7th Combinatorial Optimization Workshop
Aussois, 9–15 March, 2003

Small Instance Relaxations for the Traveling Salesman Problem

Gerhard Reinelt
Klaus Wenger

Discrete Optimization Research Group
Institute of Computer Science
University of Heidelberg

Small Instance Relaxations

- **Definition:** SIR is linear relaxation obtained by lifting inequalities from polytopes of small problem instances. B&C.

- **Philosophy:** Get rid of polyhedral studies; avoid tailored separation procedures; automized wellspring of problem-specific cutting-planes.

- **Ideal:** Easy dimension reduction and easy computation of linear descriptions of low-dimensional problem-specific polytopes.

- **Problems:** How to reduce given to small instance? Complex facial structures.

- **Experience:** Linear Ordering Problem, Weighted Betweenness Problem and TSP.

SIRs for the TSP

- **Past:** Preliminary experiments exploiting STSP(9) by Christof and Reinelt (1996).

- **Pool:** Facets of TSP polytopes STSP(k) with $6 \leq k \leq 10$. Available in SMAPO. [Nor55,BC91,CJR91,CR96,CR01].

k	#facets	#classes
6	100	4
7	3,437	6
8	194,187	24
9	42,104,442	192
10	≥51,043,900,866	≥15,379

- **Representative:** One TT $f^T y \geq f_0$ per symmetry class. Permute indices.

- **Procedure:** support graph (V_n, E_n, x^*), choose k-way cut, shrink graph to k vertices, separate resulting weight vector y^* from STSP(k), eventually lift.

Separation in the Low-Dim Case

- **QAP:** $\mathcal{NP}$-hard. We solve 1 per class. Representative $f^T y \geq f_0$.

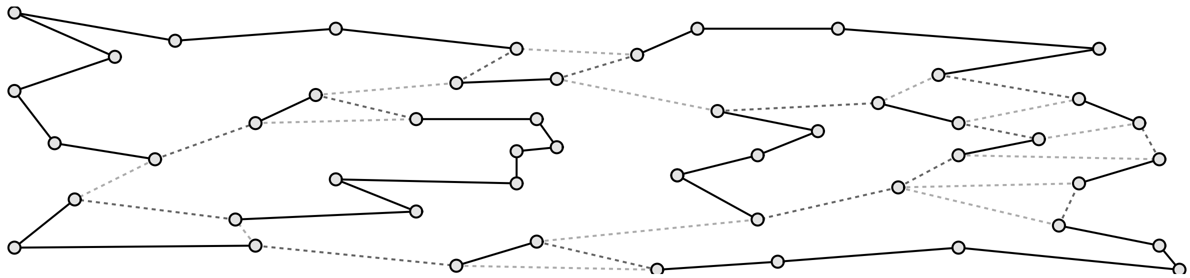
$$\min_{\pi \in \Pi_k} \sum_{\{i,j\} \in E_k} f_{\{\pi(i),\pi(j)\}} y^*_{\{i,j\}} < f_0 \quad ?$$

- **Heuristic:** GRASP-sparse by Pardalos, Pitsoulis and Resende. Per iteration: construct good permutation and improve by local search. Several iterations.

- **Exact:** We compute all π with lhs $< f_0$. B & B originally by Burkard and Derigs.

- **Bottleneck:** Number of classes explodes with increasing k and QAPs get tougher. Low-dim separation is clearly the bottleneck of the SIR approach.

Generation of Small Graphs



- **Universal:** Choice of partition is steered by x^* (not by f). No tailored separation.

- **Wanted:** Small y^* values to violate $\geq$ TT inequalities (coeffs ≥ 0).

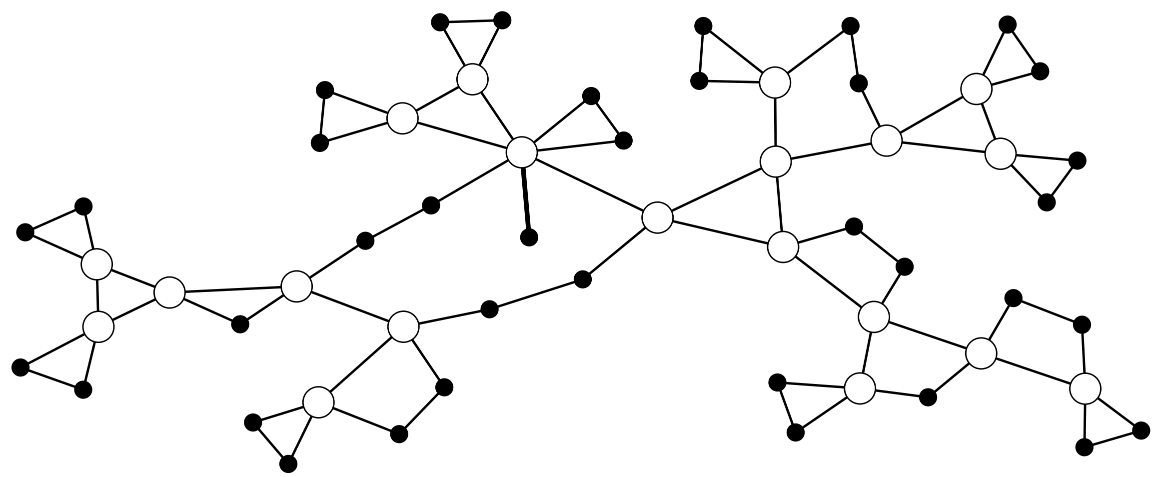
- **Mincuts:** Min k-way cuts. Literature: single cut for $k \leq 6$. We: many for $k \geq 6$.

- **Perfect:** Min k-way cut where every subset is global mincut. We compute ALL such perfect cuts for $6 \leq k \leq 10$.

- **Further:** Near-min k-way cuts [KS96] and local vertex neighbourhoods.

Exploiting the Cactus-Tree

- **Cactus:** Connected graph in which every edge is within at most one cycle.
- **Representation:** Cacti can be used to represent all global mincuts of a weighted graph compactly. Structure visible.



- **Construction:** Very fast for TSP [Wen02].
- **Enumeration:** Using the cactus, we enumerate ALL mincut partitions of a TSP support graph into $6 \leq k \leq 10$ elements. Some skipped later. Short 1-paths.

Many Minimum Cut Partitions

- **Example:** Instance pr2392. Support graph after SEC and simple comb separation. Shrunken 1-paths. Cactus construction: 3442 global mincuts, 0.03 sec (2.6 GHz).

k	number	time in sec
6	1123	0.92
7	1306	1.02
8	1581	1.22
9	1983	1.56
10	2632	2.12

- **Example:** Support graph of intance pcb1173 with 1472 global mincuts.

k	number	time in sec
6	3384	1.11
7	5971	1.99
8	10907	3.07
9	19984	6.54
10	35522	11.66

Padberg-Rinaldi Shrinking

- **Shrinkable:** $S \subset V_n$ called shrinkable if weight vector is outside STSP after shrinking S when it was outside before shrinking.

- **Which:** [PR91] give sufficient condition for S to be shrinkable. Usually only applied for $|S| = 2, 3$. With cactus: higher $|S|$ ok. Also use complete linear descriptions of STSP(k). Diminishing return.

- **Why:** We use PR shrinking (also 1-squares) to remove redundancy from support graph and to reduce huge number of shrunken graphs before QAPs.

- **Failure:** We did not find further sufficient conditions for save shrinking.

Separation Strategy

- **Quality:** We consider mincut partitions to be of highest quality. Near-min k-way cuts and vertex neighbourhoods only after mincut partitions.

- **Increasing:** We start with $k = 6$ and only $k++$ if no cut found using STSP(k).

- **Skipping:** If PR shrinking can be applied to a small graph, we can skip the graph safely (use what you already know).

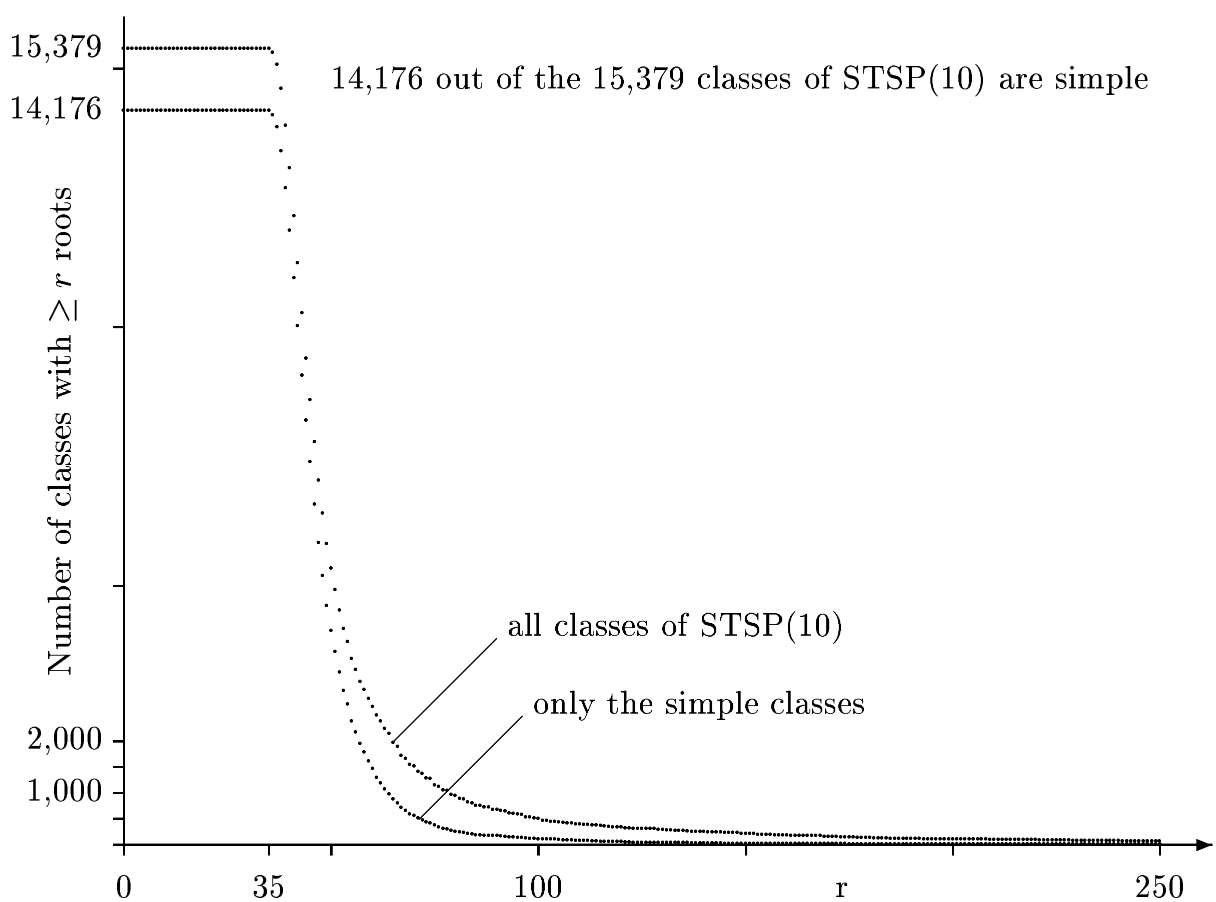
- **Isomorphism:** We maintain a pool of small graphs for which no cut was found. A new shrunken graph is only tested if it is not isomorphic to one in the pool.

- **Speed:** GRASP for QAP in practice.

Making Choices for STSP(10)

- **Overflow:** Too time consuming to exploit all classes. Experiments → go for facets with many roots. Rarely cuts with few roots and they don't stay binding.

- **Choice:** At least 100 roots → 500 classes.



Some Computational Results

Only mincut partitions, GRASP.

- **rat195:**

	SCs	SCs + SIR
B&C nodes	37	21
Time/sec	26	107
#SIR cuts	0	1393
#Tested	0	649
#Success	0	85
#PR skip	0	106
#Iso skip	0	72

- **d198:**

	SCs	SCs + SIR
B&C nodes	17	7
Time/sec	7	32
#SIR cuts	0	963
#Tested	0	497
#Success	0	106
#PR skip	0	1001
#Iso skip	0	388

Comparison with Local Cuts

- **Reference:** ABCC. TSP Cuts Which Do Not Conform to the Template Paradigm. Jünger/Naddef (Eds.) *LNCS 2241*, 2001.

- **Similarity:** Both SIR and local cut approach by ABCC choose partitions, shrink, separate in low-dim space, and lift.

- **Templates:** important basis for both.

	SIR	ABCC
template?	small STSPs	no fall back
#cuts/graph	0 or many	0 or 1
speed/graph	slower (QAP)	faster
maximal k	10	about 30
scaleability?	restricted	better
space per cut	usually more	less (V_0)
facets of?	STSP	GTSP
are valid?	verifiable	see proofs
cut effect	some global	all local
cut choices?	roots/angle...	no: 1 cut
partitions	min k-way	BFS neighbhds

Conclusions

Small Instance Relaxations work for the TSP.

SIR more successful for LOP than for TSP.

SIR as alternative approach to solving TSPs.

SIR alone or on top of existing templates.

Number of subproblems can be reduced.

QAPs cause problems with running times.

Classical templates are a very good basis.

Enumeration of mincut partitions interesting.

Extensive tuning is still required.

There will be paper with full details.