HW4:

李钊佚 SA22011035

Exercise 1:

Solution 1:

The Algorithm "$4/3$-approx-TSP" can be described as follows:

Algorithm 1: $4/3$ - approx-TSP.

* Input: a graph G whose edge weights are either 1 or 2, start point $v_0 \in V(G)$, a known "min-2-matching" algorithm MIN-2-MATCHING for polynomial time.
* Output: a tour path S of TSP problem for G and v_0

o 1: utilize known MIN-2-MATCHING on G to get a optimal solution M for min-2-matching problem.

o 2: according to the defination of "2-matching", we know that $M = \{E_1 \cup E_2 \cup \cdots E_k\}$, where $E_i \cap E_j = \emptyset$ and E_i and E_j share no common vertex for $\forall i \neq j$, while for $\forall i \in \{1, 2, \cdots, k\}$, $\exists$ a connected graph G_i, s.t. $E_i = E(G_i)$.

o 3: $S = E_1$

04: for $2 \le i \le k$ do:

05: $e_1^* = \max_{w(e)} \{e \mid e \in S\}$

06: $e_2^* = \max_{w(e)} \{e \mid e \in E_k\}$

07: $V_1, V_2 = \text{vertex}(e_1^*)$

08: $V_3, V_4 = \text{vertex}(e_2^*)$

09: $e_3^* = \text{edge}(V_1, V_3)$

10: $e_4^* = \text{edge}(V_2, V_4)$.

11: $S = S - \{e_1^*, e_2^*\} + \{e_3^*, e_4^*\}$.

12: return S

Prove the $4/3$-approximatio ratio for the Algorithm:

Prove:

define the notion $\Delta(i)$ means in i-th iteration

$$\Delta(i) = w(S - \{e_1^*, e_2^*\} + \{e_3^*, e_4^*\}) - w(S)$$

denote in the start of i-th iteration $(1 \le i \le k-1)$

$S \triangleq S_i$.

denote in the end of ith iteration $(1 \le i \le k-1)$

$$S \triangleq S_{i+1}.$$

So naturally we can induce that:

$$\Delta(i) = w(S_{i+1}) - w(S_i).$$

Let $T(i)$ denote the number of edges whose weight $=2$ in S_i. $(1 \le i \le k)$

So obviously we can induce that

① $\Delta(i) \le 1$, when $T(i) \ge 1$

② $T(i+1) = T(i) + \Delta(i)$

So the following equation exists!

$$T(k) = T(k-1) + \Delta(k-1) = \cdots = T(1) + \Delta(1) + \Delta(2) + \cdots + \Delta(k-1)$$
$$= T(1) + \sum_{i=1}^{k-1} \Delta(i).$$

$$\Rightarrow \sum_{i=1}^{k-1} \Delta(i) = T(k) - T(1)$$

According to the algorithm, We know that

$$\#(\text{weight}=2 \text{ edges in } S_k) - \#(\text{weight}=2 \text{ edges in } S_1) \le k.$$

$$\Rightarrow \sum_{i=1}^{k-1} \Delta(i) \le k$$

Considering that: for each $i \in \{1, 2, \cdots, k\}$.

$$|E_i| \ge 3$$

So we have that $|M| = |\bigcup_{i=1}^{k} E_i| = \sum_{i=1}^{k} |E_i| \geq 3k$.

(because $E_i \cap E_j = \phi$)

So: [$cost^*$ denotes the cost of optimal solution].

$$\frac{cost(\text{Algo: } 4/3\text{-approx-TSP})}{cost^*(\text{MIN-2-MATCHING})} \leq \frac{W(S)}{W(M)}$$

$$= \frac{W(M) + \sum_{i=1}^{k+1} \Delta(i)}{W(M)} = 1 + \frac{\sum_{i=1}^{k+1} \Delta(i)}{W(M)}$$

$$\leq 1 + \frac{k}{|M|} \leq 1 + \frac{k}{3k} = \frac{4}{3}$$

and

$cost^*(TSP) \geq cost^*(\text{MIN-2-MATCHING})$,

for each solution p for TSP,

we know that p is a solution for MIN-2-MATCHING.

To conclude:

$$\frac{cost(\text{Algo: } 4/3\text{-approx-TSP})}{cost^*(TSP)} \leq \frac{cost(\text{Algo: } 4/3\text{-approx-TSP})}{cost^*(\text{MIN-2-MATCHING})} \leq 4/3$$

Exercise 2:

Solution 2:

First, I would give a proof showing that the LP-FACILITY LOCATION algorithm achieves a 10-approximation ratio for this case.

Proof:

After running the LP-FACILITY LOCATION on this case, the service cost for client j in cluster k is at most:

$$C_{i_k,j}^2 \le (C_{ij} + C_{ijk} + C_{ikjk})^2$$

$$= \left(C_{ij}^2 + C_{ijk}^2 + C_{ikjk}^2\right) \cdot (1^2 + 1^2 + 1^2) \quad \text{... Cauchy inequality.}$$

$$\le (V_j^{*2} + V_{jk}^{*2} + V_{jk}^{*2}) \cdot (1+1+1) \quad \text{... complementary slackness}$$

$$\le 9 \cdot V_j^{*2} \quad \text{... according to the description of Algorithm.}$$

the bound of facility is the same as the proof in Lecture 10,

$$f_{ik} = f_{ik} \cdot \sum_{i:\text{neighbor of } j_k} x^*_{ij_k} \le \sum_{i:\text{neighbor of } j_k} f_i \cdot x^*_{ij_k}$$

$$\le \sum_{i:v} f_i \cdot y_i^*$$

Since $j_k, j_{k'}$ cannot have a common neighbor for $k \neq k'$, the total facility cost is,

$$\sum_k f_{ik} \le \sum_k \sum_{i:\text{neighbor of } j_k} f_i \cdot y_i^* \le \sum_{i \in F} f_i \cdot y_i^*$$

So the total cost is at most

$$9 \cdot \sum_{j \in D} v_j^{*2} + \sum_{i \in F} f_i \cdot y_i^*$$

$$\le 9 \cdot OPT_{LP} + OPT_{LP} \quad \text{--- weak duality.}$$

$$\le 10 \cdot OPT \quad (\text{Integer LP} \subseteq \text{LP})$$

where OPT is the optimal solution for ILP.

To conclude, We proved that the algorithm LP-FACILITY LOCATION is a 10-approx algorithm for this case #.

Exercise 3.

[P.S. I looked up to 'cs.princeton.edu/courses/archive/fall14/cos521/lecnotes/lec15.pdf' for references.]

Solution3: first, we introduce the corresponding SDP. problem.

define $n+1$ vectors:

$u_0, u_1, \cdots, u_n$., the first vector u_0 is a dummy vector(fixed), Let us set $\|u_0\|^2 = 1$.

and consider a general-clause-form: $y_i \vee y_j$.

We would like to say that if $u_i = u_0$, this means set y_i= TRUE, if $u_i = -u_0$, this means set y_i= FALSE.

So, $y_i \vee y_j$ can be expressed as follows:

$$1 - \frac{1}{4}(u_0 - u_i)(u_0 - u_j) = \begin{cases} 0, & u_i = u_j = -u_0 \\ 1, & u_i = u_0 \text{ OR } u_j = u_0 \end{cases}$$

(*).

So, the SDP problem is to find these vectors satisfying $|u_i|^2 = 1$, for all i as to maximize $\sum_{\text{clause-}l} v_l$, where v_l is the expression for lth clause. (*)

So, the Algorithm 'SDP-MAX-2SAT' could be described as follows:

Algorithm SDP-MAX-2SAT

1. Solve the above SDP problem to obtain the optimal solution. $u_1, u_2 \cdots u_n$
2. choose a uniformly random unit vector $\vec{r} \in \mathbb{R}^n$.
3. if $\langle \vec{r}, \vec{u_0} \rangle \cdot \langle \vec{r}, \vec{u_i} \rangle \geq 0$, set $\hat{y_i}$ = TRUE
   otherwise, set y_i = FALSE
4. output $\{y_1, y_2, \cdots, y_n\}$.

Then, we will prove that the above algorithm is a 0.878-approximation.

Proof.

Tips: $\hat{u}$ represent: rounded
u_i represent: relaxed

e.g. for $y_i \vee y_j$.

$$1 - \frac{1}{4}(\hat{u}_0 - \hat{u}_i)\cdot(\hat{u}_0 - \hat{u}_j) = \frac{1}{4}(1+\hat{u}_0\cdot\hat{u}_j) + \frac{1}{4}(1+\hat{u}_0\cdot\hat{u}_i) + \frac{1}{4}(1-\hat{u}_i\hat{u}_j)$$

We will consider the above expression term by term.

e.g.

$$1 + \hat{u}_0\hat{u}_j = \begin{cases} 2, & \hat{u}_j = \hat{u}_0 \\ 0, & \hat{u}_j = -\hat{u}_0 \end{cases}.$$

$$\Rightarrow E[1+\hat{u}_0\hat{u}_j] = 2\cdot P(\hat{u}_j = \hat{u}_0)$$

$$= 2\cdot\frac{\pi-\theta_{j,0}}{\pi} = 2\cdot\frac{\pi-\theta_{j,0}}{\pi(1+\cos\theta_{j,0})}\cdot(1+\cos\theta_{j,0})$$

$$\geq 0.878\cdot(1+\cos\theta_{j,0})$$

$$= 0.878\cdot(1+u_0\cdot u_j) \quad ①$$

$$1 - \hat{u}_i\hat{u}_j = \begin{cases} 2, & \hat{u}_i = -\hat{u}_j \\ 0, & \hat{u}_i = \hat{u}_j \end{cases}$$

$$\Rightarrow \begin{cases} \hat{u}_i = \hat{u}_j \iff \langle u_i, r\rangle \cdot \langle u_j, r\rangle \geq 0 \\ \hat{u}_i = -\hat{u}_j \iff \langle u_i, r\rangle \cdot \langle u_j, r\rangle \leq 0 \end{cases}$$

从而 $E[1-\hat{u}_i\hat{u}_j] = \frac{2\theta_{ij}}{\pi} = \frac{2\theta_{ij}}{\pi(1-\cos\theta_{ij})}(1-\cos\theta_{ij})$

$$\geq 0.878(1-\cos\theta_{ij})$$

$$= 0.878(1-u_i\cdot u_j) \quad ②$$

结合①、②可知：

$$E\left[\frac{1}{4}(1+\hat{u}_i\hat{u}_0)+\frac{1}{4}(1+\hat{u}_j\cdot\hat{u}_0)+\frac{1}{4}(1-\hat{u}_i\cdot\hat{u}_j)\right]$$

$$= \frac{1}{4}\times\left\{0.878\left[(1+u_iu_0)+(1-u_i\cdot u_j)+(1+u_ju_0)\right]\right\}$$

$$= 0.878\times OPT_{SDP} \geq 0.878\cdot OPT_{MAX\text{-}2SAT}.$$

#