COMPLEXITY ANALYSIS OF CLIQUE PROBLEM

Priyanka Saxena

Department Of Computer Science (AIM & ACT)
Banasthali Vidhyapith
Rajasthan, India
s.priyankasaxena309@gmail.com

Ms. Dipanwita Thakur

Department of Computer Science(AIM& ACT)
Banasthali Vidhyapith
Rajasthan, India
Dipanwita.thakur@gmail.com

ABSTRACT

MAXIMUM CLIQUE PROBLEM the most relevant problem in Graph theory, known for years still do not have its polynomial time solution. Many algorithms have been proposed, still the problem lie the same i.e. to find the Clique in the polynomial time.

The Clique problem is to figure out the sub graph with the maximum cardinality. Maximum clique in a random graph is NP-Hard problem, actuated by many prominent problems such as mobile networks, computer vision, & social network analysis.

The paper is focused to serve an absolute review on Maximum Clique Problem. This paper encompass the survey done on the most studied algorithms of the Clique Problem with their results showing the best, average, and worst timing algorithms of Clique Problem. Though capturing the complete research in this regard is over the latitude of this paper, still it is tried to capture most of the ideal paper from similar approaches.

KEYWORDS: Maximum Clique problem; sub graph; maximum cardinality; conjointly adjacent; social network analysis;

INTRODUCTION

The objective of this work is to survey and compare all the exact, approximation and heuristic clique problems defined till now. We have compared the time complexities of all the algorithms

CLIQUEPROBLEM: Assuming Z=(X, Y) an approximate undirected simple graph where X consists of the set of n vertices & $Y \subseteq X*X$ is the set of edges in Z. An edge {a, b} is in Y only if $\{a, b\} \subseteq X$ and vertex is adjacent to vertex b. A clique can be defined as the set of vertices X such that each pair of vertices in the clique L is adjacent in the graph Z. Finding the clique of a fixed size K is a well- known NP-Complete Problem known as K-Clique. Clique is a basic NP-Complete problem. It is postured as the independent problem: Given a random graph Z & a positive integer $K \leq |X|$, Z will have the clique of size K or more?

Then the problem is to know the maximum clique of size Ꞷ (Z).

The problems in clique problem are:

1. Graph Z, and an integer S is taken, does Z have a clique consisting of S vertices?
2. Given a graph Z find all the cliques with the maximum cardinality.
3. Listing all the maximal cliques.
4. The decision problem of analyzing does the graph Z contains a clique larger than a given size.

The clique problem can be solved in polynomial time if there is a finite set of given vertices or when there are finite number of cliques present in the graph Z, but when the number of cliques starts to grow exponentially we cannot calculate the solution in polynomial time the problem then becomes the NP-Hard problem given that it includes the version of NP-Complete.

The clique problem resides with two well-studied problems:

Maximum Clique Problem: The maximum clique is the maximal clique with the maximum cardinality or weight.

Maximal Clique Problem: The maximal clique is the clique that is not a subset of any other clique (independent set).

RELATEDWORK

Addressing the decision and optimization algorithm with an Exact Algorithm for example the Backtracking search algorithm. The back tracking algorithm works in the following manner, it additionally formulate a set C (originally vacant) by electing a candidate vertex from the candidate set D (originally full) and adding them to C. After electing a vertex, from the set D the candidate set is renewed and extract the vertices that cannot take part in the emerging Clique.

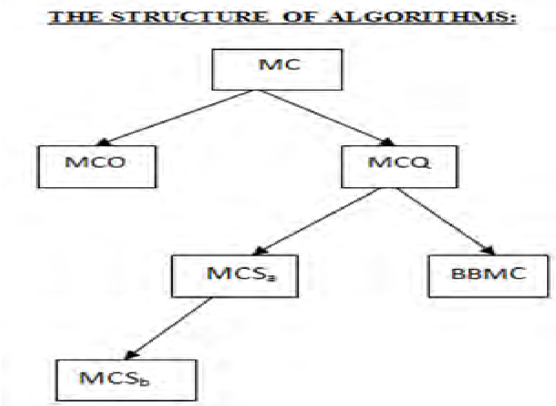
Assuming that the candidate set D becomes empty i.e. the set C becomes maximal (if it becomes maximal) then keeps it and then backtracks. If D is not vacant, search is run on it, i.e. nominating from D and adding it to C.

There are various scenarios were we do not require the search such as:

1. If P contains the insufficient set of vertices to dismount the champion list, in such a case the search may be dropped.

2. If we are using the Graph Coloring, an upper bound at the time of search can be calculated by using graph coloring, i.e. if the whole candidate set D is colored with n colors then it will have the clique no more than n vertices. [1]

There are various other parameters that may be considered into account while electing the candidate vertex, contrasting approaches of search, various algorithms to color the graph and particular orders in which it can be done. [1]



THE ALGORITHMS:

Classic Bron-Kerbosch algorithm: Bron and Kerbosch designed a backtracking method that requires only polynomial storage space and prohibits all the possibility of computing the same clique over again .Time complexity of this algorithm is $O(3^{n/3})$.It particularizes all the maximal cliques. [5]

Modified approach of Bron-Kerbosch algorithm: David Eppstain presented a modified version of Bron-Kerbosch that visits the graph using a degeneracy ordering, having the complexity O $(d*n*3^{d/3})$. Given d, degeneracy of the graph, an ordering in which each vertex has d or less neighbors which can be found out in the linear time. [6]

Disadvantage: The overhead discussed algorithms were designed for discovering all the maximum cliques in a graph, but solving the maximum clique problem requires discovering only one maximum clique. To solve this problem below discussed algorithms were proposed.

Tarjan and Trojanowski: Presented a recursive algorithm for maximum independent set problem, having time complexity O $(2^{n/3})$.The problem finding the Maximum Independent Set in the graph is equivalent to the Maximum Clique Problem. [7]

The algorithm uses recursive, backtracking approach and concept of connected component and dominance. O $(2^{n/3})$ is much faster than enumeration of all independent set.

[8]Are defined result with score complexity of O $(2^{0.288n})$ was achieved. Further Robson [9] refined the best known worst- case complexity to O $(2^{n/4})$.

Branch and Bound: Developed by Carraghan and Pardalos [10] it is represented as EA i.e. enumerative algorithm is the basic branch & bound algorithm. Despite its simplicity, this algorithm is established as an extensive step for the exact solving of MCP and act as the basis for many later improved exact clique algorithms

VColor-BT-u: Deniss Kumlander presented a better heuristic based vertex coloring and backtracking for MCP. Used color classes and backtracking methodology.[11]

It outperforms EA and REA (Reverse enumerative Algorithm, proposed by Osterg'ard). [12]

The algorithm first finds the maximum clique that is present in each color class and then assigns it to the special *array b*. Now as a result b[i]is the maximum clique for the sub graph formed by {L_i, L_{i-1}…L_l} vertices while searching backwards [3].

MC: MC [21; 1] It is refined and simple algorithm but too simple to posses any practical importance.

Fahle's Algorithm 1is implemented by using two sets: L amplifying clique (originally vacant) and D (candidate set) (originally having each vertex of the graph Z).L is maximal when D is empty & if |L| is *maxima* then. L becomes the champion.

If |D|+|L| are too limited to detrain the champion search, then the search is aborted. Else the search continues over the vertices in D, choose a vertex x and intensify the clique $L^1$where L^1=L∪ {x} and a new candidate set D^1 as the set of vertices in D that are adjacent to the vertex x and keeps on counting in L^1.It is called MC.

DRAWBACK: MC generates recursive calls that fail immediately. Drawback overcame in MCQ.

MCQ: MCQ [14; 1] is revised version ofMC.MCQ implements the binomial search with the following advances:

1. Due to Welsh and Powell [4] the greedy sequential coloring algorithm is used to color the graph which is lead by the candidate set. Due to which an upper bound can be given on the size of the clique L and it is added to D.
2. Sorting of the candidate set D occurs in the non decreasing color order.
3. An array X of class vertex can be sorted in O (n (log (n))) time by using Java's Arrays. Sort(X) method & then it is ordered as default by CompareToMethod in the class vertex X.

   MCS: Are fined version of greedy coloring based algorithm is proposed as MCS [13].A recoloring strategy using greedy approximate coloring procedure and outperform its predecessors [14; 15].

The gist of pruning in MCS is that if a vertex x∈Y, having color number K x< ($|L^*|$-|L|), then the vertex x needs not to be searched from. So the vertices to be searched in the candidate set D are reduced.

DLS: Pullan Wand Hoos H**,** Dynamic Local Search [16] is actually as light simplification of deep adaptive greedy search [17].The family of stochastic local search algorithms i.e., Dynamic local search (DLS)[16],phased local search (PLS) [18]and cooperating local search(CLS)[19],share analogous strategies of clique expansion, plateau search and search stagnation.

The algorithm is accomplished through efficient supporting data structures leads to smaller overall CPU times.

PLS: Pullan W, Phased Based Local Search [18], to endure with graphs of different structures, bonds three sub-algorithms i.e. random selection, random selection within among those with highest vertex degree, and random selection within those with lowest vertex penalty, all three approaches uses different vertex selection rules.

The performance results of PLS compared to DLS-MC shows that PLS is more efficient than DLS-MC for all DIMACS instances.

CLS: W. Pullan, F.Mascia, M. Brunato, Cooperating Local Search (CLS) [19], further advance over PLS by covering four low level heuristics which are effective for different instance types, namely: Greedy Search (GREEDY), Level Search (LEVEL), Focus Search (FOCUS), Penalty Search (PENALTY).

CLS shows excellent performances on both DIMACS and BOSHLIB benchmarks.

BBMC: San Segundo's Bit Board Max Clique algorithm [20]. BBMC takes a color-class perspective. BB Color begins building the first color class and then finds the next color class and soon until D (Candidate Set) is exhausted. Color Classes are combined using the pigeon whole sort, which result in a list of non decreasing color order and this is further used in BB Max-Clique method to cut off search as in MCQ and MCS_a.

Now if the currently formed clique depicts both maximal and maximum characteristics then it is saved by BBMC through specialized save method else, if C does not depict the maximal behavior then a recursive call is made to BBMaxClique.

ELS: S.Balaji [2] The Edge Based Local Search is a two phased local search method for MCP.

The support of a vertex for each x ∈ X in the graph Z, can be defined by

$S(x) = d(x) + \Sigma_{y \in N(x)} d(y)$

The quantity $\Sigma_{y \in N(x)} d(y)$ is called as the sum of degree of vertices that are adjacent to x.

ELS is efficient for achieving the state of art performance which can be proved by the computational results on BHOSLIB and DIMACS benchmark graphs in average running time of maximum clique problem.

SEARCH STRATEGY AND CRITERIA

We have studied all the algorithms of various types such as exact algorithms, Approximation algorithm and heuristic algorithm and had compared their time complexities to find out the best result. We have tried to compare the different algorithms and had tried to find out which algorithm works the best for which application may be that algorithm gives the worst time complexity in the other application. We had tried to find the disadvantages and drawbacks of the proposed algorithms the result is provided in the comparison table, specifying the best algorithm time complexity till date.

RESULT

YEAR	NAME/Author	CODE	DETAILS	COMPLEXITY	REMARKS
1973	Bron, Coen; Kerbosch, Joep/**Classic Bron-Kerbosch algorithm**	Yes	Back tracking method , excludes all The possibility of computing the Same clique twice requires only polynomial storage space. ,	Not mentioned By the author	O (3n/3), enumerates all maximal Cliques.
1977	R.E. Tarjan and A.E. Trojanowski/**MIS/MCP**		The algorithm uses a recursive, or Backtracking scheme and concept of connected component and dominance	$O(2^{n/3})$	It is much faster than Enumeration of all independent set.
1990	R.Carraghan &P.M. Pardalos/ **Branch &Bound algorithm** also denoted as **EA (Enumerative algorithm)**	Yes	In EA, at the depth in the binomial search the vertices are arranged in non-decreasing degree order with cut-off based on the size of the maximum clique known so far. The basic B&B approach	Not mentioned by the author	Branch & Bound acts as the basis for the other Branch & Bound algorithms It is used for the generation of random Graphs. So the experimental results for EA are fully reproducible.
1992	Pardalos &Rodgers/ **0-1encoding**		If the search decides that the vertex x is present in the clique then $X_x$is assigned value1,& if rejected then 0 RULE:¬adjacent(y, X)→x_y+x_x	Not mentioned By the author	
1997	**Graph coloring & Fractional coloring**		Greedy approach is applied to color the candidate set. If, Size of the current clique + number of colors used ≤ the size of the largest clique then that branch of search is cutoff	Not mentioned By the author	Vertices are selected in the opposite manner as proposed in branch & bound i.e. in non-increasing degree order. If the selection of vertices is allowed to be chosen in free manner then it will give the similar effect as that of MC.
2001	Battiti& Protasi/**RLS**		Reactive Tabu local search, it is past sensitive Scheme to find the amount of diversification	Not mentioned By the author	MCP algorithm, shows better results as compared its antecedents

2002	Patric R.J. ¨Ostergard/ **REA(Cliquer)**		Based on B & B, Uses reverse Ordering of vertices as in EA Search process: Find clique which is largest, having vertices extracted from the set Sn= {xn} register its size in L[n]. Search proceeds to find the clique which is largest in the set Si= {xi, xi+1...xn} using the value in L [i+1] as a bound.	Not mentioned By the author	Has a dynamic Programming flavor. Vertices colored as: 1. Greedily 2. Non- decreasing Color order. EA performs better than REA.
2002	Torsten Fahle/ **Algorithm1(MC)**		Inherently MC with the advancement of free selection of vertices instead of dynamic maintenance of vertex degree in the candidate set and fixed iteration.	Not mentioned by the author	DF is then advanced to associate a coloring Bound, parallel to that in Wood.
2002	**Enhanced algorithm1 i.e.algorithm2 named Domain Filtering**		Forced accept &reject steps	Not mentioned by the author	
2003	Jean-Charles R´egin/ **CPR**		Model uses a matching in a duplicated graph to deliver a bound within search ,a Not Set as used in the Bron Kerbosch enumeration Algorithm& vertex selection using the pivoting strategy	Not mentioned by the author	Constraint programming, Using B & B and filtering of vertices to tighten the candidate set D
2003	Tomita/**MCQ**	Yes	MCQ is an extended version of MC, Implements the binomial search, with advancements i.e.: Due to Welsh and Powell [4] the greedy sequential coloring algorithm is used to color the graph which is lead by the candidate set. Due to which an upper bound can be given on the size of the clique L and it is added to D. Sorting of the candidate set D occurs in the non-decreasing color order.	O(n.log(n))	MCQ uses simple degree ordering.

2004	Katayama, Hamamoto,& Narihisa,/**KLS**		Iterated local search	Not mentioned By the author	Achieves: Good performance: MANN instances from DIMACS. Bad performance: Keller and Brock instance
2004	Grosso, A., Locatelli, M., & Croce, F.D./ **DAGS**		Greedy algorithm including plateau search	Not mentioned by the author	1. The best performing algorithm. 2. Achieves high competitive results as compared with state-of- the art Algorithms former to DLS
2006	Deniss Kumlander/**VColor-BT-u**		Based on Color Classes and Backtracking	Not mentioned By the author	Outperform EA and REA
2006	S. Busygin /**QUALEX-MS**		Greedy algorithm, nonlinear programming formulation is used to derive the vertex weights.	Not mentioned by the author	Better Performance on Brock instances of DIMACS

2006	Pullan W/**PLS**		Phased based local search ,a robust form of DLS	Not mentioned by the author	Perform better then DLS on DIMACS benchmarks (exceptKeller6)
2007	Tomita/ **MCR**	Yes	Replica of MCQ with advancement in: Initial ordering Essentially non-decreasing degree with Tie-breaking on the sum of the degrees of adjacent vertices. This ordering is then modified during search via the coloring routine number Sort as in MCQ	Not mentioned By the author	MCR is compared to MCQ in E. Tomita &Toshikatsu Kameda B&B over 8 Ofthe66instances of the DIMACS benchmarks showing an improvement in MCR over MCQ. MCR is our MCQ With style= 3
2007	Janez Konc and Duˇsanka Janeˇziˇc / **MaxCliqueDyn algorithm**		Dynamic reordering of vertices in Candidate Set is done using current degree i.e. which is prior to the coloring	Not mentioned by the author	Reordering takes place high-up in the backtrack tree & is expensive. Is controlled by parameter L_{emit}

2010	Pablo San Segundo and Crist´obal Tapia/**early version of BBMC(BB-MCP)**	Yes	It is the bit-set encoding of MCSa with following features: 1. Bit strings are used to represent sets. 2. The candidate set is colored through BBMC colors by using static sequential ordering; the ordering in BBMC is same as that of MCSa. 3. BBMC depicts the neighborhood of vertex and the inverse neighborhood is depicted as bit strings. 4. A color class broad view is taken while coloring.	Not mentioned By the author	
2010	Tomita/**MCS**	Yes	B & B based on sub graphs coloring. MCS is MCR with two modifications: 1.MCSuses"an adjacent ordered set of	Not mentioned by the author	MCS is much better than MCR. An improved version of sub graph
			Vertices for approximate coloring" 2. When coloring vertices Re-NUMBER) Extension of MCQ is used (Called as repair mechanism). Uses static coloring order. Uses Static coloring Order plus color repair mechanism		coloring algorithm, Performs better than its all predecessors on many DIMACS Instances.
	MCS is divided into two: 1. **MCS_a** 2. **MCS_b**	Yes Yes	The partition based upper bound can be improved by using MaxSAT. The graph structure of propositional and exploitation is combined to solve the MaxClique.	Not mentioned by the author Not mentioned by the author	
2010	Li& Quan/**Max SAT encodings for maximum clique**	Yes		Not mentioned by the author	Hard random graphs and DIMACS MaxClique benchmarks show positive results when applied.

2011	Pablo San Segundo/ **BBMC**	Yes	A version of BBMC with the color repair Steps from Tomita's MCS.	Not mentioned By the author	BB-ReCol1 can Result in the candidate set becoming a multistep resulting in redundant re-exploration of search space. Exploration of search space with subsequent poor performance.
2011	W. Pullan, F.Mascia, M. Brunato/**CLS**		Cooperating local search, advance than PLS and paralleled algorithm	Not mentioned by the author	DIMACS &BOSHLIB benchmarks provide the best results when CLS is applied. The best performing MCP Algorithm.

2012	Renato Carmo & Alexander P. Z¨uge		Presented an experimental study over 8 algorithms including MCQ,MCR,MCS,MCS_a,,MCS_b, MaxCliqueDyn	Not mentioned By the author	Declared Bron Kerbosch Algorithm provides a united framework for all algorithms. Their algorithm can be viewed as iterative version of MaxClique.
2013	S.Balaji/**ELS**	Yes	Edge based local search for vertex Cover and equivalently for MCP	Not mentioned By the author	Performance on Both DIMACS and BOSHLIB benchmarks is equivalent or better than PLS
2014	KK Singh, K K Naveena, G Likhitaa/ **TALS**		Target aware local search, Incorporating Prohibition time for diversification	Not mentioned By the author	Outperform DLS Excluding some instances of P-hat, gen-400,keller-6of DIMACS benchmarks

ACKNOWLEDGMENTS

I consider it my privilege to express gratitude and sincere thanks to my guide, Mrs. Dipanwita Thakur for taking me under her supervision, encouraging me to enter into the domain of Theory Of Computation for giving me this wonderful opportunity to conduct my study in this area with her constant support, help and interaction.

I am also thankful to my university i.e. Banasthali University for allowing me to do my research and providing with all the equipments and workspace and funding my project.

CONCLUSION

There is no deterministic polynomial time algorithm that can find clique of size $n^{1-\epsilon}$for any $\epsilon>0$, until NP=ZPP. The best leading known polynomial time approximation algorithm for the clique problem has O (n/ (logn) $^{2)}$ time complexity. Cellular Learning Automata until now has not been considered for solving the clique problem, though if we see the history Cellular learning automata meets the requirement of applications such as rumor diffusion, in data classification, image processing, social network analysis, clustering the wireless sensor networks , VLSI Placement and many more. Moreover Cellular Learning Automata is studied to be highly efficient and performs very well in the applications based on the clique problem. So later we are going to propose an algorithm based on cellular learning Automata, who knows, it may open a whole new world to their searches and the million dollar problem of P=NP or not may result in something new.

REFRENCES

[1] Prosser, P. Exact algorithms for maximum clique: A computational study. Algorithms 2012, 5, 545–587

[2] S. Balaji, "A New Effective Local Search Heuristic for the Maximum clique problem ", World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical and Quantum Engineering Vol: 7 No: 5, 2013

[3] Singh, Krishna Kumar, and Ajeet Kumar Pandey. Survey of Algorithms on Maximum Clique Problem.

[4] D.J.A. Welsh and M.B. Powell, "An upper bound to the chromatic number of a graph and its application to time-tabling problems", The Computer Journal, Vol 10 (1967), p. 85.

[5] Coen Bron, Joep Kerbosch, Algorithm 457: finding all cliques of an undirected graph, Communications of the ACM, v.16 n.9, p.575-577, Sept. 1973 [doi>10.1145/362342.362367

[6] D. Eppstein, M. Löffler, and D. Strash. Listing all maximal cliques in sparse graphs in near-optimal time. In ISAAC (1), pages 403--414, 2010.

[7] R.E. Tarjan and A.E. Trojanowski, Finding a maximum independent set, *SIAM J. Compute*., *Vol. 6*: 537–546, 1977.

[8] Fedor V. Fomin , Fabrizio Grandoni , Dieter Kratsch, Measure and conquer: a simple $O(2^{0.288n})$ independent set algorithm, Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, p.18-25, January 22-26, 2006, Miami, Florida [doi>10.1145/1109557.1109560].

[9] J. M. Robson. Finding a maximum independent set in time O (2n/4). Technical Report 1251--01, LaBRI, Université Bordeaux I, 2001.

[10] Randy Carraghan and Panos M. Pardalos. An exact algorithm for the maximum clique problem. Operations Research Letters, 9:375–382, 1990.

[11] Singh, Krishna Kumar,Lakkaraju Govinda. A simple and efficient heuristic algorithm for maximum clique problem.IntelligentSystemsandControl (ISCO), 2014IEEE8th InternationalConferenceon. IEEE,2014.

[12] Östergård, P.R.J. 2002. A fast algorithm for the maximum clique problem. Discrete Applied Mathematics, 120:195–205

[13] Tomita, E., Sutani, Y., Higashi, T., Takahashi, S., Wakatsuki, M.: A simple and faster branch-and-bound algorithm for finding a maximum clique. In: Rahman MS, Fujita S. (eds.) Proceedings of the 4th International Workshop on Algorithms and Computation. Lecture Notes in Computer Science, vol. 5942, pp. 191–203. Springer, Berlin (2010)

[14] Tomita, E., Seki, T.: An efficient branch and bound algorithm for finding a maximum clique. In: Calude, C., Dinneen, M., Vajnovszki, V. (eds) Discrete Mathematics and Theoretical Computer Science. LNCS, vol. 2731, pp. 278–289, Springer, Berlin (2003)

[15] Tomita, E., Kameda, T.: An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. J. Global Optim. 37, 95–111 (2009); J. Global Optim. 37, 95–311 (2007)

[16] Pullan, W.J., Hoos, H.H.: Dynamic local search for the maximum clique problem. J. Artif. Intell. Res. **25**, 159–185 (2006)

[17] Grosso, A., M. Locatelli, and F. Delia Croce. (2004). "Combining Swaps and Node Weights in an Adaptive Greedy Approach for the Maximum Clique Problem." *Journal of Heuristics* 10(2), 135–152

[18] W. Pullan, Phased local search for the maximum clique problem, J. Comb. Optim. 12 (3) (2006) 303–323

[19] W.Pullan,F.Mascia,M.Brunato,"Cooperatinglocalsearchforthemaximumcliqueproblem".JournalofHeuristics,17(2):181–199,2011

[20] San Segundo, P., Rodriguez-Losada, D., Jimenez, A.: An exact bit-parallel algorithm for the maximum clique problem. Comput. Oper. Resour.**38**(2), 571–581 (2011)

[21] T. Fahle, Simple and fast: Improving a branch-and-bound algorithm for maximum clique, in: Proc. of ESA, 2002, pp. 485–498