

# 4005-800 ALGORITHMS

## HOMEWORK 6

Christopher Wood

May 3, 2012

### PROBLEM 1 - 16.1-2.

*Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.*

#### Solution.

In the context of the activity selection problem, the notion of a greedy algorithm is one that chooses an activity at each iteration such that we are left with as much resources (time) for other activities as possible. Thus, by choosing the last activity  $a_k$  to start (assuming that all activities are sorted in monotonically decreasing order by start time), we are guaranteed to leave as much time for other compatible activities as possible that be selected before the start of  $a_k$ . This is because choosing an activity with an earlier start time would consume more resources, and therefore is not greedy.

We can also show that this approach yields an optimal solution to the original problem solution. Consider, for example, any non-empty subsequence of activities  $S' = \langle a_1, a_2, \dots, a_n \rangle$ , which is in monotonically decreasing order based on start times. We now prove that the largest set of compatible activities for  $S'$  is  $1 + (\text{the largest set of compatible activities in } \langle a_2, a_3, \dots, a_n \rangle)$ . In other words,  $a_1$  is included in some maximum-size subsequence of mutually compatible activities of  $S'$ .

*Proof.* Let  $A$  be the largest set of mutually compatible activities in  $S'$ , and let  $a_j$  be the activity with the latest start time. If  $a_j = a_1$ , then the result follows immediately. If  $a_j \neq a_1$ , then consider  $A' = (A - \{a_j\}) \cup \{a_1\}$ . Note that  $A'$  is a mutually compatible set of activities since  $s_1 \geq s_j$ . That is, there is more time before  $a_1$ , and by picking  $a_j$  there is nothing after (since  $a_j$  has the latest start time). Now, since  $|A'| = |A|$ , the result follows immediately. ■

### PROBLEM 2 - 26.1-4.

*Let  $f$  be a flow in a network, and let  $\alpha$  be a real number. The **scalar flow product**, denoted  $\alpha f$ , is a function  $V \times V \rightarrow \mathbb{R}$  defined by*

$$(\alpha f)(u, v) = \alpha \times f(u, v)$$

*Prove that the flows in a network form a **convex set**. That is, show that if  $f_1$  and  $f_2$  are flows,*

then so is  $\alpha f_1 + (1 - \alpha)f_2$  for all  $\alpha$  in the range  $0 \leq \alpha \leq 1$ .

**Solution.**

To show that flows in a network form a **convex set**, we consider the expression of scalar flow products  $\alpha f_1 + (1 - \alpha)f_2$ , assuming that  $f_1$  and  $f_2$  are valid flows in the same network. Manipulating this expression yields the following:

$$\begin{aligned}\alpha f_1 + (1 - \alpha)f_2 &= \alpha f_1 - \alpha f_2 + f_2 \\ &= \alpha(f_1 - f_2) + f_2\end{aligned}$$

Now, we consider this expression on a case-by-case basis.

**Case 1:**  $\alpha = 0$

When  $\alpha = 0$ , the expression  $\alpha(f_1 - f_2) + f_2$  reduces to  $f_2$ , which is known to be a valid flow.

**Case 2:**  $\alpha = 1$

When  $\alpha = 1$ , the expression  $\alpha(f_1 - f_2) + f_2$  reduces to  $f_1$ , which is known to be a valid flow.

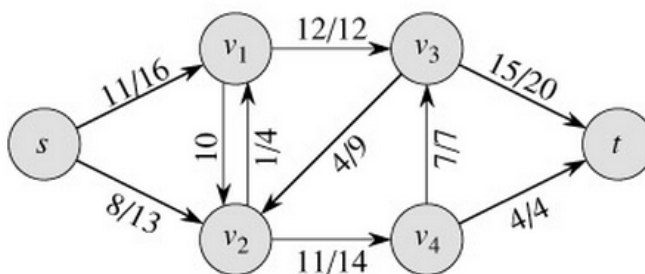
**Case 3:**  $0 < \alpha < 1$

TODO

**PROBLEM 3 - 26.2-2.**

In Figure 26.1(b), what is the flow across the cut  $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$ ? What is the capacity of this cut?

**Solution.** The graph in Figure 26.1(b) is shown below.



By considering the cut  $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$ , we know the edges that cross the cut boundary are  $\{(u, v) : u \in S, v \in T, (u, v) \in E\} = \{(s, v_1), (v_2, v_1), (v_3, v_2), (v_4, v_3), (v_4, t)\}$ . The net flow of this  $s - t$ -cut is then as follows:

$$\begin{aligned}f(S, T) &= f(s, v_1) + f(v_2, v_1) + f(v_3, v_2) + f(v_4, v_3) + f(v_4, t) \\ &= 11 + 1 + (-4) + 7 + 4 \\ &= 19\end{aligned}$$

Since the capacity of the cut is defined as the sum of the edge capacities from all vertices  $u \in S$  to  $v \in T$ , we have the following capacity:

$$\begin{aligned} c(S, T) &= c(s, v_1) + c(v_2, v_1) + c(v_4, v_3) + c(v_4, t) \\ &= 16 + 4 + 7 + 4 \\ &= 31 \end{aligned}$$