CS 170 Final Project

Qin Bian, Xinyu Feng, Tianyu Xu

May 2 2016

General Idea:

First, we use the graph density formula[1] $D = \frac{|E|}{|V|(|V|-1)}$ to determine if the input graph is dense or not. If it is a sparse graph ($D < 0.5$), we formulate the problem to be a integer linear programming problem. Our objective is to maximize a weighted number of matches subject to the constraint that no cycle in the graph of length more than k is chosen. In order to speed up the program, our algorithm first solves the relaxed problem where the bound on the cycle length is dropped, and then if it finds a solution with cycles longer than permitted it adds the corresponding constraints to these cycles in the next iteration. Specifically, we will solve the following LP problem[2]:

Maximize $\sum_{e \in E} y_e w_e$
such that
for all $v \in V$ $\sum_{e \in \delta_v^{(-)}} y_e = f_e^i$ where $\delta_v^{(-)}$ is the set of all incoming edges to v
for all $v \in V$ $\sum_{e \in \delta_v^{(+)}} y_e = f_e^o$ where $\delta_v^{(+)}$ is the set of all outgoing edges to v
$f_e^o = f_e^i = \{0, 1\}$
$y_e = \{0, 1\}$
$\sum_{e \in C} y_e \leq |C| - 2$ for $c \in C_{over}$ where C_{over} is the set of oversize cycles

The running time is bounded by exponential time as we use simplex in Matlab. If it is a dense graph($D > 0.5$), we will run the following greedy algorithm: we will randomly select a vertex and find a cycle that has ≤ 5 vertices by running DFS. If such a cycle is found, we will remove the cycle and test if the rest of graph is sparse. If it is sparse, we then switch to the linear programming algorithm described above. Otherwise, we will remove this vertex and repeat the same process with another randomly selected vertex. The running time of this greedy algorithm is $O(|V| * (|V| + |E|))$ as DFS requires $O(|V| + |E|)$ time and we will at most run DFS on $O(|V|)$ vertices.

References:

1. https://en.wikipedia.org/wiki/Dense_graph
2. http://web.stanford.edu/iashlagi/papers/pnasChain.pdf