1 Clique Problem

- *Algorithm:* Finding maximum clique in a given graph (algo. 1)
- *Input:* An undirected graph with a set of vertices V and a set of edges E.
- *Complexity:* $\mathcal{O}(2^{n^{1-\epsilon/r^{1+\epsilon}}})$
- *Data structure compatibility:* Undirected Graphs.
- *Common applications:* DNA computing, sociology studies, data mining.

Clique Problem

The clique problem is to find the maximum clique in a given undirected graph. A clique is defined as all nodes are connected to each other through an edge, namely a complete subgraph.

Description

1. Problem Clarification

   Suppose we have an undirected graph with a set of vertices and a set of edges, we wish to find a fully connected subgraph with maximum nodes, which is known as a *maximum complete subgraph*. The input shall be the undirected graph.

2. Algorithm Clarification

   We will introduce a basic algorithm used to solve the clique problem, the method used is called **B&B framework**. (Branch and Bound) In algo. 1, it uses two key sets, C represents the clique and P represents all candidate vertices. The formal definition of P is :

   $$P \subset V \backslash C \cap \{\forall v \in P, \forall u \in C, u, v \in E\}$$

   The key operation of this algorithm is on line 10 and 11, line 10 is the bounding criteria, and line 11 is the branching condition.

   (a) Bound: Since we need to maximize $C*$, from the definition of the graph, $|C| + |P|$ necessarily defines a maximum bound for the clique, and if our currently discovered clique is larger than $|C| + |P|$, we shall stop since we have already found the largest clique.

   (b) Branch: During the branch operation, the property of P must be maintained. So each time we update P as $P \cap$ adjacent vertices to p. And p is chosen as the vertices from smallest degree to biggest degree.

   After each iteration, it will lead to either a bigger clique, or the function will halt on the bound condition.

3. Applications with Details and complexity

   The clique problem can be applied into many specific fields, such as DNA computing, sociology studies, data mining.

   We will discuss about an abstract application, that the clique problem can be related with a standard quadratic programming problem in graph. The problem is defined as

$$\text{To maximize } \sum_{i=1}^{n} w_i x_w, \text{ given that } x_i x_j = 0 \quad \forall i,j \in E, x_i^2 - x_i = 0$$

   Let $A_G = (a_{ij})_{i,j \in V}$ be the adjacency matrix of G, and $g(x) = x^T A_G x$. The global optimal solution is $x*$ that

$$\omega(G) = \frac{1}{1 - g(x*)}$$

   The $\omega(G)$ is the maximum clique of G. The computational complexity of this problem is given by $\mathcal{O}(2^{n^{1-\epsilon/r^{1+\epsilon}}})$.

Algorithm 1: Maximum Clique Problem

```
Input  : A graph G
Output: Clique C
Function Main(G):
                                        /* C* is the current discovered maximum clique */
    C* ← ∅;
    Clique(C*,∅, V);
    return C*;
end
Function Clique(set C*, set C, set P):
     /* C is clique, P is candidate vertex set */ /* |C| means the cardinality of C */
    if |C| > |C*| then
        C* ← C;
    end if
    if |C| + |P| > |C*| then
        for p sorted with degree-ascending in P do
            P ← P \ {p} ;
            C' ← C ∪ {p};
            P' ← P ∩ vertices adjacent to p;
            Clique(set C*, set C', set P') ;
        end for
    end if
end
```

References

- Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., & Protasi, M. (1999). Complexity and approximation: Combinatorial optimization problems and their approximability properties. Berlin: Springer.

- Babel, L., & Tinhofer, G. (1990). A branch and bound algorithm for the maximum clique problem. Methods and Models of Operations Research, 34(3), 207217.
- Engebretsen, L., & Holmerin, J. (2003). Towards optimal lower bounds for clique and chromatic number. Theoretical Computer Science, 299(13), 537584.