

Tutorial 4: Flow

Artificial Intelligence

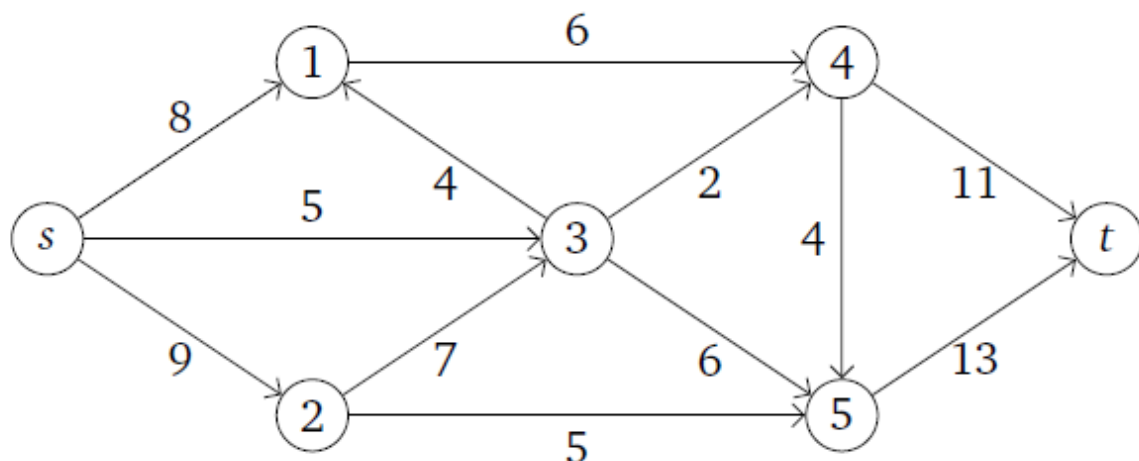
G.Guérard

Les étudiants doivent faire des groupes de 3-4 afin de faire un brainstorming pour chaque exercice. Il n'est pas demandé aux étudiants d'avoir une connaissance parfaite de la théorie, vous devez les guider afin qu'ils trouvent une solution aux problèmes. Vous pouvez commencer le TP par de simples questions de cours.

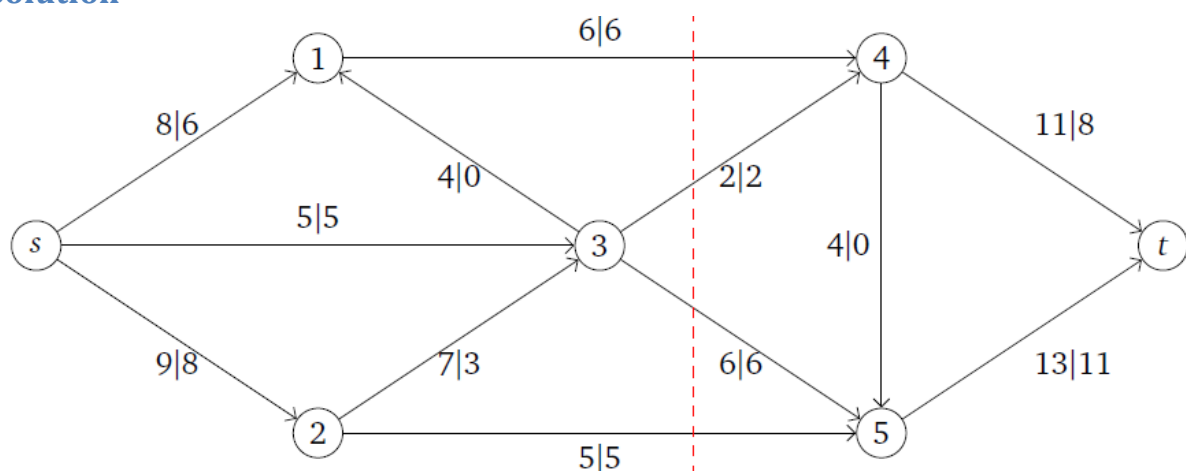
Flow

Exercise 1

Using the Ford-Fulkerson method, compute a maximal flow in the following network:

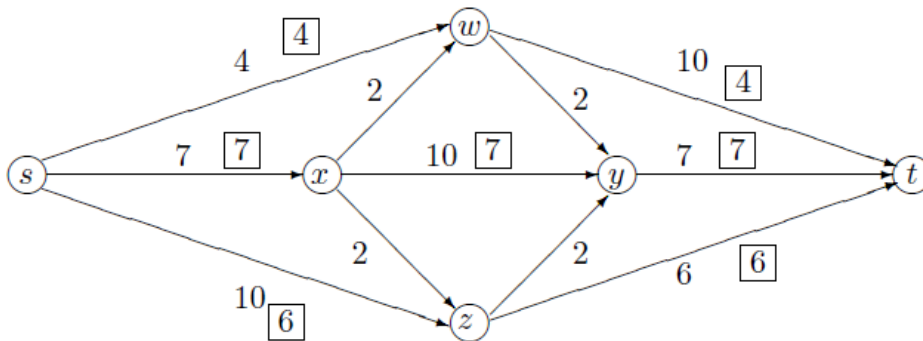


Solution



Exercise 2

The figure below shows a flow network on which an s-t flow is shown. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbers have no flow being sent on them.) What is the value of this flow? Is this a maximum s-t flow in this graph? If not, find a maximum s-t flow. Find a minimum s-t cut. (Specify which vertices belong to the sets of the cut.)



Solution

The value of this flow is 17. No, this isn't a maximum flow. We can find a maximum flow by looking at the residual network. In this case, from the given flow, we can get a max flow with only a single augmenting path, s-z-y-x-w-t. So the maximum flow has value 19.

We can find a minimum cut using the residual network for the final maximum flow. Starting at s, find all vertices that are reachable from s in the residual network. This forms one set in the minimum cut. The other vertices form the other set part of the cut. So a minimum cut in this case are the two sets {s; z} and {x; w; y; t}.

You can check that the capacity of this cut (i.e. the total capacity of the edges from {s; z} to {x; w; y; t}, consisting of the directed edges (s; w); (s; x); (z; y); and (z; t) is 19, as the Max Flow/Min Cut Theorem guarantees.

Exercise 3

Describe an efficient algorithm that finds a new maximum flow if the capacity of a particular edge increases by one unit. Describe an efficient algorithm that finds a new maximum flow if the capacity of a particular edge decreases by one unit.

Solution

Suppose that the edge from u to v increases its capacity by one. From the previous maximum flow F, just make a new iteration of Ford-Fulkerson algorithm with the modified graph: The residual flow graph increases the capacity of edge (u; v) with one. Make a graph search to see if there is any path in the residual flow graph along which the flow can increase. If there is one, there must be a flow of

size one (because all flows are integers). If there is no flow in the residual flow graph F is still the maximum flow.

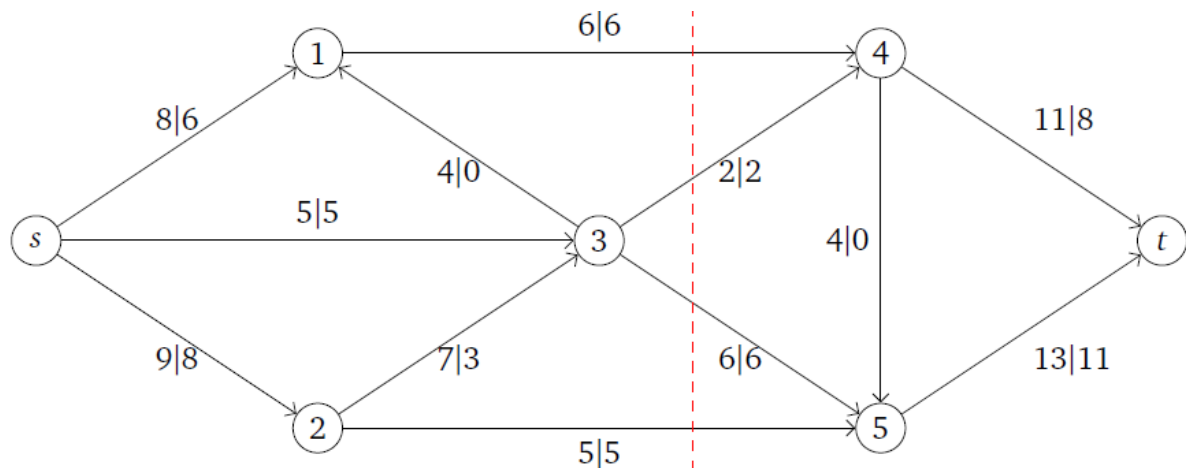
Suppose that the edge from u to v decreases its capacity by one. If the previous maximum flow F didn't use the full capacity of $(u; v)$ such change doesn't count at all. Otherwise we update the flow as follows:

Since the flow entering u is one unit more than the flow leaving it and the flow entering v is one unit less than the flow leaving it, there is no way we must transfer a unit flow from u to v . Thus, search in the residual flow graph for a path from u to v along which the flow can increase by one. If there is such a path we update F with the flow.

If there is no such a path, we must reduce the flow from s to u and from v to t with one unit. We do this by finding a path from u to s in the residual flow graph along which the flow can increase by one and a path from t to v along which flow can increase by one. (There must be such paths, because we had a flow from s to t via $(u; v)$.) Then, update F with these two flows.

Exercise 4

Considering the following flow, how to increase the flow?



Solution

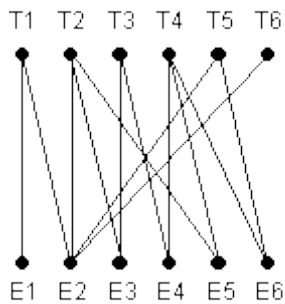
The minimal cut is done, so we know bottlenecks on the graph. The sum of capacities at source is 22, and 24 at the sink, we conclude that we can't have more than 22. So we need to find a path from 1 to t to increase the flow by 2, and a path from 2 to t to increase the flow by 1. The previous exercise gives an algorithm an idea to reach this goal. We can increase capacities on minimal cut edges.

Assignment

Exercise 5

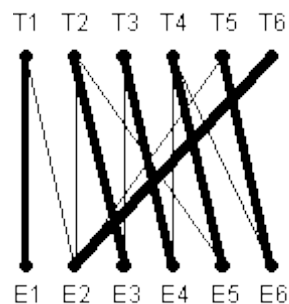
In the bipartite graph opposite, the vertices $T1 \dots T6$ represent workers and vertices $E1 \dots E6$ represent jobs. An edge connects a worker with an employment if the worker has the necessary

qualifications to occupy this employment. How allocate jobs to the workers to minimize the number of unemployed persons?



Solution

Assign a job to a person is equal to "select" an edge. T_i are connected to a fictive source with a capacity of 1 (same for the E_i to a fictive sink). The edges of the bipartite graph have a capacity of 1. It then searches a maximum flow.



Exercise 6

Suppose you are organizing a conference where researchers present articles they have written. Researchers who want to present an article send a paper to the conference organizers. The conference organizers have access to a committee of reviewers who are each willing to read up mB articles each. Each paper submission gets reviewed by up to mA reviewers. Moreover, each submission has a particular topic and each reviewer has a specialization for a set of topics, so papers on a given topic only get reviewed by those reviewers who are experts on that topic. The conference organizers need to decide which reviewers will review each article (or equivalently, which articles will be reviewed by which reviewers). Explain how they could use a flow network to solve this problem.

Solution

There is a set A of articles and a set B of reviewers. Add a directed edge (α, β) to the graph if some article α in A could be reviewed by some reviewer β in B , namely the article is on a topic that the reviewer is an expert in. Set the capacity of that edge to be 1. Add a vertex s (source) and a vertex t (terminal). For each α in A , add an edge (s, α) of capacity mA . For each β in B , add an edge (β, t) of capacity mB . Run Ford-Fulkerson to get the maximum flow and compute the corresponding cut (A, B) . This gives the set of edges which correspond to the assignment of articles to reviewers.

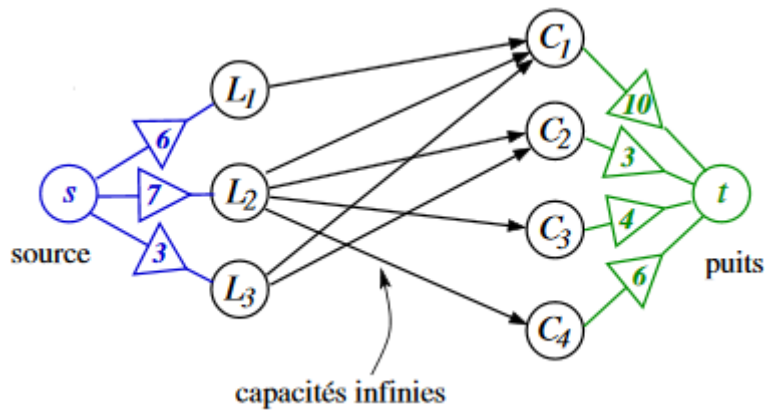
Exercise 7

Let be a set of machines and tasks. A machine L_i can be used at most a_i time. A task C_j can be used at most b_j time. Table of supply and demand represents machines and tasks, and the ability to perform a task on a given machine (blank). Trouver le maximum d'affectations possibles.

	C_1	C_2	C_3	C_4	a_i
L_1					6
L_2					7
L_3					3
b_j	10	3	4	6	

Solution

Create bipartite graph machines / tasks, and then do a fictive source (fictive sink) with capacities of arcs a_i (and b_j). Then solve the flow problem.



	C_1	C_2	C_3	C_4	a_i
L_1	6				6
L_2	4	0	0	3	7
L_3	0	3			3
b_j	10	3	4	6	