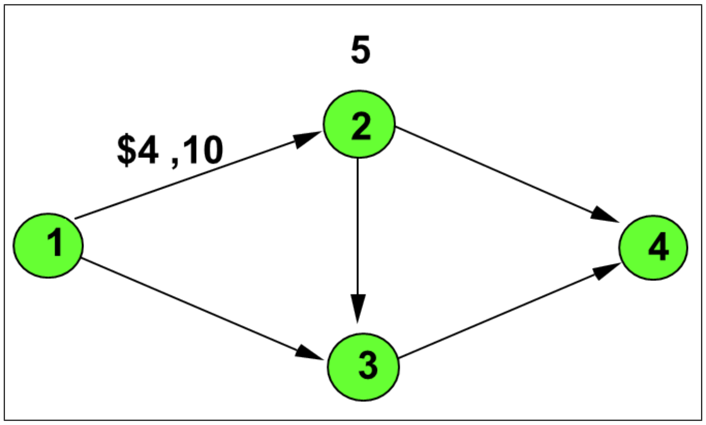

Figure 13: The arc capacity u_{ij} and the cost c_{ij} per unit flow through arc (i, j)

Remark 1.2. *For many applications, b_i and u_{ij} will have integer values, and implementation will require that the flow quantities x_{ij} also be integer. Fortunately, just as for the transportation problem, this outcome is guaranteed without explicitly imposing integer constraints on the variables because of the basic feasible solution (BFS) being integer valued.*

Remark 1.3. *If we constrain the flow value to be 1 (demand at the terminal node t), and all capacities are set to 1, it is pretty clear that the minimum cost flow problem is equivalent to finding the shortest path.*

Remark 1.4. *The transportation problem (or the linear assignment problem) is a network-flow model without intermediate locations, see Fig. 14. To formulate the transportation problem presented as a minimum cost flow problem, a supply node is provided for each source, as well as a demand node for each destination, but no transshipment nodes are included in the network. All the arcs are directed from a supply node to a demand node, where distributing x_{ij} units from source i to destination j corresponds to a flow of x_{ij} through arc (i, j). The cost c_{ij} per unit distributed becomes the cost c_{ij} per unit of flow. Since the transportation problem does not impose upper bound constraints on individual x_{ij}, Eq. (13) does not apply.*

1.4 Linear Assignment Problem

The linear assignment problem can be described by the way of the following example. Typically, we have a group of n applicants applying for n jobs, and the non-negative cost c_{ij} of assigning the i-th applicant to j-th job is known. The objective is to assign one job to each applicant in such a way as to achieve the minimum possible total cost. The problem is also known as weighted bipartite matching. Define binary variables x_{ij} such that x_{ij}= 1, it indicates that we should assign applicant i to job j. Otherwise (x_{ij} = 0), we should not assign applicant i to job j. The mathematical formulation is as follows:

$$
\begin{aligned}
&\min & &\sum_i \sum_j c_{ij} x_{ij} & & & (14)\\
&\text{subject to} & &\sum_j x_{ij} = 1 & &(\text{for each } i) & (15)\\
& & &\sum_i x_{ij} = 1 & &(\text{for each } j) & (16)\\
& & &x_{ij} \in \{0, 1\} & & & (17)
\end{aligned}
$$

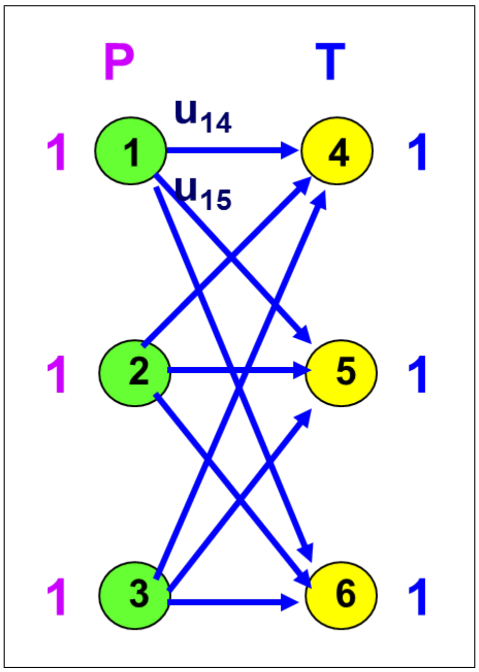

Figure 14: Network for linear assignment problem with P (people) set and T (task) set and the cost c_{ij} (or utilities u_{ij}) for assignment.

Formulate the following linear assignment problem:

The manager of Hotel in Johannesburg have four workers: Agness, Rose, Joyce and Zainab. The manager needs to have one of them clean the bathroom of the hotel, another sweeps the floor of the hotel, the third to be a receptionist and the last person to cook food for the guests and management, but they each demand different pay for their different tasks. The table below represents the costs c_{ij} of the workers i doing the jobs (tasks) j, where ZAR is the South African Rand.

The above problem is the of perfect matching in a bipartite graph since for the bipartite graph $G = (V, E)$ the matching $M \subset E$ is such that every $i \in V$ there is exactly an edge $e \in M$, i.e. every vertex has at exactly one incident edge in M. Hence the problem (14)-(17) can also be written as the maximum weight perfect matching

Table 1: Data showing the cost of workers doing different jobs

	Jobs			
Workers	Cleaning	Sweeping	Receptionist	Cooking
Agness	ZAR9000	ZAR7500	ZAR7500	ZAR800
Rose	ZAR3500	ZAR8500	ZAR5500	ZAR6500
Joyce	ZAR12500	ZAR9500	ZAR9000	ZAR10500
Zainab	ZAR4500	ZAR11000	ZAR9500	ZAR11000

$$\max \quad \sum_e c_e x_e \tag{18}$$

$$\text{subject to} \quad \sum_{\text{e incident to } i} x_e = 1 \qquad (\text{for each } i \in V) \tag{19}$$

$$x_e \in \{0, 1\} \tag{20}$$

where c_e is the weight or utility of the edge e. For the case when the matching is not perfect the constraint (19) is replaced with

$$\sum_{\text{e incident to i}} x_e \leq 1, \, \forall i \in V,$$

then this problem is known as bipartite matching problem.

Consider that a matching gives an assignment of people to tasks. You want to get as many tasks done as possible, i.e. you want a maximum matching, one that contains as many edges as possible. Create an instance of bipartite matching. Then create an instance of a network flow where the solution to the network flow problem can easily be used to find the solution to the bipartite matching. The edges used in the maximum network flow will correspond to the largest possible matching!

1.5 Minimum Spanning Tree Problem

A spanning tree is a tree (i.e., a connected acyclic graph) that spans (touches) all the nodes of an undirected network[4]. The cost of a spanning tree is the sum of the costs (or lengths) of its edges. In the minimum spanning tree problem, we wish to identify a spanning tree of minimum cost (or length). A spanning tree of a graph has the following features:

- Every node in the network is connected to every other node by a sequence of arcs from the subnetwork, where the direction of the arcs is ignored.
- The subnetwork contains no loops, where a loop is a sequence of arcs connecting a node to itself, where again the direction of the arcs is ignored.

A subnetwork that satisfies the above two properties is called a spanning tree.

[4]Minimum spanning tree for directed graph can also be constructed but it has limited applications.