

Heuristic Algorithm for Finding the Maximum Independent Set with Absolute Estimate of the Accuracy

Damir N. Gainanov¹, Nenad Mladenović², Varvara Rasskazova³, and Dragan Urosević⁴

¹ Ural Federal University, Ekaterinburg, Russia
damir.gainanov@gmail.com,

² Serbian Academy of Sciences and Arts, Belgrade, Serbia
nenadmladenovic12@gmail.com,

³ Moscow Aviation Institute, Moscow, Russia
varvara.rasskazova@mail.ru,

⁴ Serbian Academy of Sciences and Arts, Belgrade, Serbia
draganu@turing.mi.sanu.ac.rs

Abstract. The paper presents an algorithm for finding the maximum independent set in an undirected graph with absolute estimate of the accuracy. Special notions are introduced and theoretical results in the area of deviation of approximate solution from the exact one are presented. Also the paper presents results of computational experiments on the complementary graphs DIMACS.

Keywords: Algorithm · Maximum independent set · Absolute estimate of the accuracy

1 Introduction

The problem of the maximum independent set of vertices in an undirected graph (MIS Problem) has applications in many areas. So, for example, the vertices of the social graph can be interpreted as users accounts. In this case, the independent set will correspond to different accounts, and the clique in the complementary graph will correspond to fake ones. Another example is the location of enterprises. In this context, the independent set will correspond to the optimal location. Thus, the relevance of the research does not cause doubts in terms of practical significance.

MIS Problem is a classical \mathcal{NP} -hard. In complexity theory 3-satisfiability problem (3-SAT) reduces to it, which in turn reduces to the standard hard satisfiability problem (SAT). This effect indicates that the problem under consideration have special scientific interest. As it is known, for today the problem of

equality of classes is not solved and there is no polynomial algorithm for any hard problem. In this connection, approximate and heuristic algorithms are developing actively. The disadvantage of approximate algorithms is that estimates of the accuracy obtained are usually greatly overstated. So, for example, in [1] there was developed 2-approximate algorithm for vertex cover problem which guarantees the deviation of not more than 2 times. However for problems with known exact solution this estimate is measured by units of vertices. Another approximate algorithm from [2] implements a greedy approach for the same problem and allows to obtain the estimate $\mathcal{O}(\log n)$, which often leads to significant quality losses. Heuristic algorithms are very effective but any estimates are absent for them. For example, in [3]–[5] algorithms for maximum clique problem (MCP) were developed. Note that MCP mentioned above is the dual \mathcal{NP} -hard for MIS Problem.

In [6] an inference algorithm for monotone Boolean functions with absolute estimate of the accuracy was developed. The additive criterion has the advantage for solving large-dimensional problems. If the solution include many non-zero components then the deviation measured by units of components does not lead to significant losses for practical purposes. In this paper, this idea was continued to solve MIS Problem.

2 The Maximum Independent Set in the Graph

Let us consider an undirected graph $G = (V, \mathcal{E})$, in which

$$V = \{v_i, i = [n]\}$$

is the set of vertices, and

$$\mathcal{E} = \{e = (v_i, v_j), i, j \in [n]\}$$

is the set of edges.

Definition 1. For an integer $k \in [n - 1]$ we call a vertex $v \in V$ of the graph $G = (V, \mathcal{E})$ a k -vertex if $|\mathcal{N}(v)| = k$ and the neighborhood of this vertex is the complete induced subgraph.

Definition 2. For integers $k \in [n - 1]$ and m we call a vertex $v \in V$ of the graph $G = (V, \mathcal{E})$ a (k, m) -vertex if $k = |\mathcal{N}(v)|$ and in the neighborhood of this vertex there are absent m edges to be the complete induced subgraph.

Proposition 1. If the vertex v is a k -vertex in the undirected graph $G = (V, \mathcal{E})$ then there exists the maximum independent set of vertices $S \subseteq V$ such that $v \in S$.

Proof. Let us consider the arbitrary maximum independent set S of the graph and the k -vertex $v \in V$. It is easy to see that among elements of the set $\{v\} \cup \mathcal{N}(v)$

there is vertex u such that $u \in S$. Indeed otherwise one could find an independent set $S' \subseteq V$ such that $S' = \{v\} \cup S$ and $|S'| > |S|$. This contradicts the maximality of the independent set S .

Now let us consider two possible cases.

- 1) If $v \in S$ then we are done.
- 2) If $v \notin S$ and $u \in S$ for some vertex $u \in \mathcal{N}(v)$ then for S one can find the set

$$S' = S - \{u\} \cup \{v\}$$

which is also the independent set by definition of k -vertex. By construction we have $|S'| = |S|$. Thus we obtained the maximum independent set S' such that $v \in S'$ as was to be proved.

Note that Proposition 1 underlies the idea of developing the heuristic algorithm with the absolute estimate of the accuracy.

3 The Algorithm for MIS Problem with Absolute Estimate of the Accuracy

We denote the set of all independent sets of the graph G as $\mathcal{S}(G)$ and the set of all maximum independent sets of the graph G as $\mathcal{S}_{\max}(G)$. Let $\max S(G) = |S|: S \in \mathcal{S}_{\max}(G)$ be the cardinality of the maximum independent set of the graph G .

Proposition 2. *Let two graphs $G_1 = (V_1, \mathcal{E}_1)$ and $G_2 = (V_2, \mathcal{E}_2)$ are such that $V_1 = V_2$, and $\mathcal{E}_1 \subseteq \mathcal{E}_2$. Then*

$$\mathcal{S}_{\max}(G_2) \subseteq \mathcal{S}(G_1) .$$

Proof. Consider an arbitrary independent set $S \in \mathcal{S}(G_2)$ of the graph G_2 . By the definition of the independent set we have:

$$\forall u, v \in V_2: u, v \in S \quad (u, v) \notin \mathcal{E}_2 .$$

By the hypothesis of the proposition, we have $\mathcal{E}_1 \subseteq \mathcal{E}_2$ and $V_1 = V_2$. As a consequence for any $S \in \mathcal{S}(G_2)$ we have

$$\forall u, v \in V_2: u, v \in S \quad (u, v) \notin \mathcal{E}_1 .$$

and thus

$$S \in \mathcal{S}(G_1) . \tag{1}$$

Then for any set of vertices of the graph G such that $S \in \mathcal{S}(G_2)$, the inclusion (1) holds, that is,

$$\mathcal{S}(G_2) \subseteq \mathcal{S}(G_1) .$$

At the same time by definition of the maximum independent set of the graph, we have that any maximum independent set of the graph is clearly the independent set of the graph, so

$$\mathcal{S}_{\max}(G_2) \subseteq \mathcal{S}(G_2) \subseteq \mathcal{S}(G_1) ,$$

as was to be proved.

Corollary 1. *Let $G_1 = (V, \mathcal{E}_1)$ and $G_2 = (V, \mathcal{E}_2)$ be graphs such that $\mathcal{E}_1 \subseteq \mathcal{E}_2$. Then*

$$\max S(G_1) \geq \max S(G_2) .$$

Proof. Let us consider the maximum independent set $S \in \mathcal{S}_{\max}(G_2)$. According to Proposition 2 we have $S \in \mathcal{S}(G_1)$. By definition for any $S \in \mathcal{S}(G_1)$ there exists $S' \in \mathcal{S}_{\max}(G_1)$ such that $|S'| \geq |S|$. Then

$$\max S(G_1) = |S'| \geq |S| = \max S(G_2) ,$$

as was to be proved.

The following proposition underlies the absolute estimate of the accuracy of the approximate solution.

Proposition 3. *Let $G = (V, \mathcal{E})$ be a graph in which vertices v and u are not adjacent. And let $G' = (V', \mathcal{E}')$ be a graph in which $V' = V$ and $\mathcal{E}' = \mathcal{E} \cup \{(v, u)\}$. Then*

$$\max S(G) \geq \max S(G') \geq \max S(G) - 1 . \quad (2)$$

Proof. The inequality $\max S(G) \geq \max S(G')$ follows from Corollary 1 since $\mathcal{E} \subseteq \mathcal{E}'$. Let us prove the inequality $\max S(G') \geq \max S(G) - 1$. Let S be the maximum independent set of the graph G .

1) Suppose that $v \notin S$ and $u \notin S$. Then $S \in \mathcal{S}_{\max}(G')$. Indeed otherwise we would obtain that there exists the maximum independent set $S' \in \mathcal{S}_{\max}(G')$ such that $|S'| > |S|$. According to Proposition 1, we obtain that $S' \in \mathcal{S}(G)$, but this contradicts the maximality of the set S under consideration. Thus, in this case, we have:

$$\max S(G) = |S| = \max S(G') \geq \max S(G) - 1 .$$

2) Suppose that $v \in S$ and $u \notin S$. If the edge (v, u) is added, then $S \in \mathcal{S}_{\max}(G')$ as was shown above.

3) Suppose that $v \in S$ and $u \in S$. If the edge (v, u) is added, then we obtain $S \notin \mathcal{S}_{\max}(G')$. In this case we can find a set S' such that $S' = S - \{v\}$. Then $S' \in \mathcal{S}_{\max}(G')$ and

$$|S'| = |S| - 1 .$$

by construction. By the definition of the maximum independent set of the graph, we have:

$$\max S(G') \geq |S'| = |S| - 1 = \max S(G) - 1 ,$$

as was to be proved.

Corollary 2. *Let $\{e_1, \dots, e_k\}$ be a subfamily of k vertex pairs that are not edges of the graph $G = (V, \mathcal{E})$. And let the graph $G' = (V', \mathcal{E}')$ is such that $V' = V$ and $\mathcal{E}' = \mathcal{E} \cup \{e_1, \dots, e_k\}$. Then*

$$\max S(G) \geq \max S(G') \geq \max S(G) - k .$$

Proof. For proof it suffices to apply Proposition 3 k times.

Proposition 4. *If the vertex v is a (k, m) -vertex in the graph $G = (V, \mathcal{E})$ then there exists a maximal independent set $S \subseteq V$ such that $v \in S$ and*

$$|S| \geq \max S(G) - m .$$

Proof. According to definition there are missed m edges in the neighborhood of the vertex v to be the complete induced subgraph. Suppose these edges are $\{e_1, \dots, e_m\}$. Then the vertex v is a k -vertex in the graph G' , which is obtained from the graph G by the addition of these m edges $\{e_1, \dots, e_m\}$. According to Proposition 1 there exists S' : $S' \in \mathcal{S}_{\max}(G')$ such that $v \in S'$.

According to Corollary 2 we have:

$$|S'| = \max S(G') \geq \max S(G) - m .$$

It follows from Proposition 2 that $S' \in \mathcal{S}(G)$. By definition there exists a maximal independent set S such that $|S| \geq |S'|$ and, as a consequence

$$|S| \geq |S'| \geq \max S(G) - m ,$$

as was to be proved.

Based on Proposition 4 one can propose an efficient recursive algorithm for solving MIS Problem. Let us define some features of this algorithm.

On the input of the algorithm $\mathcal{B}(G)$ the graph $G = (V, \mathcal{E})$ is given. For example, it could be given by the adjacency matrix. It is clear that each iteration of the algorithm leads to the update of the adjacency matrix, when a certain vertex is included in S and removed from V_0 together with neighborhood.

In arrays $k[v], m[v]$ (of length n) the values of parameters k and m are stored for corresponding vertices. These values are also updated at each iteration of the algorithm.

The function $MinMaxParam(v)$ returns consequentially the vertex v with the minimum value of the parameter m and with the maximum value of the parameter k . In other words, at first vertices are ordered by increasing m . Then among selected vertices with the minimum m the vertex with the maximum k is selected. For example, if among vertices of the initial graph there is a unique vertex with the value 0 of the parameter m , then the function $MinMaxParam(v)$ will return this vertex, regardless of the value of the parameter k .

So the algorithm allows to find the independent set S and to calculate the absolute estimate $(\max S(G) - |S|)$.

Algorithm $\mathcal{B}(G)$: for solving MIS Problem with an absolute estimate of the accuracy

Require: undirected graph $G = (V, \mathcal{E})$

Ensure: the independent set $S (S \subseteq V)$ and the estimate Est for deviation from the exact solution

```

1:  $V_0 \leftarrow V$  {the current set of vertices consists of applicants for inclusion in
   the independent set}
2:  $S \leftarrow \{\}$ 
3:  $Est \leftarrow 0$ 
4: while  $V_0 \neq \{\}$  do
5:   for all  $v \in V_0$  do
6:     calculate  $k[v]$ 
7:     calculate  $m[v]$ 
8:   end for
9:    $v_0 \leftarrow MinMaxParam(v)$ 
10:   $S \leftarrow S \cup \{v_0\}$ 
11:   $Est \leftarrow Est + m[v_0]$ 
12:   $V_0 \leftarrow V_0 - \{v_0\} - \mathcal{N}(v_0, V_0)$  {remove from the set of applicants the
   selected vertex together with neighborhood}
13: end while

```

This algorithm was implemented and used to find the maximum independent set in complementary graphs DIMACS. It is clear that in the initial graph the solution obtained will correspond to the clique. Then it became possible to compare results with other effective algorithms.

3.1 Properties of the Algorithm

The following corollary shows that in some cases the Algorithm $\mathcal{B}(G)$ allows to find the exact solution of the MIS Problem.

Corollary 3. *If $Est = 0$ then the independent set S found by Algorithm $\mathcal{B}(G)$ is the maximum.*

Proof. If $Est = 0$ then $m[v] = 0$ for all $v \in S$. This means that all these vertices are k -vertices. Thus, according to Proposition 1 the independent set obtained is the maximum.

It is clear that $Est = 0$ is not always satisfied. This motivates to study particular classes of graphs for which the condition of the Corollary 3 hold.

Proposition 5. *For any connected graph $G = (V, \mathcal{E})$, where $|V| > 2$, there exists the maximum independent set that contains all hanging vertices of this graph.*

Proof. Consider the connected graph $G = (V, \mathcal{E})$, where $|V| > 2$. Let $v \in V$ be hanging vertex. Implementation of the algorithm $\mathcal{B}(G)$ allows different states of V_0 . Let's consider each of them in detail.

1) The vertex v is a hanging and unique k -vertex. Then $S \leftarrow S \cup \{v\}$ as required.

2) The vertex v is a hanging and not unique k -vertex. Consider the vertex u adjacent to the hanging vertex v . The vertex u can not be included in the set S at the first step, since the neighborhood of the vertex u is not the complete

induced subgraph. Indeed the vertex v is hanging and adjacent to u , that is the vertex v is in the neighborhood of the vertex u . The graph $G = (V, \mathcal{E})$ is connected, then in the neighborhood of the vertex u (in the first step, when V_0 coincides with V) there are also other vertices. But the vertex v can not be adjacent to any of them, since $\deg(v, V_0) = 1$ and the vertex adjacent to v is the vertex u . Thus, the vertex u is not a k -vertex and will not be included in S on the first step of the algorithm.

A similar situation occurs at any step if at least one vertex remains in the neighborhood of the vertex u except for the hanging vertex v . Vertices in the neighborhood of the vertex u will still not be adjacent, and therefore the vertex u will not be a k -vertex.

Let at some step all vertices adjacent to u be removed except for the hanging vertex v . This can happen when vertices adjacent to vertices from the neighborhood of the vertex u will be included in S . In this situation the vertex u becomes a k -vertex and can be included in S . However, the vertex v can also be included in S , which does not reduce the cardinality of S .

Indeed, let $s: |S| = s$ be the cardinality of S at some step. And let the vertex u be included in S and the vertex v be not included, then

$$S \leftarrow S \cup \{u\} \implies |S| = s + 1 .$$

Now suppose that the vertex v be included in S and the vertex u be not included, then

$$S \leftarrow S \cup \{v\} \implies |S| = s + 1 ,$$

that is, one can include the hanging vertex without decreasing the cardinality of the independent set. Note that if the above situation occurs, then all vertices which are included in S are k -vertices. Thus, according to Corollary 3 there exists the maximum independent set of vertices which contains the hanging vertex v .

Arguing similarly for each hanging vertex we get what was required to prove.

Thus, we can formulate a more general proposition about the Algorithm $\mathcal{B}(G)$.

Proposition 6. *The Algorithm $\mathcal{B}(G)$ implemented in the class of trees allows to find the exact solution of the MIS Problem.*

Proof. According to Corollary 3 the independent set found by the Algorithm $\mathcal{B}(G)$ is maximum if $Est = 0$. In the case when initial graph is a tree at each step its hanging (or isolated) vertices could be included in S . In this way $Est = \sum_{v \in S} m[v] = 0$ and the independent set S is the maximum as required.

Note that Proposition 6 is the property of the Algorithm $\mathcal{B}(G)$. In addition to this algorithm, the method of dynamic programming from [7] can be effectively used to solve the MIS Problem in the class of trees. However, there are still structural distinctions between these two approaches.

3.2 Complexity of the Algorithm

Let us discuss the algorithmic complexity of the Algorithm $\mathcal{B}(G)$.

For each vertex v it is necessary to find the number of vertices in the neighborhood and the number of edges that should be added. We remove the vertices $v \cup \mathcal{N}(v, V_0)$ and the edges $e \in G(\{v\} \cup \mathcal{N}(v, V_0))$ until V_0 becomes empty. Let n and m be the number of vertices and the number of edges of the initial graph respectively.

Then we obtain the following estimate. The common number of iterations is less than or equal to n . Each iteration needs no more than $\mathcal{O}(n \cdot m)$ actions to calculate parameters and no more than $\mathcal{O}(m)$ actions to remove selected vertex together with neighborhood. Thus, the complexity of the Algorithm $\mathcal{B}(G)$ is $\mathcal{O}(n \cdot n \cdot m + n \cdot m) = \mathcal{O}(n^2 \cdot m)$.

4 Computational Results

The table below shows the results of computational experiments with complementary graphs DIMACS. The table gives the results of the comparison with effective algorithms from [3]–[5].

In the column $\omega(G)$ there is maximal clique of the corresponding graph, and the exact solution (maximum clique) is highlighted in bold. In the column $\mathcal{B}(G)$ the results for proposed heuristic algorithm are given.

Table 1. Computational results with complementary graphs DIMACS

	DIMACS	$\omega(g)$	approximate solution	Median Runtimes In Seconds					$\mathcal{B}(G)$
				IncMaxCLQ [3]	BBMCX [4]	RPC [5]			
						$\delta=1$	$\delta=2$	$\delta=3$	
1	C125.9	34	33						0,001
2	C250.9	44	42	333.2	1144	1041.2	987.8	971.2	0,005
3	DSJC1000 5	15	13	261.2	211				0,529
4	DSJC500 5	13	11		1.66				0,082
5	MANN a27	126	126	0.43	0.318				0,01
6	MANN a45	345	344	114.1	41.4	31.2	34.6	85.6	48,151001
7	MANN a81	1100	1099						696,870972
8	brock400 2	29	22	259.9	132	107.7	103.8	104	0,024
9	brock400 4	33	22	197.7	20.2	79.7	76.9	77.1	0,04
10	brock800 2	24	18		3586	2971.6	2845.8	2842.2	0,19
11	brock800 4	26	17		2532	1405.8	1341.4	1333.3	0,191
12	hamming8-4	16	16						0,011
13	p hat300-1	8	8						0,038
14	p hat300-2	25	25						0,029
15	p hat300-3	36	34	0.87	0.531	1.1	1	1.2	0,015
16	p hat700-1	11	8						0,363
17	p hat700-2	44	43	1.25	1.42	2.3	2.3	2.4	0,691
18	p hat700-3	62	61	357.4	718	894.5	841.3	854.4	0,144

It can be seen that the best-known solution was obtained in some cases only but the approximate solution is rather close to the best-known or exact ones. Furthermore, the algorithm shows high efficiency in terms of time complexity of calculations.

5 Conclusion

The paper presents the algorithm for MIS Problem with the absolute estimate of the accuracy of the approximate solution. Special notions were introduced and theoretical results of research were presented. The properties of the algorithm are investigated and, in particular, its convergence in the class of trees is shown. The results of computational experiments on complementary graphs DIMACS are presented, which demonstrate rather good results in terms of the quality of the approximate solution and high efficiency in terms of the time complexity of the calculations.

Acknowledgement. The results of the work were obtained within the framework of the state task of the Ministry of Education and Science of RF no. 2.2461.2017/4.6.

References

1. Skiena, S.: The Algorithm Design Manual. Springer (2008)
2. Korte, B., Vygen, J.: Combinatorial Optimization. Springer (2008)
3. Li, C., Fang, Z., Xe, K.: Combining MaxSAT reasoning and incremental upper bound for the maximum clique problem. In: Proceedings of the 25th International Conference on Tools with Artificial Intelligence. pp. 939–946 (2013)
4. Batsyn, M., Nikolaev, A., San Segundo, P.: Infra-chromatic bound for exact maximum clique search. Computers & Operations Research 64, 293–303 (2015)
5. Batsyn, M., Nikolaev, A., San Segundo, P.: Reusing the same coloring in the child nodes of the search tree for the maximum clique problem. In: LNCS. vol. 8994, pp. 275–280 (2015)
6. Gainanov, D.N., Rasskazova, V.A.: An inference algorithm for monotone Boolean functions associated with undirected graphs. Bulletin of the SUSU 9(3), 17–30 (2016)
7. Dasgupta, S., Papadimitriou, C., Vazirani, U.: Algorithms. McGraw-Hill (2006)