Scientia Iranica D (2021) 28(3), 1493–1496

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

http://scientiairanica.sharif.edu

Tight online conflict-free coloring of intervals

M.A. Abam*

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.

Received 28 October 2019; received in revised form 6 August 2020; accepted 2 November 2020

KEYWORDS

Frequency assignment;
Conflict-free coloring;
Intervals;
On-line algorithms;
Computational geometry.

Abstract. This study revisited the problem of online conflict-free coloring of intervals on a line, where each newly inserted interval must be assigned a color upon insertion such that the coloring remains conflict-free, i.e., for each point p in the union of the current intervals, there must be an interval I with a unique color among all intervals covering p. The best-known algorithm uses $O(\log^3 n)$ colors, where n is the number of current intervals. A simple greedy algorithm was presented that used only $O(\log n)$ colors. Therefore, the open problem raised in [Abam, M.A., Rezaei Seraji, M.J., and Shadravan, M. "Online conflict-free coloring of intervals", *Journal of Scientia Iranica*, **21**(6), pp. 2138–2141 (2014).] was resolved.

1. Introduction

Background. In a cellular network, each base station has a coverage area (usually denoted by a disk) and can only give services to clients being inside its coverage area. In general, several base stations may cover a client (i.e., their coverage areas have a common point). This may lead to interference of signals for the client. Thus, one would like to assign frequencies to the base stations such that for each client within the coverage area of at least one base station, there is a base station with a unique frequency covering the client. The main objective of this course is to do this using a few distinct frequencies. Even et al. [1] modeled this problem by the concept of *conflict-free coloring*, defined next.

The Conflict-Free coloring (CF-coloring) of a set of n objects with respect to a (possibly infinite) family of ranges is a coloring of objects with the following properties: for any range $r \in R$ intersecting at least one object, there is an object $o \in S(r)$ with a unique color in $S(r)$, where $S(r)$ is the set of objects intersecting r. It is clear that a conflict-free coloring always exists as we can use n different colors. However, one would like to use a few colors for this purpose. Note that if we take S to be a set of disks (the coverage area of base stations) and $\mathcal{R}$ to be the set of all points in $\mathcal{R}^2$ (clients), then we get the frequency-assignment problem discussed above. In this paper, a case where objects are intervals in $\mathbb{R}^1$ and ranges are points is only considered.

Related work. The offline CF-coloring where all objects are given in advance has been studied a lot in the last two decades. Even et al. [1] were the first to present CF-coloring of points with respect to disks using $O(\log n)$ colors, which is tight in the worst case. Then, Har-Peled and Smorodinsky [2] extended these results by considering other range spaces like rectangles and pseudo disks. For more recent work on the CF-coloring problem, see [3–9].

Chen et al. [10] were the first to study the online version of CF-coloring of points with respect to intervals: Upon the arrival of a point, a color is assigned to it and the color cannot be changed since then. The coloring must remain conflict-free at

*. *Tel.: +98 21 6616 6653; Fax: +98 21 6601 9246*
E-mail address: abam@sharif.edu

doi: 10.24200/sci.2020.54600.3826

all times. They presented a deterministic algorithm using $O(\log^2 n)$ colors and a randomized algorithm using $O(\log n)$ colors with a high probability for a set of n points. The best-known lower bound for both randomized and deterministic algorithms, also valid for the offline case, is $\Omega(\log n)$ colors [11]. They also showed that the known simple greedy algorithm that held the unique maximum invariant (see the next section) might require $\Omega(\sqrt{n})$ colors and obtain an upper bound on the number of colors used by the algorithm as an open problem– they conjectured the bound to be close to $\Omega(\sqrt{n})$. For other interesting variants of the online CF-coloring problem, see [12] and references therein. Abam et al. [13] considered the online CF-coloring of intervals with respect to points. They presented a simple greedy algorithm that uses $O(\sqrt{n})$ colors. Then, they gave a more sophisticated algorithm that used $O(\log^3 n)$ colors (big O). They left out an open problem whether there was a CF-coloring algorithm using $o(\log^3 n)$ colors (small o). It is worth mentioning that the offline CF-coloring of intervals with respect to points can be done with at most 3 colors [1].

Recently, de Berg and Markovic [14] studied the dynamic CF-coloring of rectangles (and some other objects) under insertions and deletions. For n arbitrary rectangles whose coordinates come from a fixed universe of size N, they use $O(\log^2 N \log^2 n)$ colors, at the cost of only $O(\log n)$ re-colorings per insertion and deletion. The fully dynamic and kinetic versions of intervals were considered in [15] as well.

Problem definition. We revisit the problem of online CF-coloring of intervals on a line. Here, intervals are arriving one by one and upon the arrival of an interval, we should assign a color to this interval and its color cannot be changed later. At any time, the coloring must remain conflict-free, i.e., for each point p in the union of the current intervals, there must be an interval with a unique color among all intervals covering p.

Our results. This paper affirmatively answers the open problem whether there is a CF-coloring algorithm for intervals with respect to points using $o(\log^3 n)$ colors. It is demonstrated that the known simple greedy algorithm that holds the unique-maximum invariant (see the next section) only uses $O(\log n)$ colors. There is a simple proof showing that any online CF-coloring of intervals with respect to points needs $\Omega(\log n)$ colors. Therefore, the problem of online CF-coloring of intervals is settled down.

Paper organization. The paper is organized as follows. Section 2 analyzes the known simple greedy algorithm and shows that it uses only $O(\log n)$ colors. This paper will close with a few concluding remarks and solutions to open problems in Section 3.

2. Simple CF-coloring using $O(\log n)$ colors

Our algorithm is similar to the simple greedy algorithm given in [13] that holds the unique-maximum invariant– this greedy algorithm indeed is a well-known algorithm in the area of CF-coloring and researchers in this field are interested in knowing how well this algorithm works (for instance, see [10]). Suppose that $\mathcal{I}$ is a set of n intervals arriving through time one by one and suppose that for each point $p \in \mathbb{R}^1$, $\mathcal{I}(p)$ is the set of all intervals containing p at the current time. We denote colors by non-negative integer numbers and denote the color of an interval I by $c(I)$. The algorithm guarantees to hold the unique-maximum invariant (UM invariant for short): $\mathrm{mc}(p) = \max_{I \in \mathcal{I}(p)} c(I)$ is unique in the multiset $\{c(I) : I \in \mathcal{I}(p)\}$ for all $p \in \mathbb{R}^1$. If this holds, the coloring, of course, is a CF-coloring as the interval with the maximum color among intervals containing p has a unique color. Next, details of the algorithm will be discussed.

2.1. Coloring algorithm

The maximum color used so far in a variable m is maintained; at the beginning, $m = 1$. Upon the arrival of an interval I, the set $S_I \subset \{1, \cdots, m\}$ of *forbidden colors* for I is computed first in a sensethat if one of them is assigned to I and then, the UM invariant does not hold anymore. Then, a color is assigned to I as follows. If $S_I = \{1, \cdots, m\}$, m is increased by 1 and $c(I)$ is set to be m. Otherwise, $c(I)$ is set to be the smallest unforbidden color from the set $\{1, \cdots, m\}$. Algorithm 1 shows a simple implementation of the above algorithm.

2.2. Analysis

Imagine we know all n intervals in advance (see Figure 1(a)). Let $p_1, p_2, ..., p_m$ be the list of the distinct interval endpoints, sorted from left to right ($m \leq 2n$ as some endpoints may coincide). Consider the partitioning of $\mathbb{R}^1$ into the elementary intervals $(\infty : p_1), [p_1 : p_1], (p_1 : p_2), [p_2 : p_2], \cdots, (p_{m1} : p_m), [p_m : p_m], (p_m : +\infty)$ (see Figure 1(b)). The list of elementary intervals consists of open intervals between two consecutive endpoints p_i and p_{i+1}, alternated with closed intervals consisting of a single endpoint. The reason that we treat the points p_i as intervals is that the

```
j=0
repeat
    j = j + 1
    c(I) = j
until UM invariant holds
```

Algorithm 1. Online-CF-coloring (I).

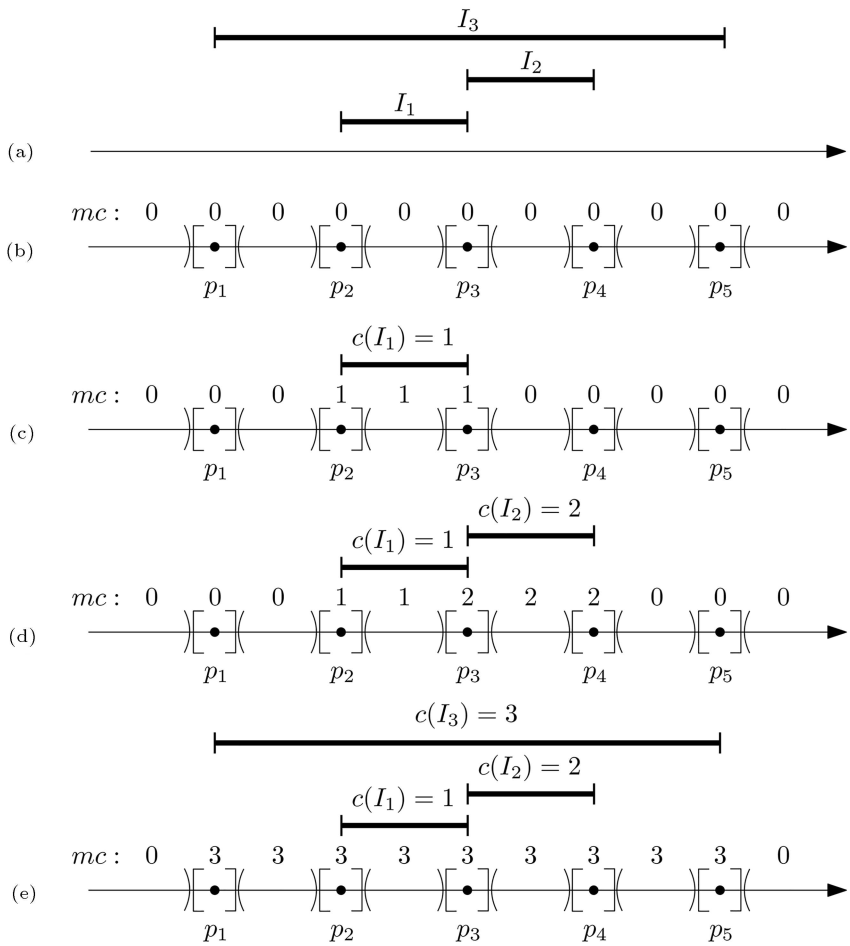

Figure 1. (a) All three input intervals in advance, (b) elementary intervals and their mc when no interval arrives, and (c)-(e) the status when the intervals I_1, I_2, and I_3 arrive one by one in the given order. Upon the arrival of I_3, $c(I_3)$ becomes 3 as the first color holding the UM variant is 3. Therefore, $\mathrm{mc}(e) = 3$ for all elementary intervals e except the rightmost and leftmost elementary intervals.

set of intervals covering p_i is not necessarily the same as the set of intervals covering p where p can be any point close to p_i. For an elementary interval e, let $\mathrm{mc}(e)$ be the maximum color covering e. At the beginning when no interval has arrived, $\mathrm{mc}(e) = 0$ for all (at most) $4n+1$ elementary intervals. It is clear that $\mathrm{mc}(e)$ is not decreasing through time for each elementary interval e (see Figure 1(c)–(e)). Let $\mathcal{I}_{i,j}$ be the set of the input intervals whose color is at least i and at most j. We will show that $|\mathcal{I}_{10,+\infty}| \leq n/2$. Let I be a member of $\mathcal{I}_{10,+\infty}$. Upon the arrival of I, we assign a color greater than 9 to it. This implies that assigning any color less than 10 would not hold the UM invariant. Therefore, there must be 9 elementary intervals $\{e_1, \cdots, e_9\}$ such that $\{\mathrm{mc}(e_1), \cdots, \mathrm{mc}(e_9)\} = \{1, 2, \cdots, 9\}$. After coloring I, all $\mathrm{mc}(e_i)$ $(i = 1, \cdots, 9)$ become the color of I which is at least 10. Then, it can be imagined that all these 9 elementary intervals with mc less than 10 are killed after the insertion of I—note that these elementary intervals are still alive, but their mc is at least 10. Therefore, each member of $\mathcal{I}_{10,+\infty}$ kills 9 elementary intervals with mc less than 10 and since we have at most $4n + 1$ elementary intervals, we can simply conclude that 9. $|\mathcal{I}_{10,+\infty}| \leq 4n + 1$; yielding $|\mathcal{I}_{10,+\infty}| \leq n/2$ for $n \geq 2$. Similarly, if we apply the same argument to the set $\mathcal{I} - \mathcal{I}_{1,9}$, we can conclude that $|\mathcal{I}_{19,+\infty}| \leq n/4$—note that $\mathcal{I} - \mathcal{I}_{1,9} = \mathcal{I}_{10,+\infty}$ has size of at most $n/2$, as shown above. Of course. This can be extended to $|\mathcal{I}_{9i+1,+\infty}| \leq n/2^i$. For i greater than $\log_2 n$, we know that $n/2^i < 1$. Therefore, $\mathcal{I}_{9i+1,+\infty}$ is empty for $i > \log_2 n$. This shows that the greedy algorithm uses at most $9 \log_2 n = O(\log n)$ colors in total.

Theorem 1. *There is an online CF-coloring algorithm for a set of n intervals in the online model that uses $O(\log n)$ colors.*

As mentioned in [13], although there is a CF-coloring of intervals with respect to points using 3 colors in the offline model, it can be easily shown [11] that in the online model, any such CF-coloring must use $\Omega(\log n)$ colors in the worst case. Consider

$\mathcal{I} = \{I_1, \cdots, I_n\}$ where $I_j = [1, p_j]$ and $p_j = j$. Suppose that the intervals in $\mathcal{I}$ arrive in the increasing order of their indices. In this scenario, it is easy to see that CF-coloring of the intervals is equivalent to CF-coloring of points p_i with respect to intervals, needing $\Omega(\log n)$ colors [1]. Indeed, if we assign the color of each interval I_r to its right endpoint (i.e. p_r), we can show that among points p_k at any interval $[i, j]$, one has a unique color. This lower bound on the number of colors demonstrates that our algorithm is tight in the worst case.

3. Conclusion

This study revisited the problem of online conflict-free coloring of intervals on a line, where each newly inserted interval must be assigned a color upon insertion such that the coloring would remain conflict-free. Subsequently, this paper affirmatively answered the open problem whether there was a CF-coloring algorithm for intervals with respect to points using $o(\log^3 n)$ colors. Indeed, it was shown that the known simple greedy algorithm that maintained the maximum color unique only used $O(\log n)$ colors. This matched the known lower bound $\Omega(\log n)$ for the number of colors. This problem could be simply extended to the online CF-coloring of rectangles with respect to points. As an interesting special case, rectangles with one side on the x-axis can be studied first.

Acknowledgements

We would like to thank Maryam Chahian Borujeni for her partial work on the problem.

References

1. Even, G., Lotker, Z., Ron, D., and Smorodinsky, S. "Conflict-freee colorings of simple geometric regions with applications to frequency assignment in cellular networks", *SIAM J. Comput.*, **33**(1), pp. 94–136 (2003).
2. Har-Peled, S. and Smorodinsky S. "Conflict-free coloring of points and simple regions in the plane", *Discr. Comput. Geom.*, **34**(1), pp. 47–70 (2005).
3. Ashok, P., Dudeja, A., Kolay, S., and Saurabh, S. "Exact and fixed parameter tractable algorithms for max-conflict-free coloring in hypergraphs", *SIAM J. Discrete Math.*, **32**(2), pp. 1189–1208 (2018).
4. Cheilaris, P., Gargano, L., Rescigno, A.A., and Smorodinsky, S. "Strong conflict-free coloring for intervals", *Algorithmica*, **70**(4), pp. 732–749 (2014).
5. Cui, Z. and Hu, Z. "On variants of conflict-free-coloring for hypergraphs", *Discrete Applied Mathematics*, **220**, pp. 46–54 (2017).
6. Fekete, S.P. and Keldenich, P. "Conflict-free coloring of intersection graphs", *International Journal of Computational Geometry and Applications*, **28**(3), pp. 289–307 (2018).
7. Gargano, L. and Rescigno, A.A. "Complexity of conflict-free colorings of graphs", *Theor. Comput. Sci.*, **566**, pp. 39–49 (2015).
8. Keller, C. and Smorodinsky, S. "Conflict-free coloring of intersection graphs of geometric objects", In *Proc. SODA*, pp. 2397–2411 (2018).
9. Reddy, I.V. "Parameterized algorithms for conflict-free colorings of graphs", *Theoretical Computer Science*, **745**, pp. 53–62 (2018).
10. Chen, K., Fiat, A., Kaplan, H., Levy, M., Matousek, J., Mossel, E., Pach, J., Sharir, M., Smorodinsky, S., Wagner, U., and Welzl, E. "Online conflict-free coloring for intervals", *SIAM J. Comput.*, **36**(5), pp. 1342–1359 (2007).
11. Pach, J. and Toth, G. "Conflict-free colorings", *Discrete and Computational Geometry, The Goodman-Pollack Festschrift*, B. Aronov, S. Basu, J. Pach and M. Sharir, Editors, Springer Verlag, Heidelberg, pp. 665–671 (2003).
12. Bar-Noy, A., Cheilaris, P., and Smorodinsky, S. "Conflict-free coloring for intervals: from offline to online", In *Proc. ACM Sympos. on Parallelism Algorithms Architectures*, pp. 128–137 (2006).
13. Abam, M.A., Rezaei Seraji, M.J., and Shadravan, M. "Online conflict-free coloring of intervals", *journal of scientia iranica, Journal of Scientia Iranica*, **21**(6), pp. 2138–2141 (2014).
14. De Berg, M. and Markovic, A. "Dynamic conflict-free colorings in the plane", *Comput. Geom.*, **78**, pp. 61–73 (2019).
15. De Berg, M., Leijsen, T., Markovic, A., van Renssen, A., Roeloffzen, M., and Woeginger, G.J. "Fully-dynamic and kinetic conflict-free coloring of intervals with respect to points", *Int. J. Comput. Geometry Appl.*, **29**(1), pp. 49–72 (2019).

Biography

Mohammad Ali Abam is an Assistant Professor at the Department of Computer Engineering at Sharif University of Technology in Tehran, Iran and has been working in the Sharif faculty since 2011. He received his BS and MS degrees in Computer Engineering from the Sharif University of Technology respectively in 1999 and 2001, and PhD degree in Computer Sciences from the Eindhoven University in the Netherlands in 2007. He was a Postdoc Researcher at Aarhus University in Denmark and Dortmund University in Germany from 2008 till 2010.