

15-451 Algorithms, Fall 2017

Recitation 11 Worksheet

I. Safe Moving An office building has one (1) safe where valuables are kept. There are two (2) rooms in the building numbered $\sqrt{7}$ and 43. The distance between the rooms is 1 mile. There is a sequence of requests where an employee in some room needs to access the safe. If the employee is in the room with the safe, it's free. If the employee is in the other room, it costs \$1. Management is monitoring these activities, and has the option to move the safe from time to time to a different room. The cost of moving the safe is \$ p .

The requests are adversarially generated, and future requests are unknown by management. After each request management has the option of moving the safe from one place to another. Management's goal is to obtain a deterministic algorithm with low total cost. (The total cost includes the employee movement costs plus the costs incurred by moving the safe.) The criterion of any management algorithm is the competitiveness of the algorithm, as defined in class.

We will analyze a counter-based algorithm, where the threshold is $2p$. This means that when a request is from the room without the safe, the algorithm processes it in the current room (at a cost of 1), and increments the counter. If the counter has reached $2p$ it moves the safe to the just requested room (at a cost of p), and resets the counter to 0.

Prove that this algorithm is 3-competitive.

Solution: Call this online algorithm A , and call the adversary's algorithm B . Here S_X is where algorithm X currently has the safe located.

$$\Phi(S_A, c, S_B) = \begin{cases} 2c & \text{if } S_A = S_B \\ 3p - c & \text{if } S_A \neq S_B \end{cases}$$

It's just a matter of going through all the cases and proving that in each case the amortized cost to A of the operation is at most three times the cost to B of the operation. In the analysis below we let C_A be the cost of the operation to A . C_B is analogous. We let AC_A be the amortized cost to A .

- Case: The request is free to A : The cost to A is 0, the change in potential is 0, and the cost to B is ≥ 0 .
- Case: The request costs A 1:
If A and B are in the same state then $C_A = C_B = 1$. Also, $\Delta\Phi = 2$. So $AC_A = 3$. Thus $AC_A \leq 3C_B$.
If A and B are in different states, then $C_A = 1, C_B = 0$. Also, $\Delta\Phi = -1$. So $AC_A = 0$. and we have $AC_A \leq 3C_B$.

- Case: A moves the safe:

In this case c changes from $2p$ down to 0 .

If $S_A = S_B$ before and $S_A \neq S_B$ after, then the potential changes from $4p$ down to $3p$, for a net change of $-p$. $C_A = p$, so $AC_A = 0 \leq 3C_B$.

If $S_A \neq S_B$ before and $S_A = S_B$ after, then the potential changes from p before to 0 after, for a net change of $-p$. $C_A = p$, so $AC_A = 0 \leq 3C_B$.

- Case: B moves the safe:

$$\Delta\Phi \leq |(3p - c) - 2c| = |3p - 3c|$$

Since $0 \leq c \leq 2p$, we know that $|3p - 3c| \leq 3p$. So $AC_A \leq 3p \leq 3C_B$.

This completes the proof.

II. 2-SAD As explored in a previous recitation, 3-SAT's slightly less capable relative, 2-SAT, is solvable in linear time. Now consider the following modification: given a CNF with at most 2 literals per clause and a constant k , is there a satisfying assignment which satisfies at least k clauses?

Turns out, this problem is NP-complete. Prove it.

Solution: NP is trivial, just nondeterministically guess the assignment.

NP Hard: Reduce 3-SAT: Let n be the number of clauses. For each 3-SAT clause (p, q, r) , translate to a group of clauses $\{p, q, r, s, (\bar{p} \vee \bar{q}), (\bar{q} \vee \bar{r}), (\bar{r} \vee \bar{p}), (\bar{s} \vee p), (\bar{s} \vee q), (\bar{s} \vee r)\}$ where s is a new variable we create for each clause. Let $k = 7n$.

Easy to check that any satisfying 3-SAT assignment has some setting of s where 7 of these clauses are satisfied (do some casework on how many of each clause is satisfied). Moreover, note no more than 7 can be satisfied.

Check that for any unsatisfied clause, strictly less than 7 can be satisfied.

Thus any satisfying assignment produces exactly $7n$ satisfied terms and any non-satisfying assignment gives $< 7n$.

Notes: think of s as a 'slack' variable which allows satisfiability to $= 7$ regardless of how many satisfied literals there are (as long as there is at least 1). You can notice that if we fix $s = T$, then if (p, q, r) are all satisfied there are only 6 satisfied, while if $s = F$ the same is true for if exactly 1 of the literals is true in the satisfying assignment.

NP Hard: Reduce Independent Set: For each vertex, have term (v) , and for each edge (u, v) , have $|V| + 1$ copies of $(\bar{u} \vee \bar{v})$. Now set $k = |E|(|V| + 1) + i$ where i is the size of the independent set you want. We first see that any independent set of size i being true (and all other variables being false) will satisfy k clauses, as it will satisfy all edge clauses and i vertex clauses. Now if the true variables are not an independent set, then they can satisfy at most $(|E| - 1)(|V| + 1) + |V| = |E|(|V| + 1) - 1 < k$ clauses, as since it is not independent, some edge clause was not satisfied.