Exercise 3.3

zondag 10 september 2023 16:12

1. The approximation ratio is $n-2$.
2. Informally: the rough idea of the 3-approximation algorithm is to not set just one of the variables in the selected clause to True, but to set all three of them to True. This prevents the problem where the 'wrong' variable is selected continuously, in a similar way to what was done in the weighted vertex-cover problem.

(i) To determine (a bound on) the approximation ratio of Greedy-CNF, we first consider that the algorithm may return a solution of size $n-2$, and in particular, will never produce a solution of a larger size. To see why, we consider the following invariant over the loop in the algorithm:

at the start of an iteration of the loop, all clauses in C have the property that all variables contained in them are not true.

Proof of invariant:

initialization: prior to the first execution of the loop, none of the variables have been set to true. Hence, the invariant holds.
(This assumes all variables start out as false, or as unset.)

maintenance: during an iteration, exactly one variable is set to true. Then, all clauses containing that variable are removed from C. This implies that all remaining clauses do not contain any variable which was set to true in this loop iteration. Thus, as long as these clauses did not contain any variable set to true before this iteration of the loop, they will not contain any variable set to true after this iteration either. Since the loop invariant guarantees us that this condition is met before the iteration, we can now conclude it must hold after the iteration as well.

termination: at the end of the loop, C is empty (by construction). Hence, the invariant holds trivially.

This invariant is particularly useful with respect to the last iteration of the loop; prior to the start of that iteration, all variables in the clause that will be selected are not true. In the final iteration, exactly one of them will be set to true. Hence, at least two variables will remain false after the final iteration of the loop. This implies at most $n-2$ variables can be set to true by the algorithm, and hence the size of the solution returned is at most $n-2$.

Furthermore, it is straightforward to see that a lower bound on the size of the optimal solution is 1; any valid solution must have that at least one variable is set to true (with the possible but nonsensical exception of $n=0$, i.e. the problem with 0 clauses; we will ignore this case).

This gives us that

$$\begin{aligned} \text{Greedy-CNF}(C,X) &\leq n-2 \\ &\leq (n-2)\cdot 1 \qquad \text{note: } LB=1 \\ &\leq (n-2)\cdot LB \\ &\leq (n-2)\cdot OPT, \end{aligned}$$

which gives us an approximation ratio of $n-2$, which is tight; to see why, consider any number of clauses of the form (x_j, x_{n-1}, x_n).
j = clause number

(ii)

We can construct a 3-approximation algorithm as follows:

3-Greedy-CNF(C,X)

1. $C = \{C_1, \dots, C_m\}$ is a set of clauses, $X = \{x_1, \dots, x_n\}$ a set of variables.
2. while $C \neq \emptyset$ do
   a. Take an arbitrary clause $C_j \in C$.
   b. Let x_{j1}, x_{j2} and x_{j3} be the variables in C_j.
   c. Set x_{j1}, x_{j2} and x_{j3} to true.
   d. Remove all clauses from C that contain at least one of x_{j1}, x_{j2} and x_{j3}.
3. end while
4. return X

Let z be the number of iterations taken by the loop.
In this case, we can see that a lower bound on the size of OPT is given by z; at least one variable from each clause selected must be set to true, since otherwise, there would be at least one clause without any variables set to true. ~~note that, since all variables in a selected clause are set to true at the same time, it must be the case that the sets of variables in the clauses are disjoint. This ensures the number of selected clauses~~

Furthermore, the algorithm sets $3z$ variables to true (by definition of z and line 2c of the algorithm. This gives

$$\begin{aligned} \text{3-Greedy-CNF}(C,X) &= 3z \qquad \text{using } LB = z \\ &\leq 3\cdot LB \\ &\leq 3\cdot OPT \end{aligned}$$

This proves that 3-Greedy-CNF is a 3-approximation algorithm.

(iii)

LPR-CNF(C,X)

1. Solve the relaxed linear program corresponding to the given problem:

$$\begin{array}{lll} \text{Minimize} & \sum_{i=1}^{n} \lambda_i & \\ \text{Subject to} & \sum_{i\in\{1\leq i\leq n \mid x_i \Rightarrow c_j\}} \lambda_i \geq 1 & \text{for all clauses } c_j \in C \\ & 0 \leq \lambda_i \leq 1 & \text{for } 1 \leq i \leq n \end{array}$$

2. solution $\leftarrow \{x_i \in X \mid \lambda_i \geq \frac{1}{3}\}$
3. return solution

$$\begin{aligned} ALG(I) &= \text{number of variables set to true} \\ &= |X^*| \\ &= \textstyle\sum_{x_i \in X^*} 1 \\ &\leq \textstyle\sum_{x_i\in X^*} 3y_i \qquad \text{since } y_i \geq \tfrac{1}{3} \text{ for all } x_i \in X^* \\ &\leq \textstyle\sum_{i=1}^{n} 3y_i \qquad \text{since } y_i \geq 0 \text{ for all } 1 \leq i \leq n \\ &= 3\textstyle\sum_{i=1}^{n} y_i \\ &= 3\,LB(I) \qquad \text{since } LB = \textstyle\sum_{i=1}^{n} y_i = \text{solution to relaxed LP} \\ &\leq 3\,OPT(I) \end{aligned}$$