# Problem

# Notations

The key mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

|  |  |
| --- | --- |
| **Symbol** | **Description** |
| S | the source node |
| T | the sink node |
| edges | the weights of the edges |
| Nk | the kth edge |
| f(e) | the flow of edge |
| c(e) | the biggest flow of edge |

# Analysis of Problem

The given paragraph describes the problem of maximizing the total oil flow from source S to sink T through a network of pipelines. Each pipeline has a limited capacity, and hence, the maximum flow cannot be achieved by simply routing the maximum flow through all the pipelines. To solve this problem, a linear programming model is proposed that considers the constraints of flow conservation and maximum flow capacity of each pipeline.

The input to the model is a directed graph where each edge has a weight representing the maximum flow capacity of that pipeline. The output of the model is the maximum flow from source to sink. The model defines an objective function and constraints. The objective function is to maximize the flow from source to sink, and the constraints are flow conservation and maximum flow capacity of each pipeline. Each pipeline's flow is represented as a variable, and the flow conservation constraint ensures that the inflow to each node equals the outflow from that node. Additionally, the flow through each pipeline must be within the feasible flow range.

In summary, the paragraph describes a linear programming model to maximize the total oil flow from source to sink through a network of pipelines with limited capacity. The model considers flow conservation and maximum flow capacity constraints and defines an objective function to maximize the flow from source to sink.

# Model building and solving

G = (V,E) is a directed graph. Meanwhile.V is the set of nodes and E is the set of edges. Each edge e ∈ E has a non-negative capacity c(e). Suppose there are two special nodes S and T, denoting the source node and the sink node, respectively. Our goal is to find the maximum flow from S to T.

## **Define start, end and edge weights**

Table 2: Defining the edge

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | S | V1 | V2 | V3 | V4 | T |
| S | - | N1:(0,1,4) |  | N2:(0,3,3) | N3:(0,4,4) | - |
| V1 | - | - | N4:(1,2,2) | N5:(1,3,1) | - | - |
| V2 | - | - | - | N6:(2,3,2) | - | N7:(2,t,4) |
| V3 | - | - | - | - | N9:(3,4,3) | N8:(3,t,2) |
| V4 | - |  |  | N10:(4,3,2) | - | N11:(4,t,3) |
| T | - | - | - | - | - | - |

Notes:

·S is the source node and T is the sink node.

· (i , j , w) represents the side from Vi to Vj，where w denotes the weight (the maximum flow of this edge). Nk denotes the k-th edge.

·- indicates that the two nodes are unrelated or unlinkable in this flow direction.

## Define the objective function and constraints

### Defining the constrains （1）Constrains of flow conservation

The constraint is implemented by a two-dimensional array. The size of this array is ((T-1)×2)×n, where there are T-1 nodes and n is the number of edges. Each row of this array represents a constraint and each column represents an edge. Each element of the array represents the relationship between the corresponding constraint and the edge. The element is 1 or -1, if there is a relationship between the constraint and the edge. Otherwise , the element is 0 . Specifically, if the starting point of the i-th edge is j, the element in the i-th column of the j-th row is -1 . Correspondingly , the element in the i-th column of the (j+T-1)-th row is 1. By the same token, we can see that if the ending point of the i-th edge is j, the element in the i-th column of the j-th row is 1 . And, the element in the i-th column of the (j+T-1)-th row is -1. These constraints ensure that the flow of each node is conserved and that the flow of each edge does not exceed its capacity.

The constraint matrix is as follows:

|  |  |
| --- | --- |
|  | () |

The flow balance constraint is shown in Equation 2:

|  |  |
| --- | --- |
|  | () |

**（2）Constrains of flow feasibility**

The actual definition of the range of variables. In practical applications, the pipe capacity is taken into account.

|  |  |
| --- | --- |
|  | () |

### Defining the object function

In this problem, we want to maximize the flow rate from the starting point to the end point. Therefore, we can define the objective function as the sum of the flow rates from the starting point to the end point. Specifically, we can express the objective function as Equation (4):

|  |  |
| --- | --- |
|  | () |

where V is the successor node to the starting point S and f(S,V) denotes the traffic from the starting point S to the node V. The meaning of this objective function is to maximize the traffic from the starting point to the end point.

The objective function is built as follows:

·Step1: Create a one-dimensional array fun of length n, where n is the number of edges. Each element of this array represents the weight of the corresponding edge. In this problem, the weight of each edge is its capacity. So we can initialize each element of the array fun to 0. The array is [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0].

·Step2: In this problem, we want to maximize the flow from the starting point to the end point, so we need to set the weight of the starting point s to -1. Solving the linear programming problem, we will maximize the flow from the starting point to the end point. Specifically, we can set the ith element of the array fun to -1 if the starting point of the ith edge is S . Otherwise , it is 0.

## Solving the maximum flow problem based on linear programming algorithm

|  |  |
| --- | --- |
|  | () |

As shown in the figure below , **the solution results in a maximum flow of 7.0. And the flow through each of the 11 sides is [3. 0. 4. 2. 1. 0. 2. 2. 2. 0. 1. 3.]**

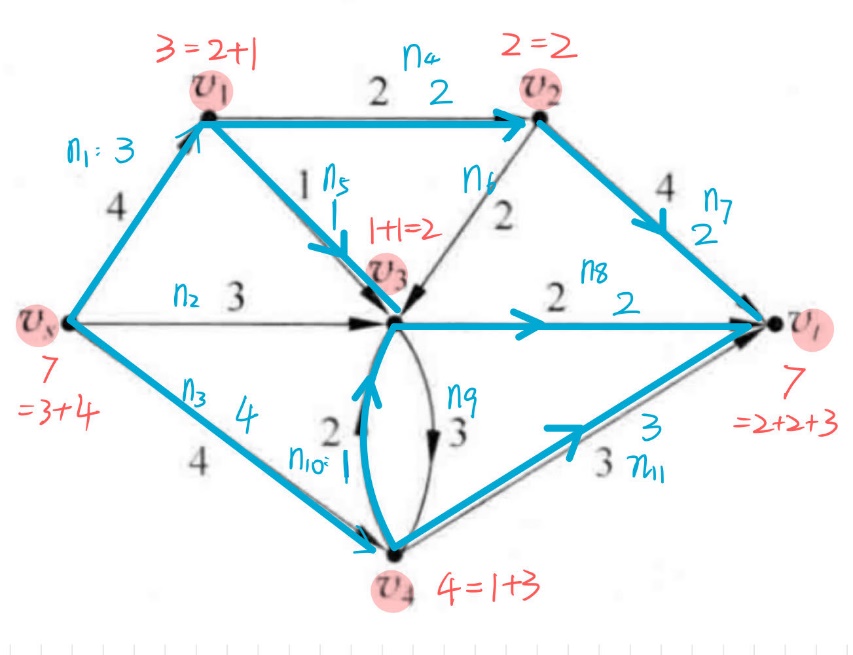


Figure 1：Pipeline transmission volume arrangement

It can be seen that the oil delivery path passes through **S-V1-V2-V3-V4-T**. The oil delivery path is shown in the following figure 2.

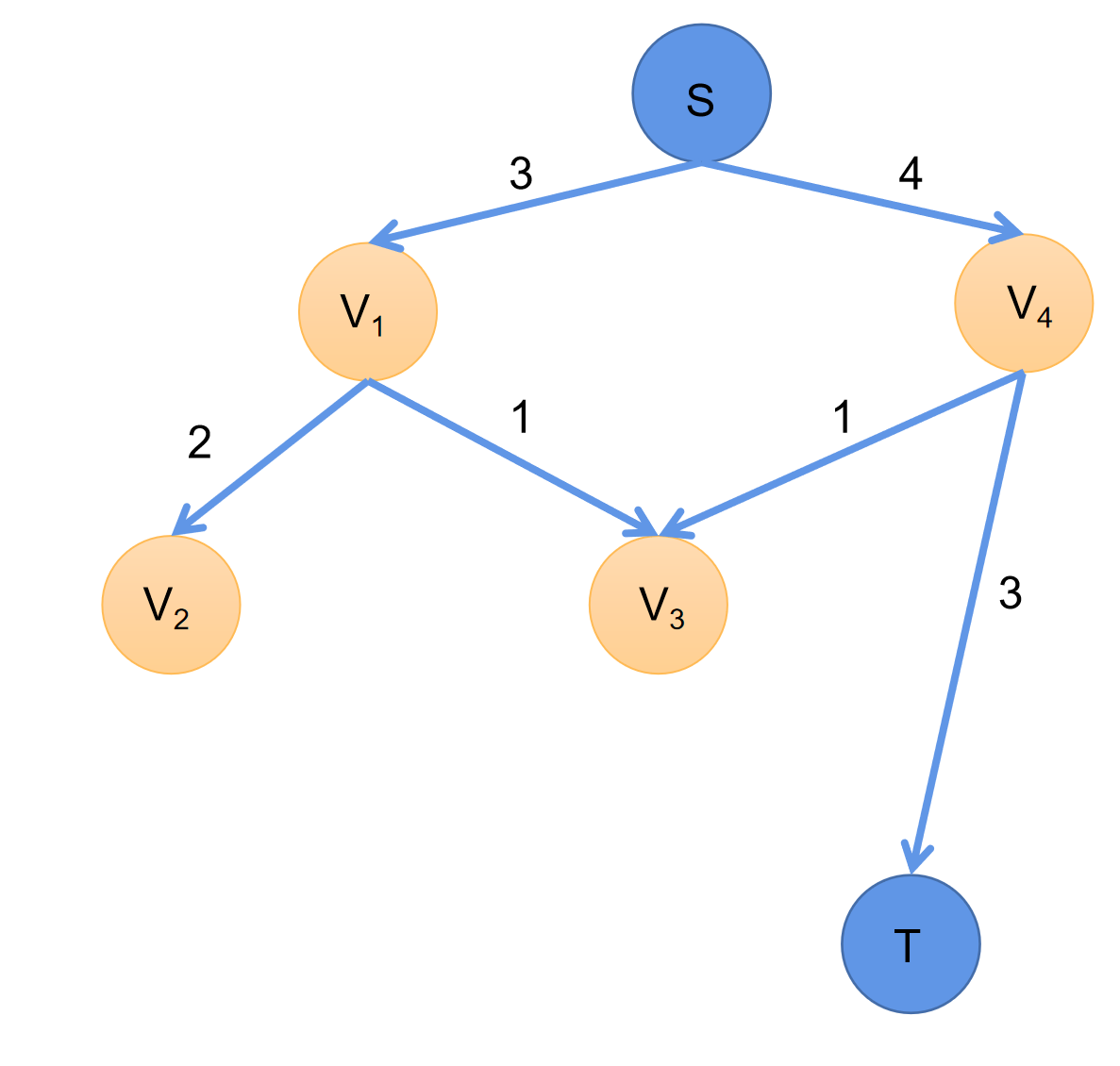


Figure 2: Schematic diagram of oil transmission route

# Appendices

|  |
| --- |
| Appendix |
| Introduce: Python |
| import scipy.optimize  import numpy  # 定义起点和终点  S = 0  T = 5  # 定义边的权重 [Vi,Vj,边权]  edges = numpy.array([[S, 1, 4], [S, 3, 3], [S, 4, 4], [1, 2, 2], [1, 3, 1],  [2, 3, 2], [2, T, 4], [3, T, 2], [3, 4, 3], [4, 3, 2], [4, T, 3]])  def solve():  #定义边的个数  n = len(edges)  # 定义约束条件  constrains = numpy.array([[0 for i in range(n)] for j in range((T - 1) \* 2)])  # 定义目标函数  fun = numpy.array([0 for i in range(n)])  # 定义边的权重范围 变量的取值范围  bound = [(0, edges[i, 2]) for i in range(0, n)]  for i in range(n):  if edges[i, 0] == S:  fun[i] = -1  else:  constrains[edges[i, 0] - 1][i] = -1  constrains[edges[i, 0] + T - 2][i] = 1  if edges[i, 1] != T:  constrains[edges[i, 1] - 1][i] = 1  constrains[edges[i, 1] + T - 2][i] = -1  #print(fun) #print(constrains) #print(bound)  # 使用线性规划求解 res结果  res = scipy.optimize.linprog(fun.tolist(), constrains.tolist(), [0 for i in range((T - 1) \* 2)], None, None, bound)  print(-res.fun) #目标函数的最小值 即最大值的相反数 7.0  print(res.x) #似目标函数最大的变量取值 [3. 0. 4. 2. 1. 0. 2. 2. 0. 1. 3.]  #res.x是一个一维数组，表示使目标函数最大的变量取值。在这段代码中，res.x的长度为11，因为有11条边。  #res.x[i]表示第i条边的流量。  if \_\_name\_\_ == '\_\_main\_\_':  solve() |