20172983
pla85900	85900	143564780

Number of operations

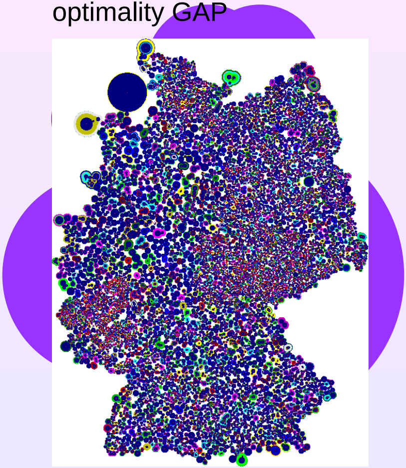
Function	pcb3038	usa13509	pla85900
`lin_kernighan`	141	468	1842
`lin_kernighan` winers	91	261	1169
average number of `flip`	61	99	108
`lk_search`	19,855	95,315	376,897
`lk_search` winers	1,657	9,206	29,126
`flip`	180,073	1,380,545	5,110,340
undo `flip`	172,396	1,336,428	4,925,574
size of `flip`	75	195	607
`flip` size ≤ 5	67,645	647,293	1,463,090
`next`	662,436	6,019,892	14,177,723
`prev`	715,192	4,817,483	13,758,748
`sequence`	89,755	773,750	2,637,757

Implementations:

Implementation	pcb	Instance usa	pla	% total time flip%	n/p%	seq%
arrays	7.2	246.6	10422.5	97	1	1
A + RB	1.6	21.6	265.9	85	1	1
list	50.8	5929.4	>50000	0	93	6
L + 2-search	15.7	426.7	24047.9	0	55	44
L + index + n/p	1.8	65.6	697.3	92	1	1
L + 3 levels	2.3	18.5	81.4	38	19	4
L + 2 layers	1.2	10.1	43.9	26	6	1
Binary Tree	1.4	12.6	52.9	17	24	6

- How do we obtain bounds or warranties?
- Assign disjoint circles to every city.
- Assign disjoint moats to set of cities.
- Add two times each ratio and band-width to get a valid bound.
- How do we find each circle and moat width?

15,112 cities in Germany, 0.74% optimality GAP



Previous Definitions:

- V Set of cities.
- E Set of allowed connections between cities, i.e. $E = \{(a,b) : a,b \in V, a \neq b\}$.
- c Cost of each edge.
- $\delta(S)$ Edges crossing the boundary of set S, i.e. $\delta(S) = \{(a,b) \in E : a \in S, b \in V \setminus S\}$.

IP Formulation:

$$
\begin{aligned}
\min \quad & \sum (c_e x_e : e \in E) \\
s.t. \quad & \sum (x_e : e \in \delta(\{v\})) = 2 \quad \forall v \in V \\
& \sum (x_e : e \in \delta(S)) \geq 2 \quad \forall \emptyset \subsetneq S \subsetneq V \\
& x_e \in \{0,1\} \quad \forall e \in E
\end{aligned}
$$

Some problems of the IP formulation:

- Just as hard as counting possible permutations.
- Number of variables is $|V|(|V|-1)/2$.
- No efficient algorithm is known.

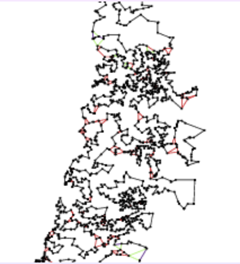
Continuous Relaxation (SEP):

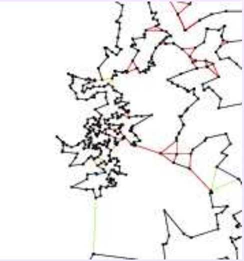
$$
\begin{array}{llll}
\min & \sum (c_e x_e : e \in E) & & \\
\text{s.t.} & \sum (x_e : e \in \delta(\{v\})) = 2 & \forall v \in V & (r_v) \\
& \sum (x_e : e \in \delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V & (W_S) \\
& x_e \in [0,1] & \forall e \in E &
\end{array}
$$

Can be solved efficiently (Ellipsoid method).

Bounds from the SEP relaxation:

0.69% GAP for instance chile5445

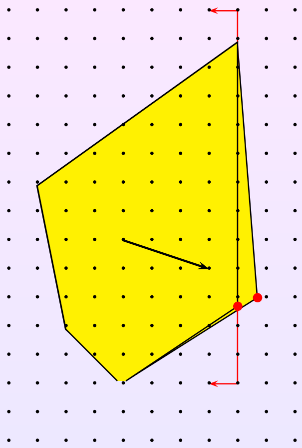




IP through LP

First proposed by Dantzig, Fulkerson and Johnson (1954) for the TSP.

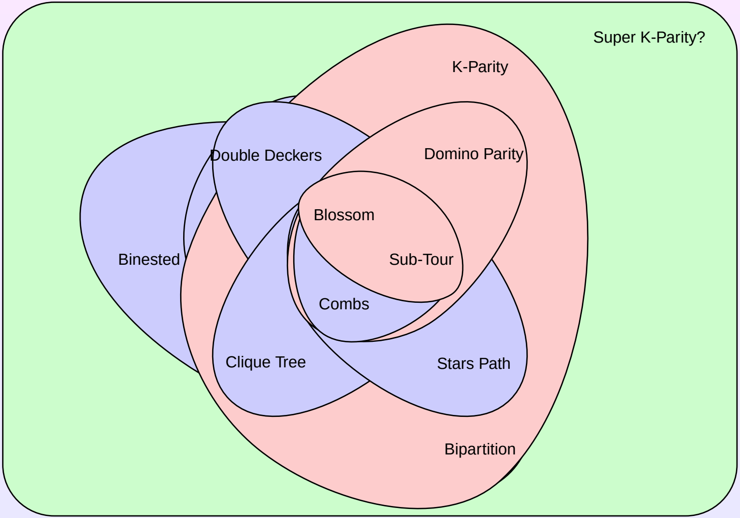
1. Consider continuous relaxation.
2. Let x^* be the continuous optimal solution.
3. Is x^* integer?, then finish.
4. Find valid inequality for integer points.
5. Add it to our LP formulation.
6. Go back to 2.



(Some) structural cuts for the TSP

- Sub-tour (separable)
- Blossom (Edmonds 1965)(separable)
- Combs (Chvátal 1973, Grötschel y Padberg 1979)
- Clique-Tree (Grötschel y Pulleyblank 1986)
- Star, Path (Fleischmann 1988, Cornuéjols et al. 1985)
- Bipartition (Boyd y Cunningham 1991)
- Binested (Nadeff 1992)
- Double Deckers (Applegate et. all 1994)
- Domino Parity (Letchford 2000)(planar)
- K-Parity (Cook et. al. 2004)(planar)

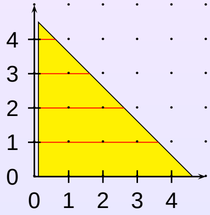
How do they relate?



General IP Cuts

Local Cuts for the TSP

- Idea: get cuts automatically.
- Base: use a simplified version of the problem.
- Example: Gomory-Chvátal cuts (1958).
  - Consider a single (basic) constraint with a fractional integer variable.
  - Rounding of the constraint give us a valid cut.
  - In theory, can solve any IP problem.

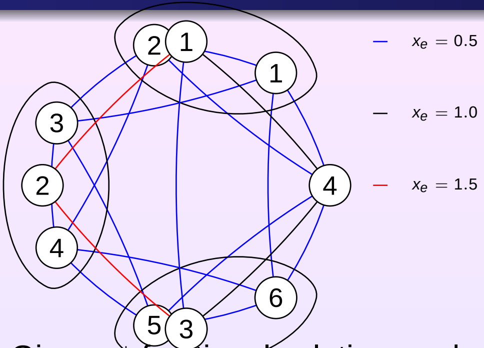


- $x_2 \in \mathbb{Z}, x_1 \in \mathbb{R}^+$
- $P = \{(x_1, x_2) : x_1 + x_2 \leq 4.5\}$.
- $x_1 + x_2 \leq 4.5, x_1 \geq 0 \Rightarrow x_2 \leq 4.5$
- $x_2 \leq 4$.

Non-structured cuts

Local Cuts for the TSP:

- Shrink to a small GTSP (16-48 cities).
- Separate on small problem.
- If successful, add expanded cut to original problem.
- Numerical issues.
- Extension to MIP.
- What if everything fails, what do we do?



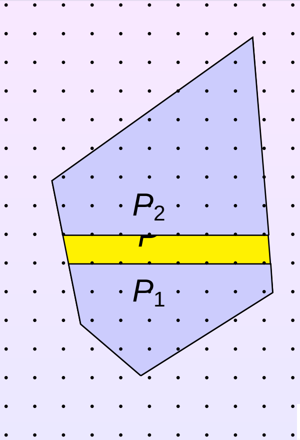
Given x^* fractional solution, and P polyhedron:

$x^* \in P$? Let $\{v_k : k = 1, \ldots, K\}$ extreme points of P.

Between enumeration and Lineal Programming

Strong Branching (divide to conquer)

- Create easier sub-problems.
- Fix variables upper/lower bounds.
- Solve each resulting sub-problem.
- Choose biggest impact.
- Together with cutting plane approach at each node.



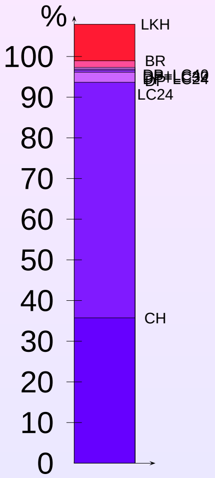
Numerical Examples on chile5445

Optimal Solution: 40011.091Km

Conf.	Value	Time	GAP (%)
Subtour	39755.198	134	0.639
Heuristic separation	39846.738	25518	0.470
Local Cuts (24)	39994.941	14509	0.040
Domino Parity	40001.294	10863	0.024
DP + LC 24	40002.578	14160	0.021
DP + LC 32	40003.294	21159	0.019
DP + LC 40	40004.291	60269	0.017
DP + LC + Branching	40008.475	+3 dias	0.007
LKH	40031.459	46	-0.051
First solution	44594.459	3	-11.455

Numerical Results (Closed GAP over SEP)

Conf.	(%)
Heuristic separation	35.773
Local Cuts (24)	93.689
Domino Parity	96.171
DP + LC 24	96.673
DP + LC 32	96.953
DP + LC 40	97.343
DP + LC + Branching	98.978
LKH	107.960



Conclusions

- TSP offers a reference for IP in general.
- Strategy depends on the real objective:
  - Find feasible solution.
  - Find good solution.
  - Optimality.
- Most important techniques for IP where (are) born in the TSP.
- Importance of having good bounds.
- Numerical Issues.
- Looking for general cuts for MIP (local cuts).

Thanks for your patience!
Questions?

References I

- David Applegate, Robert E. Bixby, Vašek Chvátal, and William Cook, *Finding cuts in the TSP (a preliminary report)*, Tech. Report 95-05, DIMACS, Rutgers University, New Brunswick, NJ, 1995.
- ______, *Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems*, Math. Prog. **97** (2003), 91–153.

References II

- S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems*, Proceedings 33rd Annual Symposium on Foundations of Computer Science (Los Alamitos, CA), IEEE Computer Society Press, 1992, pp. 12–23.
- S. Arora, *Polynomial time approximation schemes for euclidean TSP and other geometric problems*, Proceedings 37th Annual Symposium on Foundations of Computer Science (Los Alamitos, CA), IEEE Computer Society Press, 1996, pp. 2–11.

References III

S. Shani and T. Gonzalez, *P-complete approximation problems*, Journal of the Association for Computing Machinery **23** (1976), 555–565.