Institut for Matematik og Datalogi
Syddansk Universitet

May 16, 2017
JFB

On-Line Algorithms – F17 – Lecture 19

Lecture, May 9

We will covered much of the article "The Advice Complexity of a Class of Hard Online Problems", Joan Boyar, Lene M. Favrholdt, Christian Kudahl, Jesper W. Mikkelsen. *Theory of Computing Systems*, First Online 2016. The publication is also available through the course's homepage.

Lecture, May 16

We will finish the article "The Advice Complexity of a Class of Hard Online Problems". We will also discuss the exam and possibly begin on the problems for May 17.

Problems for May 17

1. Consider the following advice for the makespan scheduling problem on N identical machines: The index of one machine which should not be used, except for items with load at least $N/2$.

   (a) Specify how many bits of advice are used, and define an algorithm to use it effectively (specify what the algorithm should do with jobs with different loads).

   (b) Show the effect this advice would have on the sequence used in the lower bound proof of Theorem 12.1 in the textbook. How would your algorithm schedule this sequence and what would the performance ratio be, compared to OPT? (Note that if your algorithm does not do better on this sequence than GREEDY does, you should redefine your algorithm.)

   (c) Give an example input sequence showing that the performance ratio compared to OPT is still at least $2 - \frac{1}{N}$.

(d) Give an example input sequence showing that the performance ratio compared to OPT is at least 2. Can the performance ratio be even higher?

2. Consider the graph $G = (V, E)$, where $V = \{v_1, v_1, v_3, v_4, v_5, v_6, v_7\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), (v_1, v_6), (v_7, v_2), (v_7, v_3), (v_7, v_4), (v_7, v_5), (v_7, v_6)\}$. Find an optimal vertex cover.

   Find a $(7, 5, 2)$-covering design of size 3.

   Show how to use this covering design for the graph G. What is your performance ratio?

3. Show that Maximum Matching is in AOC, but Simple Knapsack is not.

4. Show that CLIQUE is AOC-Complete.

5. Show that the Independent Set problem is NP-hard. For Independent Set, the input is a graph, $G = (V, E)$, and the output is a subset $I \subseteq V$ such that for any two vertices $u, v \in I$, there is no edge between them in G, i.e., $(u, v) \notin E$. Such a set I is called an *independent set*. The goal is to output a maximum size independent set. Reduce from 3-SAT.

   Show that the Vertex Cover problem is NP-hard. Reduce from Independent Set.

6. Show that the Cycle Finding problem can be solved in polynomial time. The input is a graph, and the goal is to find a smallest subset of the vertices containing a cycle.