SOME OF THESE PROBLEMS MAY BE DIFFICULT.

Metrical Task Systems

All problem so far introduced this semester are special cases of the much more general *metrical task system* problem.

A metrical task system consists of a set $\mathcal{M}$ of *states*, a metric d on $\mathcal{M}$, a set $\mathcal{R}$ of *tasks*, and a *cost* function *cost* on $\mathcal{R} \times \mathcal{M}$, where $0 \leq cost(r, x) \leq \infty$ for each $r \in \mathcal{R}$ and $x \in M$.

Before each step, there is a current state, say, $x \in \mathcal{M}$. There is a request $r \in \mathcal{R}$. The algorithm must change state to a new state $y \in \mathcal{M}$. The cost to the algorithm is the cost of moving, plus the cost of serving the request at y, namely:

$$d(x, y) + cost(r, y)$$

Competitiveness is defined in the usual way.

WFA for a Metrical Task System

The work function algorithm (WFA) is defined for any metrical task system, on page 132 of Borodin & El-Yaniv. It is also proved that if a MTS has n states, WFA is $(2n - 1)$-competitive.

Every MTS has an *opt-graph*. This is a graph where each node is a reachable *offset function*. For all the examples we have seen so far, the opt-graph is finite. Find an MTS with only two states that has an infinite opt-graph.

The Underemployed Watchman

Using the above definition, show that the underemployed watchman problem is a special case of the metrical task system problem.

What you need to do is identify the set of states, the metric, the set of tasks, and the cost function. Draw the opt-graph.

Paging

Using the above definition, show that, for any given k, the k-paging problem is a special case of the metrical task system problem.

What you need to do is identify the set of states, the metric, the set of tasks, and the cost function. Draw the opt-graph.

Other Metrical Task Systems

The Extremely Lazy Underemployed Watchman

The watchman is so lazy that his discomfort from changing chairs is greater than the inconvenience of leaning over to push the button on the opposite wall. Let us say that he pays 1 to change

chairs, and ε to push the button on the opposite wall without changing chairs, where $0 < \varepsilon \leq 1$. As before, it costs zero to push the near button.

Draw the opt-graph for $\varepsilon = \frac{1}{N}$, where N is a positive integer. The larger N is, the more states there are in the opt-graph. As you see the pattern, you'll be able to represent a graph with as many states as necessary. What happens if ε is irrational?

Find the deterministic and randomized competitiveness of this problem for $\varepsilon = \frac{1}{N}$ where N is an integer. What is the limiting value of this competitiveness as N grows without bound?

The Remote Read Caching Problem

Assume that there are k pages in the cache, and the total number of pages is m. If a fault occurs, a page can be ejected and a new page brought in. This costs 1. Alternatively, we can remotely read the requested page at a cost of ε, where $0 < \varepsilon \leq 2$.

You would like to find the deterministic and randomized competitiveness of this problem for $\varepsilon = \frac{1}{N}$ where N is an integer, and then find the limiting value of this competitiveness as N grows without bound.[1] Let's take the smallest non-trivial case of the remote read caching problem, namely $k = 2$ and $m = 3$. Can you answer the questions for that one example?

You might want to restrict your attention to the case that $\varepsilon = \frac{1}{N}$, where N is an integer. Start with $N = 1$ and then $N = 2$. The limiting case is where N is extremely large.

Different Adversaries

Our textbook discusses various adversaries. The three we will consider are:

1. The *strong* adversary. This adversary has complete knowledge of the past, present, and future.

2. The *adaptive online* adversary. This adversary has knows our past sequence of services, but does not know our present service. He must serve each request without knowing our service of that request, but then is informed immediately after the step.

3. The *oblivious* adversary. This adversary is never told what services we choose.

The competitiveness of any deterministic algorithm against any adversary is the same, so when we discuss deterministic online algorithms, we do not vary the adversary.

A randomized algorithm can perform better against a weaker adversary.

Given an online problem, let C_{str} be the best possible competitiveness of any randomized algorithm against the oblivious adversary, let C_{adon} be the best possible competitiveness of any randomized algorithm against the oblivious adversary, and let C_{obl} be the best possible competitiveness of any randomized algorithm against the oblivious adversary. let C_{det} be the best possible competitiveness of any deterministic algorithm against any adversary.

1. Prove the following statements:

[1] I do not actually expect you to solve those problems.

(a) $C_{\text{obl}} \leq C_{\text{adon}} \leq C_{\text{str}}$

(b) $C_{\text{str}} = C_{\text{det}}$

(c) $C_{\text{str}} \leq C_{\text{obl}} \cdot C_{\text{adon}}$ [Karp and Wigderson]

2. Consider the game shown in Figure 1. There are two players, **Black** (the adversary) and **Red** (us). The state of the game at any point in time is I, II, III, IV, V, or VI. Think of a "pebble" as being located on the current state. There is also a **Referee** whose job it is to move the pebble and to keep track of the money. **Red** is never told how much money is paid. If the state is I or II, **Black** chooses one of the four states III, IV, V, or VI, and the pebble moves along the edge to that chosen state. **Black** is debited the amount shown on that edge. At the same time, the **Referee** tells **Red** the letter (a or b) on that edge. **Red** is not told what state the pebble is moved to. If the state is III, IV, V, or VI, **Red** chooses one of the two states I or II, and the pebble moves along the edge to that chosen state. **Red** is debited the amount shown on that edge. **Black** is not told what the new state is. Each player pays in his own kind of money (dollars, pounds, euros, or whatever).

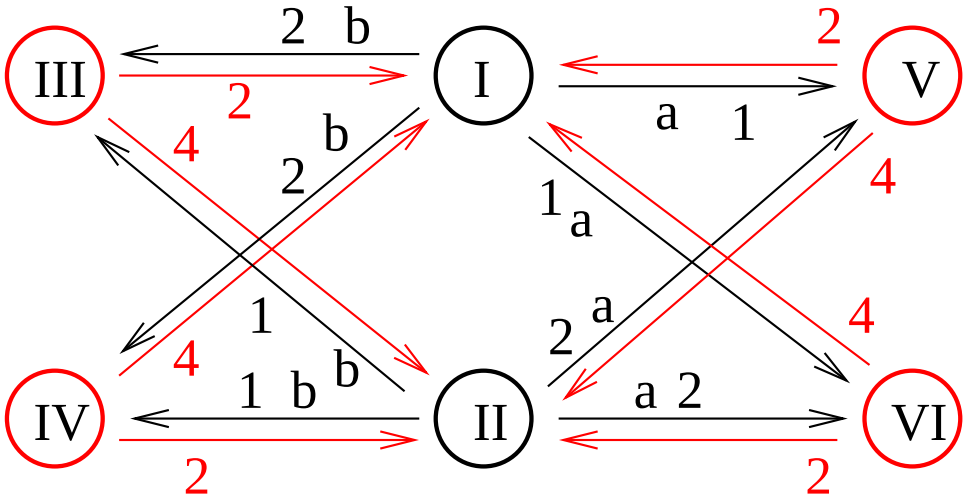


Figure 1: A Game

Find C_{obl}, C_{adon}, and C_{str} for this game. The strong adversary knows what moves we will make in the future in response to any give request sequence. The adaptive online adversary knows where the pebble is at all times, but does not know where we will move it in the future. The oblivious adversary does not know where the pebble is when it is his turn.

3. Find C_{adon} for the underemployed watchman problem.

4. Find C_{adon} for the paging problem with $k = 2$, where there are a total of only 3 pages altogether.

The k-Server Problem

1. Show how, for any given metric space M and any k, the k-server problem on M is a metrical task system problem.

Consider the special case where $k = 2$ and $M = \{x, y, z\}$.

2. Draw the opt-graph for the uniform case, *i.e.*, $d(x, y) = d(x, z) = d(y, z) = 1$.

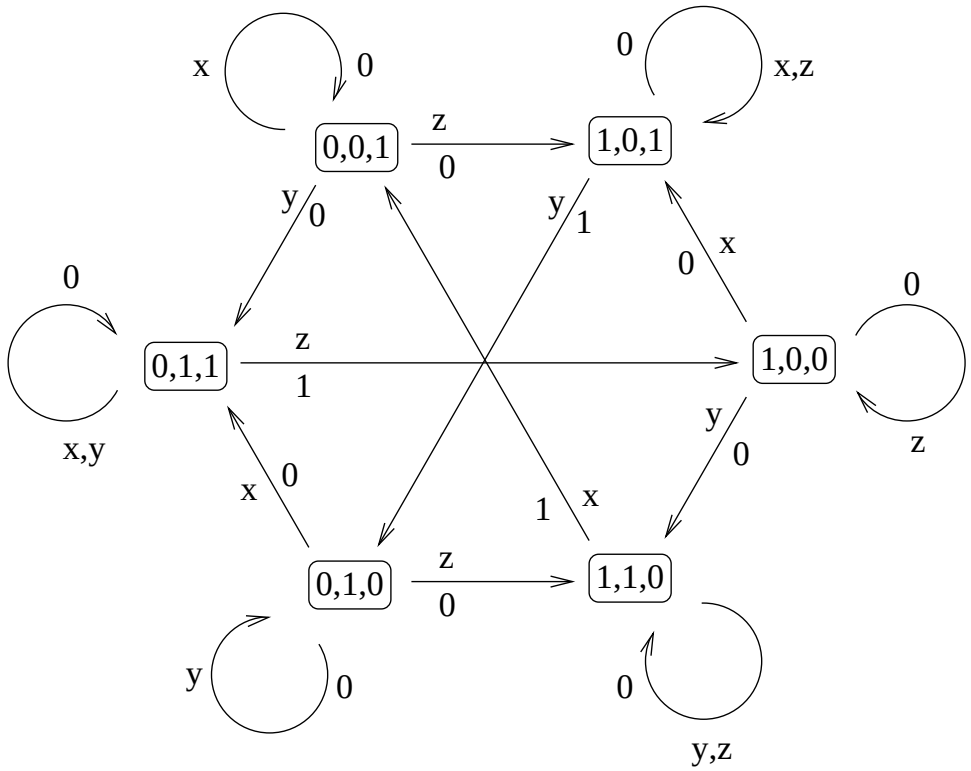


Figure 2: Opt-graph for the 2-server problem in a uniform metric space of three points

3. Draw the opt-graph for the 3-4-5 triangle case, *i.e.*, $d(x, y) = 3$, $d(x, z) = 4$, and $d(y, z) = 5$.

What you need to do is identify the set of states, the metric, the set of tasks, and the cost function.

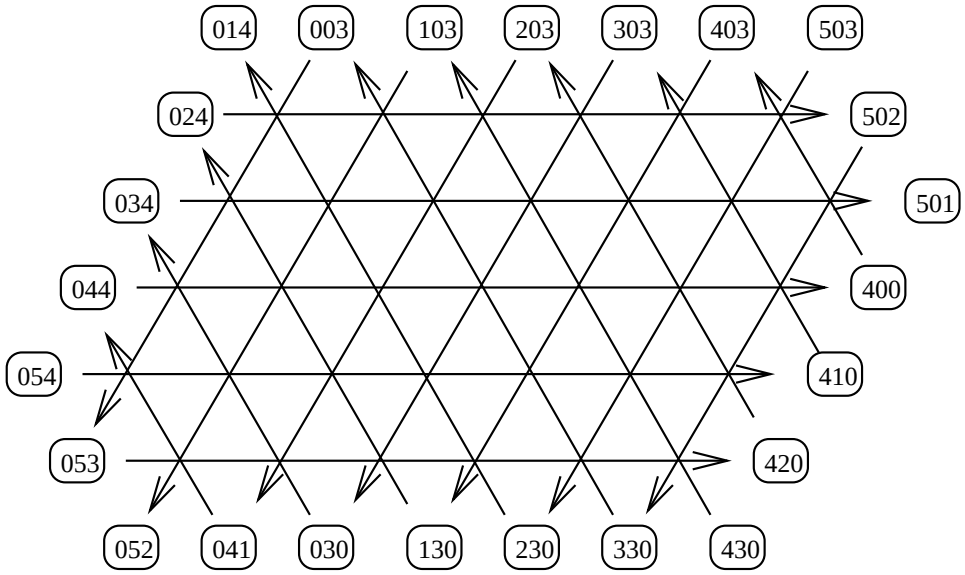


Figure 3: Opt-graph for the 2-server problem in 3-4-5 metric space. there are 24 nodes. Labels on the edges are omitted. Not all edges are shown.