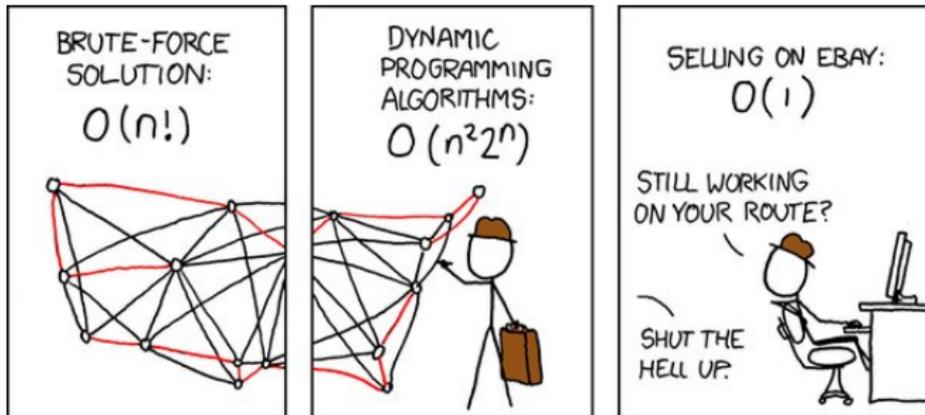


Combinatorial Optimization - Lecture 14 - TSP

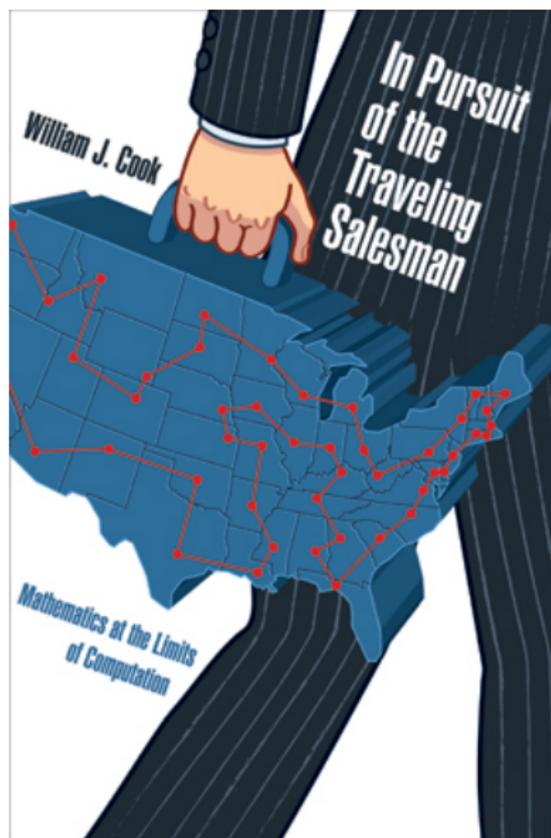
EPFL

2012



Plan

- Simple heuristics
- Alternative approaches
- Best heuristics: local search
- Lower bounds from LP
- Moats



Simple Heuristics

- Nearest Neighbor (NN)
- Greedy
- Nearest Insertion (NI)
- Farthest Insertion (FI)

From ConcordeTSP app →

Algorithm Choice

Exact Solver

Tour Heuristics

TSP Art

Geo Duality

Edge Gen

TSP Move

Compute a heuristic Tour

NN

Greedy

NI

Algo

FI

Christo

2-Opt

3-Opt

LK

CLK

Initial Tour

NN

Random

Animation

On

Off

Load

Log

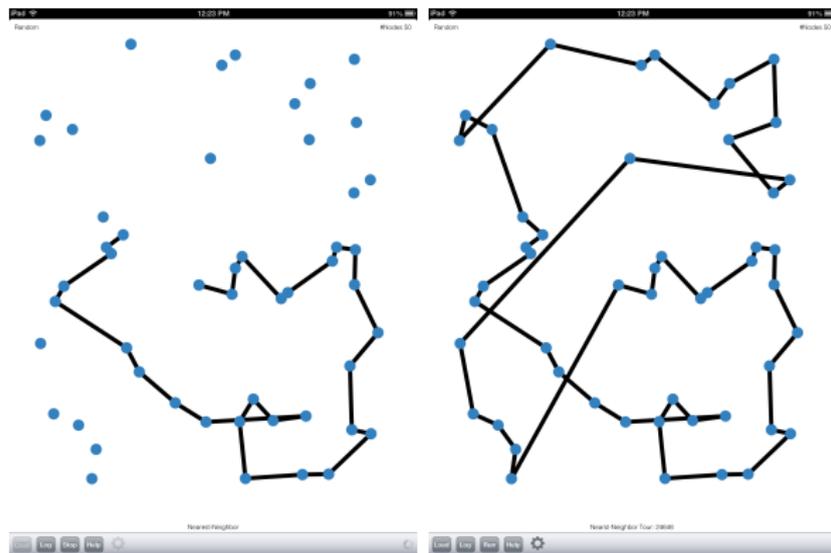
Run

Help



Nearest Neighbor

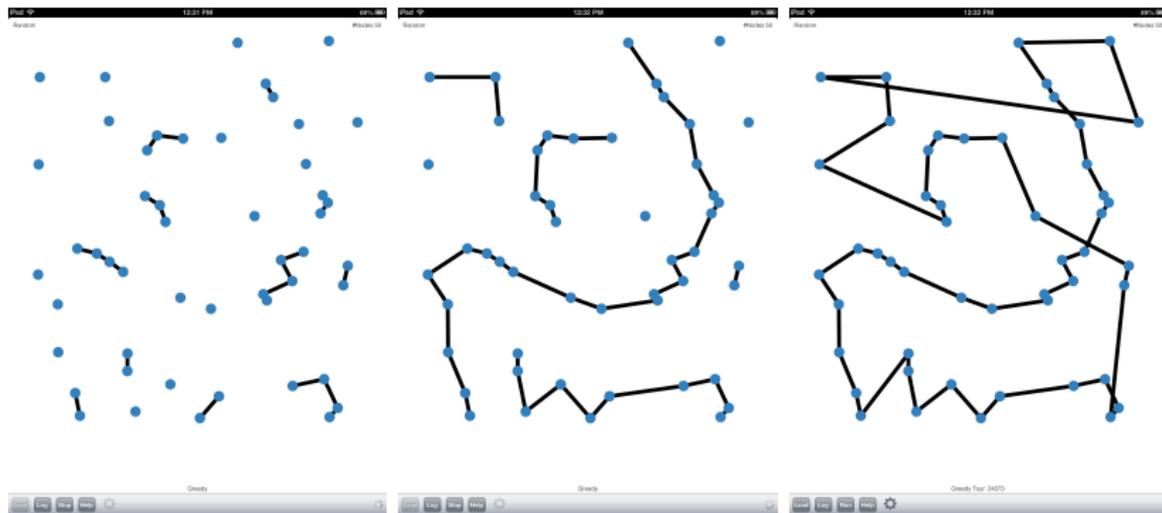
Starting from a random vertex, repeatedly add an edge to its nearest neighbor, unless it creates a non-Hamilton cycle.



- The last edge can be forced to be arbitrarily bad, so cannot be a k -approximation for any k .
- Average performance: $\kappa = 1.25$
- Advantage: Salesman makes it up as he goes along.

Greedy

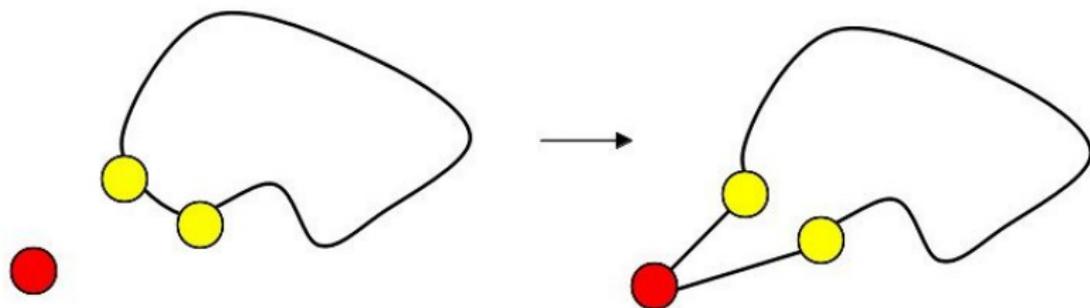
Repeatedly add the shortest edge, unless it creates a degree-3 vertex or a non-Hamiltonian cycle



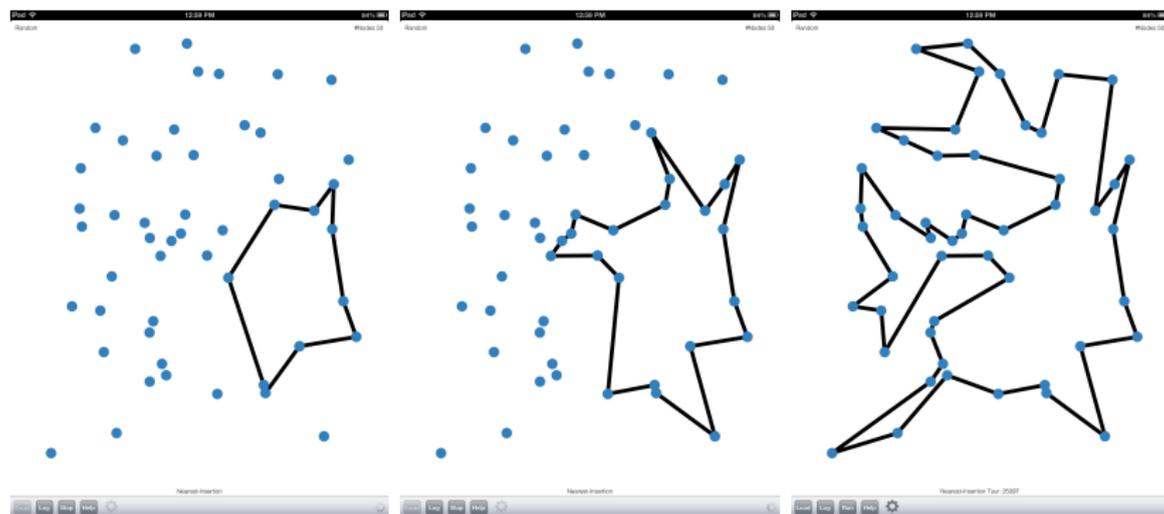
- Also no k -approximation.
- Average performance: $\kappa = 1.14$

Nearest Insertion

Start with a 3-cycle, take the vertex nearest to the cycle, and insert this in the cheapest way.



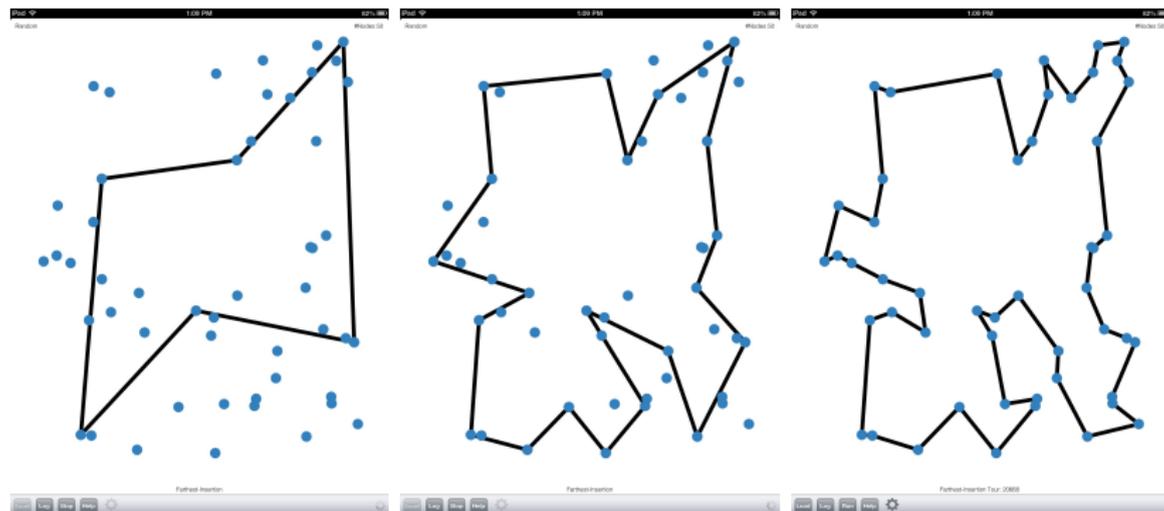
Nearest Insertion



- 2-approximation algorithm
- $\kappa = 1.20$

Farthest Insertion

Start with a 3-cycle, take the vertex **farthest** from the cycle, and insert this in the cheapest way.



- Not known if it is a k -apx...
- But $\kappa = 1.10!$

Comparison

	$k - \text{apx}$	κ	speed
Nearest Neighbor	no	1.25	fast
Greedy	no	1.14	fast
Nearest Insertion	2	1.20	fast
Farthest Insertion	?	1.10	fast
Christofides	1.5	1.09	slow

The averages are from two survey papers. They are obtained by testing the heuristics many times on standard libraries with examples of different sizes, mostly random Euclidean instances.

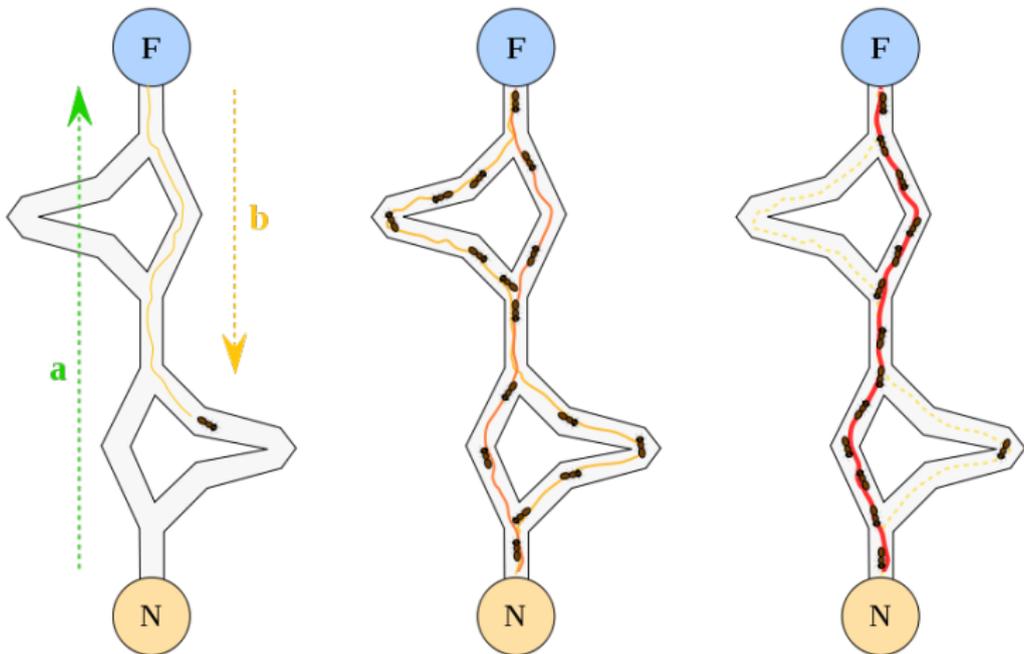
Alternative/silly approaches

- DNA
- Bacteria
- Amoeba
- Bees, pigeons, chimps...
- Humans
- **Ants**
(I, for one, welcome our new insect overlords)
- **Genetic Algorithms**
- Neural Networks



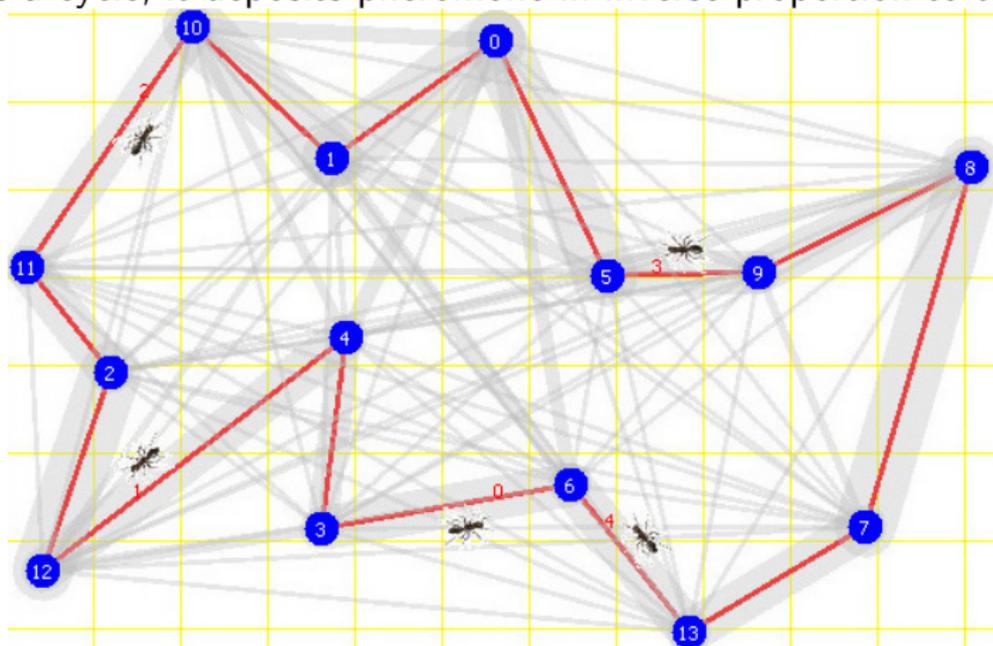
Ant Colony Optimization

Ants find shortest paths from Nest to Food by leaving pheromone on any path that led them to the Food. At intersections they prefer the branch with more pheromone.



Ant Colony Optimization

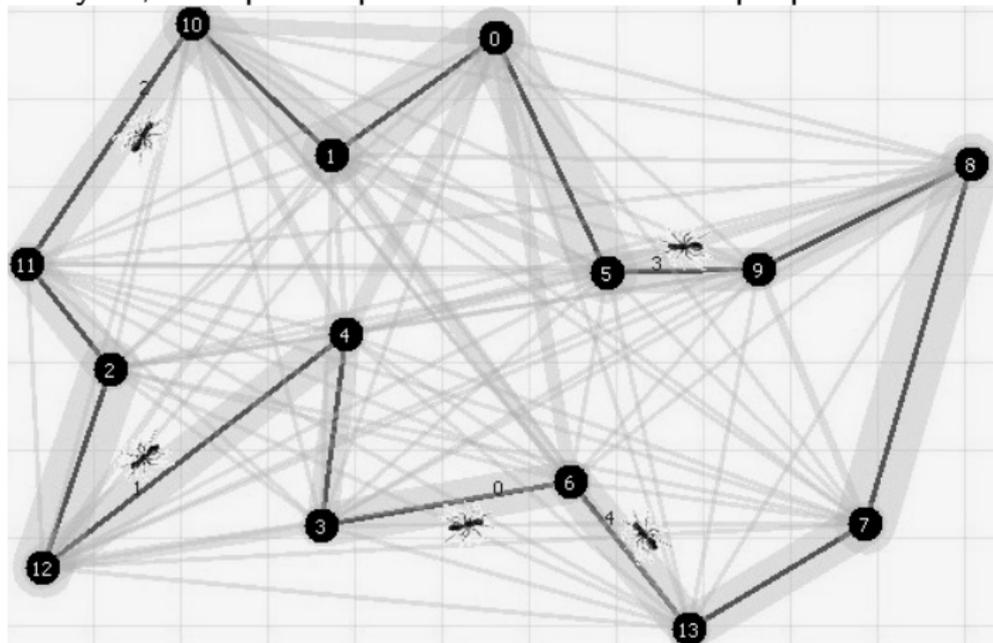
Something similar can be done for TSP. Each will remember which vertices it has already visited, and not visit those again. When it finishes a cycle, it deposits pheromone in inverse proportion to the length.



Reasonably effective, though not as good as the heuristics we saw. But interesting to study for other reasons.

Ant Colony Optimization

Something similar can be done for TSP. Each will remember which vertices it has already visited, and not visit those again. When it finishes a cycle, it deposits pheromone in inverse proportion to the length.

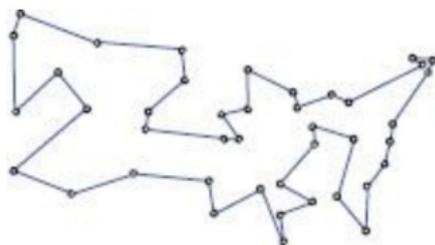


Reasonably effective, though not as good as the heuristics we saw. But interesting to study for other reasons.

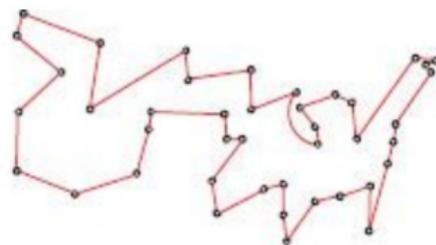
Genetic Algorithms (letting solutions evolve)

For instance, the blue and red tours mate by taking an alternating cycle in their union, then taking the symmetric difference with that alternating cycle, and making sure the result is a tour. This is repeated over many generations.

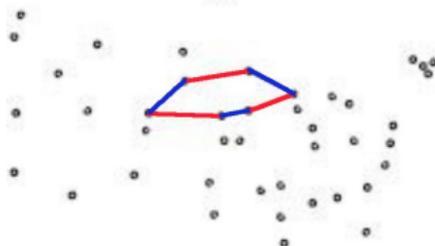
Effective; a version like this holds several records for huge examples.



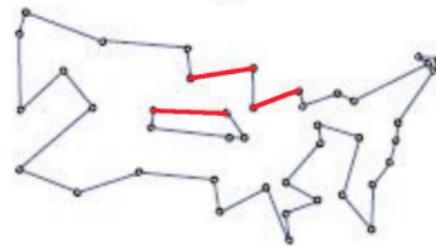
(1)



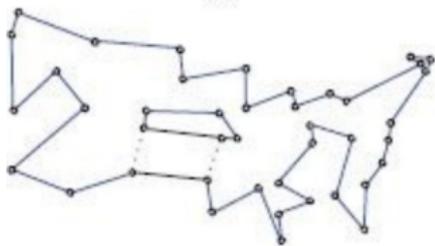
(2)



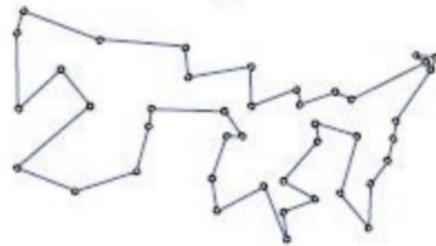
(3)



(4)



(5)



(6)

Local Search

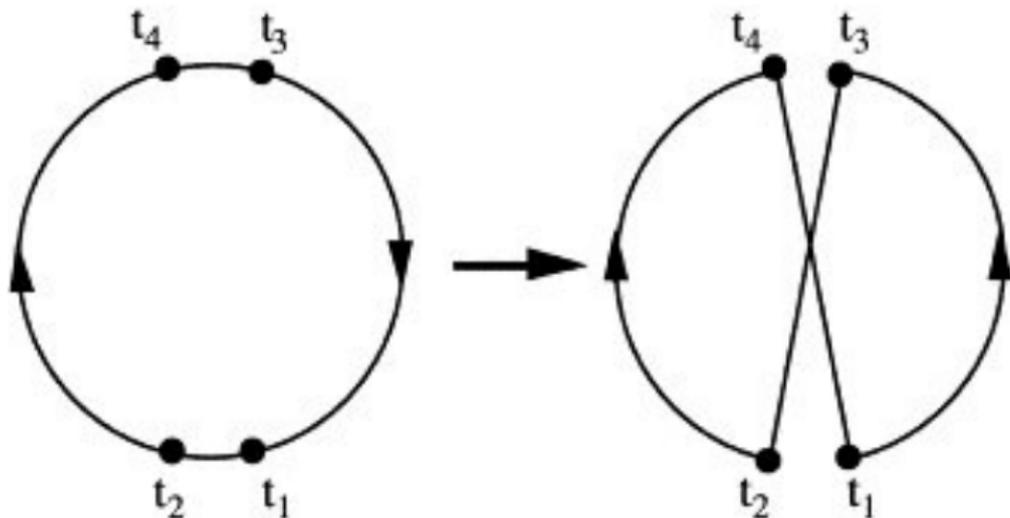
Starting from any Hamilton cycle, make iterative improvements.

- 2-opt
- 3-opt
- 4-opt, 5-opt, ...
- Lin-Kernighan (variable k -opt)

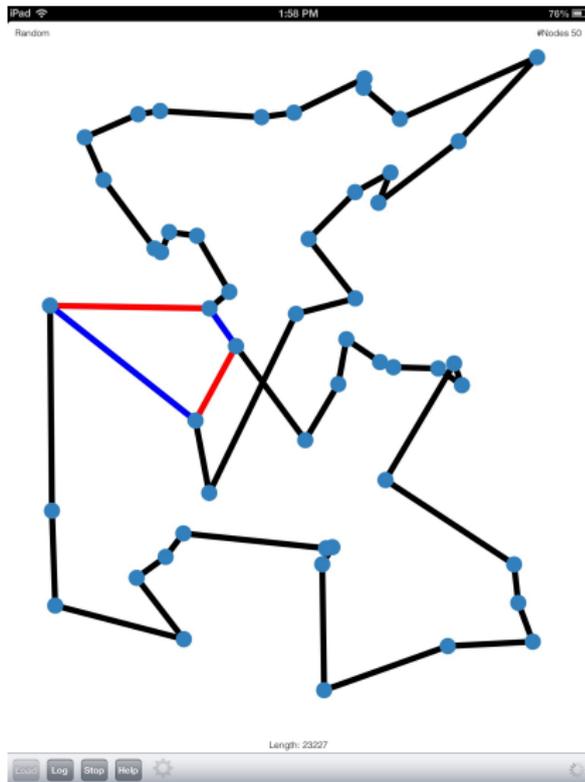
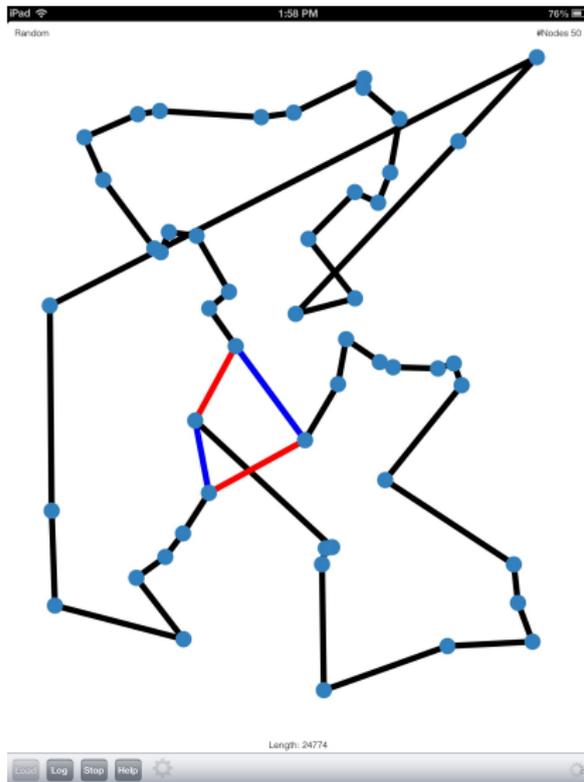
No approximation bounds; may get stuck in local optimum.

2-opt

In a **2-opt move** we remove 2 non-adjacent edges from the Hamilton cycle, then replace them by the 2 edges that give you another Hamilton cycle.

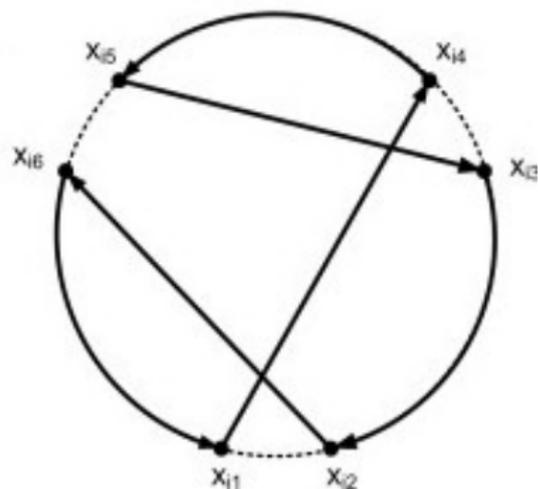
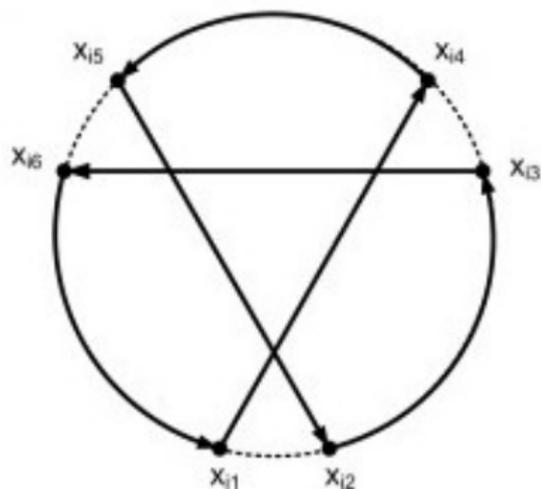


2-opt

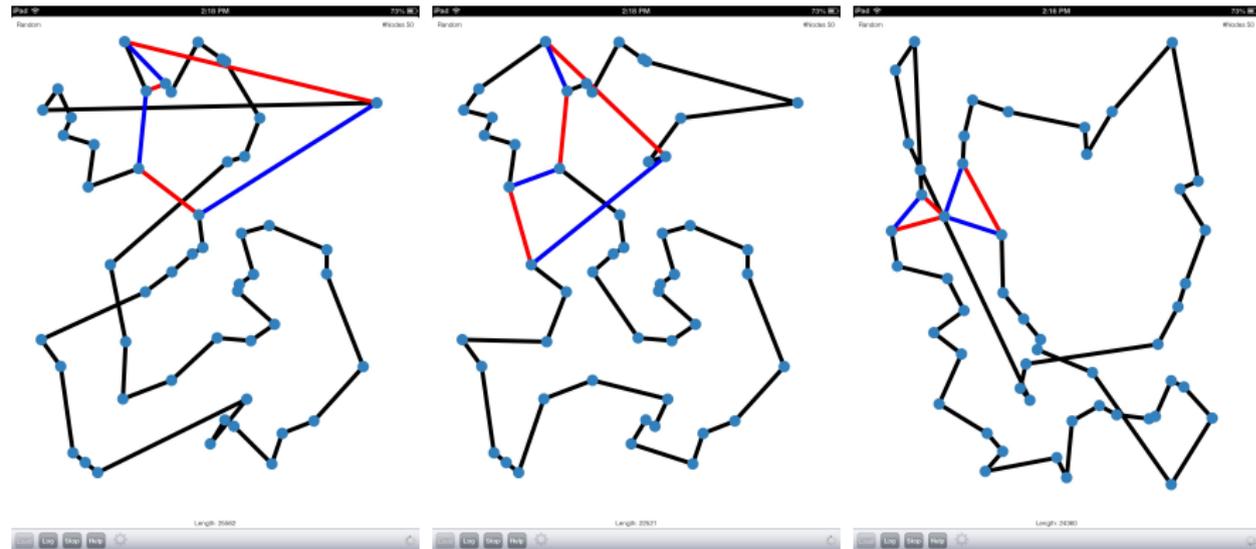


3-opt

Remove 3 non-adjacent edges from the Hamilton cycle, then replace them by the cheapest 3 edges that give you another Hamilton cycle.



3-opt



Comparison

	$k - \text{apx}$	κ
Nearest Neighbor	no	1.25
Nearest Insertion	2	1.20
Greedy	no	1.14
Farthest Insertion	?	1.10
Christofides	1.5	1.09
2-Opt	no	1.05
3-Opt	no	1.03
Lin-Kernighan	no	1.02
Chained Lin-Kernighan	no	1.01

- 2-Opt and 3-Opt are started on a Nearest Neighbor cycle, then repeated until they can improve the cycle no more.
- Lin-Kernighan is more complicated, but basically looks for a k -opt that makes the biggest improvement.
- Chained Lin-Kernighan adds “kicks”, random k -opts to get away from local minima.

Integer Program

IP for TSP

$$\text{minimize } \sum_{e \in E(G)} w_e x_e \quad \text{with } 0 \leq x \leq 1, x \in \mathbb{Z},$$

$$\text{(degrees)} \quad \sum_{e \in \delta(v)} x_e = 2 \quad \text{for } v \in V,$$

$$\text{(subtours)} \quad \sum_{e \in \delta(S)} x_e \geq 2 \quad \text{for } S \subset V(G) \quad (S \neq \emptyset, V(G)).$$

- The minimum integral solutions are exactly the minimum Hamilton cycles.
- But the number of constraints is exponential: There are roughly $2^{|V(G)|}$ of the subtour constraints.

Subtour LP Relaxation

$$\text{minimize } \sum_{e \in E(G)} w_e x_e \quad \text{with } 0 \leq x \leq 1, \quad x \in \mathbb{Z},$$

$$\text{(degrees)} \quad \sum_{e \in \delta(v)} x_e = 2 \quad \text{for } v \in V,$$

$$\text{(subtours)} \quad \sum_{e \in \delta(S)} x_e \geq 2 \quad \text{for } S \subset V(G) \quad (S \neq \emptyset, V(G)).$$

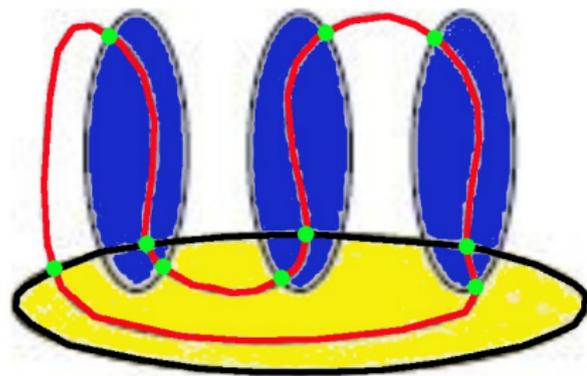
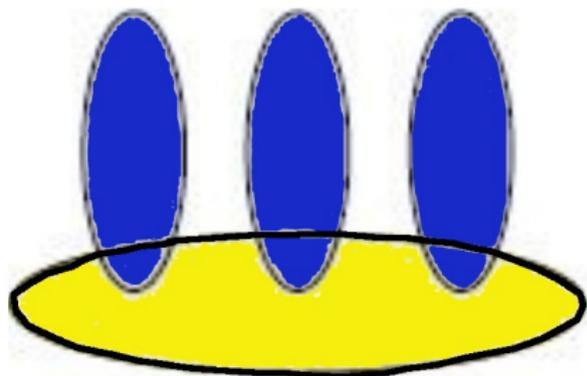
- The relaxation can nevertheless be solved in polynomial time, but its polytope is not *integral*, and the solution is usually fractional (but only with 1/2's).
- But the solution does give a *lower bound* for the length of the minimum Hamilton cycle. It is at least 2/3 of the optimum, and typically within 1% (for Euclidean TSP).

Cutting Planes

- To do better than the subtour bound, more inequalities must be added. Geometrically, this amounts to finding planes that cut off fractional solutions.
- There is a whole catalog of increasingly complicated structures that can give cutting planes.

Cutting Planes

One example of a cutting plane is a *comb inequality*:



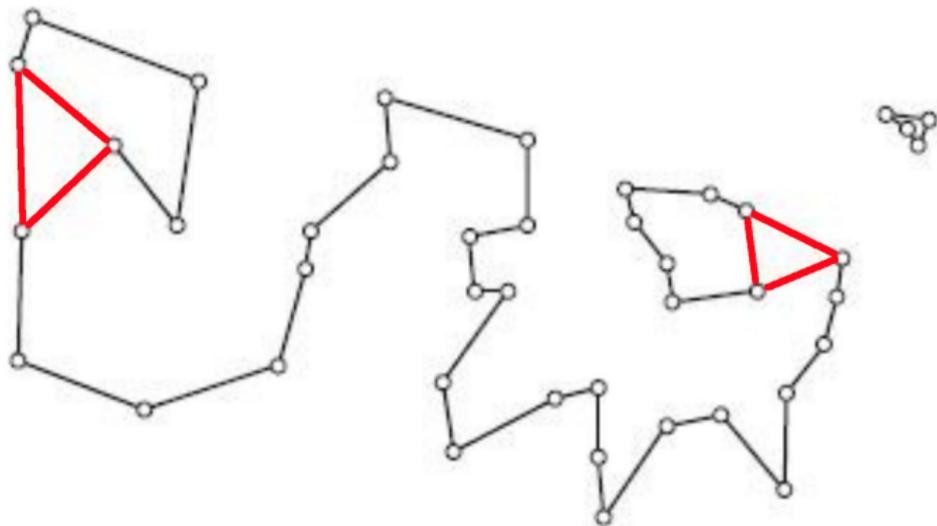
$$\sum_{e \in \delta(S_1)} x_e + \sum_{e \in \delta(S_2)} x_e + \sum_{e \in \delta(S_3)} x_e + \sum_{e \in \delta(S_4)} x_e \geq 10$$

It is like a sum of 4 subtour constraints, but stronger because it has 10 instead of 8 on the right.

(Each tooth of the comb has 2 edges going out, and the handle has 4.)

Example

Here is a fractional solution to the *Degree LP Relaxation* (with the degree constraints, but no subtour constraints), with edges with $x_e = 1/2$ in red:

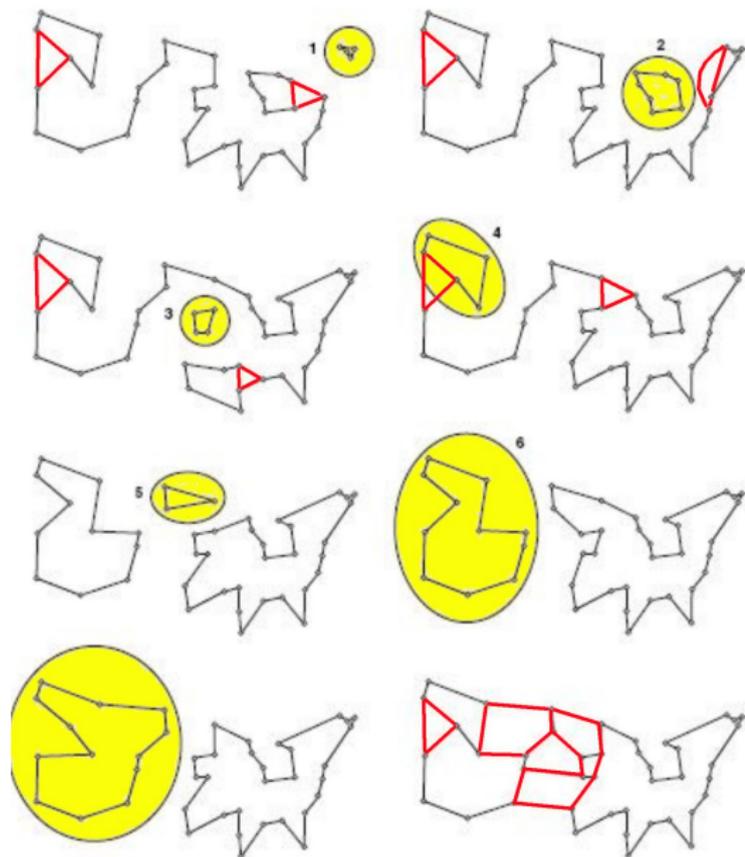


(It's a map of 42 cities in the US, used as one of the first TSP challenges in the 50s.)

Example

Here we introduce subtour constraints until we have a solution of the subtour LP relaxation. The yellow sets are the S .

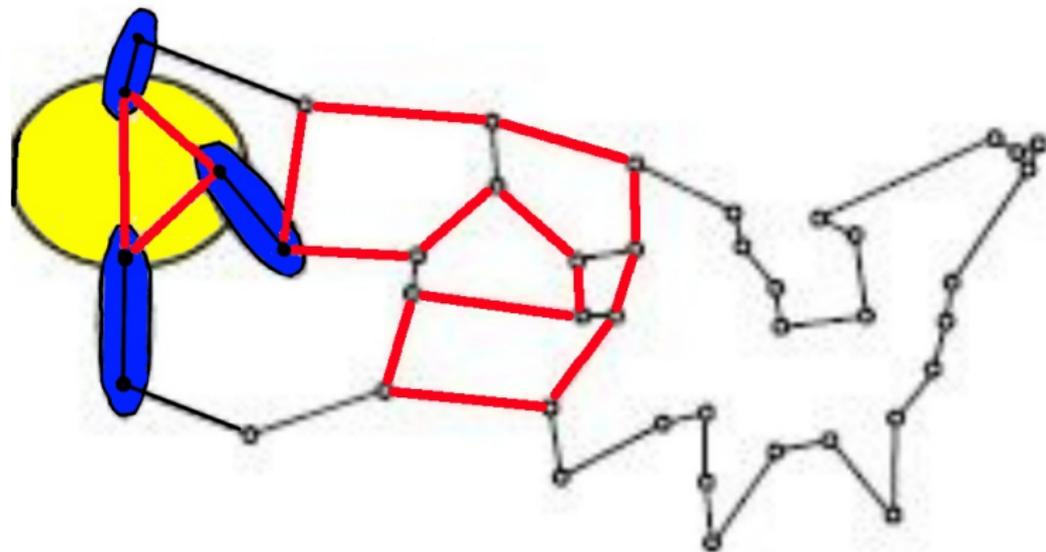
For instance, an island violates a subtour constraint, so adding that constraint to the LP excludes the current solution, and should give a “better” solution.



Example

After adding only 7 subtour constraints we obtained a solution that satisfies all subtour constraints. But it's fractional.

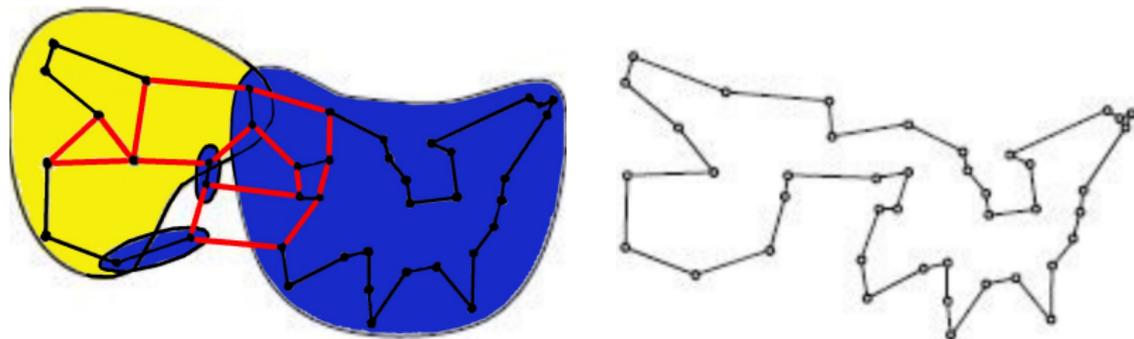
To get closer to a real cycle, we find (somehow) four sets in a comb configuration that violate the comb inequality:



The handle has 3 outgoing edges, giving 9 total.

Example

Adding the last comb inequality and solving the LP gives the following solution. It again has four sets violating a comb inequality (with one oversized handle):



On the right is a solution of the resulting LP; it is a cycle, hence it is a minimum Hamilton cycle.

So LP can sometimes find a solution, but more often it is used to get a lower bound.

Concorde

On the right is a part of the log of the Concorde app as it exactly solves an instance with 500 vertices.

- It starts by doing Chained Lin-Kernighan 10 times. This gives a cycle of length 59491.
- Then it looks for lower bounds with LP. From 57446 it goes to 58966, and continues like this until it finds the lower bound 59491, so the cycle found was optimal. (Other times it improves it by LP.)
- You can see that some of the inequalities it adds are subtours, some are combs, among other things.
- It took 112 seconds.

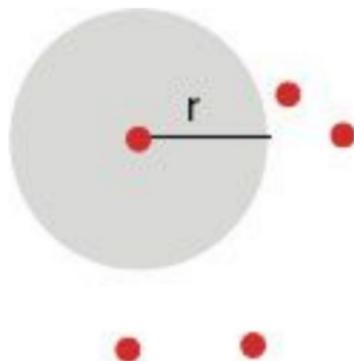
```
iPad
Method: Concorde Exact Solver
Problem: Random
Start 2012-12-18 17:02:03 +0000
Building tour with Chained Lin-Kernighan
LK Initial Run: 59608
LK Run 0: 59583
LK Run 1: 59738
LK Run 2: 59495
LK Run 3: 59491
LK Run 4: 59508
LK Run 5: 59585
LK Run 6: 59508
LK Run 7: 59558
LK Run 8: 59771
LK Run 9: 59589
Starting Tour: 59491
Found 14 connect cuts
Found 12 segment cuts
Found 20 fast blossoms
Found 8 G-H blossoms
Found 46 block combs
Found 5 exact subtours
Found 13 tighten_lp cuts
Found 2 CLOSE tighten_lp cuts
Found 7 double deckers
Found 6 CLOSE double deckers
LP Value 1: 57446.000000
Found 35 connect cuts
Found 10 segment cuts
Found 16 fast blossoms
Found 6 G-H blossoms
Found 26 block combs
Found 16 exact subtours
Found 34 tighten_lp cuts
Found 8 CLOSE tighten_lp cuts
Found 40 double deckers
Found 5 CLOSE double deckers
Found 26 handling cuts
LP Value 2: 58966.000000
Found 3 connect cuts
Found 24 segment cuts
Found 10 fast blossoms
Found 10 G-H blossoms
Found 73 block combs
Found 1 exact subtours
Found 36 tighten_lp cuts
Found 17 CLOSE tighten_lp cuts
Found 53 double deckers
Found 15 CLOSE double deckers
Found 33 handling cuts
Found 22 teathed combs
LP Value 3: 59237.517857
```

Dual of the degree LP

The degree LP is the LP relaxation without the subtour constraints; we'll consider its dual program. The dual constraints are

$$r_u + r_v \leq w_{uv} \quad \text{for } uv \in E(G).$$

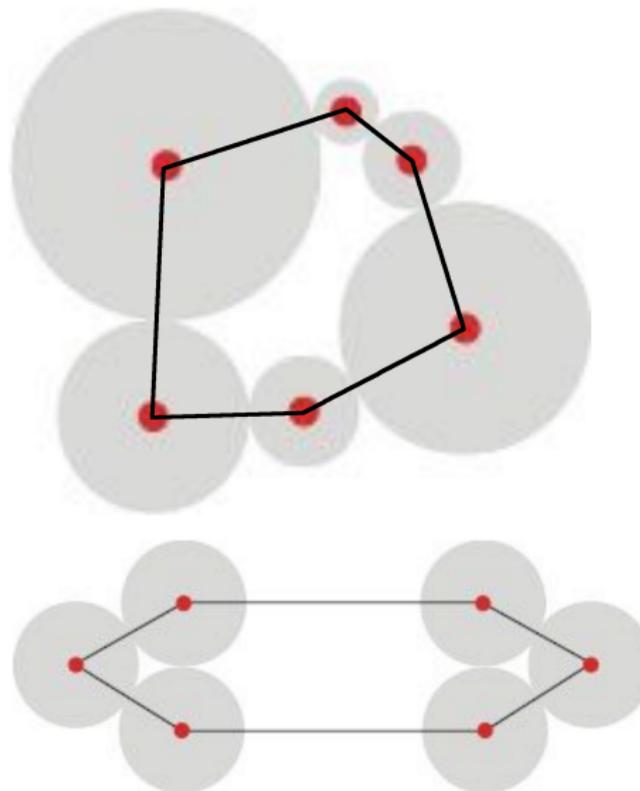
We can interpret each r_v as the width of a circle around v . Then the constraint says that the circles may not overlap.



Zones

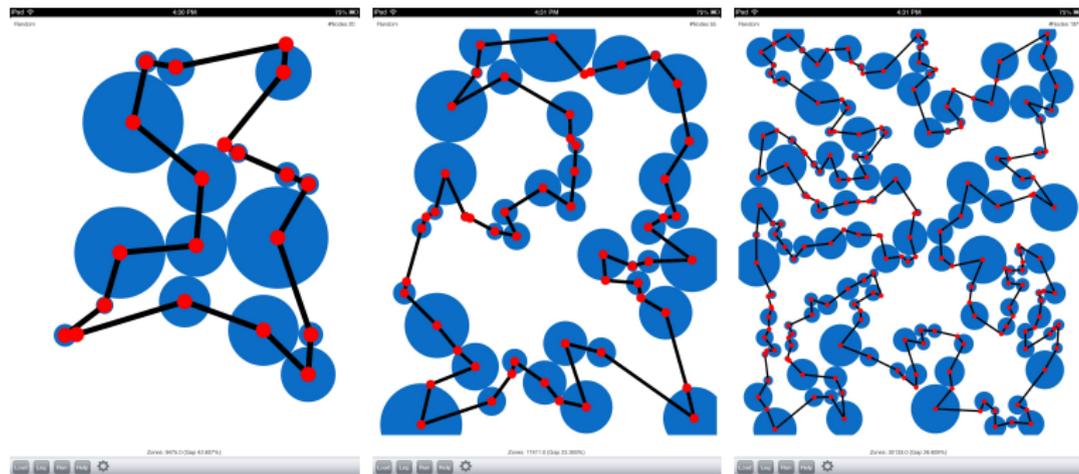
If we're lucky, the solution of this LP will look like the first one on the right, where there is a Hamilton cycle that stays within the circles. By the duality theorem, that means it is minimum.

But in the second one, we see that this does not always happen.



Zones

Here are some examples.



These have approximation factors 1.42, 1.23, and 1.26. But there is no guaranteed approximation factor.

Dual of the degree LP with subtour constraints added

Now suppose that in the primal degree LP, we add a subtour constraint for some set $S \subset V(G)$. That will change some of the dual constraints. Specifically, those for $e = uv$ with $e \in \delta(S)$ become

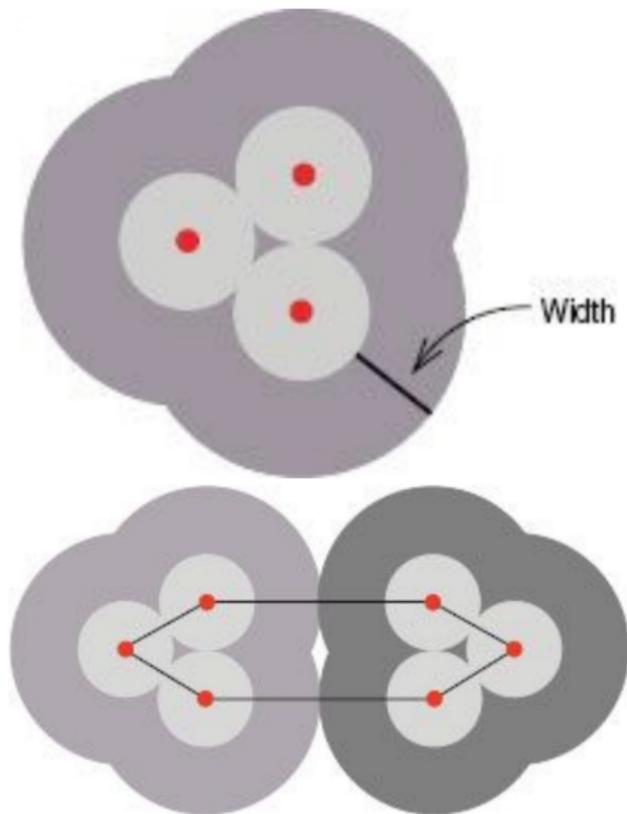
$$r_u + r_v + y_S \leq w_e.$$

This amounts to putting a *moat* around the zones of the vertices in S , i.e. a strip of a fixed width placed around those circles.

Moats

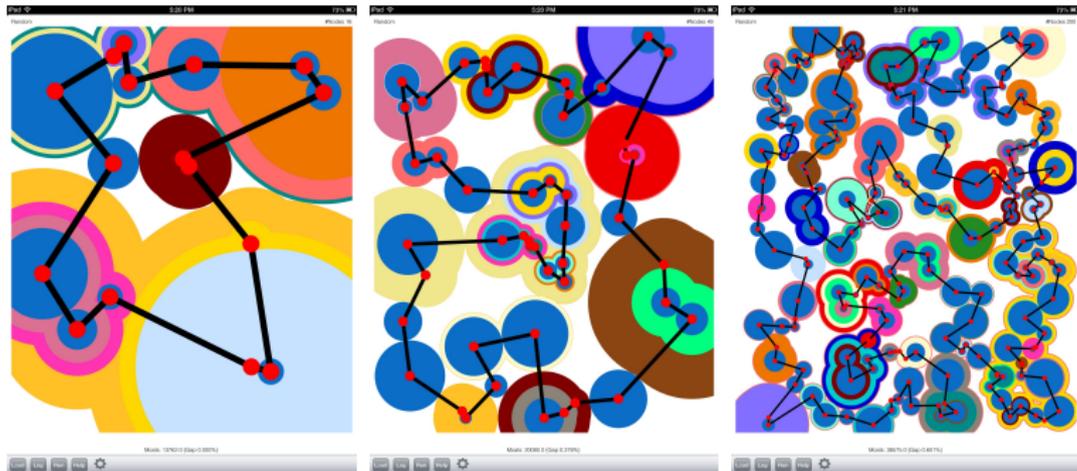
Here we've added a subtour constraint for S being three vertices.

Here you see that two moats solve the problem we saw earlier.

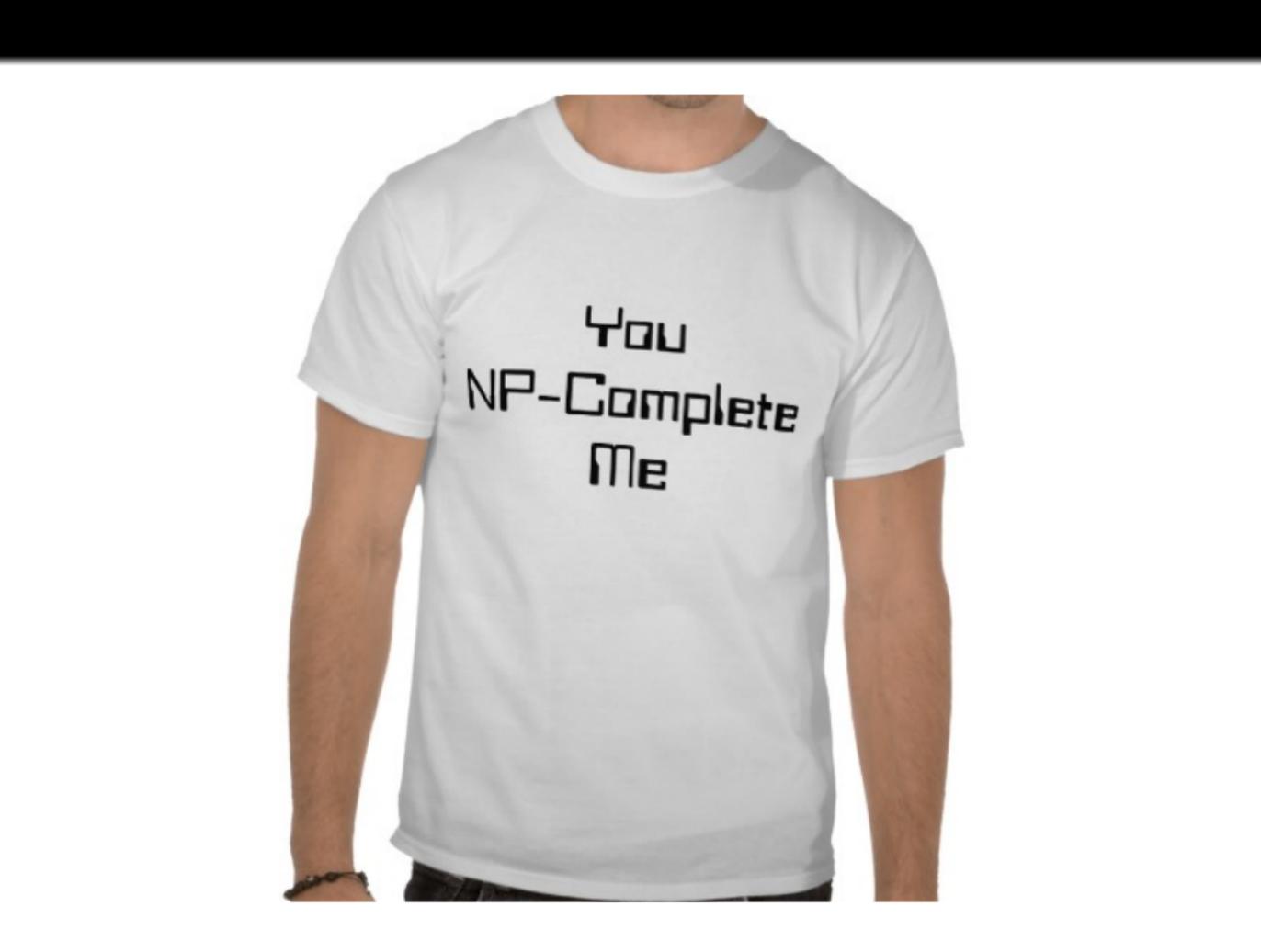


Moats

Here are some lovely examples.



These have approximation factors 1.00, 1.003, and 1.006. Sadly, the dual solution only gives the zones and moats, but not the path (in the pictures Concorde computed that separately). So this only gives an upper bound for the minimum Hamilton cycle.

A person is shown from the chest up, wearing a white t-shirt. The t-shirt has the text "You NP-Complete Me" printed on it in a black, monospaced font. The text is arranged in four lines: "You" on the first line, "NP-Complete" on the second line, and "Me" on the third line. The fourth line is empty. The person is also wearing a black beaded bracelet on their left wrist.

You
NP-Complete
Me

