

Some of these slides have been borrowed from Dr. Paul Lewis, Dr. Joe Felsenstein. Thanks!

Paul has many great tools for teaching phylogenetics at his web site:

<http://hydrodictyon.eeb.uconn.edu/people/plewis>

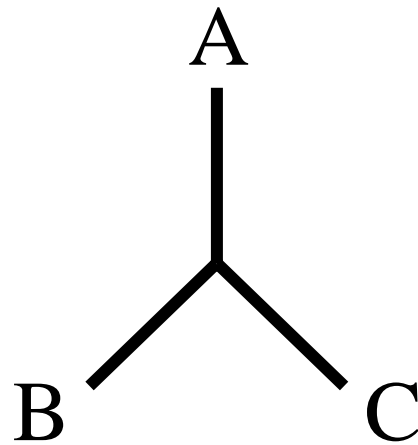
# Tree Searching

---

Parsimony and ML give us ways to deciding whether one tree is fits our data better than another tree, but . . .

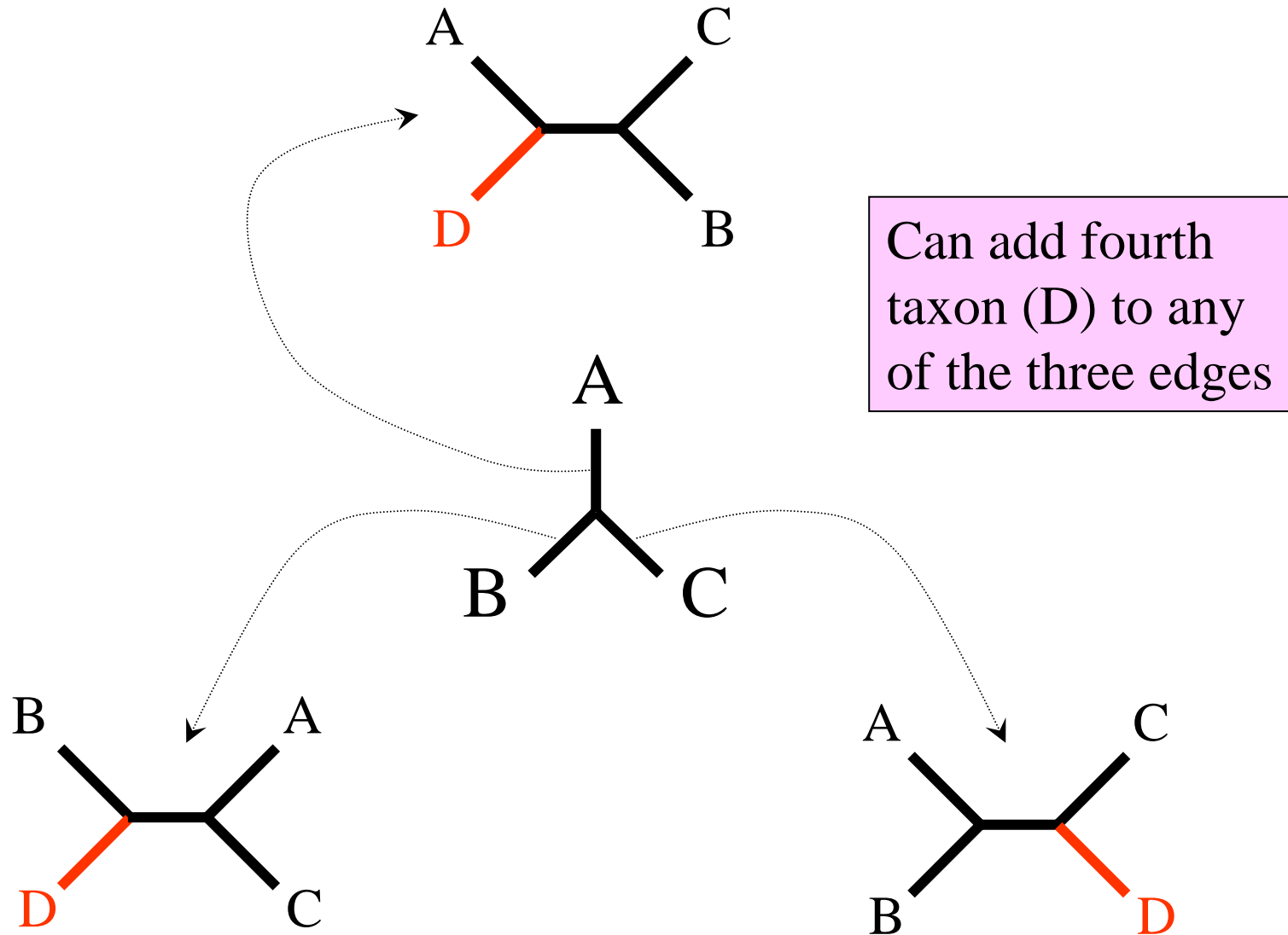
How do we find the best tree?  
(or one that is good enough)

# Exhaustive Enumeration



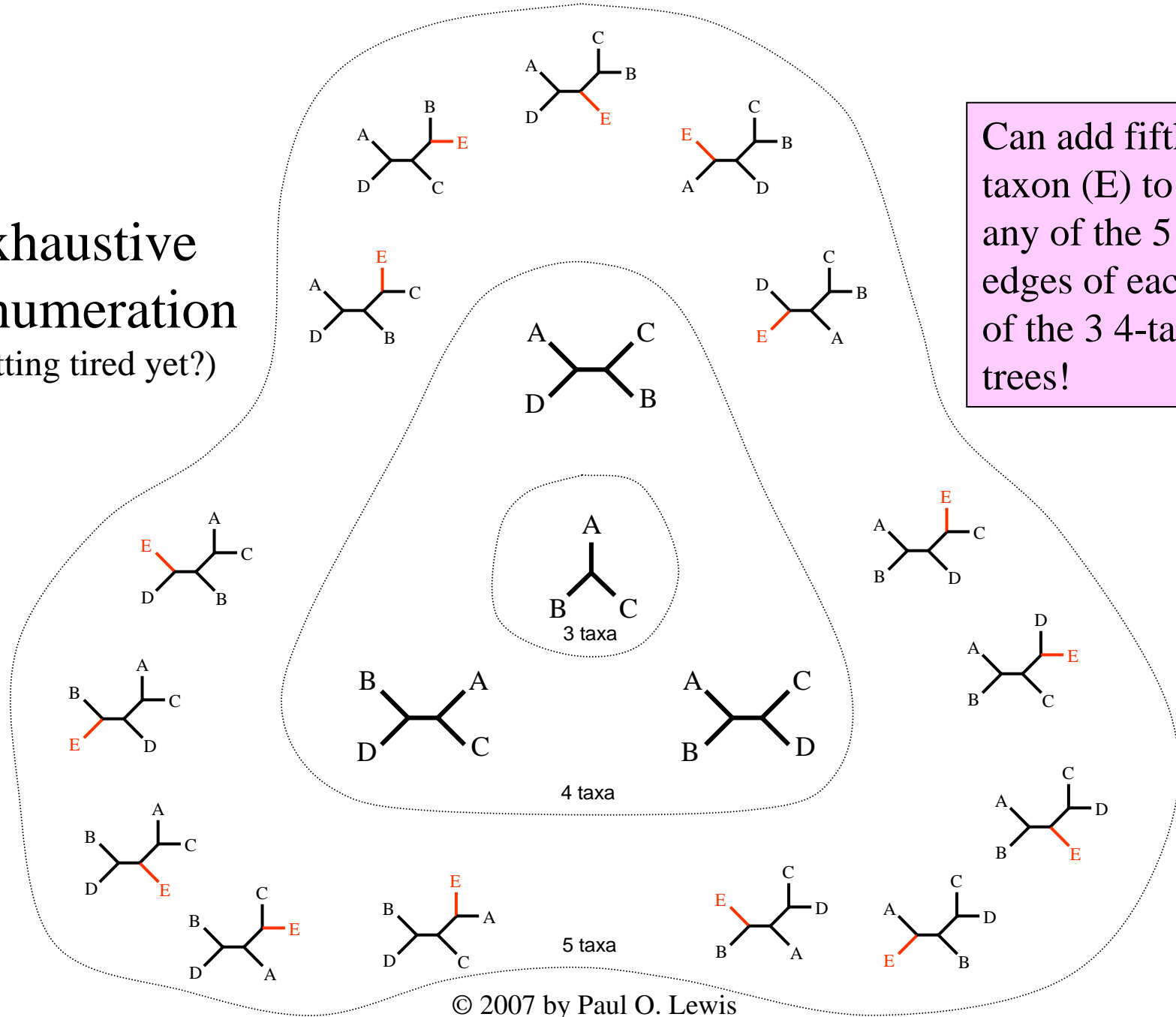
With the first three taxa, create the trivial unrooted tree

# Exhaustive Enumeration...



# Exhaustive Enumeration (getting tired yet?)

Can add fifth  
taxon (E) to  
any of the 5  
edges of each  
of the 3 4-taxon  
trees!



Tips	Number of unrooted (binary) trees	
4	3	
5	15	
6	105	
7	945	
8	10,395	
9	135,135	
10	2,027,025	
11	34,459,425	
12	654,729,075	
13	13,749,310,575	
14	316,234,143,225	
15	7,905,853,580,625	
16	213,458,046,676,875	
17	6,190,283,353,629,375	
18	191,898,783,962,510,625	
19	6,332,659,870,762,850,625	
20	22,164,309,5476,699,771,875	
21	8,200,794,532,637,891,559,375	
22	319,830,986,772,877,770,815,625	
23	13,113,070,457,687,988,603,440,625	> 21 moles of trees
24	563,862,029,680,583,509,947,946,875	

For  $N$  taxa:

$$\# \text{ unrooted, binary trees} = \prod_{i=3}^{N-1} (2i - 3)$$

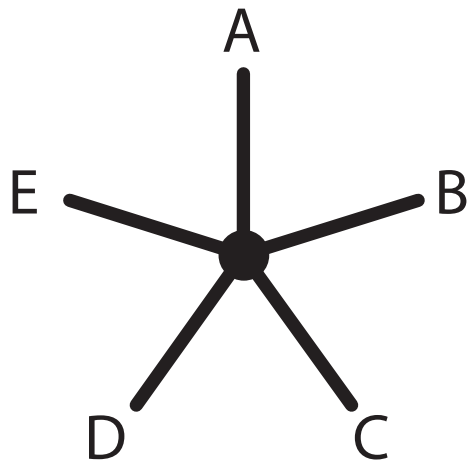
$$= \prod_{i=4}^N (2i - 5)$$

$$\# \text{ rooted, binary trees} = \prod_{i=3}^N (2i - 3)$$

$$= (2N - 3)(\# \text{ unrooted, binary trees})$$

# Star decomposition

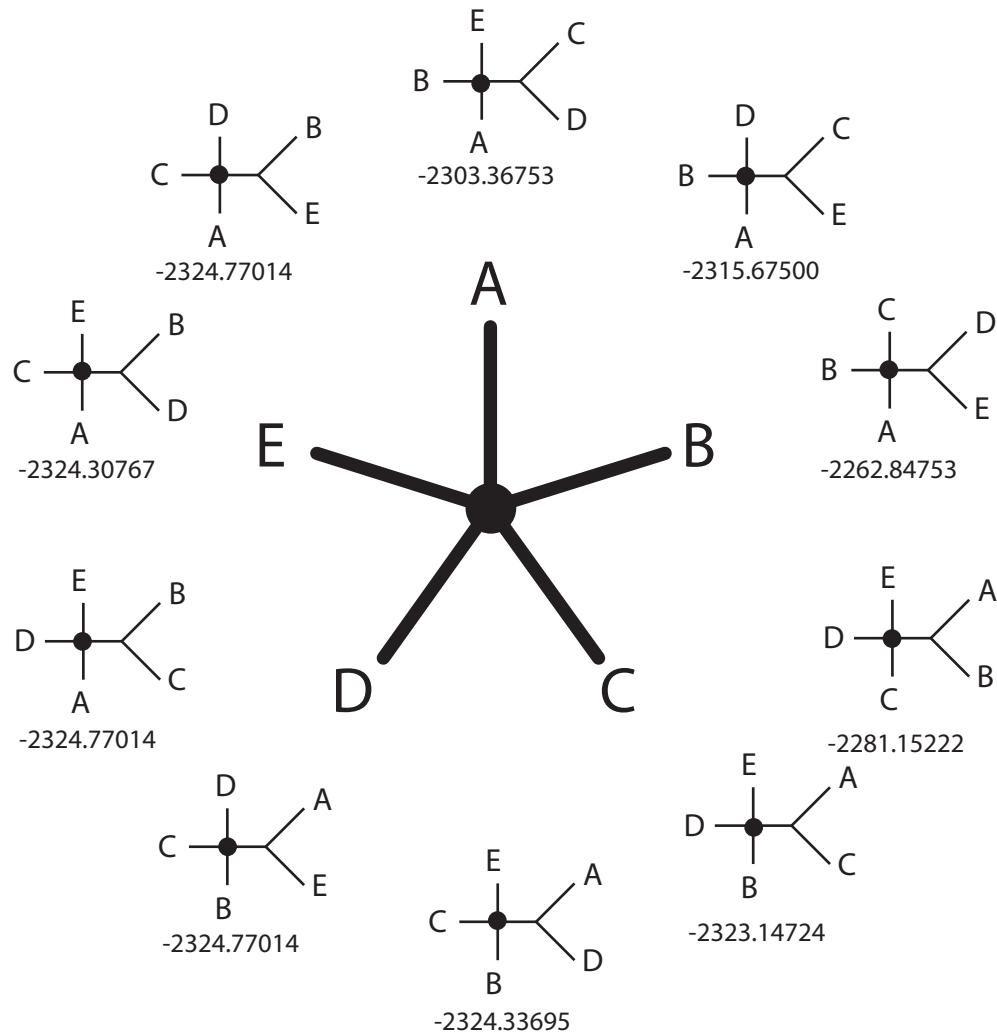
---



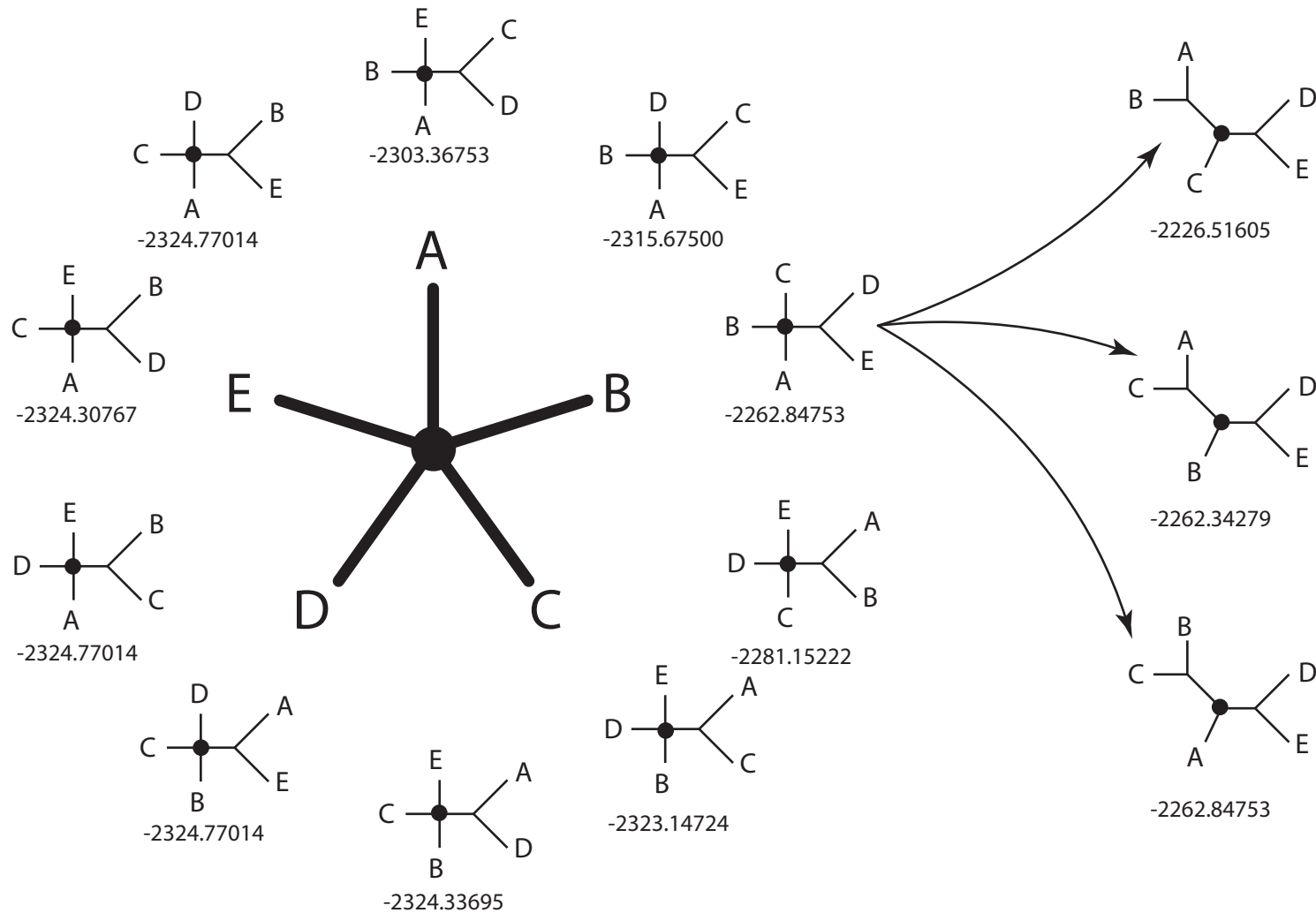


# Star decomposition

---



# Star decomposition



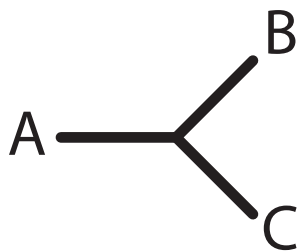
## Star decomposition

---

- Very greedy. Once a pair of species are joined, they will not be separated.
- Neighbor-joining is star decomposition under the balanced minimum evolution criterion

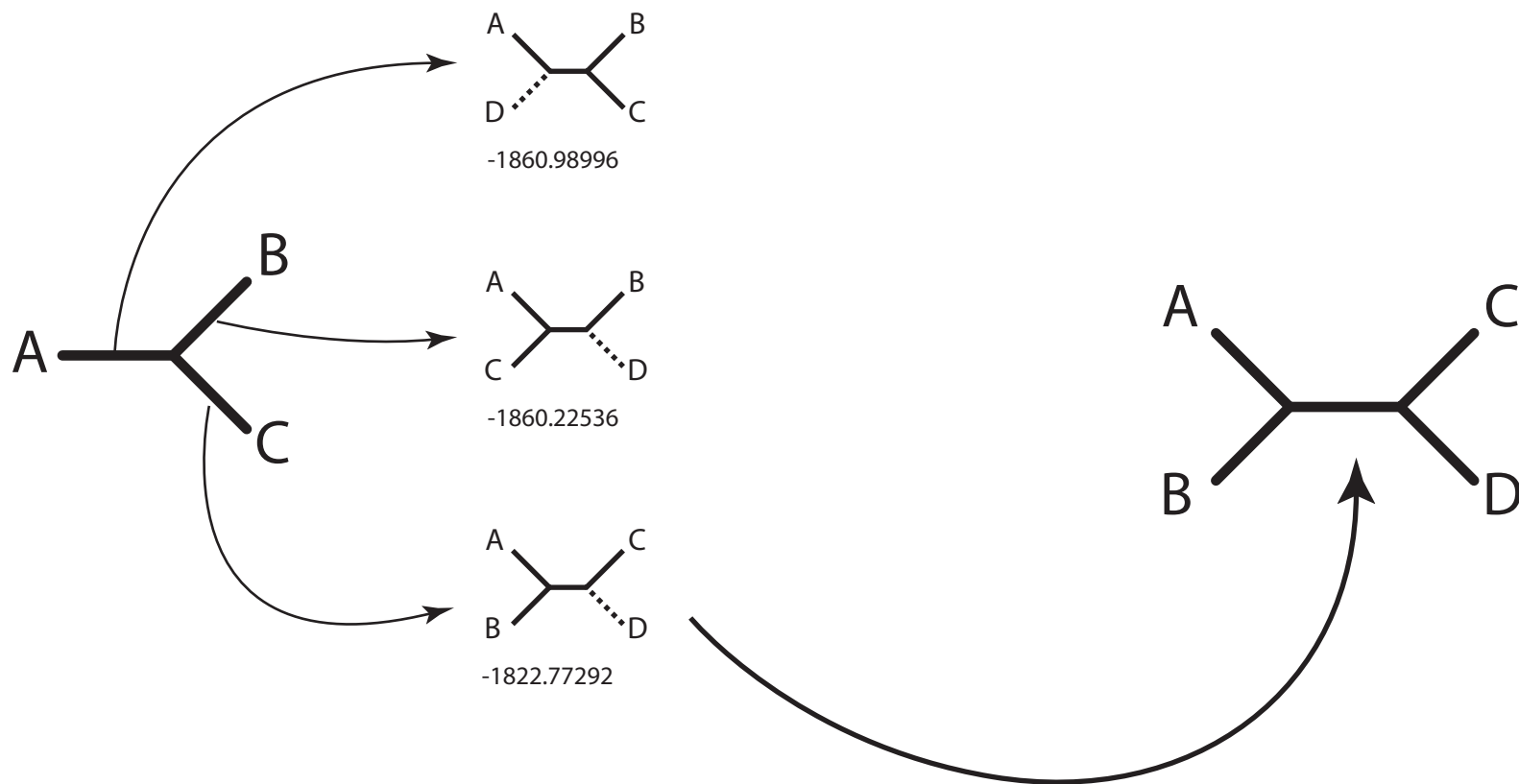
## Stepwise addition

---



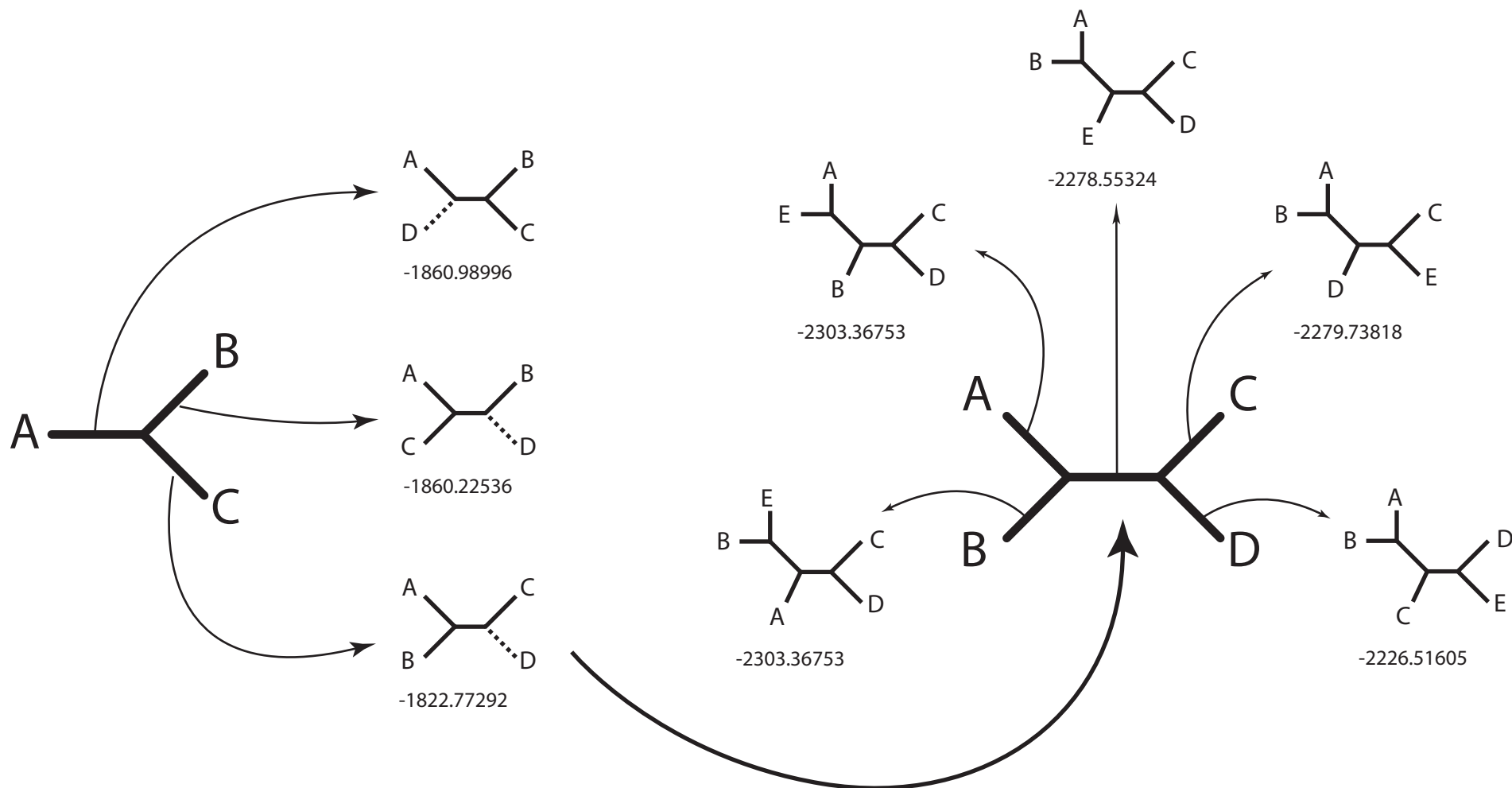
# Stepwise addition

---



# Stepwise addition

---



# Heuristic hill-climbing

---

Problem:

- Stepwise addition and star decomp. may not return the best tree
- exhaustive searching is infeasible.

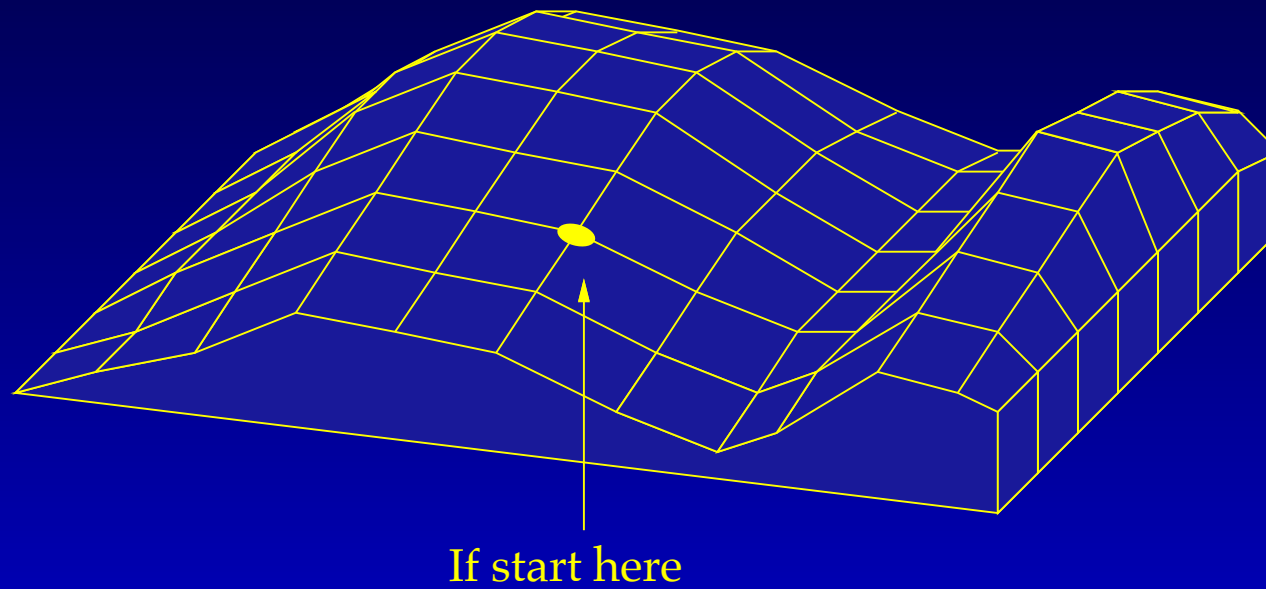
hill-climbing:

1. Start with a tree
2. Score the tree
3. Consider a new tree within the neighborhood of the current tree:
  - (a) Score the new tree.
  - (b) If the new tree has a better score, use it as the “current tree”
  - (c) Stop if there are no other trees within the neighborhood to consider.

These are **not** guaranteed to find even one of the optimal trees.

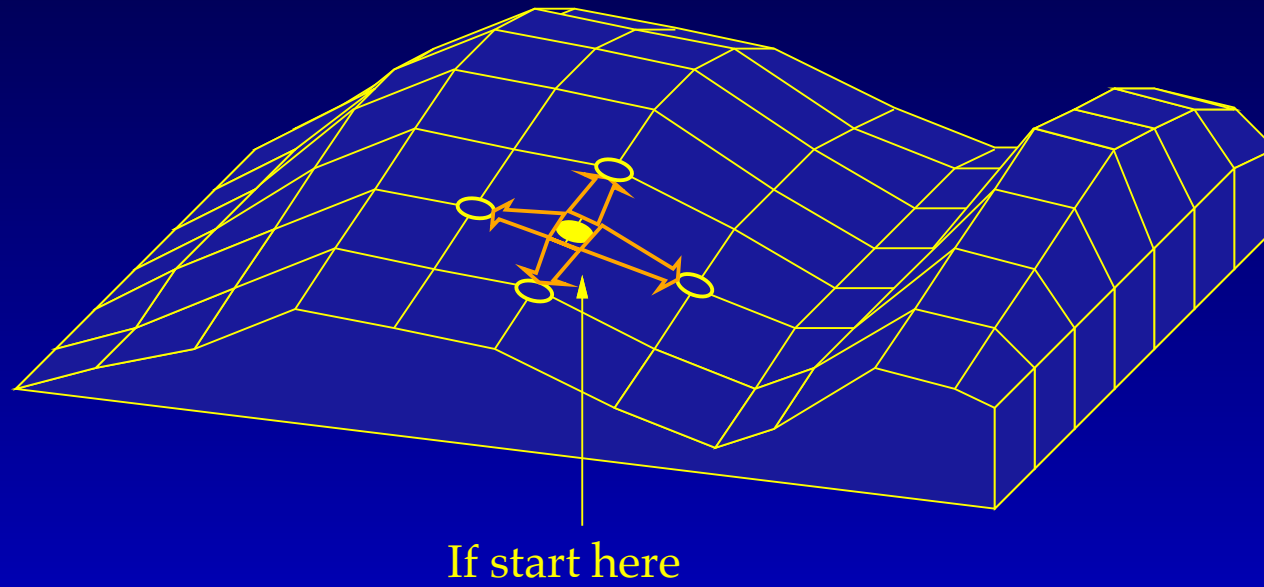
The most common way to explore the neighborhood of a tree is to swap the branches of the tree to construct similar trees.

# Greedy search for a maximum

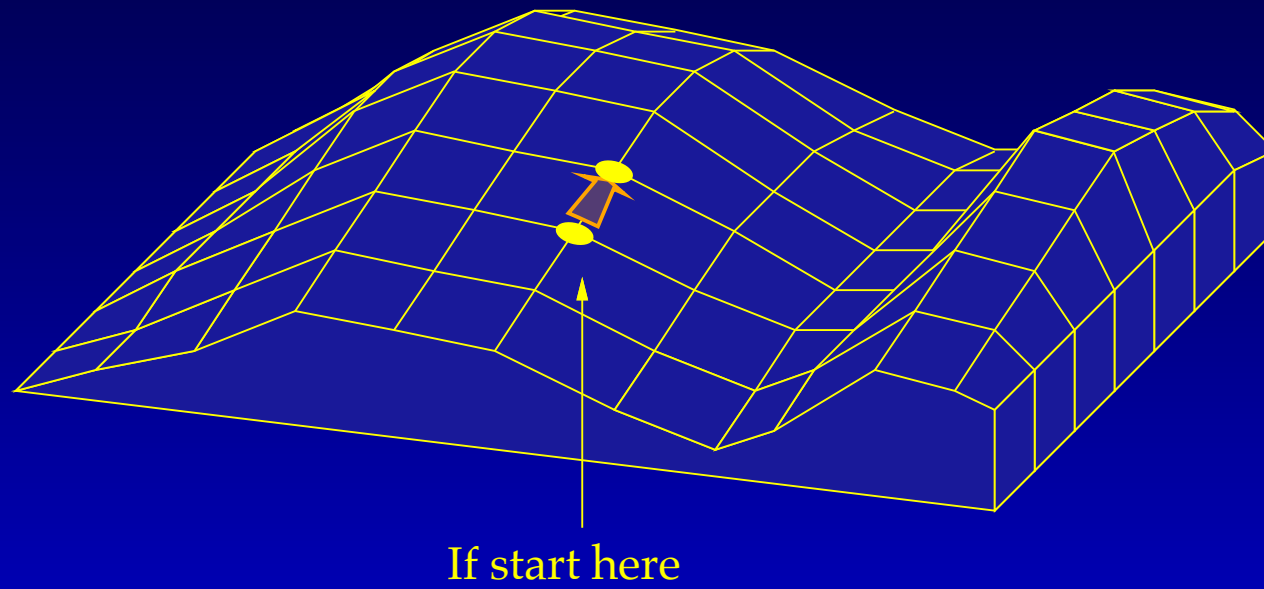




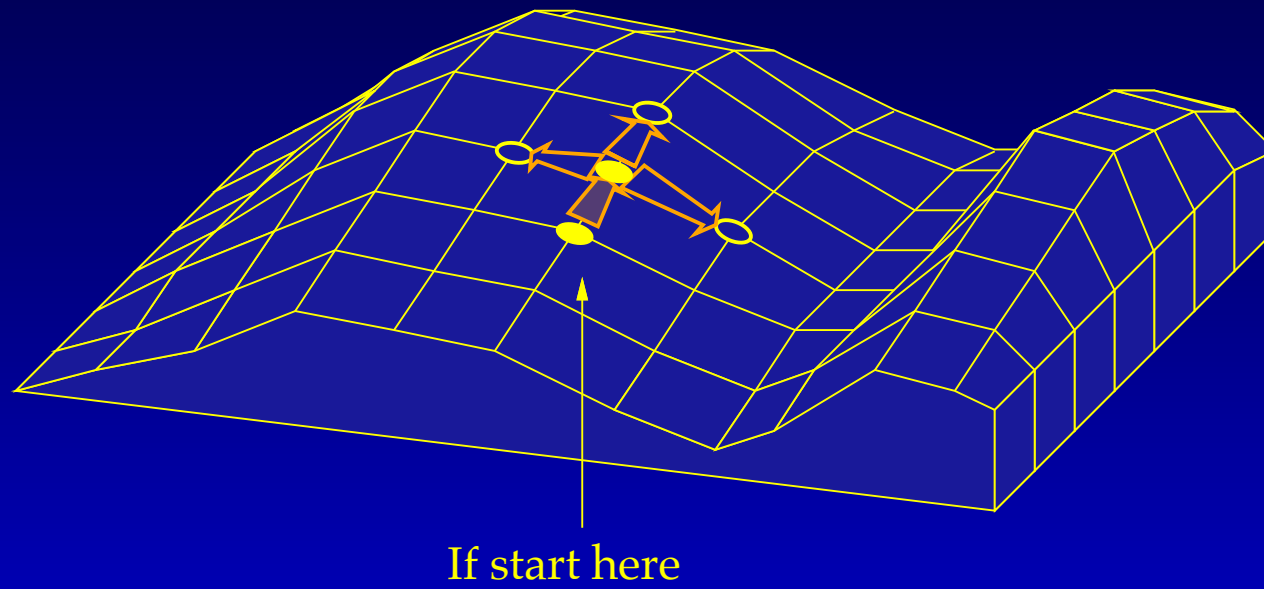
# Greedy search for a maximum



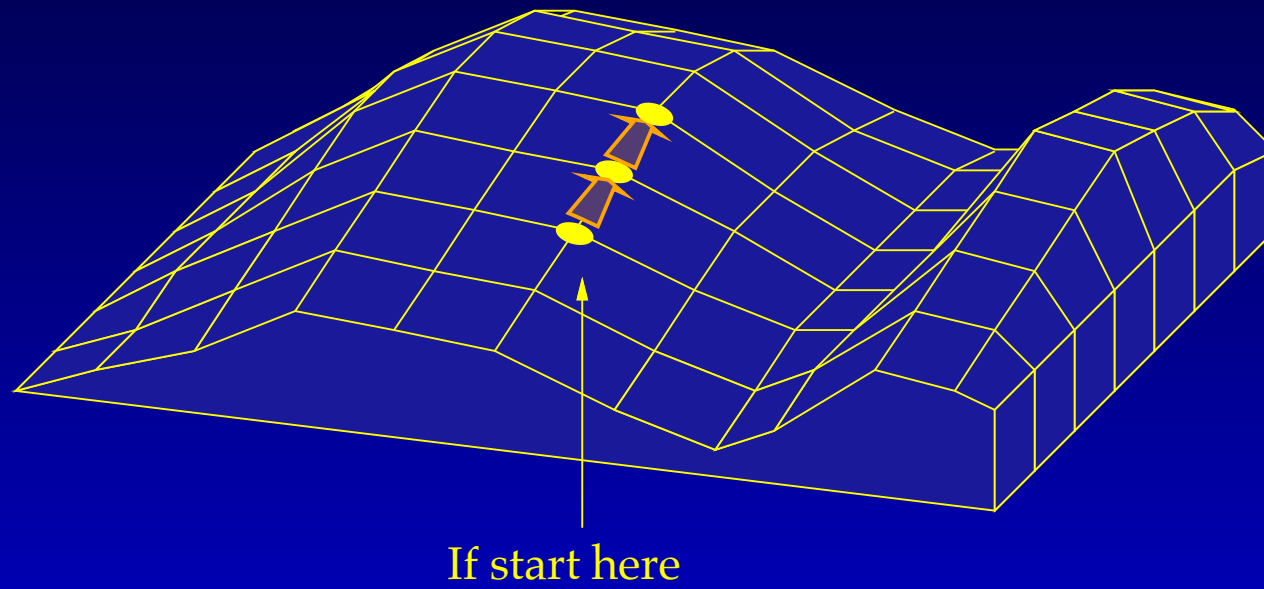
# Greedy search for a maximum



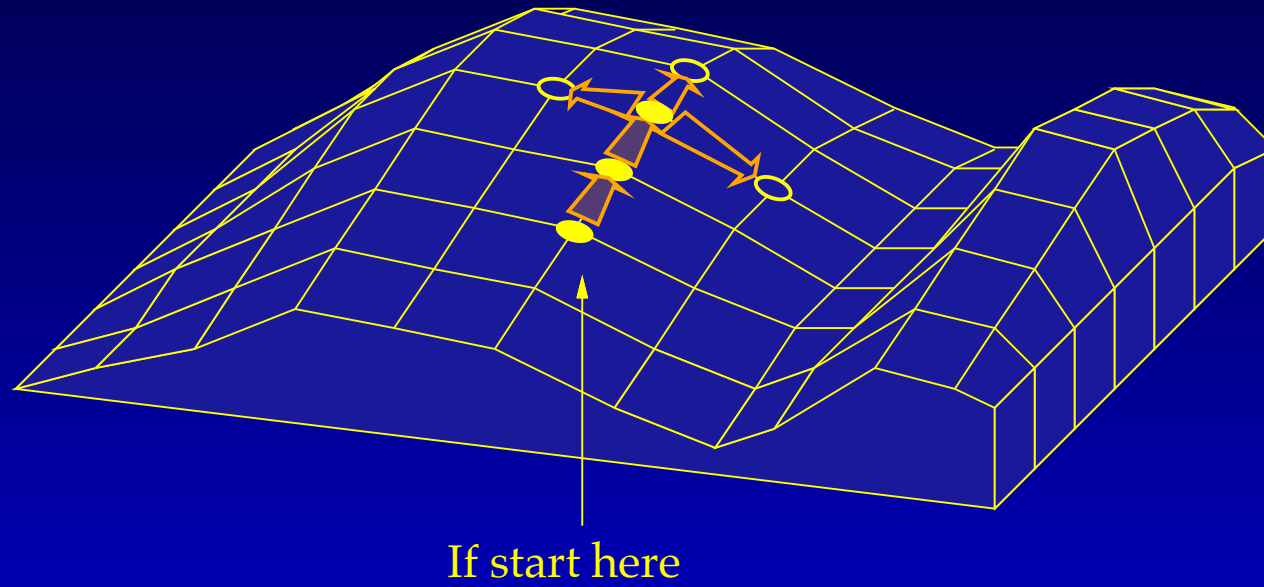
# Greedy search for a maximum



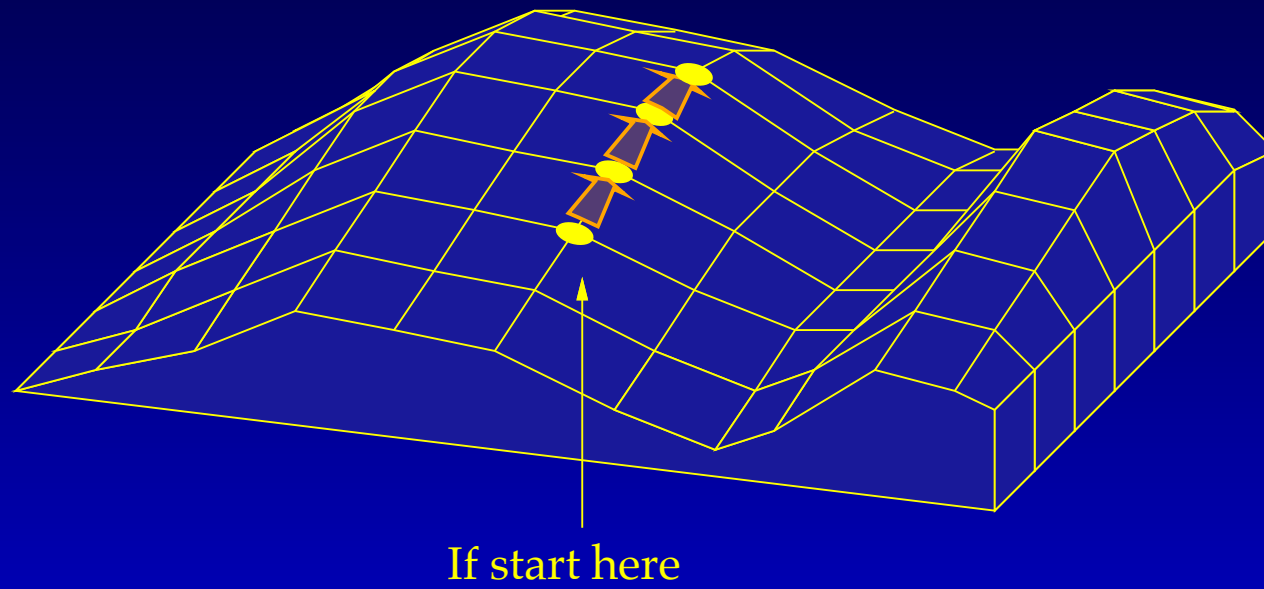
# Greedy search for a maximum



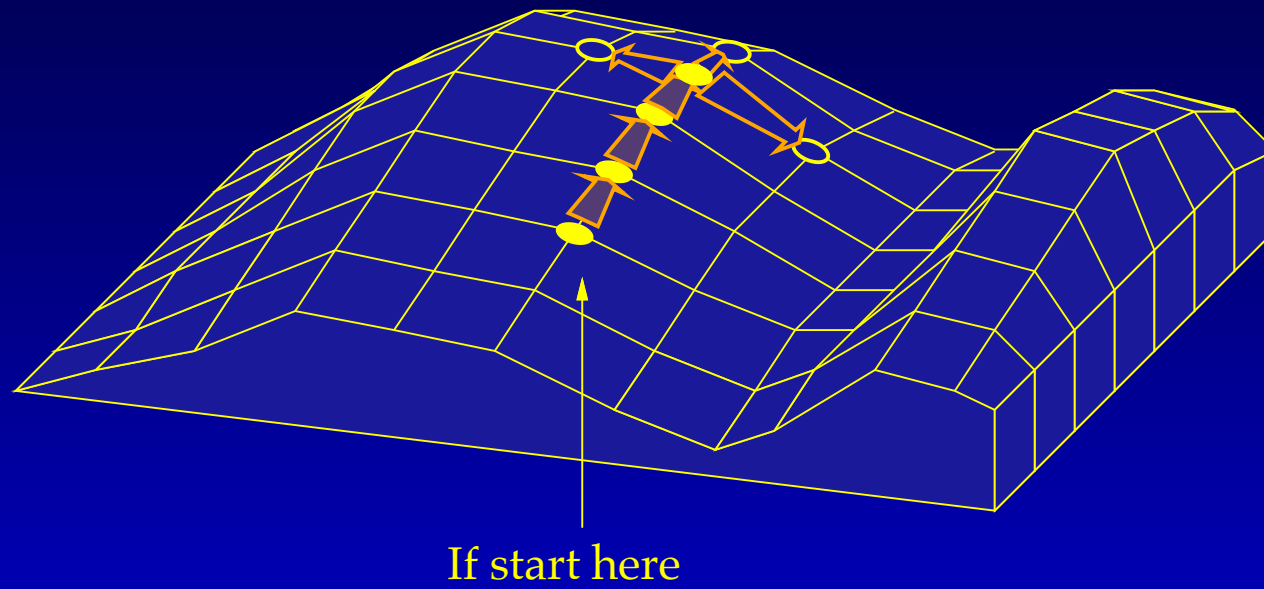
# Greedy search for a maximum



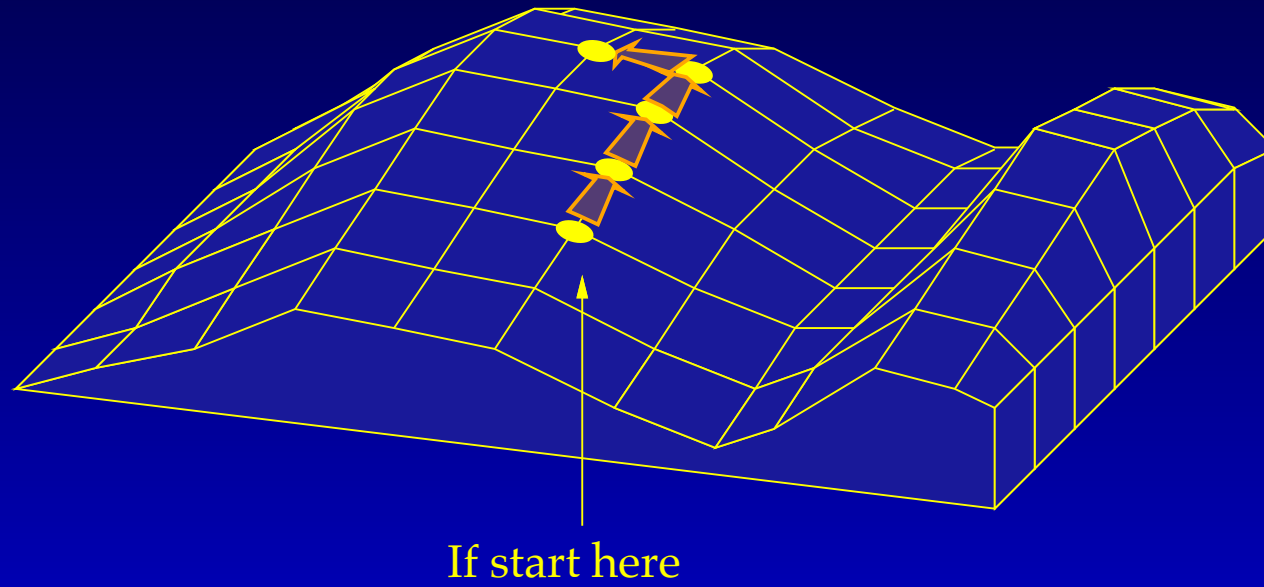
# Greedy search for a maximum



# Greedy search for a maximum

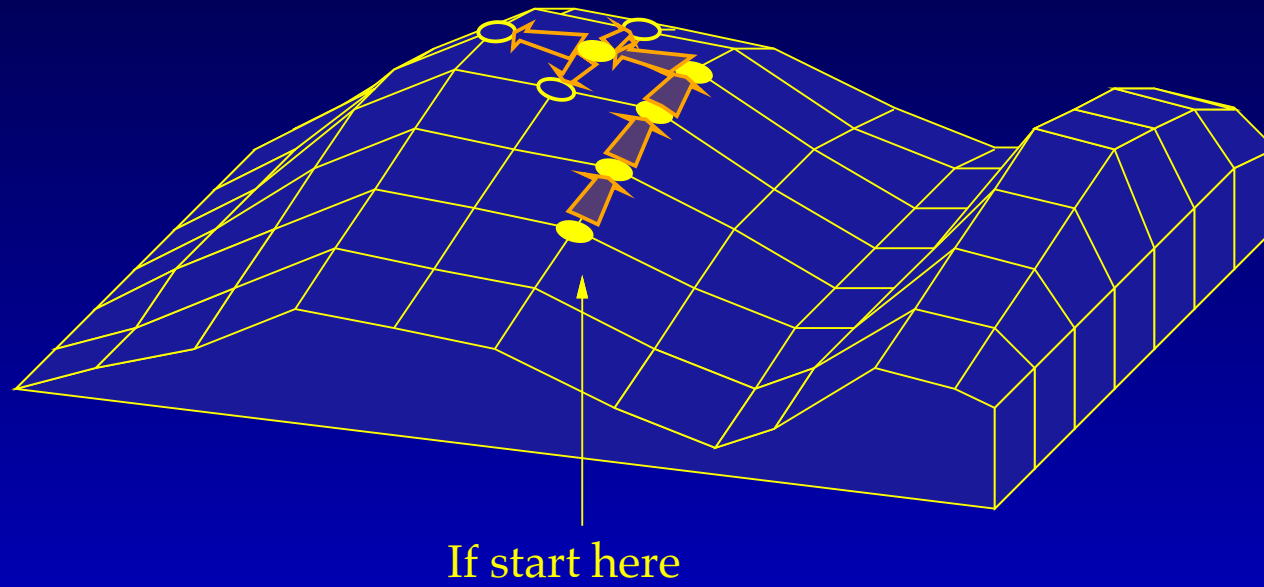


# Greedy search for a maximum

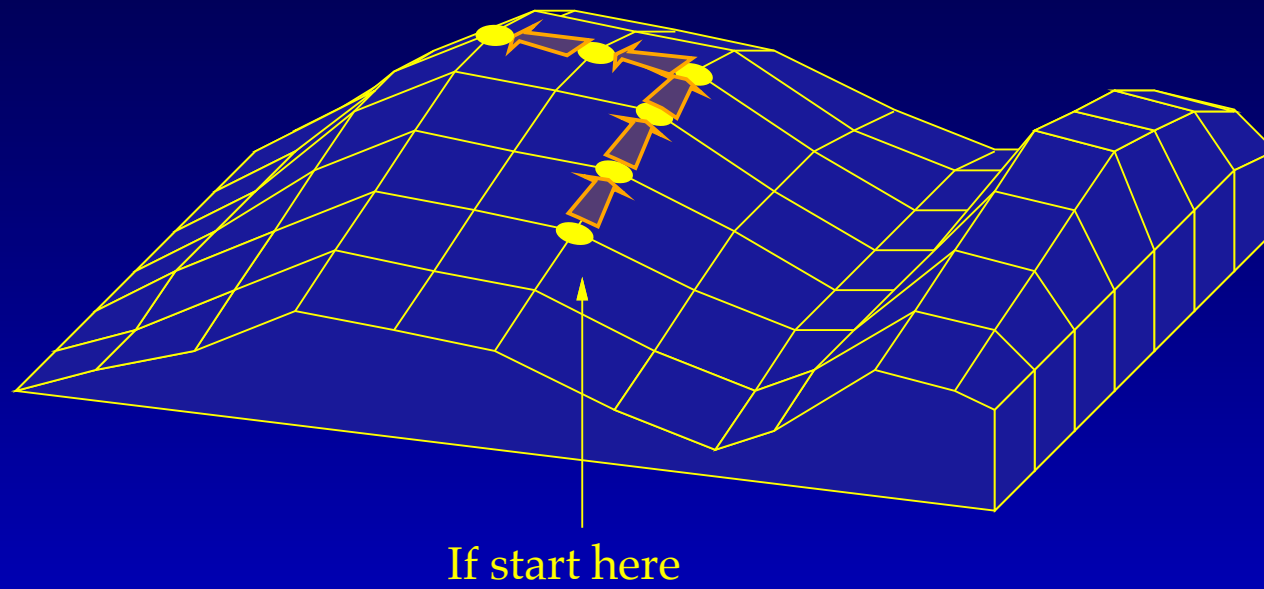




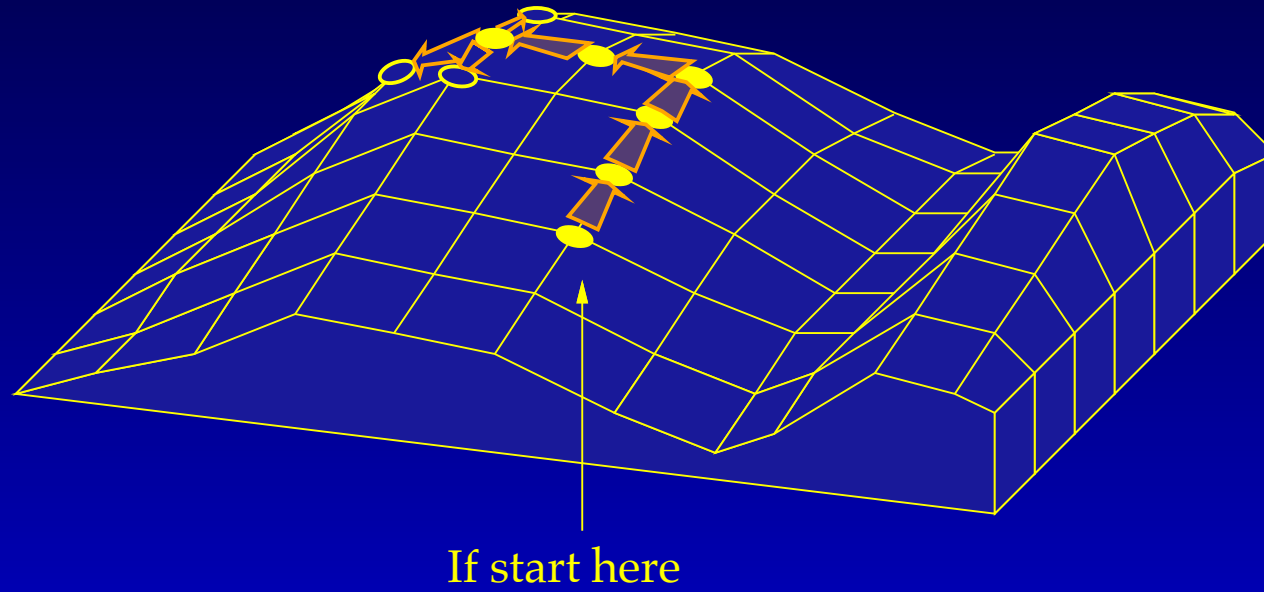
# Greedy search for a maximum



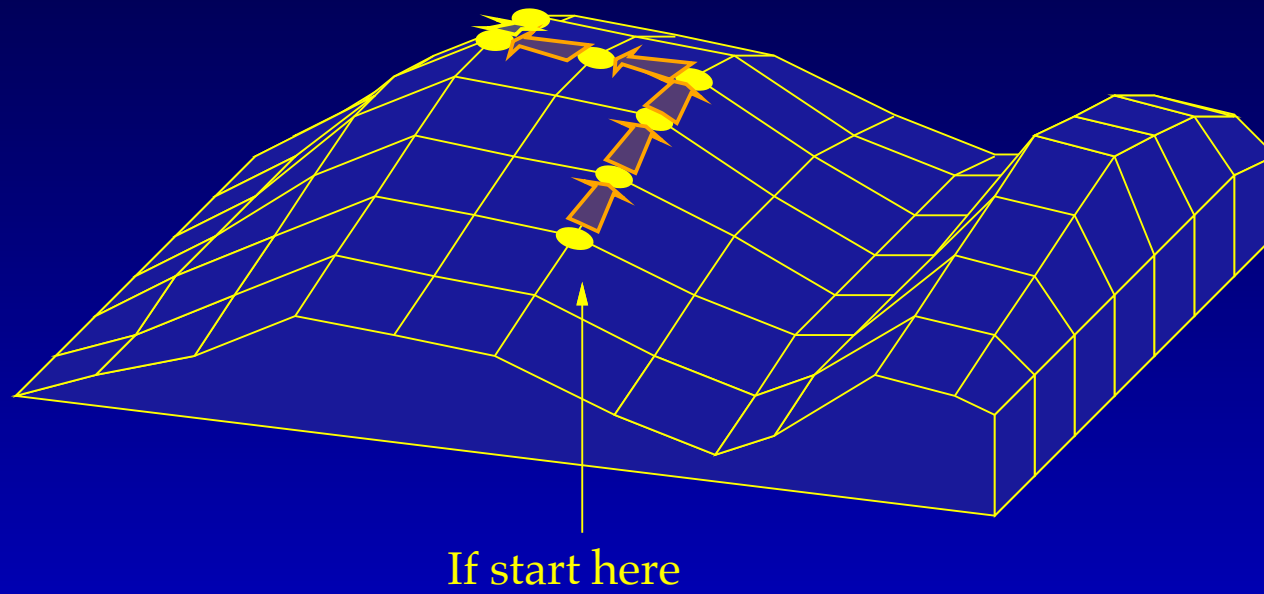
# Greedy search for a maximum



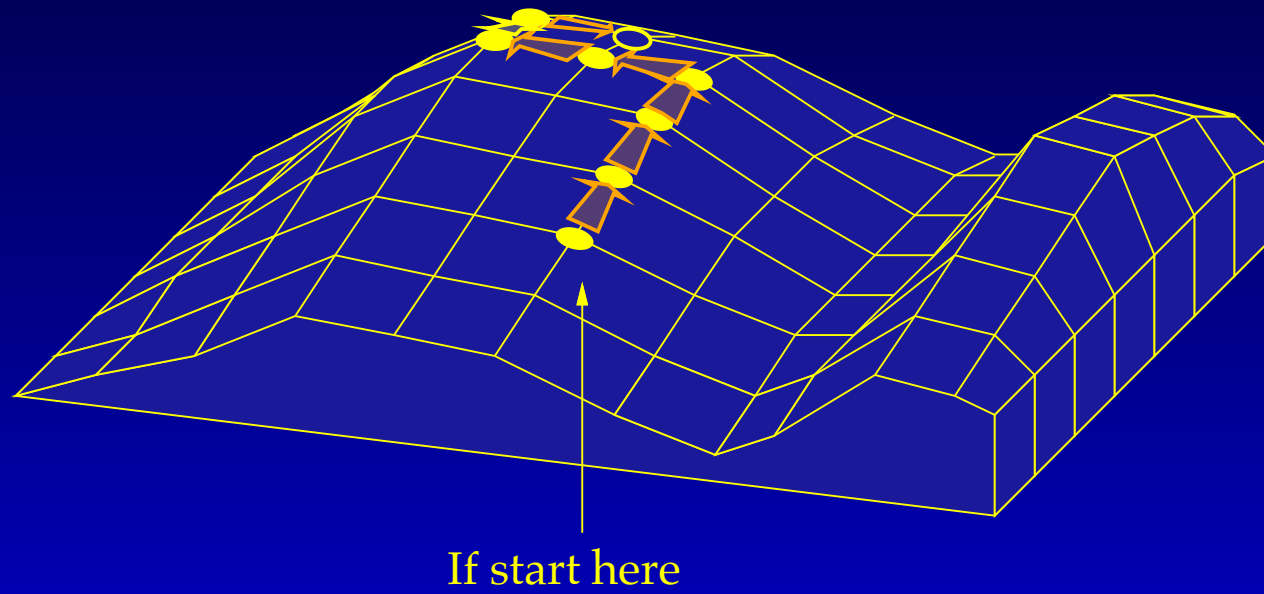
# Greedy search for a maximum



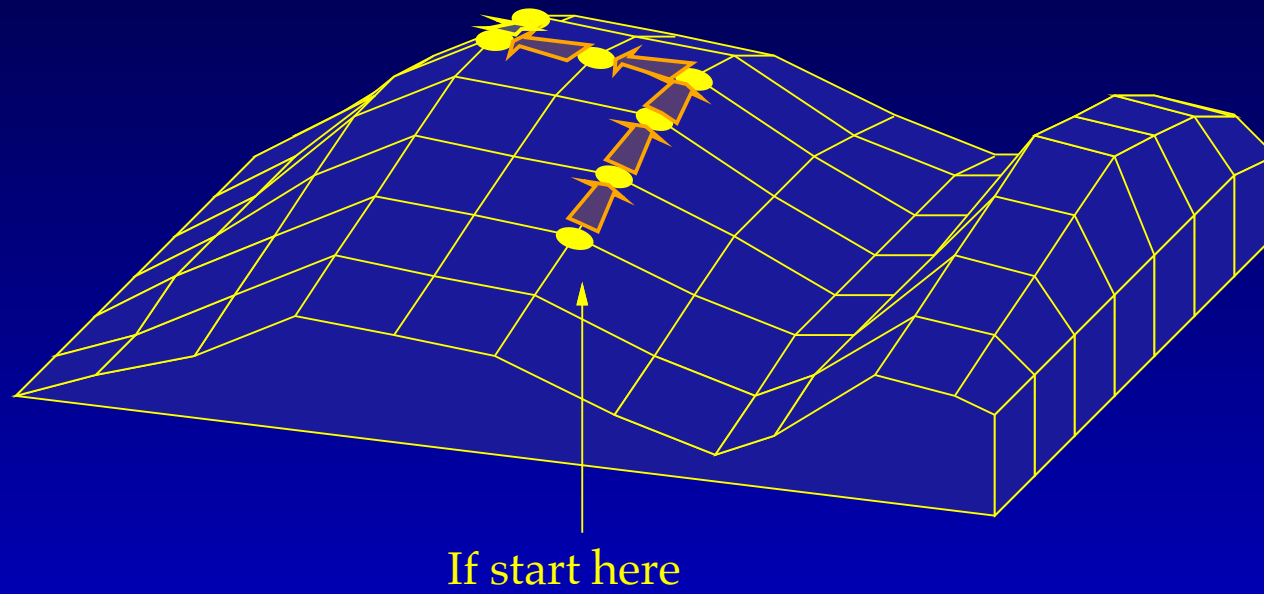
# Greedy search for a maximum



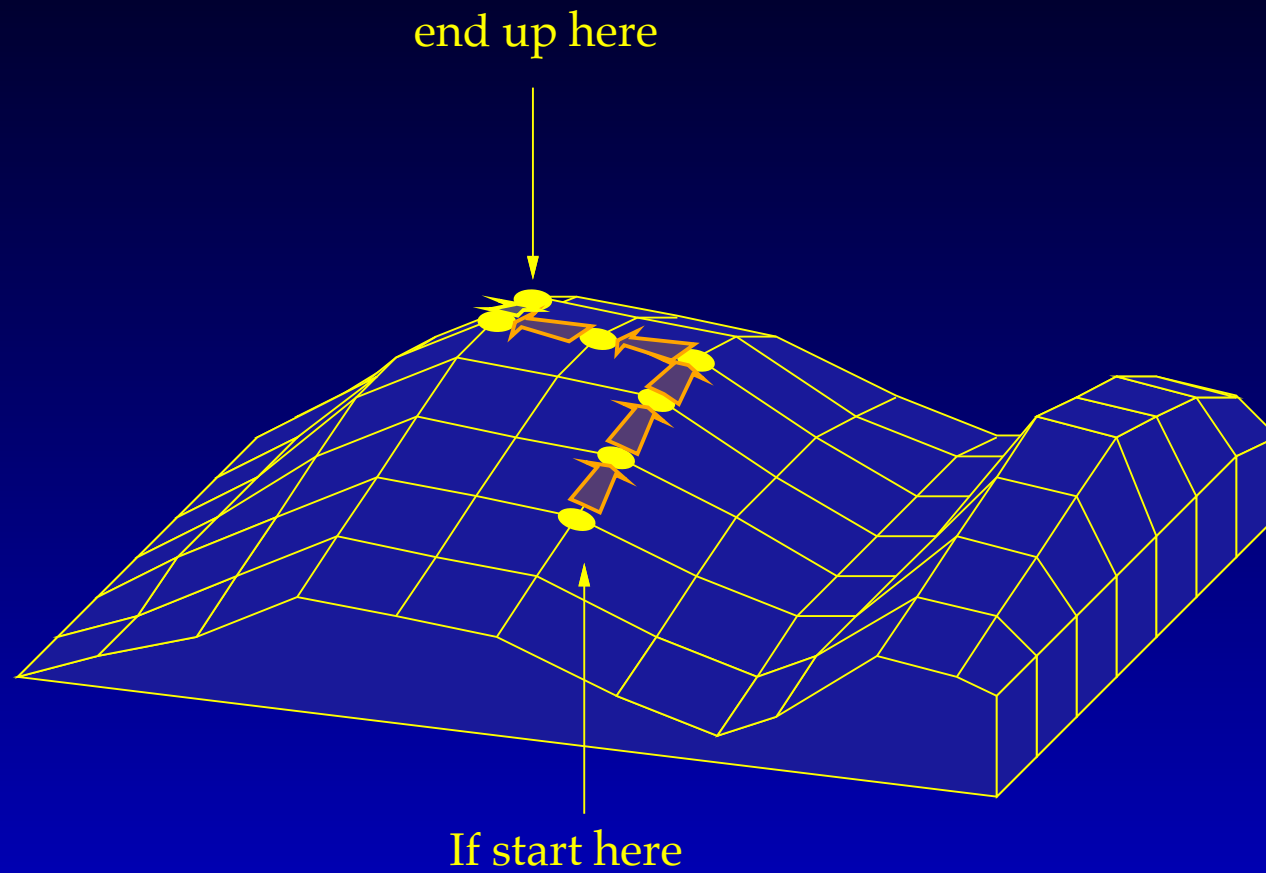
# Greedy search for a maximum



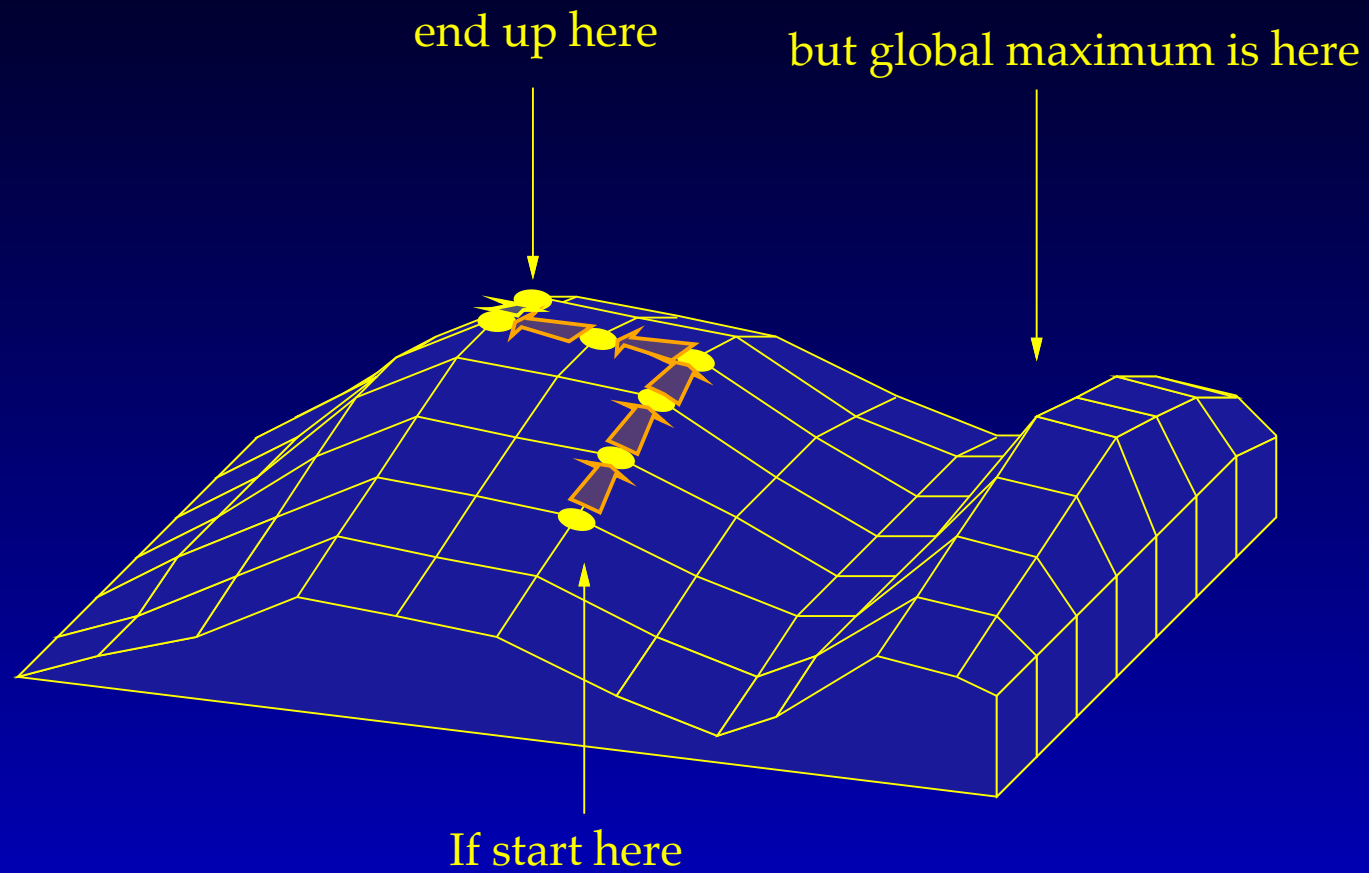
# Greedy search for a maximum



# Greedy search for a maximum

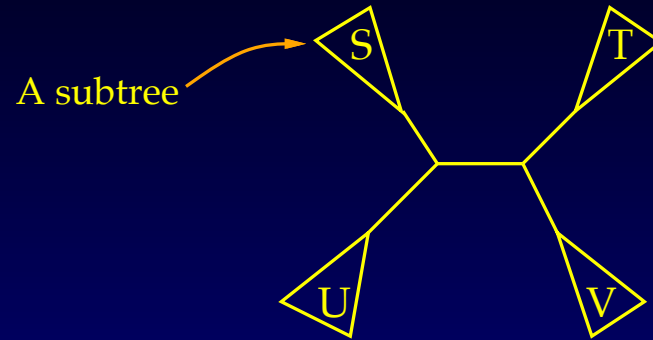


# Greedy search for a maximum

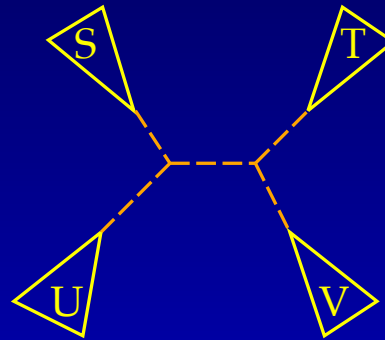




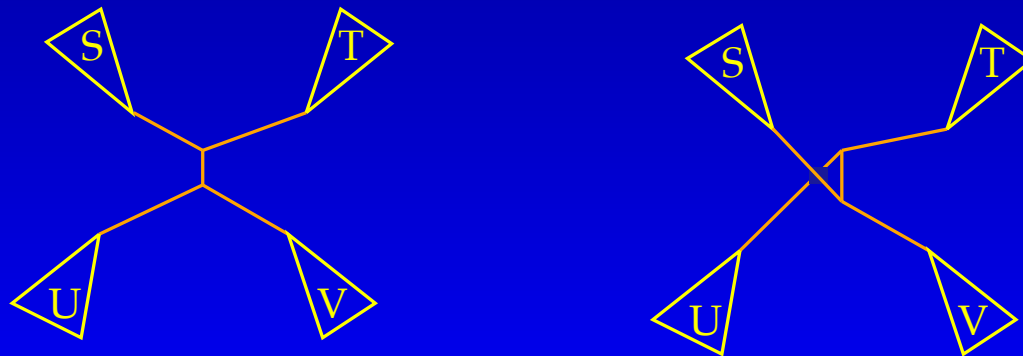
# Nearest-neighbor rearrangements



is rearranged by dissolving the connections to an interior branch



and reforming them in one of the two possible alternative ways:



## Nearest Neighbor Interchanges searches

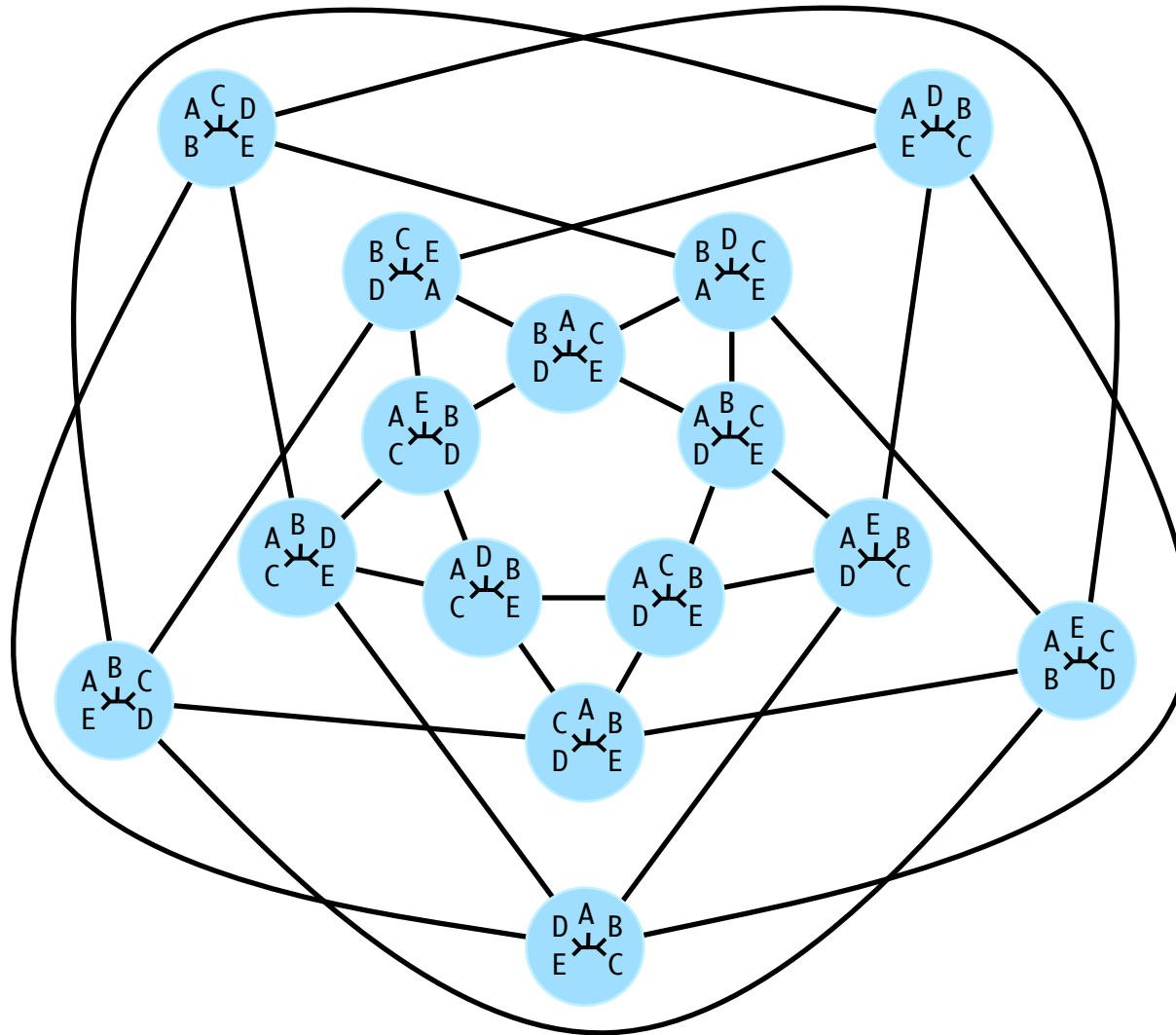
---

1. Consider the two possible NNI neighbors “around” each internal edge in the tree
2. Return the tree set of trees that is at least as good as all of the NNI neighbors.
3. Number of rearrangements scored *per tree*  $N$  taxa :

$$\# \text{ rearrangements scored} = 2 \times (N - 3)$$

But, there is no upper bound on the number of trees encountered on the path from the initial tree to the final tree.

## Schoenberg graph – edges connect NNI neighbors



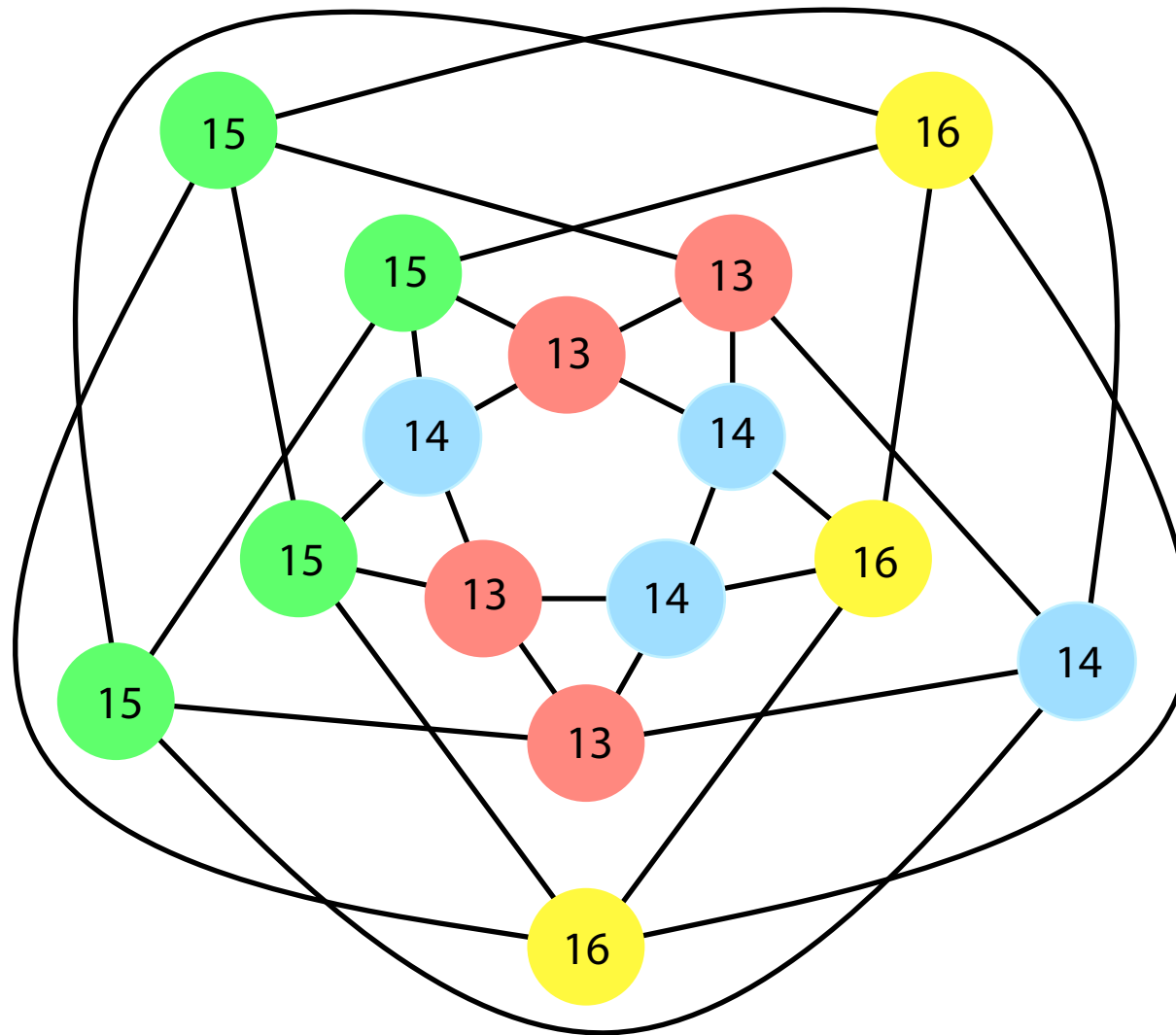
## **Tree “Islands” (Maddison, 1991) are possible**

---

The following Schoenberg graph shows the scores of the 15 trees on the following dataset (contrived data by POL):

A ACGCAGGT  
B ATGGTGAT  
C GCTCACGG  
D ACTGTCGT  
E GTTCTGAG

## Schoenberg graph with parsimony scores



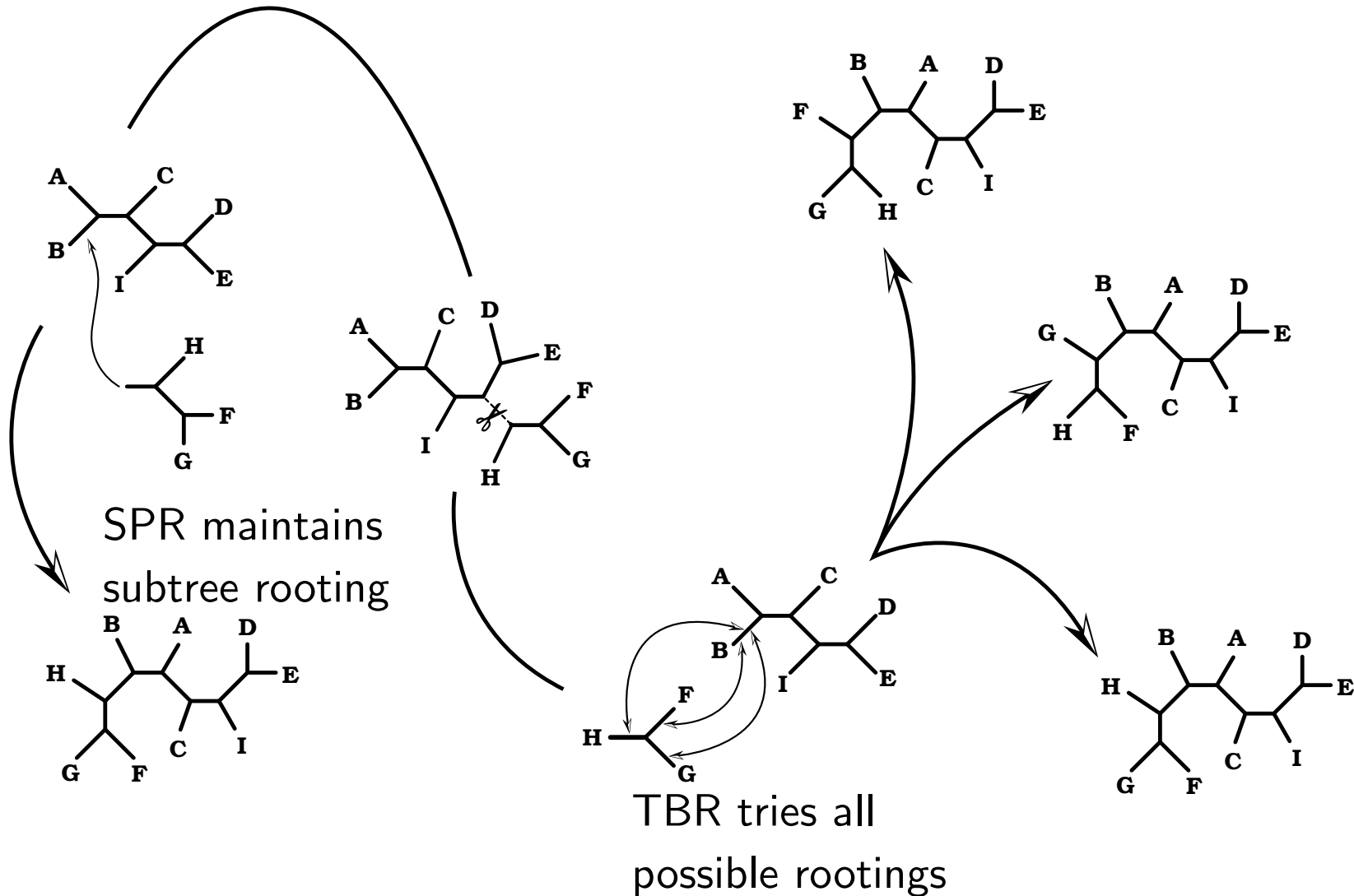
## **Tree Islands implications**

---

1. Islands can be larger than 1 tree – so we must consider ties if we want to find all optimal trees.
2. There can be more than 1 island for good scores: swapping to completion on all optimal trees found in a search is not guaranteed to succeed.
3. The delimitation of an island depends on tree changing operation used.

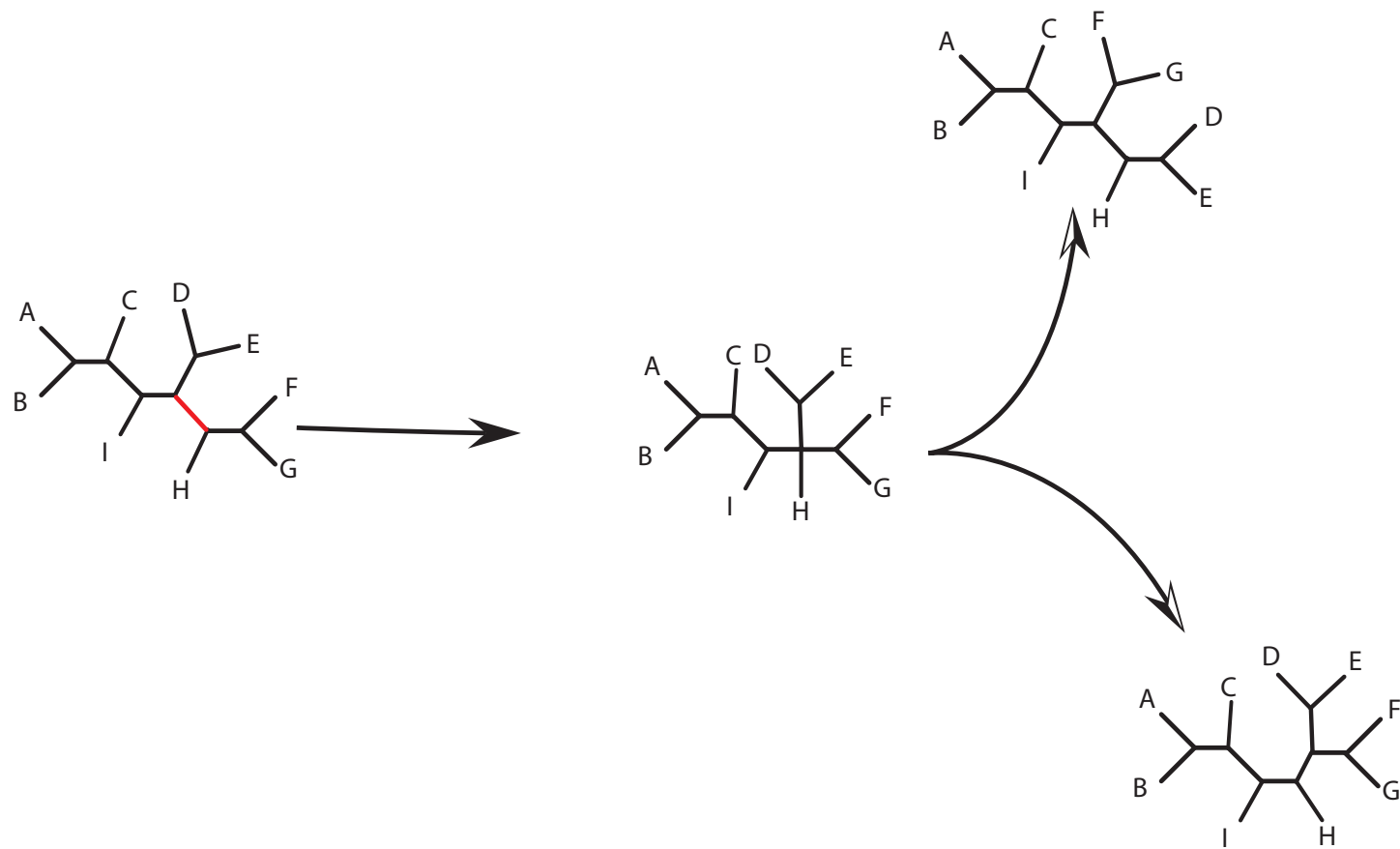
# Subtree Pruning Regrafting (SPR) and Tree Bisection Reconnection (TBR)

---



# 1-Edge-contract Refine

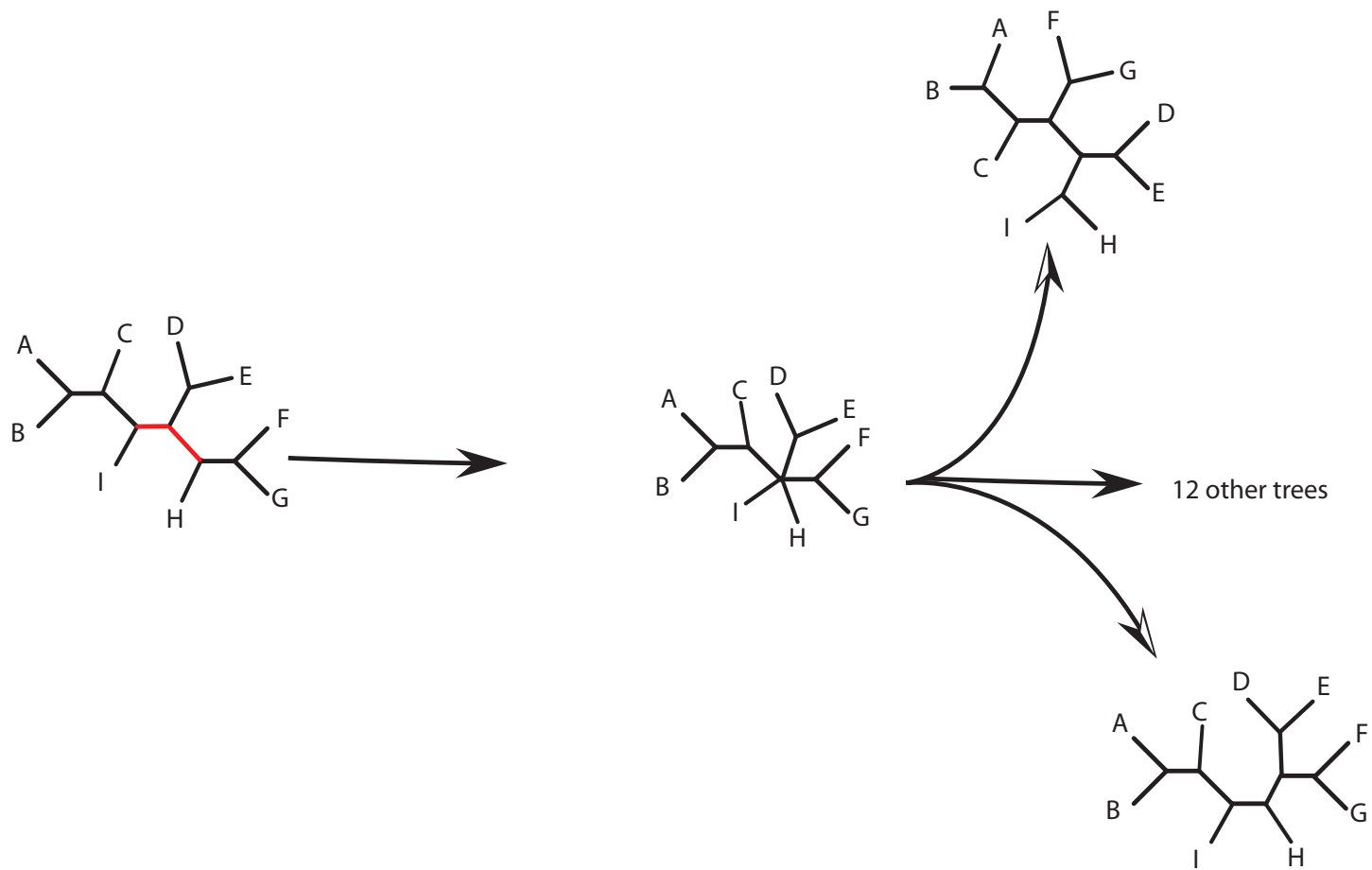
---





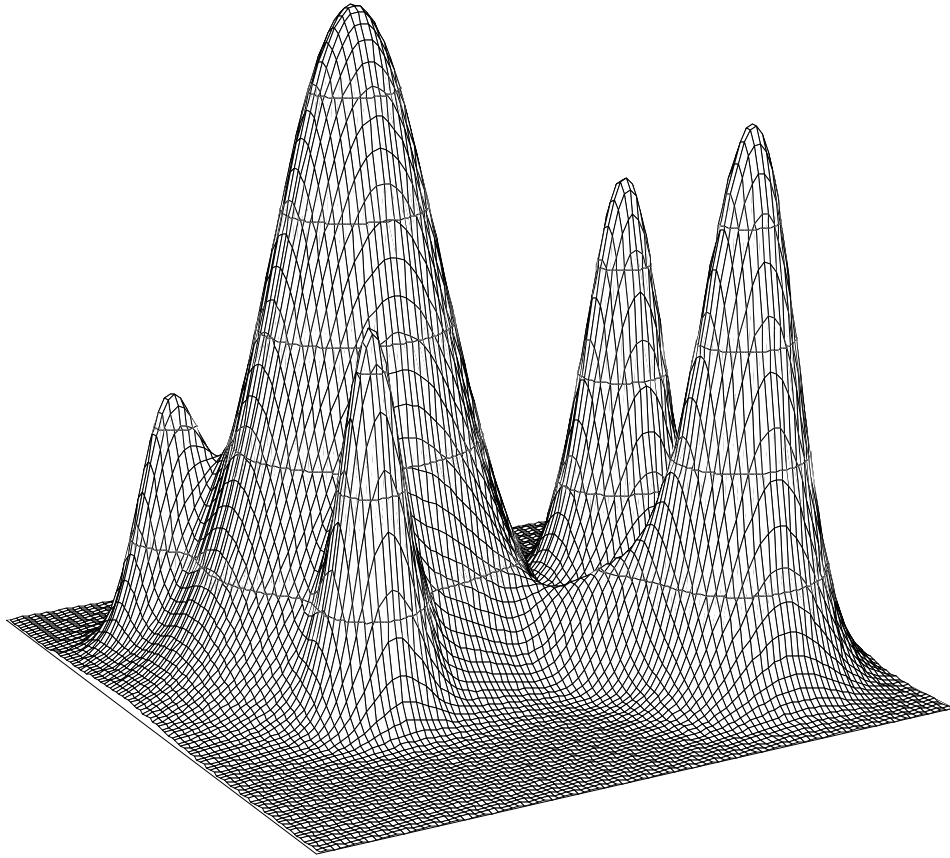
## 2-Edge-contract Refine

---



# Heuristics explore “Tree Space”

---



Many commonly-used methods are “hill-climbers.”

Multiple optima found by repeating searches from different origins.

Severity of the problem of multiple optima depends on step size.

## **Branch-swapping tradeoff**

---

Using more drastic moves (such as TBR instead of NNI):

- + you can jump between “hills”
- many neighboring trees to score
- if your current tree is good, high % of neighbors have much worse likelihood.

## **Nested optimizaton**

---

Recall that we have to optimize parameters for each tree:

<http://phylo.bio.ku.edu/mephytis/brlen-opt.html>

## **Need to get branch support**

Recall that bootstrapping is even more expensive:

<http://phylo.bio.ku.edu/mephytis/boot-sample.html>