

GCLIQUE: An Open Source Genetic Algorithm for the Maximum Clique Problem

Shalin Shah
sshah100@jhu.edu

Abstract

A clique in a graph is a set of vertices that are all connected to each other. A maximum clique is a clique of maximum size. A graph may have more than one maximum cliques. The problem of finding a maximum clique is a strongly hard NP-hard problem. It is not possible to find an approximation algorithm which finds a maximum clique that is a constant factor of the optimum solution. In this work, we present a genetic algorithm for the maximum clique problem that is able to find optimum or close to optimum solutions to most DIMACS graphs. The genetic algorithm uses new crossover mechanisms which are able to find reasonably good cliques which can then be used in other applications downstream. We also provide C++ code for our algorithm. Results show that our algorithm is able to find maximum cliques for most DIMACS instances, and if not, close to optimum solutions for the other instances.

Keywords: maximum clique, randomized algorithm, genetic algorithm, approximation algorithm, k-opt moves, local search, open source, C++, intersection crossover

1 Methods

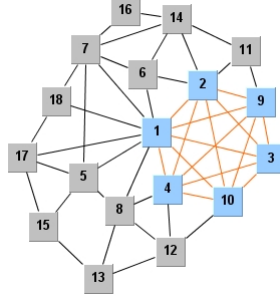
This work implements a genetic algorithm for the maximum clique problem.

This genetic algorithm uses new types of crossover mechanisms to find reasonably good solutions. The intersection crossover that generates a new offspring based on an intersection of the two clique solutions in the two parents is inspired by [1].

The genetic algorithm uses a population of individuals (solutions) and performs crossover and mutation on them. It initializes the population using random solutions. If in consecutive 10 generations, there is no improvement, it does a random restart of the population. In each generation, the algorithm keeps a cache of the global best solution, and propagates the global best individual to the next generation.

For the entire population, the algorithm performs the intersection crossover and mutation. After each step, it performs 2-opt local improvement. Mutation happens only if the fitness (clique size) of the offspring is less than or equal to at least one of the parent solutions.

The maximum clique problem has been well studied in the past. [2] and



[3] are surveys on the problem with detailed descriptions on various algorithms like approximations. This work [4] is a fast algorithm for the maximum clique problem. [5] is another survey on maximum clique algorithms.

The algorithm could be used to feed the clique found to other algorithms like the one mentioned in [6].

2 Results

In most of the DIMACS instances, the algorithm is able to find the optimum or close to optimum clique (except the san1000 instance). The results are shown in the following table. The git repository can be found in [7] at the following URL: <https://github.com/shah314/clique>

The DIMACS problem instances are available at [8] and also linked from the git page. If a faster and simple solution in Java is desired, the work in [9] could be useful.

3 Conclusion

In this article, we presented an open source implementation of a genetic algorithm for the maximum clique problem in C++. The algorithm is quite fast, and the speed can be tuned by setting the appropriate number of generations. Our intersection crossover takes the common nodes in the two parent cliques and then completes it. This is inspired by [1] and is a very effective way of finding cliques reasonably close to the optimum. Further types of crossover mechanisms can be explored.

Table 1: GCLIQUE Results

DataSet	Vertices	Edges	Best Known	This Algorithm	Duration (Sec-onds)
brock200_1	200	14834	21	21	9
C125.9	125	6963	34	34	0
c-fat200-1	200	1534	12	12	0
DSJC500.5	500	125248	13	13	3
gen200_p0.9_44	200	17910	44	40	5
hamming6-2	64	1824	32	32	0
johnson32-2-4	496	107880	16	16	0
keller4	171	9435	11	11	0
MANN_a27	378	70551	126	125	1
p_hat300-1	300	70551	8	8	0
san200_0.7_1	200	13930	30	30	0
san1000	1000	250500	15	10	1

References

- [1] Ashay Dharwadker. *The clique algorithm*. Institute of Mathematics, 2006.
- [2] Panos M Pardalos and Jue Xue. The maximum clique problem. *Journal of global Optimization*, 4(3):301–328, 1994.
- [3] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.
- [4] Patric R.J. Ostergard. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1):197 – 207, 2002. Special Issue devoted to the 6th Twente Workshop on Graphs and Combinatorial Optimization.
- [5] Qinghua Wu and Jin-Kao Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693 – 709, 2015.
- [6] Shalin Shah. Jcol: A java package for solving the graph coloring problem. *Journal of Open Source Software*, 5(48):1843, 2020.
- [7] Shalin Shah. <https://github.com/shah314/clique>, 2020.
- [8] http://iridia.ulb.ac.be/~fmascia/maximum_clique/dimacs-benchmark.
- [9] Shalin Shah. Randomized heuristic for the maximum clique problem, 2016.