CSC 2420 Spring 2017, Assignment 1

Due date: February 13, 2017 (start of class)

It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should try to improve upon the presentation of the result.

If you would like to discuss any questions with someone else that is fine BUT at the end of any collaboration you must spend at least one hour playing video games or watching two periods of Maple Leaf hockey or maybe even start reading a good novel before writing anything down.

Unless stated otherwise, all subquestions are worth 10 points. If you do not know how to answer a question, state "I do not know how to answer this (sub) question and you will receive 20% (i.e. 2 of 10 points) for doing so. You can receive partial credit for any reasonable attempt to answer a question BUT no creiodt for arguments that make no sense.

In class I can clarify any questions you may have about this assignment.

1. Consider the makespan problem for the identical machines model with m machines. We sketched the proof that the worst case *competitive ratio* $\frac{C_{Greedy}}{C_{OPT}}$ of the natural online greedy algorithm is $\leq 2 - \frac{1}{m}$ for every input. We also gave an example showing that this ratio is "tight" (i.e. cannot be improved) for this algorithm.

   - Finish the proof that was sketched in Lecture 1; that is, use the fact that $C_{OPT} \geq \sum_{1 \leq i \leq n} p_i/m$ for any sequence of n input jobs and $C_{OPT} \geq p_i$ where job J_i has "load" p_i.
   - Argue for $m = 2$ machins that *any* (not necessarily greedy) online algorithm would have competitive ratio no better than $\frac{3}{2}$ so that the above bound is tight for $m = 2$ for any online algorithm.
   - Suppose jobs now are transient with a job arriving at some time a_i and departing at some (unknown at arrival) time d_i. Does the same competitive ratio hold at any time during the execution of the Greedy algorithm? If no, then give a counter example; if yes, then argue why the proof still applies. Here as before, a job must be scheduled on one of the m machines at the time of arrival.
   - Consider the original makespan problem (i.e. permanent jobs) but now in the random order ROM model. Show that the (expected) approximation ratio $\frac{E[C_{Greedy}]}{C_{OPT}} \geq 2 - \epsilon$ for any $\epsilon > 0$. NHere the expectation is with respedct to the uniform distribution on input arrival order.
     If you cannot prove the stated claim then prove any ratio greater than 1. Hint: generalize the nemesis sequence for the adversarial competitive ratio.

2. Consider the knapsack problem with input items $\{(v_1, s_1), \ldots, (v_n, s_n)\}$ and capacity C. Without loss of generality the sizes s_j of all items are at most C. Consider the following "natural" greedy algorithms which initially sorts the input set and then schedules greedily (i.e. takes the item if it fits). For each algorithm provide input instances which show that these algorithms will not achieve a c-approximation for any constant c.
   Note: For definiteness, assume all input values are integral which in principle could make an inapproximation result harder. But here it should be easy to derive appropriate integral examples.

   - *Greedy by value*: Sort the items $I_j = (v_j, s_j)$ so that $v_1 \geq v_2 \ldots \geq v_n$.
   - *Greedy by size*: Sort the items so that $s_1 \leq s_2 \ldots \leq s_n$.
   - *Greedy by value-density*: Sort the items so that $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \ldots \geq \frac{v_n}{s_n}$

3. For the knapsack problem, consider the algorithm that returns the maximum of "Greedy by value" and "Greedy by value-density" as defined in the previous question. Return the better of the two solutions. Show that this algorithm is a 2-approximation for the knapsack problem by showing the following:

   - Let item t be the first item that is rejected by Greedy by value density. That is, when $v_1/s_1 \geq v_2/s_2 \ldots \geq v_n/s_n$ then $\sum_{i=1}^{t-1} s_i \leq C$ and $\sum_{i=1}^{t} s_i > C$ where C is the capacity bound. (We can assume there is such a t since otherwise if all items fit in the knapsack then any greedy algorithm will be optimal.) Show that $\sum_{i=1}^{t} v_i \geq OPT$
   - Show how the above fact implies that the algorithm that returns the maximum of "Greedy by value" and "Greedy by value-density" is a 2-approximation.

4. The following refers to the underlying optimization problem in the multi-minded conbinatorial auction (CA) problem and its relation to the maximum weighted independent set problem in $(k+1)$-claw free graphs.

   There are n agents and each agent i is intersted in certain desireable subsets $S_i^1, \ldots S_i^{n_i}$ where each subset $S_i^j \subset U$ has size at most s and value v_i^j for agent i. The goal is to allocate at most one desireable subset to each agent so that the allocated subsets are disjoint and the total value of all allocated subsets is maximized.

   - Show how to formulate the above problem in terms of the maximum weighted independent set problem in $(k+1)$-free graphs for an appropriate k.
   - Consider the weighted maximum independent set problem in $(k+1)$-claw free graphs and the "greedy-by-value" approximation algorithm for this problem. Use a "charging argument" to show that this algorithm provides an k approximation. What conclusion can you state for approximating the above multi-minded CA optimization problem?

5. This problem concerns the weighted interval scheduling problem (WISP) on two machines. That is, the intervals (s_i, f_i, w_i) are to be scheduled on two machines without intersection. Here $s_i(resp. f_i, resp. w_i)$ is the start time (resp. finish time, resp. weight or value) of the i^{th} input job. (We usually allow intersection when $f_i = s_j$ for some $i < j$.) Let the intervals $\{I_1, \ldots, I_n\}$ be sorted so that $f_1 \leq f_2 \ldots \leq f_n$.

   Give an optimal DP for WISP on two machines.
   Hint: Use the idea for the one machine WISP discussed in class.

6. Consider the following $\{0, 1, 3\}$ knapsack problem with input items $\{(v_1, s_1), \ldots, (v_n, s_n)\}$ and capacity C. (Assume all parameters are integral). Each of the items (v_i, s_i) is either not used or used once or three times in the knapscak and as in the classical $\{0, 1\}$ knapsack, the capacity cannot be exceeded. To be precise: maximize $\sum_{1 \leq i \leq n} n_i v_i$ subject to $\sum_{1 \leq i \leq n} n_i s_i \leq C$ and $n_i \in \{0, 1, 3\}$ for all i.

   Give an optimal pseudo polynomial time DP algorithm for this problem. State the time complexity of your algorithm. (Note: the time complexity should be a polynomial in terms of n and C.)

7. Consider the exact max-2-sat problem. Define the extended Hamming neighbourhood $N'(\tau) = N_1(\tau) \cup \{\bar{\tau}\}$. That is, it is the Hamming distance 1 augmented with the component–wise complement of τ. Complete the proof that shows that the oblivious local search algorithm (with neighbiurhood N') achieves a $\frac{3}{4}$ totality ratio. That is, at every local optimum, the solution has weight at least $\frac{3}{4}$'s of the sum of the weights of all clauses.