Problem Solving Homework (Week 5)

161180162 Xu Zhiming

April 10, 2018

JH Chapter 3

3.6.1.3

Proof. According to the definition, we can verify the following three conditions:

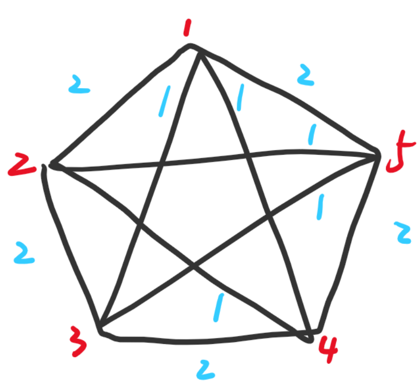
1. Suppose $\alpha \in M(x)$, then $\alpha \in f_x(\alpha)$ because, if we flip no variable in α, α is mapped to itself
2. $\alpha \in M(x)$ and $\beta \in f_x(\alpha)$. Then we have α and β have at most one variable that is not the same. If $\alpha = \beta$, from 1. we have $\alpha \in f_x(\beta)$. If they are not the same, flipping the only different variable will transform α to β, therefore, $\alpha \in f_x(\beta)$
3. k can be the number of all variables in MAX-SAT, γ_1 and α have one different variable. γ_k and β have one different variable. γ_i and γ_{i+1} have one different variable for $i = 1, 2, \cdots, k-1$. These assignments satisfy the definition.

□

TSP

2-exchange

Proof. The following graph servers as a counterexample. If the original solution is $\{1, 2, 3, 4, 5, 1\}$, 2-exchange will never lead to $\{1, 3, 5, 2, 4, 1\}$, which is optimal.



□

(n-1)-exchange

Proof. The former graph can also serve as a counterexample, i.e., (n-1)-exchange is still not accurate, which means the problem itself is wrong. □

Accurate local search

(1) No, it is not accurate.
Suppose the cut generated by this algorithm is S. Let v be a vertex in S. Consider the set E_v of edges incident to v. If we move v from S to $S' = V \backslash S$ Since S is a local optimum, moving v to S' does not increase the cut value. Therefore:

$$\sum_{u \in S', (u,v) \in E} w_{u,v} \geq \sum_{u \in S', (u,v) \in E} w_{u,v}$$

$$2 \sum_{u \in S', (u,v) \in E} w_{u,v} \geq \sum_{u \in S', (u,v) \in E} w_{u,v} + \sum_{u \in S', (u,v) \in E} w_{u,v}$$

$$\sum_{u \in S', (u,v) \in E} w_{u,v} \geq \frac{1}{2} \sum_{u : (u,v) \in E} w_{u,v}$$

$$\sum_{u \in S, (u,v') \in E} w_{u,v'} \geq \frac{1}{2} \sum_{u : (u,v') \in E} w_{u,v}$$

$$\therefore 2c(S) = 2 \sum_{u \in S, v \in S', (u,v) \in E} w_{u,v} \geq \sum_{e \in E} w_e \geq OPT$$

That is, this algorithm generates a solution that is no smaller than half the optimal one(s).

(2) The worst case time complexity is exponential.