University of Science and Technology of China

Optimization Algorithm

MATH5015P

Course Project Report

Author:
Jiazhi He: PB19151772

May 14, 2022

Contents

1 Problem Description

1.1 Research Background

- The assignment problem is a special form of the transportation problem, and it is a special linear programming problem. The process of solving the assignment problem is to find out a matching relationship between the assigned task and the assigned object with the highest execution efficiency or the lowest cost.
- Transportation problem: How to transport goods from a series of origins (such as: factories, warehouses) to a series of destinations (such as: warehouses, customers) at the lowest possible cost.
- The essence is to find a set of transportation relations with the minimum weight that satisfy the constraints in a weighted (complete) bipartite graph.

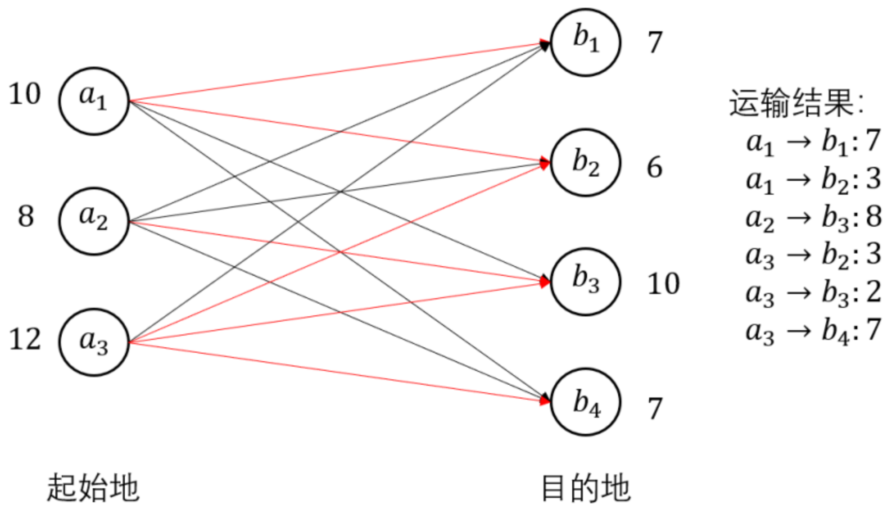

Figure 1: Schematic diagram of the transportation problem

Model 1: transportation problem. Suppose we have m origins, n destinations, the supply of each origin is a_i, the demand of each destination is b_j, the unit transportation cost from a starting point i to a destination j is c_{ij}, and the transportation volume of each side is x_{ij}, then the above transportation problem can be expressed in the following form:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} && (1)\\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = a_i, (i = 1, 2, \cdots, m) && (2)\\
& \sum_{i=1}^{m} x_{ij} = b_j, (j = 1, 2, \cdots, m) && (3)\\
& x_{ij} \geq 0 && (4)
\end{aligned}
$$

- Assignment Problem: A special class of transportation problems. It is only necessary to find the corresponding relationship in the bipartite graph shown in the transportation problem.
- The main feature of the assignment problem that is different from the general transportation problem is that the relationship between the node pairs is changed from the original continuous variable of the transportation value to the discrete variable of whether there is an assignment relationship.

Model 2: One-to-one assignment (maximum matching). Consider the assignment problem of one-to-one relationship, we have n assigned objects and assigned tasks, and the cost of the object i to execute the task j is c_{ij}, x_{ij} is a decision variable indicating whether an assignment relationship occurs. Then the model of the one-to-one assignment problem is as follows:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} && (5)\\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, (i = 1, 2, \cdots, n) && (6)\\
& \sum_{i=1}^{n} x_{ij} = 1, (j = 1, 2, \cdots, n) && (7)\\
& x_{ij} \in \{0, 1\} && (8)
\end{aligned}
$$

- The ability to describe the one-to-one traditional assignment problem is still relatively limited. Therefore, the research on the modeling and solution of a series of variants of the assignment problem is an indispensable part of the field of optimization modeling.

- In this report, I will give an optimization model and solution algorithm for a variant of the assignment problem based on specific requirements.

From the correspondence between assigned tasks and assigned objects, variant problems can be divided into the following categories:

- **one-to-many:** It means that the same assigned object needs to complete multiple assigned tasks, and each assigned task only needs to be completed once; vice versa. (such as: teaching assistant correcting coursework)
- **many-to-many:** It means that the same assigned object needs to complete multiple assigned tasks, and each assigned task needs to be completed multiple times. (such as: review questions)

In addition, the variant of the assignment problem also comes from the limitations of **objective function** and **constraints** in real-world applications, such as: considering the additional cost of executing multi-task objectives, the upper limit of a single object to perform tasks , etc. We will mainly focus on the modeling and solution of a variant problem under the one-to-many relationship.

On the basis of the one-to-one assignment problem, we can give a model of the one-to-many assignment problem by adjusting the constraints.

Model 3: one-to-many assignment. Suppose there are m assigned objects and n assigned tasks, and satisfy $m \leq n$. At the same time, we give the number of tasks performed by each assigned object i the lower limit l_i and upper limit u_i, c_{ij}, x_{ij} still use the definition in Model2, then one-to-many assignment problem can be written as:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{9}$$

$$\text{s.t.} \quad l_i \leq \sum_{j=1}^{n} x_{ij} \leq u_i, (i = 1, 2, \cdots, m) \tag{10}$$

$$\sum_{i=1}^{m} x_{ij} = 1, (j = 1, 2, \cdots, n) \tag{11}$$

$$x_{ij} \in \{0, 1\} \tag{12}$$

The one-to-many assignment problem can be further relaxed into the many-to-many assignment problem

Model 4: many-to-many assignment. Suppose there are m assigned objects and n assigned tasks, and satisfy $m \le n$. At the same time, we give the number of tasks performed by each assigned object i the lower limit $l_i^{(1)}$ and upper limit $u_i^{(1)}$, each assigned task needs to be executed by at least $l_i^{(2)}$ at most $u_i^{(2)}$ assigned objects, c_{ij}, x_{ij} still use the definition in Model2, then the many-to-many assignment problem can be written as:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} && (13)\\
\text{s.t.} \quad & l_i^{(1)} \le \sum_{j=1}^{n} x_{ij} \le u_i^{(1)}, (i = 1, 2, \cdots, m) && (14)\\
& l_i^{(2)} \le \sum_{i=1}^{m} x_{ij} \le u_i^{(2)}, (j = 1, 2, \cdots, n) && (15)\\
& x_{ij} \in \{0, 1\} && (16)
\end{aligned}$$

Let us consider review questions.

Problem Description: Given m reviewers and n manuscripts, assign manuscripts to reviewers according to the following conditions:

- For each manuscript, there must be q reviewers.
- For each reviewer, review at most p manuscripts.
- For each manuscript i, the decision of reviewer j is B_{ij}. It means to want if $B_{ij} = 2$, maybe if $B_{ij} = 1$, do not want if $B_{ij} = 0$.
- If a conflict of interest exists and the reviewers cannot review the manuscript, the decision is $B_{ij} = -1$.

Parameter Description: $c_{n\times m}$ is the weight matrix used to evaluate the relevance between reviewers and manuscripts, for manuscripts i and reviewers j, there are

$$c_{ij} = \begin{cases} 0 & \text{if } B_{ij} = 2 \\ \text{CostRef}(> 0) & \text{if } B_{ij} = 1 \\ \text{CostScale}(\text{Costscale} > \text{CostRef}) & \text{if } B_{ij} = 0 \\ \infty & \text{if } B_{ij} = -1 \end{cases}$$

Model: integer programming model

$$
\begin{aligned}
\min_{S_{ij}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} S_{ij} c_{ij} && (17)\\
\text{s.t.} \quad & \sum_{j} S_{ij} = q && (18)\\
& \sum_{i} S_{ij} \leq p && (19)\\
& S_{ij} \in \{0, 1\} && (20)
\end{aligned}
$$

The above model can be transformed into a minimum cost flow model.

Network flow refers to the distribution of flow in a directed graph with capacity on each side, so that the flow of each side will not exceed its capacity.

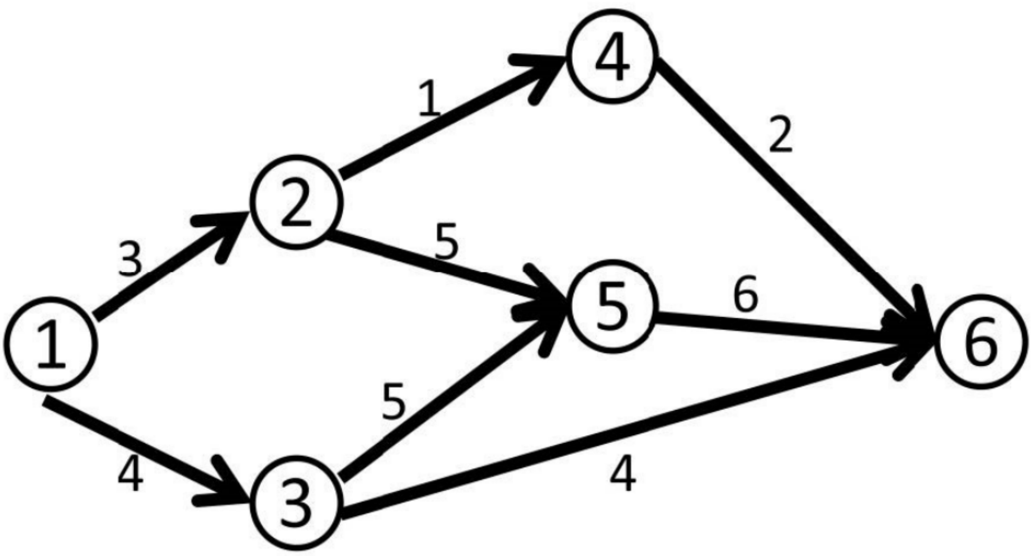

Figure 2:

Given a graph structure $G(V, E)$ similar to the above figure, the node set V contains source points s and sink points t whose demand is not zero, each edge (i, j) in the edge set E has capacity u_{ij} and cost c_{ij}.
Min-Cost flow.

$$
\text{minimum cost :} \quad \min \sum_{(i,j) \in E} c_{ij} x_{ij}
$$

$$\text{flow conservation :} \quad \text{s.t.} \quad \sum_j S_{ij} = q$$

$$\text{source stream value :} \quad \sum_{i \in V_s^+} x_{si} = f_0, \left(\sum_{i \in V_t^-} x_{it} = f_0 \right)$$

$$\text{capacity constraints :} \quad x_{ij} \leq u_{ij}$$

The transportation model is a special minimum-cost flow model.

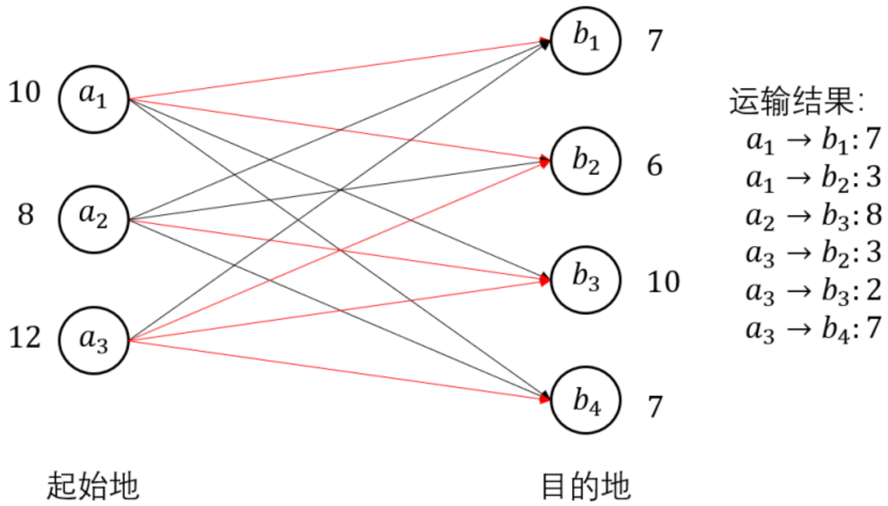

Figure 3:

The network flow model of the review problem is given below

Variable description: x_{ij} is the flow value on each edge.

Parameter Description:

- $V_p := \{p_1, \cdots, p_n\}$: manuscript collection
- $V_r := \{r_1, \cdots, r_m\}$: reviewer collection
- s : source point
- t : sink point
- $V = V_r \cap V_p \cap \{s, t\}$: node set
- $E_{in} = \{(s, p_i) : \forall p_i \in V_p\}$: source edge set

- $E_{out} = \{(r_j, t) : \forall r_j \in V_r\}$: sink edge set

- $E = E_{in} \cup E_{out} \cup \{(p_i, r_j) : B_{ij} \geq 0\}$: edge set

- c_{ij}: the cost of edge $(e_i, e_j) \in E$, where

$$c_{ij} = \begin{cases} 0 & \text{if } B_{ij} = 2 \\ \text{CostRef}(> 0) & \text{if } B_{ij} = 1 \\ \text{CostScale}(\text{Costscale} > \text{CostRef}) & \text{if } B_{ij} = 0 \end{cases}$$

- u_{ij} : the capacity of edge $(e_i, e_j) \in E$, where

$$u_e = \begin{cases} q & \text{if } e \in E_{in} \\ p & \text{if } e \in E_{out} \\ 1 & \text{if } e \in \{(p_i, r_j) : B_{ij} \geq 0\} \end{cases}$$

Model: network flow model

$$\min_{x_{ij}} \quad \sum_{p_i \in V_p} \sum_{r_j \in V_r} c_{ij} x_{ij} \tag{21}$$

$$\text{s.t.} \quad \sum_{p_i \in V_p} x_{si} = \sum_{r_j \in V_r} x_{jt} = nq \tag{22}$$

$$\sum_{j:(i,j) \in E} x_{ij} = \sum_{k:(k,i) \in E} x_{ki} \quad \forall i \in V \backslash \{s, t\} \tag{23}$$

$$x_{ij} \in \{0, 1\} \tag{24}$$

The above model can be solved by network flow algorithm or integer programming algorithm.

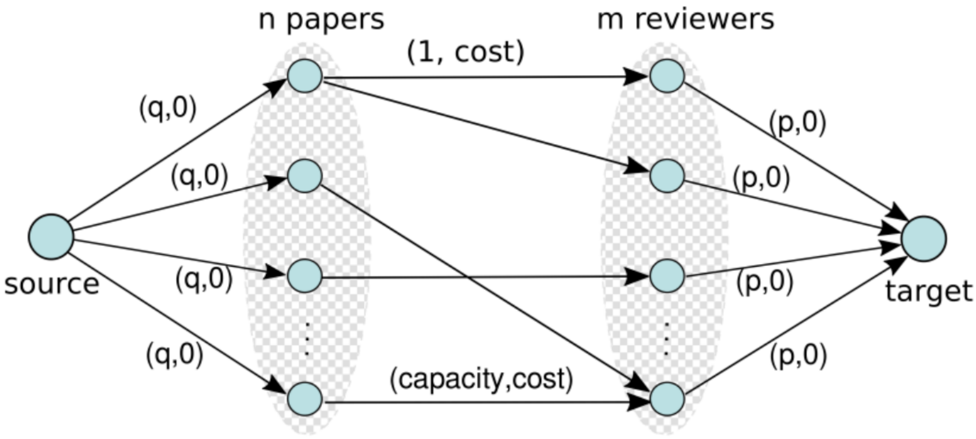

Figure 4: Minimum cost flow network for paper review

1.2 Optimization Modeling

In the assignment problem of one-to-many relationship, when the execution order of multiple tasks assigned by an object has an impact on the optimization goal, how should we characterize this type of problem?

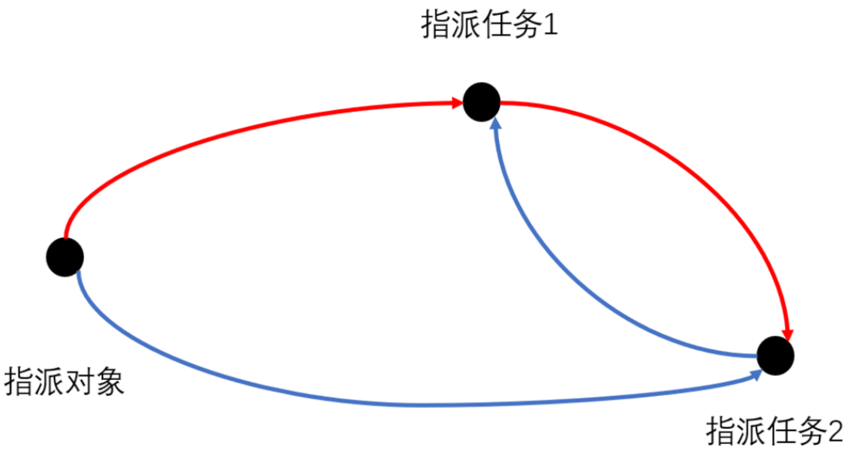

Figure 5: relationships between tasks in the Assignment problem

Here are two common modeling ideas to characterize the ordering relationship in the one-to-many assignment problem:

- Bundle assignment tasks to construct new task nodes
- Use network flows to characterize order relations

Scene introduction: There is an existing excavation machine assignment task (as shown in the figure below), which requires m machines to be assigned to n work sites to complete the excavation task ($m \leq n$), and the excavation machine movement costs are incurred when the machine moves between different task locations.

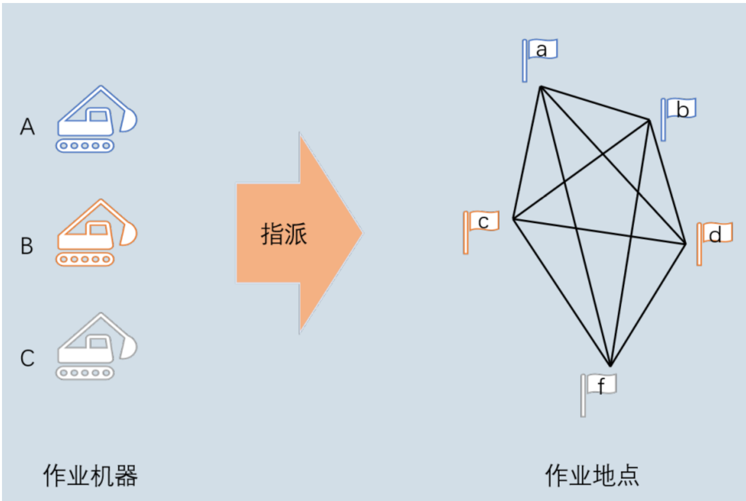

Figure 6:

Problem Description
Input parameters:

- Number of mining equipment
- Number of work sites
- The cost of equipment traveling to different locations for work
- The cost of moving equipment between locations

Constraints:

- A piece of equipment goes to at least one location for work

- A piece of equipment can go to k places at most for work
- A location only needs to be executed once
- During the moving process, the equipment shall not pass through other workplaces to interfere with the work

Optimization goal:

- Minimize operating costs

The structure in the figure below is

- The complete bipartite graph between the working machine and the working place
- The complete graph of the work site

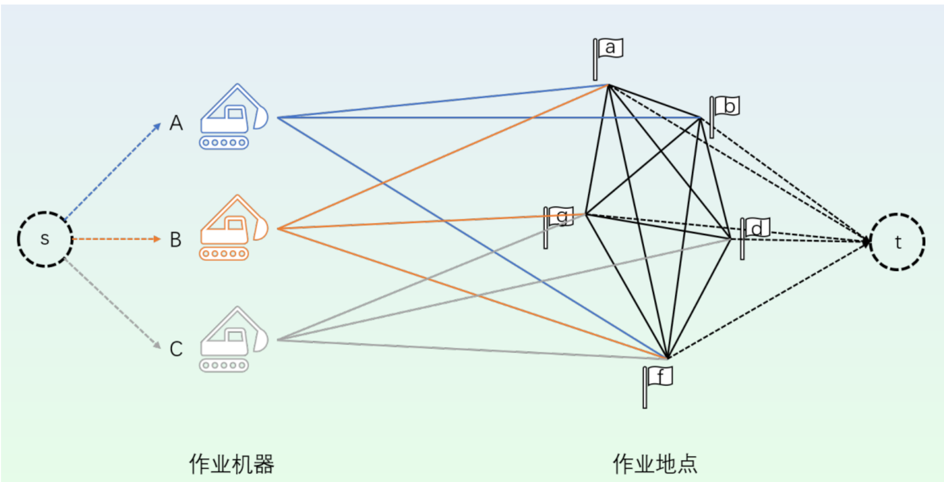

Figure 7:

In the graph structure, we need to add the attributes corresponding to the input parameters, and give the specific model representation of the objective function and constraints. First, give the definition of the weight of each edge in the graph structure:

- The weight of the edge directly connected from the node representing the device to the node representing the location represents **the cost of assigning the device to the location**, as shown in the figure below c_{Aa}

- The weight of an edge connected from a node representing a location to another node representing a location represents **the cost of a device moving and performing the task of the next location after performing the task of the previous location**, as shown in the figure below c_{df} as shown.

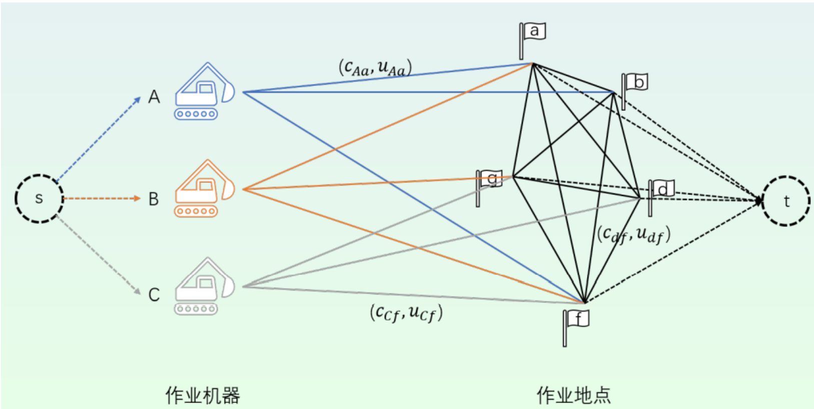

Figure 8:

The upper limit of the capacity of each edge in the graph:

- We define x_e to represent the flow value of edge e, there is $x_e \leq 1$.
- The upper limit of the capacity of each side is 1, there is $u_{Aa} = 1$.

Constraint conversion:

- A location only needs to be executed once
- the equipment does not pass through other working places during the moving process $\iff$ the flow value of the node is conserved, and the flow value is 1.

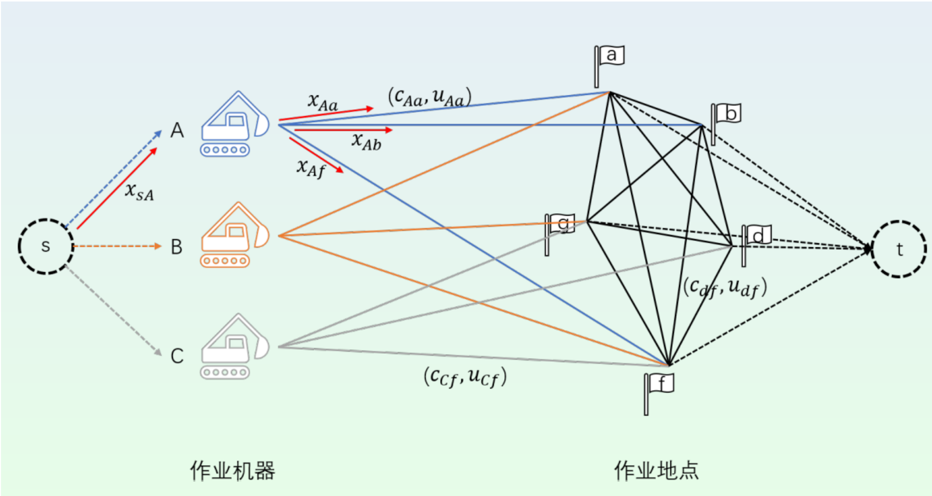

Figure 9:

Constraint Transformation:

- Assign at least 1 tasks per device $\iff$ The flow value from source to each device is 1

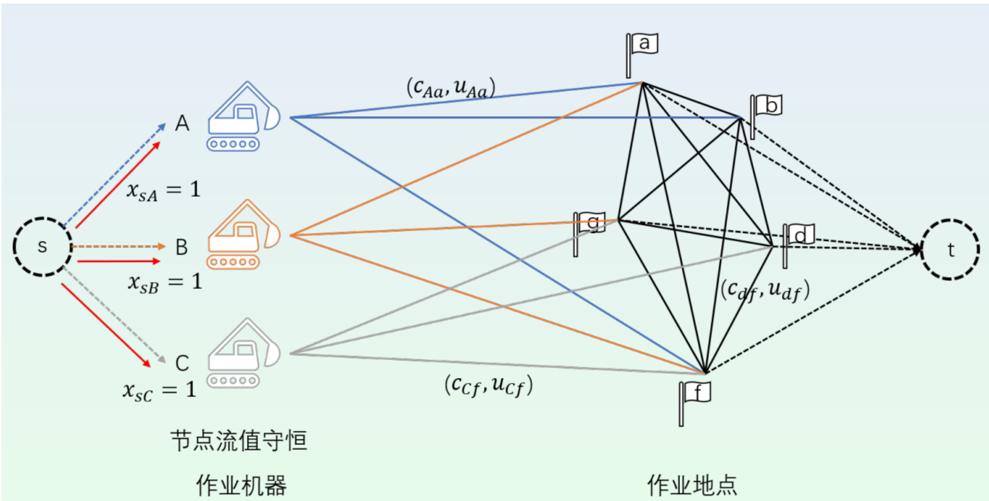

Figure 10:

After sorting out the above expressions, the explanations of the variables and parameters of the following model are given:

Table 1: Model Variables and Parameter Interpretation.

Symbols	**Meaning**
V	node set
$i \in V$	node
$E \subset \{(i,j) \mid i,j \in V\}$	directed edge set
$G(V,E)$	graph structure
$s \in V$	source point
$t \in V$	sink point
$V_i^+ = \{j \in V \mid (i,j) \in E\}$	Head node set corresponding to the edge with i as the tail node
$V_i^- = \{j \in V \mid (j,i) \in E\}$	Tail node set corresponding to the edge with i as the head node
c_{ij}	the cost of edge (i,j)
m	the number of equipment (total flow value)
x_{ij}	the flow value of edge (i,j), **decision variable**

Model: One-to-many assignment considering order relationship.

$$\min \quad \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{25}$$

$$\text{s.t.} \quad \sum_{i\in V_s^+} x_{si} = m \tag{26}$$

$$\sum_{j\in V_i^+} x_{ij} = \sum_{k\in V_i^-} x_{ki} = 1, \forall i \in V \backslash \{s,t\} \tag{27}$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in E \tag{28}$$

2 Test method

2.1 Input description

Manually input, use `workcost` to store the cost of equipment in each location, and use `trancost` to store the cost of equipment transfer between different task locations. For example, enter the following data during testing

equipment	place1	place2	place3	place4	place5
X001	272	116	105	156	227
X002	213	232	270	263	232
X003	113	283	180	142	276

Figure 11: Cost Matrix between two sides, objects and tasks

equipment	place1	place2	place3	place4	place5
place1	0	126	227	256	299
place2	275	0	100	128	233
place3	169	249	0	263	109
place4	223	242	200	0	177
place5	173	206	265	191	0

Figure 12: Cost Matrix among tasks

The code is as follows:

```
#Input the data then get the solution for problem
workcost=[[272,116,105,156,227],
[213,232,270,263,232],
[113,283,180,142,276]]#cost of equipment's work
trancost=[[0,126,227,256,299],
[275,0,100,128,233],
[169,249,0,263,109],
[223,242,200,0,177],
[173,206,265,191,0]]#cost of transition from place to place
```

2.2 Output description

The output of the test is

```
tractor 1 should go to site 2 with cost 116
tractor 1 then should go to site 3 with cost 100
tractor 1 then should go to site 5 with cost 109
project of tractor 1 cost 325
tractor 2 should go to site 1 with cost 213
project of tractor 2 cost 213
tractor 3 should go to site 4 with cost 142
project of tractor 3 cost 142
total cost= 680.0
Wall time: 127 ms
```

Figure 13: Output

Consistent with the expected results. Expressed in the following form by Assignment list

- tractor 1: place2 $\longrightarrow$place3 $\longrightarrow$place5,cost $= 116 + 100 + 109 = 325$
- tractor 2: place1,cost $= 213$
- tractor 3: place4,cost $= 142$

So the Total Cost $= 680$.

Assignment graph is represented as follows:

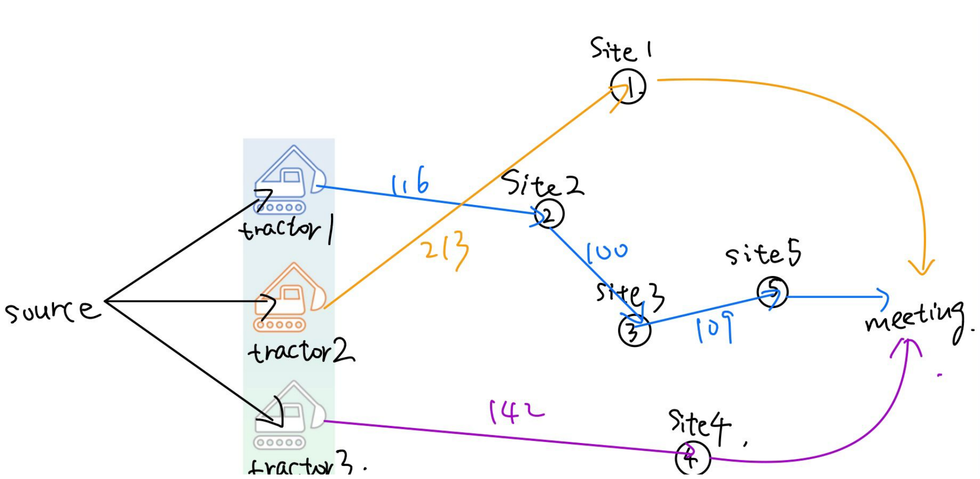

Figure 14: Assignment Graph

So I solved this variant of the assignment problem using `ortools`.

My code is open source and available on Github[1].

[1] https://github.com/He-jiazhi/Optimization-Algorithms-Course-Project/tree/main/Project2-AssignmentProblem