15-210

Parallel and Sequential Algorithms and Data Structures

Lecture 4

Divide-and-Conquer Continued

SYNOPSIS

- The Euclidian Travelling Salesperson Problem
- Divide-and-Conquer Heuristic Algorithm
- Analysis of Costs

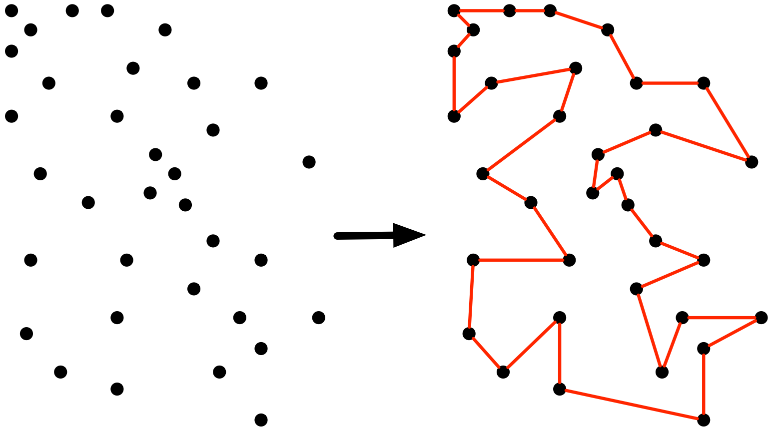
THE EUCLIDIAN TSP

- Given a set of points in a *n*-dimensional Euclidian space.
  - What is a Euclidian space?

- Find the shortest Hamiltonian cycle.
  - What is a Hamiltonian cycle?

- We get a planar Euclidian Traveling Salesperson Problem when the points are in 2-dimensional space.

The Planar TSP



SYNOPSIS

- The Euclidian Travelling Salesperson Problem
- Divide-and-Conquer Heuristic Algorithm
- Analysis of Costs

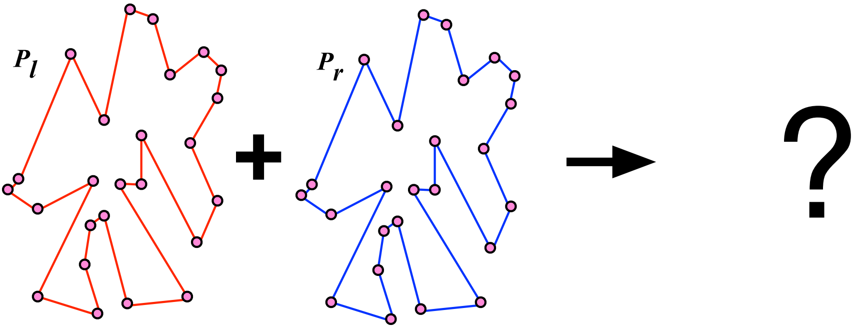
A Divide-and-Conquer Heuristic

- What is a heuristic?

- Approximation algorithm
  - Resulting tour length is guaranteed to be close to the actual minimum tour length
  - If you spend enough work (but polynomial).

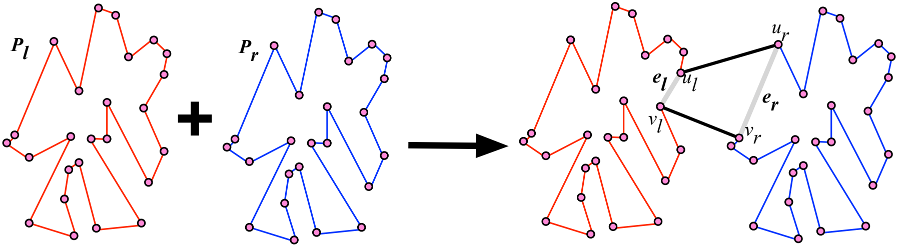
- The Divide-and-Conquer does work both before and after the recursive calls.

A Divide-and-Conquer Heuristic



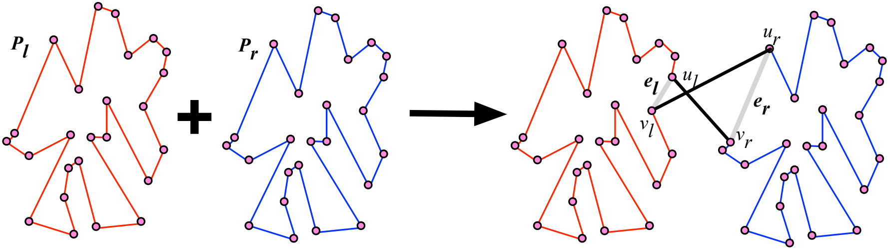
- Assume P_ℓ and P_r have tour lengths T_ℓ and T_r.
- Tour length for the combination?

A Divide-and-Conquer Heuristic



$$\color{red}{T_\ell} + \color{blue}{T_r} + \underbrace{\|u_\ell - u_r\| + \|v_\ell - v_r\|}_{\text{Add these}} - \underbrace{\|u_\ell - v_\ell\| - \|u_r - v_r\|}_{\text{Subtract these}}$$

A Divide-and-Conquer Heuristic



$$\color{red}{T_\ell} + \color{blue}{T_r} + \underbrace{\|u_\ell - v_r\| + \|v_\ell - u_r\|}_{\text{Add these}} - \underbrace{\color{gray}{\|u_\ell - v_\ell\| - \|u_r - v_r\|}}_{\text{Subtract these}}$$

A Divide-and-Conquer Heuristic

- Try all pairs of edges e_ℓ from P_ℓ and e_r from P_r
  - How many pairs are there?
- For each pair of edges, find the smallest increase.
- Then combine the small tours into a large tour.

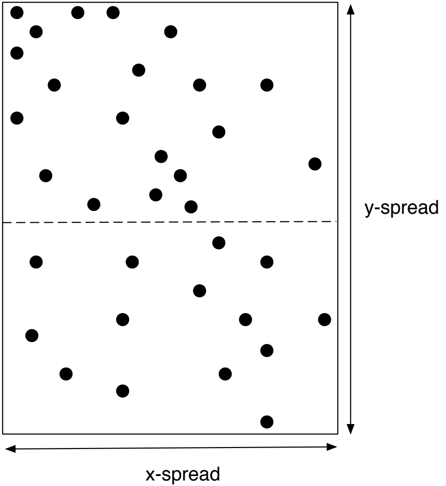
A Divide-and-Conquer Heuristic

```
fun eTSP(P) =
  case (|P|)
    of 0, 1 ⇒ raise TooSmall
     | 2 ⇒ {(P[0], P[1]), (P[1], P[0])}
     | n ⇒ let
              (P_ℓ, P_r) = splitLongestDim(P)
              (L, R) = ( eTSP(P_ℓ) || eTSP(P_r) )
              (c, (e'_ℓ, e'_r)) =
              minval <_#1 {(swapCost(e_ℓ, e_r), (e_ℓ, e_r)) :
                                e_ℓ ∈ L, e_r ∈ R}
            in
              swapEdges(append(L, R), e'_ℓ, e'_r)
      end
```

A Divide-and-Conquer Heuristic

```
fun eTSP(P) =
  case (|P|)
    of 0, 1 ⇒ raise TooSmall
     | 2 ⇒ {(P[0], P[1]), (P[1], P[0])}
     | n ⇒ let
               (Pℓ, Pr) = splitLongestDim(P)
               (L, R) = (eTSP(Pℓ) || eTSP(Pr))
               (c, (e′ℓ, e′r)) =
               minval <#1 {(swapCost(eℓ, er), (eℓ, er)) :
                                 eℓ ∈ L, er ∈ R}
           in
               swapEdges(append(L, R), e′ℓ, e′r)
    end
```

Splitting the Points



- Split at the median along the longer spread dimension.

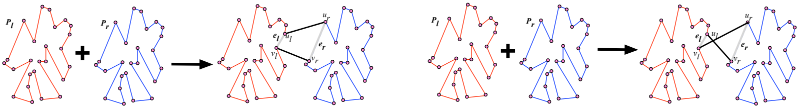
Swap Cost

- Given $e_\ell = (u_\ell, v_\ell) \in L$ and $e_r = (u_r, v_r) \in R$

$$\text{swapCost}((u_\ell, v_\ell), (u_r, v_r)) = \text{Cost Added} - \text{Cost Removed}$$

$$\begin{aligned}\text{Cost Added} = \min(&\|u_\ell - u_r\| + \|v_\ell - v_r\|\,, \\ &\|u_\ell - v_r\| + \|v_\ell - u_r\|)\end{aligned}$$

$$\text{Cost Removed} = \|u_\ell - v_\ell\| + \|u_r - v_r\|$$



Swapping Edges

- `swapEdges(append(L,R),e'`$_\ell$`,e'`$_r$`)`
- Appends the Tour edge lists from subproblems
- Then removes and adds appropriate edges.

SYNOPSIS

- The Euclidian Travelling Salesperson Problem
- Divide-and-Conquer Heuristic Algorithm
- Analysis of Costs

```
fun eTSP(P) =
  case (|P|)
  of 0,1 ⇒ raise TooSmall
   | 2 ⇒ {(P[0], P[1]), (P[1], P[0])}
   | n ⇒ let
            (P_ℓ, P_r) = splitLongestDim(P)          O(n) work  O(log n) span  (Why?)
            (L, R) = ( eTSP(P_ℓ) || eTSP(P_r) )       2W(n/2) work  S(n/2) span
            (c, (e'_ℓ, e'_r)) =
            minval <_#1 {(swapCost(e_ℓ, e_r), (e_ℓ, e_r)) :
                              e_ℓ ∈ L, e_r ∈ R}        O(n^2) work  O(log n) span  (Why?)
           in
             swapEdges(append(L, R), e'_ℓ, e'_r)       O(log n) span  (Why?)
        end
```

Cost Analysis

$$W(n) = 2W(n/2) + O(n^2)$$

$$S(n) = S(n/2) + O(\log n)$$

$$S(n) \in O(\log^2 n)$$

Cost Analysis

- Solve (directly)

$$W(n) = 2W(n/2) + k \cdot n^{1+\varepsilon}$$

for constant $\varepsilon > 0$.

  - Depth is $\log_2 n$ (Is this technically right?)
  - At level i, we have 2^i nodes each costing $k \cdot (n/2^i)^{1+\varepsilon}$

$$\begin{aligned}
W(n) &= \sum_{i=0}^{\log n} k \cdot 2^i \cdot \left(\frac{n}{2^i}\right)^{1+\varepsilon} \\
&= k \cdot n^{1+\varepsilon} \cdot \sum_{i=0}^{\log n} 2^{-i \cdot \varepsilon} \\
&\leq k \cdot n^{1+\varepsilon} \cdot \sum_{i=0}^{\infty} 2^{-i \cdot \varepsilon} \\
W(n) &\in O(n^{1+\varepsilon})
\end{aligned}$$

(Why?)

SUMMARY

- Euclidian Traveling Salesperson Problem
  - Divide-and-Conquer Heuristic
  - Processing before and after the subproblem solutions.
- Cost Analysis