**Reading Assignment 1**

In the paper *Dynamic Local Search for the Maximum Clique Problem* the author describes and analyzes a new algorithm (2006) Dynamic Local Search (DLS). DLS is used to find an optimal solution for the maximum clique problem. The maximum clique problem is finding the largest clique of vertices in a graph. A clique is a group of vertices where every vertex has an edge going to every other vertex in the clique. Solving the maximum clique problem is NP-hard so DLS is an approximation algorithm and therefore does not always give an optimal solution. The performance of DLS is evaluated by comparing it to other state-of-the-art maximum clique approximation algorithms using DIMACS benchmarks. DLS outperforms other approximation algorithms on almost every problem instance and seems to be the new leader in solving the maximum clique problem.

The inputs of DLS are the graph, target clique size, penalty delay, and maximum number of steps. DLS is initialized by choosing a random vertex and creating a one vertex clique. DLS then alternates between two phases: an improvement phase and plateau phase. The improvement phase searches the vertices neighboring the current clique. If a vertex has edges to all the vertices in the current clique it then adds that vertex to the clique. The plateau phase also searches the vertices neighboring the current clique. If a vertex has edges to all the vertices in the clique except one it adds that vertex and removes the vertex which did not have an edge to the new vertex. In both phases the vertex with the smallest weight is chosen first if there are multiple available vertices. A vertex with a weight greater than ten cannot be chosen in the improvement or plateau phases. Vertices that are in DLS’s current clique have their weights increased incrementally. All vertex weights that are not zero are decreased incrementally with the penalty value update cycle. The penalty value update cycle is defined by the penalty delay (e.g. the penalty delay is a number of steps and the vertex weights are decremented every time a multiple of the number of steps is reached). After the penalty value update cycle function is called the current clique is reduced to the last vertex added to it if the penalty delay is greater than one. If the penalty delay is one the clique is reduced to a random vertex from the graph. The DLS algorithm then starts over from the improvement phase. This weight system helps prevent DLS from searching over vertices that it has already searched. If the current clique size in DLS reaches the target clique size after an improvement phase it returns the current clique.

The DIMACS benchmarks are from the Second DIMACS Implementation Challenge and includes 80 different problem instances. DLS was run 100 times for each instance. For three out of the 80 instances DLS was not able to consistently find the target clique size within the allotted number of steps. The other algorithms that DLS was compared to were QUALEX-MS, RLS, DAGS, and k-opt. Each of these algorithms are the dominant algorithm (based on success rate, CPU time, largest clique size, and average clique size) for one to four of the 80 problem instances. DLS is the dominant algorithm for 72 of the 80 problem instances. One of the problem instances called brock800\_1 (which QUALEX-MS is the dominant algorithm) is especially difficult because the average number of edges for each vertex is higher than the average number of edges for vertices in the maximum clique. Using a high penalty delay (e.g. 45) improves DLS’s performance when solving problems like brock800\_1. A lower penalty delay (e.g. 10) is prefered for problems where the average number of edges for all vertices is lower than the average number of edges for vertices in the maximum clique. Overall DLS seems to perform very well as is, but further optimisations of the algorithm should be looked into.