SHANGHAI JIAO TONG UNIVERSITY

# Algorithm Analysis HW:6

Author: *519030910100 Huangji Wang F1903004*

Course: *Fall 2021, AI2615: Algorithm*
Date: *December 31, 2021*

**Problem 1 (50 points)**

Consider the following variant of the scheduling problem. We have $n$ jobs with size (time required for completion) $p_1, ..., p_n \in Z^+$. Instead of fixing the number of machines $m$ and minimizing the makespan, we fix the makespan and minimize the number of machines used. That is, you can decide how many (identical) machines to use, but each machine can be operated for at most $T \in Z^+$ units of time. Assume $p_i \leq T$ for each $i = 1, ..., n$. Your objective is to minimize the number of machines used, while completing all the jobs.

(a) (20 points) Show that this minimization problem is NP-hard.

(b) (20 points) Consider the following local search algorithm. Initialize the solution where n machines are used such that each machine handles a single job. Do the following update to the solution until no more update is possible: if there are two machines such that the total size of the jobs on the two machines is less than $T$, update the solution by using only one machine to complete all these jobs (instead of using two machines). Show that this algorithm gives a 2-approximation.

(c) (10 points) Provide a tight example showing that the algorithm in (b) can do 1.5-approximation at best.

(a) Proof:

To prove this minimization problem is NP-hard is to prove that there exists $k \in R$ such that **deciding** whether $OPT \leq k$ is NP-hard. Therefore, we can just fix the machine number $k$ and the problem is changed into checking whether $k$ is feasible in this job scheduling. Actually, this step aims at proving that the new decision problem is in NP.

Secondly, this problem can be reduced into Makespan Problem. I present a reduction from Makespan Problem. Given a Makespan Problem instance $S = \{p_1, ..., p_n\}, m \in Z^+, \min T$, we construct a **Scheduling** instance $S' = \{p_1, ..., p_n\}, k \in Z^+, \min T' \leq T$ as follows.

The job set $S$ is the same because these two problems aim at solving the schedule of $n$ jobs. The machine's number $k$ in Scheduling is the same as the number $m$ in Makespan because at this moment we have already fix the $k$. The goal is to check whether the jobs can be done with the help of $k$ machines within total time cost $T$. Therefore, we can easily make sure that the new decision problem is equivalent to the Makespan

Problem. Notice that Makespan Problem is NP-Complete, and thus the Scheduling problem is also NP-Complete and thus the original problem is NP-Hard.

(b) Proof:

Firstly, we can make sure that after enough rounds no more update is possible, which means that for any machine $i$ with workload $w_i$, it cannot find another machine $j$ such that $w_i + w_j \leq T$, which implies that $\forall i, j \in [1, n], w_i + w_j > T$. Thus, we can take two machine as one pair. For the optimal solution $OPT$, it has the total workload at most $OPT * T$. And the workload of this should satisfy that

$$OPT * T \geq workload = \Sigma(w_1 + ... + w_{ALG}) > \frac{ALG}{2} * T$$

, which implies that $ALG < 2 * OPT$ and this implies that this algorithm gives a 2-approximation.

(c) Example:

Jobs with $3, 1, 2, 4$ and T = 5 will have that:

$$w_1 = \{3, 1\}, w_2 = \{2\}, w_3 = \{4\}$$

and the optimal solution is

$$w_1 = \{1, 4\}, w_2 = \{2, 3\}$$

This implies that the local search algorithm can reach 3 while the optimal solution can reach 2 and the result is $3 = 2 * 1.5$, which means the algorithm in (b) can do 1.5-approximation at best.

**Problem 2 (50 points)**

Choose any one of the following questions. (You are encouraged to solve as many the remaining questions as possible in your mind.)

(a) Given an undirected graph $G = (V, E)$ with $n = |V|$, decide if $G$ contains a clique with size exactly $\frac{n}{2}$. Prove that this problem is NP-complete.

(b) Given an undirected graph $G = (V, E)$, the $3 - coloring$ problem asks if there is away to color all the vertices by using three colors, say, red, blue and green, such that every two adjacent vertices have different colors. Prove that 3-coloring is NP-complete.

(c) Given two undirected graphs $G$ and $H$, decide if $H$ is a subgraph of $G$. Prove that this problem is NP-complete.

(d) Given an undirected graph $G = (V, E)$ and an integer $k$, decide if $G$ has a spanning tree with maximum degree at most $k$. Prove that this problem is NP-complete.

(e) Given a ground set $U = 1, ..., n$ and a collection of its subsets $S = S_1, ..., S_m$, the exact cover problem asks if we can find a subcollection $T \in S$ such that $\bigcup_{S \in T} S = U$ and $S_i \cap S_j = \emptyset$ for any $S_i, S_j \in T$ . Prove that exact cover is NP-complete.

(f) Given a collection of integers (can be negative), decide if there is a subcollection with sum exactly 0. Prove that this problem is NP-complete.

(g) In an undirected graph $G = (V, E)$, each vertex can be colored either black or white. After an initial color configuration, a vertex will become black if all its neighbors are black, and the updates go on and on until no update is possible. (Notice that once a vertex is black, it will be black forever.) Now, you are given an initial configuration where all vertices are white, and you need to change $k$ vertices from white to black such that all vertices will eventually become black after updates. Prove that it is NP-complete to decide if this is possible.

(a) Given an undirected graph $G = (V, E)$ with $n = |V|$, decide if $G$ contains a clique with size exactly $\frac{n}{2}$. Prove that this problem is NP-complete.

**1.Prove that $f$ is in NP.**

First of all, Clique Problem is in NP , as a clique set $S$ can be served as a certificate, and it can be verified in polynomial time whether $S$ is a clique set and whether $|S| = k$.

**2.Present the reduction $g \leq_k f$ for an NP-complete problem $g$.**

To show that clique problem is NP complete, we present a reduction from Independent Set. Given a Independent instance $G = (V, E), k \in Z^+$ construct a clique instance $G' = (V', E'), k' \in Z^+$ as follows:

The vertex set is $V' = V$, which is defined as follows. For each vertex $v \in V$ in the Independent Set instance, it can be directly mapped to the vertex in $V'$.

The edge set is $E' = \overline{E}$, which is defined as follows.

$$\forall u, v \in V, (u, v) \notin E \implies (u, v) \in E' = \overline{E}$$

Define $k' = k$.

**3.yes $g$ to yes $f$.**

Suppose $(G = (V, E), k)$ is a yes independent set instance. There exists a vertex cover $S \in V$ with $|S| = k$. We will prove $\overline{S}$ corresponding $S$ is a clique set in $G'$. Thus, we can make sure that there exists an independent set of size $k$ in $G = (V, E)$. For this independent set, we consider its complement graph and we can find that its complement graph is a complete graph. Otherwise, it will contradict the truth that any two vertex will not have edge in the origin graph. Therefore, when we build a yes instance of $g$ and since $|S| = |S'| = k = k'$, it implies the yes instance of $f$.

**4.yes $f$ to yes $g$.**

In the similar way, suppose $(G' = (V', E'), k')$ is a yes clique set instance. Then, there exists a clique with size exactly $k'$. Thus, if we take the complement graph of this clique, we can have a graph with no edges and $k'$ vertices by the definition. Due to the reason that there are no edges among the $k'$ vertices, we can ensure that this is exactly an independent set with size $k'$. Note that the new independent set is a complement graph of the origin $G'$ and thus we build a yes instance of $f$ and it implies the yes instance of $g$.

At last, even if we prove that for all clique problem with size $k$ is NP-Complete, it is useless to prove the specialize situation of $\frac{n}{2}$.

Let $(G = (V, E), k)$ be an instance of the clique problem, and suppose G has n nodes. We produce an instance of the $\frac{n}{2}$ clique problem, as follows:

- $k = \frac{n}{2} \implies$ actually G has a $\frac{n}{2}$ clique if and only if it has a clique of size $k$. And at this point these two points are the same.
- $k > \frac{n}{2} \implies$ we can add $2k - n$ isolated nodes and thus the graph in $G$ can have a $\frac{2*k}{2} = k$ clique because the size of the new graph is $n + (2k - n) = 2k$. Thus the new graph has a $\frac{n}{2}$ clique if and only if the origin graph $G$ has a clique of size $k$.
- $k < \frac{n}{2} \implies$ we can add $n - 2k$ nodes and add connect all the new nodes to each other and to the original graph $G$, leading to a new graph with $2 * (n - k)$ nodes and at least a clique with size $n - 2k$. Next, we know there exists a clique with size $k$ in the origin graph(and it has no connection with the added $n - 2k$ nodes). Therefore, if we add all the connections between $n - 2k$ nodes and $k$ nodes it can yield a new clique with size $n - 2k + k = n - k$. The new graph has size $2(n - k)$ and it can have a clique with $n - k$.

Thus, based on these three steps we can form a reduction from the clique problem to $\frac{n}{2}$ clique problem and it is obviously the reduction time is in polynomial since the adding method can cost at most $O(N^2)$. Therefore, the $\frac{n}{2}$ clique problem is an NP-complete problem.

(b) Given an undirected graph $G = (V, E)$, the $3-coloring$ problem asks if there is away to color all the vertices by using three colors, say, red, blue and green, such that every two adjacent vertices have different colors. Prove that 3-coloring is NP-complete.

**1.Prove that $f$ is in NP.**

First of all, $3-coloring$ Problem is in NP , as a coloring set $S$ can be served as a certificate, and it can be verified in polynomial time whether $S$ is a $3-coloring$ set.

**2.Present the reduction $g \leq_k f$ for an NP-complete problem $g$.**

To show that $3-coloring$ problem is NP complete, we present a reduction from SAT Set. Based on the same condition that $G = (V, E), S = \{x_1, ..., x_n\}$, given a SAT problem instance with graph $G' = (V', E'), S' = \{x_{1,1}, x_{1,2}, x_{1,3}, ..., x_{n,1}, x_{n,2}, x_{n,3}\}$ where the element $x_{i,j}$ means the colored status of the $i$-th node. $j = 1(red), 2(blue), 3(green)$. The SAT instance $\phi$ can be composed by $\forall v \in V$:

$$\phi_V = \bigcap_{v \in V} (x_{v,1} \vee \neg x_{v,2} \vee \neg x_{v,3}) \wedge (x_{v,2} \vee \neg x_{v,1} \vee \neg x_{v,3}) \wedge (x_{v,3} \vee \neg x_{v,1} \vee \neg x_{v,2})$$

and $\forall (u, v) \in E$, we suppose that $\phi_{e,i} = (x_{u,i} \vee \neg x_{v,i}) \wedge (x_{v,i} \vee \neg x_{u,i})$

$$\phi_E = \bigcap_{(u,v) \in E} \phi_{e,1} \wedge \phi_{e,2} \wedge \phi_{e,3}$$

The final SAT instance is

$$\phi = \phi_V \wedge \phi_E = true$$

Construct a $3-coloring$ instance as follows:

Both the edge sets are the same while the vertex set in the SAT problem is generated by dividing each vertex into 3 conditions: red, blue and green and thus $|V'| = 3|V|$.

The value assignment solution $S$ can be applied both in these two problems.

WLOG, I suppose that $x_i = 1(red), 2(blue), 3(green)$.

**3.yes $g$ to yes $f$.**

Suppose $(G = (V, E), k)$ is a yes SAT problem instance. Thus, we can find a value assignment(if have) for the instance $\phi$, which implies two end points in the same edge cannot be colored the same color for all edges. Notice that the color domain has just 3 types, this is exactly the true instance of the $3-coloring$ problem and the value assignment solution in the SAT problem can be the solution of the $3-coloring$ problem. Thus, yes $g$ implies yes $f$.

**4.yes $f$ to yes $g$.**

In the similar way, suppose $G = (V, E), S = \{x_1, ..., x_n\}$ is a yes $3-coloring$ set instance. Then, every two adjacent vertices have different colors. This exactly means that $\forall(u,v) \in E, x_u \neq x_v$. Therefore, we can make sure that no two adjacent vertices have the same color based on the problem and thus the intersection of each edge should turns out to be 1 and the final result of the SAT problem instance should be *true*. Thus, yes $f$ to yes $g$.

Above all, $3 - coloring$ problem is NP-complete.

(c) Given two undirected graphs $G$ and $H$, decide if $H$ is a subgraph of $G$. Prove that this problem is NP-complete.

This time I try to use some different method to prove this problem is NP-complete. The main idea is that the reduction will be based on the clique problem. To find a subgraph is to find the isomorphism subgraph. Thus, to translate this problem to a subgraph isomorphism problem, we can try to let $H$ be the complete graph $K^k$ and thus the answer to the new problem for $G$ and $H$ can be changed into the clique problem for $G$ and k. We know that the clique problem is NP-complete(based on problem(a)) and thus this subgraph isomorphism problem is also NP-complete.

(d) Given an undirected graph $G = (V, E)$ and an integer $k$, decide if $G$ has a spanning tree with maximum degree at most $k$. Prove that this problem is NP-complete.

The idea is to do reduction from Hamiltonian Path. If k=2 it is exactly the Hamilton Path and if larger it is obviously harder than $k = 2$ condition and thus it is NP-complete.

[Note:] Due to lacking of time, for the next problems I might just provide my reduction idea and I won't give the full proof.

(e) Given a ground set $U = 1, ..., n$ and a collection of its subsets $S = S_1, ..., S_m$, the exact cover problem asks if we can find a subcollection $T \in S$ such that $\bigcup_{S \in T} S = U$ and $S_i \cap S_j = \emptyset$ for any $S_i, S_j \in T$ . Prove that exact cover is NP-complete.

This problem aims at finding a best set cover for the ground set $U$ because there is one restriction that any two subsets in the subcollection have no same elements. Thus, this problem can be reduced into a minimum vertex cover problem. The instance is presented as followed:

For the minimizing vertex cover problem $G = (V, E)$, we can let the ground set $U = E$ and the collection of its subsets $S_i$ be the set of edges that directly connect the $i$-th vertex, where all the vertices are taken from the vertex set $V$. To find the minimum vertex cover in $G = (V, E)$ is to find the minimum set cover in the original problem. (Minimum vertex cover is also NP-complete and this implies the exact set cover is NP-complete).

[**Question**] I think this seems true but I don't get well through this problem.

(f) Given a collection of integers (can be negative), decide if there is a subcollection with sum exactly 0. Prove that this problem is NP-complete.

Try to reduce it to the Subset Sum problem and based on $k$. We choose the problem of subset sum with arbitrary $k'$ and we just need to add $S'$ with $-k$ to build $S = S' \cup \{-k\}$

$$S = \{a_1, ..., a_n, -K\} \text{ has zero iff } S' = \{a_1, ..., a_n\} \text{ has K}$$

(g) In an undirected graph $G = (V, E)$, each vertex can be colored either black or white. After an initial color configuration, a vertex will become black if all its neighbors are black, and the updates go on and on until no update is possible. (Notice that once a vertex is black, it will be black forever.) Now, you are given an initial configuration where all vertices are white, and you need to change $k$ vertices from white to black such that all vertices will eventually become black after updates. Prove that it is NP-complete to decide if this is possible.

**Problem 3**

How long does it take you to finish the assignment (include thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Write down their names here.

I take 5 hours in total to finish the assignment(include thinking and discussing).
Difficulty: 3.
Collaborators: wikipedia in problem(2).(c)