A* (BF*) WITH NON-ADDITIVE COST & EVALUATION FUNCTIONS

Andrey Balabokhin
Neema Kanapala

A* ALGORITHM

- Additive Evaluation function, $f = g + h$

- Additive cost function, $g = g(n') + c(n, n')$

A* always finds the shortest path to the goal node, if *h* is admissible.

TYPES OF COST FUNCTION

- Additive Cost Function

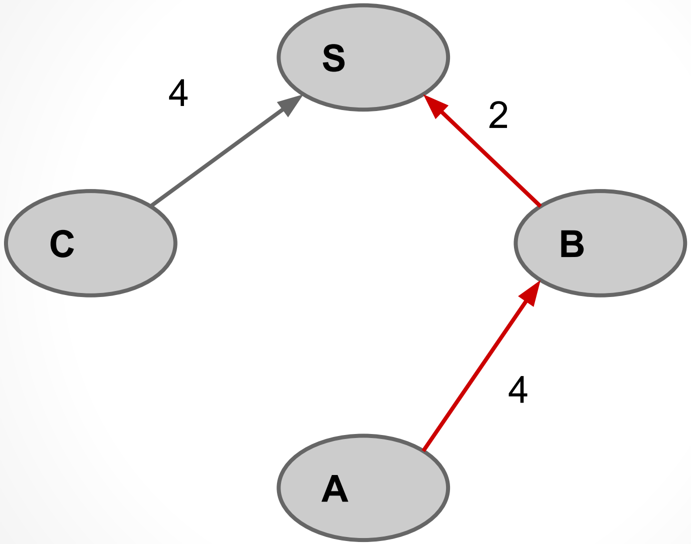
- Non-Additive Costs

COST FUNCTIONS

Additive costs	sum [$c(p_i, p_{i+1})$]
NON-ADDITIVE COSTS	
Multiplicative costs	product [$c(p_i, p_{i+1})$]
Mean	avg [$c(p_i, p_{i+1})$]
Median	middle value in a sorted list of costs
Mode	most frequent [$c(p_i, p_{i+1})$]
Range	max [$c(p_i, p_{i+1})$] - min [$c(p_i, p_{i+1})$]
Max-cost	max [$c(p_i, p_{i+1})$]
Last edge cost	$c(p_{n-1}, p_n)$

RANGE

$$g(P) = \max [c(p_i, p_{i+1})] - \min [c(p_i, p_{i+1})]$$

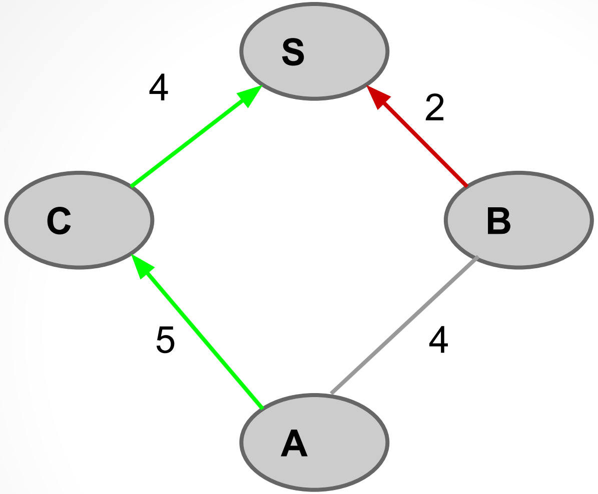


$g(P_1) = 4\text{-}2 = 2$

P_1 - (A, B, S)

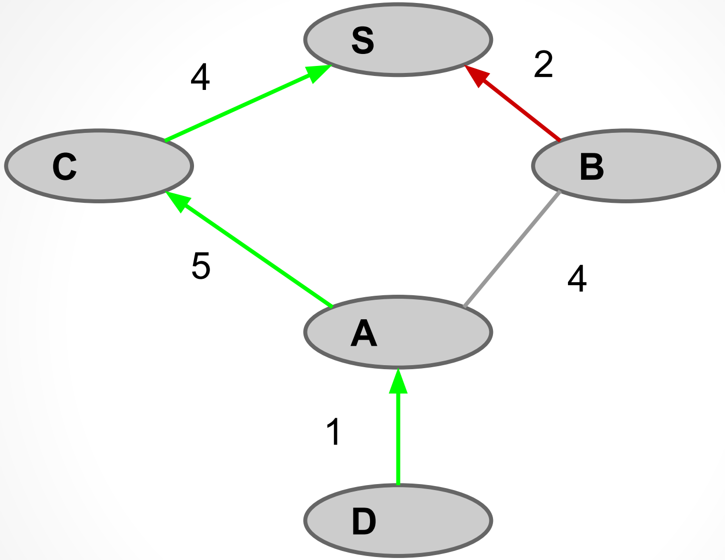
Arrows indicate the parent nodes

RANGE



$g(P_1) = 5-4 = \mathbf{1}$	$g(P_2) = 4-2 = \mathbf{2}$

RANGE



g(P_2) = 4-1 = **3**

P_2 - (D, A, C, S)

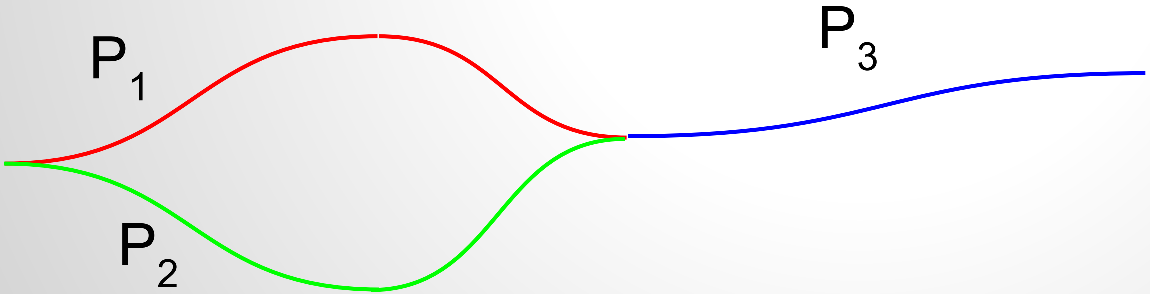
g(P_1) = 5-1 = **4**

P_1 - (D, A, B, S)

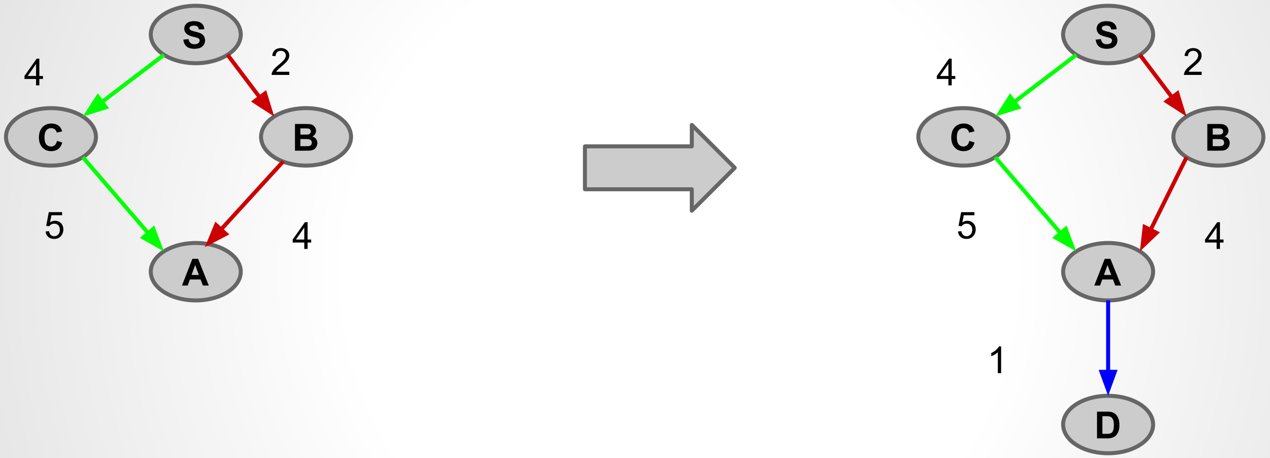
but **4 > 3 !**

ORDER PRESERVING COST ORDER FUNCTION

Let P_1 and P_2 be any two paths from "s" node to "n" node and P_1P_3 and P_2P_3 paths after concatenation of a new path P_3, if $g(P_1) >= g(P_2) \Rightarrow g(P_1P_3) >= g(P_2P_3)$, g is said to be order preserving.

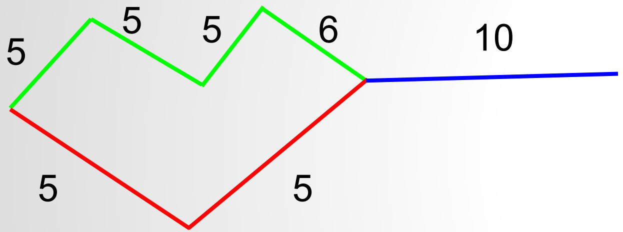


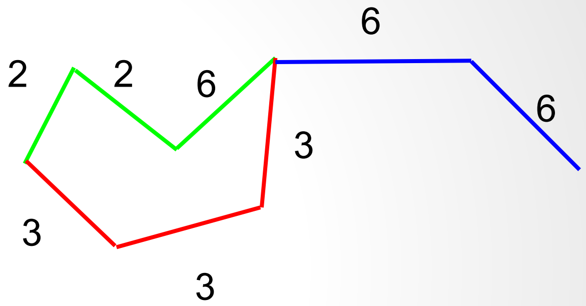
RANGE IS NON-ORDER PRESERVING



$g(P_1) = 4-2 = 2$ $g(P_2) = 5-4 = 1$ $g(P_1) > g(P_2)$	$g(P_1P_3) = 4-1 = 3$ $g(P_2P_3) = 5-1 = 4$ $g(P_1P_3) < g(P_2P_3)$

Order-preserving property for the mean and mode cost functions

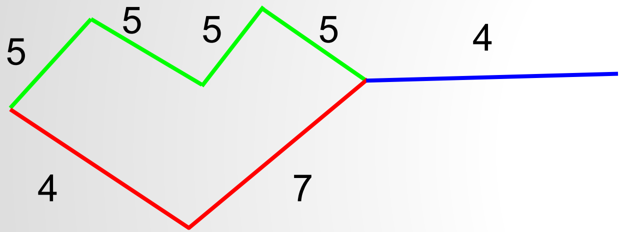


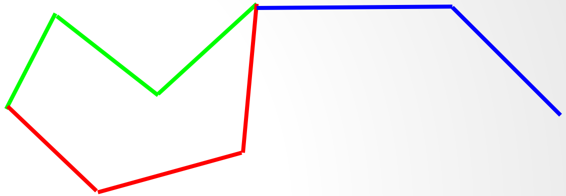


Legend:

g(P) - Mean edge cost	g(P) - Mode (most frequent $[c(p_i, p_{i+1})]$)
red line - P_1, $g(P_1) = 5$	red line - P_1, $g(P_1) = 3$
green line - P_2, $g(P_2) = 5.25$	green line - P_2, $g(P_2) = 2$
red line, blue line - $g(P_1P_3) = 6.7$	blue line - P_3, $g(P_1P_3) = 3$, $g(P_2P_3) = 6$
green line, blue line - $g(P_2P_3) = 6.2$	
$g(P_1) < g(P_2)$ but $g(P_1P_3) > g(P_2P_3)$	$g(P_1) > g(P_2)$ but $g(P_1P_3) < g(P_2P_3)$

Order-preserving property for the median and max cost functions





Legend:

g(P) - Median edge cost	g(P) - Max
- P_1, $g(P_1) = 5.5$	- P_1, - P_2, - P_3
- P_2, $g(P_2) = 5$	Let $g(P_1) >= g(P_2)$
- $g(P_1P_3) = 4$	if $g(P_3) >= g(P_1) => g(P_1P_3) = g(P_2P_3) = g(P_3)$
- $g(P_2P_3) = 5$	if $g(P_3) < g(P_1) =>$
$g(P_1) > g(P_2)$ but $g(P_1P_3) < g(P_2P_3)$	$g(P_1P_3) = g(P_1) >= \max(g(P_2), g(P_3)) = g(P_2P_3)$

If P is a path with nodes p_1, p_2, …, p_{n-1}, p_n and $c(p_i, p_{i+1})$ is an cost value for the edge between nodes p_i and p_{i+1}. Cost measure function g(P) can be any function:

Cost Function		Is g(P) order-preserving?
Additive costs	sum $[c(p_i, p_{i+1})]$	Yes
Multiplicative costs	product $[c(p_i, p_{i+1})]$	Yes when $c(p_i, p_{i+1})$ is positive
Mean	avg $[c(p_i, p_{i+1})]$	No
Median	middle value in a sorted list of costs	No
Mode	most frequent $[c(p_i, p_{i+1})]$	No
Range	max $[c(p_i, p_{i+1})]$ - min $[c(p_i, p_{i+1})]$	No
Max-cost	max $[c(p_i, p_{i+1})]$	Yes
Last edge cost	$c(p_{n-1}, p_n)$	Yes

BEST FIRST* (BF*) ALGORITHM

- “ *f* ” is an arbitrary function

eg: $f = \text{sqrt}[g\text{^}2 + h\text{^}2]$

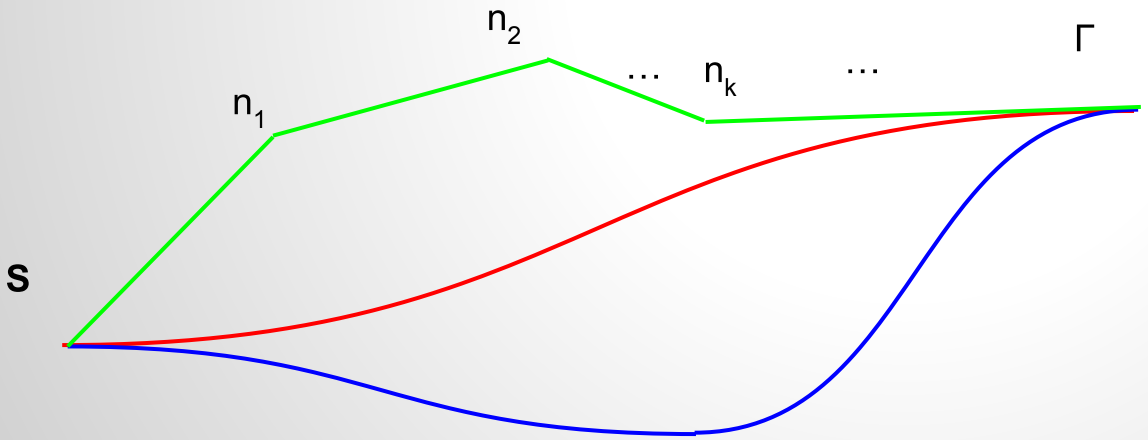
$f = \max(g, h)$

- Builds path along the least values of $f(P_i)$

- A* is a special case of BF*

Admissibility of BF*

If in every graph searched by BF* there exists at least one optimal solution path along which f attains its maximum value on the goal node, then, then BF* is admissible.



- optimal path P_1

- not optimal path P_2

- not optimal path P_3

$g(P_1) <= g(P_2)$, $g(P_1) <= g(P_3)$

$n_1, n_2, \ldots, n_k, \ldots, \Gamma$ - nodes along P_1

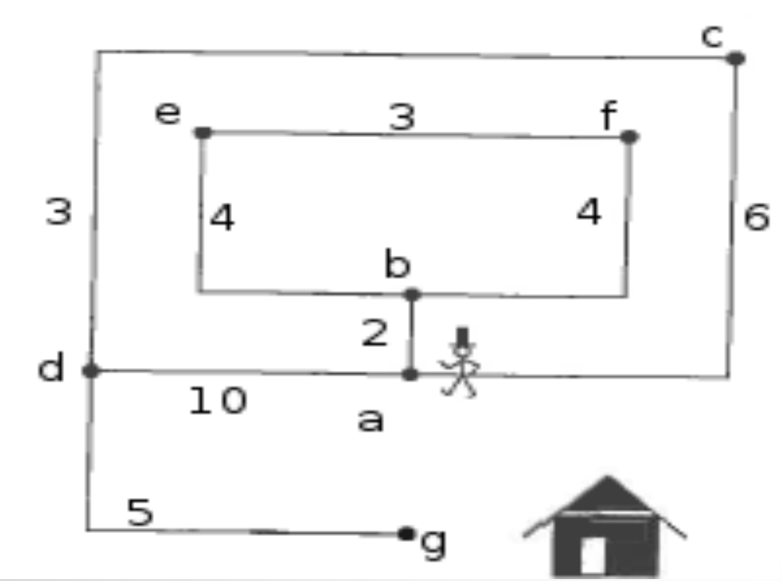
BF* is admissible if **$f(\Gamma) = \max(f(n_i))$** along optimal path P_1

BF* ALGORITHM REQUIREMENT

- **“ f ” must be** Order-preserving for all the paths from start node “s” to goal node “Γ”.

- **“ f ” must be** admissible.

BF* EXAMPLE WITH MAX-COST FUNCTION



$g(P_i) = \max [c(p_i, p_{i+1})]$

$f(P_i) = g(P_i)$

h - trivial

f - order-preserving

f - admissible, as it is not decreasing with increasing number of nodes

=> f(goal node) = $\max(f_i)$

THANK YOU !

QUESTIONS ?

References

1. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving (The Addison-Wesley series in artificial intelligence) 1984; Section 3.3 ("Some Extensions to Nonadditive Evaluation Functions")