Algorithms Seminar

Jake Christensen

April 7, 2015

1 April 3

The questions posed to the reading group are as follows:

Problem 1. Let G be a graph with demands at all of its edges and vertices. Let k be the number of machines we can place at a vertex. When placed a vertex, a machine can service the demand for that vertex, all edges incident to it, and its neighbors. Find an expression that gives the time required to service all demands and the placement of guards that minimizes that time.

Special Case: k = n, demands on nodes only.

If k equals n (i.e. every vertex gets a guard), the maxspan is equal to $max\frac{d_i}{\delta_i+1}$ across all i, where d_i is the demand for vertex i. This is because each node with simultaneously serviced by itself and all its neighbors. See Figure 1 for an example.

Problem 2. Let G be a graph with demands at all of its edges and vertices. Let k be the number of machines we can place at a vertex. We have one machine that can move between vertices. When at a vertex, it can service the demand for that vertex, all edges incident to it, and its neighbors. Find the path that minimizes the makespan (time to service all demands).

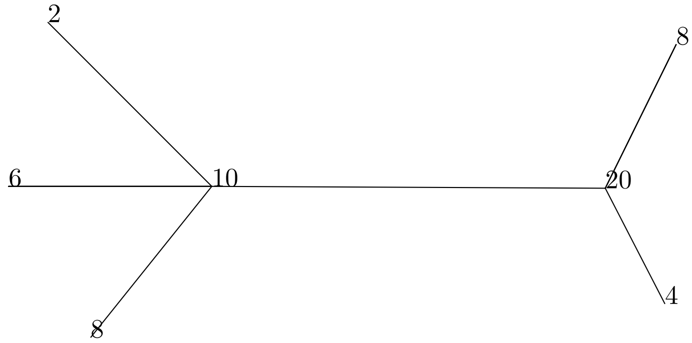

Figure 1: Makespan: $\frac{20}{4} = 5$

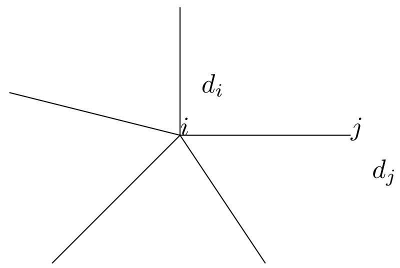

Figure 2: Generalized

Problem 3. Without parallel servicing, the static problem becomes a scheduling problem, with each machine being able to work on the demand for their node or the demand for any one of their neighbors. The goal is to satisfy the demand as soon as possible.

LP:x_{ij} is the time vertex i spends doing work on j.

Minimize C subject to $\sum j \in I_i x_{ij} \leq C \forall i$ and $\sum i \in I_i x_{ij} \geq d_j \forall j$.

For a visual, see Figure 2.

This notation also with demands on edges.

Problem 4. In the special case of a tree, is there a combinatoric algorithm to find the minimum dominating set in linear time? This problem has been solved. (Shika found it, I can't seem to find the same paper).

Problem 5. Using k machines where $k > k_{min}$, find optimal positioning and schedule.

The LP is the same as before, but with $x_{ij} = 0$ if there is no machine at i.

There are $O(n)$ states, where a state is described as (u, v, b), where b is a bit that represents a guard being at u.

$f(u, v, I) = min\{...\}$, $O(1)$ choices.

Problem 6. Mobile machine without parallel processing.