COMP 360: Algorithm Design Instructor: Yang Cai

Assignment 4 Solution

General Rules: In solving these questions you may consult books but you may not search on the web for solutions. You must write up your final solution yourself. You should drop your solutions in the assignment drop-off box located in the Trottier Building. No late assignments accepted.

1. (15 pts) For any $A \subseteq \mathbb{R}$, let $f(A) = \sum\limits_{a \in A} a$. Given $B = \{b_1, ..., b_n\}$ with $b_i \in (0, 1]$, let $\mathcal{C}(B) = \{C : C = \bigcup\limits_{i=1}^{m} C_i$ is a partition of B such that $f(C_i) \in (0, 1],\ \forall i \in [m]\}$.
   Consider the following problem: Given some B as above, find $m^* = \arg\min\limits_{C \in \mathcal{C}(B)} |C|$.
   Give a $\frac{3}{2}$-approximation algorithm for this problem that runs in polynomial time.

   Solution

   (a) Algorithm: For i from 1 to n, put b_i in a bin already containing items such that the bin's total weight will be ≤ 1 if such a bin exists. Otherwise, put b_i into a new bin.
   Let $s(B) = \sum\limits_{i=1}^{n} b_i$. Let k the number of bins used by our algorithm, k^* the minimal number of bins. We know that $k^* \geq \lceil s(B) \rceil$. On the other hand, $k \leq 2 * \lceil s(B) \rceil$ since after running the algorithm, all but at most one bin contains weight $\geq 1/2$.

   (b) Algorithm: same as above but assume $b_1 > b_2, ..., > b_n$ (sort before starting). Let $j = \lceil 2k/3 \rceil$. If bin j contains an item of weight $> 1/2$, then so do bins $1, ..., j-1$ which implies $k^* \geq j$. Otherwise all items in bin j are of weight $\leq 1/2$. Then so are all items in all bins $j+1, .., k$. Thus there are at least $2(k-j)+1$ items in bins $j, ..., k$. These items did not fit in bins $1, ..., j-1$. This means $s(B) > min\{j-1, 2(k-j)+1\} \geq j-1$. Thus $k^* \geq \lceil s(B) \rceil \geq j$, hence we are done.

2. (15 pts)*Maximum Weight Clique:* Given an undirected graph $G = (V, E)$. Each vertex $i \in V$ has a weight w_i. The problem asks you to find a clique $S \subseteq V$ such that the total weight is maximized.
   (a) (7 pts) Formulate this problem as an Integer Linear Program (ILP).
   (b) (8 pts) What is the Linear Program relaxation (LP) of this problem and what is the **Integrality Gap** of your LP relaxation?

   Solution:

   (a) Let $x_i \in \{0, 1\}$ represent whether vertex i is selected (1 if it's selected, 0 if not). Then the ILP is as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1, \quad \forall i \neq j, (i,j) \notin E \\ & x_i \in \{0,1\}, \quad \forall i \in V \end{aligned}$$

   For any feasible vector x and i, j such that $x_i = x_j = 1$, then i must connect to j in G since if $(i, j) \notin E$, $x_i = x_j = 1$ contradicts with the first constraint of ILP. Thus the set S corresponded to vector x is a clique.

On the other side, for any clique $S \subseteq V$, let x be the corresponding vector of S. Then for any $(i, j) \notin E$, $x_i + x_j$ must be at most 1. Since $x_i + x_j > 1$ implies $x_i = x_j = 1$, i.e., $i, j \in S$. Since S is a clique, $(i, j) \in E$. Contradiction. Thus x is feasible in the ILP.

(b) The LP relaxation is as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1, \quad \forall i \neq j, (i, j) \notin E \\ & x_i \geq 0, \quad \forall i \in V \end{aligned}$$

Now we consider the graph with n nodes and no edge (all nodes are isolated). Each node i has weight $w_i = 1$. Then the maximum weight clique for this graph contains a single node, with total weight 1. For the LP, the vector $x = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})^T$ is feasible with objective value $\frac{n}{2}$. We will prove this is the optimal solution for the LP. For any feasible solution x, since $E = \emptyset$, $x_i + x_j \leq 1$ for all $i \neq j$. Then the objective

$$\sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} x_i = \frac{1}{2} \sum_{i=1}^{n} (x_i + x_{j+1}) \leq \frac{n}{2} \tag{1}$$

where $x_{n+1} = x_1$. Thus for this graph $\frac{\text{OPT(LP)}}{\text{OPT(ILP)}} = \frac{n}{2}$.

The Integrality Gap for this relaxation is at least $\frac{n}{2}$ where n is the number of nodes in the graph. (If you come up with an example with any non-constant Integrality Gap, you'll get full marks).

3. (15 pts) Let $X = \{x_1, \ldots, x_n\}$ be set of n boolean variables. Consider a 3-CNF formula over the set X which does not contain negated variables. An example of such a formula is

$$(x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_3 \vee x_5) \wedge (x_1 \vee x_2 \vee x_3)$$

Our goal is to find the smallest set of variables such that if you set them TRUE, the CNF will evaluate to TRUE. Give a 3-approximation algorithm for this problem based on LP rounding.

Solution

Let the CNF be $\Phi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$. Formulate the following integer linear program for the problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i + x_j + x_k \geq 1, \quad \forall x_i, x_j, x_k \in X \quad \text{s.t.} \quad x_i \vee x_j \vee x_k = C_l \\ & x_i \in \{0, 1\} \quad \forall x_i \in X \end{aligned}$$

If $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is optimal solution to this ILP, then it is also the assignment which makes Φ to evaluate to 1 and has the smallest number 1's in it. Now let's relax the ILP to a linear program:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i + x_j + x_k \geq 1, \quad \forall x_i, x_j, x_k \in X \quad \text{s.t.} \quad x_i \vee x_j \vee x_k = C_l \\ & x_i \geq 0 \quad \forall x_i \in X \end{aligned}$$

Let $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)$ be the optimal solution for the LP and the optimal value of LP $\leq$ optimal value of ILP, i.e. $\sum_{i=1}^{n} \hat{x}_i \leq \sum_{i=1}^{n} x^*$. Then let $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ be the assignment after rounding $\hat{x}$'s fractional values, i.e. $\forall i \ \tilde{x}_i = 1$ if $\hat{x}_i \geq \frac{1}{3}$ and $\tilde{x}_i = 0$ otherwise.

First of all let's make sure that $\tilde{x}$ makes Φ to evaluate to 1. Since $\hat{x}_i + \hat{x}_j + \hat{x}_k \geq 1$ for every i, j, k such that x_i, x_j, x_k forming a clause, then at least one of $\hat{x}_i$ or $\hat{x}_j$ and $\hat{x}_k$ has to be at least $\frac{1}{3}$ which means respectively one of $\tilde{x}_i, \tilde{x}_j, \tilde{x}_k$ will be rounded to 1, so for every clause there will be at least one variable set to 1.

To show that this gives a 3-approximation we need to show that $\sum_{i=1}^{n} \tilde{x}_i \leq 3 \sum_{i=1}^{n} x_i^*$.

$$\sum_{i=1}^{n} \tilde{x}_i \leq \sum_{i=1}^{n} 3\hat{x}_i \leq 3 \sum_{i=1}^{n} x_i^*.$$

4. (15 pts) You have a knapsack of capacity W, and there are n items where item i has value v_i and weight w_i. Let V be the optimal value you can achieve. Design a PTAS to find a subset of items whose total value is at least V and its total weight at most $(1 + \epsilon) \cdot W$.

   Solution
   Let $\bar{w}_i = \lfloor \frac{w_i}{\theta} \rfloor \theta$ and $\hat{w}_i = \lfloor \frac{w_i}{\theta} \rfloor$ where w_i weight of item i.
   Consider the following recursive definition of $m[i, w]$, the maximum value that can be attained with weight less than or equal to w, using items up to i.
   $m[0, w] = 0$
   $m[i, w] = m[i-1, w]$ if $w_i > w$
   $m[i, w] = \max(m[i-1, w], m[i-1, w - w_i] + v_i)$ if $w_i \leq w$
   This gives a dynamic program with runtime $O(nW)$, where n the number of items. Run the dynamic program with values v, weights $\hat{w}_i$, with $\theta = \lfloor \frac{W\epsilon}{n} \rfloor$. Then we get at least the optimal total value and the runtime is $n\lfloor \frac{W}{\theta} \rfloor = n\lfloor \frac{n}{\epsilon} \rfloor$. Let S be the chosen set of items. We know $\sum_{i \in S} \hat{w}_i \leq \frac{W}{\theta}$. Thus the total weight of the items can be given upper bound:
   $\sum_{i \in S} w_i \leq \sum_{i \in S} (\bar{w}_i + \theta) = n\theta + \sum_{i \in S} \bar{w}_i = n\theta + \theta \sum_{i \in S} \hat{w}_i \leq W + n\theta = (1 + \epsilon)W.$