Assignment-3

Praneeth Kacham
2015CS10600

Question-1

Integer-Program

$$\begin{aligned} \min_{x} \quad & \sum_{e} c_e x_e \\ \text{s.t.} \quad & \sum_{e \in P_i} x_e \geq 1 \\ & x_e \in \{0, 1\} \end{aligned}$$

Dual of Relaxed problem

$$\begin{aligned} \max_{y} \quad & \sum_{i=1}^{k} y_i \\ \text{s.t.} \quad & \sum_{i: e \in P_i} y_i \leq c_e \qquad \forall e \in E \\ & y_i \geq 0 \end{aligned}$$

Algorithm 1 Primal-Dual Algorithm for Multi-Cut problem

```
procedure MULTICUT(T = (V, E), r ∈ V, c_e ≥ 0 ∀e, (s_i, t_i) i = 1...n)
    F ← ∅
    while F is not a multi-cut do                                          ▹
        i be the index of the unseparated pair (s_i, t_i) having highest depth(lct(s_i, t_i))
        Increase y_i such till edge e becomes tight
        F ← F ∪ {e}
    end while
end procedure
```

In reverse delete step, go through the edges in the reverse order in which they are addded to F. Delete an edge e if $F - \{e\}$ is a feasible multi-cut. Return F finally.

Theorem 1.

$$cost(F) = \sum_{e \in F} c_e \leq 2 * OPT \tag{1}$$

Proof. Let y be the dual feasible solution given by the algorithm. Hence, $\sum_{i=1}^{k} y_i \leq OPT$. We also have

$$\begin{aligned}
cost(F) &= \sum_{e \in F} c_e \\
&= \sum_{e \in F} \sum_{i: e \in P_i} y_i && \text{(Since, an edge is added only when tight)} \\
&= \sum_{i=1}^{k} y_i |F \cap P_i|
\end{aligned}$$

Claim 1. $y_i > 0 \Rightarrow |F \cap P_i| \leq 2$

Proof. Let $a \rightsquigarrow b$ denote the set of edges in the path from a to b. Suppose there is an i such that $y_i > 0$ and $|F \cap P_i| > 2$. Let u be the lowest common ancestor of s_i and t_i. Let e be the edge which became tight by increasing y_i.(Note: e might have been deleted from F in deletion step). As $|F \cap P_i| > 2$, we can, without loss of generality assume that $|F \cap (s_i \rightsquigarrow u)| \geq 2$. Let e_1, e_2 be two edges in $|F \cap (s_i \rightsquigarrow u)|$ such that e_1 is closer to r that e_2. Let pair (s_l, t_l) caused the addition of e_1 and $(s_{l'}, t_{l'})$ caused the addition of e_2. We claim that our reverse deletion step would have deleted the edge e_2 and hence obtain a contradiction. Given that $y_i > 0$, we can conclude that $depth(lct(s_i, t_i)) > depth(lct(s_l, t_l))$ and $depth(lct(s_i, t_i)) > depth(lct(s_{l'}, t_{l'}))$. Otherwise, edges e_1 or e_2 would have been added to F earlier that e and y_i couldn't have been raised. We can also conclude that all the pairs for which e_2 is the earliest separator that has been added to F have their lct depth lower that $depth(lct(s_i, t_i))$.Again, otherwise, e_2 would have been added before e and hence y_i couldn't have been raised. This implies that all those pairs for which e_2 is the earliest separator that has been added to F have e_1 in the path between the nodes. It is also easy to see that e_1 has been added to F after e_2. Thus, in reverse delete step we observe e_2 after e_1 and hence we will remove e_2 from F as e_1 separates all the pairs which require e_2. Hence having e_1 and e_2 both in F is a contradiction. Which gives $y_i > 0 \Rightarrow |F \cap P_i| \leq 2$. □

From the claim above, we have $y_i|F \cap P_i| \leq 2y_i$. Hence, the $cost(F) = \sum_{i=1}^{k} y_i |F \cap P_i| \leq \sum_{i=1}^{k} 2y_i \leq 2OPT$. □

Question-2

Let F' be the set of all edges initially added by the algorithm. Let F be the set of edges returned by the algorithm which deletes edges in any order i.e., F is a feasible solution for the problem and $\forall e \in F$, $F - \{e\}$ is not feasible. Let C_i be the connected components in the iteration when ith edge is added by the primal-dual algorithm. By, $C_i^G \subseteq C_i$ denotes the connected components whose dual variable is increased in the ith iteration and define $C_i^N = C_i - C_i^G$. Thus, $C_0 = \{\{s_i\}, \{t_i\} | (s_i, t_i) \in \mathcal{P}\}$, $C_0^G = C_0$ and $C_0^N = \emptyset$. Define graph G_i as follows: vertex set is given by $V_i = \{v_j | j = 1, \ldots, |C_i|\}$ and $E_i = \{(v_k, v_l) | \exists (u, v) \in F, u, v$ in different connected components in $C_i\}$.

Claim 2. *G_i is a forest for all i.*

Proof. It is easy to see that the vertex set of G_i along with the edges F' is a forest. As $F \subseteq F'$, we have that G_i is a forest. □

Claim 3. *All vertices corresponding to the components C_i^N in the graph G_i have degree ≥ 2.*

Proof. Suppose there is a component in C_i^N with degree 1 in the graph G_i. Given that the component is in C_i^N, we have that the connected coponent doesn't separate any pair (s_i, t_i). So, the only edge that is coming into the connected component doesn't connect any pair (s_i, t_i). Hence, this edge can be deleted from F without affecting feasibility. This contradicts the fact that $F - \{e\}$ is unfeasible for all the edges $e \in \{F\}$. Hence, all vertices corresponding to the components C_i^N have a degree ≥ 2. □

Theorem 2. *The algorithm is a 2-approximation to steiner-forest problem.*

Proof. Using the above claims, the proof that this gives 2-approximation goes exactly like the proof of 2-approximateness of the reverse-deletion algorithm. □

Question-3

Notation : For $S \subseteq V$, $\delta^+(S) = \{(i,j) \in A | i \notin S, j \in S\}$. Let $\mathcal{S}$ denote the set $\{S \subseteq V | r \notin S\}$.
Integer Program

$$\begin{aligned} \min_x \quad & \sum_{e \in A} c_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta^+(S)} x_e \geq 1 \qquad \forall S \in \mathcal{S} \\ & x_e \in \{0,1\} \end{aligned}$$

Dual of Relaxation

$$\begin{aligned} \max_y \quad & \sum_{S \in \mathcal{S}} y_S \\ \text{s.t.} \quad & \sum_{S : e \in \delta^+(S)} y_S \leq c_e \qquad \forall e \in A \\ & y_S \geq 0 \end{aligned}$$

On the set F returned by the algorithm, do a reverse delete step which goes through the edges in the reverse

Algorithm 2 Primal-Dual Algorithm for MinCost Branching Problem

procedure MIN-BRANCHING($G = (V, A), r \in V, c_e \geq 0\ \forall e$)
  $F \leftarrow \emptyset$
  $\mathcal{C} \leftarrow \{\{v\} | v \in V - \{r\}\}$
  while F is not feasible **do** ▷
    Raise the y_S value for all $S \in C$ till an edge $e = (u, v)$ becomes tight
    Let $u \in C_1 \in \mathcal{C}$ and $v \in C_2 \in \mathcal{C}$
    $\mathcal{C} \leftarrow (\mathcal{C} - \{C_2\}) \cup \{C_1 \cup C_2\}$
    $F \leftarrow F \cup \{e\}$
  end while
end procedure

order they are added to F and delete e from F if $F - \{e\}$ is feasible. Let F' be the set finally obtained after the reverse delete step. Let $\mathcal{C}^i$ be the $\mathcal{C}$ in ith iteration.

Claim 4. $\sum_{S \in \mathcal{C}^i} |\delta^+(S) \cap F'| \leq |\mathcal{C}^i|$

Proof. Suppose the claim doesn't hold. Then $\mathcal{D} = \{C \in \mathcal{C}^i || \delta^+(C) \cap F'| \geq 1\}$ is non-empty. Let C be a minimal element of $\mathcal{D}$ i.e., $\forall S \subset C : S \notin \mathcal{D}$. It is easy to see that C cannot be set of single vertex.(We would have deleted all but one of the arcs in to the vertex in the reverse delete stage.) From algorithm, we can see that C is made up of two components C_1 and C_2 and one of them is still growing. Let C_1 be active and C_2 be passive. We also have that there is an edge $e \in F$ going from C_1 to C_2. From, minimality of C, we have $C_1 \notin \mathcal{D}$. We also have that, C_2 with the arcs in F in ith iteration has a vertex from which all the vertices can be visited and this vertex is the end-point of the arc e. From this we can conclude that $|\delta^+(C) \cap \delta^+(C_1) \cap F'| \geq 1$. Let e' be an arc in $\{\delta^+(C) \cap \delta^+(C_1) \cap F'\}$. Let e' be added in an iteration j in which a component $C' \supset C_2$ was growing. But, C_2 was already passive which implies that there was another component $C'' \subset C'$ which was also growing. Let e'' be the arc which stops the component C'' from growing. Now in the reverse delete step, we see e'' before seeing e'. We claim that e' would have been deleted in the reverse delete step. r could reach all the vertices of C_2 using the edge e''. Now e' is redundant as all the vertices of C_2 could be reached using the vertex which can reach all the vertices of C_2. Hence, e' is deleted which contradicts our assumption. □

Proof. of optimality:
We have $\sum_{e \in F'} c_e = \sum_{e \in F'} \sum_{S \in \mathcal{S} : e \in \delta^+(S)} y_S = \sum_{S \in \mathcal{S}} y_S |\delta^+(S) \cap F'|$. We claim that $\sum_{S \in \mathcal{S}} y_S |\delta^+(S) \cap F'| \leq \sum_S y_S$.

We prove this by induction on iterations of the algorithm. Initially it is trivially true as both LHS and RHS are true. Assume that the inequality is true before ith iteration. In ith iteration LHS increases by $\epsilon \sum_{S \in C^i} |\delta(S) \cap F'|$ and RHS increases by $\epsilon |C^i|$. From the claim above, we have that increase in LHS is less than increase in RHS. Hence, LHS $\leq$ RHS, after the end of ith iteration. Hence, the algorithm is optimal. $\square$

Question-4

Define $A \cdot X = \sum_{i,j} a_{ij} x_{ij}$.
Primal SDP

$$\begin{aligned} \max_X \quad & \sum_{i<j} w_{ij}(1 - x_{ij})/2 \\ \text{s.t.} \quad & x_{ii} = 1 \quad \forall i \\ & X \succeq 0 \end{aligned}$$

Dual

$$\begin{aligned} \min_\gamma \quad & \frac{1}{2}\sum_{i<j} w_{ij} + \frac{1}{4}\sum_i \gamma_i \\ \text{s.t.} \quad & W + diag(\gamma) \succeq 0 \end{aligned}$$

We have W is a symmetric matrix with $w_{ii} = 0$. To show weak duality, we need to show that given $X \succeq 0$, $x_{ii} = 1 \ \forall i$, $W + diag(\gamma) \succeq 0$, we have

$$\frac{1}{2}\sum_{i<j} w_{ij}(1 - x_{ij}) \leq \frac{1}{2}\sum_{i<j} w_{ij} + \frac{1}{4}\sum_i \gamma_i$$

Lemma 3. *If X, Y are positive semidefinite matrices, then $X \cdot Y \geq 0$*

Proof. Given matrices X, Y we have $X \cdot Y = \text{tr}(X^T Y)$. As X, Y are p.s.ds, we can write $X = LL^T$ and $Y = MM^T$. Hence, $\text{tr}(X^T Y) = \text{tr}(LL^T MM^T) = \text{tr}(L^T MM^T L) = \text{tr}(L^T M (L^T M)^T) = \|L^T M\|_F^2 \geq 0$. Second equality follows from the fact that $\text{tr}(AB) = \text{tr}(BA)$. $\square$

Proof.

$$\begin{aligned} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_{ij}) \leq \frac{1}{2}\sum_{i<j} w_{ij} + \frac{1}{4}\sum_i \gamma_i \\ \Leftrightarrow \quad & -\frac{1}{2}\sum_{i<j} w_{ij}x_{ij} \leq \frac{1}{4}\sum_i \gamma_i \\ \Leftrightarrow \quad & -\frac{1}{4}\sum_{i \neq j} w_{ij}x_{ij} \leq \frac{1}{4}\sum_i \gamma_i && \text{(Since, } x_{ij} = x_{ji} \text{ \& } w_{ij} = w_{ji}) \\ \Leftrightarrow \quad & -\frac{1}{4}\sum_{i,j} w_{ij}x_{ij} \leq \frac{1}{4}\sum_i \gamma_i && \text{(Since, } w_{ii} = 0) \\ \Leftrightarrow \quad & 0 \leq \sum_{i,j} w_{ij}x_{ij} + \sum_i \gamma_i \\ \Leftrightarrow \quad & 0 \leq \sum_{i,j} w_{ij}x_{ij} + \sum_i \gamma_i x_{ii} && \text{(Since, } x_{ii} = 1) \\ \Leftrightarrow \quad & 0 \leq (W + diag(\gamma)) \cdot X \end{aligned}$$

Given that the last inequality is true as both matrices are p.s.ds, we can follow the bi-implications backward and obtain what is required. $\square$

Question-5

Part-a

For each vairable x_i in the satisfiability problem we will have a variable y_i which takes the values from the set $\{-1,1\}$. We also have a variable $y_0 \in \{-1,1\}$. x_i is assigned the truth value TRUE if $y_i \dot{y}_0 = 1$ and FALSE otherwise. Let the first variable in ith clause be x_{i_1} and second variable be x_{i_2}. For a clause $x_1 \vee x_2$, we have the integer expression $\frac{1}{4}(3+y_0y_1+y_0y_2-y_1y_2)$ and similar expressions for other forms of 2-SAT clauses. Let there be n clauses. The following integer program models a Max-2SAT problem.

$$\begin{aligned} \max_{y} \quad & \sum_{i=1}^{n} \frac{1}{4} w_i (3 \pm y_0 y_{i_1} \pm y_0 y_{i_2} \pm y_{i_1} y_{i_2}) \\ \text{s.t.} \quad & y_j \in \{-1,1\}\ \forall j \end{aligned}$$

Sign of $\pm$ depends on the corresponding clause.

Part-b

The integer program in part-a can be relaxed to get a Semi-definite program by replacing y_j with vector $\vec{y_j}$ and replacing the product $y_j y_k$ with $\langle \vec{y_j}, \vec{y_k} \rangle$ and adding a condition that $\|y_j\|^2 = 1$.
Rounding the semi-definite solution:
Let $\vec{y_0}^*, \vec{y_1}^*, \ldots, \vec{y_m}^*$ be the optimal semi-definite solution. Pick a random plane given by $a^T x = 0$. Variable y_i is assigned 1 if $a^T \vec{y_i}^* > 0$ and -1 otherwise.

Claim 5. *$\forall i, j$, the following hold true*

$$\begin{aligned} E[1 + y_i y_j] &\geq 0.878[1 + \langle y_i, y_j \rangle] \\ E[1 - y_i y_j] &\geq 0.878[1 - \langle y_i, y_j \rangle] \end{aligned}$$

Proof. We have the following,

$$y_i y_j = \begin{cases} +1 \text{ with probability } 1 - \cos^{-1}(\langle \vec{y_i}^*, \vec{y_j}^* \rangle)/\pi \\ -1 \text{ with probability } \cos^{-1}(\langle \vec{y_i}^*, \vec{y_j}^* \rangle)/\pi \end{cases} \tag{2}$$

Hence, $E[1 + y_i y_j] = 2[1 - \cos^{-1}(\langle \vec{y_i}^*, \vec{y_j}^* \rangle)/\pi] = (2/\pi)\cos^{-1}(-\langle \vec{y_i}^*, \vec{y_j}^* \rangle) \geq 0.878(1 + \langle \vec{y_i}^*, \vec{y_j}^* \rangle)$. Similarly, we can show that the other inequality is also true. □

Theorem 4. *Rounding as defined is 0.878 approximate solution for MAX-2-SAT.*

Proof. Consider a clause $(x_1 \vee x_2)$. The corresponding expression in terms of $y_i's$ is given by $(3 + y_1 y_0 + y_2 y_0 - y_1 y_2)/4$. Then expression has value 1 if the clause is satisfied and 0 otherwise. The expression can also be written as $((1 + y_1 y_0) + (1 + y_2 y_0) + (1 - y_1 y_2))/4$. But, from the previous claim, $E[((1 + y_1 y_0) + (1 + y_2 y_0) + (1 - y_1 y_2))/4] \geq 0.878((1 + \langle \vec{y_1}^*, \vec{y_0}^* \rangle) + (1 + \langle \vec{y_2}^*, \vec{y_0}^* \rangle) + (1 - \langle \vec{y_1}^*, \vec{y_2}^* \rangle))/4$. Hence, the expected value for integer rounding is $\geq 0.878 OPT\_SDP \geq 0.878 OPT$. □

Part-c

We can combine, the 0.75 approx algorithm and MAX-2-SAT algorithm to obtain a better approximation ratio. We have that, for clauses of size greater than 3, 0.789 fraction of them are true in 0.75-approx algorithm and 0.75 fraction for clauses of size < 3. If, with α probability 0.75 algorithm is run and $(1-\alpha)$ probability MAX-2-SDP is run, we obtain a $\min\{\alpha 0.789, \alpha 0.75 + (1-\alpha)0.878\}$ algorithm. This is maximized when $\alpha 0.039 = (1-\alpha)0.878 \Rightarrow \alpha = 22.5/23.5 = 0.957$ and the best approximation is ~ 0.755.