Travelling Salesman Problem

-Amisha Gupta, Aarjav Hansoti, Varun Khadayate, Ankit Jha, Rugved Kulkarni

What is the TSP?

The Travelling Salesman Problem (TSP) is an optimization problem which can be imagined as a salesman who wishes to visit several cities but wants to do so in a way that his total distance travelled is minimum and he ends up back where he started i.e. a closed loop. The complexity of TSP increases as the number of cities are increased. It is classifies as an NP-Hard problem.

History

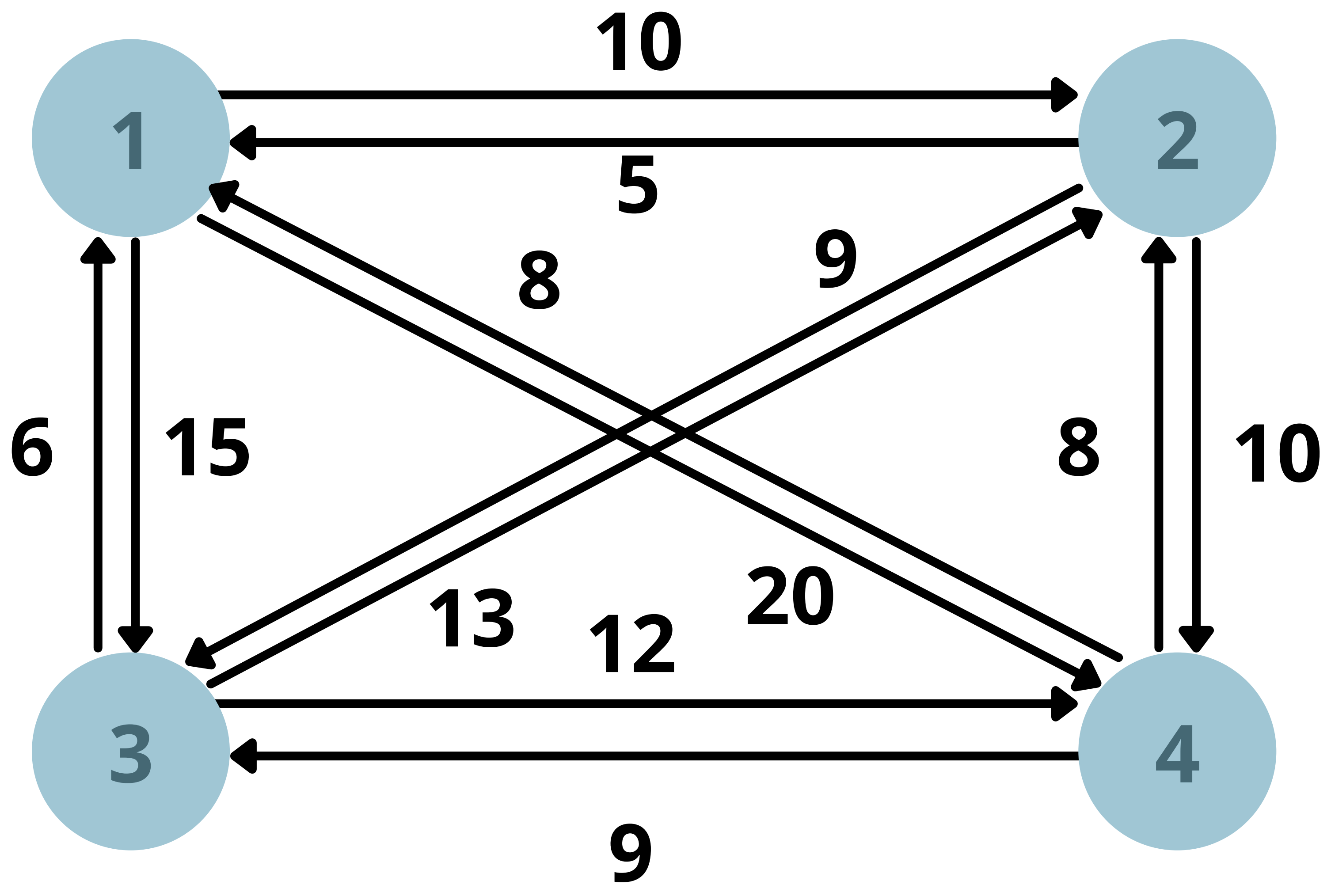
The traveling salesman problem was mathematically formulated by Irish mathematician W.R. Hamilton and by the British mathematician Thomas Kirkman. The general form of the TSP appears to have been first studied by mathematicians during the 1930s in Vienna by Karl Menger. Menger called it the "Messenger Problem" the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points. Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points are not known. The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general, does not yield the shortest route.

Greedy Approach VS Dynamic Approach

- DP is an exact algorithm, at least as it is usually used. There are DP algorithms for TSP. Thus, these algorithms will solve the problem exactly.

- The greedy approach doesn't always give the optimal solution for the travelling salesman problem. Greedy heuristics are sometimes used for solving TSPs. As the number of vertices grows, the run time of those heuristics grows too, but it does not grow exponentially.

Example 1



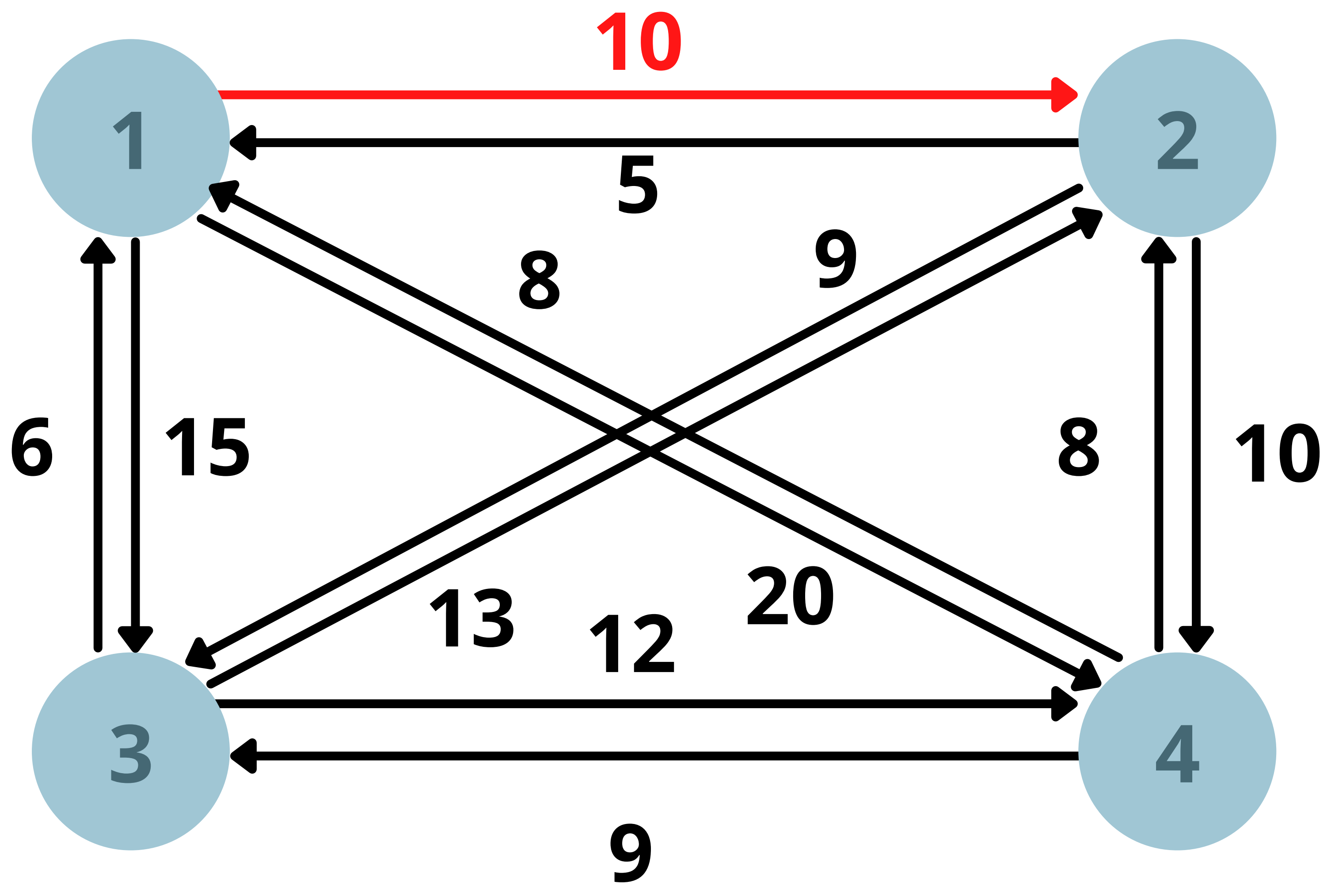
Formula:

C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

Algorithm:

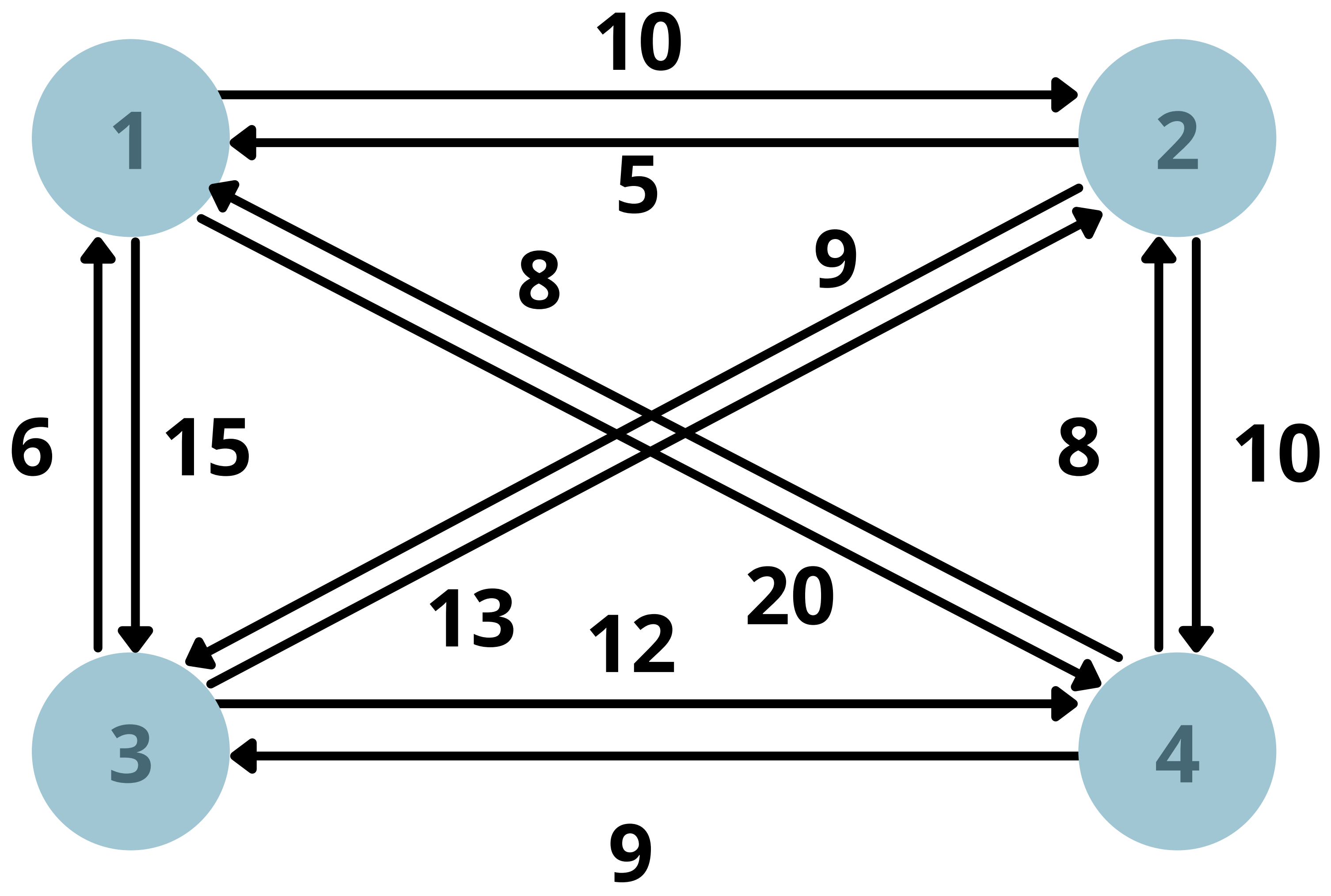
C ({1}, 1) = 0
for s = 2 to n do
  for all subsets S Є {1, 2, 3, … , n} of size s and containing 1
    C (S, 1) = ∞
  for all j Є S and j ≠ 1
    C (S, j) = min {C (S – {j}, i) + d(i, j) for i Є S and i ≠ j}
Return minj C ({1, 2, 3, …, n}, j) + d(j, i)

Example 1



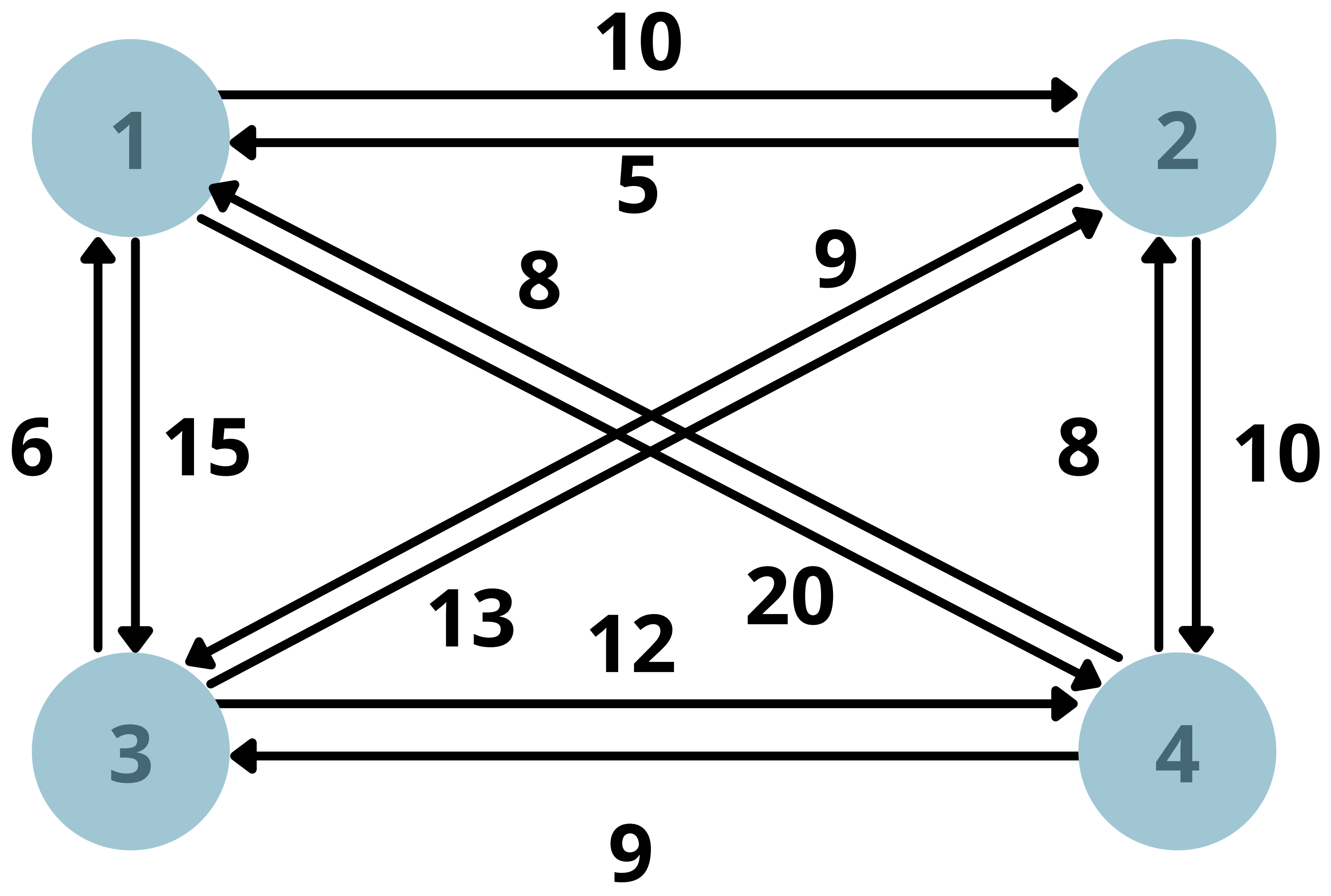
	1	2	3	4
1	0	10		
2		0		
3			0	
4				0

Example 1



	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Example 1



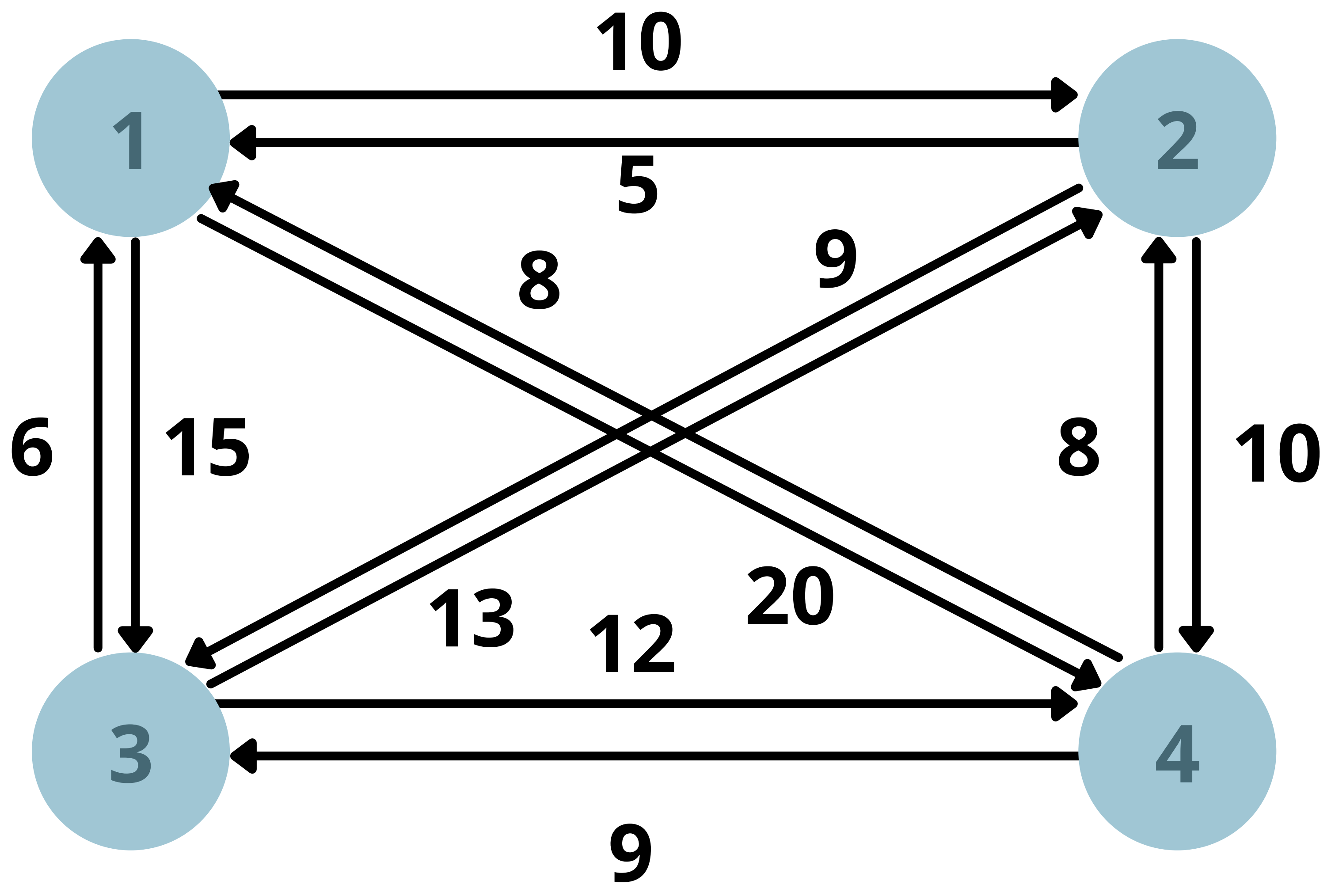
Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = Φ

Cost(2,Φ,1) = d(2,1) = 5

Example 1



Formula:

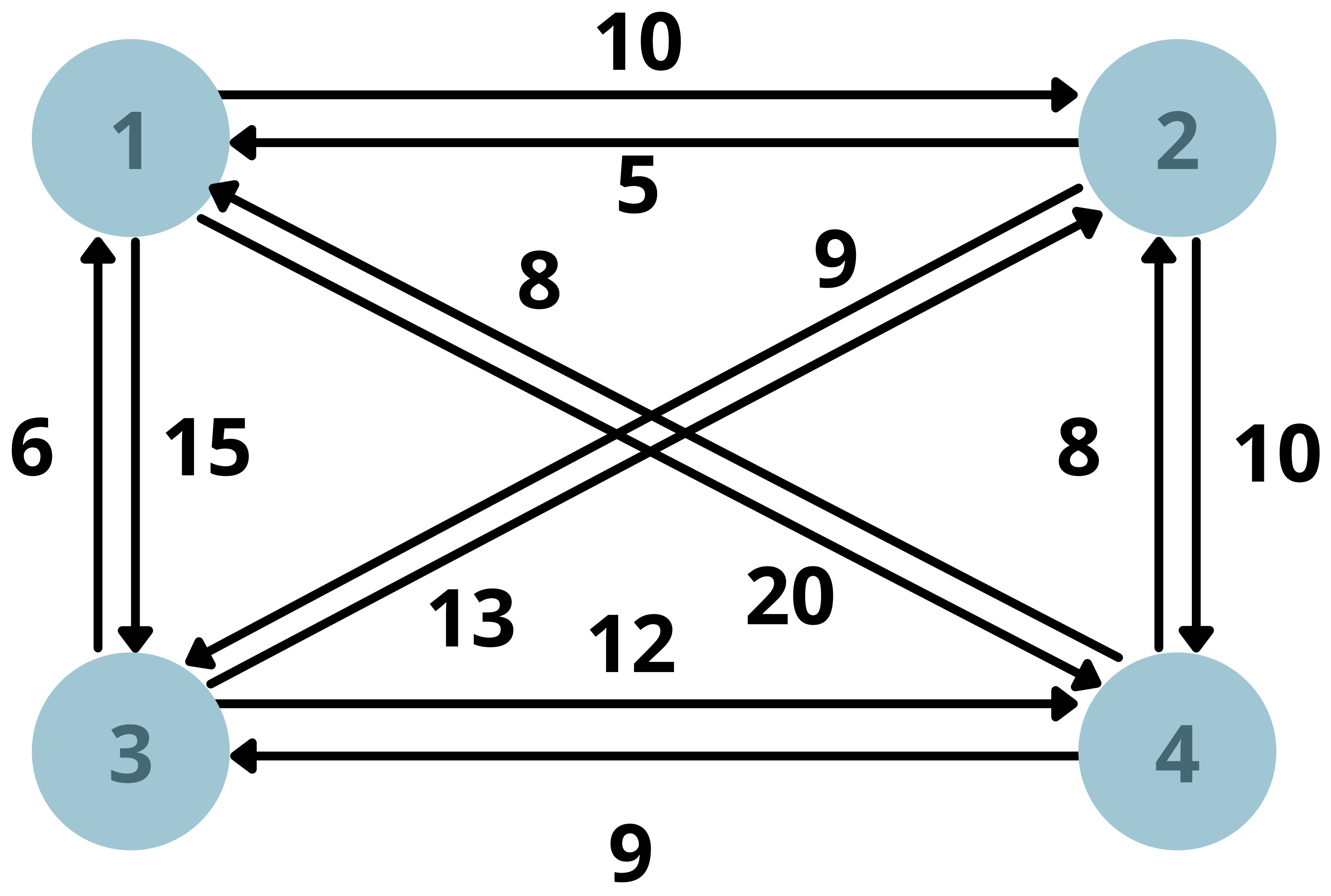
C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = Φ

Cost(2,Φ,1) = d(2,1) = 5

Cost(3,Φ,1) = d(3,1) = 6

Example 1



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

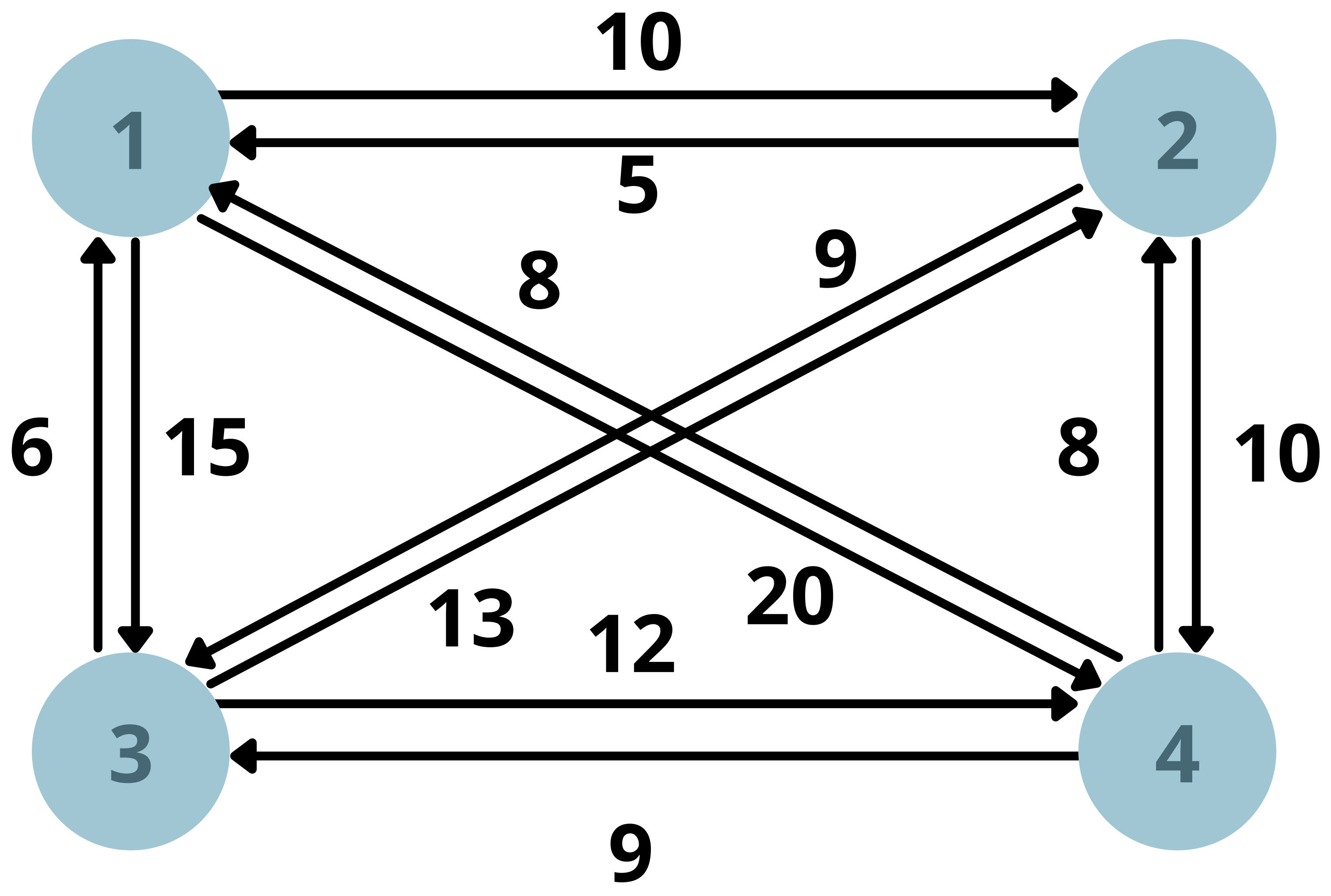
S = Φ

Cost(2,Φ,1) = d(2,1) = 5

Cost(3,Φ,1) = d(3,1) = 6

Cost(4,Φ,1) = d(4,1) = 8

Example 1



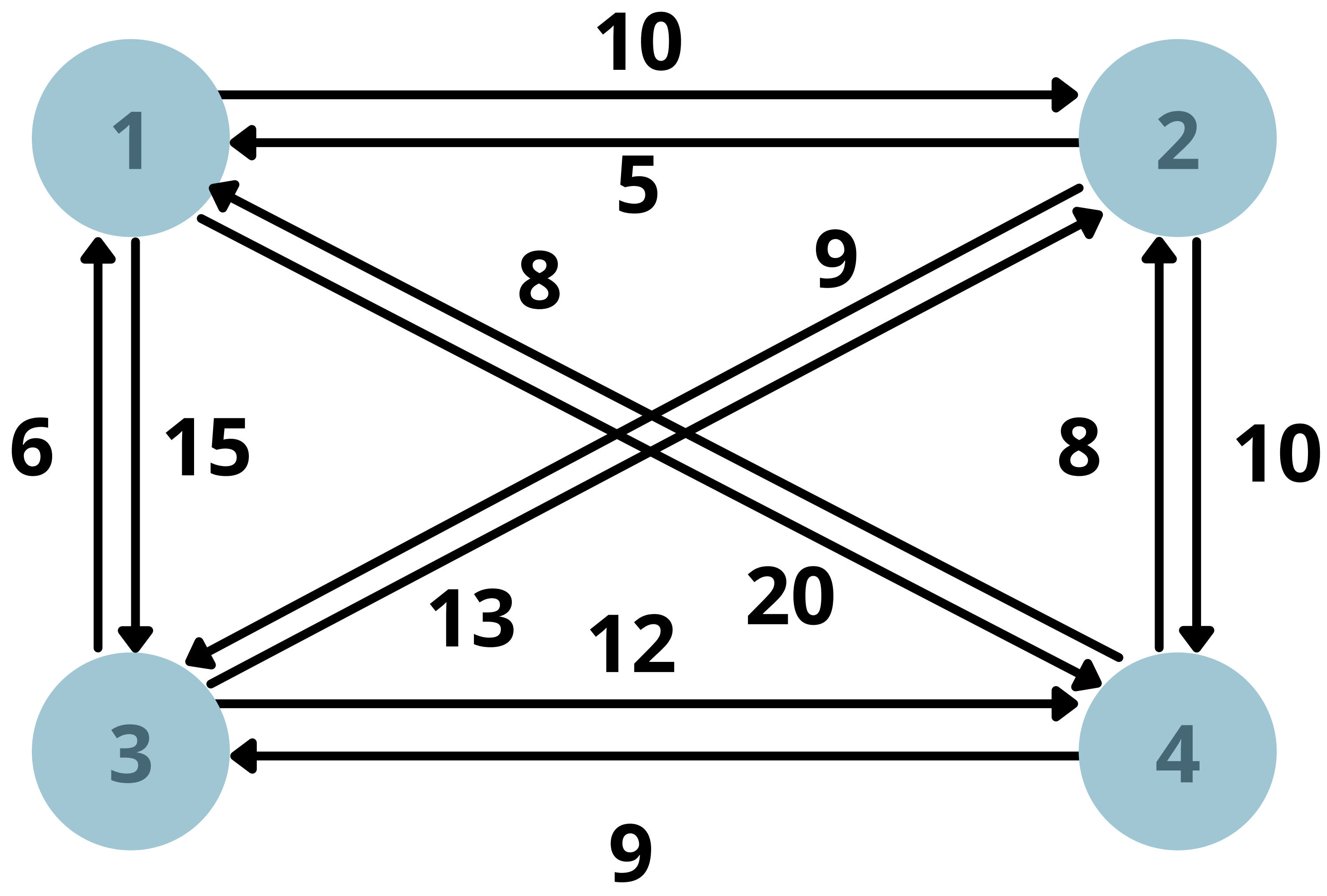
Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 1

Cost(2,{3},1) = d[2,3] + Cost(3,Φ,1) = 9 + 6 = 15

Example 1



Formula:

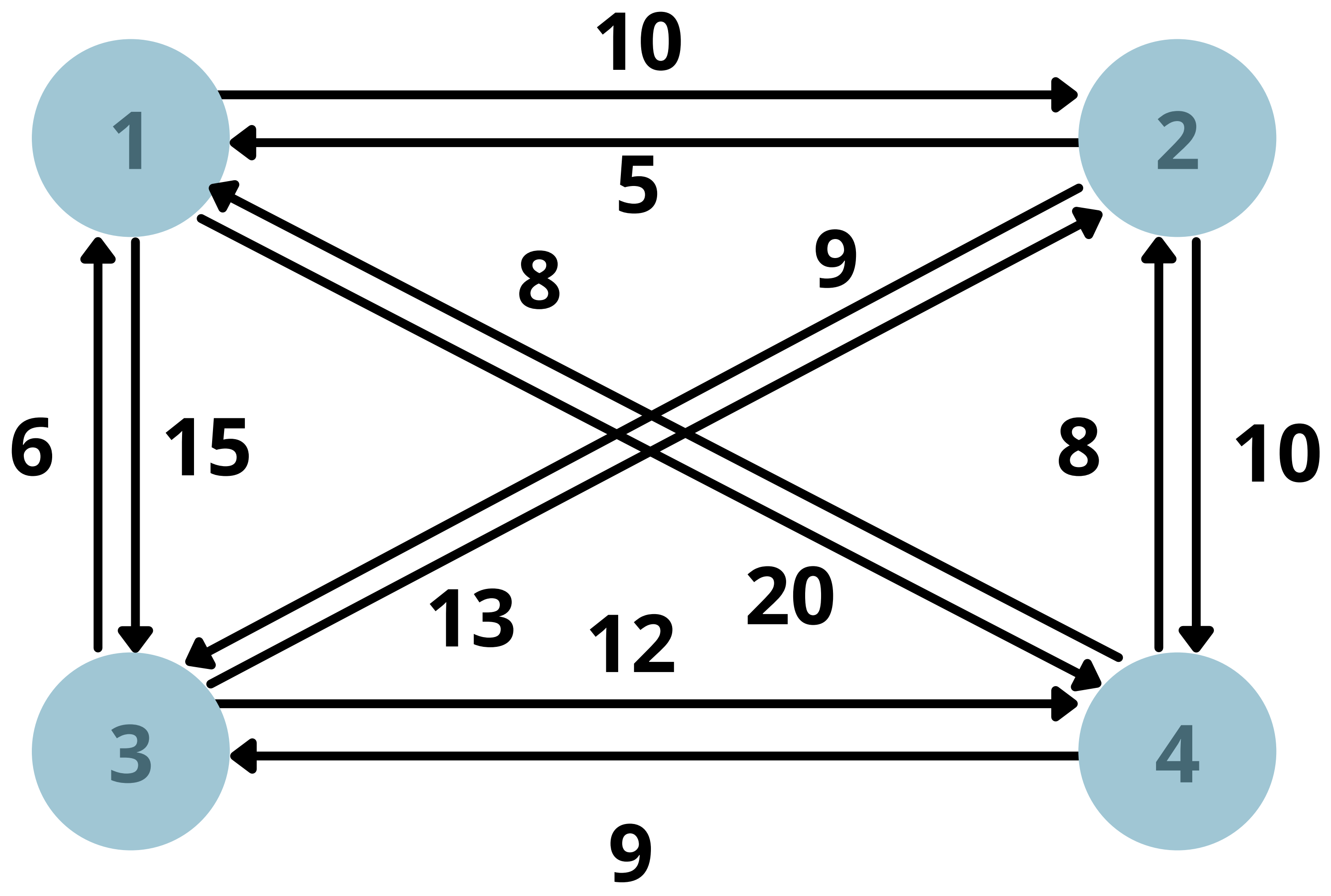
C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 1

Cost(2,{3},1) = d[2,3] + Cost(3,Φ,1) = 9 + 6 = 15

Cost(2,{4},1) = d[2,4] + Cost(4,Φ,1) = 10 + 8 = 18

Example 1



Formula:

C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 1

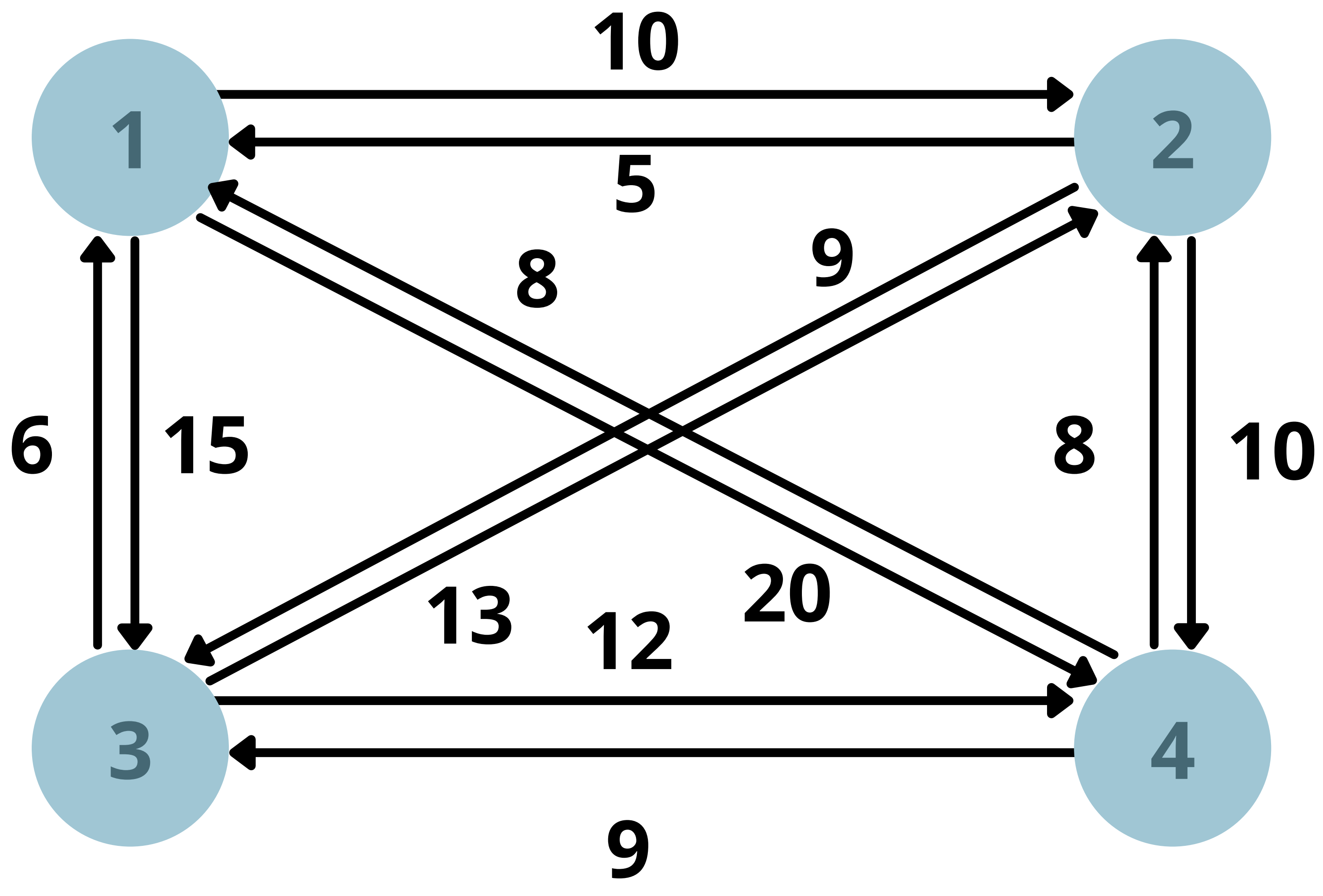
Cost(2,{3},1) = d[2,3] + Cost(3,Φ,1) = 9 + 6 = 15

Cost(2,{4},1) = d[2,4] + Cost(4,Φ,1) = 10 + 8 = 18

Cost(3,{2},1) = d[3,2] + Cost(2,Φ,1) = 13 + 5 = 18

Cost(3,{4},1) = d[3,4] + Cost(4,Φ,1) = 12 + 8 = 20

Example 1



Formula:

$C(j, S, i) = \min C(S - \{j\}, i) + d(j, i)$ where $i \in S$ and $i \neq j c(S, j)$

S = 1

$Cost(2,\{3\},1) = d[2,3] + Cost(3,\Phi,1) = 9 + 6 = 15$

$Cost(2,\{4\},1) = d[2,4] + Cost(4,\Phi,1) = 10 + 8 = 18$

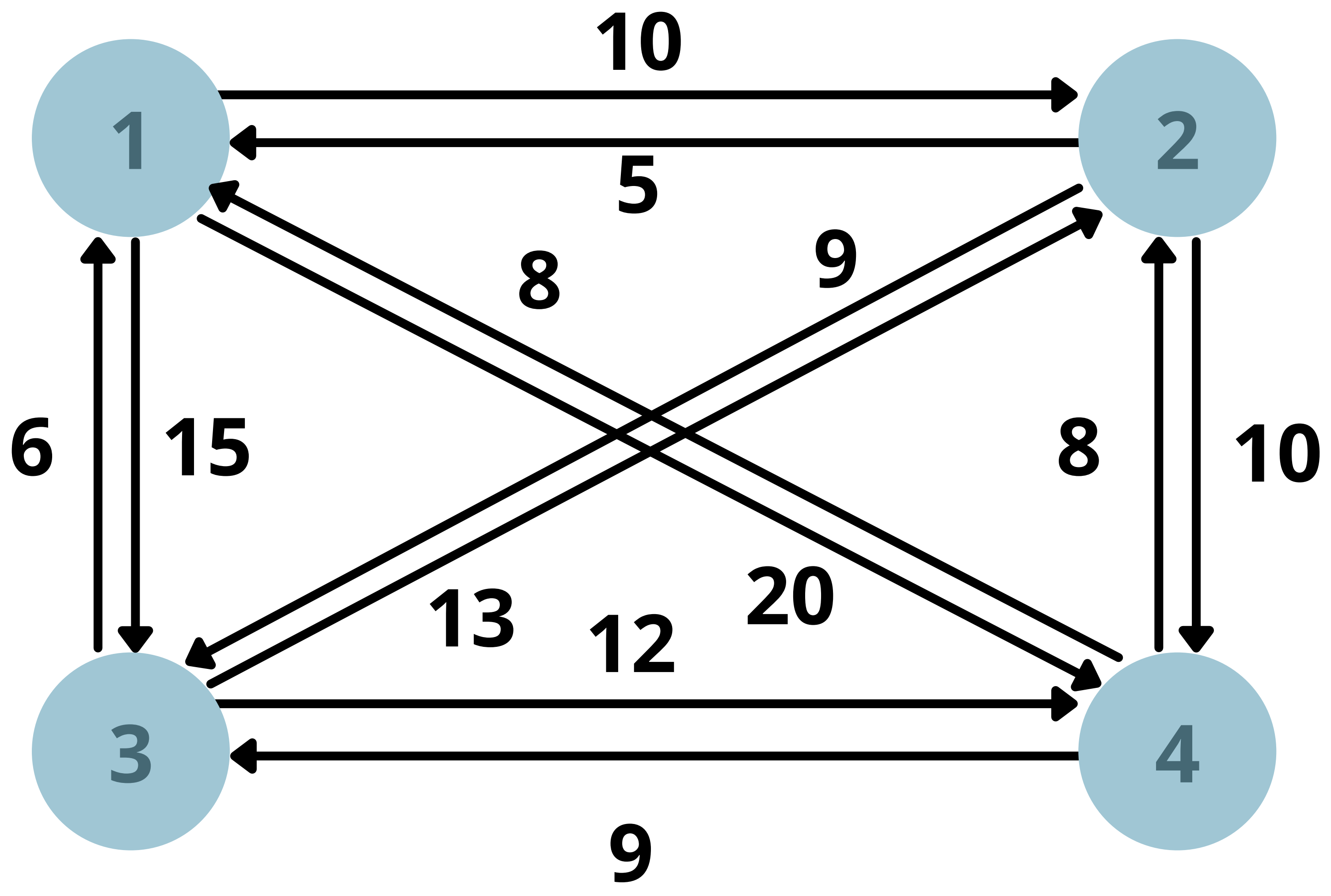
$Cost(3,\{2\},1) = d[3,2] + Cost(2,\Phi,1) = 13 + 5 = 18$

$Cost(3,\{4\},1) = d[3,4] + Cost(4,\Phi,1) = 12 + 8 = 20$

$Cost(4,\{3\},1) = d[4,3] + Cost(3,\Phi,1) = 9 + 6 = 15$

$Cost(4,\{2\},1) = d[4,2] + Cost(2,\Phi,1) = 8 + 5 = 13$

Example 1



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

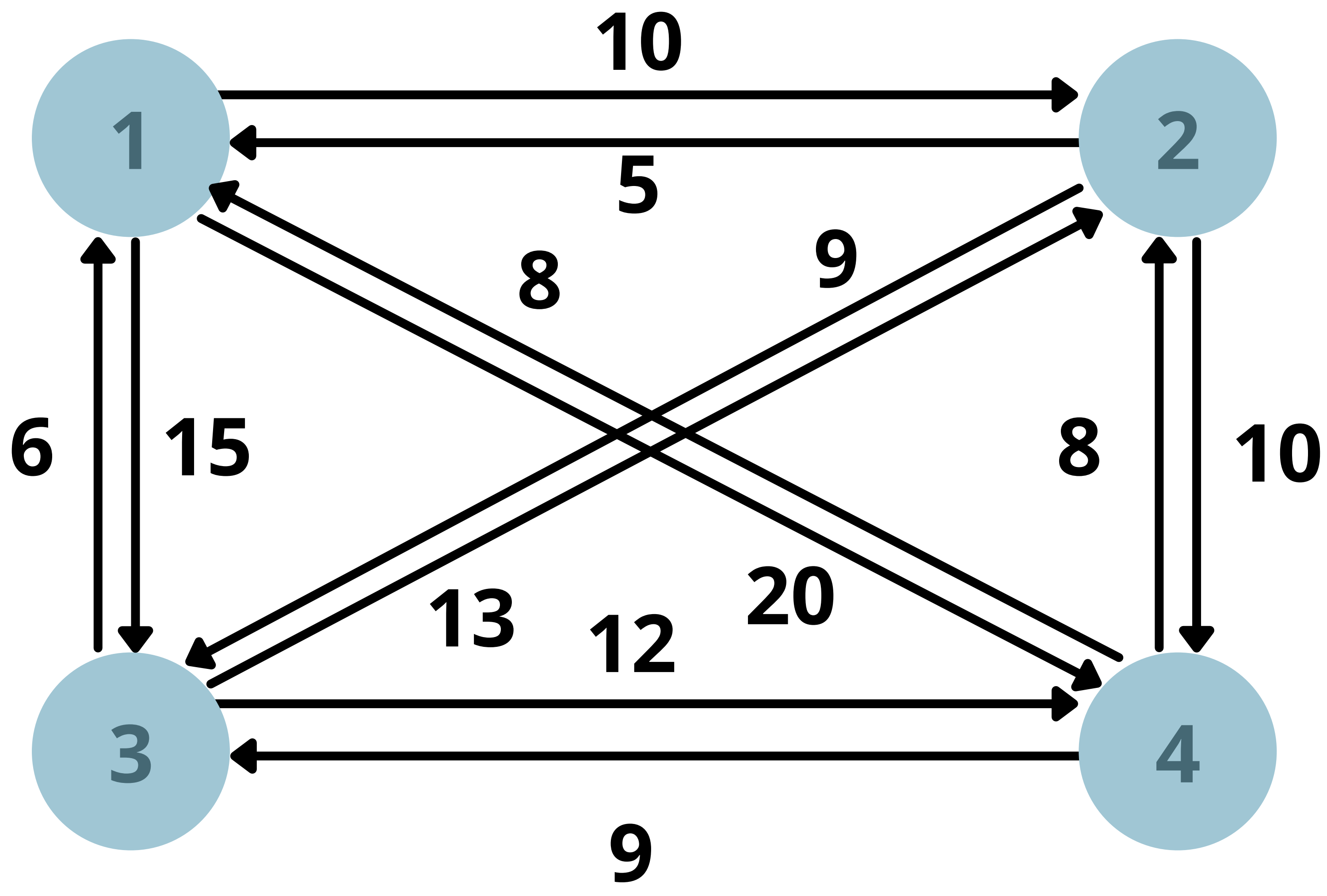
S = 2

Cost(2,{3,4},1)= min{ d[2,3]+Cost(3,{4},1)=9+20=29

d[2,4]+Cost(4,{3},1)=10+15=25=25

= 25

Example 1



Formula:

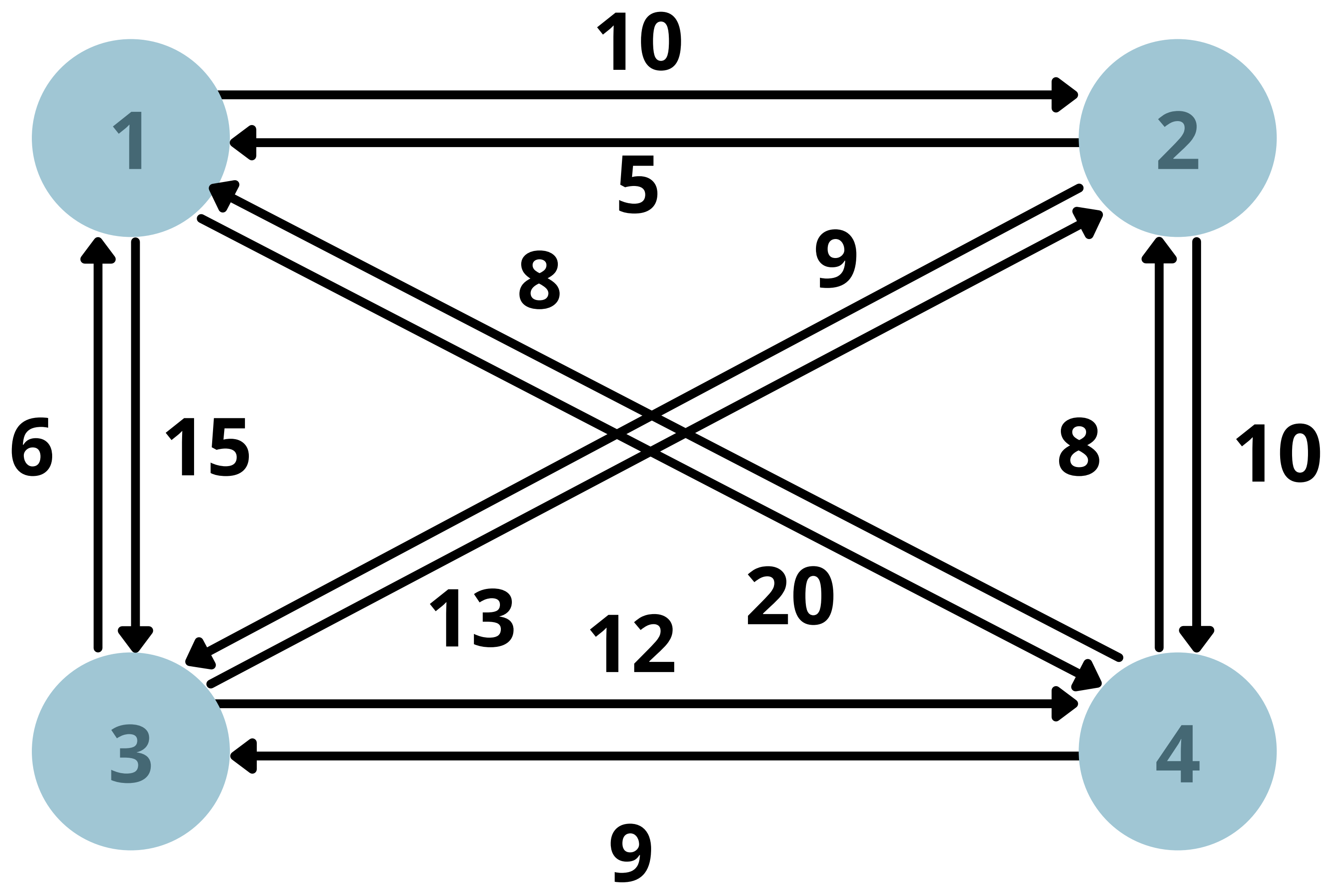
C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 2

$$\text{Cost}(3,\{2,4\},1) = \min \begin{cases} d[3,2]+\text{Cost}(2,\{4\},1)=13+18=31 \\ d[3,4]+\text{Cost}(4,\{2\},1)=12+13=25 \end{cases}$$

= 25

Example 1



Formula:

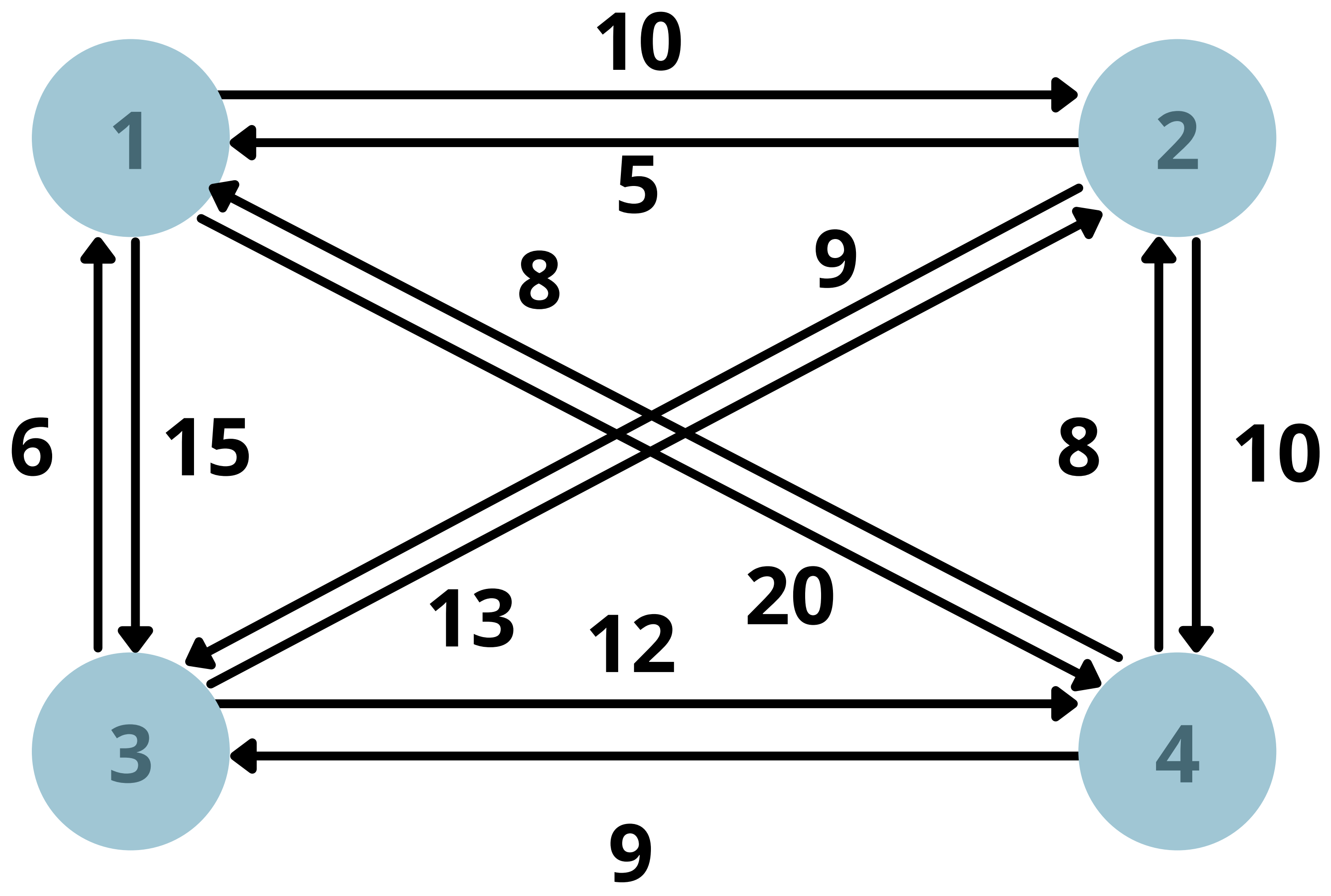
C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 2

$$\text{Cost}(4,\{2,3\},1) = \min\begin{cases} d[4,2]+\text{Cost}(2,\{3\},1)=8+15=23 \\ d[4,3]+\text{Cost}(3,\{2\},1)=9+18=27 \end{cases}$$

= 23

Example 1



Formula:

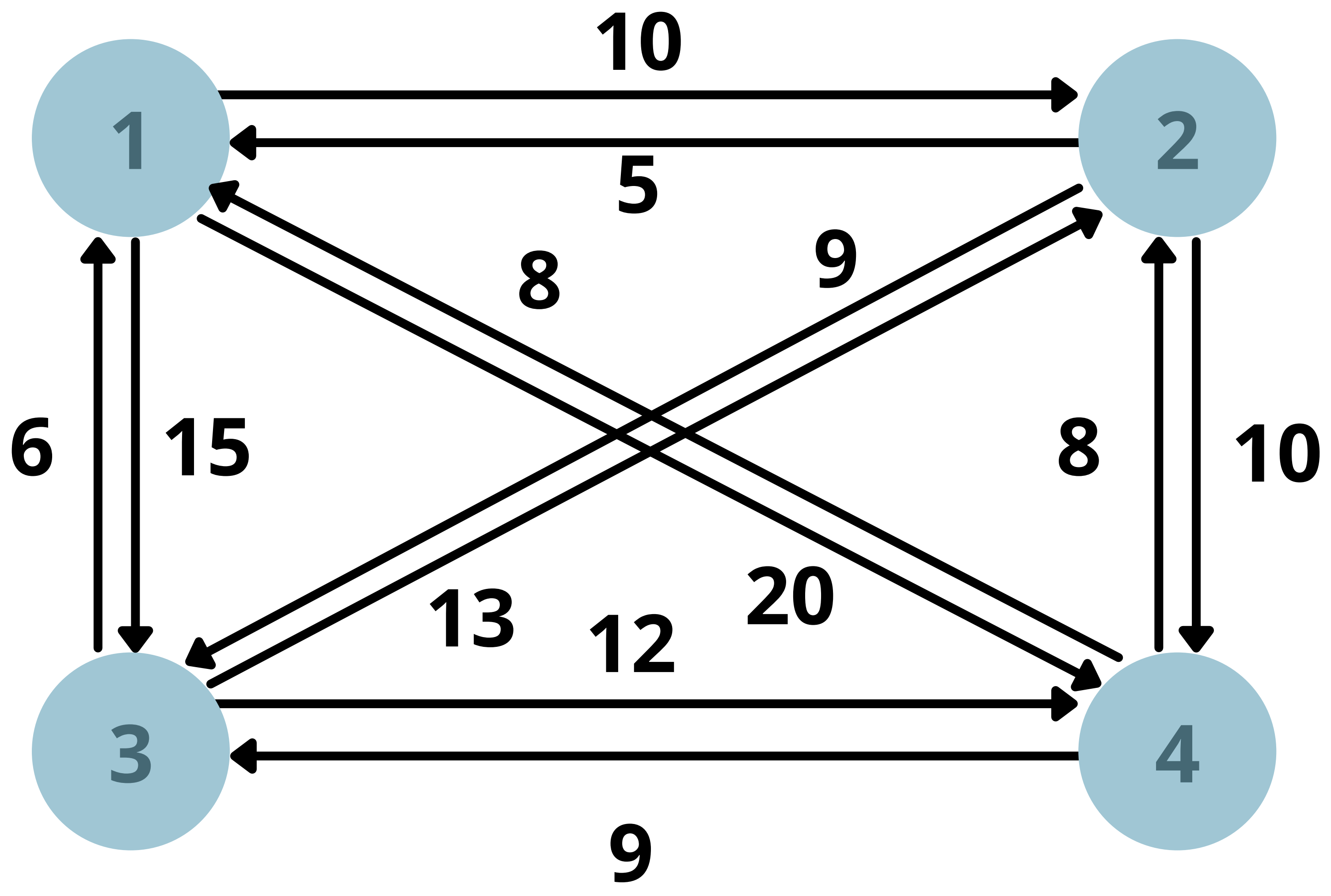
C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 3

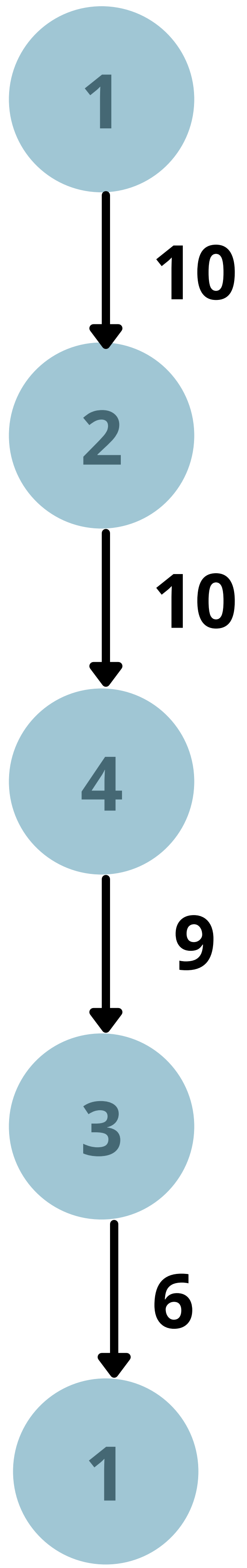
$$\text{Cost}(1,\{2,3,4\},1) = \min \begin{cases} d[1,2]+\text{Cost}(2,\{3,4\},1)=10+25=35 \\ d[1,3]+\text{Cost}(3,\{2,4\},1)=15+25=40 \\ d[1,4]+\text{Cost}(4,\{2,3\},1)=20+23=43 \end{cases}$$

= 35

Example 1



DP Method:

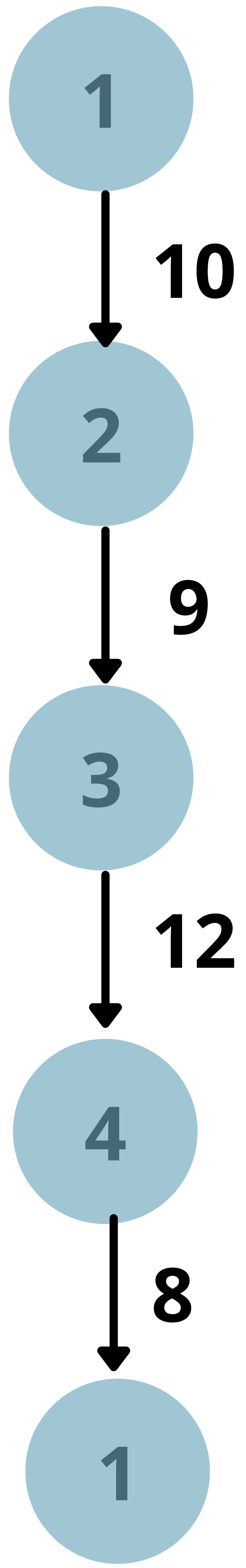


Minimum Cost Path= 35

Example 1

	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Greedy Method:

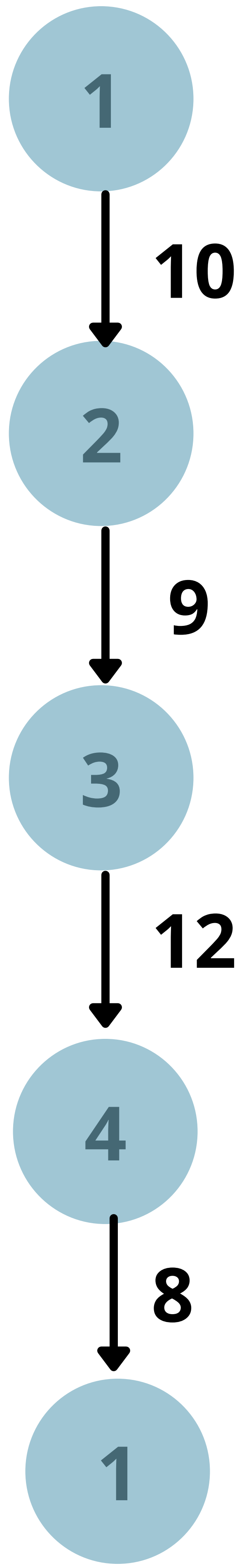


Minimum Cost Path= 39

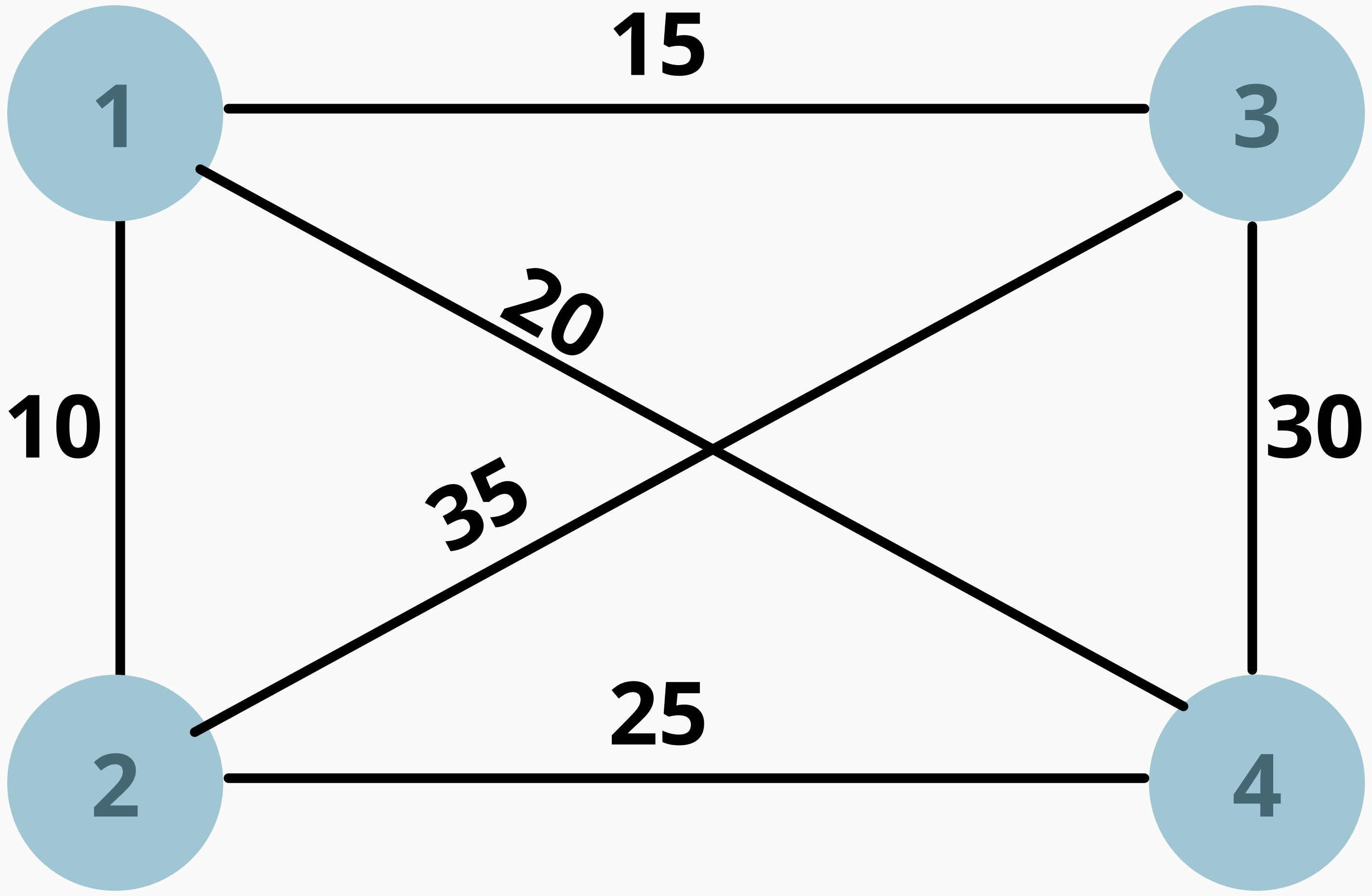
Example 1

	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Greedy Method:

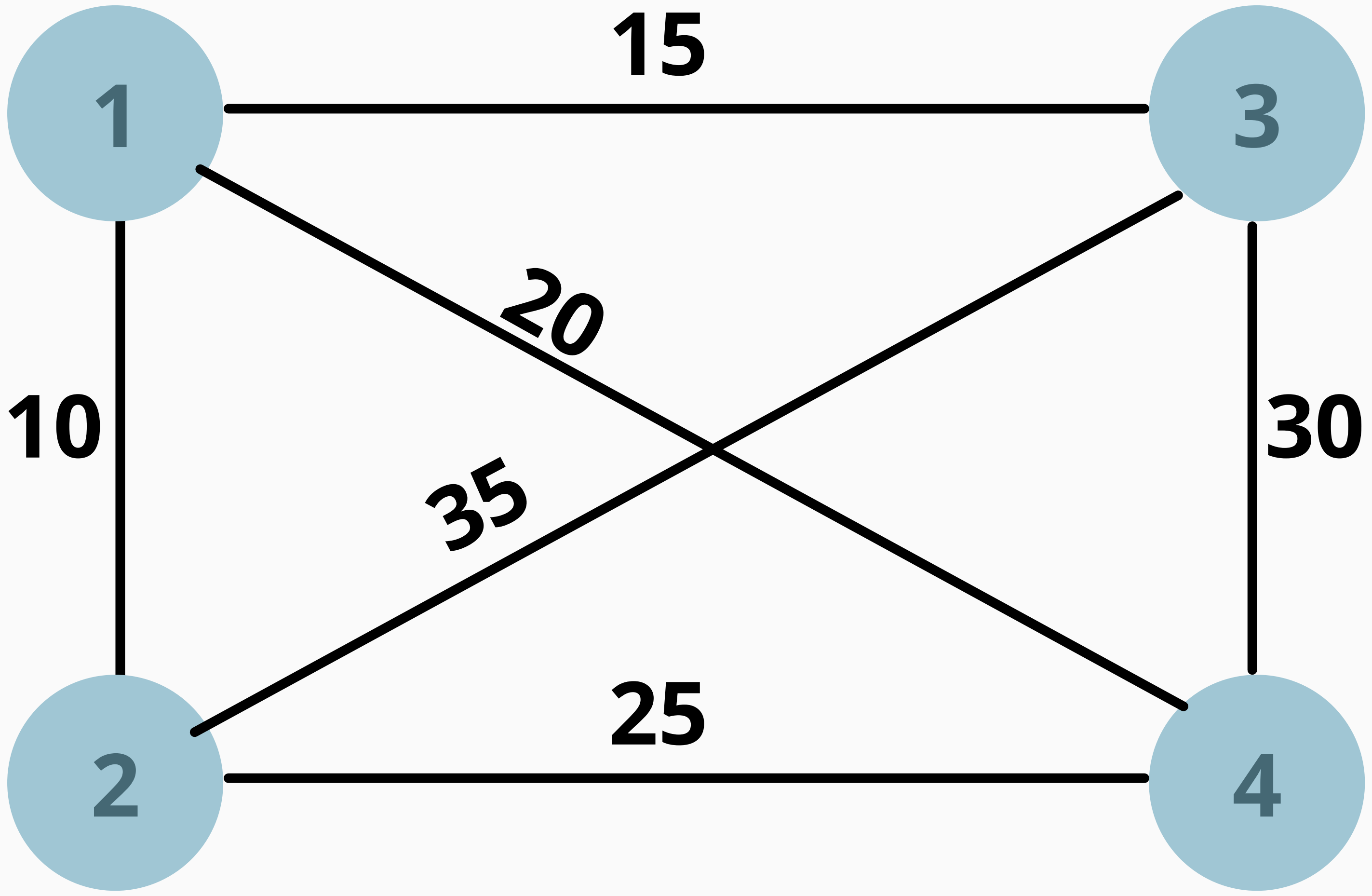


Minimum Cost Path= 39

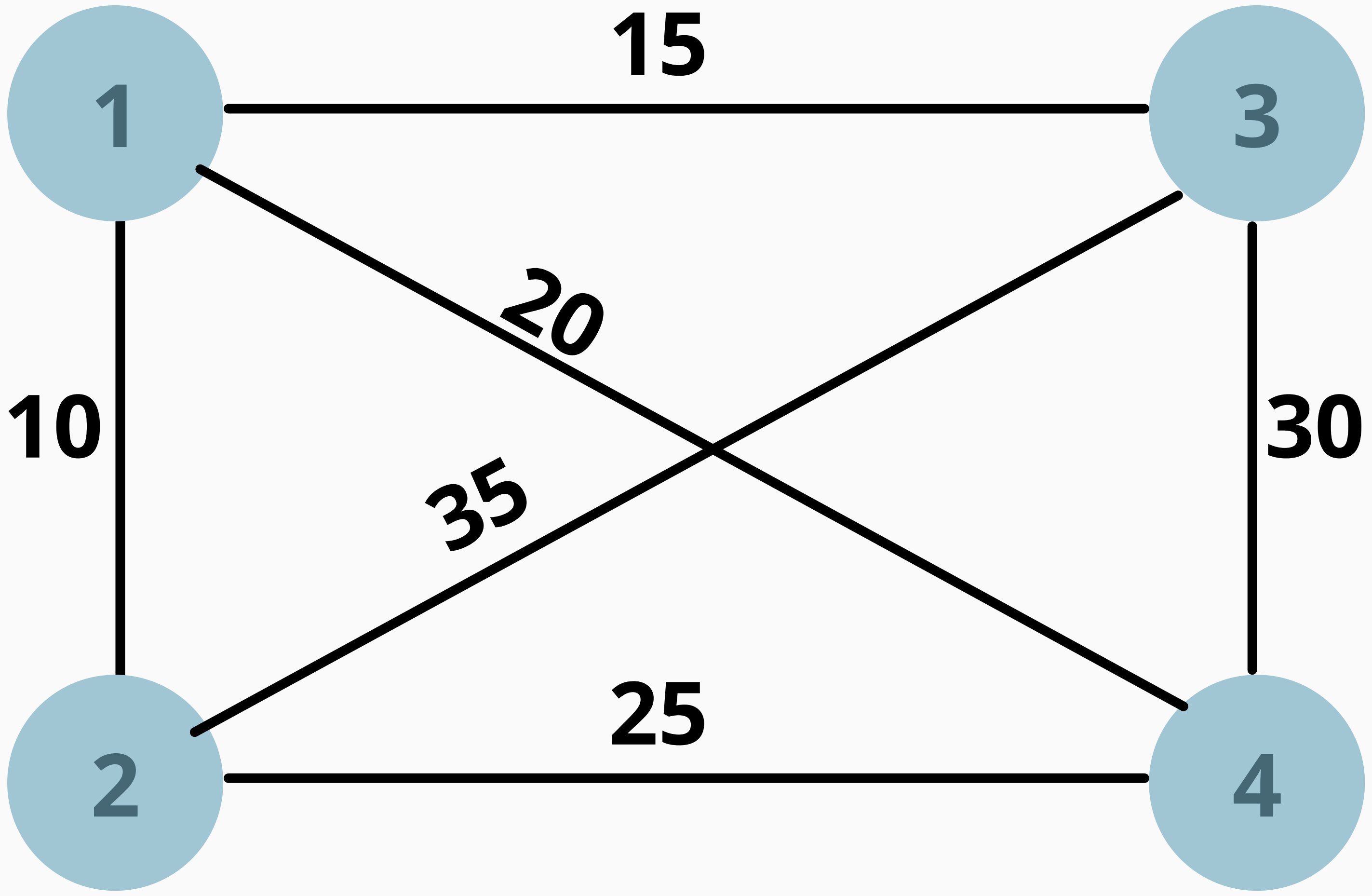


Example II

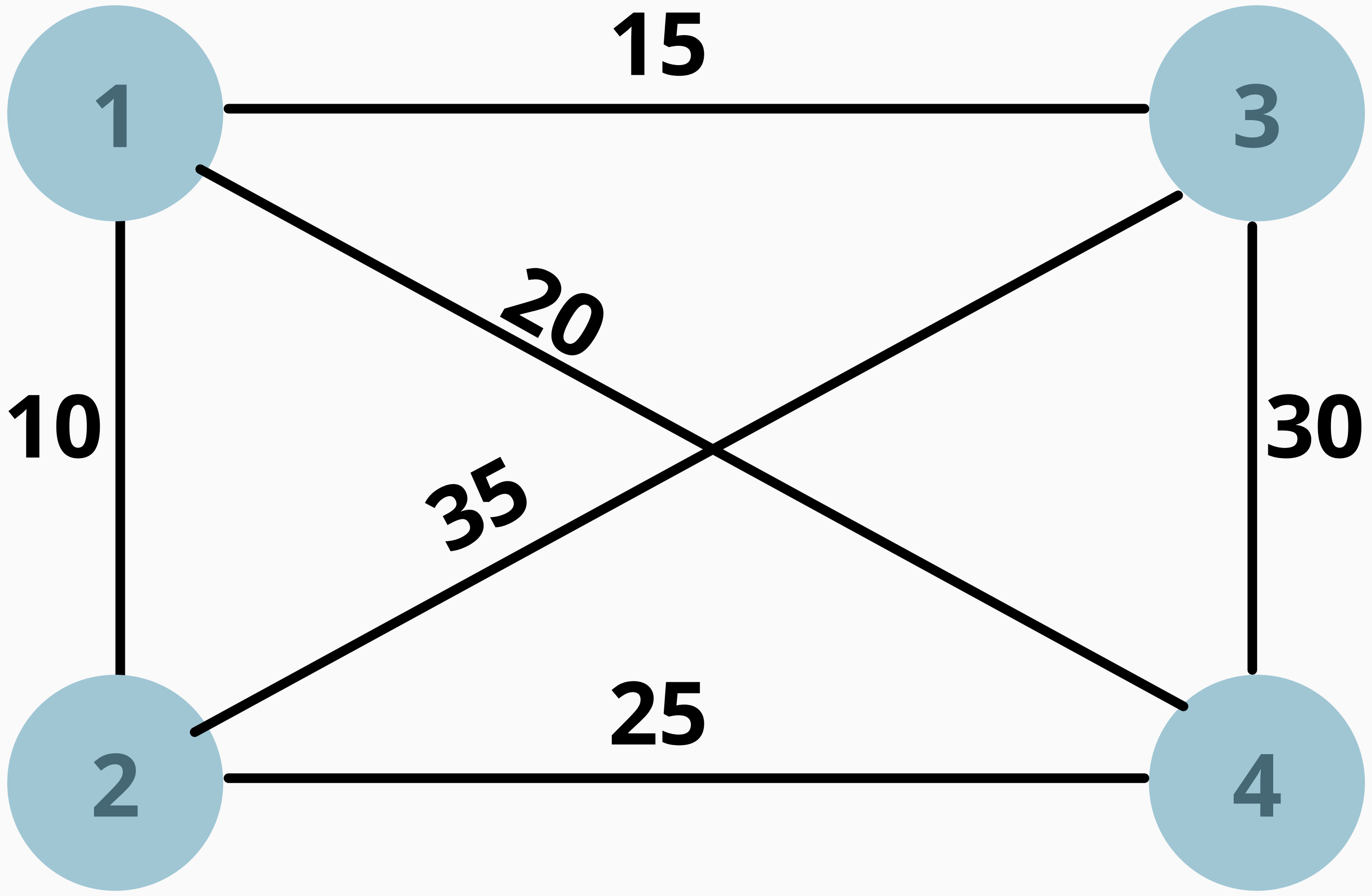
Adjacency Matrix



Adjacency Matrix



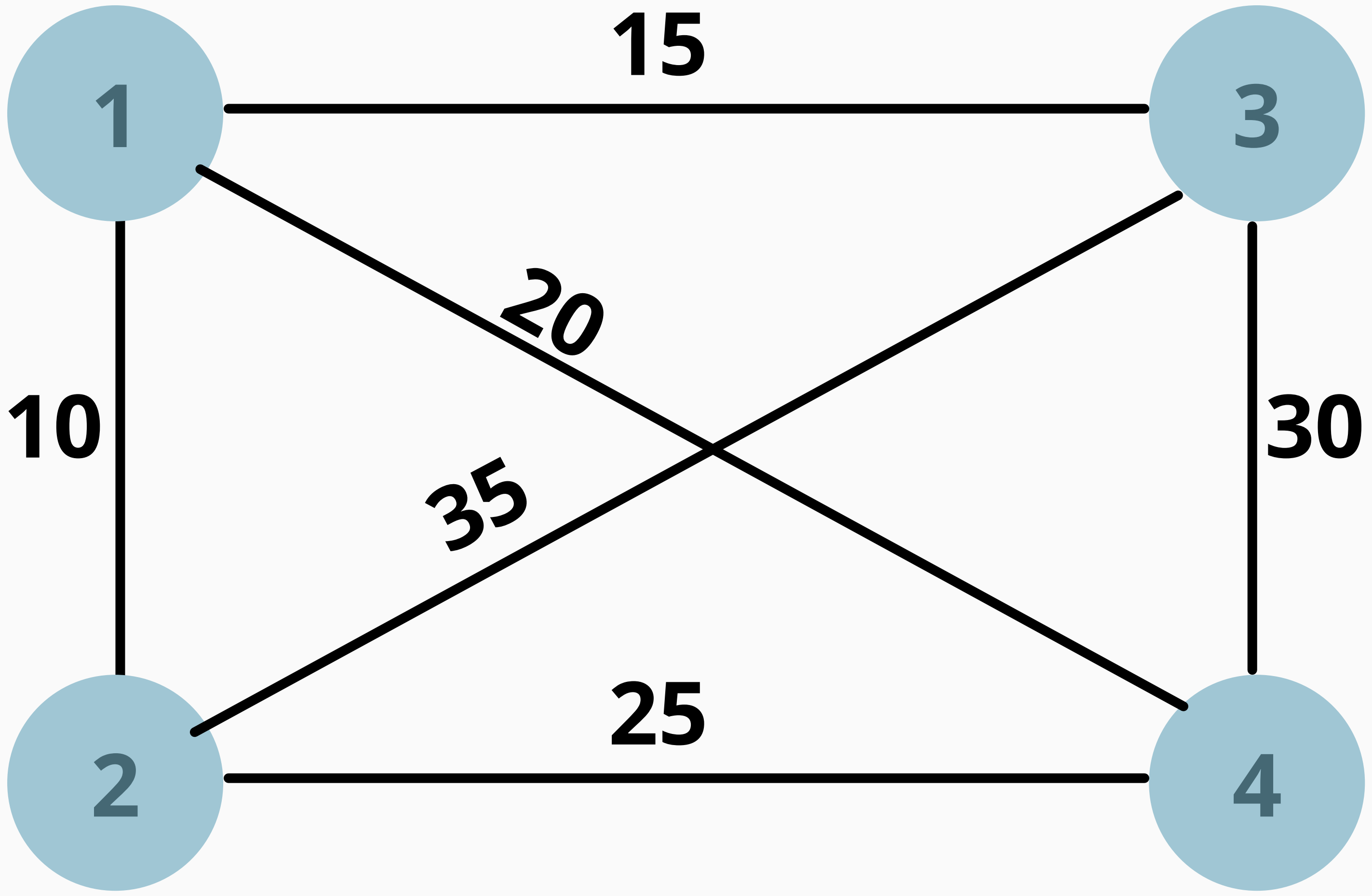
	1	2	3	4
1	0	10	15	20
2	10	0	35	25
3	15	35	0	30
4	20	25	30	0



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = Φ

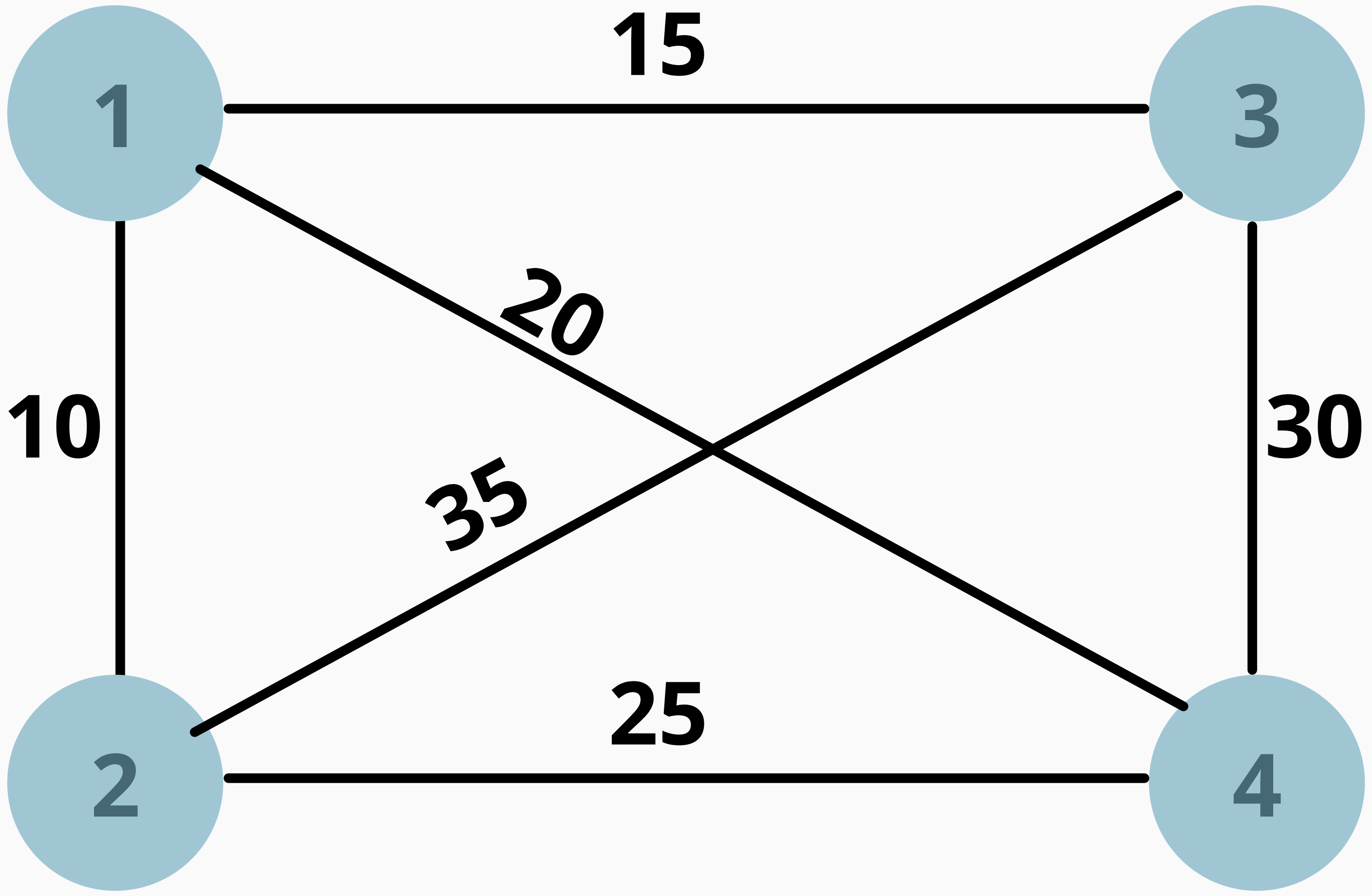


Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = Φ

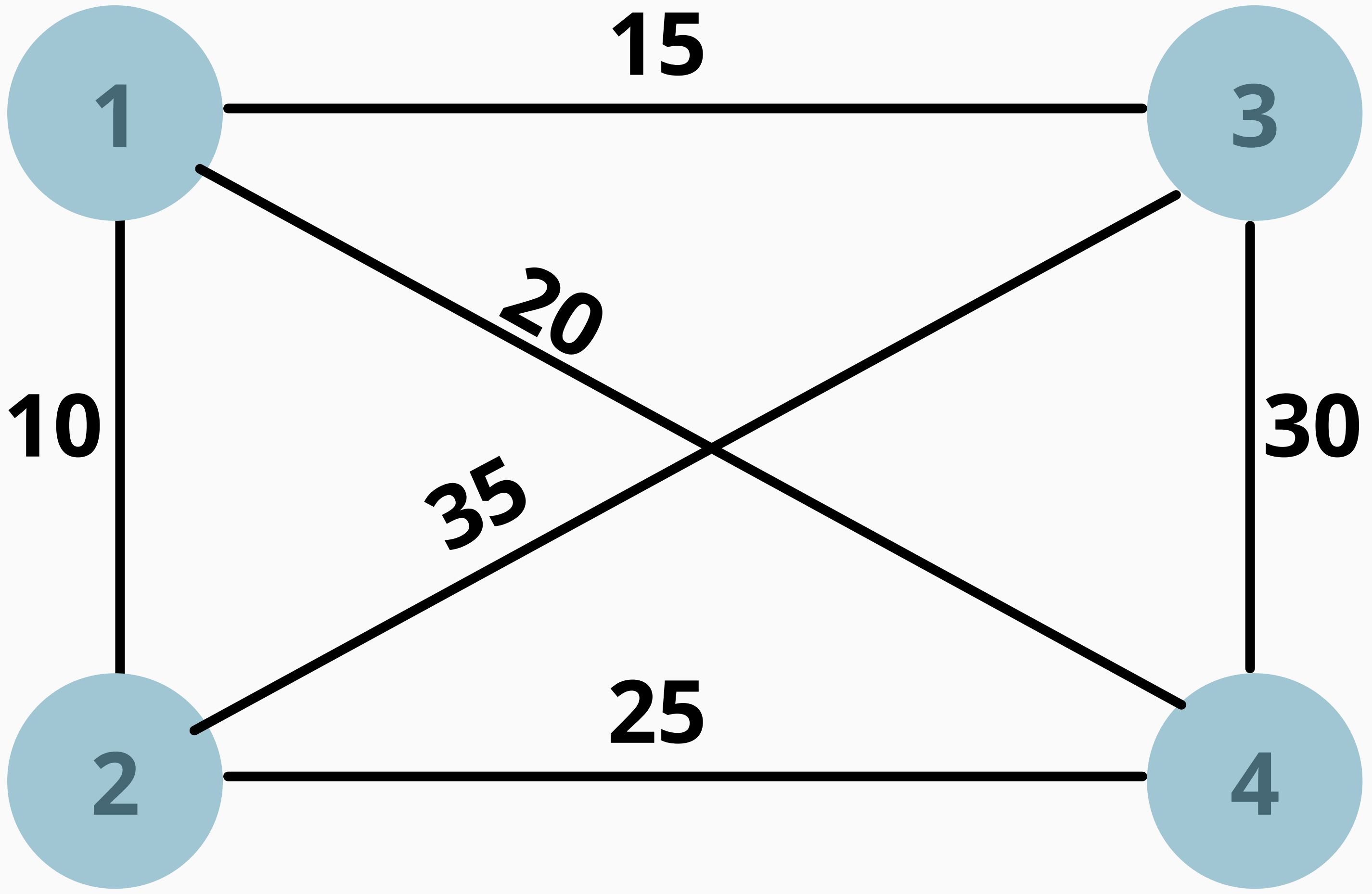
Cost(2,Φ,1) = 10
Cost(3,Φ,1) = 15
Cost(4,Φ,1) = 20



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 1



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 1

Cost(2,{3},1) = 50

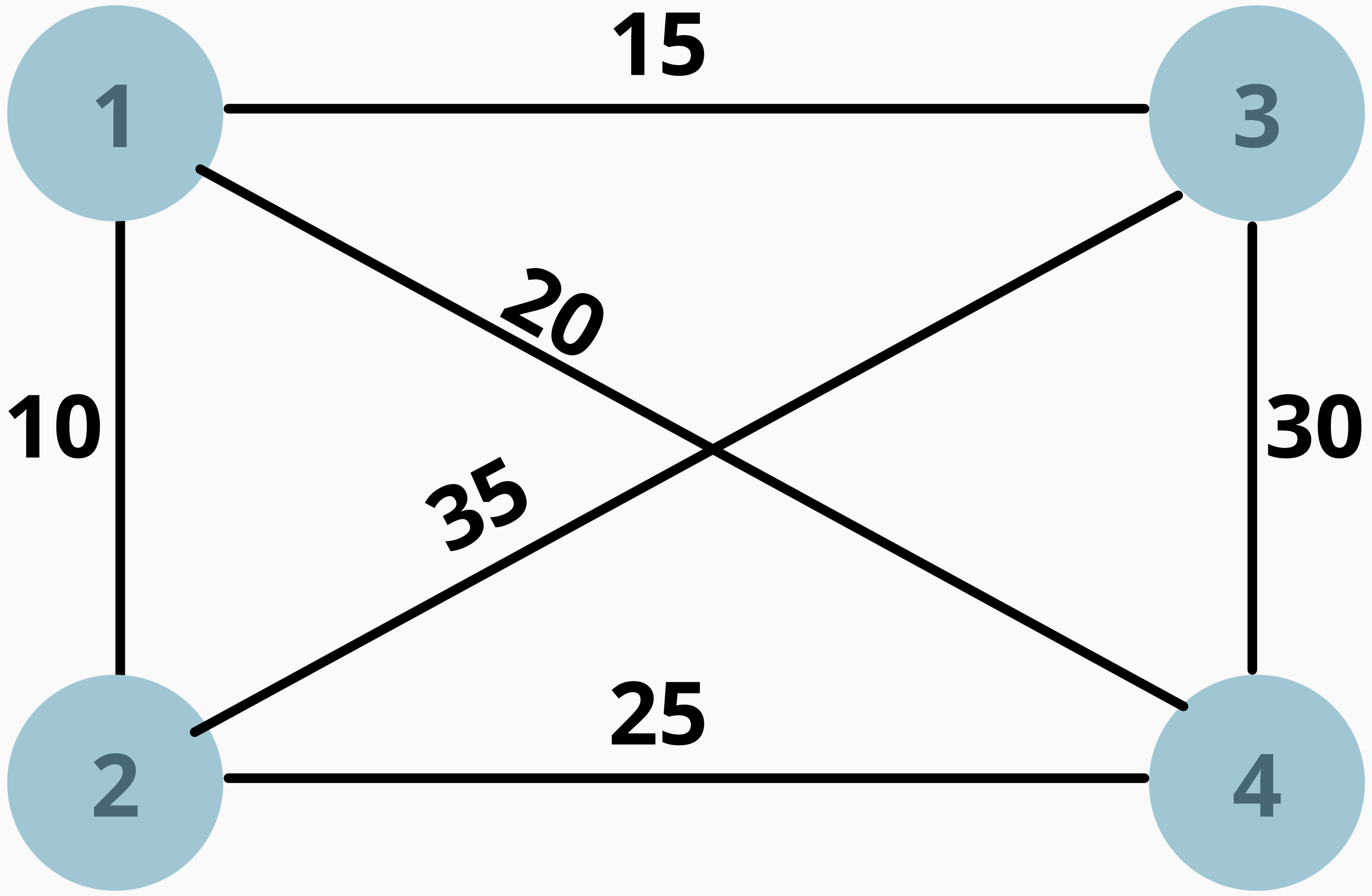
Cost(2,{4},1) = 45

Cost(3,{2},1) = 45

Cost(3,{4},1) = 50

Cost(4,{2},1) = 35

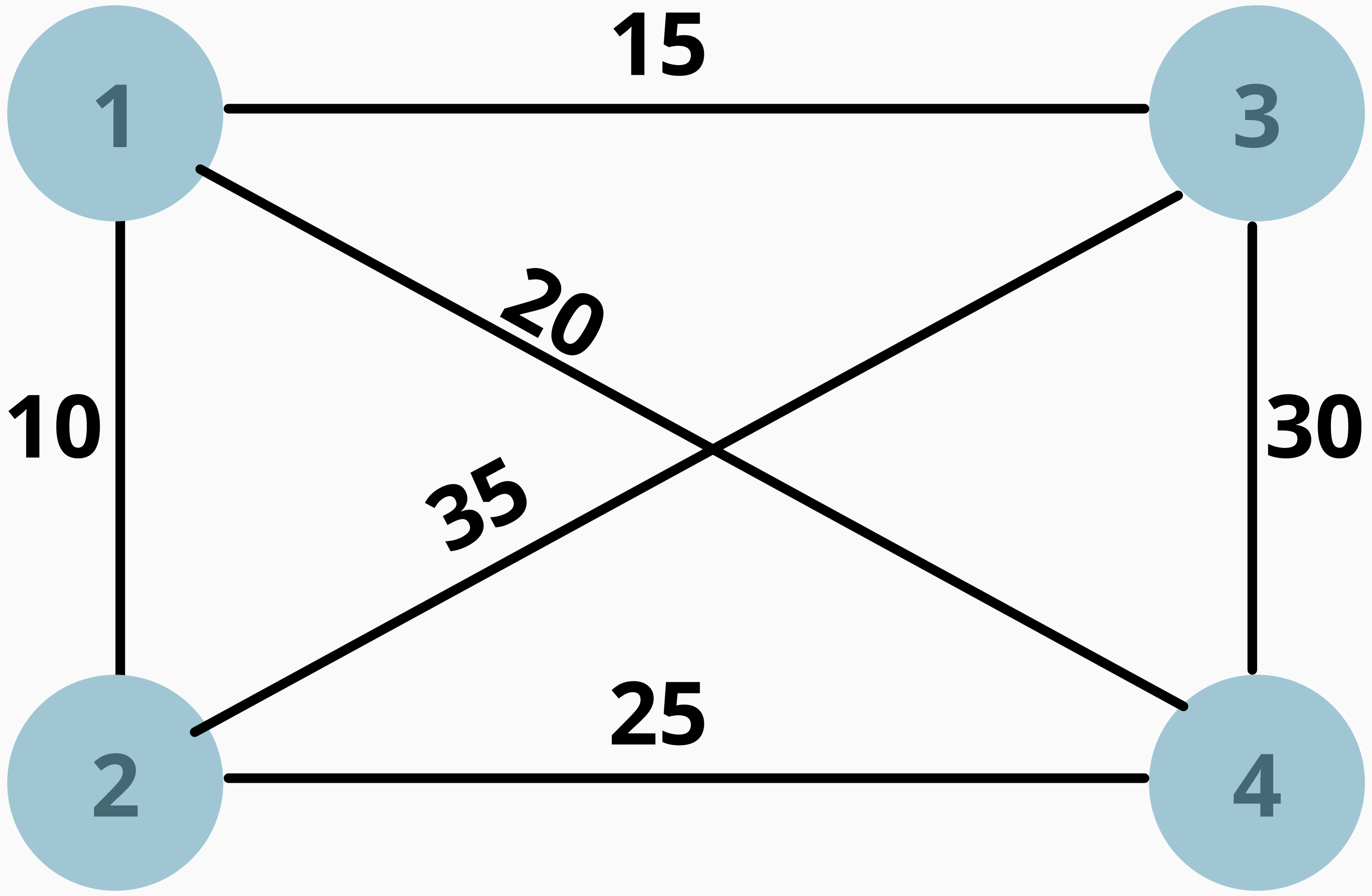
Cost(4,{3},1) = 45



Formula:

$C(j, S, i) = \min C(S - \{j\}, i) + d(j, i)$ where $i \in S$ and $i \neq j c(S, j)$

S = 2



Formula:

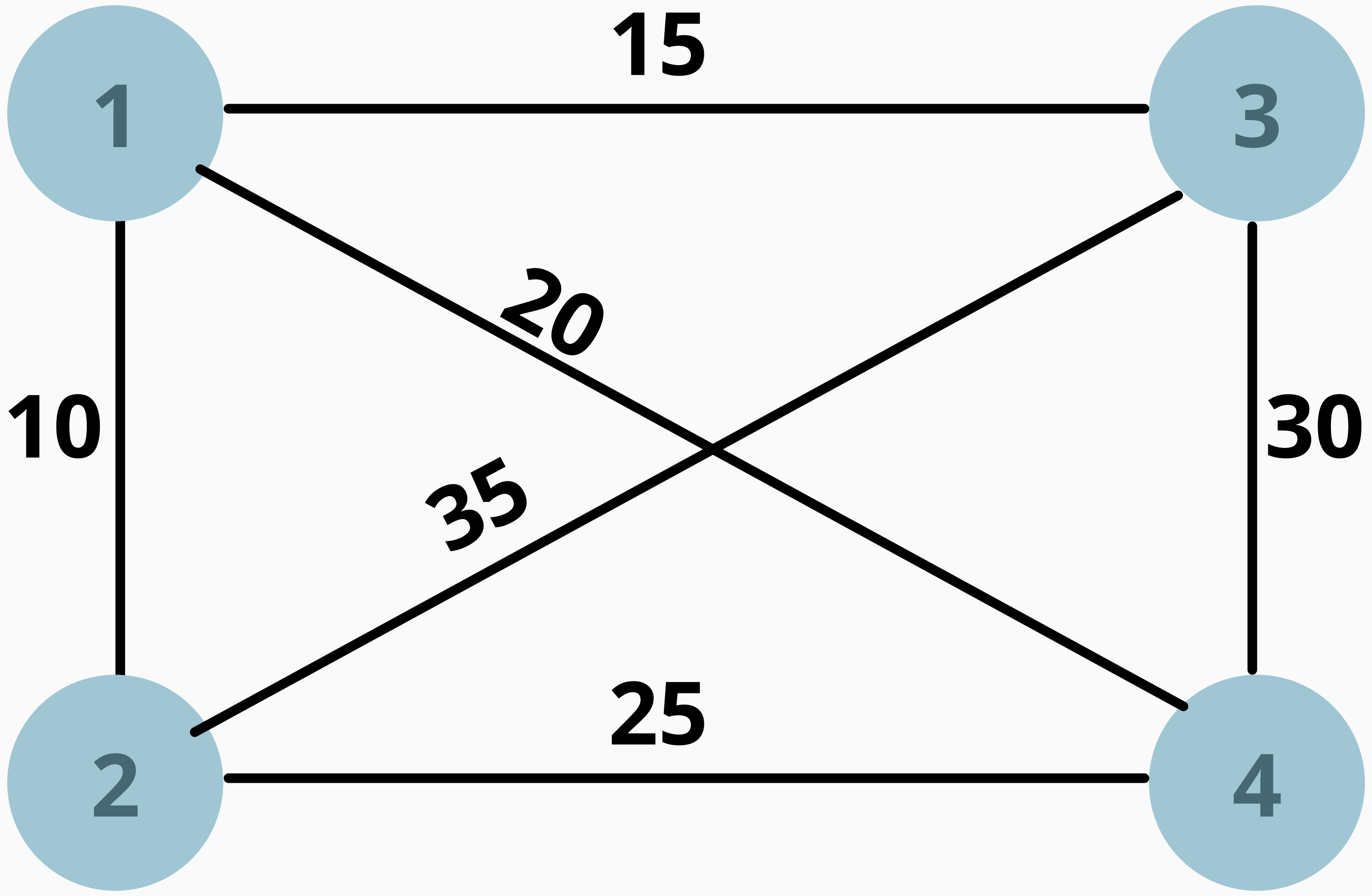
C(j, S, i) = min C(S – {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 2

Cost(2,{3,4},1) = 70

Cost(3,{2,4},1) = 65

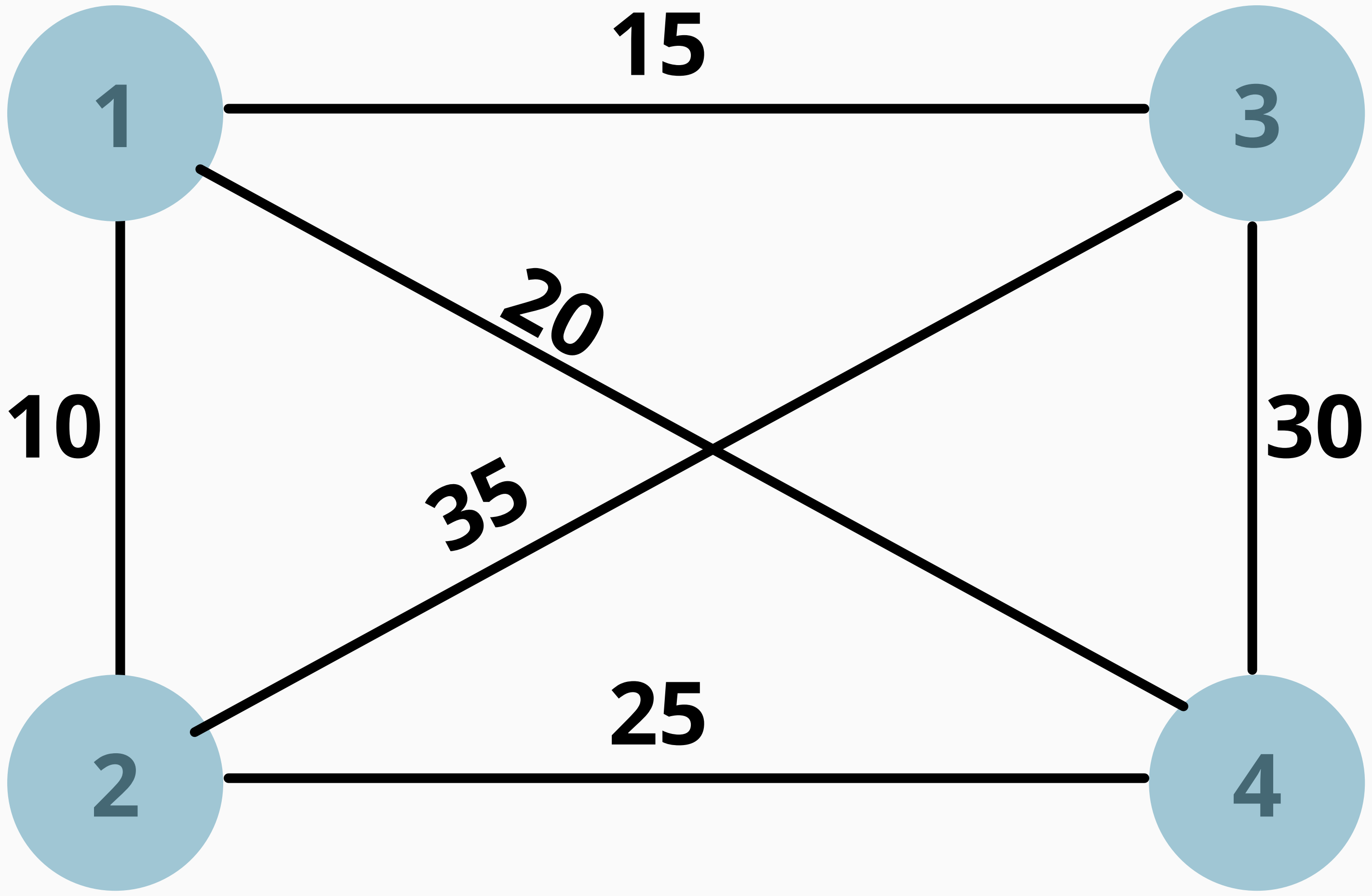
Cost(4,{2,3},1) = 75



Formula:

C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 3

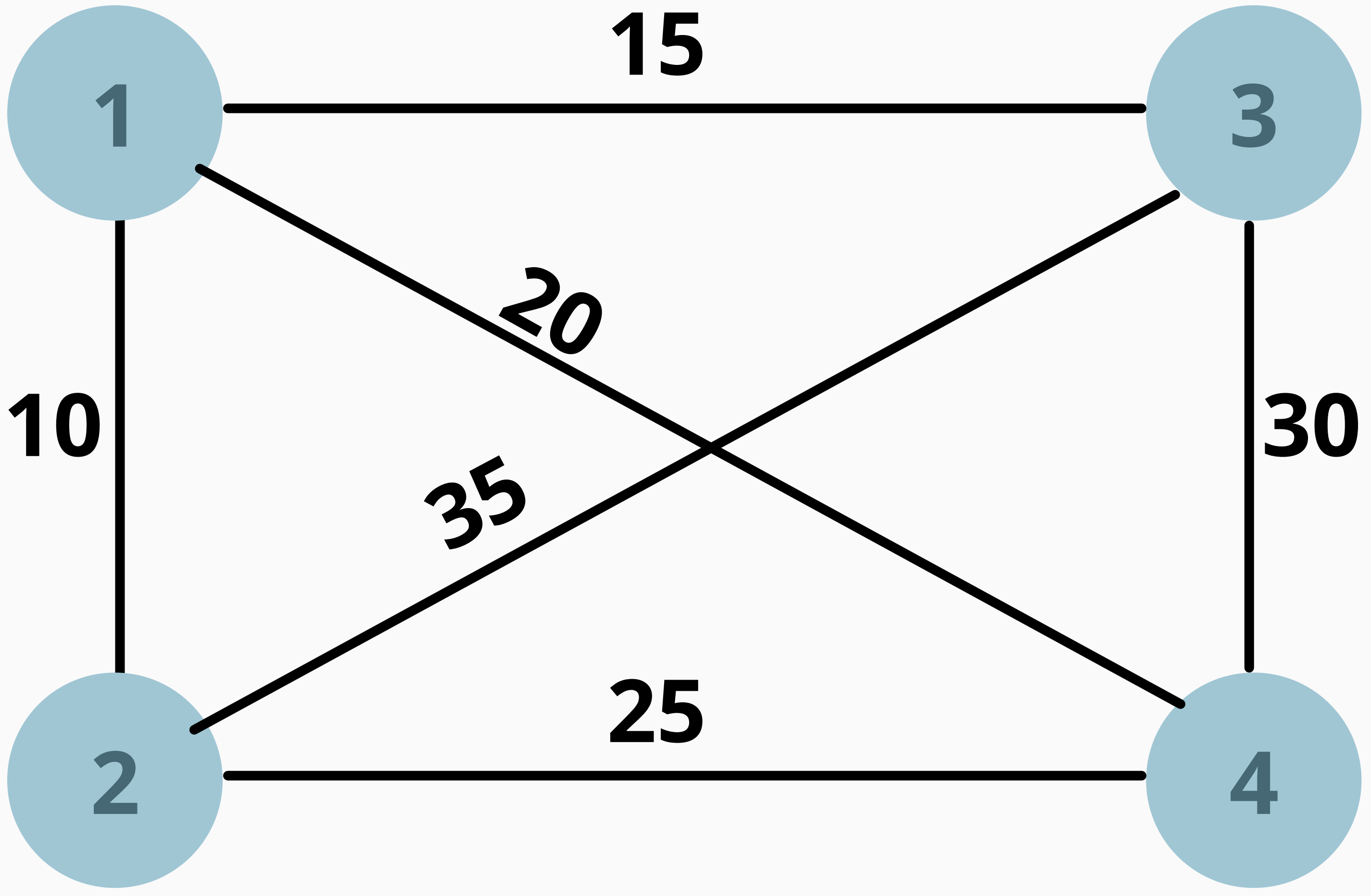


Formula:

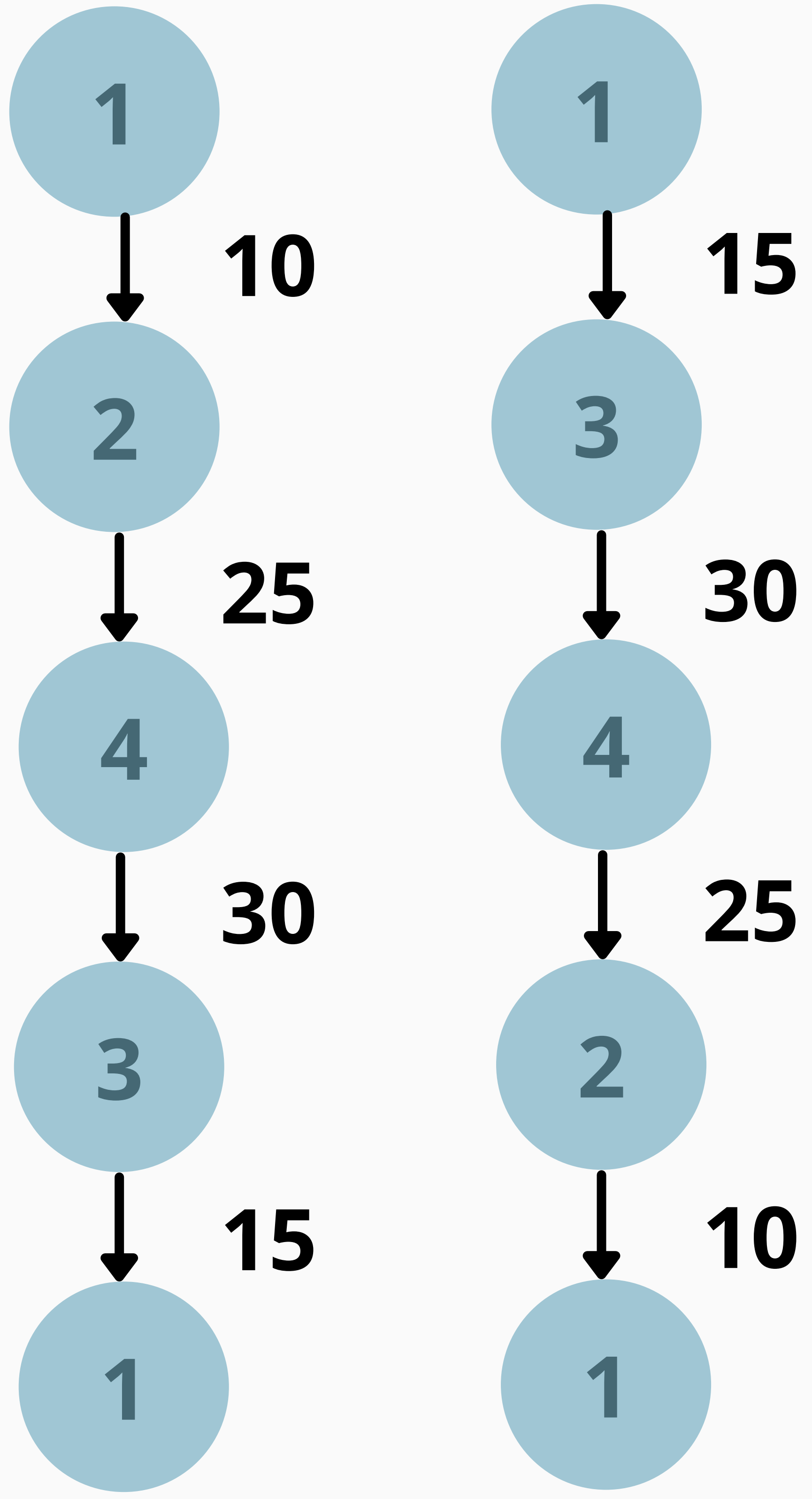
C(j, S, i) = min C(S − {j},i) + d(j, i) where i ∈ S and i ≠ jc(S, j)

S = 3

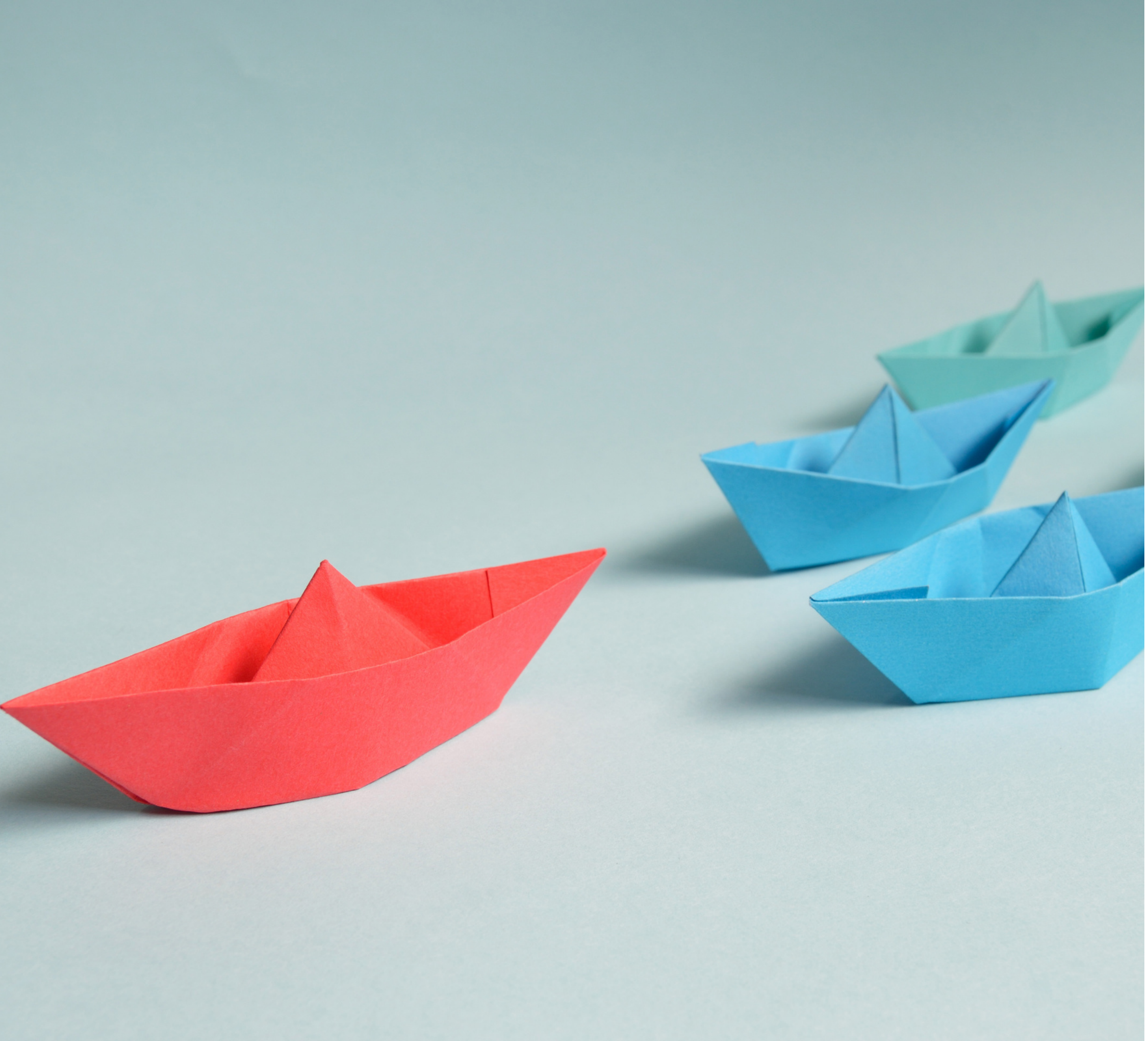
Cost(1,{2,3,4},1) = 80



DP Method:



Minimum Cost Path= 80



Thank You