Lecture on

Local Search
Hill Climbing Algorithm

Dr. Partha Pakray

Local Search

Local search methods work on complete state formulations. They keep only a small number of nodes in memory.

Local search is useful for solving optimization problems:

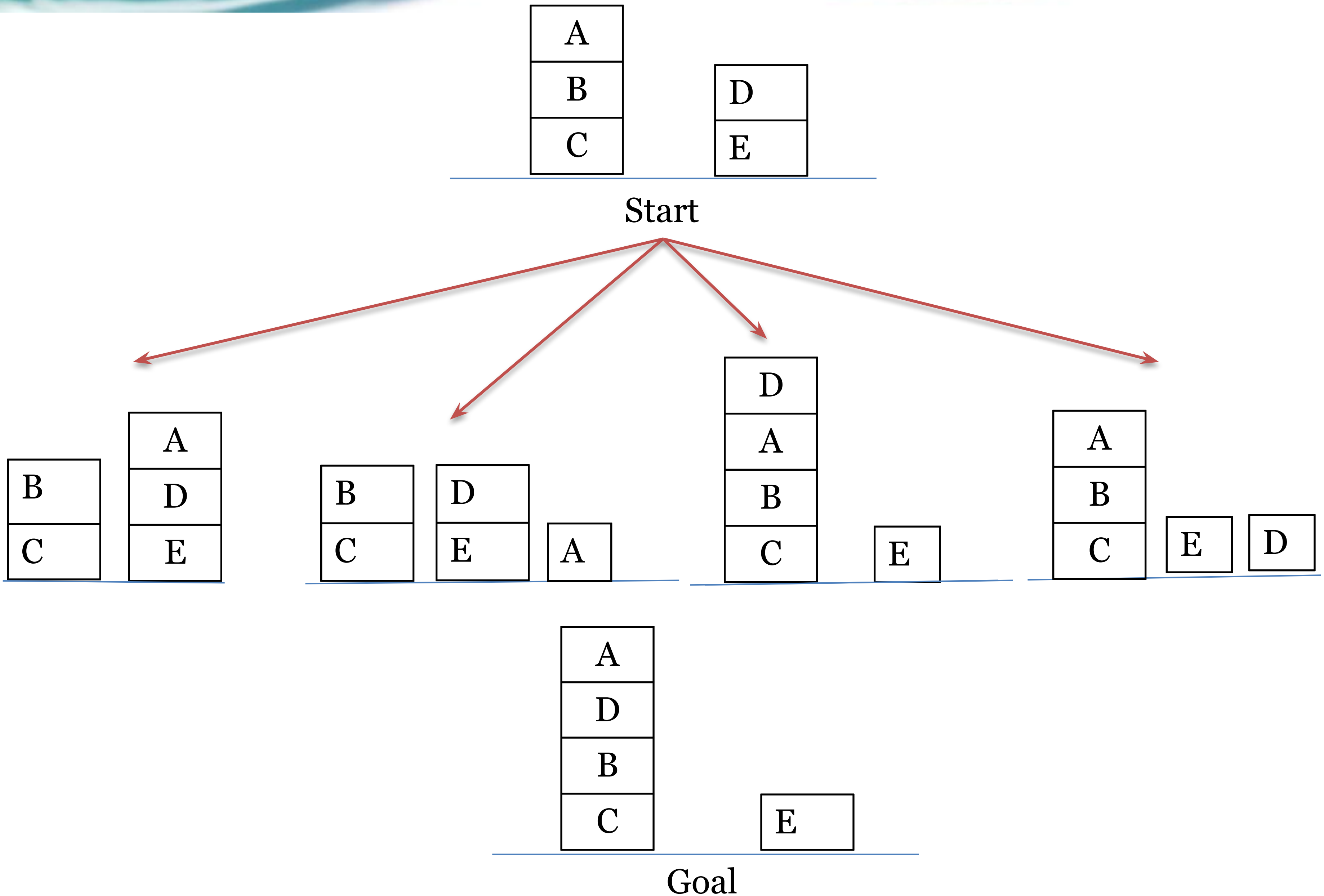
- **Often it is easy to find a solution**
- **But hard to find the best solution**

Algorithm goal: find optimal configuration (e.g., TSP),

- Hill climbing
- Gradient descent
- Simulated annealing

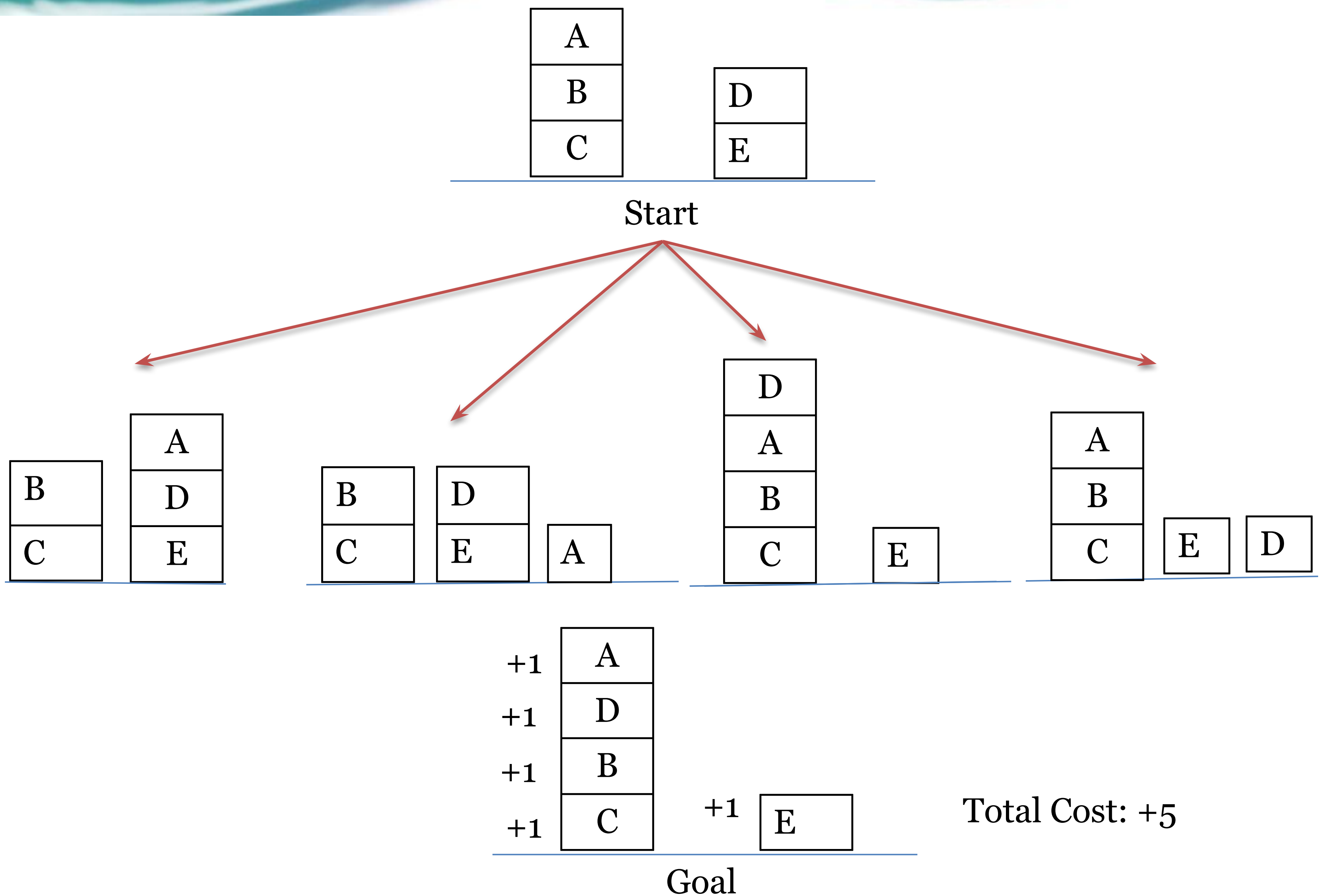
- For some problems the state description contains all of the information relevant for a solution. Path to the solution is unimportant.
- Examples:
  - map coloring
  - 8-queens
  - cryptarithmetic

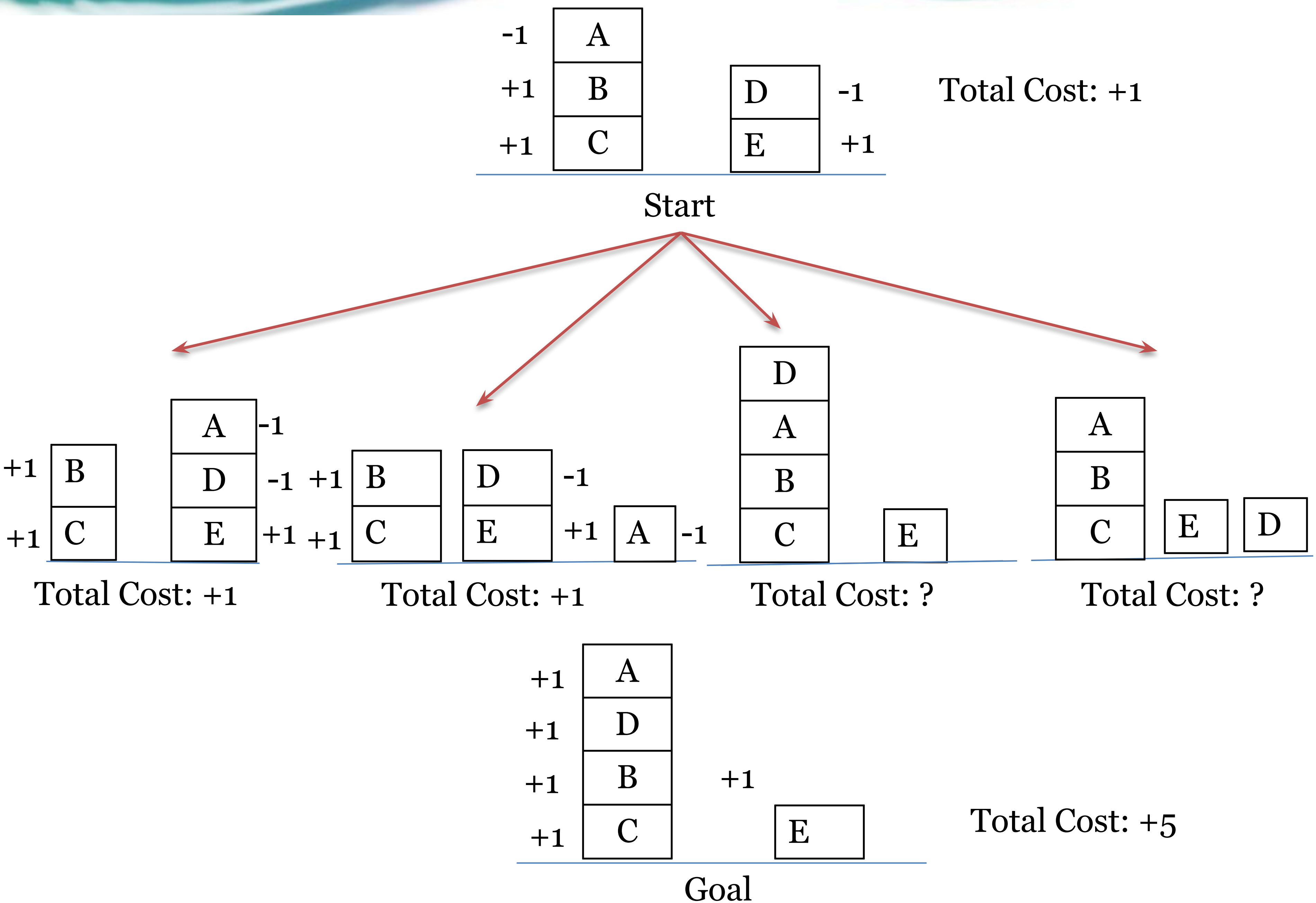
Block Worlds Domain


A
B
C
D
E
Start
B
C
A
D
E
B
C
D
E
A
D
A
B
C
E
A
B
C
E
D
A
D
B
C
E
Goal

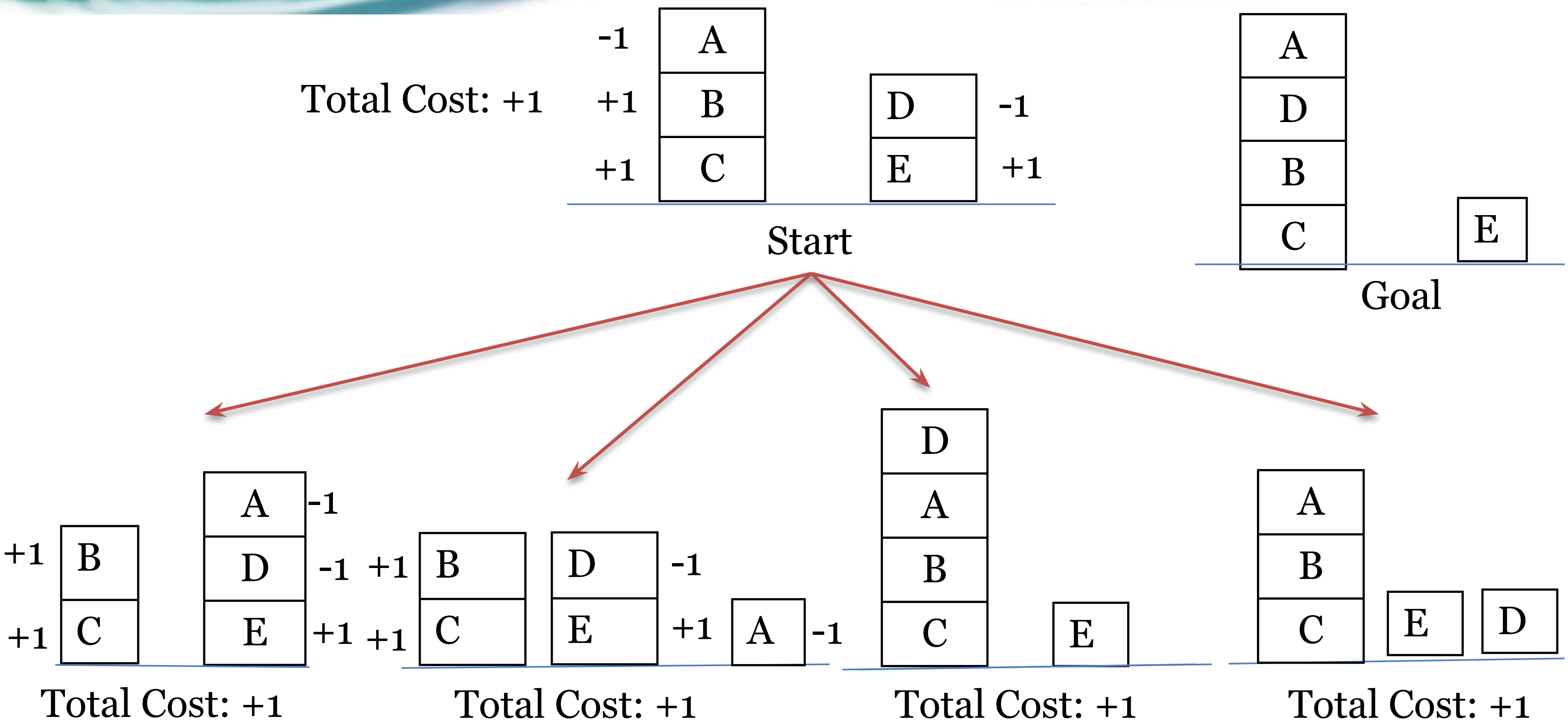
Rule: Heuristic Function

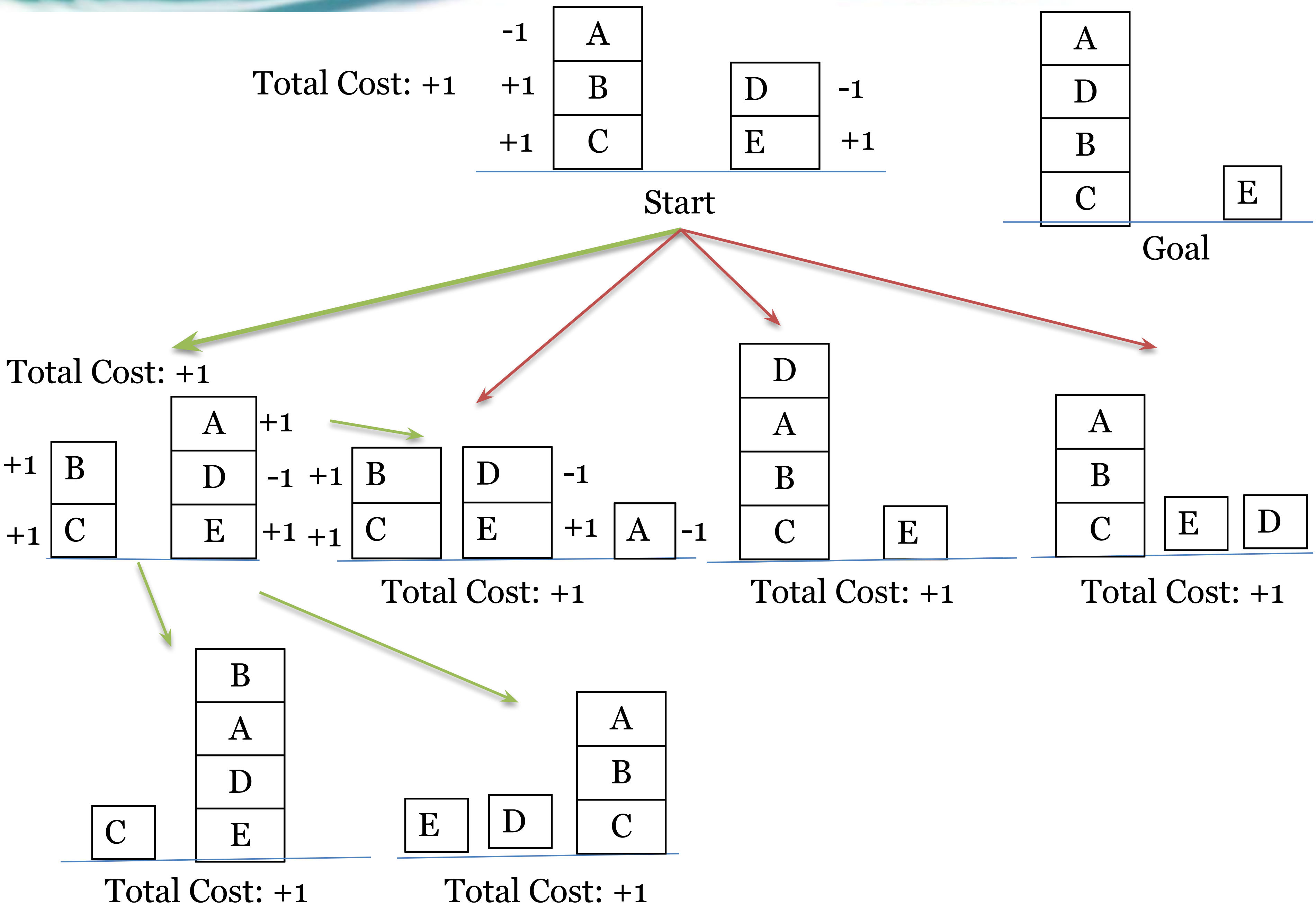
h1(n)= add 1 if the block is on the correct location
subtract 1 if the block is on the incorrect location


A
B
C
D
E
Start
B
C
A
D
E
B
C
D
E
A
D
A
B
C
E
A
B
C
E
D
+1
A
+1
D
+1
B
+1
C
+1
E
Goal
Total Cost: +5



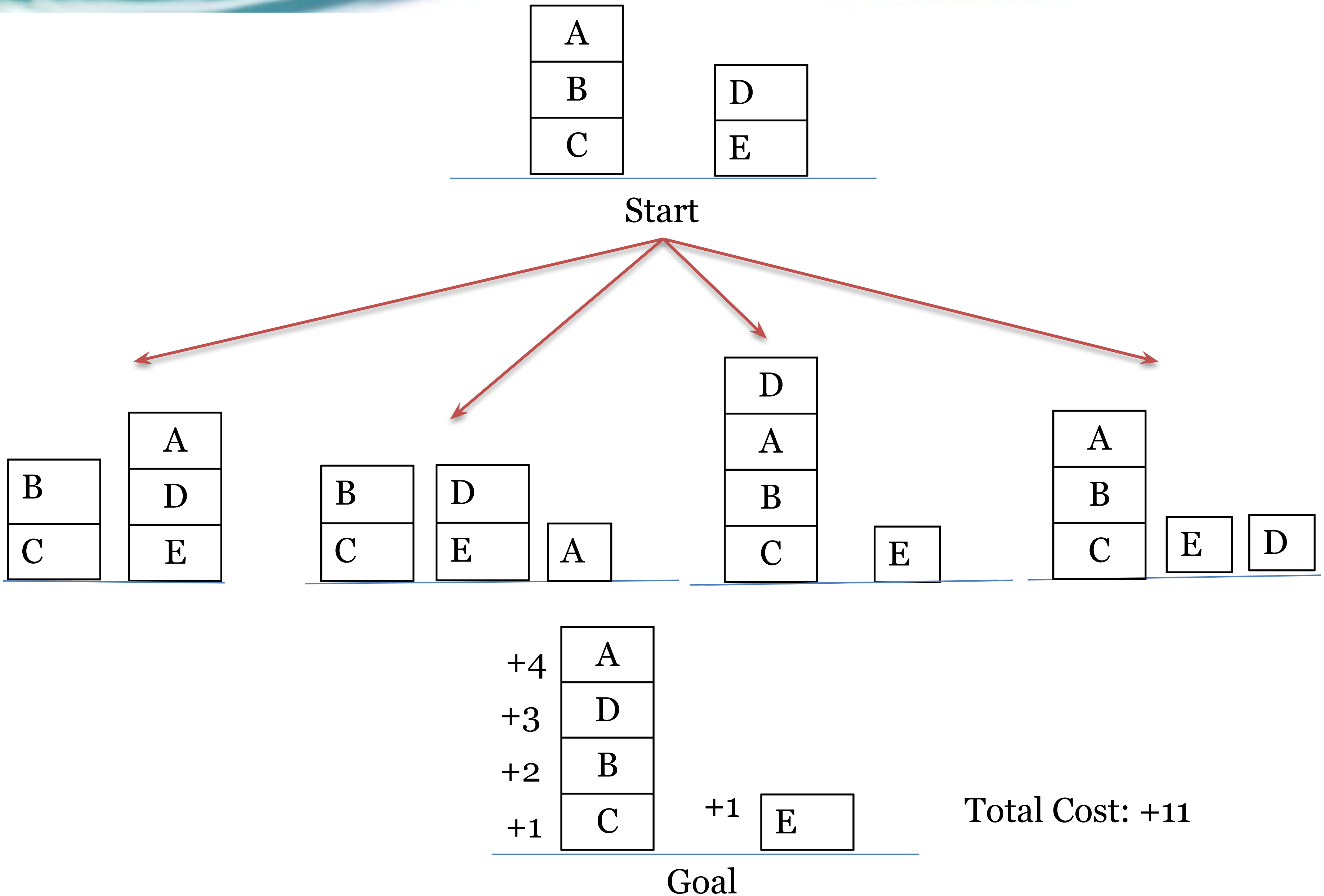
-1 A
+1 B
+1 C
D -1
E +1
Total Cost: +1
Start
+1 B
+1 C
A -1
D -1
E +1
Total Cost: +1
+1 B
+1 C
D -1
E +1
A -1
Total Cost: +1
D
A
B
C
E
Total Cost: ?
A
B
C
E
D
Total Cost: ?
+1 A
+1 D
+1 B
+1 C
+1
E
Total Cost: +5
Goal


Total Cost: +1
-1
A
+1
B
+1
C
D
-1
E
+1
Start
A
D
B
C
E
Goal
+1
B
+1
C
A
-1
D
-1
E
+1
Total Cost: +1
+1
B
+1
C
D
-1
E
+1
A
-1
Total Cost: +1
D
A
B
C
E
Total Cost: +1
A
B
C
E
D
Total Cost: +1

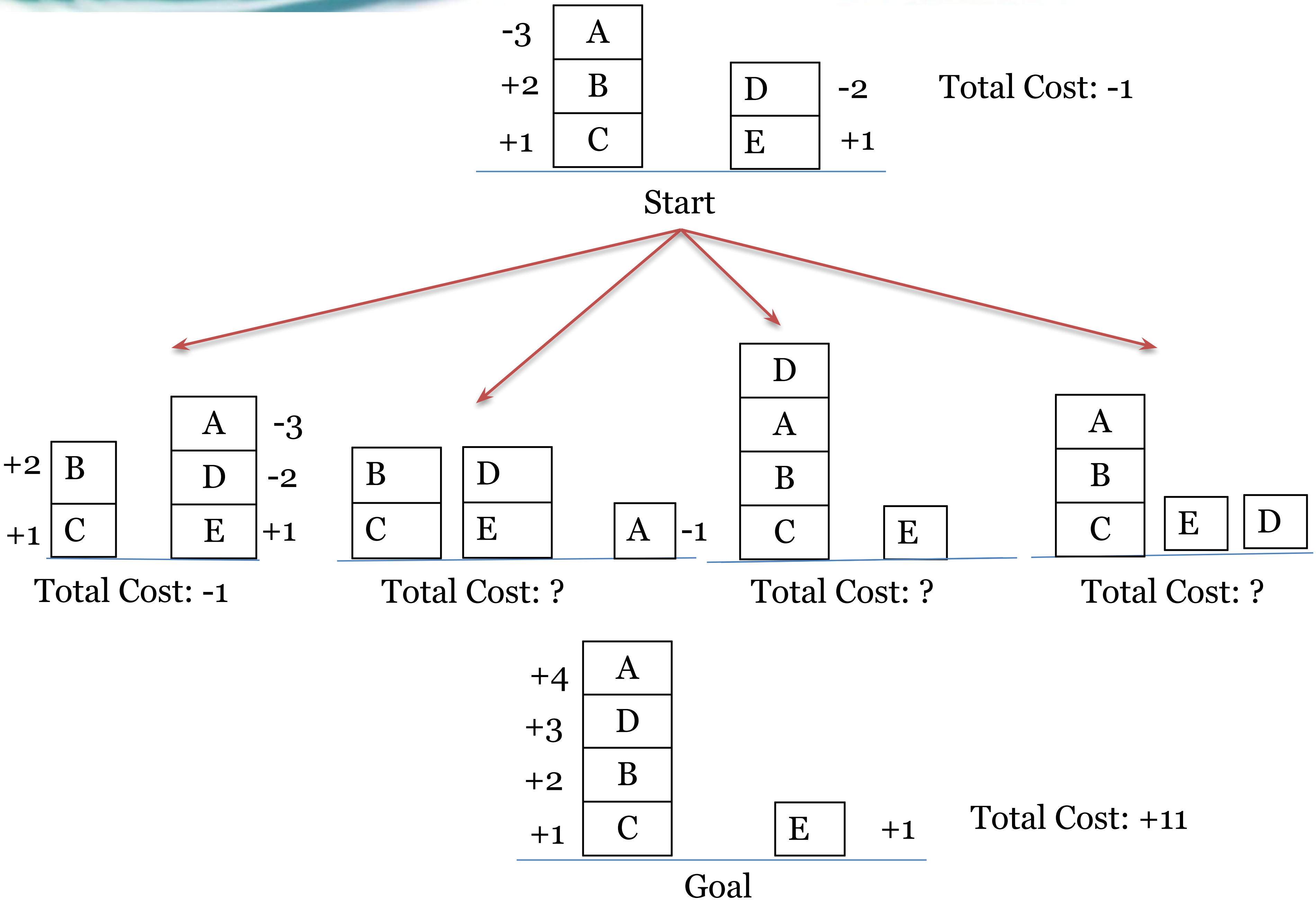

Total Cost: +1
-1
+1
+1
A
B
C
D
E
-1
+1
Start
A
D
B
C
E
Goal
Total Cost: +1
+1
+1
B
C
A
D
E
+1
-1
+1
+1
+1
B
C
D
E
-1
+1
A
-1
Total Cost: +1
D
A
B
C
E
Total Cost: +1
A
B
C
E
D
Total Cost: +1
B
A
D
E
C
Total Cost: +1
E
D
A
B
C
Total Cost: +1

Rule

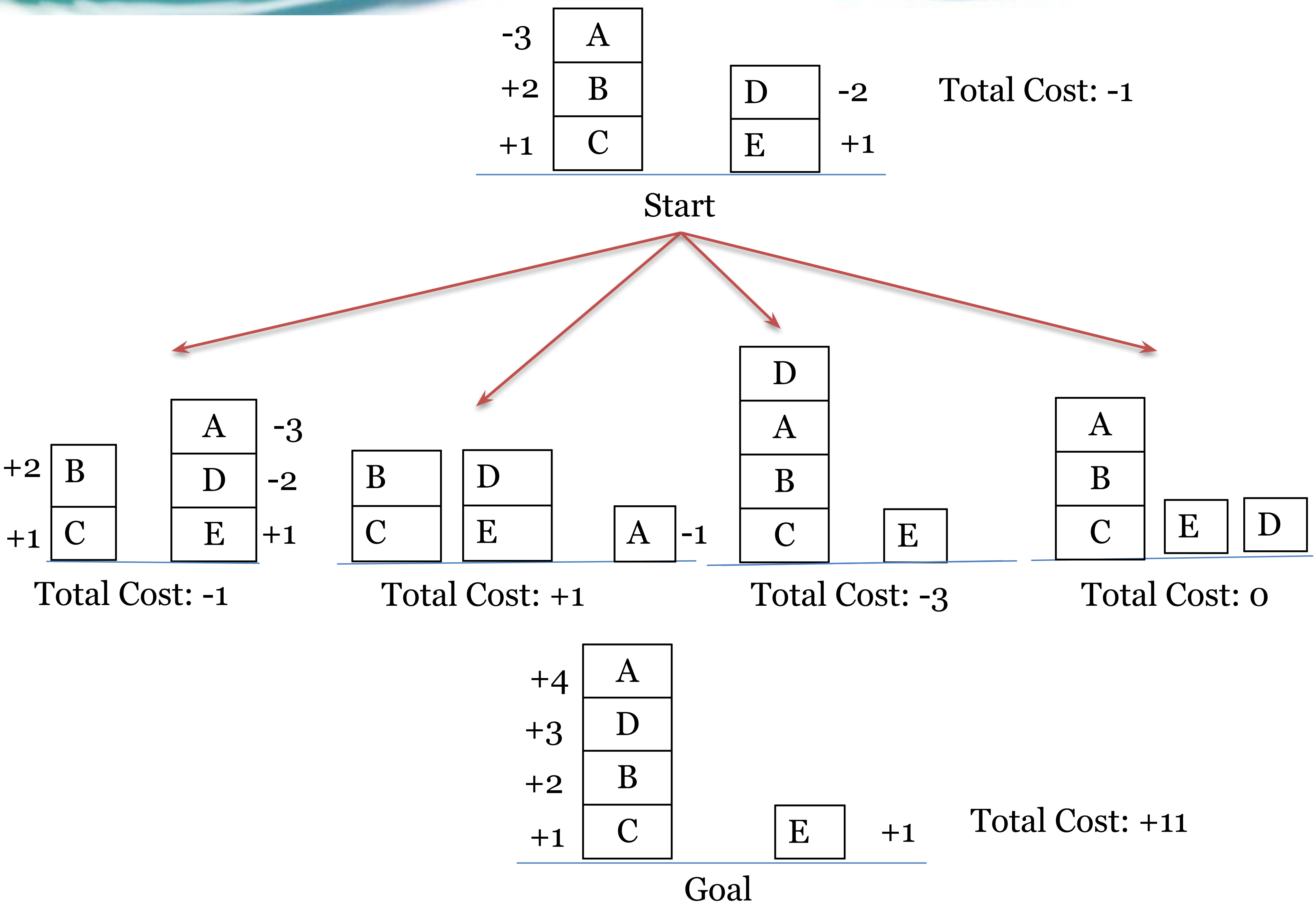
h2(n)= add 1 for every block in the correct structure that the block in sitting on
subtract 1 for every block in the incorrect structure



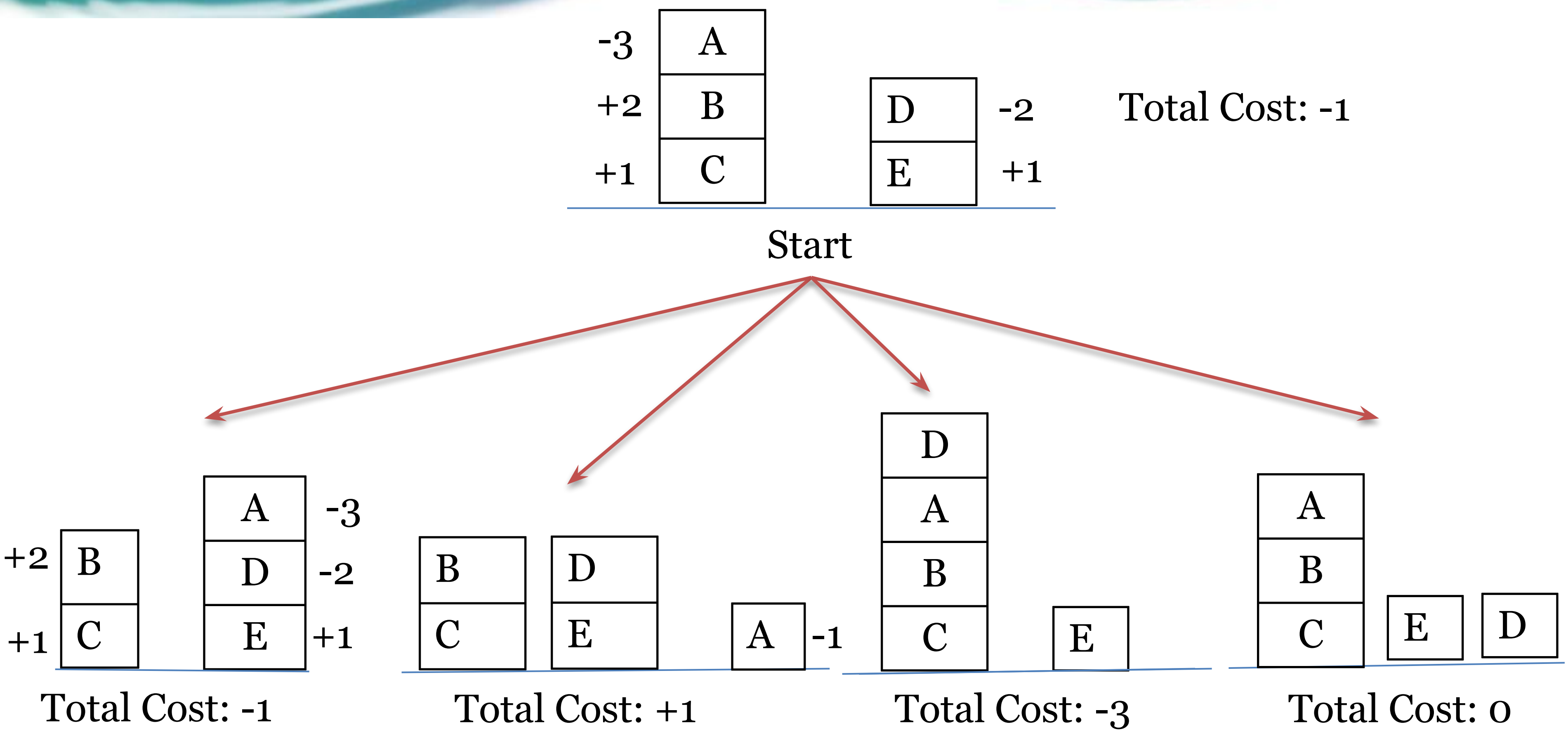
Total Cost: +11

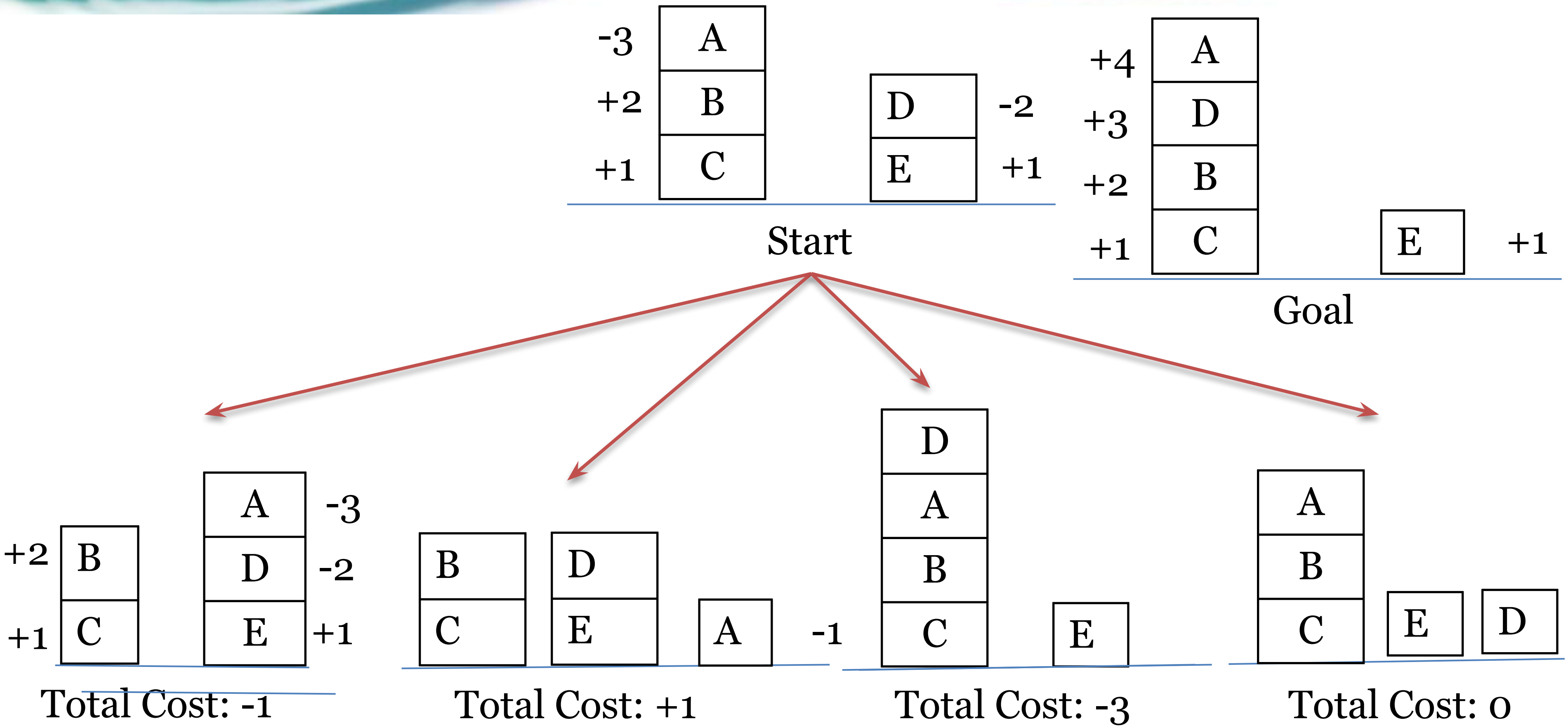


-3 A
+2 B
+1 C
D -2
E +1
Total Cost: -1
Start
+2 B
+1 C
A -3
D -2
E +1
Total Cost: -1
B
C
D
E
A -1
Total Cost: ?
D
A
B
C
E
Total Cost: ?
A
B
C
E
D
Total Cost: ?
+4 A
+3 D
+2 B
+1 C
E +1
Total Cost: +11
Goal

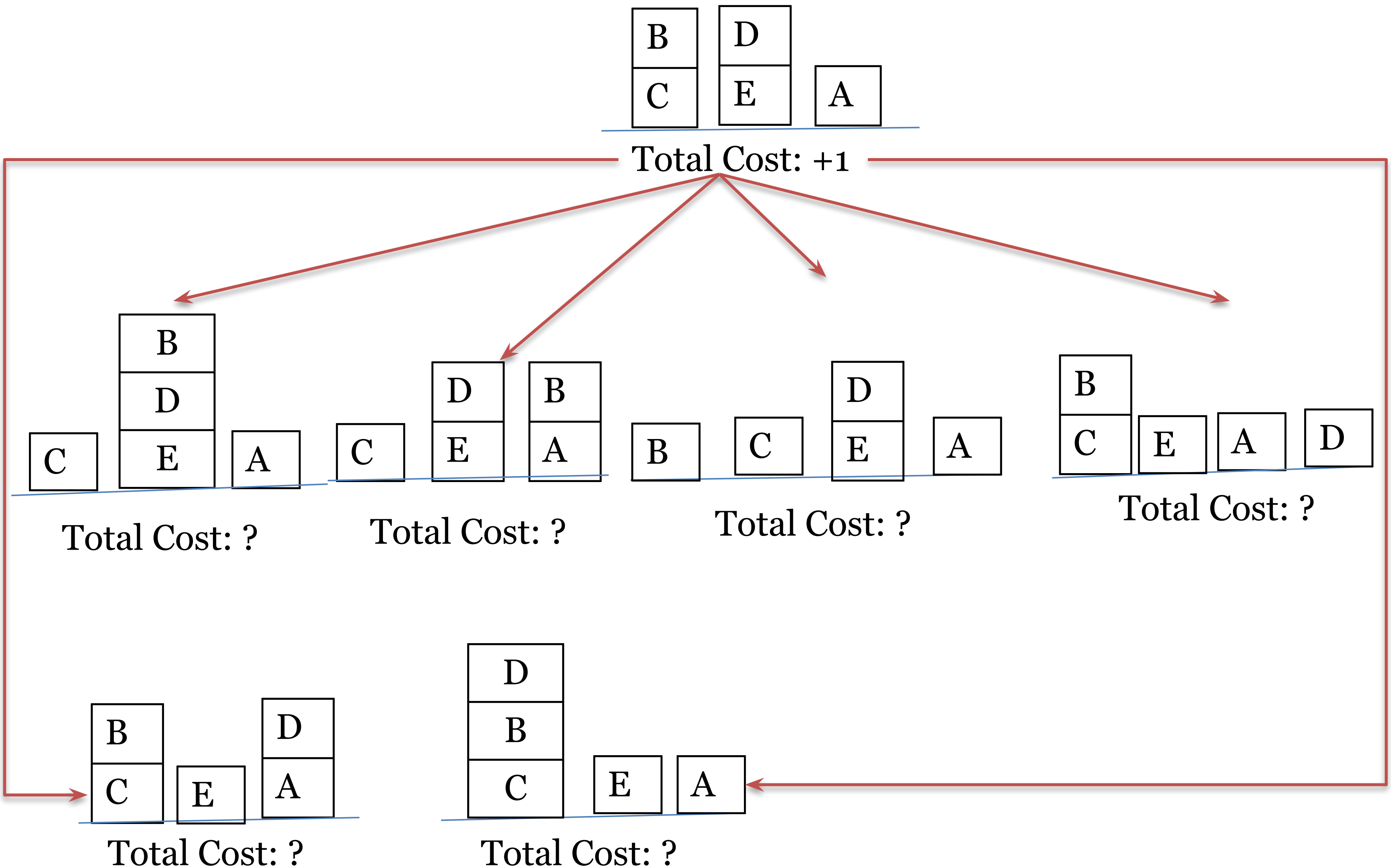


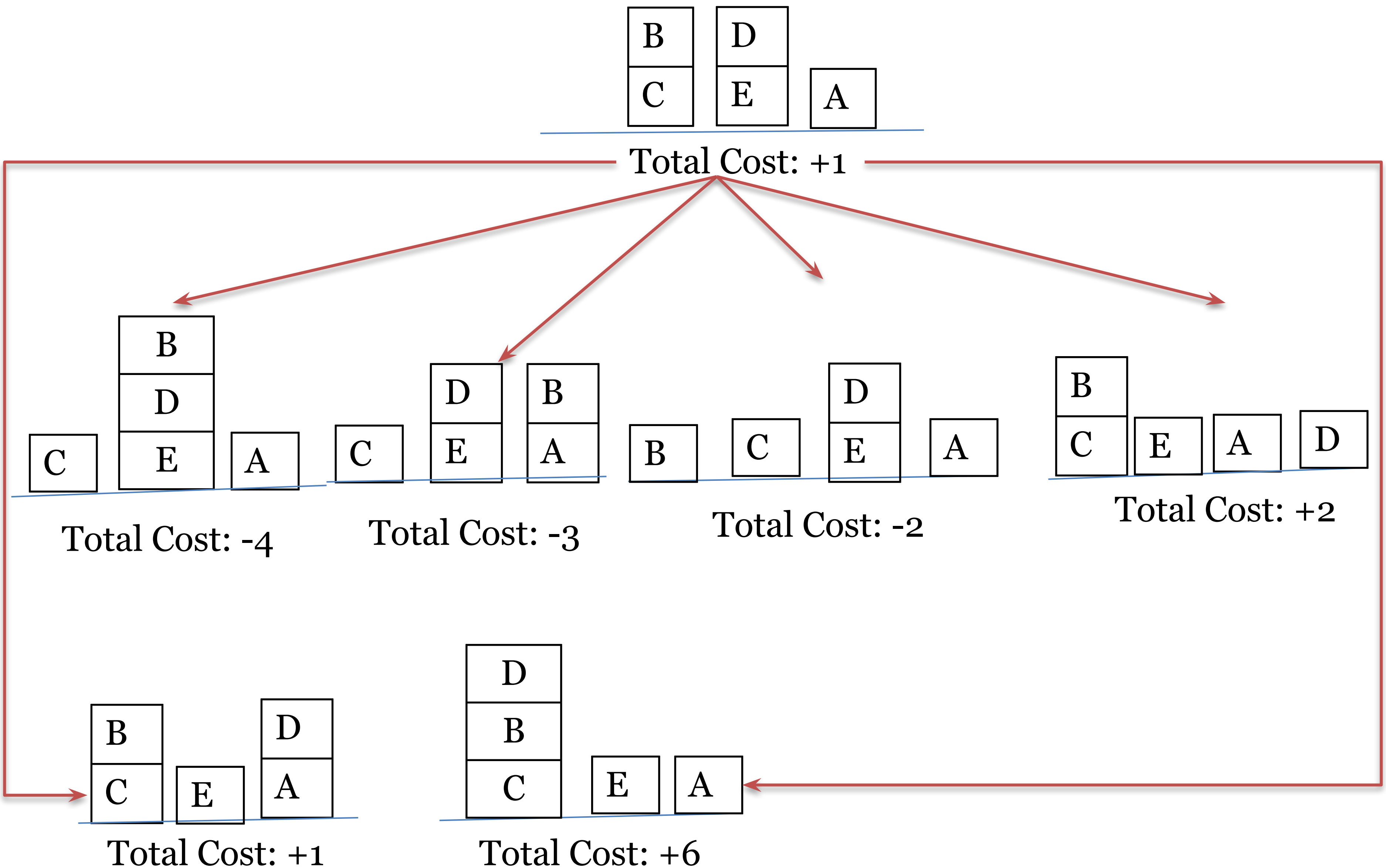
-3
A
+2
B
+1
C
D
-2
E
+1
Total Cost: -1
Start
+2
B
+1
C
A
-3
D
-2
E
+1
Total Cost: -1
B
C
D
E
A
-1
Total Cost: +1
D
A
B
C
E
Total Cost: -3
A
B
C
E
D
Total Cost: 0
+4
A
+3
D
+2
B
+1
C
E
+1
Total Cost: +11
Goal


-3
A
+2
B
+1
C
D
-2
E
+1
Total Cost: -1
Start
+2
B
+1
C
A
-3
D
-2
E
+1
Total Cost: -1
B
C
D
E
A
-1
Total Cost: +1
D
A
B
C
E
Total Cost: -3
A
B
C
E
D
Total Cost: 0

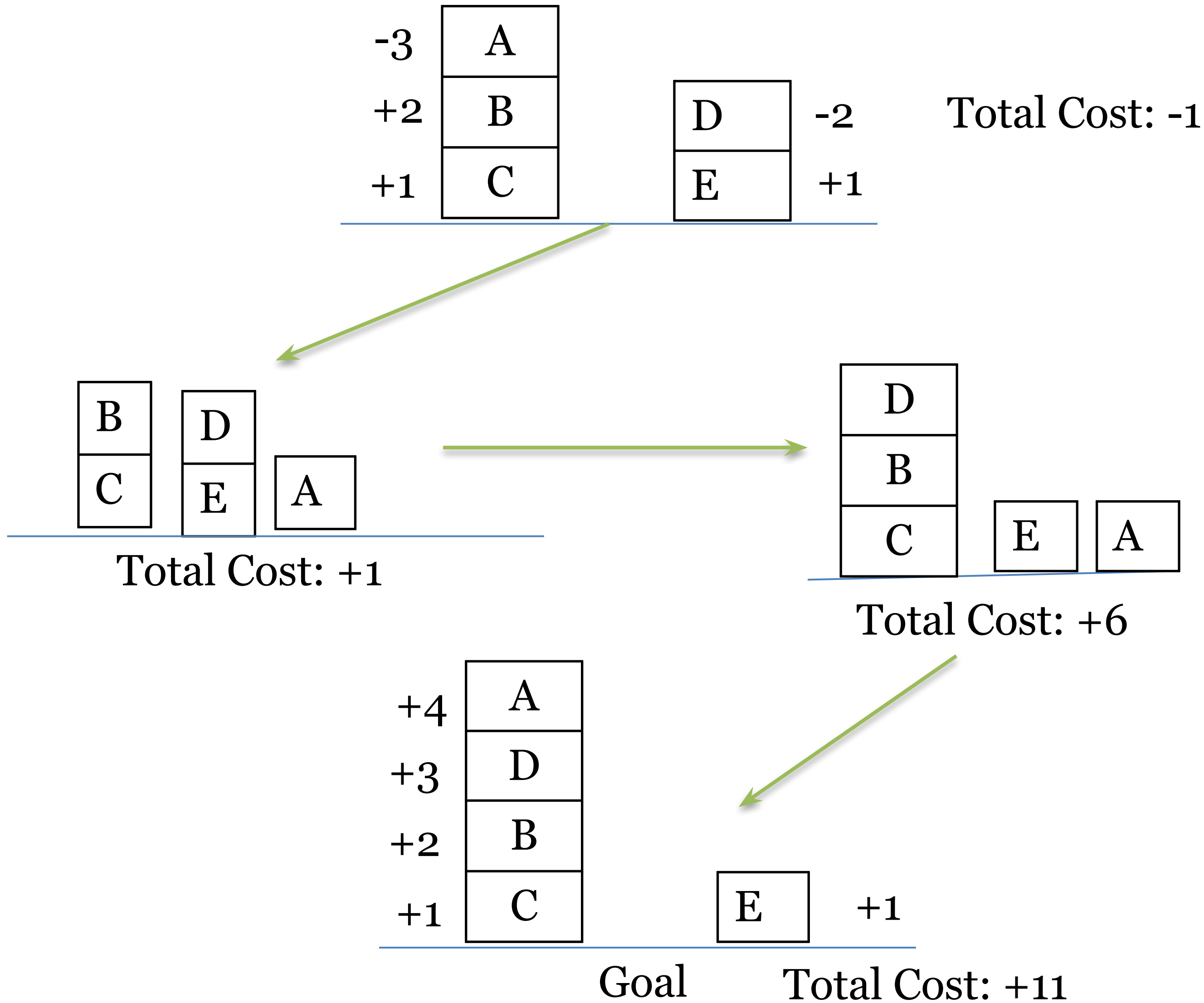


-3 A
+2 B
+1 C
D -2
E +1
Start
+4 A
+3 D
+2 B
+1 C
E +1
Goal
+2 B
+1 C
A -3
D -2
E +1
Total Cost: -1
B
C
D
E
A -1
Total Cost: +1
D
A
B
C
E
Total Cost: -3
A
B
C
E
D
Total Cost: 0

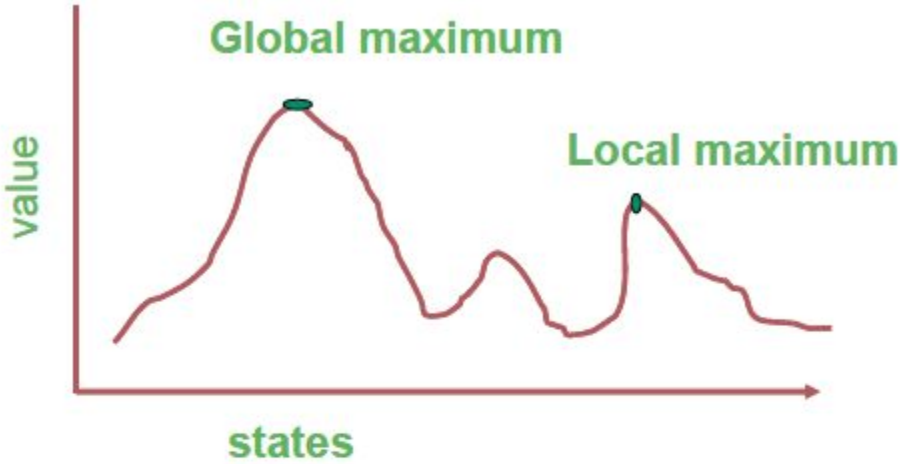

B
C
D
E
A
Total Cost: +1
B
D
E
C
A
Total Cost: ?
D
E
C
B
A
Total Cost: ?
B
C
D
E
A
Total Cost: ?
B
C
E
A
D
Total Cost: ?
B
C
E
D
A
Total Cost: ?
D
B
C
E
A
Total Cost: ?


B
D
C
E
A
Total Cost: +1
B
D
E
C
A
Total Cost: -4
D
B
C
E
A
Total Cost: -3
B
C
D
E
A
Total Cost: -2
B
C
E
A
D
Total Cost: +2
B
C
E
D
A
Total Cost: +1
D
B
C
E
A
Total Cost: +6

Final Solution



Local Maxima



Hill Climbing (Cont…)

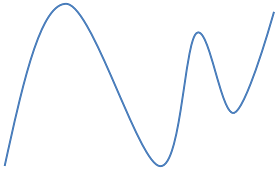
Local maxima
Once the top of a hill is reached the algorithm will halt since every possible step leads down.

Plateaux
If the landscape is flat, meaning many states have the same goodness, algorithm degenerates to a random walk.

Ridges
If the landscape contains ridges, local improvements may follow a zigzag path up the ridge, slowing down the search.

Surface

Surface
h1: Local Maxima



Surface
h2: Global Maxima