0.1 Vertex independent set

- *Algorithm:* greedy algorithm (algo. 2)
- *Input:* An undirected graph $G(V, E)$
- *Complexity:* $\mathcal{O}(n^3)$
- *Data structure compatibility:* Graph can be represented by either adjacency matrix(dense graph) or adjacency list(sparse graph)
- *Common applications:* Theory computer science

Problem. Vertex independent set

Given an undirected graph $G(V, E)$, and a subset V' of V is vertex independent set if and only if for any two different nodes u, v in V', there is no edge between them.

Description

We have studied the clique problem in the homework. Vertex independent set problem, however, is somewhat the opposite of clique. We have an undirected graph $G(V, E)$, a subset V' of V is called vertex independent set if and only if for any two different nodes u, v in V', there is no edge between them.

Moreover, we are interested in maximal independent set(MIS) as well. Here MIS is defined as the vertex independent set with largest cardinality in the graph. We are interested in finding algorithms to find such an MIS.

MIS problem has many theoretical applications and can be used to prove the time complexity of many other theoretical problems in the field of theory computer science. For example, we can use the linear programming(LP) knowledge we learned from lab 8 and transform the MIS problem into its dual which is minimum set cover(MSC) problem. Usually we can reduce some hard problem to the MIS and prove its complexity.

We first try with brute force. Assume there are n nodes in the graph. We need to list all the possibilities of choosing nodes in the graph. You can either choose a node or not. There should be $\mathcal{O}(2^n)$ choices. After choosing, we need to test all the edges between them, which takes $\mathcal{O}(n^2)$ time because there are $n^2/2$ pairs to check. The pseudo code is in the first algorithm below. In 2017, it has been proved that there exists an exact algorithm for MIS with time complexity of $\mathcal{O}(1.1996^n)$ using polynomial space, where n is the size of nodes in the graph[2]. Although this method is much better than brute force, it is still unachievable to get an exact answer for the problem. To tackle the MIS problem, we need to go with approximation algorithms.

One good choice is to go with greedy. We choose the node v with smallest degree and add to a set. After that, we delete all the neighbors of v from the graph G. We repeat this until G becomes empty. The pseudo code is in the second algorithm below. To find the node with maximum degree, we need to check all the edges, which takes $\mathcal{O}(n^2)$. Then we need to do this for $\mathcal{O}(n)$ nodes in worst case. Thus the overall time complexity for greedy is $\mathcal{O}(n^3)$. The polynomial time complexity is quite satisfactory for us, but the sad story is that we have no guarantee on the quality of the MIS we get using greedy algorithm. It is proved by Bazgan in 2004 that the MIS problem is Poly-APX-complete and thus no approximation algorithm can achieve a constant factor in polynomial time[1].

Algorithm 1: brute force

Input : An undirected graph $G(V, E)$
Output: MIS of $G(V, E)$

```
n = |V|
MIS = 0
size = 0
for i = 0; i < 2^n; i++ do
    correct = True
    represent i = a_1, a_2, ..., a_n in base 2
    for j = 1; j < n+1; j++ do
        for k = 1; k < n+1; k++ do
            if a_j and a_k and (j, k) ∈ E then
                correct = False
            end if
        end for
    end for
    if correct and sum(a_1, a_2, ..., a_n) > size then
        size = sum(a_1, a_2, ..., a_n)
        MIS = i
    end if
end for
return S
```

Algorithm 2: greedy algorithm

Input : An undirected graph $G(V, E)$
Output: MIS of $G(V, E)$

```
S = ∅ while G is not empty do
    v be node of minimum degree in G
    S = S ∪ v
    remove v and its neighbours from G
end while
return S
```

References.

[1] Cristina Bazgan, Bruno Escoffier, and Vangelis Th Paschos. “Completeness in standard and differential approximation classes: Poly-(D) APX-and (D) PTAS-completeness”. In: *Theoretical Computer Science* 339.2-3 (2005), pp. 272–292 (cit. on p. 1).

[2] Mingyu Xiao and Hiroshi Nagamochi. “Exact algorithms for maximum independent set”. In: *Information and Computation* 255 (2017), pp. 126–146 (cit. on p. 1).