Problem Solving Homework (Week 3)

161180162 Xu Zhiming

2018 年 6 月 3 日

JH Chapter 2

3.3.2.7

```
Divide-And-Conquer-MVC(G)
Find all strongly connected components with
    Tarjan's algorithm
Find the minimum vertex cover of each SCC, respectively,
    with the algorithm introduced in 3.3.2.4
Combine vertex covers of each SCC
return the global minimum vertex cover
```

REVISE: Modify Step 3 in Algorithm 3.3.2.4 from exhaustive search to divide-and-conquer, i.e., $(G, m) \in VC \iff (G_1, m-1) \in VC) \vee ((G_2, m-1) \in VC)$.

3.3.2.8

证明. To prove the given statement, It is needed to show that there exists a *Var-parameterized polynomial-time algorithm* for Max-Sat. What follows is one of such algorithms. Suppose there are n Boolean functions in such CNF.

```
  Var-Para-For-Max-Sat(Φ)
1 Make all possible assignments for all Φ:
        // With this assignment, test whether CNF is true
2       if assignment {A_{k_1}, A_{k_2}, ..., A_{k_n}} for
        {Φ_1, Φ_2, ..., Φ_n} satisfies CNF
3             return TRUE
4 return FALSE
```

Since it has tested all possible assignments, and will return true if such CNF can be satisfied, yield false, otherwise. It solves the problem. Then, we prove it to be *Var-parameterized polynomial-time* where $Var(\Phi_i)$ is the variables occurring in Φ_i. The time complexity for test a series of assignments in line 2 is $O(n)$. However, line 1 will run

$$\sum_{i=1}^{n} 2^{Var(\Phi_i)}$$

times. Hence the total running time is

$$O\left(n \cdot \sum_{i=1}^{n} 2^{Var(\Phi_i)}\right) = O\left(n^2 \cdot 2^{\max(Var(\Phi_i))}\right).$$

Therefore, Max-Sat is fixed-parameterized polynomial-time tractable according to *Var*. □

3.3.2.9

```
Max-Set-Cover(X, F, k)
Set T = ∅ and C = ∅
if |T| < k and |C| < p + Δ−1
      Pick a set S_i ∈ S\T that covers the largest
      number of elements in X\C
      Branch on Max-Set-Cover(T ∩ {S_i}, C ∩ S_i) and,
      ∀x ∈ S_i\C branch on Max-Set-Cover(T, C ∩ {x})
else
      if |T| = k
            Store T
      else
            (|C| = p + Δ−1) store a solution covering C
            (if possible)
Output the best among solutions stored
```

REVISED: Set cover problem. $Pot(X)$ is the set of all subsets of the set X. Similar to VCP, use divide-and-conquer.

Divide: $((X_i, F_i), k-1)$: Select any $S_i \in F$. Let $X_i = X \backslash S_i, F_i = f(F, i)$, function f deletes S_i and all elements of S_i in $S_1, \cdots, S_{i-1}, S_{i+1}, \cdots, S_l$. Using induction, $((X_T, F_T), 0)$ is trivial. Complexity is $O(Pat^{pat}|X|)$.

真・多项式时间的最大流算法

REVISE NOTE: Push relabel in CLRS

前置重贴标签算法在任何流网络上的运行时间都为 $O(V^3)$。因为每个节点的重贴标签操作数不会超过 $O(V)$，而所有节点的重贴标签操作总次数不会超过 $O(V^2)$。算法花在重贴标签上的时间不会超过 $O(VE)$，饱和推送的总次数为 $O(VE)$。综合可得，复杂度为

$$O(V^3 + VE) = O(V^3).$$