Section 20.4: The 2-OPT Heuristic for the TSP

The Traveling Salesman Problem (TSP)

Input: complete undirected graph $G = (V, E)$, real-valued edge costs c_e.

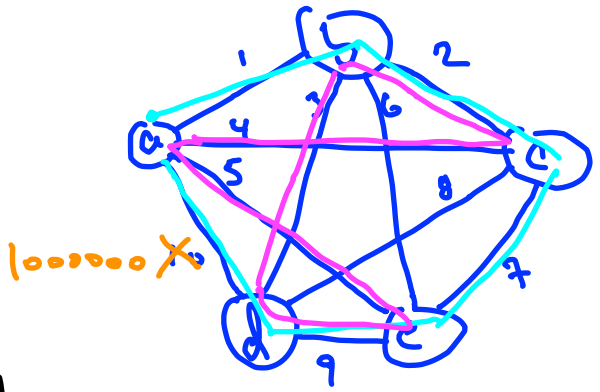
Output: A tour T (i.e., cycle visiting each vertex exactly once) with minimum-possible sum of edge costs $\sum_{e \in T} c_e$.

Fact: NP-hard. (as we'll eventually prove)

Quiz #1

Nearest neighbor heuristic

1. Start at arbitrary vertex a.
2. Repeat until all vertices visited:
   - if current vertex is v, travel to closest unvisited vertex (minimizing c_{vw})
3. Return to starting vertex



Question: What is the total cost of an optimal tour and the total cost of the heuristic's tour?

(a) 23 and 29
(b) 24 and 29
(c) 25 and 29
(d) 24 and 30

Quiz #2

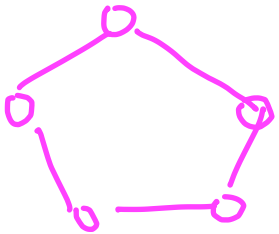
For a TSP instance with n vertices, what is the maximum number of edges that can be shared by two distinct tours?

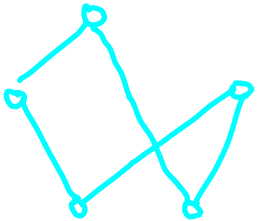
(a) $\log_2 n$

(b) $n/2$

(c) $n-2$

(d) $n-1$

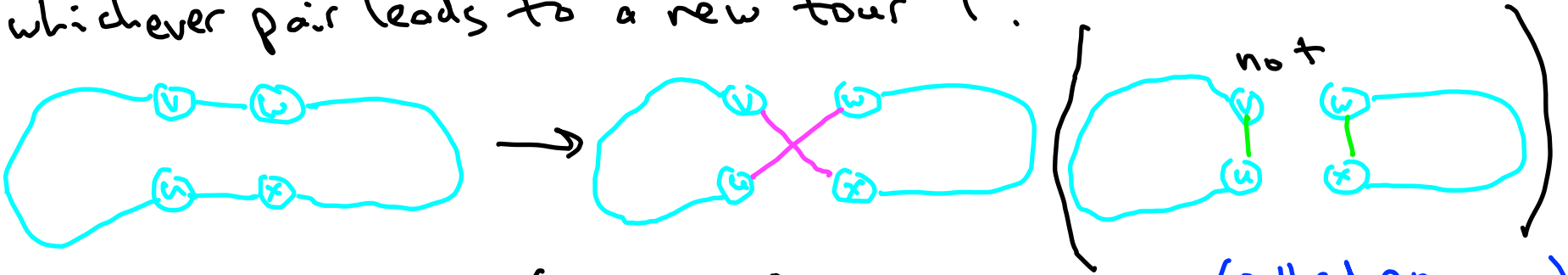




2-Changes

2-Change: (1) Given a tour T, remove two edges $(v,w), (u,x)$ of T that share no endpoints,

(2) Add either the edges (v,x) & (u,w) or the edges (v,u) & (w,x), whichever pair leads to a new tour T'.



Decrease in cost: $(c_{vw} + c_{ux}) - (c_{uw} + c_{vx})$

edges removed — edges added

(called an improving 2-change if decrease > 0)

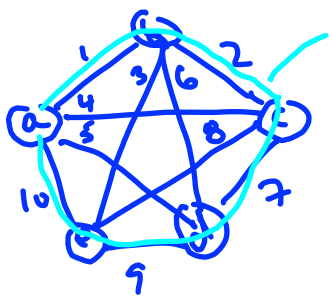
The 2-OPT Heuristic

(1) $T :=$ initial tour (e.g. output of NearestNeighbor heuristic)

(2) while there is an improving 2-change ($^{\text{edges } (v,w)}_{\text{and } (u,x)}$)

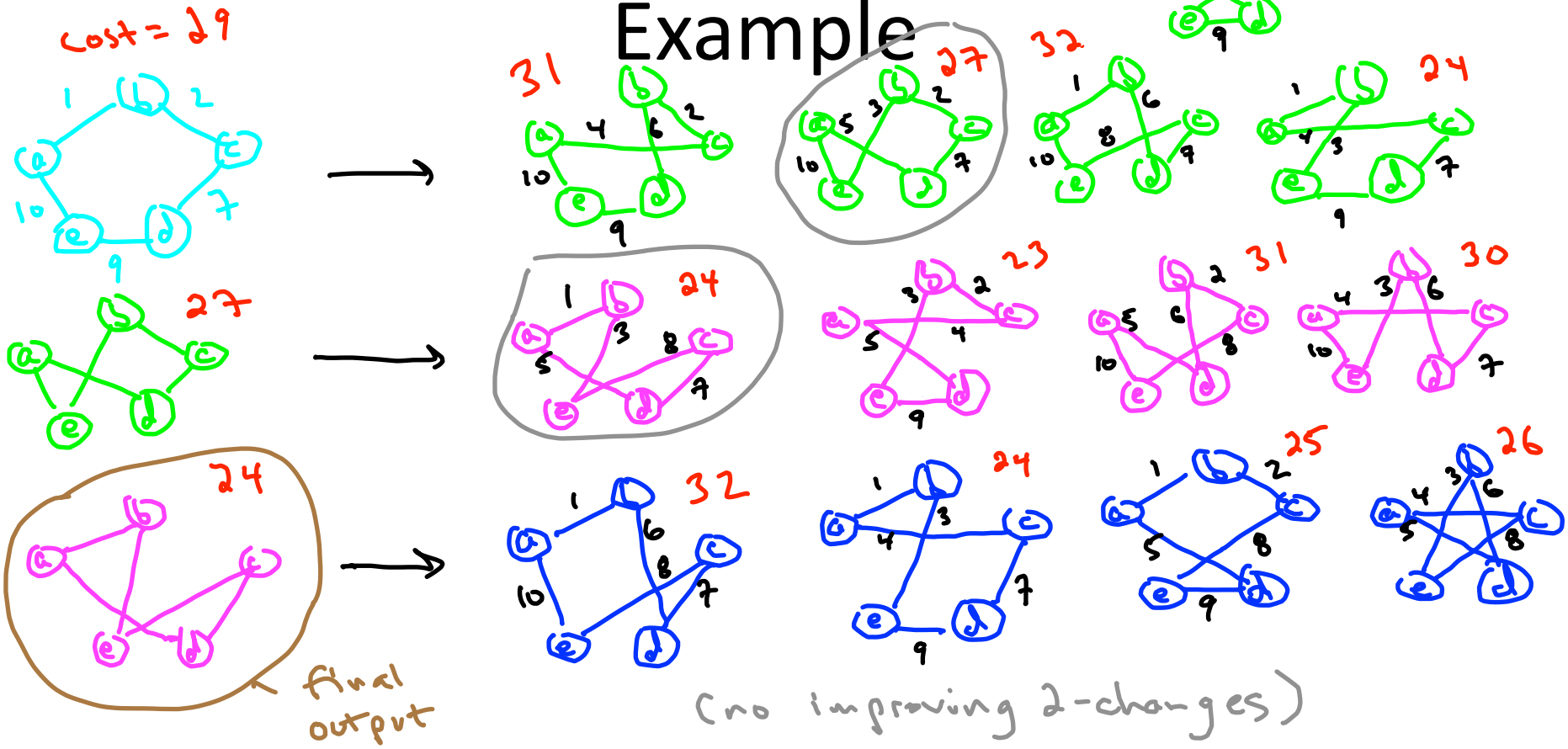
- $T := 2\text{Change}(T, (v,w), (u,x))$

(3) return T



greedy tour

Example


cost = 29
1
2
10
7
9
31
4
6
2
10
9
27
1
3
2
5
3
10
7
32
1
6
8
10
9
25
1
2
5
8
9
24
1
3
4
7
9
27
24
1
3
8
5
7
23
3
2
5
4
9
31
2
6
5
10
8
30
4
3
6
10
7
24
final output
32
1
6
8
10
7
24
1
3
4
7
9
25
1
2
5
8
9
26
3
6
4
5
8
(no improving 2-changes)

2-OPT: Running Time

- guaranteed to halt within exponential # of iterations
  (each tour better than the last ⇒ no repeats)
- each iteration runs in $O(n^2)$ time (n = # of vertices)

Bad news: in pathological examples, # of iterations can be exponential (in n).

Good news: (1) empirically, usually halts after subquadratic # of iterations.

(2) "anytime algorithm" – can interrupt after prescribed amount of time (returns best tour found so far)

2-OPT: Solution Quality

Bad news: No approximate correctness guarantee.

Good news: Empirically, typically returns tour with total cost close to the minimum possible (e.g., $\leq 3\%$ or $\leq 20\%$, depending on the application).