A New Algorithm of Online Edge-Coloring Algorithm

Masakuni TAKI, Mikihito SUGIURA*, Toshinobu KASHIWABARA*

SUMMARY We consider an edge coloring game of a graph by two persons who are an adversary and an edge-painter. The former strategy is to add or to delete edges in the graph successively. The strategy is called input. The latter colors the edge as soon as an edge is added in the graph. Then it is never changed the color of edges colored. The latter is not given any further information of the adversary's. We call the coloring strategies of the painter online edge-coloring algorithm.

In this paper the painter is limited the amount of colors. When the painter can use k colors, the adversary inputs to make bipartite graph that always has maximum degree of less than equal k. A minimum ratio which is a ratio between the number of edges colored of an online edge-coloring algorithm A for adversary's family of input and that of offline edge-coloring algorithm is called a competitiveness coefficient of an online edge-coloring algorithm A. It is better algorithm that the online edge-coloring algorithm has larger value of this coefficient. We have proved that a competitiveness coefficient of arbitrary deterministic online edge-coloring algorithm is 0, and present a randomized online-coloring algorithm that the competitiveness coefficient is greater than equal $\frac{1}{4}$ for oblivious adversary.

Key word: *edge coloring, online algorithm, competitive analysis, two persons game.*

1. Introduction

Given an undirected graph $G(V, E)$, an edge-coloring is an assignment of indices $1,...,n$ to the edges of G such that no two edges incident on a vertex have the same label. The indices are referred as colors, and the smallest value of n for which such a coloring can be achieved is called the chromatic index of the graph.

The online coloring algorithm is defined as follows. It is not given all of input data in advance. As soon as it is given an input data, it colors the input data one after another. It is not given any further information of input after it colors. In general they use a competitive analysis to analyze the online algorithm, because an absolute performance for the online algorithm dose often not make sense. This is to compare cost of online algorithm with that of offline algorithm for the same family of input data. The offline algorithm is an algorithm that deals with a family of input data all of which are known in advance.

In this paper we consider an edge-coloring game of a graph by two persons who are an adversary and an edge-painter. The former strategy is to add or to delete edges in the graph successively. The strategy is called input. Since the adversary makes a family of input. On the other hand, as soon as an edge is added in the graph, the latter has to color the edge. Then it is never changed the color of edges colored. The latter is not given any further information of the adversary's. We call the coloring strategies of the painter online edge-coloring algorithm.

We analyze a deterministic online edge-coloring algorithm and a randomized online edge-coloring algorithm under the following cases. Here, we introduce a new case that is not yet studied. The painter is limited the amount of colors. When the painter can use k colors, the adversary inputs to make bipartite graph that

* Auther is With the Department of Information and Computer Sciences, Osaka University Toyonaka-shi

always has maximum degree of less than equal k. The chromatic number of bipartite graph is equal to the maximum degree of the graph [3]. A minimum ratio which is a ratio between the number of edges colored of an online edge-coloring algorithm A for adversary's family of input and that of offline edge-coloring algorithm is called a competitiveness coefficient of an online edge-coloring algorithm A. It is better algorithm that the online edge-coloring algorithm has larger value of this coefficient. We have proved that a competitiveness coefficient of arbitrary deterministic online edge-coloring algorithm is 0, and present a randomized online-coloring algorithm that the competitiveness coefficient is greater than equal $\frac{1}{4}$ for oblivious adversary.

2. Preliminaries

Here we wish to refer to some notations about graph in [1], and the graph is undirected and is allowed to have multiple edges except on special occasions.

Theorem 1: The chromatic number of arbitrary graph G is equal to the maximum degree of the graph or is greater than it [1].

Theorem 2: The chromatic number of bipartite graph G is equal to the maximum degree of the graph [1].

We consider a two persons game by an edge-painter and an adversary. Let Π be a problem with a finite set I of input instances (fixed size), and a finite set of deterministic algorithms A. For input $i \in I$ and algorithm $a \in A$, let $C(i, a)$ denote the cost of algorithm a on input i. An edge-painter wants to make algorithms whose cost is smaller and an adversary wants to make input instances whose cost is larger. For probability distributions p over I and q over A, let i_p denote a random input chosen according to p and a_p denote a random algorithm chosen according to q.

Theorem 3 (Yao's Minimax Principle)[2]: For all distributions p over I and q over A,

$$\min_{a \in A} E[C(i_p,a)] \leqq \max_{i \in I} E[C(i,a_q)]$$

where $E[X]$ is expectation of X.

There are an oblivious adversary and an adaptive adversary for adversary of randomized online edge-coloring algorithm. When the former dose input, he cannot get the information that edge-painter had chosen colors for family of input before. When the latter dose input, he can get the information that edge-painter had chosen colors for family of input before. Since the adaptive adversary is stronger than the other.

Let a graph $G_0=(V \cup W, E, k)$ be the beginning of a graph G, here $|V|$, $|W| \gg k$, $E=0$. Adversary adds or deletes edges (v, w) (here, $v \in V$, $w \in W$) to G_0 successively, under the following conditions : the degree of all vertices are always less than equal k, and then the graph is always bipartite graph which has maximum degree of less than equal k on the way of inputting. The adversary knows the strategies of the edge-painter. There are colors which are $color_1$, $color_2$,..., $color_h$. When the adversary adds an edge, the edge-painter colors the edge then the color is different from each of adjacent edges. When these $color_1$, $color_2$,..., $color_h$ are all used on adjacent edges, he must not color even if these $color_1$, $color_2$,..., $color_h$ are all different on adjacent edges. The edge-painter never changes the color of the edge colored.

The edge-painter's aim is to make large number of edges colored according to the families of input of the adversary. The adversary's aim is to give the families of input that the ratio between number of edges colored of online edge-coloring algorithm and that of offline edge-coloring algorithm is made smaller. For every adding edge (v, w) the degree of vertices v and w are increased one, and for every deleting edge (v, w) the degree of vertices v and w are decreased one.

We define a competitiveness coefficient for analysis of deterministic online-edge coloring algorithm. Let $f_A(e_1,e_2,...,e_M)$ be the number of edges colored of deterministic online edge-coloring algorithm A according to the family of input $e_1,e_2,...,e_M$, and $f_O(e_1,e_2,...,e_M)$ be the number of edges colored of optimal offline edge-coloring algorithm according to the family of input $e_1,e_2,...,e_M$.

Definition 1: A deterministic online edge-coloring algorithm A is said to be C-competitive such that for any families of input $e_1,e_2,...,e_M$, a constant b exists,

$$f_A(e_1,e_2,...,e_M) - C \times f_O(e_1,e_2,...,e_M) \geqq b.$$

And a constant b must be independent of M, however k need not independence. The competitiveness coefficient of A, denoted C_A^{det} , is the supremum of C such that A is C-competitive.

Oblivious adversary is the same that he makes all of families of input previously, because he has no influence that edge-painter colored edges before. Let $f_R(e_1,e_2,...,e_M)$

be the number of edges colored of randomized online edge-coloring algorithm R according to the family of input $e_1,e_2,...,e_M$, then the $f_R(e_1,e_2,...,e_M)$ is the random variable.

Definition 2: For oblivious adversary a randomized online edge-coloring algorithm R is said to be C-competitive such that for any families of input $e_1,e_2,...,e_M$, a constant b exists,

$E[f_R(e_1,e_2,...,e_M)]-C\times f_o(e_1,e_2,...,e_M)\geqq b.$

And a constant b must be independent of M, however k need not independence. The competitiveness coefficient of R, denoted C_R^{obl}, is the supremum of C such that R is C-competitive.

Let P be a probability distribution of choosing families of input. The $f_A(e_1,e_2,...,e_M)$ and the $f_o(e_1,e_2,...,e_M)$ are random variables under P. For a deterministic online edge-coloring algorithm A, defined its competitiveness coefficient under P, C_A^P, to be supremum of C such that a constant b exists,

$E[f_A(e_1,e_2,...,e_M)]-C\times E[f_o(e_1,e_2,...,e_M)]\geqq b.$

Yao's Minimax Principle implies that

$$C_R^{obl}\leqq \max_A C_A^P.$$

An adaptive adversary can decide his input by knowing the information of previously inputs of edge-painter. For such families of input decided $e_1,e_2,...,e_M$, the $f_R(e_1,e_2,...,e_M)$ is the random variable and the $f_o(e_1,e_2,...,e_M)$ is also the random variable.

Definition 3: For an adaptive adversary a randomized online edge-coloring algorithm R is said to be C-competitive such that for the family of input $e_1,e_2,...,e_M$ made by the adaptive adversary.

A constant b exists,

$E[f_R(e_1,e_2,...,e_M)]-C\times E[f_o(e_1,e_2,...,e_M)]\geqq b.$

And a constant b must be independent of M, however k need not independence. The competitiveness coefficient of R, denoted C_R^{ada}, is the supremum of C such that R is C-competitive.

From definitions, these competitiveness coefficients are

$0\leqq C_A^{det}$, C_R^{ada}, $C_R^{obl}\leqq 1.$

Definition 4 : For any randomized online edge-coloring algorithm R according to family of input $e_1,e_2,...,e_M$, let $g(b_i)$ is as follows: when b_1 , b_2 ,$\cdots b_i,\cdots b_n$, are edges added,

$g(b_i)=0$: b_i is not colored for adding edge b_i, or

$g(b_i)=1$: v is colored for adding edge b_i.

By linearity of expectation $E[f_R(e_1,e_2,...,e_M)]$ is as follows:

$E[f_R(e_1,e_2,...,e_M)]=E[g(b_1)]+E[g(b_2)]+\cdots+E[g(b_n)]$

from $E[f_R(e_1,e_2,...,e_M)]=g(b_1)+g(b_2)+\cdots+g(b_n)$.

Definition 5 : D_R^{obl} is the supremum of D such that $E[f_R(e_1,e_2,...,e_M)]-D\times$[total of adding edges of $e_1,e_2,...,e_M$] $\geqq b$.

And a constant b must be independent of M, however k need not independence.

Then $D_R^{obl}\leqq C_R^{obl}$ from

$f_o(e_1,e_2,...,e_M)\leqq$[total of adding edges of $e_1,e_2,...,e_M$].

Definition 6 : For oblivious adversary of D_R^{obl} the family of inputs $e_1,e_2,...,e_M$ is as follows:

a constant b exists such that the equation $E[f_R(e_1,e_2,...,e_M)]-D\times$[total of adding edges of $e_1,e_2,...,e_M$] $\geqq b$ is made up when $D=D_R^{obl}$, or

a constant b dose not exist such that the equation $E[f_R(e_1,e_2,...,e_M)]-D\times$[total of adding edges of $e_1,e_2,...,e_M$] $\geqq b$ is made up when $D>D_R^{obl}$.

Color in this paper has linear order as follows : $color_1<color_2<\cdots$.

If new color is used, it must be smallest number that is not yet used.

A set of colors on edges incident from a vertex v is denoted Sv.

3. Online edge-coloring problem

We present two cases about online edge-coloring algorithm. One is that the edge-painter colors all of edges added. Another is that the edged-painter need not color any of edges added. We study these two cases about deterministic online edge-coloring algorithm and randomized online edge-coloring algorithm.

3.1 Deterministic online edge-coloring algorithm

Lemma 1: Let A be a deterministic online edge-coloring algorithm that must try to color all of edges added. Then $C_A^{del}=0$.

Proof We prove it by reductio ad absurdum. We assume that for any families of input $e_1,e_2,...,e_M$, A has a constant b according to $C>0$ and

$f_A(e_1,e_2,...,e_M)-C\times f_O(e_1,e_2,...,e_M)\geqq b.$

Here we think that the family of input is as follow.

1. Repeat k-1 times adding (v_1, w_1).
2. Repeat k-1 times adding (v_2, w_2).

: .

：.

By the assumption, $Sv_1 \cup Sv_2 \cup \ldots$ is a subset of $\{color_1, color_2, \cdots, color_{2k-2}\}$.

A subset of $\{color_1, color_2, \cdots, color_{2k-2}\}$ is finite, since there exist adequate sequence x_1, x_2, and $Sv_{x_1} = Sv_{x_2}$ if this repeating are continued enough.

3. And then this pattern is disconnected. After that, next following edge (v_{x_j}, w_0). is added. Edge-painter cannot color an element of S_{x_j} on the edge (v_{x_j}, w_0). Since one of remaining colors without k-1 elements of S_{x_j} needs to color on this edge.
4. Add (v_{x_2}, w_0).
   Edge-painter cannot color this edge, because adjacent edges to this edge are already colored with all of k colors.
5. Delete (v_{x_2}, w_0).
6. Repeat step 4 and step 5 enough.

For this family of input it can color all of edges if only it dose not make $Sv_{x_1} = Sv_{x_2}$ according to offline edge-coloring algorithm.

By the assumption $C \times f_O(e_1,e_2,\ldots,e_M)$ becomes large enough, but $f_A(e_1,e_2,\ldots,e_M)$ is a constant. Since a constant b, which is satisfied with the assumption, dose not exists. This is the contradiction. □

On the above 2, the continuing enough is the maximum $\binom{2k\text{-}1}{k\text{-}1} \times (k\text{-}1) + 1$, and the maximum number of vertices $|V|$ and $|W|$, for making the families of inputs of the above, are

$$\binom{2k\text{-}1}{k\text{-}1} \times (k\text{-}1) + 1 \text{ and } \left(\binom{2k\text{-}1}{k\text{-}1} \times (k\text{-}1) + 1\right) \times 2 + 1$$

respectively.

Lemma 2: Let A be a deterministic online edge-coloring algorithm that need not color any of edges added. Then $C_A^{det} = 0$.

Proof We prove it by reductio ad absurdum. We assume that for any families of input $e_1,e_2,\ldots,e_M$, A has a constant b according to $C > 0$ and

$f_A(e_1,e_2,\ldots,e_M) - C \times f_O(e_1,e_2,\ldots,e_M) \geqq b$.

Here we think that the family of input is as follow.

1. Repeat adding or deleting edge (v_1, w_1) until it is colored. If it is not colored, then it is finished after repeating enough.
2. Repeat step 1 *k-1* times.
   If it is not finished on the way of repeating, then there are colors (：$color_1, color_2, \cdots, color_{k-1}$) on k-1 multiple edges between v_1 and w_1.
3. Repeat adding or deleting edge (v_2, w_2) until it is colored. If it is not colored, then it is finished after repeating enough. If it is colored, then it is colork.
4. Repeat adding or deleting edge (v_2, w_2) until it is colored. If it is not colored, then it is finished after repeating enough.
5. Repeat step 4 *k-1* times.
   If it is not finished on the way of repeating, then there are colors $(color_1, color_2, \cdots, color_{k-1})$ on k-1 multiple edges between v_2 and w_2.
6. Delete edge (v_2, w_1).
7. Repeat adding or deleting edge (v_1, w_0) until it is colored. If it is not colored, then it is finished after repeating enough. If it is colored, then it is colork.
8. Add (v_2, w_0).
   Edge-painter cannot color this edge, because adjacent edges to this edge are already colored with all of k colors.
9. Delete (v_2, w_0).
10. Repeat step 8 and step 9 enough.

For this family of input, at least repeating edges without coloring can paint adequate colors except edges enough repeated according to offline edge-coloring algorithm.

By the assumption $C \times f_O(e_1,e_2,\ldots,e_M)$ becomes large enough, but $f_A(e_1,e_2,\ldots,e_M)$ is a constant. Since a constant b, which is satisfied with the assumption, dose not exists. This is the contradiction. □

Theorem 4: Let A be a deterministic online edge algorithm. Then Then $C_A^{det} = 0$.

Proof From lemma 1 and lemma 2 it is evident.

3.2 Randomized online edge-coloring algorithm

Theorem 5: Let R be an randomized online edge algorithm.

Then $C_A^{obl} \leqq \frac{X\text{-}1}{X}$, here $X = \binom{k+1}{2} \times ((k\text{-}1)(k+1)+2)$.

Proof For proving this theorem we apply Yao's Minimax Principle to the competitiveness coefficient of randomized online edge-coloring algorithm.

We prove it by reductio ad absurdum.

Let C_R^{obl} be any randomized online edge algorithm R.

We assume that $C_R^{obl} = \frac{X\text{-}1}{X} + \delta$. Here,

$$\frac{X\text{-}1}{X} < C_R^{obl} = \frac{X\text{-}1}{X} + \delta \leqq 1.$$

From Yao's Minimax Principle, there exist a deterministic online edge-coloring algorithm, which is $C_A^P \geqq \frac{X\text{-}1}{X} + \delta$, according to any families of inputs chosen of a probability distribution. Then we think a family of input chosen of a probability distribution as follow,

1. Repeat k-1 times adding (v_1, w_1).
2. Repeat k-1 times adding (v_2, w_2).
   ⋮.
   ⋮.
3. Repeat k-1 times adding (v_{k+1}, w_{k+1}).
   There exist adequate x_1, and x_2, and $Sv_{x_1} = Sv_{x_2}$ if A dose not color all of edges.
4. After that, choose 2 vertices v_{y_1} and v_{y_2} in the same probability from v_1, v_2,..., and add edges (v_{y_1}, w_0) and (v_{y_2}, w_0).
5. Delete all of edges added by steps from 1 to 4.
6. Repeat $\binom{k+1}{2}$ times from step1 to step 5
7. Repeat Y times, which are repeated enough times from step 1 to 6.

Let a period to be from step 1 to 6. Then these edges added are X edges a period.

Then $E[f_A(e_1,e_2,...,e_M)]$ is $(X\text{-}1)Y$, and $E[f_O(e_1,e_2,...,e_M)]$ is XY. Under like these families of inputs chosen of a probability distribution P, any deterministic online edge-coloring algorithm A is satisfied with next equation:

$E[f_A(e_1,e_2,...,e_M)] - C_A^P \times E[f_O(e_1,e_2,...,e_M)] \geqq b$.

Let B be the left side of the above equation.

$$B < (X\text{-}1)Y - (\frac{X\text{-}1}{X} + \delta) \times XY = -\delta \times XY.$$

But there is no constant b that is satisfied with B, because Y is large enough. This is the contradiction of the assumption, hence $C_R^{obl} \leqq \frac{X\text{-}1}{X}$. □

Lemma 3: Let R be a randomized online edge-coloring algorithm that must try to color all of edges added. Then $C_R^{obl}=0$.

Proof We prove it by reductio ad absurdum. We assume that for any families of input $e_1,e_2,...,e_M$, R has a constant b according to $C>0$ and

$E[f_R(e_1,e_2,...,e_M)] - C \times f_O(e_1,e_2,...,e_M) \geqq b$.

Here we think that the family of input is as follow.

1. Repeat k-2 times adding (v_1, w_1).
2. Add 2 times (v_1, w_2).
3. Repeat k-2 times adding (v_2, w_2).
4. Add 2 times (v_2, w_3).
5. Repeat k-2 times adding (v_3, w_3).
6. Delete all of edges between v_1 and w_2, and between v_2 and w_3. Then $Sv_1 = Sv_2 = Sv_3$. Let $color_x$ and $color_y$ to be 2 colors that are not yet used.
7. Add (v_1, w_1) again.
   As edge (v_1, w_1) cannot be colored with the element of Sv_1, this edge is colored with either of $color_x$ or $color_y$ except k-2 element of Sv_1. Let this edge to be colored with $color_x$.
8. And then add (v_1, w_2).
   This edge is colored with the rest of two, i.e. $color_y$.
9. Next. Add (v_2, w_1). This edge is also colored with $color_y$.
10. Add (v_2, w_3).
    This edge is colored with the rest of two, i.e. $color_x$.
11. Delete (v_1, w_1) that was added at step 7.
12. And then add (v_1, w_3).
    This edge cannot be colored, because adjacent edges to this edge are already colored with all of k colors.
13. Delete (v_1, w_3).
14. Repeat step 12 and step 13 enough.

For this family of input, at least repeating edges without coloring can paint adequate colors except edges enough repeated according to offline edge-coloring algorithm.

By the assumption $C \times f_O(e_1,e_2,...,e_M)$ becomes large enough, but $E[f_R(e_1,e_2,...,e_M)]$ is a constant. Since a constant b, which is satisfied with the assumption, dose not exists. This is the contradiction. □

Algorithm 1

Next, we study a randomized online edge-coloring algorithm that often need not color any of edges added. We pay attention to edges going out from each vertices, as following: colorless edges going out from each vertices are surely made more than half, instead the edges which are tried to color should be colored absolutely. All of vertices have matrix $1 \times k$. Let A_v be a matrix of a vertex v.

Procedure algorithm 1 $(G_0, e_1,e_2,...,e_M)$.

(a) For matrix of all of vertices u, we assign an element whose probability is a half, as follows which ① or ②, where k is even number:

① $A_u[1]=0, A_u[2]=0, ..., A_u[\frac{k}{2}]=0, A_u[\frac{k}{2}+1]=1,$

$A_u[\frac{k}{2}+2]=1, ..., A_u[k]=1$, or

② $A_u[1]=1, A_u[2]=1, ..., A_u[\frac{k}{2}]=1, A_u[\frac{k}{2}+1]=0,$

$A_u[\frac{k}{2}+2]=0, ..., A_u[k]=0.$

(b) All of subindex of matrices are keeping to the *unmark*.

(c) *while* adversary inputs do

i . *if* an edge (v, w) is added, then

A : when the minimum subindex of *unmark* of matrix of v is a and the minimum subindex of *unmark* of matrix of w is b, change the subinedx a of matrix of v and the subinedx b of matrix of w from *unmark* to *mark*.

B: *if* $A_v[a]=1$ and $A_w[b]=1$, then an adversary tries to color edge (v, w).

When an adversary tries to color edge (v, w), it is decided to color or not by the color of edges going out from v or w. If he colors, he should use the minimum color of unused one among all of adjacent edges.

else edge (v, w) should not be colored.

ii . elseif delete edge (v, w) existed, then,

when edge is deleted, the subindex marked of A_v and A_w are returned to *unmark*.

Lemma 4: For any edge (v, w) added, when the minimum subindex of *unmark* of matrix of v is a and the minimum subindex of *unmark* of matrix of w is b, the probability of $A_v[a]=A_w[b]=1$ is $\frac{1}{4}$.

Proof Number of matrices assigned of each vertices in $V \cup W$ is $2^{|V|+|W|}$ altogether and each of them has even probability. Then $A_v[a]=1$ are $2^{|V|+|W|-1}$ cases and $A_w[b]=1$ of $2^{|V|+|W|-1}$ cases are $2^{|V|+|W|-2}$ cases. This is that it dose not depend on any input before adding edge (v, w). Therefore the probability of $A_v[a]=A_w[b]=1$ is $\frac{1}{4}$.

Lemma 5: $C^{obl}_{Algorithm1} \leqq \frac{1}{4}$

Proof We prove it by reductio ad absurdum. We assume $C^{obl}_{Algorithm1} > \frac{1}{4}$. For any families of input $e_1,e_2,...,e_M$, there exists a constant b and

$E[f_{Algorithm1}(e_1,e_2,...,e_M)] - C^{obl}_{Algorithm1} \times f_O(e_1,e_2,...,e_M) \geqq b$ -------(*).

For large enough L, we think that there is a family of input which is consisted of L elements of $(a, \bar{a}, a, \bar{a}, ..., a, \bar{a})$. Then $f_O(a,\bar{a},a,\bar{a},...,a,\bar{a})=\frac{L}{2}$.

We pay attention to a case that under this family of input any edge $a=(v, w)$ is added and is colored. Let a be the minimum subindex of *unmark* of matrix of v and b be the minimum subindex of *unmark* of matrix of w. Only if $A_v[a]=1$ and $A_w[b]=1$, then edge-painter tries to color edge (v, w). From lemma 4, this probability is $\frac{1}{4}$.

Thus, $E[f_{Algorithm1}(a,\bar{a},a,\bar{a},...,a,\bar{a})]$

$=E[g(a)]+E[g(a)]+\cdots+E[g(a)]=\frac{1}{4}+\frac{1}{4}+\cdots+\frac{1}{4}$

$=\frac{1}{4}\times\frac{L}{2}$.

So, the left side of (*)$=\frac{1}{4}\times\frac{L}{2}-C^{obl}_{Algorithm1}\times\frac{L}{2}$

$=(\frac{1}{4}-C^{obl}_{Algorithm1})\times\frac{L}{2}$.

Then, $\frac{1}{4} - C^{obl}_{Algorithm1} < 0$ and L is large enough. Since a constant b, which is satisfied with the (*), dose not exists. This is the contradiction. □

Theorem 6: $C^{obl}_{Algorithm1}=\frac{1}{4}$

Proof Let (v, w) be an edge to try to color. Let a be the minimum subindex of unmark of matrix of v and b be the minimum subindex of unmark of matrix of w. Then it was $A_v[a]=A_w[b]=1$. The number of edges colored, (which are going out from v or w just before adding (v, w)), is at most $\frac{k}{2}$-1 respectively even if any matrices are assigned to each vertices. Since the total of edges colored which are going out from both vertices is at most $(\frac{k}{2}-1)\times 2=k-2$. So the edge (v, w) can be colored. □

Theorem 7: $C^{ada}_{Algorithm1}=0$

Proof We think a family of input as follows:

1. repeat to add (v, w) until not to be able to colored.

From algorithm 1, an edge without color should appear. An $(v, w)_1$ denotes the edge.

2. repeat to add or to delete $(v, w)_1$ enough.

When these family of input are $e_1,e_2,...,e_M$, $E[f_{Algorithm1}(e_1,e_2,...,e_M)]$ is a constant. But because offline edge-coloring algorithm can be colored the only

edges which are repeated at 2-stage at least, $E[f_{Algorithm1}(e_1,e_2,...,e_M)]$ can be large enough.
Since $C^{ada}{}_{Algorithm1} \leqq 0$. □

Algorithm 2

An aim of algorithm 2 is to cut down edges which are not able to colored as edge-painter tries to color edges added with probability $\frac{1}{2}$.

Procedure algorithm 2(G_0, e_1,e_2,...,e_M).
(a) While edge (v, w) is added do
i .The edge-painter tries to color the edge (v, w) with probability $\frac{1}{2}$.

When the edge (v, w) is tried to color, the edge is actively decided to color or not to color by color of edges already going out from v and w. When the edge is colored, it should be colored with minimum color which is not used among of all adjacent edges.

Lemma 6: $C^{obl}{}_{Algorithm2} \leqq \frac{1}{2}$

Proof We prove it by reductio ad absurdum. We assume $C^{obl}{}_{Algorithm2} > \frac{1}{2}$. For any families of input $e_1,e_2,...,e_M$, there exists a constant b and
$E[f_{Algorithm2}(e_1,e_2,...,e_M)] - C^{obl}{}_{Algorithm2} \times f_O(e_1,e_2,...,e_M) \geqq b$ ------- (*).
For large enough L, we think that there is a family of input which is consisted of L elements of ($a, \bar{a}, a, \bar{a}$, ..., a, $\bar{a}$). Then $f_O(a,\bar{a}, a,\bar{a}, ..., a,\bar{a},) = \frac{L}{2}$.

The edge-painter tries to color any edge a=(v, w) added. When the probability $\frac{1}{2}$ is to try to color, the edge is actively colored. This is also for any edges added.
Thus, $E[f_{Algorithm2}(a,\bar{a}, a,\bar{a}, a,\cdots a,\bar{a},)]$
$=E[g(a)] + E[g(a)] + \cdots + E[g(a)]$
$= \frac{1}{2} + \frac{1}{2} + \cdots + \frac{1}{2} = \frac{1}{2} \times \frac{L}{2}$.

So, the left side of (*) $= \frac{1}{2} \times \frac{L}{2} - C^{obl}{}_{Algorithm2} \times \frac{L}{2}$

$= (\frac{1}{2} - C^{obl}{}_{Algorithm2}) \times \frac{L}{2}$.

Then, $\frac{1}{2} - C^{obl}{}_{Algorithm2} < 0$ and L is large enough. Since a constant b, which is satisfied with the (*), dose not exists. This is the contradiction. □

We look for the infimum of C_R^{obl} with $C_R^{obl} \leqq C_R^{obl}$ from definition.

Lemma 7: For a randomized online edge-coloring algorithm R trying to color with a constant provability, there exist adequate i and M in a family of oblivious adversary D_R^{obl}. This is consisted of M elements of (a_1, a_2, ..., a_i, $\bar{a}_i$, a_i, $\bar{a}_i$, ⋯ , a_i, $\bar{a}_i$,). Let edges which are added be b_1, b_2,..., a_i, a_i,..., there exist a minimum $E[g(a_i)]$ among $E[g(b_1)]$, $E[g(b_2)]$,..., $E[g(a_i)]$, $E[g(a_i)]$,....Here $\bar{a}_i$ means deleting a_i.

Proof We assume that there is no family of D_R^{obl} such that. Let a family of D_R^{obl} be $e_1,e_2,...,e_N$. And then let edges added be b_1, b_2,..., b_L. Let $E[g(b_j)]$ be a minimum among $E[g(b_1)]$, $E[g(b_2)]$,..., $E[g(b_L)]$, where $1 \leqq j \leqq L$. Then we think that a family of input which L edges are added are as follows:
($e_1,e_2,...,b_j,\bar{b}_j, b_j,\bar{b}_j, ...$) . From definition of D_R^{obl}, there exist a constant b that satisfies next equation, $E[f_R(e_1,e_2,...,b_j,\bar{b}_j, bj,\bar{b}_j, ...)] - D \times$ [total of adding edges of $e_1,e_2,...,b_j,\bar{b}_j, b_j,\bar{b}_j, ...] \geqq b$-------(**), where $D=D_R^{obl}$.
$E[f_R(e_1,e_2,...,b_j,\bar{b}_j, b_j,\bar{b}_j, ...)] = E[g(b_1)] + E[g(b_2)] + \cdots + E[g(b_j)] + E[g(\bar{b}_j] + \cdots \leqq E[g(b_1)] + E[g(b_2)] + \cdots + E[g(b_n)] = E[f_R(e_1,e_2,...,e_N)]$
Since at $D > D_R^{obl}$ a constant b, which is satisfied with the (**), dose not exists. Therefore ($e_1,e_2,...,b_j,\bar{b}_j, b_j,\bar{b}_j, ...$) is one of a family of D_R^{obl}, but this is contradictory to the assumption. □

We call an adding of a family of input (a_1, a_2, ..., a_i, (v, w)) a family ρ_1 of input. We make a family ρ_2 of input from a family ρ_1 of input. The ρ_2 is as follows ; let degree of v and of w just before adding (v, w) of a family ρ_1 of input be x and y respectively. Edges that go out from w to some elements of V are added k-1-y times after edges that go out from v to some elements of W are added k-1-x times following (a_1, a_2, ..., a_i), and after that (v, w) is added.

Lemma 8: For a randomized online edge-coloring algorithm R trying to color with a constant provability, the next holds good.
($E[g((v, w))]$ according to (v, w) added with ρ_1)$\geqq$($E[g((v, w))]$ according to (v, w) added with ρ_2.)

Proof When $x=y=k$-1, an equal sign holds good.
When $x \leqq k$-2 or $y \leqq k$-2, the algorithm R tries to color (v, w) with a constant provability. But number of color may be increased because edges going out from v and w just before adding (v, w) of a family ρ_2 of input increases more than edges going out from v and w just before

adding (v, w) of a family ρ_1 of input. So the possibility of not being able to color increases even if the R tries to color according to (v, w) added with ρ_2. Therefore this case also holds good. □

Lemma 9 : $D^{obl}_{Algorithm2} > \frac{1}{4}$

Proof From lemma 7 and lemma 8 one of a family of oblivious adversary of $C^{obl}_{Algorithm2}$ is L edges added like as $(a_1, a_2, \cdots, a_i, (v, w), \overline{(v, w)}, (v, w), \overline{(v, w)}, \cdots,)$. (After some adequate inputs are continued, same edges add or delete alternatively.) Let only edges added from those above be $(b_1, b_2, \cdots, (v, w), (v, w), \cdots)$. Then there are $E[g((v, w))]$ that is minimum of $E[g(b_1)]$, $E[g(b_2)]$, $\cdots$, $E[g((v, w))]$, $E[g((v, w))]$, and degree of both v and w just before (v, w) added are k-1. If $E[g((v, w))] > \frac{1}{4}$, then $E[f_{aligorithm2}(a_1, a_2, \cdots, a_i, (v, w), \overline{(v, w)}, (v, w), \overline{(v, w)}, \cdots,)] = E[g(b_1)] + E[g(b_2)] + \cdots + E[g((v,w))] + E[g((v, w))] + \cdots > \frac{1}{4} \times L$. Therefore the lemma holds good such that we must prove $E[g((v, w))] > \frac{1}{4}$.

Let sets of color, which colored edges going out from v and w just before (v, w) added, be S_v and S_w.

When $|S_v| + |S_w| \geqq k$, it is possible not to color (v, w).

When $|S_v \cup S_w| \leqq k$-1, it is possible to color (v, w) even if edges are colored with any color.

When algorithm2 colors (v, w) and $|S_v \cup S_w| \leqq k$-1, $g((v, w))=1$. The probability of $|S_v \cup S_w| \leqq k$-1 is greater than $|S_v \cup S_w| \leqq k$-1.

(The probability of algorithm2 to color (v, w)) × (the probability of $|S_v \cup S_w| \leqq k$-1) ≧ (the probability of algorithm2 to color (v, w)) × (the probability of $|S_v \cup S_w| \leqq k$-1) $= \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$. Therefore $g((v, w))=1$, then the proba bility is greater than .

Since $E[g((v, w))] > \frac{1}{4}$. □

Theorem 8: $< C^{obl}_{Algorithm2} \leqq \frac{1}{2}$

Proof From lemma 9

and $C^{obl}_{Algorithm} \geqq D^{obl}_{Algorithm2}$,

$C^{obl}_{Algorithm} \geqq D^{obl}_{Algorithm2} > \frac{1}{4}$.

And from lemma 8, $C^{obl}_{Algorithm} \leqq \frac{1}{2}$.

Theorem 9: $C^{ada}_{Algorithm2} = 0$

Proof We think that the family of input is as follow.

1. Repeat adding or deleting edge (v_1, w_1) until it is colored.
2. 2. Repeat step 1 k-1 times.

Then there are colors (: $color_1$, $color_2$, $\cdots\cdots$, $color_{k-1}$) on k-1 multiple edges between v_1 and w_1.

3. Repeat adding or deleting edge (v_2, w_1) until it is colored, then it is $color_k$.
4. Repeat adding or deleting edge (v_2, w_2) until it is colored.
5. Repeat step 4 k-1 times, then there are colors (: $color_1$, $color_2$, $\cdots\cdots$, $color_{k-1}$) on k-1 multiple edges between v_2 and w_2.
6. Delete edge (v_2, w_1).
7. Repeat adding or deleting edge (v_1, w_0) until it is colored, then it is $color_k$.
8. Add (v_2, w_0).

Edge-painter cannot color this edge, because adjacent edges to this edge are already colored with all of k colors.

9. Delete (v_2, w_0).
10. Repeat step 8 and step 9 enough.

For this family of input, at least repeating edges without coloring can paint adequate colors except edges enough repeated according to offline edge-coloring algorithm.

$E[f_O(e_1,e_2,...,e_M)]$ becomes large enough, but $E[f_{Algorithm2}(e_1,e_2,...,e_M)]$is a constant.

Since $C^{ada}_{Algorithm2} \leqq 0$. □

Here, number of bits of random numbers in the algorithm 2 is equal to total of edges added.

4. Conclusion

We have proved that a competitiveness coefficient C^{det}_A of arbitrary deterministic online edge-coloring algorithm is 0, and present a randomized online-coloring algorithm that the competitiveness coefficient C^{obl}_R is $\frac{1}{4}$ and a randomized online-coloring algorithm that the competitiveness coefficient is $\frac{1}{4} < C^{obl}_R \leqq \frac{1}{2}$ for oblivious adversary. Next study is the presentation of a randomized online-coloring algorithm that the competitiveness coefficient C^{obl}_R is more lager probability.

References

[1] Robin J. Wilson and John J. Watkins, " Graphs : An Introductory Approach ", John Wiley & Sons Inc, 1990.

[2] Rajeev Motwani, Prabhakar Raghavan, " Randomized Algorithms ", Cambridge University Press, 1995.

[3] Amotz Bar-Noy, Rajeev Motwani, Joseph Naor, " The greedy algorithm is optimal for online edge coloring ", Information Processing Letters 44, 251-253, 1992.