A Speed-up for Helsgaun's TSP Heuristic by Relaxing the Positive Gain Criterion

Sabrina Ammann, Birte Ostermann, Sebastian Stiller, Timo de Wolff

Traveling Salesman Problem (TSP)

- **Complete, undirected graph** $G = (V, E)$
- **Vertices** $V = \{1, \ldots, n\}$
- **Edges** $E = \{\{i,j\} \mid i,j \in V,\ i \neq j\}$ with **costs** assigned by cost function $c : E \to \mathbb{R}_{>0}$
- A **tour** T is a **Hamiltonian cycle** of G, i.e. a cycle that includes every vertex exactly once.

Goal: Find a tour T in which the **sum of costs** $c(T)$ over the edges traversed is **minimal**.
Challenge: The TSP is NP-hard!
$\Rightarrow$ **Heuristics** can find a tour in a relatively short running time without guarantee for optimality.

A tour through 15112 cities in Germany: Instance *d*15112 of TSPLIB [3].

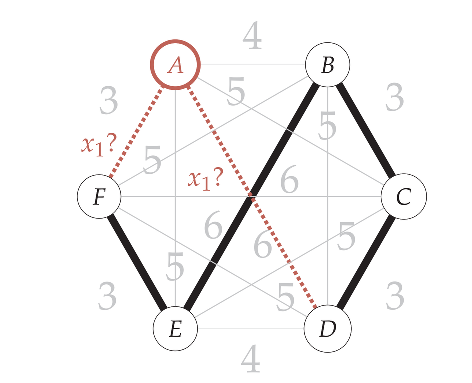
Lin-Kernighan-Helsgaun (LKH)

LKH is a state-of-the-art TSP heuristic that iteratively tries to improve a given tour T, i.e. tries to establish a new tour T' with $c(T') < c(T)$.
Idea: Exchange edges along **alternating cycles** $(x_1, y_1, \ldots, x_k, y_k)$:

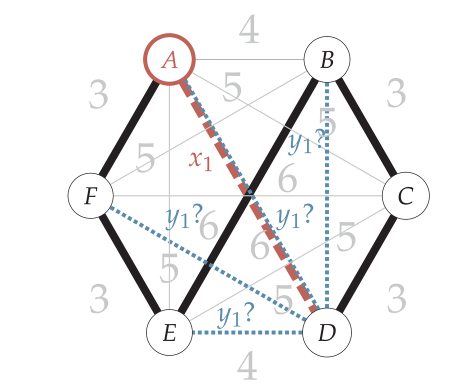
- Select edges x_i to be deleted from T and edges y_i to be added
- Exchanging edges x_i, y_i, we restrict to certain **heuristic criteria**:
- E.g.: Total cost difference of deleted and added edges is positive

Positive Gain Criterion [1]: Given deleted edges $x_1, \ldots, x_i$ and added edges $y_1, \ldots, y_{i-1}$ we limit the selection for y_i by the condition:

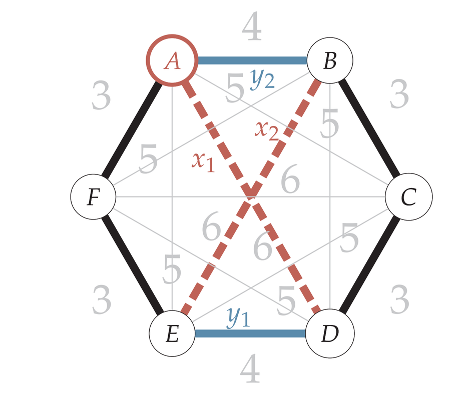
$$\text{Gain } G_i = \sum_{\ell=1}^{i} c(x_\ell) - c(y_\ell) > 0$$



Start tour: $c(\mathbf{T}) = 24$. At random vertex A, choose edge x_1 to delete.



Choose y_1 to be added: Selecting (D, E) gives gain $G_1 = 6 - 4 = 2$.

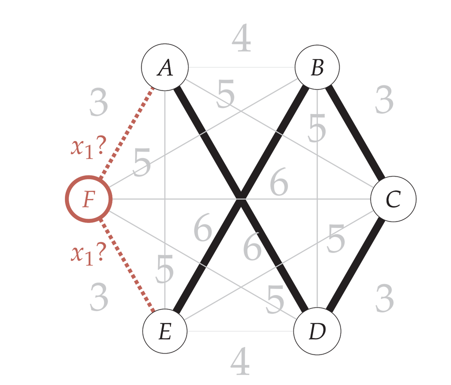


After the next edge exchange, we obtain the tour T' with $c(T') = 20$.

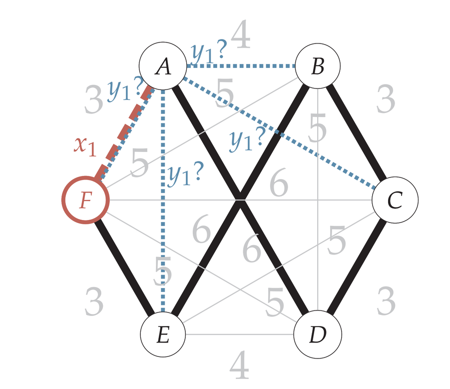
Relaxed Positive Gain Criterion

Given deleted edges $x_1, \ldots, x_i$ and added edges $y_1, \ldots, y_{i-1}$ we limit the selection for y_i by the following condition (R), assuming $G_0 > 0$:

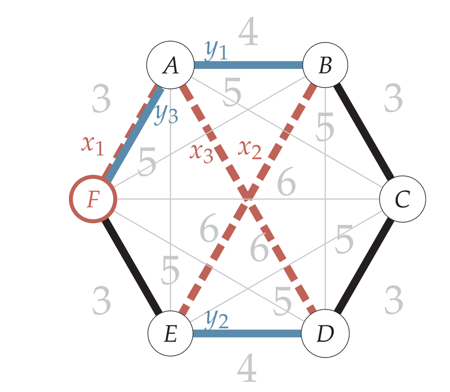
$$G_{i-1} = \sum_{\ell=1}^{i-1} c(x_\ell) - c(y_\ell) > 0 \quad \textbf{or} \quad G_i = \sum_{\ell=1}^{i} c(x_\ell) - c(y_\ell) > 0 \quad \text{(R)}$$



$c(\mathbf{T}) = 24$. At vertex F, select x_1.



Select y_1: We now **allow** $G_1 \leq 0$.



After edge exchanges: $c(T') = 20$.

Experimental Setup

We implement (R) in LKH 3.0.8 [2] by K. Helsgaun and compare to original code on various benchmark instances, e.g. from TSPLIB [3].

- Computer cluster: one node, 256 AMD EPYC 7742 64-CPU
- 10 or 100 runs per instance
- Test candidate set types [1,4] and number of candidates
- Further algorithm engineering

Computational Results

Relaxed LKH vs. LKH 3.0.8 with POPMUSIC [4] candidate sets and five candidates. Display of the **averaged** results over all **small**, **medium** and **large** instances, resp.

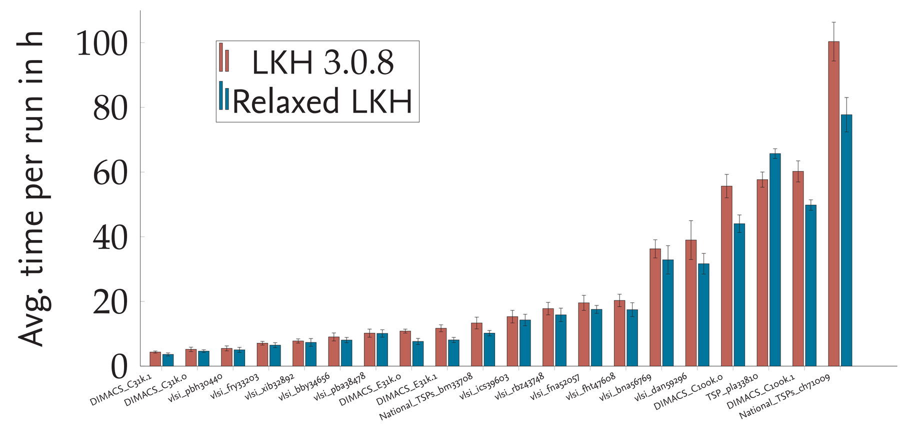
Version	Problem size	#Instances	Minimal cost	Avg. gap to optimum	Avg. time per run
LKH 3.0.8	small	123	$3,031,129$	0.0077%	3.5 s
Relaxed LKH	$(n < 1,000)$		$3,031,129$	0.0072%	**4.0 s**
LKH 3.0.8	medium	140	$4,865,171$	0.083%	42.5 min
Relaxed LKH	$(1,000 \leq n \leq 30,000)$		$4,864,873$	0.076%	40.9 min
LKH 3.0.8	large	20	$32,826,656$	Optimum	25.35 h
Relaxed LKH	$(30,000 < n < 1 \times 10^6)$		$32,824,384$	not known	21.91 h

Using the Relaxed Positive Gain Criterion:

- The **quality of solutions** does not differ significantly.
- Small instances: On average 0.5 s (14%) increase of running time per run.
- Medium instances: On average ca. 90 s (4%) decrease of running time per run.
- **Large instances**: On average **3.4 h** (**13.6%**) **decrease** of **running time per run**, for individual large instances, up to 30% decrease of running time.

Speed-up on Large Instances

Relaxed LKH vs. LKH 3.0.8 in average time per run on large instances that completed ten runs in one month.



References

[1] Helsgaun, K., *An effective implementation of the Lin–Kernighan traveling salesman heuristic*, European journal of operational research 126 (2000), no. 1, 106–130.
[2] Helsgaun, K., *LKH*, http://webhotel4.ruc.dk/~keld/research/LKH-3/, January 2023, Version 3.0.8, accessed 03/23.
[3] Reinelt, G., *TSPLIB—a traveling salesman problem library*, ORSA journal on computing 3 (1991), no. 4, 376–384.
[4] Taillard, E. D., Helsgaun, K., *POPMUSIC for the Travelling Salesman Problem*, European Journal of Operational Research 272 (2019), no. 2, 420–429.

Find our paper on ArXiv 2401.16149

Technische Universität Braunschweig

ProvideQ