Discussiones Mathematicae
Graph Theory 37 (2017) 415–426
doi:10.7151/dmgt.1965

ON SEQUENTIAL HEURISTIC METHODS FOR THE MAXIMUM INDEPENDENT SET PROBLEM

Ngoc C. Lê

Faculty of Mathematics and Computer Science
Technische Universität Bergakademie Freiberg
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology

e-mail: lechingoc@yahoo.com

Christoph Brause and Ingo Schiermeyer

Faculty of Mathematics and Computer Science
Technische Universität Bergakademie Freiberg

e-mail: brause@math.tu-freiberg.de
ingo.schiermeyer@tu-freiberg.de

Abstract

We consider sequential heuristics methods for the Maximum Independent Set (MIS) problem. Three classical algorithms, VO [11], MIN [12], or MAX [6], are revisited. We combine Algorithm MIN with the α-redundant vertex technique [3]. Induced forbidden subgraph sets, under which the algorithms give maximum independent sets, are described. The Caro-Wei bound [4, 14] is verified and performance of the algorithms on some special graphs is considered.

Keywords: maximum independent set, heuristic, MIN, MAX, VO, vertex ordering.

2010 Mathematics Subject Classification: 05C69, 05C85.

1. Introduction

In a simple graph G, a set of vertices is *independent* (or *stable*) if no two vertices in this set are adjacent. The cardinality of a maximum size independent set in G is called the *independence number* of G and denoted by $\alpha(G)$. The problem of determining an independent set of maximum cardinality finds important applications in various fields, some examples are computer vision and pattern recognition. It is well-known that the problem is generally NP-hard. Hence, exact algorithms for even with some hundred-vertex graphs are impractical. In this sense, heuristic algorithms are good candidates.

Sequential heuristics methods generate a maximal independent set through repeated addition of a vertex into an independent set or repeated deletion of a vertex from the original graph. Borowiecki *et al.* [2] called the two strategies *best-in* and *worst-out* strategies, respectively. Decisions on which vertex to be added in or moved out next are based on certain indicators associated with candidate vertices. For example, a possible best-in heuristic constructs a maximal independent set by repeatedly adding in a vertex that has the smallest degree among candidate vertices. In this case, the indicator is the degree of a vertex. On the other hand, a possible worst-out heuristic can start with the whole vertex set V and then repeatedly remove a vertex out of V until V becomes independent.

Three well known heuristic algorithms are Vertex Order (VO) [11], MIN [12], and MAX [6]. Algorithm MAX follows worst-out strategy using degree indicator, i.e., it repeatedly removes maximum degree vertices until every remaining vertex is of degree zero, i.e., they compose an independent set. MIN and VO follow best-in strategy with the same indicator, i.e., they repeatedly choose the minimum degree vertices and add them to the being constructed independent set if no conflict occurs.

Moreover, while MIN and MAX update the indicators every time when a vertex is added in or moved out, we call this approach as *new* strategy, VO does not, but follows so-called *old* strategy.

Based on the three above algorithms, one can think about a greedy heuristic method based on old worst-out strategy working like first order the vertex set of a graph G in decreasing degree order. Then the algorithm proceeds through the list, adds a vertex to the being constructed independent set if it has no neighbor in the remaining graph and removes it from G. The process was repeated until the list is empty. However, a deeper analysis shows that actually this algorithm and Algorithm VO produce the same maximal independent set for every graph.

All three above algorithms give a maximal independent set in polynomial time. However, under some restrictions, these maximal independent sets become maximum. We investigate on these conditions for algorithms in Section 4.

Some others useful techniques for solving the problem are transformation methods. In this paper, we consider α-redundant vertex deletion. We call a vertex $u \in V(G)$ *α-redundant* for G if $\alpha(G) = \alpha(G - u)$ [3]. We can repeat the deletion of vertices until we got a simple (enough) graph, i.e., a graph that we already have efficient algorithm for solving the problem.

The advantage of graph transformation methods is that they can lead to a maximum independent set. However, graph transformation methods can only work (well) under some restrictions, say they need some special structures to be contained in the graph. More on this technique can be found in [8]. In Section 3, we combine the strength of MIN algorithm and this method.

Let us introduce some notations which will be used throughout this paper. We consider only finite undirected, simple graphs $G = (V(G), E(G))$. For $u, v \in V(G)$, we denote $u \sim v$ if $uv \in E(G)$ and $u \nsim v$ if $uv \notin E(G)$. Let $N_G(x)$ be the *neighbourhood* of a vertex $x \in V(G)$, $N_G[x] = N_G(x) \cup \{x\}$ be the *closed neighbourhood* of x, and $\deg_G(x) = |N_G(x)|$ be the *degree* of x in G. For a vertex x and a vertex subset or an induced subgraph U, we denote $N_U(x) := N_G(x) \cap U$, similarly for $N_U[x]$, and $\deg_U(x)$. We also denote $N_G(U) = \bigcup_{u \in U} N_G(u) \backslash U$ and for a set $W \subset V(G)$, let $N_U(W) = N_G(W) \cap U$. If no confusion arises, we write $N(.)$, $N[.]$, and $\deg(.)$ instead of $N_G(.)$, $N_G[.]$, and $\deg_G(.)$, respectively, for short. Let $\delta(G) = \min_{u \in V(G)} |N(u)|$.

For a set $U \subset V(G)$, we denote $G[U]$ as the *induced subgraph* of G on U and $G - U := G[V(G) \backslash U]$. For short, we write $G - v$ for $G - \{v\}$.

If $F_1, F_2, \ldots, F_k$ are graphs, then we say that G is $\{F_1, F_2, \ldots, F_k\}$-*free* if G does not contain a copy of any of the graphs $F_1, F_2, \ldots, F_k$ as an induced subgraph.

As usual, we denote $K_{m,n}$ as a complete bipartite graph with cardinalities of the two parts m and n, respectively, and P_n as a chordless path having n vertices. For a $K_{1,m}$, the vertex of degree m is called the *center-vertex* while the m others vertices are called *leaves*.

2. Caro-Wei Bound

Given a graph G, we denote $k_{MIN}(G)$, $k_{MAX}(G)$, and $k_{VO}(G)$ the smallest cardinalities of the maximal independent sets obtained by the MIN, MAX, and VO algorithms, respectively. Caro [4] and Wei [14] independently used MIN algorithm to discover a lower bound on $\alpha(G)$ in terms of the degree sequence of G, i.e.:

$$\alpha(G) \geq k_{MIN}(G) \geq \sum_{v \in V(G)} \frac{1}{\deg(v) + 1}.$$

As they observed, the above bound is sharp, i.e., we have the equality if G is a union of disjoint cliques. Griggs [6] also showed that Algorithm MAX can be used to prove the Caro-Wei bound. Surprisingly, the VO algorithm also can be employed to obtain this bound as shown in the following observation.

Proposition 1.

$$k_{VO}(G) \geq \sum_{v \in V(G)} \frac{1}{\deg(v) + 1}.$$

Proof. The proof mimics the similar proofs for Algorithms MIN [14] and MAX [6].

We consider the VO algorithm. Let (u_i), $i = 1, 2, \ldots, k_{VO}$, be the (ordered) vertices added in the resulting maximal independent set. Let $H_1 := G$ and $H_{i+1} := H_i - N_{H_i}[u_i]$, for $i = 1, 2, \ldots, k_{VO}$. It is obvious that each vertex v belongs to $N_{H_i}[u_i]$ for some u_i. Moreover, if $v \in N_{H_i}[u_i]$, then v appears after u_i in the list generated by the algorithm, i.e., $\deg_{H_i}(u_i) \le \deg_G(u_i) \le \deg_G(v)$. Hence,

$$k_{VO}(G) = \sum_{i=1}^{k_{VO}} \frac{\deg_{H_i}(u_i)+1}{\deg_{H_i}(u_i)+1} \ge \sum_{i=1}^{k_{VO}} \sum_{v \in N_{H_i}[u_i]} \frac{1}{\deg(v)+1} = \sum_{v \in V(G)} \frac{1}{\deg(v)+1}. \qquad \blacksquare$$

We refer the readers to the result of Borowiecki *et al.* [2] about a Caro-Wei-like bound using potential function of vertices, a generalization of degree.

3. MIN and α-Redundance

Now, we describe a modified version of Algorithm MIN (see Algorithm 1) based on α-redundant vertex method. Clearly, a vertex u is α-redundant if there exists some maximum independent set not containing u. Based on this, Gerber and Lozin [5] obtained the following result which can be derived by Lemma 1 of [5].

Lemma 2 [5]. *Given a graph G containing an induced $K_{1,m}$, $\{u, v_1, v_2, \ldots, v_m\}$, where u is the center vertex (i.e., the vertex of degree m). If there exist no vertices $u_1, u_2, \ldots, u_m$ such that $\{u, u_1, u_2, \ldots, u_m\}$ is an independent set and $u_i \sim v_i$ for $i = 1, 2, \ldots, m$, then u is an α-redundant vertex for G.*

It leads to the following consequence.

Corollary 3. *Given a graph $G = (V, E)$, a vertex u of G, and vertices $v_1, v_2 \in N(u)$ such that $v_1 \nsim v_2$. If there exist no vertices u_1, u_2 such that $\{u, u_1, u_2\}$ is independent and $\{u, u_1, u_2, v_1, v_2\}$ induces a $K_{2,3}$ or a banner or a P_5, then u is α-redundant.*

By this corollary, we introduce the following notation. Given a graph G and a vertex u of G, let us say that u *enjoy property P for G* if there exist vertices $v_1, v_2 \in N(u)$ such that $v_1 \nsim v_2$ and there exist no vertices u_1, u_2 such that $\{u, u_1, u_2\}$ is independent and $\{u, u_1, u_2, v_1, v_2\}$ induces a $K_{2,3}$ or a banner or a P_5. Then Corollary 3 states that: Given a graph G and a vertex u of G, if u enjoys property P for G then u is α-redundant for G.

Remind that if the neighborhood of u contains no non-edge, then u is a simplicial vertex [13] and we know that u belongs to some maximum independent set, i.e., we can add u to the being constructed independent set.

Algorithm 1 MMIN(G)

Input: A graph G

Output: A maximal independent set of G.

1: $I := \emptyset$; $i := 1$; $H_i := G$;
2: **while** $V(H_i) \neq \emptyset$ **do**
3: Choose $u \in V(H_i)$ such that $\deg_{H_i}(u) = \delta(H_i)$;
4: **if** u enjoys property P **then**
5: $H_{i+1} := H_i - u$; $i := i + 1$;
6: **end if**
7: $I := I \cup \{u\}$; $i := i + 1$; $H_i := H_{i-1} - N_{H_{i-1}}[u]$;
8: **end while**
9: **return** I

Consider an arbitrary graph G, and let $n = |V(G)|$. The algorithm repeatedly chooses a minimum degree vertex u, then it checks and removes u if it enjoys property P. We can find a minimum degree vertex of G in time $\mathcal{O}(n^2)$. Moreover, for a vertex u, we can check if it is simplicial in time $\mathcal{O}(n^2)$ (Step 4). Given that (v_1, u, v_2) induces a P_3, we can check if there exist vertices u_1, u_2 such that $\{u, u_1, u_2, v_1, v_2\}$ induces a $K_{2,3}$, a banner, or a P_5 in time $\mathcal{O}(n^2)$. For each u, such a test can be performed in time at most $\mathcal{O}(n^2)$. Clearly, Algorithm MMIN gives a maximal independent set. Hence, we have the following result.

Theorem 4. *For a graph $G = (V, E)$, Algorithm MMIN gives a maximal independent set in time $\mathcal{O}(n^5)$, where $n = |V(G)|$.*

It is worth to notify that in [9], we did the same for Algorithm MAX and obtain Algorithm MMAX. Say, after choosing a vertex of maximum degree u, before removing it, we look for an α-redundant vertex in the neighborhood of u and remove such vertex instead of u.

4. Forbidden Induced Subgraphs

In this section, we describe sufficient conditions for heuristic algorithms mentioned in the above sections. Mahadev and Reed [11] characterized a graph class, for which a maximum independent set can be obtained by Algorithm VO. Harant *et al.* [7] and Zverovich [15] obtained the similar results for Algorithm MIN. In [9], we obtained the forbidden subgraphs for Algorithms MAX and MMAX. These results are summarized in the following theorem.

Theorem 5 [7, 9, 11, 15]**.** *Let* (*see Figure* 1)

$\mathcal{F}_1 = \{F_1, F_2, F_3, F_4, F_5, F_6\}$,

$\mathcal{F}_2 = \{F_1, F_3, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}\}$,

$\mathcal{F}_3 = \{F_1, F_4, F_5, F_6, F_7, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, F_{21}, F_{22}, F_{23}, F_{24}\}$,

$\mathcal{F}_4 = \{F_4, F_{15}, F_{19}, F_{20}, F_{21}, F_{24}, F_{25}, F_{26}, F_{27}\}$, *and*

$\mathcal{F}_5 = \{F_1, F_5, F_7, F_8, F_{14}, F_{15}, F_{18}, F_{20}, F_{21}, F_{24}, F_{37}, F_{38}, F_{39}\}$.

Then Algorithm VO always generates a maximum independent set for $\mathcal{F}_1$-free graphs, analogously, Algorithm MIN for $\mathcal{F}_2$-free and $\mathcal{F}_3$-free, Algorithm MAX for $\mathcal{F}_4$-free, and Algorithm MMAX for $\mathcal{F}_5$-free.

The following result describes a forbidden subgraph set for Algorithm MMIN.

Theorem 6. *Algorithm MMIN always gives us a maximum independent set for $\mathcal{F}_6$-free graphs, where*

$$\mathcal{F}_6 = \{F_1, F_7, F_{14}, F_{15}, F_{20}, F_{24}, F_{28}, F_{29}, F_{30}, F_{31}, F_{32}, F_{33}, F_{34}, F_{35}, F_{36}\}.$$

Proof. We basically follow the idea used in [15] with replacing the MIN algorithm by MMIN algorithm.

Let G be an $\mathcal{F}_6$-free connected graph. Suppose that the algorithm fails for G and G is a minimal graph (inclusive sense) with respect to this property. Then there exists some $u_0 \in V(G)$ such that

1. u_0 is of minimum degree in G,
2. u_0 does not enjoy property P for G, and
3. u_0 does not belong to any maximum independent set of G.

Claim 7. *Every maximum independent set of G contains $N(u_0)$.*

Proof. If the statement does not hold, then there is a maximum independent set I of G and a vertex $v \in N(u_0)\backslash I$. Let $G' = G - v$. Then clearly, I is independent in G'. Hence, $\alpha(G') \geq |I| = \alpha(G)$. So, $\alpha(G') = \alpha(G)$, i.e., every maximum independent set of G' is a maximum independent set of G.

We show that u_0 does not enjoy property P for G'. Then by the minimality of G, u_0 belongs to some maximum independent set J of G' which is also a maximum independent set of G, a contradiction.

To show that u_0 does not enjoy property P for G', we have to show that for arbitrary vertices $v_1, v_2 \in N_{G'}(u_0) \subseteq N_G(u_0)$ such that (v_1, u_0, v_2) induces a P_3, there exist vertices u_1, u_2 in G' such that $\{u_0, u_1, u_2, v_1, v_2\}$ induces a P_5 or a banner or a $K_{2,3}$. Since u_0 does not enjoy property P for G, for such v_1, v_2, there exist vertices $u_1, u_2 \in V(G)$ such that $\{u_0, u_1, u_2, v_1, v_2\}$ induces a $K_{2,3}$ or a banner or a P_5 in G. Note that, such $u_1, u_2 \notin N_G(u_0)$, hence, $u_1, u_2 \in V(G')$. Thus, u_0 does not enjoy property P for G'. □

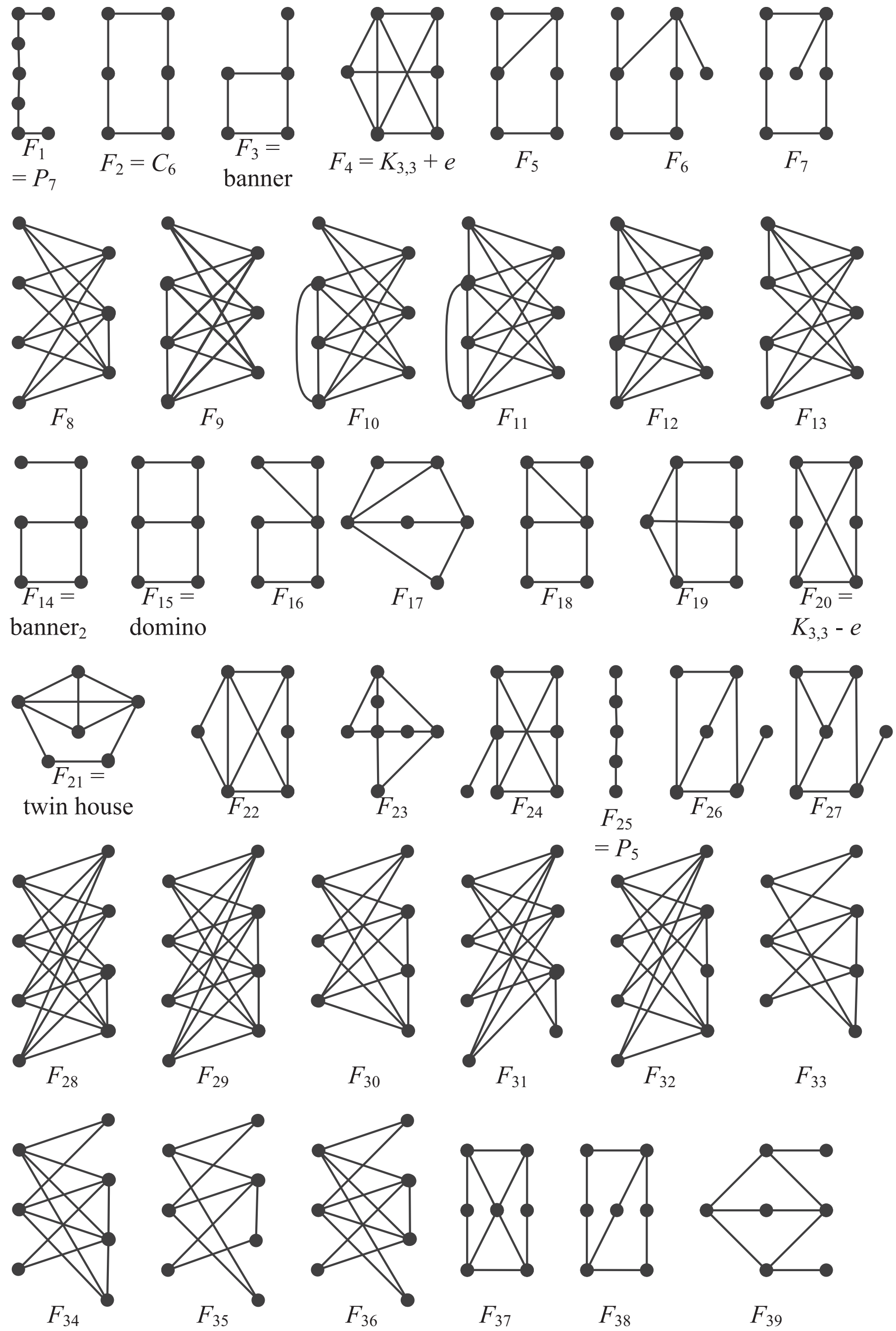

Figure 1. Forbidden induced subgraphs for some heuristic greedy algorithms.

Let S be a maximum independent set of G and $T = V(G)\backslash S$. Then $u_0 \in T$ and $N_G(u_0) \subseteq S$.

Claim 8. *Let $u \in T$ be at distance two from u_0. Then $|N_S(u)| \geq 2$*

Proof. Since the distance between u_0 and u is two, there exists some $w \in N_S(u_0) \cap N_S(u)$. If the statement is not true, then $S' = (S\backslash\{w\}) \cup \{u\}$ is a maximum independent set of G and $N(u_0) \nsubseteq S'$, a contradiction to Claim 7. □

Claim 9. *There exist vertices $u_1, u_2 \in T$ and $v_1, v_2 \in S$ such that $\{v_1, v_2, u_0, u_1, u_2\}$ induces a $K_{2,3}$.*

Proof. Since u_0 does not belong to any maximum independent set, u_0 is not simplicial, and thus there exist vertices $v_1, v_2 \in N(u_0)$ such that (v_1, u_0, v_2) induces a P_3. Because u_0 does not enjoy property P for G, there exists some u_1, u_2 such that $\{u_0, u_1, u_2, v_1, v_2\}$ induces a $K_{2,3}$ or a banner or a P_5. By symmetry, we only have to consider the two following cases.

Case 1. $\{u_0, u_1, u_2, v_1, v_2\}$ induces a P_5 and $u_1 \sim v_1$, $u_2 \sim v_2$. Since both u_1, u_2 are of distance two from u_0, by Claim 8, $|N_S(u_1)|, |N_S(u_2)| \geq 2$. Consider the two following subcases.

1.1. There exists some $v_3 \in N_S(u_1) \cap N_S(u_2)$. We have $v_3 \nsim u_0$, otherwise $\{u_0, u_1, u_2, v_1, v_2, v_3\}$ induces an F_{15}, a contradiction. Since $(S\backslash\{v_1, v_2, v_3\}) \cup \{u_0, u_1, u_2\}$ is not independent, there exists some $v_4 \in S\backslash\{v_1, v_2, v_3\}$ such that v_4 is adjacent to at least one of vertices u_0, u_1, u_2. Now, $\{u_0, u_1, u_2, v_1, v_2, v_3, v_4\}$ induces an F_7 or an F_{14} or an F_{15} depending on whether v_4 is adjacent to exactly one vertex or two or three vertices of $\{u_0, u_1, u_2\}$, a contradiction.

1.2. $N_S(u_1) \cap N_S(u_2) = \emptyset$. Then there exists some $v_3 \in N_S(u_1)\backslash N(u_2)$ and $v_4 \in N_S(u_2)\backslash N(u_1)$. We have $v_3 \nsim u_0$ (similarly, $v_4 \nsim u_0$), otherwise $\{u_0, u_1, u_2, v_1, v_2, v_3\}$ induces an F_{14}, a contradiction. Now, $\{v_3, u_1, v_1, u, v_2, v_4\}$ induces an F_1, a contradiction.

Case 2. $\{u_0, u_1, u_2, v_1, v_2\}$ induces a banner and $u_2 \nsim v_1$. Since u_2 is of distance two from u_0, there exists some $v_3 \in N_S(u_2)\backslash\{v_2\}$. Then $\{v_1, v_3, u_0, u_1, u_2\}$ induces an F_{14} or an F_{15}, a contradiction, or a $K_{2,3}$, depending on whether v_3 is adjacent to none, one, or two vertices among $\{u_0, u_1\}$. □

Claim 10. *There exist no vertices $u_1, u_2 \in T$ and $v_1, v_2, v_3, v_4 \in S$ such that $\{u_0, u_1, u_2, v_1, v_2, v_3, v_4\}$ induces a $K_{3,4}$.*

Proof. By contradiction, suppose that there exist vertices $u_1, u_2 \in T$ and $v_1, v_2, v_3, v_4 \in S$ such that $\{u_0, u_1, u_2, v_1, v_2, v_3, v_4\}$ induces a $K_{3,4}$. Let H be a maximal induced complete bipartite subgraph of G with parts A and B such that $\{v_1, v_2, v_3, v_4\} \subseteq A \subseteq S$ and $\{u_0, u_1, u_2\} \subseteq B \subseteq T$. Consider the two following cases.

Case 1. $|B| < |A|$. Since $\deg(u_0) \leq \deg(v_1)$, there exists some $t \in N(v_1)\backslash (N(u_0) \cup B)$. This also implies $t \in T$. Consider the two following subcases.

1.1. t is adjacent to every vertex of A. Then t is adjacent to some $u_i \in B$, otherwise we have a contradiction with the maximality of H. Without loss of generality, suppose that $t \sim u_1$. Then $\{u_0, u_1, u_2, t, v_1, v_2, v_3, v_4\}$ induces an F_{29} or an F_{28} depending on whether $t \sim u_2$ or not, a contradiction.

1.2. t is non-adjacent to some vertex of A. Without loss of generality, assume that $t \nsim v_2$. Then we show that more generally $t \nsim v_j$ for every $v_j \in A\backslash\{v_1\}$. Indeed, by contradiction suppose that $t \sim v_j$ for some $v_j \in A\backslash\{v_1, v_2\}$. Then $t \sim u_k$ for every $u_k \in B\backslash\{u_0\}$, otherwise $\{u_0, u_k, t, v_1, v_2, v_j\}$ induces an F_{20}, a contradiction. Now, $\{u_0, u_1, u_2, t, v_1, v_2, v_j\}$ induces an F_{30}, a contradiction. Then the assertion is shown.

It follows that $\{u_0, u_1, u_2, t, v_1, v_2, v_3, v_4\}$ induces a graph containing one of the following graphs F_{24}, F_{31}, or F_{32} as an induced subgraph depending on the adjacency between t and $\{u_1, u_2\}$, a contradiction.

Case 2. $|B| \geq |A| \geq 4$, i.e., there exists some $u_3 \in B\backslash\{u_0, u_1, u_2\}$. Since $S' = (S\backslash A) \cup B$ is not independent (otherwise S' is a maximum independent set containing u_0, a contradiction), there exists some $w \in S\backslash A$ such that w is adjacent to at least one vertex of B, assume that $w \sim u_j$. Note that w cannot be adjacent to all u_i belonging to B because of the maximality of H. Assume that $w \nsim u_k$ for some $u_k \in B$. If w is adjacent to some vertex $u_l \in B\backslash\{u_j, u_k\}$, then $\{u_j, u_k, u_l, w, v_1, v_2\}$ induces an F_{20}, a contradiction. If w is non-adjacent to any vertex of B but u_j, then $V(H) \cup \{w\}$ induces a graph containing F_{24} as an induced subgraph, a contradiction. □

Now, by Claim 9, let $u_1, u_2 \in T$ and $v_1, v_2 \in S$ be such that $\{v_1, v_2, u_0, u_1, u_2\}$ induces a $K_{2,3}$. Let $A = N_S(\{u_0, u_1, u_2\})$. Since $|(S\backslash A) \cup \{u_0, u_1, u_2\}| < |S|$ (otherwise we have a maximum independent set containing u_0, a contradiction), $|A| \geq 4$. Moreover, since $(S\backslash\{v_1, v_2\}) \cup \{u_i, u_j\}$ is not independent for every two vertices u_i, u_j of u_0, u_1, u_2 (otherwise we have a maximum independent set not containing all neighbors of u_0, a contradiction with Claim 7), there exist vertices $v_3, v_4 \in N_S(\{u_0, u_1, u_2\})\backslash\{v_1, v_2\}$ such that $|N_{\{u_0,u_1,u_2\}}(\{v_3, v_4\})| \geq 2$. By Claim 10, $|N_{\{u_0,u_1,u_2\}}(v_3)|$ or $|N_{\{u_0,u_1,u_2\}}(v_4)|$ is smaller than three.

If $N_{\{u_0,u_1,u_2\}}(v_3) = 2$ (similarly for the case $|N_{\{u_0,u_1,u_2\}}(v_4)| = 2$), then $\{u_0, u_1, u_2, v_1, v_2, v_3\}$ induces an F_{20}, a contradiction.

If $|N_{\{u_0,u_1,u_2\}}(v_3)| = 1$ and $|N_{\{u_0,u_1,u_2\}}(v_4)| = 3$ (or vice versa), then $\{u_0, u_1, u_2, v_1, v_2, v_3\}$ induces an F_{24}, a contradiction.

The remaining case is $|N_{\{u_0,u_1,u_2\}}(v_3)| = |N_{\{u_0,u_1,u_2\}}(v_4)| = 1$. Without loss of generality, we assume that v_3 is adjacent to u_1 but neither to u_0 nor u_2. Since $\deg(u_0) \leq \deg(v_3)$, there exists some $u_3 \in N(v_3)\backslash N(u_0)$.

If $u_3 \nsim u_1$, then $\{u_0, u_1, u_3, v_1, v_2, v_3\}$ induces an F_{14} or an F_{15} or an F_{20}

depending on the adjacency between u_3 and $\{v_1, v_2\}$, a contradiction. Then let us assume that $u_3 \sim u_1$.

If u_3 is adjacent to u_2 and not adjacent to v_1, v_2, then $\{u_0, u_1, u_2, u_3, v_1, v_2\}$ induces an F_{20}, a contradiction.

If u_3 is adjacent to u_2 and adjacent to exactly one of v_1, v_2, then $\{u_0, u_1, u_2, u_3, v_1, v_2, v_3\}$ induces an F_{33}, a contradiction.

If u_3 is adjacent to u_2, v_1, v_2, then $\{u_0, u_1, u_2, u_3, v_1, v_2, v_3\}$ induces an F_{34}, a contradiction.

If $u_3 \nsim u_2$ and u_3 is adjacent to exactly one vertex of v_1, v_2, then $\{u_0, u_2, u_3, v_1, v_2, v_3\}$ induces an F_{14}, a contradiction.

If u_3 is not adjacent to v_1, v_2, u_2, then $\{u_0, u_1, u_2, u_3, v_1, v_2, v_3\}$ induces an F_{35}, a contradiction.

If $u_3 \nsim u_2$ and u_3 is adjacent to v_1, v_2, then $\{u_0, u_1, u_2, u_3, v_1, v_2, v_3\}$ induces an F_{36}, a contradiction. ■

5. Comparison

The following results are obvious.

Proposition 11.

- *F_7 contains F_2, and $F_8, \ldots, F_{13}$ contain F_4.*
- *$F_{14}, \ldots, F_{24}, F_{26}, F_{27}$ contain F_3.*
- *F_{28}, F_{29} contain F_8, and $F_{30}, \ldots, F_{36}$ contain F_{21}.*

Proposition 12.

- *Every $\mathcal{F}_1$-free graph is $\mathcal{F}_2$-free and $\mathcal{F}_3$-free.*
- *Every $\mathcal{F}_2$-free graph and every $\mathcal{F}_3$-free graph is $\mathcal{F}_6$-free.*

Some observations from the above results are:

- MIN performs better than VO, and
- MMIN performs better than MIN,

all in forbidden induced subgraphs set sense. Now, we compare the greedy heuristic algorithms by considering their performances on some special graphs. Given two graphs G_1, G_2, let $G_1 + G_2$ denote the disjoint union of G_1 and G_2, i.e., $V(G_1+G_2) = V(G_1) \cup V(G_2)$ and $E(G_1+G_2) = E(G_1) \cup E(G_2)$, and let $G_1 \vee G_2$ denote the graph obtained from $G_1 + G_2$ by adding edges from each vertex of G_1 to each vertex of G_2.

Proposition 13. *For every integer p, there exist graphs G such that*

$$k_{MMIN}(G) - k_{MIN}(G) > p.$$

Proof. Let $G := (K_{p+3} + K_{p+3}) \vee \overline{K_{p+3}}$. Then $k_{MIN}(G) = 2$ while $k_{MMIN}(G) = p + 3 = \alpha(G)$. ■

6. Discussion

So far, there are not many results about polynomial time solution for the MIS problem in some subclasses of P_7-free graphs, except for (P_7,banner)-free graphs [1], and $(P_7, K_{1,m})$-free graphs [10]. Our results for Algorithm MMIN can be considered as a contribution in subclasses of P_7-free graph. Remind that the complexity of the problem for the class of P_7-free graphs is still an open question.

Our results in this direction also follow the approach of Mahadev and Reed [11], Harant *et al.* [7], Zverovich [15], and Lê *et al.* [9]. Moreover, our forbidden induced subgraph set for Algorithm MMIN covers the two sets for Algorithm MIN in [7] and [15].

Besides, greedy heuristic methods can be easily implemented and they also have low complexity compared with augmenting methods used by Alekseev and Lozin [1] and Lozin and Milanič [10].

Our combined methods also suggest that we can combine other (conditionally) exact methods with greedy methods to obtain interesting algorithms, especially in choosing the next vertex in general by best-in or worst-out strategies.

Acknowledgement

This research is supported by National Foundation for Science and Technology Development (NAFOSTED) of Vietnam, project code: 101.99-2016.20.

We would like to express the appreciation to the anonymous reviewer for his/her very useful advices, comments, and corrections.

References

[1] V.E. Alekseev and V.V. Lozin, *Augmenting graphs for independent sets*, Discrete Appl. Math. **145** (2004) 3–10. doi:10.1016/j.dam.2003.09.003

[2] P. Borowiecki, F. Göring, J. Harant and D. Rautenbach, *The potential of greed for independence*, J. Graph Theory **71** (2012) 245–259. doi:10.1002/jgt.20644

[3] A. Brandstädt and V.V. Lozin, *A note on α-redundant vertices in graphs*, Discrete Appl. Math. **108** (2001) 301–308. doi:10.1016/S0166-218X(00)00239-0

[4] Y. Caro, New results on the independence, Technical Report, Tel-Aviv University (1979).

[5] M.U. Gerber and V.V. Lozin, *On the stable set problem in special P_5-free graphs*, Discrete Appl. Math. **125** (2003) 215–224. doi:10.1016/S0166-218X(01)00321-3

[6] J.R. Griggs, *Lower bounds on the independence number in terms of the degrees*, J. Combin. Theory Ser. B **34** (1983) 22–39. doi:10.1016/0095-8956(83)90003-5

[7] J. Harant, Z. Ryjáček and I. Schiermeyer, *Forbidden subgraphs and MIN-algorithm for independence number*, Discrete Math. **256** (2002) 193–201.
doi:10.1016/S0012-365X(02)00571-X

[8] N.C. Lê, C. Brause and I. Schiermeyer, *New sufficient conditions for α-redundant vertices*, Discrete Math. **338** (2015) 1674–1680.
doi:10.1016/j.disc.2014.07.002

[9] N.C. Lê, C. Brause and I. Schiermeyer, *Extending the Max Algorithm for maximum independent set*, Discuss. Math. Graph Theory **35** (2015) 365–386.
doi:10.7151/dmgt.1811

[10] V.V. Lozin and D. Rautenbach, *Some results on graphs without long induced paths*, Inform. Process. Lett. **88** (2003) 167–171.
doi:10.1016/j.ipl.2003.07.004

[11] N.V.R. Mahadev and B.A. Reed, *A note on vertex orders for stability number*, J. Graph Theory **30** (1999) 113–120.
doi:10.1002/(SICI)1097-0118(199902)30:2⟨113::AID-JGT5⟩3.0.CO;2-#

[12] O. Murphy, *Lower bounds on the stability number of graphs computed in terms of degrees*, Discrete Math. **90** (1991) 207–211.
doi:10.1016/0012-365X(91)90357-8

[13] D.J. Rose, R.E. Tarjan and G.S. Lueker, *Algorithmic aspects of vertex elimination of graphs*, SIAM J. Comput. **5** (1976) 266–283.
doi:10.1137/0205021

[14] V.K. Wei, A lower bound on the stability number of a simple graph, Technical Report, Bell Laboratories (1981).

[15] I.E. Zverovich, *Minimum degree algorithms for stability number*, Discrete Appl. Math. **132** (2004) 211–216.
doi:10.1016/S0166-218X(03)00402-5

Received 29 December 2015
Revised 1 February 2017
Accepted 1 February 2017