IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E102-A NO. 9
SEPTEMBER 2019

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER *Special Section on Discrete Mathematics and Its Applications*

An Approximation Algorithm for the Maximum Induced Matching Problem on C_5-Free Regular Graphs

Yuichi ASAHIRO[†a)], **Guohui LIN**[††b)], **Zhilong LIU**[†††c)], ***Nonmembers*, *and* Eiji MIYANO**[†††d)], ***Member***

SUMMARY In this paper, we investigate the maximum induced matching problem (MaxIM) on C_5-free d-regular graphs. The previously known best approximation ratio for MaxIM on C_5-free d-regular graphs is $\left(\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8}\right)$. In this paper, we design a $\left(\frac{2d}{3} + \frac{1}{3}\right)$-approximation algorithm, whose approximation ratio is strictly smaller/better than the previous one when $d \geq 6$.

key words: *induced matching problem, C_5-free regular graph, approximation algorithm*

1. Introduction

Let $G = (V, E)$ be a simple unweighted graph, where V and E are the set of vertices and the set of edges, respectively. Two edges are called *adjacent* if they have a common vertex. A *matching* in the graph G is a subset of edges, no two of which are adjacent. A matching M is *induced* if no two vertices belonging to different edges of M are adjacent. In other words, an induced matching M in G is formed by the edges of a 1-regular induced subgraph of G. An induced matching is often called the *strong matching* [5], [7].

The Maximum Induced Matching problem (MaxIM) is that of finding an induced matching of maximum cardinality in an input graph. The MaxIM problem was originally introduced by Stockmeyer and Vazirani [14] as a variant of the Maximum Matching problem and motivated as the Risk-Free Marriage problem. Induced matchings have applications in the areas of concurrent transmission of messages in wireless ad hoc networks [1], secure communication channels in broadcast networks [6], communication network testing [14], and many other fields. Thus, MaxIM has received much attention in recent years.

The MaxIM problem is generally intractable. Stockmeyer and Vazirani [14], and Cameron [2] independently proved that MaxIM is NP-hard. Also, it remains NP-hard for several graph classes such as planar graphs of vertex degree at most four [9], bipartite graphs of vertex degree at most three [11], [13], line graphs, chair-free graphs, Hamiltonian graphs [10], and d-regular graphs for $d \geq 3$ [3].

In this paper, we focus only on d-regular graphs as input and consider the approximability of MaxIM on d-regular graphs. Zito [15] proved that a natural greedy strategy yields an approximation algorithm for MaxIM on d-regular graphs with approximation ratio $d - \frac{1}{2} + \frac{1}{4d-2}$. Then, Duckworth, Manlove, and Zito [3] improved the approximation ratio slightly into $\frac{n(d-1)}{n-2}$, i.e., asymptotically $d - 1$ for d-regular graphs of n vertices. Subsequently, Gotthilf and Lewenstein [8] provided a $\left(\frac{3d}{4} + 0.15\right)$-approximation algorithm for MaxIM on d-regular graphs by combining a greedy approach with a local search.

For subclasses of d-regular graphs, several better approximation algorithms are known. Rautenbach [12] designed a $(0.7084d + 0.425)$-approximation algorithm for MaxIM on $\{C_3, C_5\}$-free d-regular graphs. Fürst, Leichter, and Rautenbach [4] provided approximation algorithms for the following three subclasses of d-regular graphs: a $\left(\frac{9d}{16} + \frac{33}{80}\right)$-approximation algorithm for C_4-free d-regular graphs, a $\left(\frac{d}{2} + \frac{1}{4} + \frac{1}{8d-4}\right)$-approximation algorithm for $\{C_3, C_4\}$-free d-regular graphs, and a $\left(\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8}\right)$-approximation algorithm for C_5-free d-regular graphs.

The goal of this paper is to improve the previously best known $\left(\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8}\right)$-approximation algorithm for C_5-free d-regular graphs [4], and we design a $\left(\frac{2d}{3} + \frac{1}{3}\right)$-approximation algorithm, whose approximation ratio is strictly smaller/better than the previously best one when $d \geq 6$. It is important to note that our approximation algorithm works also for $\{C_3, C_5\}$-free d-regular graphs, i.e., MaxIM on $\{C_3, C_5\}$-free d-regular graphs can be better (than [12]) approximated within an approximation ratio of $\left(\frac{2d}{3} + \frac{1}{3}\right)$ for $d \geq 3$.

Related work. The inapproximability results on MaxIM for graph subclasses are also known. Duckworth, Manlove, and Zito [3] proved that for any $\varepsilon > 0$, it is NP-hard to approximate MaxIM on graphs of maximum degree three within $\frac{475}{474} - \varepsilon$, 3-regular graphs within $\frac{2375}{2374} - \varepsilon$, and bipartite graphs of maximum degree three within $\frac{6600}{6659} - \varepsilon$.

On the other hand, polynomial-time algorithms for MaxIM have been developed, for example, for chordal graphs, interval graphs [2], trees [5], circular-arc graphs [7], trapezoid graphs, k-interval-dimension graphs, and cocomparability graphs [6].

Manuscript received September 21, 2018.

[†]The author is with Kyushu Sangyo University, Fukuoka-shi, 813-8503 Japan.
[††]The author is with University of Alberta, Edmonton, Alberta, T6G 2E9, Canada.
[†††]The authors are with Kyushu Insitute of Technology, Iizuka-shi, 820-8502 Japan.
a) E-mail: asahiro@is.kyusan-u.ac.jp
b) E-mail: guohui@ualberta.ca
c) E-mail: q794023s@mail.kyutech.jp
d) E-mail: miyano@ces.kyutech.ac.jp (Corresponding author)
DOI: 10.1587/transfun.E102.A.1142

2. Preliminaries

Let $G = (V, E)$ be a simple, unweighted, and undirected graph, where V and E denote the set of vertices and the set of edges, respectively. $V(G)$ and $E(G)$ also denote the vertex set and the edge set of G, respectively. Throughout the paper, let $n = |V|$ and $m = |E|$ for any given graph. Let $G[V']$ denote a vertex-induced subgraph of $G = (V, E)$, consisting of a subset $V' \subseteq V$ and all the edges connecting pairs of vertices in V'. Also, let $G[E']$ denote an edge-induced subgraph of $G = (V, E)$, consisting of a subset $E' \subseteq E$ and the vertices that are endpoints of edges in E'. Let H be a set of graphs. A graph is *H-free* if it does not contain any graph in H as a vertex-induced subgraph.

For a vertex v in a graph G, the *open neighborhood* of v in G is $N_G(v) = \{u \in V(G) \mid \{u, v\} \in E(G)\}$ and the *closed neighborhood* of v in G is $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of v in G is denoted by $deg_G(v) = |N_G(v)|$. A graph G is *d-regular* if all the vertices in G have degree d. Throughout the paper, we assume that $d \geq 3$ since MaxIM on 1-regular and 2-regular graphs can be solved in polynomial time.

A (simple) *path* P_k with k vertices $v_1, v_2, \cdots, v_k$ is represented as a sequence $\langle v_1, v_1, \cdots, v_k \rangle$ of those k vertices where $\{v_i, v_{i+1}\}$ is an edge in P_k for each $i = 1, 2, \cdots, k-1$. The *length* of the path P is the number of edges in P, i.e., the length of P_k with k vertices is $k - 1$. A *cycle* C_k with k vertices is similarly written as $C_k = \langle v_1, v_2, \cdots, v_k, v_1 \rangle$.

For a pair of vertices v and v' in G, the *distance* between v and v' is the length of a shortest path from v to v', which is denoted by $dist_G(v, v')$. For the path $P = \{v_1, v_2, v_3, v_4, v_5, \cdots, v_k\}$ of length $k - 1$, for example, $dist_P(v_1, v_1) = 0$, $dist_P(v_1, v_2) = 1$, $dist_P(v_1, v_3) = 2$ and so on. If $dist_G(v, v') = \ell$ for two vertices v and v', then v' is called a distance-ℓ vertex of v. Let $DV_\ell(v)$ be a set of distance-ℓ vertices of v. Similarly, for a pair of edges e and e' in $E(G)$, we define the distance $dist_G(e, e')$ between two edges e and e': The *line graph* $L(G)$ of G is the graph whose vertices are the edges of G, and in which two vertices are adjacent only if they share an incident vertex as edges of G. Then, the distance $dist_G(e, e')$ between two edges e and e' in G is defined as $dist_{L(G)}(e, e')$ between two vertices e and e' in $L(G)$, i.e., the length of a shortest path from e to e' in the line graph $L(G)$ of G. For example, for $P = \langle v_1, v_2, v_3, v_4, v_5, \cdots, v_k \rangle$ again, $dist_P(\{v_1, v_2\}, \{v_1, v_2\}) = 0$, $dist_P(\{v_1, v_2\}, \{v_2, v_3\}) = 1$, $dist_P(\{v_1, v_2\}, \{v_3, v_4\}) = 2$, and so on. If $dist_G(e, e') = \ell$ for two edges e and e', then e' is called a distance-ℓ edge of e. Let $DE_\ell(e)$ be a set of distance-ℓ edges of e. Furthermore, we define the distance between an edge e and a vertex v as the length of a shortest path from one endpoint of e to v, i.e., $dist_G(e, v) = \min\{dist_G(v_e, v), dist_G(v'_e, v)\}$ for $e = \{v_e, v'_e\}$. For example, $dist_P(\{v_2, v_3\}, v_1) = 1$, $dist_P(\{v_2, v_3\}, v_4) = 1$, $dist_P(\{v_2, v_3\}, v_5) = 2$, and so on.

We say that an edge $e \in E(G)$ is *in conflict with* another edge $e' \in E(G)$ if $dist_G(e, e') \leq 2$ and the edge $e \in E(G)$ is called a *conflict edge* of $e' \in E(G)$. Then, for an edge e of a graph G, let

$$C_G(e) = \{e' \in E(G) \mid dist_G(e, e') \leq 2\} = \{e\} \cup DE_1(e) \cup DE_2(e).$$

be the set of all the conflict edges of e. Also, the set of all the conflict edges of a set $E' \subseteq E(G)$ is defined as follows:

$$C_G(E') = \bigcup_{e \in E'} C_G(e).$$

For a subset $E' \subseteq E(G)$ of edges and an edge e in G, let

$$PC_G(E', e) = C_G(e) \setminus \bigcup_{e' \in E' \setminus \{e\}} C_G(e')$$

be the set of edges that are in conflict with e but *not* in conflict with every $e' \in E' \setminus \{e\}$. The edge in $PC_G(E', e)$ is called a *private conflict edge* of e to the set E'. For example, for the graph G shown in Fig. 1, the conflict edges of e are e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, and e. Also, the private conflict edges of e to the set $M = \{e, f, f', f''\}$ are e_2, e_5, e_7, and e.

Let $OPT(G)$ be an optimal induced matching on the input G. We say that an algorithm ALG is a *σ-approximation algorithm* for MaxIM or that ALG's *approximation ratio* is at most σ if $|OPT(G)| \leq \sigma \cdot |ALG(G)|$ holds for any input G, where $ALG(G)$ is an induced matching returned by ALG.

3. Approximation Algorithm

In this section we design a $\left(\frac{2d}{3} + \frac{1}{3}\right)$-approximation algorithm for MaxIM on C_5-free d-regular graphs.

3.1 Algorithm

Here is an outline of our approximation algorithm for an input C_5-free d-regular graph G, which mainly consists of two steps. (i) In the first step, the algorithm initially finds a *maximal* induced matching M by iteratively picking an edge e into the induced matching M, and eliminating all the edges in $C_G(e)$ from the candidates of the solution. (ii) In the second step, the algorithm tries to find a larger induced matching from the temporally obtained induced matching M by a "small modification" as follows: Let M be the set of induced matching edges currently obtained. The algorithm picks one edge e from M. Then, if there exist (at least) two edges e' and e'' in $PC_G(M, e) \setminus \{e\}$ such that $dist_G(e', e'') > 2$, then the algorithm updates the "old" induced matching M to the "new" $M = (M \setminus \{e\}) \cup \{e', e''\}$. If there does not exist such an edge e in M, then the algorithm tries to find an edge e_{min} from $PC_G(M, e)$ such that $|C_G(e_{min})|$ is the minimum among $|C_G(e')|$ for every $e' \in PC_G(M, e)$. If the algorithm finds e_{min}, then it swaps e and e_{min}, i.e., updates $M = (M \setminus \{e\}) \cup \{e_{min}\}$.

The following is a description of our algorithm ALG, where let M be the induced matching obtained by ALG:

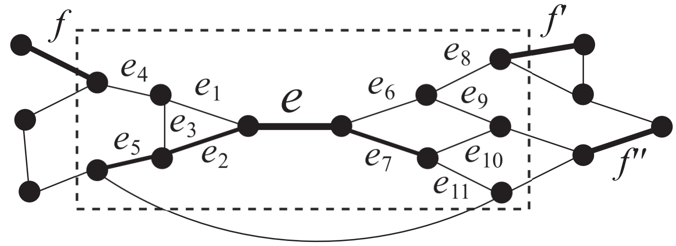


Fig. 1 Edges e_1, e_2, $\cdots$, e_{11} and e in the dotted-line rectangle are conflict edges of e. If $M = \{e, f, f', f''\}$, then the private conflict edges of e to M are e_2, e_5, e_7 and e.

Algorithm `ALG`

Input: A C_5-free d-regular graph $G = (V, E)$.
Output: An induced matching M of G.
Initialization: Set $M = \emptyset$, and obtain $C_G(e)$ and $|C_G(e)|$ for every edge $e \in E$.
Step 1. /* Find an initial maximal set M of induced matching edges. */
If $C_G(M) = E$, then go to **Step 2**; otherwise, arbitrarily select an edge e from $E \setminus C_G(M)$, set $M = M \cup \{e\}$ and repeat **Step 1**.
Step 2. /* Find a larger set M of induced matching edges */
Obtain $PC_G(M, e)$ for every $e \in M$.

(i) If there exists an edge e such that the size of a *maximal* induced matching $MAX(e)$ in $PC_G(M, e) \setminus \{e\}$ is at least two, then set $M = (M \setminus \{e\}) \cup MAX(e)$ and repeat **Step 2**.
(ii) If there exists a pair of edges $e \in M$ and $e' \in PC_G(M, e)$ such that $|C_G(e)| > |C_G(e')|$ and $|C_G(e')|$ is the minimum among $|C_G(e'')|$ for every $e'' \in PC_G(M, e)$, then set $M = (M\setminus\{e\})\cup\{e'\}$ and repeat **Step 2**.
(iii) Otherwise, go to **Termination**.

Termination. Output the solution M and halt.

[End of `ALG`]

Here is a detailed implementation of **Step 2**(i): Suppose that $PC_G(M, e)$ has k edges and let $PC_G(M, e) = \{e, e_1, e_2, \cdots, e_{k-1}\}$. Also, for each $1 \leq i \leq k-1$, let $MAX(e, e_i)$ be a maximal induced matching which is obtained by first selecting e_i from $PC_G(M, e) \setminus \{e\}$ and then selecting induced matching edges from $(PC_G(M, e) \setminus \{e\}) \setminus C_G(e_i)$ if such induced matching edges exist. In **Step 2**(i), `ALG` first obtains $k-1$ maximal induced matchings $MAX(e, e_1)$ through $MAX(e, e_{k-1})$, and then finds the set of maximum cardinality among those $k-1$ sets as $MAX(e)$. One can see that if there exists at least one maximal matching which has at least two induced matching edges, then `ALG` surely finds it in polynomial time.

Now we show the feasibility of the induced matching M output by `ALG`. One can see that if an edge e is selected into M, then all the edges in $C_G(e)$ are eliminated from candidates of the solution. Moreover, we can verify that each edge in $PC_G(M, e)$ is not in conflict with any edge in M except the edge e. Thus, the distance of any two edges in M is at least three and thus all the edges in the output M are induced matching edges. That is, `ALG` can always output a feasible induced matching M.

Next, we bound the running time of `ALG`: Clearly, **Initialization** and **Step 1** can be executed in $O(m^2)$ time. In each execution of **Step 2**(i), the number of induced matching edges in M is incremented at least by one. Hence the total number of executions of **Step 2**(i) is at most $O(m)$. Each iteration of **Step 2**(i) can be done in $O(m^2)$. Therefore, the total computational complexity of **Step 2**(i) is $O(m^3)$. As for **Step 2**(ii), if $|M| = i$ at some time point, then `ALG` has to check i private conflict edge sets, $PC_G(M, e_1)$ through $PC_G(M, e_i)$, in **Step 2**(ii). That is, the total number of executions of **Step 2**(ii) is at most $O(m^2)$. **Step 2**(ii) can be implemented in $O(m)$ time. Hence the total comutational complexity of **Step 2**(ii) is again $O(m^3)$. In the beginning of each iteration of **Step 2** we need $O(m^2)$ time to obtain $PC_G(M, e)$ for every $e \in M$. Since the iteration of **Step 2** is bounded from above by $O(m^2)$, the time complexity of **Step 2** is $O(m^4)$. Therefore, `ALG` runs in $O(m^4)$.

We make a detailed observation on **Step 2**: From the maximality of M, $\bigcup_{e \in M} C_G(e) = E(G)$ holds after **Step 1**. Now suppose that in some iteration of **Step 2**(i), `ALG` finds an edge e_1 such that a maximal induced matching $MAX(e_1)$ in $PC_G(M, e_1)$ has at least two induced matching edges. At this moment, $\bigcup_{e \in M \setminus \{e_1\}} C_G(e) = E(G) \setminus PC_G(M, e_1)$ holds since all the edges in $PC_G(M, e_1)$ are in conflict only with e_1. Moreover, from the maximality of $MAX(e_1)$, $PC_G(M, e_1) \subseteq \bigcup_{e' \in MAX(e_1)} C_G(e')$ must hold. Since `ALG` obtains a new temporal solution M' by setting $M' = (M \setminus \{e_1\}) \cup MAX(e_1)$ in **Step 2**(i), $\bigcup_{e \in M'} C_G(e) = E(G)$ is satisfied again for M'. Note that **Step 2**(ii) guarantees that when M is eventually output by `ALG`, $|C_G(e)| \leq |C_G(e')|$ must hold for every edge $e' \in PC_G(M, e)$. Therefore, from the termination condition of `ALG`, the following should be remarked:

Remark 1. When `ALG` terminates and outputs an induced matching M for an input graph G, the following three properties must be satisfied:

1. As for every private conflict edge set $PC_G(M, e)$ of e to M, any two edges in $PC_G(M, e)$ must be in conflict

with each other;

2. For every edge $e' \in PC_G(M, e)$, $|C_G(e)| \leq |C_G(e')|$ holds; and
3. $\bigcup_{e \in M} C_G(e) = E(G)$ holds, *i.e.*, M must be a maximal set of induced matching edges.

3.2 Approximation Ratio

In this section, we investigate the approximation ratio of the algorithm ALG. Now suppose that given a graph $G = (V, E)$, ALG finally outputs a set M of induced matching edges, and $|ALG(G)| = |M|$. Note that the output M by ALG cannot be enlarged by picking other two or more edges from $PC_G(M, e)$ if edge e is in M. We can obtain the following relationship between $|C_G(e)|$ and $|PC_G(M, e)|$:

Lemma 1. For any maximal set M of induced matching edges in a graph $G = (V, E)$, the following inequality is satisfied:

$$\sum_{e \in M} (|C_G(e)| - |PC_G(M, e)|) \geq 2(|E| - \sum_{e \in M} |PC_G(M, e)|).$$

Proof. Consider an edge e in a subset M of edges, the conflict edge set $C_G(e)$ of e, and the private conflict edge set $PC_G(M, e)$ of e to M. From the definitions, we know

$$\bigcup_{e \in M} (C_G(e) \setminus PC_G(M, e)) = E \setminus \left(\bigcup_{e \in M} PC_G(M, e) \right).$$

Since the private conflict edge sets are independent, the following equality holds:

$$\left| E \setminus \left(\bigcup_{e \in M} PC_G(M, e) \right) \right| = |E| - \sum_{e \in M} |PC_G(M, e)|.$$

Recall that every edge in $C_G(e) \setminus PC_G(M, e)$ must be included in at least one different conflict edge set, say, $C_G(e')$ of $e' \in M$ for $e' \neq e$. Therefore, the inequality holds. □

Now we can estimate the maximum number Γ_d of conflict edges of an edge e in d-regular graphs, which was shown in [3]:

Proposition 1 (Theorem 3.1 in [3]). For any edge e in a d-regular graph G, the number $|C_G(e)|$ of conflict edges is at most $2d^2 - 2d + 1$.

Let Γ_d be the upper bound of $|C_G(e)|$ of conflict edges over all of the edges $e \in E(G)$. One can see that the number $|C_G(e)|$ of conflict edges of the edge e gets much smaller than $2d^2 - 2d + 1$ if an edge e' in $C_G(e)$ is in a short cycle, for example, C_3 or C_4. Indeed, the following results are known [8]:

Proposition 2 (Lemmas 4 and 6 in [8]). If a cycle C_3 of length three contains an edge e in $C_G(e)$ of a d-regular graph G, then the cycle C_3 decreases the upper bound Γ_d of $|C_G(e)|$ by at least d. Moreover, if a cycle C_4 of length four contains an edge e in $C_G(e)$, then the cycle C_4 decreases the upper bound Γ_d by at least one.

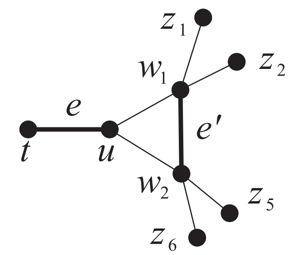


Fig. 2 An edge $e = \{t, u\}$ owns a triangle edge $e' = \{w_1, w_2\}$.

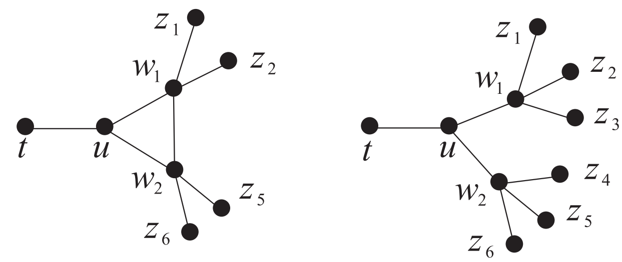


Fig. 3 Since an edge $e = \{t, u\}$ owns a triangle edge $e' = \{w_1, w_2\}$, $e' = \{w_1, w_2\}$ decreases the upper bound Γ_d of $|C_G(e)|$ by at least one.

Take a look at an edge $e = \{t, u\}$ illustrated in Figure 2. If two neighbor vertices, w_1 and w_2, of the edge e are connected by an edge $e' = \{w_1, w_2\}$, then e' is called the *triangle edge* of e, and we say that e owns the triangle edge e' or e' is the triangle edge of e. Then, we can obtain Lemma 2:

Lemma 2. If an edge e in a graph G owns a triangle edge e', then e' decreases the upper bound Γ_d of $|C_G(e)|$ by at least one.

Proof. This lemma can be obtained by a simple observation on two graphs illustrated in Fig. 3. The right graph does not have any triangle edge but the left one has one triangle edge $e' = \{w_1, w_2\}$. That is, we can think that two edges $\{w_1, z_3\}$ and $\{w_2, z_4\}$ in the right graph are replaced with one triangle edge $\{w_1, w_2\}$, or two edges are combined into one edge. Therefore, the value of Γ_d must decrease by at least one, because of the triangle edge e'. □

Now consider an edge $e = \{t, u\}$ in the solution M and the private conflict edges of e to M, $PC_G(M, e)$. Then, let $U_G(e) = (\{e' \mid dist_G(e', u) \leq 1\} \cap PC_G(M, e)) \setminus \{e\}$ and $T_G(e) = (\{e' \mid dist_G(e', t) \leq 1\} \cap PC_G(M, e)) \setminus \{e\}$. Roughly speaking, $U_G(e)$ and $T_G(e)$ are the "u-side" subset and the "t-side" subset of edges in $PC_G(M, e)$, respectively. Note that $PC_G(M, e) = U_G(e) \cup T_G(e) \cup \{e\}$ and $U_G(e) \cap T_G(e)$ may be non-empty. Moreover, let $U_G^0(e) = \{e' \in U_G(e) \mid dist_G(e', u) = 0\}$, $U_G^1(e) = U_G(e) \setminus U_G^0(e)$, $T_G^0(e) = \{e' \in T_G(e) \mid dist_G(e', t) = 0\}$, and $T_G^1(e) = T_G(e) \setminus T_G^0(e)$.

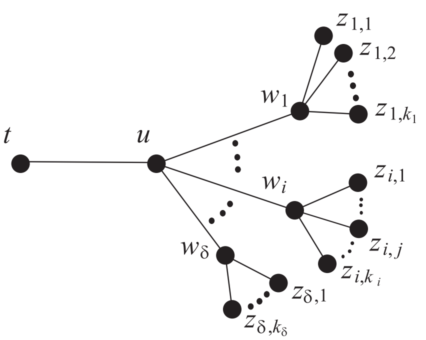


Fig. 4 $W_G(e) = V(G[U_G(e)]) \cap DV_1(u) = \{w_1, w_2, \cdots, w_\delta\}$ where w_i has k_i neighbors, $z_{i,1}$ through z_{i,k_i}.

From now on, let $|PC_G(M, e)| = \beta$. Without loss of generality, we assume that $|U_G(e)| \geq |T_G(e)|$ holds in the following. Then, we obtain the following lemma, which is quite trivial but plays a key role to estimate the approximation ratio of ALG:

Lemma 3. For each $e \in M$, $|U^1_G(e)| \geq \frac{\beta-1}{2} - (d-1)$ holds.

Proof. Clearly $|U^0_G(e)| \leq d - 1$ holds. Since $|U_G(e) \cup T_G(e)| = \beta - 1$ and $|U_G(e)| \geq |T_G(e)|$ by the assumptions, $|U_G(e)| \geq \frac{\beta-1}{2}$ is satisfied. Hence, we can obtain $|U^1_G(e)| = |U_G(e) \setminus U^0_G(e)| \geq \frac{\beta-1}{2} - (d-1)$. □

See Fig. 4. Let $W_G(e) = V(G[U_G(e)]) \cap DV_1(u) = \{w_1, w_2, \cdots, w_\delta\}$ be a set of δ neighbor vertices of u, where $\delta \leq |DV_1(u)| - 1$ holds (where "−1" comes from the edge $\{t, u\}$). Then, we define $U^1_G(e, w_i) = \{(w_i, v) \mid v \in DV_1(w_i)\} \cap U^1_G(e)$ for each $w_i \in W_G(e)$. Without loss of generality, we assume that $|U^1_G(e, w_1)| \geq |U^1_G(e, w_i)|$ for each $i = 2, \cdots, \delta$. Now, we consider the case where $|U^1_G(e, w_1)| \leq 1$ holds. Then, we obtain the following lemma:

Lemma 4. Suppose that $|U^1_G(e, w_1)| \leq 1$ and the algorithm ALG outputs a solution M. Then $|PC_G(M, e)| \leq 4d - 3$ and $|C_G(e)| + |PC_G(M, e)| \leq 2d^2 + 2d - 2$ hold for every induced matching edge $e \in M$.

Proof. From the definition, $PC_G(M, e) = \{e\} \cup U_G(e) \cup T_G(e)$. Then, by the assumption $|U_G(e)| \geq |T_G(e)|$, the following inequality holds:

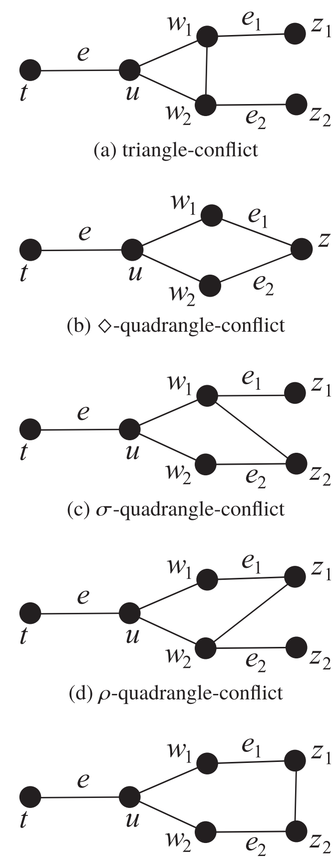
$$\begin{aligned} |PC_G(M, e)| &\leq 1 + |U_G(e)| + |T_G(e)| \\ &\leq 1 + 2|U_G(e)|. \end{aligned}$$

For a d-regular graph G, $|U^0_G(e)| \leq d - 1$ holds. The assumption $|U^1_G(e, w_1)| \leq 1$ means that $|U^1_G(e, w_i)| \leq 1$ holds for each i, $2 \leq i \leq \delta$. It follows that $|U^1_G(e)| \leq d - 1$ and $|U_G(e)| = |U^0_G(e)| + |U^1_G(e)| \leq 2(d-1)$. Therefore, $|PC_G(M, e)| \leq 1 + 4(d-1) = 4d - 3$ holds.

Since $|C_G(e)| \leq 2d^2 - 2d + 1$ as shown in Proposition 1, the inequality

$$\begin{aligned} |C_G(e)| + |PC(M, e)| &\leq (2d^2 - 2d + 1) + (4d - 3) \\ &= 2d^2 + 2d - 2 \end{aligned}$$

is obtained. □



(a) triangle-conflict

(b) ◇-quadrangle-conflict

(c) σ-quadrangle-conflict

(d) ρ-quadrangle-conflict

(e) pentagon-conflict

Fig. 5 Five types of conflicts of two edges e_1 and e_2 in $U^1_G(e)$.

Next, suppose that $|U^1_G(e, w_1)| \geq 2$ holds. We first depict all possible conflict ways of an edge of $U^1_G(e, w_1)$ and another edge of $U^1_G(e, w_i)$, where $i \neq 1$.

Recall that any two edges in $PC_G(M, e)$ (and thus any two edges in $U^1_G(e)$) are in conflict with each other to the solution M of ALG. There are five types of conflicts of two edges, say, e_1 and e_2, in $U^1_G(e)$ as follows: (a) triangle-conflict, (b) ◇-quadrangle-conflict, (c) σ-quadrangle-conflict, (d) ρ-quadrangle-conflict, and (e) pentagon-conflict. See Fig. 5 and consider two edges $e_1 = \{w_1, z_1\}$ and $e_2 = \{w_2, z_2\}$ in $U^1_G(e)$. (a) If e_1 is in conflict with e_2 since there exists the edge $\{w_1, w_2\}$ as shown in Fig. 5(a), then we say that e_1 and e_2 are in *triangle-conflict* with each other by the edge $\{w_1, w_2\}$. (b) See Fig. 5(b). If e_1 and e_2 are incident to a common vertex z and $U^1_G(e)$ does not have the edge $\{w_1, w_2\}$, then we say that e_1 and e_2 are in *◇-quadrangle-conflict* with each other. Note that if the graph shown in Fig. 5(b) has the edge $\{w_1, w_2\}$, then we regard the conflict of e_1 and e_2 as the triangle conflict caused by $\{w_1, w_2\}$. (c) If there exists the edge $\{w_1, z_2\}$ but does not exist the edge $\{w_1, w_2\}$ as shown in Fig. 5(c), then we say that e_1 and e_2 are in *σ-quadrangle-conflict* with each other by $\{w_1, z_2\}$. (d) If there exists the edge $\{w_2, z_1\}$ but does not exist the edge $\{w_1, w_2\}$ as shown in Fig. 5(d), then we say that e_1 and e_2 are in *ρ-quadrangle-conflict* with each other by $\{w_2, z_1\}$. (e) See Fig. 5(e). If there exists the edge $\{z_1, z_2\}$ but does not exist the edge $\{w_1, w_2\}$, then we say that e_1 and e_2 are in *pentagon-conflict* with each other by $\{z_1, z_2\}$. Recall, however, that all the input graphs are now C_5-free. It follows that the induced cycle $\langle u, w_1, z_1, z_2, w_2, u\rangle$ of length 5 must have at least one edge inside of it. For example, the graph has the edge $\{w_1, z_2\}$, then we regard the conflict of e_1 and e_2 as the σ-quadrangle-conflict caused by $\{w_1, z_2\}$. Therefore, we do not need to take the pentagon-conflict into account.

In the following, we slightly change the previous definition of *triangle edges*. (We call the previously defined triangle edge the *original triangle edge* in the following.) An edge in $U^1_G(e)$ is called a *triangle edge* of the edge e if its one endpoints is w_i and the other is w_j in $W_G(e) \setminus \{w_i\}$, where $w_i \neq w_1$, $w_j \neq w_1$, and $w_i \neq w_j$. That is, for example, an edge $\{w_1, w_3\}$ is not regarded as a triangle edge since its one endpoint is w_1. Let $TE_G(e)$ be the set of triangle edges. Then, we define as follows:

$$A_G(e) = U^1_G(e) \setminus (U^1_G(e, w_1) \cup TE_G(e)).$$

Every edge e_2 in $A_G(e)$ is in conflict with every edge e_1 in $U^1_G(e, w_1)$, and $|U^1_G(e, w_1)| \geq |U^1_G(e, w_i)|$ from the definition. Then, all the edges in $A_G(e)$ are divided into the following two sets, the sets of *triangle-conflict edges* and *quadrangle-conflict edges*.

Triangle-Conflict edge: If an edge e' in $A_G(e)$ is in triangle-conflict with an edge in $U^1_G(e, w_1)$, then we say that e' is a *triangle-conflict* edge. Let $TC_G(e)$ be the set of triangle-conflict edges.

Quadrangle-Conflict edge: If an edge e' in $A_G(e)$ is in $\diamond$-quadrangle, σ-quadrangle, or ρ-quadrangle-conflict with an edge in $U^1_G(e, w_1)$, then we simply say that the edge e' is a *quadrangle-conflict* edge. Let $QC_G(e)$ be the set of quadrangle-conflict edges.

From the definitions, $U^1_G(e) = TC_G(e) \cup QC_G(e) \cup U^1_G(e, w_1) \cup TE_G(e)$ and $TC_G(e) \cap QC_G(e) = \emptyset$ hold.

Recall that we are now assuming that $|U^1_G(e, w_1)| \geq 2$. We take a look at the edge $e' = \{u, w_1\}$ and calculate the cardinality of the set $C_G(e')$ of conflict edges of e'. Note that each edge in $TC_G(e)$ creates one cycle C_3 of length three, which contains e', and each edge in $QC_G(e)$ creates one cycle C_4 of length four, which contains e'. Also, each edge in $TE_G(e)$ must be an original triangle edge of e'. It follows that each edge in $TC_G(e) \cup QC_G(e) \cup TE_G(e)$ causes decrease of the upper bound Γ_d of $|C_G(e')|$ by at least one from Proposition 2 and Lemma 2.

Lemma 5. Suppose that $|U^1_G(e, w_1)| \geq 2$. Also, suppose that the algorithm ALG outputs a solution M. Then, $|C_G(e')| \leq 2d^2 - \frac{\beta}{2} - \frac{1}{2}$ holds, where $e' = \{u, w_1\}$.

Proof. See Fig. 4 again and take a look at triangle-conflict, quadrangle-conflict, and (original) triangle edges in the following:

(i) Suppose that p vertices in $\{w_2, w_3, \cdots, w_\delta\}$ of $\delta - 1$ vertices are endpoints of *triangle-conflict* edges. Then, we can verify that there are p cycles of length three which contain the edge $e' = \{u, w_1\}$. Therefore, by Proposition 2, the value of the upper bound Γ_d of e' is reduced by at least pd. Since each of those p vertices is connected to at most $d - 1$ edges in $TC_G(e)$, $|TC_G(e)| \leq p(d-1) \leq pd$ holds. Namely, we can estimate that each edge in $TC_G(e)$ reduces the value of Γ_d of e' by at least one on average.

(ii) Each edge in $QC_G(e)$ obviously generates one cycle of length four which contains the edge $e' = \{u, w_1\}$. Thus, by Proposition 2, we can also estimate that each edge in $QC_G(e)$ decreases the value of Γ_d of e' by at least one.

(iii) Clearly, each edge in $TE_G(e)$ is a triangle edge of e. Also, it is an *original* triangle edge of $e' = \{u, w_1\}$. Then, by Lemma 2, we can estimate that each edge in $TE_G(e)$ decreases the value of Γ_d of e' by at least one.

Consequently, we can estimate that each edge in $TC_G(e) \cup QC_G(e) \cup TE_G(e)$ decreases the value of Γ_d of e' by at least one. Thus, all the edges in $TC_G(e) \cup QC_G(e) \cup TE_G(e)$ decrease the value of Γ_d of e' by at least $|TC_G(e) \cup QC_G(e) \cup TE_G(e)|$ in total.

Now, recall that $U^1_G(e) = TC_G(e) \cup QC_G(e) \cup U^1_G(e, w_1) \cup TE_G(e)$. Then,

$$\begin{aligned}
&|TC_G(e) \cup QC_G(e) \cup TE_G(e)| \\
&= |U^1_G(e) \setminus U^1_G(e, w_1)| \\
&\geq |U^1_G(e)| - (d-1)
\end{aligned}$$

holds since $|U^1_G(e, w_1)| \leq d - 1$. Furthermore, since $|U^1_G(e)| \geq \frac{\beta-1}{2} - (d-1)$ as shown in Lemma 3, we obtain the following:

$$\begin{aligned}
&|TC_G(e) \cup QC_G(e) \cup TE_G(e)| \\
&\geq |U^1_G(e)| - (d-1) \\
&\geq \left(\frac{\beta - 1}{2} - (d-1)\right) - (d-1) \\
&= \frac{\beta - 1}{2} - 2(d-1).
\end{aligned}$$

Therefore, the upper bound Γ_d of e' decreases by at least $\frac{\beta-1}{2} - 2d + 2$.

From Proposition 1, we obtain the following inequalities:

$$\begin{aligned}
|C_G(e')| &\leq 2d^2 - 2d + 1 - \left(\frac{\beta - 1}{2} - 2d + 2\right) \\
&= 2d^2 - \frac{1}{2} - \frac{\beta}{2}.
\end{aligned}$$

This completes the proof of this lemma. □

From Lemma 5, we can get the following corollary:

Corollary 1. Suppose that $|U^1_G(e, w_1)| \geq 2$ and the algorithm ALG outputs a solution M. Then, $|C_G(e)| \leq 2d^2 - \frac{1}{2} - \frac{\beta}{2}$ for every induced matching edge $e \in M$.

Proof. From Lemma 5, we know that there is an edge e' in $U_G(e)$ of $PC_G(M, e)$ such that $|C_G(e')| \leq 2d^2 - \frac{\beta}{2} - \frac{1}{2}$ for any induced matching edge e. Furthermore, Remark 1 shows that $|C_G(e)| \leq |C_G(e')|$ must be satisfied for e and e'. Therefore, $|C_G(e)| \leq 2d^2 - \frac{1}{2} - \frac{\beta}{2}$ holds. □

The above corollary gives us the following lemma:

Lemma 6. Suppose that $|U^1_G(e, w_1)| \geq 2$ and the algorithm ALG outputs a solution M. Then, $|PC_G(M, e)| \leq \frac{4d^2-1}{3}$, and $|C_G(e)| + |PC_G(M, e)| \leq \frac{8d^2-2}{3}$ hold for every induced matching edge $e \in M$.

Proof. From Corollary 1, we know that for each $e \in M$, $|C_G(e)| \leq 2d^2 - \frac{1}{2} - \frac{\beta}{2}$ holds. From the definitions, $PC_G(M, e) \subseteq C_G(e)$ holds. Therefore, we obtain

$$|PC_G(M, e)| = \beta \leq |C_G(e)| \leq 2d^2 - \frac{\beta}{2} - \frac{1}{2}.$$

That is, $\beta \leq 2d^2 - \frac{\beta}{2} - \frac{1}{2}$ holds and hence β is bounded from above as follows:

$$\beta \leq \frac{4d^2 - 1}{3}. \quad (1)$$

By the definition $|PC_G(M, e)| = \beta$,

$$\begin{aligned} |C_G(e)| + |PC_G(M, e)| &\leq 2d^2 - \frac{\beta}{2} - \frac{1}{2} + \beta \\ &= 2d^2 + \frac{\beta}{2} - \frac{1}{2} \\ &\leq \frac{8d^2 - 2}{3}, \end{aligned}$$

where the last inequality comes from the above (1). This completes the proof of this lemma. □

From Lemmas 4 and 6, we have the following corollary:

Corollary 2. Suppose that a solution M is obtained by the algorithm ALG. Then, $|C_G(e)| + |PC_G(M, e)| \leq \frac{8d^2-2}{3}$ holds for every induced matching edge $e \in M$.

Proof. By Lemma 6, we know that for $|U^1_G(e, w_1)| \geq 2$,

$$|C_G(e)| + |PC_G(M, e)| \leq \frac{8d^2 - 2}{3}.$$

From the assumption $d \geq 3$ and Lemma 4, we obtain the following inequality also for $|U^1_G(e, w_1)| \leq 1$:

$$\begin{aligned} |C_G(e)| + |PC_G(M, e)| &\leq 2d^2 + 2d - 2 \\ &\leq \frac{8d^2 - 2}{3}. \end{aligned}$$

This completes the proof of this corollary. □

The following is our main theorem:

Theorem 1. The algorithm ALG is a $\left(\frac{2d}{3} + \frac{1}{3}\right)$-approximation algorithm for MaxIM on C_5-free d-regular graphs, whose running time is $O(m^4)$.

Proof. From Remark 1, the solution for an input C_5-free d-regular graph $G = (V, E)$ satisfies the inequality in Lemma 1, that is, we have obtained

$$\begin{aligned} &\sum_{e \in M} (|C_G(e)| - |PC_G(M, e)|) \\ &\geq 2(|E| - \sum_{e \in M} |PC_G(M, e)|), \end{aligned}$$

or equivalently,

$$\sum_{e \in M} (|C_G(e)| + |PC_G(M, e)|) \geq 2|E|. \quad (2)$$

From Corollary 2 and $|ALG(G)| = |M|$, we obtain:

$$\begin{aligned} &\sum_{e \in M} (|C_G(e)| + |PC_G(M, e)|) \\ &\leq \frac{|ALG(G)|(8d^2 - 2)}{3}. \end{aligned} \quad (3)$$

Suppose that $|V| = n$, and hence $|E| = \frac{nd}{2}$. Then, the above (2) and (3) give the following inequality:

$$\frac{|ALG(G)|(8d^2 - 2)}{3} \geq nd.$$

Thus,

$$|ALG(G)| \geq \frac{3nd}{8d^2 - 2}.$$

It is known [15] that the size $|OPT(G)|$ of an optimal solution is at most $\frac{nd}{4d-2}$. Therefore, the approximation ratio is as follows:

$$\frac{|OPT(G)|}{|ALG(G)|} \leq \frac{2d}{3} + \frac{1}{3}.$$

□

4. Concluding Remarks

In this paper we have considered the approximability of MaxIM on C_5-free d-regular graphs. The previously best known approximation ratio was $(\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8})$. In this paper we have provided a $\left(\frac{2d}{3} + \frac{1}{3}\right)$-approximation algorithm ALG. One can verify that the new approximation ratio of ALG is strictly better than the old one when $d \geq 6$. Recall that ALG initially finds a maximal induced matching M in **Step 1**. However, it is important to note that **Step 1** can be replaced with the $\left(\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8}\right)$-approximation algorithm as a subroutine. **Step 2** surely finds an induced matching of the same or larger size than the initial induced matching. This implies that the "hybrid" approximation algorithm achieves the approximation ratio of $\min\left\{\frac{3d}{4} - \frac{1}{8} + \frac{3}{16d-8}, \frac{2d}{3} + \frac{1}{3}\right\}$ for MaxIM on C_5-free d-regular graphs for every $d \geq 3$.

Acknowledgments

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, the Grants-in-Aid for Scientific Research of Japan (KAKENHI) Grant Numbers JP17K00016 and JP17K00024, and JST CREST JPMJR1402.

References

[1] H. Balakrishnan, C.L. Barrett, V.S.A. Kumar, M.V. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks," IEEE J. Sel. Areas Commun., vol.22, no.6, pp.1069–1079, 2004.
[2] K. Cameron, "Induced matching," Discrete Appl. Math., vol.24, no.1-3, pp.97–102, 1989.
[3] W. Duckworth, D.F. Manlove, and M. Zito, "On the approximability of the maximum induced matching problem," J. Discrete Algorithmica, vol.3, no.1, pp.79–91, 2005.
[4] M. Fürst, M. Leichter, and D. Rautenbach, "Locally searching for large induced matchings," Theor. Comput. Sci., vol.720, pp.64–72, 2018.
[5] G. Fricke and R. Laskar, "Strong matching on trees," Congressus Numerantium, vol.89, pp.239–243, 1992.
[6] M.C. Golumbic and M. Lewenstein, "New results on induced matchings," Discrete Appl. Math., vol.101, no.1-3, pp.157–167, 2000.
[7] M.C. Golumbic and R.C. Laskar, "Irredundancy in circular arc graphs," Discrete Appl. Math., vol.44, no.1-3, pp.79–89, 1993.
[8] Z. Gotthilf and M. Lewenstein, "Tighter approximations for maximum induced matchings in regular graphs," Proc. 3rd Workshop on Approximation and Online Algorithms (WAOA'05), LNCS3879, pp.270–281, 2005.
[9] C.W. Ko and F.B. Shepherd, "Bipartite domination and simultaneous matroid covers," SIAM J. Discrete Math., vol.16, no.4, pp.517–523, 2003.
[10] D. Kobler and U. Rotics, "Finding maximum induced matchings in subclasses of claw-free and P_5-free graphs, and in graphs with matching and induced matching of equal maximum size," Algorithmica, vol.37, no.4, pp.327–346, 2003.
[11] V.V. Lozin, "On maximum induced matchings in bipartite graphs," Inform. Process. Lett., vol.81, no.1, pp.7–11, 2002.
[12] D. Rautenbach, "Two greedy consequences for maximum induced matchings," Theor. Comput. Sci., vol.602, pp.32–38, 2015.
[13] I. Rusu, "Maximum weight edge-constrained matchings," Discrete Appl. Math., vol.156, no.5, pp.662–672, 2008.
[14] L.J. Stockmeyer and V.V. Vazirani, "NP-completeness of some generalizations of the maximum matching problem," Inform. Process. Lett., vol.15, no.1, pp.14–19, 1982.
[15] M. Zito, "Maximum induced matchings in regular graphs and trees," Proc. the 25th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'95), LNCS1665, pp.89–100, 1999.

Yuichi Asahiro received B.Eng., M.Eng., and D.Eeng. degrees in computer science from Kyushu University in 1994, 1996, and 1998, respectively. He is currently a professor of Department of Information Science, Kyushu Sangyo University.

Guohui Lin received his B.Sc. in Applied Mathematics and M.Sc. in Operations Research from Zhejiang University in 1993 and 1995, respectively, and his PhD in Computer Science from Chinese Academy of Sciences in 1997. He is currently a professor of Computing Science at the University of Alberta.

Zhilong Liu received the M.Eng degree from Kyushu Institute of Technology in 2015. He is currently a Ph.D. student of the Department of Systems Design and Informatics, Kyushu Institute of Technology.

Eiji Miyano received the B.Eng., M.Eng., and Dr.Eng. degrees in computer science from Kyushu University in 1991, 1993, and 1995, respectively. He is currently a professor of the Department of Systems Design and Informatics, Kyushu Institute of Technology.