Algorithm Homework 6 NP

Jingwei Zhang 201528013229095

2015-1-7

1 Problem 1

proof: Integer-programming is NPC.

Firstly, by checking $Ax \geq b$, we can verify this problem in polynomial time (matrix multiplication takes $O(mn^2)$ and comparison takes $O(m)$). Thus, Integer-programming problem is a NP problem.

Secondly, we will prove that 3SAT problem, a known NPC problem, is reducible to Integer-programming problem in polynomial time. For any instance I of 3SAT problem (its formula is ϕ), which contains m variables $x_i (i = 1, \ldots, m)$ and n clauses, we construct a instance I_i of Integer-programming as follows:

$$y_i \geq 0 \quad \text{for all } i = 1, \ldots, m \tag{1}$$

$$y_i \leq 1 \quad \text{for all } i = 1, \ldots, m \tag{2}$$

$$T_i + T_j + T_k \geq 1 \quad \text{for all clause } (X_i \vee X_j \vee X_k) \tag{3}$$

where

$$T_i = \begin{cases} y_i & \text{if } X_i = x_i \\ 1 - y_i & \text{if } X_i = \neg x_i \end{cases} \tag{4}$$

The inequalities above is equivalent to $Ax \geq b$ since every inequality can be changed to $\sum_j a_{ij} x_j \geq b_i$. $y_i = 1$ is equivalent to $x_i = true$ and $y_i = 0$ is equivalent to $x_i = false$. Then, we will prove these inequalities(or this Integer-programming problem) is equivalent to 3SAT problem.

- Suppose there exits an assignment of variables $x_i (i = 1, \ldots, m)$ such that ϕ is $true$. We assign y_i with rules mentioned above. It is obvious that inequalities (1) and (2) are satisfied. Note that (3) holds iff at least one of X_i, X_j, X_k is true. Thus, all constrains will hold.
- Suppose there exits an assignment of variables $y_i (i = 1, \ldots, m)$ such that inequalities (1), (2) and (3) hold. It is obvious that y_i equals to 0 or 1. We assign y_i with rules mentioned above. Note that (3) holds iff at least one of X_i, X_j, X_k is true. Thus, ϕ is $true$.

Thus, Integer-programming problem is NPC. ■

2 Problem 3

proof: Half-3SAT is NPC.

Firstly, by substituting the variables with the given assignment , then checking whether half of its clauses is true and half is false, we can verify this problem in polynomial time. Thus, Half-3SAT problem is a NP problem.

Secondly, we will prove that 3SAT problem, a known NPC problem, is reducible to Half-3SAT problem in polynomial time. For any instance I of 3SAT problem, which contains m variables $x_i(i = 1, \ldots, m)$ and n clauses, we construct a instance I_h of Half-3SAT as follows:

$$\phi_{Ih} = \phi_I \wedge T \wedge D \wedge \cdots \wedge D \text{ (D is repeated for } n+1 \text{ times)} \tag{5}$$

$$D = (x_{m+1} \vee x_{m+2} \vee x_{m+3}) \tag{6}$$

$$T = (x_{m+1} \vee x_{m+2} \vee \neg x_{m+2}) \tag{7}$$

ϕ_I represents the CNF of I. I_h contains $2n+2$ clauses and $m+3$ variables. Note that clause $T = x_{m+1} \vee x_{m+2} \vee \neg x_{m+2}$ is always true. Then, we will prove I and I_h are equivalent.

- Suppose there exits an assignment of variables $x_i(i = 1, \ldots, m)$ such that ϕ_I is *true*. If we assign $x_{m+1} = x_{m+2} = x_{m+3} = false$, which leads to $D = false$, ϕ_{Ih} will be separated into $n+1$ *false* clauses(all D) and $n+1$ *true* clauses(n clauses of ϕ_I and T). Thus, there exists an assignment such that exactly half the clauses of ϕ_{Ih} evaluate to false and exactly half the clauses of ϕ_{Ih} evaluate to true.

- Suppose there exists an assignment such that exactly $n+1$ clauses of ϕ_{Ih} evaluate to false and exactly $n+1$ clauses of ϕ_{Ih} evaluate to true. Since T is always *true*, if $D = true$, there are at least $n+2 > n+1$ clauses with value *true*. This assignment is not possible under previous assumption. If $D = false$, there are 1 *true* clause(T) and $n+1$ *false* clause. Thus, the remaining n clauses of ϕ_I must be *true*.

Thus, Half-3SAT problem is NPC. ∎