Problem Formulation

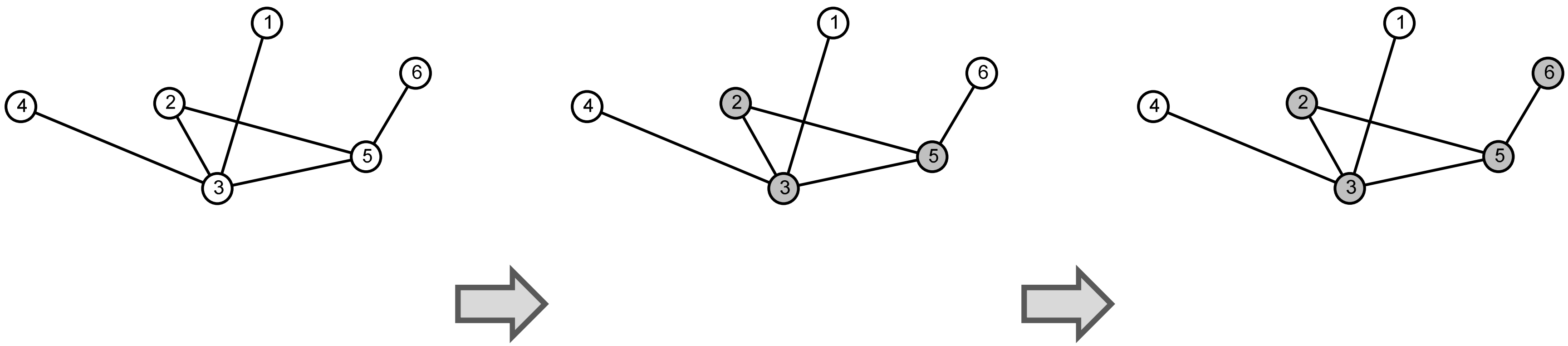
$$f\left(T_{min_i}, r_i\right) = \sum q(v_{min})|p(r_i, v_{min})|$$
$$= q(v_1)|p(r_i, v_1)| + q(v_2)|p(r_i, v_2)| + \cdots + q(v_{V-1})|p(r_i, v_{V-1})| \quad (1)$$

$$f'\left(T_{min_i}, r_i\right) = \sum |p(r_i, v_{min})|$$
$$= |p(r_i, v_1)| + |p(r_i, v_2)| + \cdots + |p(r_i, v_{V-1})| \quad (2)$$

***Assumption 1* — The local best solution for equation (1) provides the best overall solution.**

***Assumption 2* — The derived sub-graph forms a spanning tree structure.**

Algorithm Optimization



Vertex #	Degree	Weight
4	1	4
2	2	2
3	4	3
1	1	1
5	3	5
6	1	6

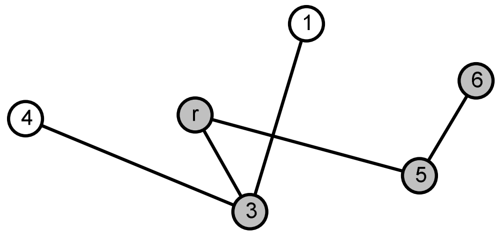
Vertex #	Degree	Weight
4	1	4
1	1	1
6	1	6

Vertex #	Degree	Weight
2	2	2
3	4	3
5	3	5

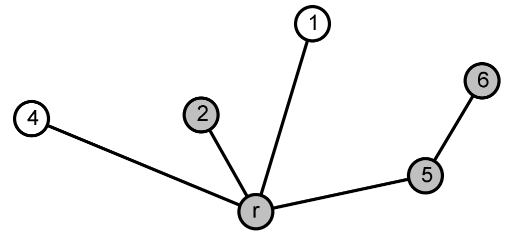
Vertex #	Degree	Weight
4	1	4
1	1	1

Vertex #	Degree	Weight
2	2	2
3	4	3
5	3	5
6	1	6

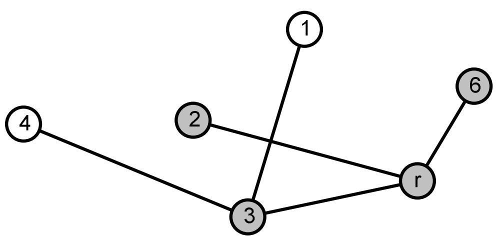
Algorithm Design



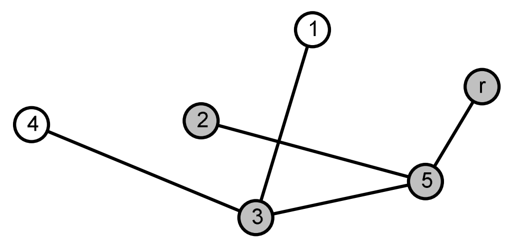
$$f(T,r) = \sum q(v)|p(r,v)|$$
$$= 4 \times 2 + 3 \times 1 + 1 \times 2 + 5 \times 1 + 6 \times 2$$
$$= 30$$



$$f(T,r) = \sum q(v)|p(r,v)|$$
$$= 4 \times 1 + 2 \times 1 + 1 \times 1 + 5 \times 1 + 6 \times 2$$
$$= 24$$



$$f(T,r) = \sum q(v)|p(r,v)|$$
$$= 4 \times 2 + 2 \times 1 + 1 \times 2 + 3 \times 1 + 6 \times 1$$
$$= 21$$



$$f(T,r) = \sum q(v)|p(r,v)|$$
$$= 4 \times 3 + 2 \times 2 + 1 \times 3 + 3 \times 2 + 5 \times 1$$
$$= 32$$

Pseudo-code

```
JADE-BFS-ALGORITHM(Matrix M)
int maxWeight ← 0;
int solutionSum ← 0;
Matrix input ← the input adjacency M;
Matrix solution ← populate with 0;
VertexNode root ← null;
Queue candidateSet ← Ø
for every v ∈ V {
    if v.getWeight > maxWeight {
        maxWeight ← v.getWeight
    }
}
for every v ∈ V {
    if v.getDegree ≠ 1 or v.getWeight = mWeight {
        candidateSet.push(v);
    }
}
while candiateSet ≠ Ø {
    Matrix localSolution ← populate with 0;
    int localSum ← 0;
    Queue temp ← Ø;
    VertexNode vroot ← candidateSet.pop;
    temp.push(vroot);
    vroot.setHops(0);
    for every v ∈ V and v ≠ vroot {
        v.setHops(∞)
    }
    while temp ≠ Ø {
        VertexNode u ← temp.pop;
        for every v ∈ V and v ≠ vroot {
            if input.adjacent(v, u) {
                if v.getHops = ∞ {
                    v.setHops(1 + u.getHops);
                    localSolution.mark(v, u);
                    temp.push(v);
                }
            }
        }
    }
    for every v ∈ V {
        localSum ← localSum + v.getWeight ´ v.getHops;
    }
    if localSum < solutionSum {
        solutionSum ← localSum
        solution ← localSolution
        root ← vroot
    }
}
```

Complexity Analysis

Space Complexity

InputMatrix* + *VertexNode*[*V*] + *OutputMatrix* + *OutputVertexNode* + *Queue

$$
\begin{aligned}
\text{Space Complexity} &= V \times \frac{V}{2} + V \times 3 + V \times \frac{V}{2} + E \\
&= V \times V + V \times 3 + E \\
&= V^2 + V \times 3 + E
\end{aligned}
$$

Time Complexity

Loop 7—11 + Loop 12—15 + Loop 16—47 × Loop 27—38 × Loop 30—37

$$
\begin{aligned}
\text{Time Complexity} &= O(V) + O(V) + O(V) \times O(V) \times O(D) \\
&= 2 \times O(V) + O(V) \times O(E) \\
&= O(V \times E)
\end{aligned}
$$