On the Effectiveness of k-Opt for Euclidean TSP

Vincent Cohen-Addad   Claire Mathieu

Sorbonne Université and CNRS   Université Paris-Diderot and CNRS

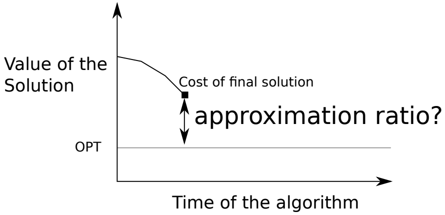
BIRS TSP workshop Sep.23-28

Local Search Algorithm

$L \leftarrow$ Arbitrary feasible solution
While $\exists\ L'$ s.t $L' \in$ Local Neighborhood (L) **and** $\text{cost}(L') < \text{cost}(L)$
$\quad L \leftarrow L'$
Output L

Why Local Search?

- Easy to implement
- Easy to run in parallel
- Competitive results in practice



Practical Success of Local Search Heuristics for TSP

- Good competitor for the DIMACS TSP challenge of early 2000s according to the report by D. Johnson et al.. In particular for Euclidean inputs.
- Best known alg. (Arora and Mitchell approach) were considered too slow for the quality of their output.

Practical Success of Local Search Heuristics for TSP

- Good competitor for the DIMACS TSP challenge of early 2000s according to the report by D. Johnson et al.. In particular for Euclidean inputs.
- Best known alg. (Arora and Mitchell approach) were considered too slow for the quality of their output.
- 2-OPT is $O(1)$-approx. for random inputs and converges in polynomial time [Englert, Roeglin, Voecking '07]

Local Search for Geometric Optimization

Previous Work: $(1+\varepsilon)$-Approximation

[N. Mustafa & S. Ray '09]: **Hitting Set Problem**

[T. Chan & S. Har-Peled '09]: **Independent Set for Pseudo-Disks**

[E. Krohn & M. Gibson & G. Kanade & K. Varadarajan '14]: **Terrain Guarding**

[C.-A., Klein, Mathieu '16 Friggstad, Rezapour, Salavatipour'16] **Clustering problems**.

etc.

Neighborhood size depends on ε.

This Talk:

- **Traveling Salesman Problem**

Our Results

Random inputs

Local search achieves a $(1+\varepsilon)$ approximation for random inputs in $\mathbb{R}^{O(1)}$.

Worst-case

There exist inputs and initial solutions such that local search may output a solution of cost at least 2OPT.

Local Search for TSP

Algorithm

$L \leftarrow$ Arbitrary feasible solution
While $\exists\ L'$ s.t. L and L' differ by $\mathcal{O}(1/\varepsilon^2)$ edges **and** $\text{cost}(L') < \text{cost}(L)$
    $L \leftarrow L'$
Output L

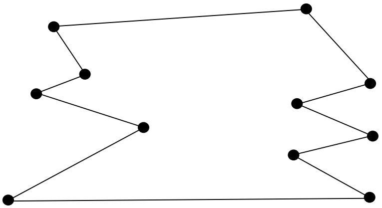


Figure: A step of the Local Search Algorithm

Local Search for TSP

Algorithm

$L \leftarrow$ Arbitrary feasible solution
While $\exists\ L'$ s.t. L and L' differ by $\mathcal{O}(1/\varepsilon^2)$ edges **and** $\text{cost}(L') < \text{cost}(L)$
$\quad L \leftarrow L'$
Output L

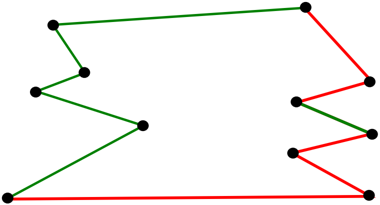


Figure: A step of the Local Search Algorithm

Local Search for TSP

Algorithm

$L \leftarrow$ Arbitrary feasible solution
While $\exists\ L'$ s.t. L and L' differ by $\mathcal{O}(1/\varepsilon^2)$ edges **and** $\text{cost}(L') < \text{cost}(L)$
$\quad L \leftarrow L'$
Output L

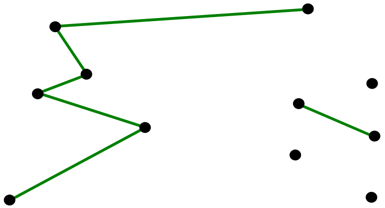


Figure: A step of the Local Search Algorithm

Local Search for TSP

Algorithm

$L \leftarrow$ Arbitrary feasible solution
While $\exists\ L'$ s.t. L and L' differ by $\mathcal{O}(1/\varepsilon^2)$ edges **and** $\text{cost}(L') < \text{cost}(L)$
    $L \leftarrow L'$
Output L

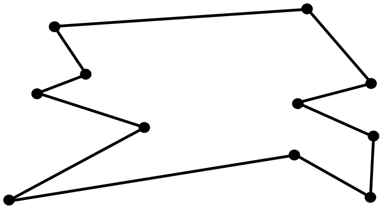


Figure: A step of the Local Search Algorithm

Theorem for today

Local search w/ neighborhood size $O(1/\varepsilon^2)$ outputs a $(1+\varepsilon)$ approx for inputs consisting of points sampled uniformly in $[0,1]^2$.

Sketch.

Intuition for the analysis: Divide the plane into regions such that

- each region contains few points;
- sum of perimeters of regions is short.

This breaks the plane into independent subproblems that are

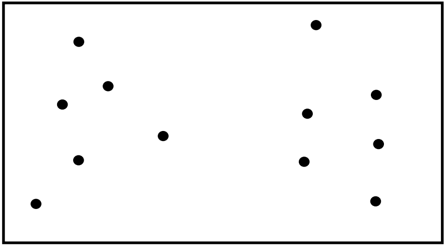
1. solved optimally by the algorithm; and
2. whose solutions are cheap to combine.

□

Defining a Recursive Dissection

Assume points are in $[0,1]^2$.

Split the box using horizontal or vertical lines and splitting into two equal size sub-boxes. Stop when the box has less than $1/\varepsilon^2$ points.



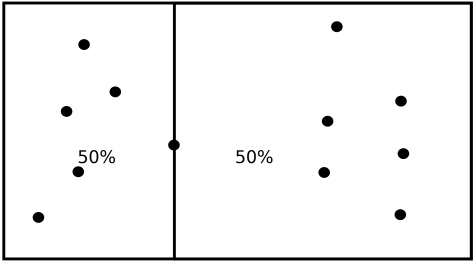
R. Karp '77

Sum of perimeters of all the boxes is $O(\varepsilon\sqrt{n})$.
Namely, $\sum_{b \in \mathcal{B}} |\text{perimeter}(b)| = O(\varepsilon\sqrt{n})$.

Defining a Recursive Dissection

Assume points are in $[0,1]^2$.

Split the box using horizontal or vertical lines and splitting into two equal size sub-boxes. Stop when the box has less than $1/\varepsilon^2$ points.



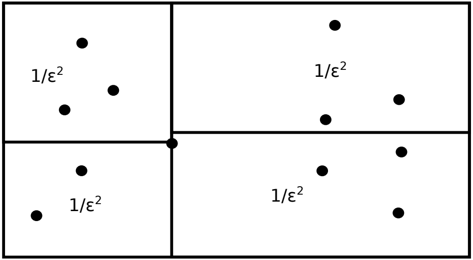
R. Karp '77

Sum of perimeters of all the boxes is $O(\varepsilon\sqrt{n})$.
Namely, $\sum_{b \in \mathcal{B}} |\text{perimeter}(b)| = O(\varepsilon\sqrt{n})$.

Defining a Recursive Dissection

Assume points are in $[0,1]^2$.

Split the box using horizontal or vertical lines and splitting into two equal size sub-boxes. Stop when the box has less than $1/\varepsilon^2$ points.

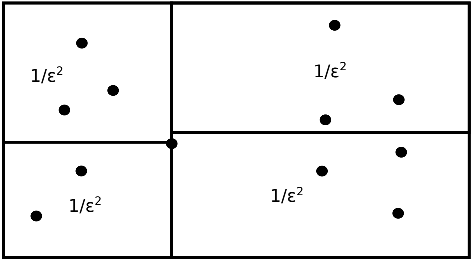


R. Karp '77

Sum of perimeters of all the boxes is $O(\varepsilon\sqrt{n})$.
Namely, $\sum_{b\in\mathcal{B}} |\text{perimeter}(b)| = O(\varepsilon\sqrt{n})$.

A Partition into Boxes

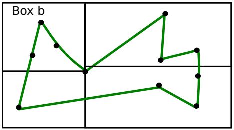
At the end: Each box contains at most $1/\varepsilon^2$ input points.



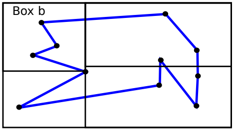
Analysis

Each box has at most $1/\varepsilon^2$ points.

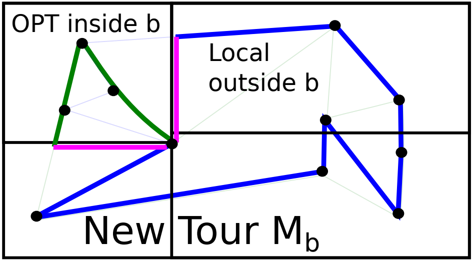
Optimal solution:

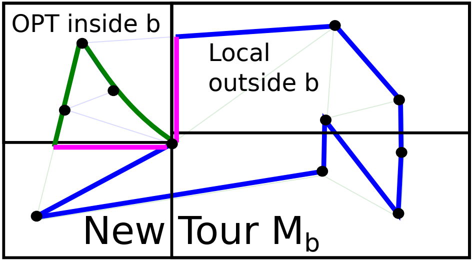


Locally optimal solution:

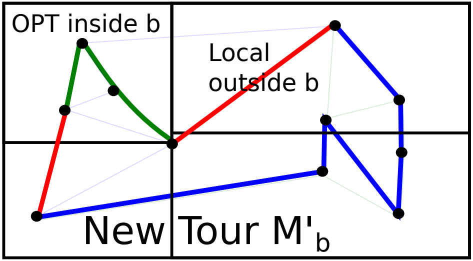


Mixed solution for box b:





Can *glue* the optimal tour within the box to the local tour outside the box by adding twice boundary of the box.



$\text{Cost}(M'_b) \le \text{Cost}(M_b)$.

M'_b differs from L by $O(1/\varepsilon^2)$ edges

Outcome

By local optimality $\text{Cost}(L) \leq \text{Cost}(M'_b) \leq \text{Cost}(M_b)$

$\text{Cost}(L) = \sum_{b'} \text{Cost}(L \text{ in } b')$

$\text{Cost}(M_b) = \text{Cost}(\text{OPT in } b) + 2\text{Perimeter of } b + \sum_{b' \neq b} \text{Cost}(L \text{ in } b')$

Local optimality implies:

$\text{Cost}(L \text{ in } b) \leq \text{Cost}(\text{OPT in } b) + 2\text{Perimeter of } b.$

Upper bound cost(L)

$$
\begin{aligned}
\mathrm{Cost}(L) &= \sum_b \mathrm{Cost}(L \text{ inside } b) \overset{\substack{\text{local}\\\text{optimality}}}{\leq} \sum_b \mathrm{Cost}(M_b \text{ inside } b) + 2|\text{perimet.}(b)| \\
&= \sum_b \mathrm{Cost}(\mathrm{OPT} \text{ inside } b) + \sum_b 2|\text{perimeter}(b)| \\
&= \mathrm{OPT} + \sum_b 2|\text{perimeter}(b)| \overset{\substack{\text{Karp's}\\\text{Lemma}}}{=} \mathrm{OPT} + \mathcal{O}(\varepsilon\sqrt{n}).
\end{aligned}
$$

Analysis

Theorem (Worst-Case)

Local Search produces a tour whose length is at most $(1+\varepsilon)OPT + \mathcal{O}(\varepsilon\sqrt{n})$.

Random Input Case

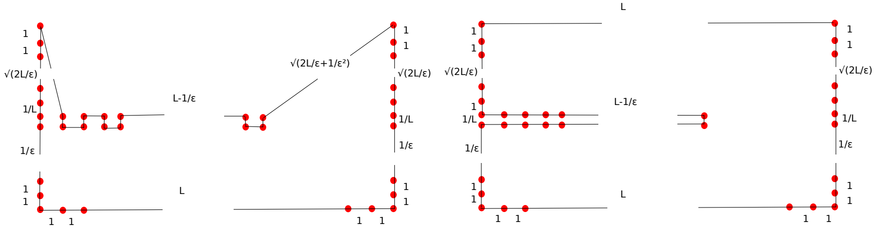
[J. Beardwood, J. Halton and J. Hammersley '59]
$\mathrm{OPT} = \Theta(\sqrt{n})$ with high probability.

Corollary (Random Input)

The Local Search Algorithm produces a tour whose length is at most $(1+\mathcal{O}(\varepsilon))OPT$.

Lower bound

In the worst-case, a local optimal could be a factor 2 away from a global optimal



Our Results

Random inputs

Local search achieves a $(1+\varepsilon)$-approximation for random inputs in $R^{O(1)}$.

Worst-case

There exist inputs and initial solutions such that local search may output a solution of cost at least 2OPT.

Extensions

More general distributions e.g.:[Englert, Roeglin, Voecking '07]?

Can we prove that Lin–Kernighan is better than k-OPT?

Can we find a fast implementation for k-OPT in the plane?

Other ways to go beyond the worst-case: what if OPT is *clear*?

Extensions

More general distributions e.g.:[Englert, Roeglin, Voecking '07]?

Can we prove that Lin–Kernighan is better than k-OPT?

Can we find a fast implementation for k-OPT in the plane?

Other ways to go beyond the worst-case: what if OPT is *clear*?

Thanks for your attention!