2019 ADA HW 4

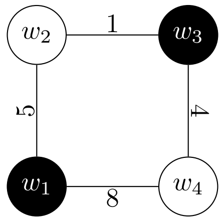
b07902064 資工二 蔡銘軒

January 6, 2020

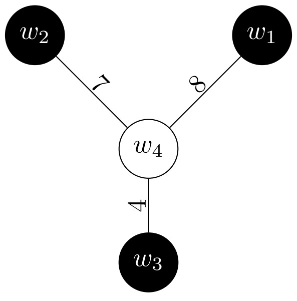
Problem E

(1)

(1-1) The maximum power enhanced is $2 \cdot (5+4+8+1) = 36$, and the optimal linking method is as follows:



(1-2) The maximum power enhanced is $2 \cdot (8+7+4) = 38$, and the optimal linking method is as follows:



(2)

(2-1) We reduce the Maximum Cut Problem to the given problem by the following:
For a given graph $G = \langle V, E\rangle$ to the Maximum Cut Problem, we construct N wands such that $N = |V|$ and each vertex corresponds to a wand. The fitness values are given by:

$$fitness(u,v) = \begin{cases} 1 \text{ if } (u,v) \in E \\ 0 \text{ if } (u,v) \notin E \end{cases} \quad \text{for } u, v \in V \text{ and } u \neq v$$

We first point out that the a linking method corresponds to a "cut" in G, and vice versa, by our construction of wands and the definition of "cut".
We prove that the linking method is optimal if and only if the edges corresponding to the links are the maximum cut set in G.

$\Longrightarrow$ direction. Let $2x$ be the enhanced power in the optimal linking method L. By the construction of wands and fitness values, there are x edges in the corresponding cut in G. Suppose the opposite that the cut is not maximum, then there is another cut C in G with y edges, where $y > x$. We can construct another linking method L' using the edges in C, and by our construction of fitness values, L' enhances the power by $2y$, contradicting the fact that L is the optimal linking method.

$\Longleftarrow$ direction. Let x be the number of edges in the maximum cut C in G. Similar to the argument above, there is a corresponding linking method that enhances power by $2x$. If there is a linking method L which enhances power by $2y$ where $y > x$, then by our construction, the links in L corresponds to a cut with y edges in G, contradicting the fact that C is the maximum cut.

(2-2) The algorithm runs n iterations. In each iteraction, it calculate the fitness values with $O(n)$ elements, so the overall time complexity of the algorithm is $O(n^2)$, which is polynomial.
As for the approximation ratio, let the power enhanced by the optimal linking method be C^*. We have

$$C^* \le 2 \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} fitness(w_i, w_j)$$

That is, C^* is less than or equal to twice the sum of all fitness values. Equality holds when $N = 2$. For $N > 2$, inequality is obvious.
Let C_i be the power enhanced by the algorithm in the *i-th* iteration, and S_i be twice the sum of all fitness values for all wands $\{w_1, w_2, \dots, w_i\}$ discovered before and in the *i-th* iteration. We show by induction that $S_i \le 2 \cdot C_i$ for $i = 2, \dots, N$.
For $i = 2$, $C_2 = S_2 = 2 \cdot fitness(w_1, w_2)$, so $S_2 \le 2 \cdot C_2$.
In the *i-th* iteration, $i > 2$, suppose $W_1 = \{u_1, u_2, \dots, u_j\}$ and $W_2 = \{v_1, v_2, \dots, v_k\}$ such that $W_1 \cap W_2 = \emptyset$ and $W_1 \cup W_2 = \{w_1, w_2, \dots, w_{i-1}\}$. Suppose the algorithm adds w_i to W_1, then we have

$$\sum_{a=1}^{k} fitness(w_i, v_a) \ge \sum_{a=1}^{j} fitness(w_i, u_a)$$

which gives

$$2C_i = 2C_{i-1} + 2 \cdot 2 \sum_{a=1}^{k} fitness(w_i, v_a) \ge S_{i-1} + 2 \cdot \sum_{a=1}^{k} fitness(w_i, v_a) + 2 \cdot \sum_{a=1}^{j} fitness(w_i, u_a) = S_i$$

Similar calculation can be applied if the algorithm adds w_i to W_2. Finally, we have

$$C^* \le 2 \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} fitness(w_i, w_j) = S_n \le 2 \cdot C_n$$

where C_n is the result returned by the algorithm. This completes the proof.

(3) We reduce problem (2-1) to the given problem with **RULE 1** and **RULE 2** by:
For the given N wands and M fitness values, we assume the fitness values are not all zero, otherwise we return 0. We add N dummy wands such that their fitness values with all other wands are zero and use it as the input for the given problem.

Let L be a linking method for the given problem, and L' be the linking method for problem (2-1) derived from L by removing the dummy points and related links. We note that the power enhanced by L is the same as the power enhanced by L', as the links conntected with the dummy points enhance no power, so their removal does not affect the power enhancement. We show that L' is optimal if and only if L is optimal.

$\Longrightarrow$ direction. Let p be the power enhanced by L and L'. Suppose L is not optimal, there exists another linking method L'' for problem (2-1) such that it enhanced q power where $q > p$. Suppose there are x black vertices and $N - x$ white vertices in L'', we can construct a linking method that enhances q power by coloring $N - x$ dummy points black and x dummy points white, contradicting the fact that L' is optimal.

$\Longleftarrow$ direction. Let p be the power enhanced by L and L'. If L' is not optimal, there exists another linking method L'' for the given problem that enhances power by q where $q > p$. We can obtain a linking method which enhances q power for problem (2-1) by removing the dummy points in L'', contradicting the fact that L is optimal.

(4) We first note that the optimal linking method in problem (3) is a complete bipartite for the reason of optimality; the graph representation of any linking method in the given problem is also a complete bipartite as required by the problem description.

We reduce problem (3) to the given problem by:
For the given N wands and M fitness values for problem (3), let m be the maximum fitness value, we

construct the input for the given problem where the wands are identical, and the fitness values for each link (u, v) is defined by $m - fitness(u, v)$. Note that the new fitness values are non-negative.
We show the set of links is optimal in the given problem if and only if it is optimal in problem (3).

$\Longrightarrow$ direction. Let L be the optimal linking method for the given problem, and $\{x_1, x_2, \ldots, x_{\frac{N^2}{4}}\}$ be the fitness values in L, then the corresponding fitness values in problem (3) is $\{m-x_1, m-x_2, \ldots, m-x_{\frac{N^2}{4}}\}$. Suppose there exists another linking method L' for problem (3) with fitness values $\{m-y_1, m-y_2, \ldots, m-y_{\frac{N^2}{4}}\}$ such that

$$\sum_{i=1}^{\frac{N^2}{4}} m - y_i > \sum_{i=1}^{\frac{N^2}{4}} m - x_i \Longrightarrow \sum_{i=1}^{\frac{N^2}{4}} y_i < \sum_{i=1}^{\frac{N^2}{4}} x_i$$

Then by the above inequality, L' is a better linking method for the given problem, contradicting the fact that L is optimal.

$\Longleftarrow$ direction. Let L be the optimal linking method in problem (3). With the similar argument above, if L is not optimal for the given problem, there exists a linking method L' such that the sum of fitness values is smaller than that of L and L' is also a better linking method for problem (3), contradiction.

(5) The reduction works as follows: Let $A = \{a_1, a_2, \ldots, a_n\}$ and W be the input to the Subset Sum Problem, where A is a set of n non-negative integers and W is a positive integer. We reduce it to the given one by: Let $H = \frac{1}{2}\sum_i^n a_i$, add $a_{n+1} = 2H + 2W$ and $a_{n+2} = 4H$ to A and denote the new set as A'. The reduction involves calculating the sum of a set and adding two elements, which can be done in polynomial time. We show that the answer to the given problem is true if and only if the answer to the Subset Sum Problem is true.

$\Longrightarrow$ direction. We have $\sum_{i=1}^{n+2} a_i = 8H + 2W$. If A' can be partitioned into two sets with sum $4H + W$, clearly one set must contain a_{n+2} and some $a_i \in A$ such that those elements in A sum up to W.

$\Longleftarrow$ direction. If there is a subset of A whose sum is W, then we add a_{n+2} to the subset, expanding its sum to $4H + W$, which is $\frac{1}{2}\sum_{i=1}^{n+2} a_i$.

Problem F

(1)

(1-1) Before starting, we prove the following:

Lemma 1. *Let $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ be two sets of positive real numbers, then $\min_{1\le i\le n} \frac{a_i}{b_i} \le \frac{\sum_i^n a_i}{\sum_i^n b_i}$*

Proof. Let $r = \min_{1\le i\le n} \frac{a_i}{b_i}$, then we have $a_i \ge r \cdot b_i$ for $1 \le i \le n \Longrightarrow \sum_i^n a_i \ge r \cdot \sum_i^n b_i$ □

Let $\text{price}(x_i) = \frac{\text{cost}(S)}{|S \cap U|}$ such that $x_i \in S$ and x_i is added to C by the discovery of S. Let U_i be the set of uncovered elements at the beginning of the *i-th* iteration. Particulary, $U_1 = X$. Consider the *i-th* iteration. Let O be the set of sets that the optimal solution chooses to cover the elements in U_i. We note that the algorithm has not yet chosen any set in O, for otherwise some elements in U_t would have been covered, contradicting the definition of U_t. Let S_t be the set the algorithm chooses in the iteration, we have

$$\frac{\text{cost}(S_t)}{|S_t \cap U_t|} \le \min_{S_i \in O_t} \frac{\text{cost}(S_i)}{|S_i \cap U_t|}$$

We note that, let m be the minimum index such that x_m is added to C by S_t, then $t \le m$, which also implies $|U_t| \le n - t + 1$. By lemma 1, for every x_k such that $x_k \in S_t$ and is added to C by S_t, we have

$$\text{price}(x_k) = \frac{\text{cost}(S_t)}{|S_t \cap U_t|} \le \min_{S_i \in O_t} \frac{\text{cost}(S_i)}{|S_i \cap U_t|} \le \frac{\sum_{S_i \in O_t} \text{cost}(S_i)}{\sum_{S_i \in O_t} |S_i \cap U_t|} \le \frac{\text{OPT}}{|U_t|} \le \frac{\text{OPT}}{n - k + 1}$$

(1-2) The algorithm runs for at most m iterations. In each iteration, it looks at at most m sets and can calculate $|S \cap U|$ in $O(n)$. The update and price assignment can all be done in $O(n)$, so the algorithm can run in $O(m^2n)$ time or faster with proper implementation.
As for the approximation ratio, by the result in (1-1), we have

$$\sum_{i=1}^{n} \text{price}(x_i) \leq \sum_{i=1}^{n} \frac{\text{OPT}}{n-i+1} = \sum_{i=1}^{n} \frac{\text{OPT}}{i} \leq (ln(n)+1) \cdot \text{OPT}$$

So the algorithm is a $(ln(n) + O(1))$-approximation.

(2)

(2-1) The reduction works as follows:
On input $M = \{M_1, M_2, \dots, M_n\}$, construct the universe $X = M$, and $F = M \cup K$, where K consists of elements of form $K_{ij} = \{M_i, M_j\}$, such that M_i and M_j can be merged. It is obvious that F is finite. The cost of each element $S \in F$ is defined by

$$\text{cost}(S) = \begin{cases} |S|, \text{ if } S \in M \\ |merge(M_i, M_j)|, \text{ if } S \in K \text{ and } S = \{M_i, M_j\} \end{cases}$$

Let $A = \{A_1, A_2, \dots, A_j\}$ be the set the Set Cover Problem chooses, we can construct a string that contains all substrings by concatenating all elements in A. If $A_i = \{M_i, M_j\} \in K$, we concatenate $merge(M_i, M_j)$.
Then we show the reduction can be done in polynomial time.

- Copy M to $X \implies O(n \cdot l)$
- Copy M to $F \implies O(n \cdot l)$
- Enumerate and compute all possible mergers and copy to $F \implies O(n^2) \cdot O(l^2) = O(n^2l^2)$. $O(n^2)$ for enumerating pairs and $O(l^2)$ for computing mergers and copy to F.

Both X and F can be done in polynoial time, so the reduction is complete.

(2-2) Let $s = a_1a_2 \dots a_{\text{OPT}_{\text{Kasumi}}}$ be the shortest string that contains all the substrings. Let $M = \{M_1, M_2, \dots, M_n\}$ be ordered according to their appearance in s from left to right. Since no string is a substring of any other string, for $i < j$, M_i must start and end before M_j. Let $b_1 = 1$, the index of the first alphabet of M_1, and $e_1 = k$ such that a_k is the end of some string $M_j \in M$ and M_j is the rightmost string that overlaps with M_1. Let b_2 be the index of the first alphabet of M_{j+1}, and define e_2 similarly to e_1. Finally we have $\{(b_1, e_1), (b_2, e_2), \dots, (b_f, e_f)\}$, where $e_f = \text{OPT}_{\text{Kasumi}}$.

Let $L = \{L_1, L_2, \dots, L_f\}$, such that $L_i = a_{b_i} \dots a_{e_i}$. Clearly each L_i is either a single string in M or a merger of two strings in M by definition of b_i and e_i, so it has a corresponding element in F of the Set Cover Problem according to our construction. This also implies that, the set $\{L'_1, L'_2, \dots, L'_f\}$, where $L'_i \in F$ and it corresponds to L_i, is a valid answer to the Set Cover Problem, with total cost $\sum_{i=1}^{f} |L_i|$.

We show that each alphabet $a_i \in s$ does not appear in more than two elements in L, which is equivalent to saying L_i does not overlap with L_{i+2} for $i = 1, 2, \dots, f-2$. Suppose the oppposite that L_i overlaps with L_{i+2} for some $i \in \{1, 2, \dots, f-2\}$, then $e_i \geq b_{i+2}$. By our construction, we have $b_{i+1} > e_i$ and $b_{i+1} < b_{i+2}$, which give $e_i < b_{i+2}$, contradiction.

Following the fact, we immediately have $\text{OPT}_{SETCOVER} \leq \sum_{i=1}^{f} |L_i| \leq 2 \cdot \text{OPT}_{\text{Kasumi}}$, which shows that the reduction is 2-approximate. Combined with the result in (1-2), the reduction gives a $2 \cdot (ln(n) + O(1)) = (2 \cdot ln(n) + O(1))$ solution to Kasumi's problem.

Reference

Problem E:

(2-1), (3) b07902026 陳玉恆 b07902028 林鶴哲
(4) b07902141 林庭風

Problem F:

(1-1) `https://www.cslog.uni-bremen.de/teaching/summer17/approx-algorithms/resource/lec3.pdf`
(2-2) `https://www.cs.dartmouth.edu/~ac/Teach/CS105-Winter05/Notes/wan-ba-notes.pdf`