Arbitrary-Oriented Color Spanning Region for Line Segments

Sukanya Maji(✉) and Sanjib Sadhu

Department of CSE, National Institite of Technology, Durgapur, India
sm.20cs1102@phd.nitdgp.ac.in, sanjib.sadhu@cse.nitdgp.ac.in

Abstract. Given a set of colored geometric objects, a color spanning region of a desired shape is a region (of that shape) that contains at least one object of each color. Here, the objective is to optimize a specific parameter of the region as mentioned in the problem definition. In this paper, we study the optimal color spanning region recognition problem of different shapes for a given set $\mathcal{L}$ of n colored line segment objects in $\mathbb{R}^2$, where each segment is associated with any one of the m colors, namely $\{1, 2, \ldots, m\}$, where $3 \leq m < n$. These are (i) an arbitrary-oriented color spanning strip of minimum width, (ii) two congruent arbitrary-oriented minimum width color spanning strips which contain disjoint subset of the members in $\mathcal{L}$, (iii) two congruent arbitrary-oriented strips of minimum width, such that their union is color spanning, and (iv) an arbitrary-oriented color spanning rectangle of minimum area. The time complexities of the proposed algorithms for these problems are: (i) $O(n^2 \log n)$, (ii) $O(n^4 \log n)$, (iii) $O(n^4 m \log m)$, and (iv) $O(n^3 m)$. Better algorithm with reduced time complexities can be achieved for problems (ii) and (iii) if some restrictions are imposed on the relative orientation of the outputs. Each of these problems needs linear space.

Keywords: Color spanning region recognition · Geometric duality · Line sweep

1 Introduction

Given a set $\mathcal{L}$ of n line segments, each segment is attached with one of the m colors ($3 \leq m < n$), the objective of this paper is to study the problem of recognizing color spanning region of different shapes minimizing a specified parameter of the region depending on the problem requirement. Here the objects in $\mathcal{L}$ may be viewed as the facilities (e.g. hospitals, post-offices, schools etc.), available in a city, and the objective is to locate a region of minimum area where at least one facility of each type is available. The facilities may be points, line segments, convex polygons, etc. The desired region may be a strip, disk, rectangle, etc. The problem is well studied in the literature starting from the work of [1], and has found a lot of applications in facility location problem [1], pattern recognition [3], database queries [12], etc.

A. Bagchi and R. Muthu (Eds.): CALDAM 2023, LNCS 13947, pp. 71–86, 2023.
https://doi.org/10.1007/978-3-031-25211-2_6

Related Work: The color spanning problem was studied by Abellanas et al. [1], where they computed a color spanning axis-parallel rectangle among a set of n colored points with m colors in $O(n(n-m)\log^2 m)$. Huttenlocher et al. [9] proposed algorithm for computing the smallest color spanning circle for a given set of n points with m colors in $O(mn\log n)$ time. The smallest color spanning strip and rectangle of arbitrary orientation for a given set of points can be computed in $O(n^2 \log n)$ and $O(n^3 \log m)$ time [6], respectively. The color spanning axis-parallel square and equilateral triangle can be determined in time $O(n\log^2 n)$ [10] and $O(n\log n)$ [8], respectively. Acharyya et al. [2] identified the smallest color spanning axis-parallel square, rectangle and circle for a colored point set around a given query point. Bae [4] computed the minimum width color spanning axis-parallel rectangular annulus for a set of points in $O((n-m)^3 n\log n)$.

Most of the research works on color spanning problem deals with the input facilities as a point set. However, in real application, it is not always reasonable to represent each facility by point only. This leads to studying the problem of recognizing a color spanning region of optimum size among a set of convex objects. For simplicity, we start research in this direction with colored line segments as the facilities. Note that, the method of solving color spanning region with point set facilities cannot be extended in a straightforward manner to handle this problem with line segment facilities. Huttenlocher et al. [9] computed the smallest color spanning axis-parallel square and disk with the line segments as facilities, in $O(n^2 \log n)$ and $O(n^2\alpha(n)\log n)$ time, respectively.

Another related problem is the k-center problem, where a given set of geometric objects need to be covered by k congruent disks or squares of minimum size. The corresponding color spanning version is finding k congruent color spanning regions among a set of colored objects as facilities to place k demand points. Here, the concept is that the i^{th} facility, denoted by r_i, can support at most $f(r_i)$ demand points (centers of the color spanning regions, each of equal size). We start studying this variation of the problem with $k = 2$ and $f(r_i) = 1$ for each facility in $\{r_1, r_2, \ldots, r_n\}$.

Depending on the problem instance, sometimes a single color spanning region may be more costly (measured in terms of width or area of the region) than the k congruent regions whose union is a color spanning. This motivates us to study further the union color spanning strips problem for a set of line segments $\mathcal{L}$. For simplicity, we have considered $k = 2$ strips in this paper.

Our Contributions: Given a set of n colored line segment objects with m different colors ($3 \leq m < n$) in $\mathbb{R}^2$, we propose algorithms for computing color spanning (CS) arbitrary-oriented (i) a pair of strips of minimum width, and (ii) a rectangle of minimum area. The specific problems that are studied in this paper, are listed below in the Table 1 along with the time complexities of the proposed algorithms. The space complexity of all these problems is $O(n)$.

Table 1. The result of arbitrarily oriented color spanning object(s)

Problems on color spanning regions for a set $\mathcal{L}$ of line segments in $\mathbb{R}^2$	Segments covered ($\mathcal{L}', \mathcal{L}'' \subseteq \mathcal{L}$) by strip(s)/rectangle	Minimizes	Time complexity
A single strip	$\mathcal{L}'$ is color spanning (CS)	Strip width	$O(n^2 \log n)$
Two congruent strips	$\mathcal{L}'$ and $\mathcal{L}''$ are CS ($\mathcal{L}' \cap \mathcal{L}'' = \phi$)	Strip width	$O(n^4 \log n)$
Two congruent parallel strips	$\mathcal{L}'$ and $\mathcal{L}''$ are CS ($\mathcal{L}' \cap \mathcal{L}'' = \phi$)	Strip width	$O(n^3)$
Union color spanning by two strips	$\mathcal{L}' \cup \mathcal{L}''$ is CS ($\mathcal{L}' \cap \mathcal{L}'' = \phi$)	Strip width	$O(n^4 m \log m)$
Union color spanning by two parallel strips	$\mathcal{L}' \cup \mathcal{L}''$ is CS ($\mathcal{L}' \cap \mathcal{L}'' = \phi$)	Strip width	$O(n^3 \log m)$
A rectangle ($\mathcal{R}$)	$\mathcal{L}'$ is CS	Area of $\mathcal{R}$	$O(n^3 m)$

2 Preliminaries and Notations

We use $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ to denote the n input line segment facilities. The subset of the segments in $\mathcal{L}$ with color $i \in \{1, 2, \ldots, m\}$ is denoted by $\mathcal{L}_i$. We use $x(p)$ and $y(p)$ to denote the x- and y-coordinate of the point p, respectively. A line passing through any two points p and q is denoted by $\ell(p, q)$. A line segment ℓ in $\mathbb{R}^2$ is said to be *covered* by a region if every point on ℓ lies inside or on the boundary of that region. A segment with its two endpoints p and q is denoted by $[p, q]$.

Definition 1 (Color spanning). *A region R in $\mathbb{R}^2$ is said to be* **color spanning** *if it contains at least one member of $\mathcal{L}$ having color i for all $i = 1, 2, \ldots, m$.*

A strip $\mathcal{V}$ is an unbounded region enclosed by two parallel lines which are called the boundaries of $\mathcal{V}$. The width of a strip $\mathcal{V}$ is determined by the perpendicular distance between its two boundaries. We use CSS to denote any color spanning strip.

Definition 2 (Minimal and minimum-CSS). *A CSS is said to be a **minimal-CSS** if it cannot be shrunk further without violating the definition 1 of the color spanning region. There may exist more than one minimal-CSS for $\mathcal{L}$. The one having minimum width among all minimal-CSSs is said to be minimum width color spanning strip, and will be denoted by **minimum-CSS**.*

2.1 A Single Color Spanning Strip of Arbitrary Orientation

Problem 1 (Single color spanning strip). *Given a set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of (possibly intersecting) line segments in $\mathbb{R}^2$; each segment $\ell_i \in \mathcal{L}$ is attached with one of m distinct colors ($3 \leq m < n$), compute a minimum-CSS $\mathcal{V}$ of arbitrary orientation.*

We use geometric duality [5] to solve this problem. Here, a point $p = (a, b)$ in the primal plane is represented by a line $p^* : \ y = ax - b$ in the dual plane, and a line $l : \ y = \mathfrak{m}x + c$ in the primal plane is represented by a point $l^* = (\mathfrak{m}, -c)$ in the dual plane. Note that, $\mathcal{V}$ may be vertical or non-vertical. We can compute the vertical strip $\mathcal{V}$ by sweeping a pair of vertical lines to locate all possible $minimal - CSS$s'. The one having minimum width is preserved as the $minimum - CSS$ as the initialization of this algorithm. This needs $O(n \log n)$ time, maintaining an array of size m for storing the number of segments of each color i $(1 \leq i \leq m)$ lying in the present position of the strip defined by the pair of sweep lines. We now concentrate on computing the smallest width non-vertical CSS. Since the point-line duality cannot handle any vertical line, if there exists any vertical line in $\mathcal{L}$, we rotate the entire set $\mathcal{L}$ by a small angle to make each segment non-vertical.

An arbitrary-oriented strip $\mathcal{V}$ is defined by its two boundaries, namely the upper boundary $ub(\mathcal{V})$ and the lower boundary $lb(\mathcal{V})$, that are mutually parallel lines; the point of intersection of $ub(\mathcal{V})$ with any vertical line lies above that of $lb(\mathcal{V})$ with the same vertical line. A strip $\mathcal{V}$ in the primal plane is mapped to a vertical line segment $\mathcal{V}^* = [lb^*(\mathcal{V}), ub^*(\mathcal{V})]$ in dual plane[1]. In duality transformation, a line segment $\ell_i = [p, q] \in \mathcal{L}$ in primal plane is mapped to a double wedge ℓ_i^* in dual plane [5], which is closure of the symmetric difference of the two half planes delimited by the lines p^* and q^*, and it does not contain any vertical line. The point of intersection of p^* and q^* is known as the *center-point* of the double wedge ℓ_i^* and is denoted by $cp(\ell_i^*)$. Let L_v be the vertical line passing through $cp(\ell_i^*)$; ℓ_{mid} be the line passing through ℓ_i^* with slope $\frac{1}{2}$(slope of p^* + slope of q^*). Each double wedge ℓ_i^* can be viewed as four rays, namely left-top $\ell t(\ell_i^*)$, left-bottom $\ell b(\ell_i^*)$, right-top $rt(\ell_i^*)$ and right-bottom $rb(\ell_i^*)$ emanating from $cp(\ell_i^*)$. The ray $\ell t(\ell_i^*)$ (resp. $\ell b(\ell_i^*)$) lies above (resp. below) ℓ_{mid} to the left of L_v, and the ray $rt(\ell_i^*)$ (resp. $rb(\ell_i^*)$) lies above (resp. below) ℓ_{mid} to the right of L_v (see Fig. 1). We refer to the union of $\ell t(\ell_i^*)$ and $rt(\ell_i^*)$ as $\mathcal{UT}(\ell_i^*)$ (*upper trace* of ℓ_i^*), and the union of $\ell b(\ell_i^*)$ and $rb(\ell_i^*)$ as $\mathcal{LT}(\ell_i^*)$ (*lower trace* of ℓ_i^*). While transforming a segment in primal plane to a double wedge using duality, the color associated with that segment remains same. The color of the double wedge (resp. line segment) ℓ^* (resp. ℓ) is denoted by $col(\ell^*)$ (resp. $col(\ell)$). The following result states the property of minimum width CSS for line segments.

Theorem 1. *A minimal-CSS $\mathcal{V}$ is defined by three segments, say $\ell_i, \ell_j, \ell_k \in \mathcal{L}$ lie inside $\mathcal{V}$, and one of its boundaries ($lb(\mathcal{V})$ or $ub(\mathcal{V})$) contains an endpoint of two segments $\in \{\ell_i, \ell_j, \ell_k\}$, and the other boundary contains an endpoint of a segment $\in \{\ell_i, \ell_j, \ell_k\}$. It may happen that both the boundaries of $\mathcal{V}$ may touch the two endpoints of a single segment $\in \{\ell_i, \ell_j, \ell_k\}$. The color of the segments defining $\mathcal{V}$ are different and none of their colors repeat inside the $\mathcal{V}$.*

[1] Due to the fact that that both the boundaries of $\mathcal{V}$ have same gradient.

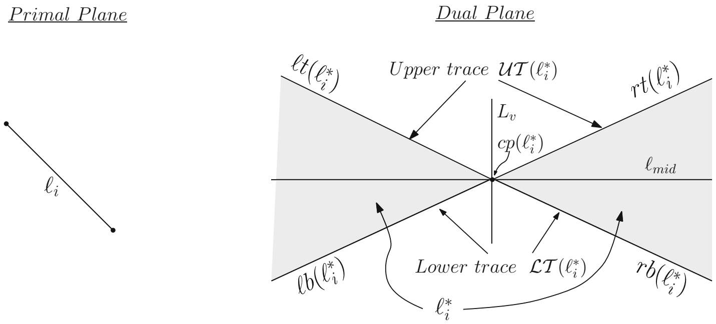

Fig. 1. Line segment ℓ_i in primal plane and its corresponding double wedge ℓ_i^* in dual plane

A segment s is said to be intersected by a double wedge ℓ^* if both $\mathcal{UT}(\ell^*)$ and $\mathcal{LT}(\ell^*)$ intersect with s. If a vertical segment in the dual plane (corresponding to a strip in the primal plane) is color spanning, it will be referred to as a *CS_segment*. The dual *CS_segment* $\mathcal{V}^*$ of a *minimal-CSS* $\mathcal{V}$ defined by three segments ℓ_i, ℓ_j and ℓ_k is shown in the Fig. 2. Due to the Theorem 1, we observe the following.

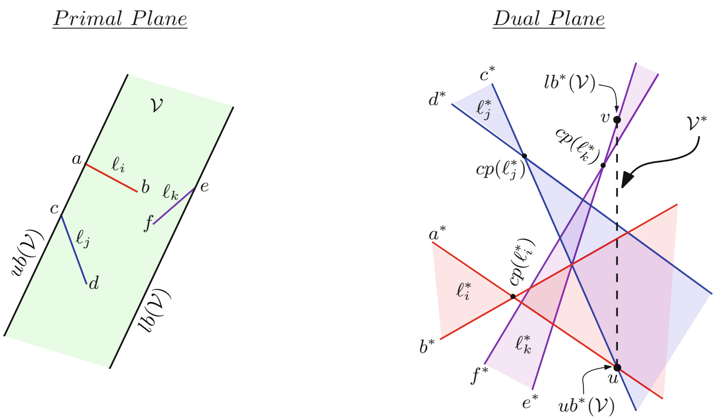

Fig. 2. Dual of strip $\mathcal{V}$ defined by three line segments ℓ_i, ℓ_j and ℓ_k.

Observation 1. *A strip $\mathcal{V}$ whose upper (resp. lower) boundary is defined by ℓ_i and ℓ_j, and lower (resp. upper) boundary is defined by ℓ_k in the primal plane, corresponds to the vertical segment $\mathcal{V}^* = [u, v]$ in the dual plane, where u is the point of intersection between the lower (resp. upper) traces of ℓ_i^* and ℓ_j^*, and v lies on the upper (resp. lower) trace of ℓ_k^* vertically above (resp. below) the point u (see Fig. 2). The width of this strip $\mathcal{V}$ is given by $\frac{|y(u)-y(v)|}{\sqrt{1+(x(u))^2}}$.*

Observation 2. *The dual $\mathcal{V}^*$ of a strip $\mathcal{V}$ is color spanning if at least one (left or right) wedge or the center-point of dual ℓ^* of ℓ of each color intersects with $\mathcal{V}^*$.*

Let $\mathcal{L}^* = \{\ell_i^* \mid \ell_i \in \mathcal{L}\}$ be the set of double wedges in the dual plane corresponding to the segments of $\mathcal{L}$ in the primal plane. We associate a vector $color[1..m]$ of length m with each $CS\_segment$. Its i^{th} entry indicates the number of double wedges of color i are intersected by the $CS\_segment$. We sweep a vertical line λ from left to right among the members in $\mathcal{L}^*$ to identify a $CS\_segment$ of minimum (dual) length.

Data Structure: We use five pointers for the four rays of each double wedge $\ell^* \in \mathcal{L}^*$; the value of these pointers corresponding to a double wedge ℓ^* depend on the position (i.e. the x-coordinate) of the vertical sweep line λ. These five pointers of all the double wedges in $\mathcal{L}^*$ are initialized to $NULL$. We now describe the significance of these pointers of a double wedge ℓ^* at a particular position, say $x{=}\alpha$, of the sweep line λ.

- $Self$**:** This pointer, associated with the ray $\ell t(\ell^*)$ (resp. $\ell b(\ell^*)$) points to $\ell b(\ell^*)$ (resp. $\ell t(\ell^*)$), and the same associated with $rt(\ell^*)$ (resp. $rb(\ell^*)$) points to $rb(\ell^*)$ (resp. $rt(\ell^*)$). From an upper trace of a double wedge, we can access its lower trace through this pointer, and vice versa.
- CS_up**:** It is associated with the rays in the lower trace $\mathcal{LT}(\ell^*)$, and it points to the upper trace $\mathcal{UT}(t^*)$ of a double wedge t^* (say), vertically above it, such that a vertical segment at the present position ($x = \alpha$) of the sweep line λ lying between the $\mathcal{LT}(\ell^*)$ and $\mathcal{UT}(t^*)$ is color spanning (See Observation 2). Note that, for the upper trace of all the members in $\mathcal{L}^*$, this pointer is always set to $NULL$.
- CS_dwn**:** It is associated with the two rays in the upper trace $\mathcal{UT}(\ell^*)$, which points to the lower trace $\mathcal{LT}(t^*)$ of a double wedge t^* (say), vertically below it, such that a vertical segment at the present position ($x = \alpha$) of the sweep line λ lying between the $\mathcal{LT}(t^*)$ and $\mathcal{UT}(\ell^*)$, is color spanning (See the Observation 2). This pointer is $NULL$ for the rays $\ell b(\ell^*)$ and $rb(\ell^*)$.
- $Same\_col\_up$**:** It is associated with the two rays in the lower trace $\mathcal{LT}(\ell^*)$ of each double wedge ℓ^*. It points to the upper trace of a double wedge t^* ($\neq \ell^*$), that lies vertically above ℓ^* and is closest one to ℓ^* among all the double wedges having the color same as that of ℓ^* at the present position of the sweep line λ, provided such a double wedge t^* exists for ℓ^*; otherwise it is set to $NULL$. Also, this pointer is $NULL$ for the rays in $\mathcal{UT}(\ell^*)$, $\forall \ell^* \in \mathcal{L}^*$.
- $Same\_col\_dwn$**:** It is associated with the two rays in the upper trace $\mathcal{UT}(\ell^*)$ of each double wedge ℓ^*. It points to the lower trace of a double wedge, say t^* ($\neq \ell^*$), if t^* lies vertically below ℓ^* and is closest one to ℓ^* among all double wedges having the color same as that of ℓ^* at the present position of the sweep line λ, provided such a double wedge t^* exists for ℓ^*; otherwise it is set to $NULL$. Also, this pointer is set to $NULL$ for the rays in $\mathcal{LT}(\ell^*)$, $\forall \ell^* \in \mathcal{L}^*$.

Algorithm: The event points of the vertical sweep line λ are the points of intersection of the dual of the endpoints of the input segments in $\mathcal{L}$ (see Observation 1) in sorted order with respect to their x-coordinates. These event points are created in $O(n^2)$ time [11], and are stored in an array A. We initialize the aforesaid pointers for each ray of the double wedges in $\mathcal{L}^*$ with their respective values at the first event position of the sweep line λ in the array A. During the sweep, we compute the $CS\_segment$ at each event point $e \in A$, and finally report the minimum length $CS\_segment$ observed.

During the sweep, the status of the sweep line λ is maintained as a list of the dual lines of $\mathcal{L}^*$ that appear on the sweep line λ in top to bottom order. The status of the sweep line is updated after processing each event point $e \in A$ as follows:

Case (i) The event point $e \in A$ corresponds to $cp(\ell_i^*)$, $\ell_i \in \mathcal{L}$:
We do the following updates:
Assign $Same\_col\_dwn(rt(\ell_i^*)) = Same\ \_col\_dwn(\ell t(\ell_i^*))$,
$Same\_col\_up(rb(\ell_i^*)) = Same\_col\_up(\ell b(\ell_i^*))$,
$CS\_up(rb(\ell_i^*)) = CS\_up(\ell b(\ell_i^*))$, and $CS\_dwn(rt(\ell_i^*)) = CS\_dwn(\ell t(\ell_i^*))$.

Case (ii) The event point $e \in A$ corresponds to the intersection of the upper (resp. lower) traces of two double wedges ℓ_i^* and ℓ_j^* of same color:
Let e be the point of intersection of upper traces of ℓ_i^* and ℓ_j^*, where $\mathcal{UT}(\ell_i^*)$ lies below $\mathcal{UT}(\ell_j^*)$ at the small distance $\epsilon > 0$ to the left of e. If at the left of e, the $Same\_col\_dwn$ of $\mathcal{UT}(\ell_i^*)$ points to the lower trace $\mathcal{LT}(\ell_k^*)$ of a double wedge ℓ_k^*, then at the event point e, the $Same\_col\_dwn$ of $\mathcal{UT}(\ell_i^*)$ needs to be updated to $\mathcal{LT}(\ell_j^*)$ provided $\mathcal{LT}(\ell_j^*)$ lies above $\mathcal{LT}(\ell_k^*)$.
Similarly, if the $Same\_col\_dwn$ pointer of $\mathcal{UT}(\ell_j^*)$ points to $\mathcal{LT}(\ell_i^*)$ just before the event e, then at the event point e, the $Same\_col\_dwn$ pointer of $\mathcal{UT}(\ell_j^*)$ will be updated to point to the old values (i.e. just before the event e) of $Same\_col\_dwn$ of $\mathcal{UT}(\ell_i^*)$; otherwise $Same\_col\_dwn$ pointer of $\mathcal{UT}(\ell_j^*)$ remains unaltered. However, the value of CS_dwn (resp. CS_up) pointer of the upper (resp. lower) trace remains unchanged at the event e.

Case (iii) The event point $e \in A$ of λ corresponds to the intersection of the upper (resp. lower) traces of the different colored double wedges ℓ_i^* and ℓ_j^*:
Suppose the upper trace of ℓ_i^* lies below that of ℓ_j^* at the $\epsilon > 0$ distance to the left of the event e. Without loss of generality, we assume that at $x = e$, the $\mathcal{UT}(\ell_i^*)$ and $\mathcal{UT}(\ell_j^*)$ are $\ell t(\ell_i^*)$ and $\ell t(\ell_j^*)$, respectively. At the event point e of λ, the CS_dwn pointers of $\ell t(\ell_i^*)$ and $\ell t(\ell_j^*)$ needs to be updated as follows.
Update of $CS\_dwn(\ell t(\ell_j^*))$: If at $\epsilon > 0$ distance to the left of the event e, the ℓ_i^* is essential[2] in the color spanning vertical segment $CS\_segment$ that spans from $\ell t(\ell_j^*)$ to $CS\_dwn(\ell t(\ell_j^*))$, then at the event e, we update the $CS\_dwn(\ell t(\ell_j^*))$ pointer to point to $Same\_col\_dwn(\ell t(\ell_i^*))$, otherwise the pointer $CS\_dwn(\ell t(\ell_j^*))$ remains unaltered.

[2] $color[col(\ell_i^*)]$ is 1 for an essential segment ℓ_i^* in the $CS\_segment$.

Update of $CS\_dwn(\ell t(\ell_i^*))$: At the event point e, the $CS\_dwn(\ell t(\ell_i^*))$ will be updated to point old value (i.e. just before the event e) of $CS\_dwn(\ell t(\ell_j^*))$, provided the lower trace of ℓ_i^* lies above that of double wedge pointed by old $CS\_dwn(\ell t(\ell_j^*))$ and the lower trace of the double wedge pointed by $CS\_dwn(\ell t(\ell_j^*))$ lies above that of $\ell t(\ell_i^*)$ before the event e.

Case (iv) The event point $e \in A$ of λ corresponds to the intersection of the lower (resp. upper) and upper (resp. lower) trace of the double wedges ℓ_i^* and ℓ_j^*, respectively:
In this case, only the status of the sweep line λ is changed.

All such aforesaid events take $O(1)$ time. For each types (i.e. aforesaid cases) of event e, if the pointer *Same_col_dwn* (resp. *Same_col_up*) associated with a double wedge, say ℓ_i^*, is updated to point to a double wedge ℓ_k^*, then the pointer *Same_col_up* (resp. *Same_col_dwn*) of the double wedge ℓ_k^* is also updated to point to ℓ_i^*. As the sweep line λ passes through each of its event point e, we update the CS_dwn (resp. CS_up) pointers of the ray involved in the upper (resp. lower) trace associated with the event point e. Also we need to update the CS_dwn (resp. CS_up) pointers of those rays whose CS_dwn (resp. CS_up) pointed to the rays associated with the event e. So this update may take linear amount of time to search for the rays whose CS_dwn (resp. CS_up) pointers need to be updated. This time can be expedited, if we use the idea of the following lemma and create two height balanced trees T_1 and T_2. These two trees are updated as the λ moves forward.

Lemma 1. *At a position, say $x = \alpha$, of the sweep line λ, if the CS_up (resp.CS_dwn) pointers of lower (resp. upper) traces of a pair of double wedges ℓ_i^* and ℓ_j^* point to the upper (resp. lower) trace of the same double wedge ℓ_k^* with $col(\ell_i) \neq col(\ell_j) \neq col(\ell_k)$, then the CS_up (resp. CS_dwn) pointers for the double wedges ℓ_{i+1}^*, ℓ_{i+2}^*, ..., ℓ_{j-1}^* lying between ℓ_i^* and ℓ_j^* also point to ℓ_k^*.*

In T_1 (resp. T_2), we store the triple (i, j, k) where the CS_up (resp. CS_dwn) pointer for the double wedges ℓ_i^*, ℓ_{i+1}^*, ..., ℓ_j^* points to the same double wedge ℓ_k^* (see the Lemma 1). The nodes in T_1 (resp. T_2) are mutually exclusive and exhaustive. These nodes are stored in T_1 (resp. T_2) with respect to the status of the λ (i.e. the ordered intersection of the double wedges with λ). We can create this T_1 (resp. T_2) in linear amount of time. We can update the CS_up pointers of the double wedges having the same value of CS_up pointers in $O(\log n)$ time using T_1 (or T_2). Hence each event needs $O(\log n)$ processing time and since there are total $O(n^2)$ events, we compute the minimal strips at each event point by checking the appropriate pointer (CS_up or CS_dwn) and report the one having minimum length among all the minimal strips obtained at each event points. The data structure of the sweep line λ at its current event point depends on the data structure of λ at its previous event point. As the sweep line λ moves forward through its event points, we need to compute $CS\_segment$ at the current event point e of λ using the data structure stored at e which can be obtained only from the information of the data structure at its previous event point. Hence as the sweep line moves through its event points e, we need to store the linear sized data structures of the previous event point of e instead of storing all the event points of λ altogether, and the same space can be reused as the λ moves forward to its next event point. Thus we obtain the following result.

Theorem 2. *The minimum width color spanning strip of arbitrary orientation for a given set of n colored line segments in $\mathbb{R}^2$ can be determined in $O(n^2 \log n)$ time and $O(n)$ space.*

2.2 Two Congruent Strips of Arbitrary Orientation

Problem 2 (Two congruent color spanning strips). *Given a set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of (possibly intersecting) line segments in $\mathbb{R}^2$; each segment $\ell_i \in \mathcal{L}$ is attached with one of m distinct colors ($3 \leq m \leq n$), the objective is to compute arbitrary-oriented two congruent color spanning strips $\mathcal{V}_1$ and $\mathcal{V}_2$ of minimum width such that the set of segments covered by $\mathcal{V}_1$ and $\mathcal{V}_2$ are disjoint.*

In the context of Problem 2, note that if a segment lies inside $\mathcal{V}_1 \cap \mathcal{V}_2$, then it is suitably considered to lie completely inside one of $\mathcal{V}_1$ and $\mathcal{V}_2$. This problem is equivalent to compute a pair of *minimal-CSS* ($\mathcal{V}_1$, $\mathcal{V}_2$), so that the width of its larger strip is minimized among all possible pair of *minimal-CSS*s. We solve this problem by considering all possible *minimal-CSS* $\mathcal{V}_1$, and for each of them we choose a *minimum-CSS* $\mathcal{V}_2$ which covers the segments that are not covered by $\mathcal{V}_1$.

Fact 1. *If the two intersecting color spanning strips $\mathcal{V}_i$ and $\mathcal{V}_j$ are disjoint with respect to the segments ($\in \mathcal{L}$) covered by them, then there exists no double wedges in $\mathcal{L}^*$, that intersect with both the corresponding CS_segments $\mathcal{V}_i^*$ and $\mathcal{V}_j^*$, respectively.*

First we compute the minimal length *CS_segment* $\mathcal{V}_i^*$ at each event point e of a sweep line λ_1 using the procedure described in the Sect. 2.1. For each such $\mathcal{V}_i^*$, we compute all the $\mathcal{V}_j^*$ which are disjoint with $\mathcal{V}_i^*$ (see the Fact 1) using another sweep line λ_2 that lies to the right of λ_1. Among all these $\mathcal{V}_j^*$, we choose the one with minimum length. We check whether $\mathcal{V}_j^*$ is disjoint with $\mathcal{V}_i^*$ or not, as follows.

After computing the *CS_segment* $\mathcal{V}_i^*$ at the event point e of λ_1, we determine the set of double wedges, say $D_i^* \subseteq \mathcal{L}^*$, that completely intersect with $\mathcal{V}_i^*$ in the sorted order using the status of the sweep line λ_1. We consider the sweep line λ_2 at one of its event point, say e' which occurs to the right of e, and let, $\mathcal{V}_{i'}^*$ be the *CS_segment* at e' and the another endpoint of $\mathcal{V}_{i'}^*$ be p'. If e' is due to the intersection of any of its double wedge in D_i^*, then we only update D_i^* by swapping the corresponding two intersecting double wedges and move to the next event of λ_2. However, if e' is not due to the intersection of any of its double wedge in D_i^*, then we compute the double wedge $d \in D_i^*$ (resp. $d' \in D_i^*$) immediately below e' (resp. p'). This can be determined in $O(\log n)$ time from D_i^*. The strips $\mathcal{V}_{i'}^*$ and $\mathcal{V}_i^*$ will be disjoint in the following two cases:

(i) The d and d' exist, and they are same, (ii) both of the d and d' do not exist.

In all other cases, $\mathcal{V}_{i'}^*$ and $\mathcal{V}_i^*$ will be overlapping. As mentioned earlier in the Sect. 2.1, all the event points of the sweep lines need not to be stored altogether and the space complexity is also linear for the Problem 2. Since there are $O(n^2)$ such event points for both λ_1 and λ_2, we obtain the following result.

Theorem 3. *For a given set of n line segments in $\mathbb{R}^2$, we can compute two congruent disjoint (with respect to the segments covered) color spanning strips of the minimum width, if such a pair exists, otherwise we report that no such pair exists in $O(n^4 \log n)$ time and $O(n)$ space.*

Problem 3 (Restricted version of the Two congruent color spanning strip). *For the same inputs as in the Problem 2, compute two congruent, minimum width color spanning disjoint strips which are parallel to each other.*

The Theorem 1 leads to the following observation.

Observation 3. *If CSSs $\mathcal{V}_1$ and $\mathcal{V}_2$ are parallel to each other, then at least one boundary of one of the two strips $\mathcal{V}_1$ and $\mathcal{V}_2$ must contain an endpoint of two different colored segments in $\mathcal{L}$ that are covered by the corresponding strip. However, if $\mathcal{V}_1$ and $\mathcal{V}_2$ are not parallel, then one boundary of each of the strips must contain two endpoints of two different colored segments in $\mathcal{L}$. Note that, both the endpoints of the same segment may also define a boundary.*

If the two disjoint CSSs $\mathcal{V}_i$ and $\mathcal{V}_j$ are mutually parallel, then their corresponding dual $\mathcal{V}_i^*$ and $\mathcal{V}_j^*$, are two disjoint vertical segments, one lying vertically above the other.

Consider a minimal $CS\_segment$ $\mathcal{V}_i^*$ at an event point $e_i \in A$, determined by the procedure described in Sect. 2.1. Now, we will determine all possible minimal $CS_segment$s $\mathcal{V}_j^*$ lying vertically above as well as below $\mathcal{V}_i^*$ in amortized $O(n)$ time. We explain the method of computing all the $CS_segment$s below $\mathcal{V}_i^*$. Suppose ℓ_i^* be a double wedge lying immediately below $\mathcal{V}_i^*$. Take two pointers *top* and *bottom*, where *top* points to ℓ_i^* and *bottom* points to a double wedge, say ℓ_j^*, below ℓ_i^*, such that the vertical segment from $\mathcal{UT}(\ell_i^*)$ to $\mathcal{LT}(\ell_j^*)$ at $x = x(e)$ is a $CS\_segment$, say $\mathcal{V}_j^*$. It can be determined from the sweep line status. The next $CS\_segment$ below $\mathcal{V}_j^*$, starting from double wedge that is just below ℓ_i^*, can be determined by shifting both the pointers *top* and *bottom* downwards through the list of double wedges in the current sweep line status array. In this way, we compute all the $CS_segment$s lying below $\mathcal{V}_i^*$ in linear amount of time. Similarly, we compute all possible minimal $CS_segment$s that lie vertically above $\mathcal{V}_i^*$. Finally, we choose the one having minimum length as $\mathcal{V}_j^*$ that pairs with $\mathcal{V}_i^*$. The entire task is done in $O(n)$ time.

We repeat the above steps to compute all possible pairs $(\mathcal{V}_i^*, \mathcal{V}_j^*)$ at each event point $e_i \in A$ by sweeping the line λ. It may happen that there exists only one $CS\_segment$ in the entire floor. In that case, only one color spanning strip will be reported. Similar to the Problem 1 in the Sect. 2.1, we need not store all the events for this Problem 3 and hence, it needs $O(n)$ space. As there are $O(n^2)$ event points in the worst case, we have the following result.

Theorem 4. *For a given set of n colored line segments in $\mathbb{R}^2$, we can compute two congruent disjoint parallel color spanning strips of the minimum width in $O(n^3)$ time and $O(n)$ space.*

2.3 Two Congruent Strips of Arbitrary Orientation Whose Union is Color Spanning

Problem 4 (Union color spanning problem). *Given a set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of (not necessarily disjoint) line segments in $\mathbb{R}^2$; each segment $\ell_i \in \mathcal{L}$ is associated with one of m distinct colors ($3 \leq m \leq n$), the objective is to compute arbitrary-oriented two congruent (i) disjoint (ii) non-disjoint strips $\mathcal{V}_1$ and $\mathcal{V}_2$ of minimum width, whose union is color spanning.*

We first compute two arbitrary-oriented disjoint strips $\mathcal{V}_1$ and $\mathcal{V}_2$. It is obvious that $\mathcal{V}_1$ and $\mathcal{V}_2$ are parallel. We use the line sweeping technique over the set of double wedges $\mathcal{L}^*$. For each color c_i ($1 \leq i \leq m$), we maintain two sorted lists of all the double wedges with respect to their lower traces and upper traces respectively, at each position $x = x(e)$, where e is the event point of the sweep line λ. These two lists are updated in constant time at each event point e as the line λ sweeps rightward. From these lists containing lower (resp. upper) traces of each color, we also compute a sorted array $first\_col\_\mathcal{LT}[1..m]$ (resp. $first\_col\_\mathcal{UT}[1..m]$) of size m at each event point e of λ, that keeps the first occurring lower (resp. upper) traces of each distinct colored double wedge that lies completely below (resp. above) the point e. These two arrays are sorted with respect to the point of intersections of the sweep line λ with the members of the array. Our algorithm executes the following tasks at each event point e of λ.

We take $\mathcal{V}_i^*$ with one of its endpoints starting at the event point e. Without loss of generality, we assume that the event point e is the intersection of two upper traces of two different colored double wedges. The other endpoint of $\mathcal{V}_i^*$ lie on the lower trace (lying vertically below e) of any one of the double wedges from the sorted array "$first\_col\_\mathcal{LT}$" and suppose this $\mathcal{V}_i^*$ intersects with the double wedges of k different colors. We use a color array $\mathcal{C}$ which keeps track of the color of wedges that completely intersect with $\mathcal{V}_i^*$ and it can be computed in $O(k)$ time. For each $\mathcal{V}_i^*$, we compute corresponding $\mathcal{V}_j^*$ which covers the double wedges of remaining $(m-k)$ colors. The two endpoints of the dual segment $\mathcal{V}_i^*$ (resp. $\mathcal{V}_j^*$) of the strip $\mathcal{V}_i$ (resp. $\mathcal{V}_j$) are denoted by top_1 (resp. top_2) and bot_1 (resp. bot_2). The color of the double wedges pointed by the top_1, top_2, bot_1 and bot_2 will be of different colors. Once we compute such a $\mathcal{V}_j^*$, we measure the length of $\mathcal{V}_j^*$. Next, we shift top_2 downward to the next upper trace below it and recompute $\mathcal{V}_j^*$. In this way we compute all possible $\mathcal{V}_j^*$ that lie below $\mathcal{V}_i^*$. Similarly we can compute all possible $\mathcal{V}_j^*$ lying above $\mathcal{V}_i^*$ in $O(n)$ time. We choose the $\mathcal{V}_j^*$ with minimum length. Now we increase the length of $\mathcal{V}_i^*$ by moving bot_1 pointer to next entry of the array "$first\_col\_\mathcal{LT}$" so that $\mathcal{V}_i^*$ covers now one extra color and we compute the corresponding minimum length $\mathcal{V}_j^*$. To obtain the optimal pair $\mathcal{V}_i^*$ and $\mathcal{V}_j^*$ at the event point e, we need not to compute all possible $\mathcal{V}_i^*$ due to the following lemma.

Lemma 2. *The function* $\max\{width(\mathcal{V}_i), width(\mathcal{V}_j)\}$ *is a convex function.*

Proof. If we increase the width of $\mathcal{V}_1$, the width of $\mathcal{V}_2$ either remains same or decreased, as the union of $\mathcal{V}_1$ and $\mathcal{V}_2$ is color spanning. □

Since, the length of $\mathcal{V}_i^*$ is increased by shifting bot_1 through the members of the sorted array "$first\_col\_\mathcal{LT}$" (of size m), to minimize the $\max\{width(\mathcal{V}_i), width(\mathcal{V}_j)\}$ we need to iterate the above procedure at most $\log m$ times (see Lemma 2) at each event point e of λ. Finally we repeat the same procedure at each event point e of λ to find the optimal solution of the Problem 4. Similar to the Problem 2, we need linear space to solve the Problem 4. Thus we obtain the following result.

Theorem 5. *For the set* $\mathcal{L}$ *of* n *line segments in* $\mathbb{R}^2$*, we can compute two congruent disjoint parallel strips of the minimum width, whose union is color spanning in* $O(n^3 \log m)$ *time and* $O(n)$ *space.*

Now, we determine two arbitrary-oriented non-disjoint strips $\mathcal{V}_1$ and $\mathcal{V}_2$, whose union is color spanning. We apply almost the same technique as well as the data structures that are used to compute two disjoint strips (first part of the Problem 4). In this case, the two vertical $CS_segment$s $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$ will lie at two different event positions of two different sweep lines λ_1 and λ_2, respectively. Both $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$ will be defined by three segments (see Observation 3). We first consider a $CS\_segment$ at an event point e of the sweep line λ_1 covering, say k colors (see the algorithm described for disjoint case). Then we compute the $\mathcal{V}_j^*$ (which covers the remaining $(m-k)$ colors) at each event e' of the sweep line λ_2 lying to the right of λ_1. We can compute the two sorted arrays (defined earlier) $first\_col\_\mathcal{LT}[1..m]$ (resp. $first\_col\_\mathcal{UT}[1..m]$) at the event point e' in $O(m)$ time. For a fixed length $CS\_segment$ $\mathcal{V}_i^*$ at e, we can determine minimum $\mathcal{V}_j^*$ at e' in $O(m)$ time. Now, the Lemma 2 says that in $O(m \log m)$ time, we can compute the optimum pair $(\mathcal{V}_i^*, \mathcal{V}_j^*)$ with one of their endpoints at e and e', respectively. Now, there are $O(n^2)$ different possible positions for each of the event points e and e'. Thus we obtain the following result.

Theorem 6. *For a given set of* n *line segments in* $\mathbb{R}^2$*, we can compute two congruent non-disjoint strips of the minimum width, whose union is color spanning in* $O(n^4 m \log m)$ *time and* $O(n)$ *space.*

2.4 Color Spanning Rectangle (*CSR*) of Arbitrary Orientation

Problem 5. *Given a set* $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ *of (possibly intersecting)* n *line segments in* $\mathbb{R}^2$*; each segment* $\ell_i \in \mathcal{L}$ *is attached with one of* m *distinct colors* $(3 \le m \le n)$*, the objective is to compute an arbitrary-oriented color spanning rectangle (CSR)* $\mathcal{R}$ *of minimum area.*

Fact 2. *A color spanning rectangle (CSR)* $\mathcal{R}$ *is the intersection of two color spanning strips, say* $\mathcal{V}_1$ *and* $\mathcal{V}_2$*, which are perpendicular to each other and boundaries of* $\mathcal{V}_1$ *and* $\mathcal{V}_2$ *pass through the opposite parallel sides of* $\mathcal{R}$ *(Fig. 3).*

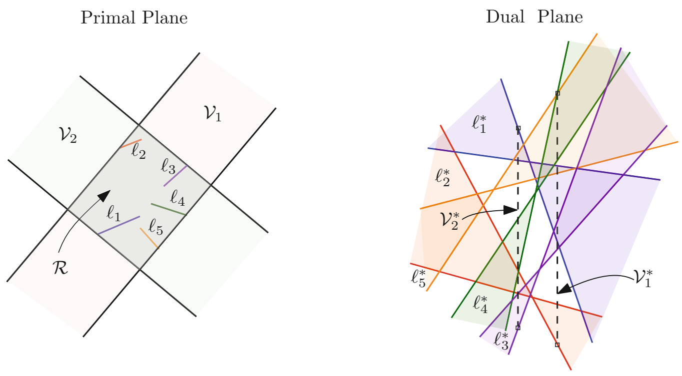

Fig. 3. Color spanning rectangle $\mathcal{R}$ in primal is represented by a pair $(\mathcal{V}_1^*, \mathcal{V}_2^*)$ in dual. (Color figure online)

Lemma 3. *One side of the minimum area CSR $\mathcal{R}$ for $\mathcal{L}$ must contain exactly two endpoints of any two segments (one endpoint of each) in $\mathcal{L}$ and each of the remaining sides of $\mathcal{R}$ must contain one endpoint of a segment in $\mathcal{L}$. Also the same colored endpoints can occur at most twice on the boundary of $\mathcal{R}$, and the color of the segments whose endpoint lie on the boundary of $\mathcal{R}$, will be distinct from remaining segments that lie completely inside $\mathcal{R}$.*

Proof. Suppose, $\mathcal{R}$ be the minimum area CSR and E be the set of segments enclosed by it. Since the area of $\mathcal{R}$ is minimum among all CSR of $\mathcal{L}$, the color of the segments in $\mathcal{L}$ whose endpoints lie on the boundary of $\mathcal{R}$ must be distinct from the others lying on or inside $\mathcal{R}$, otherwise we can rotate and/or shrink $\mathcal{R}$ to obtain another CSR with smaller area than $\mathcal{R}$, contradicting the assumption that $\mathcal{R}$ is minimum area CSR. Note that, the same colored endpoints may occur at most twice on the boundary of $\mathcal{R}$, if both the endpoints of a segment occur at the boundary of the $\mathcal{R}$. Let P be the convex hull of E. Two vertices of P must lie on a side of minimum area rectangle $\mathcal{R}$, if it encloses P [7]. □

Since the adjacent sides of a rectangle are perpendicular to each other, we observe the following.

Observation 4. *A CSR which is the intersection of two perpendicular strips $\mathcal{V}_1$ and $\mathcal{V}_2$ in the primal plane (Fact 2), can be represented by two color spanning vertical segments (CS_segment) $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$ in the dual plane such that if the $\mathcal{V}_1^*$ is at $x = x_1$, then the $\mathcal{V}_2^*$ will be at $x = \frac{-1}{x_1}$, and the set of double wedges intersecting both $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$ must be color spanning.*

Let $\mathcal{R}_{a,b}$ be the color spanning rectangle with one side defined by two points a and b of the two segments ℓ_a and ℓ_b, respectively. The four sides of $\mathcal{R}_{a,b}$ are numbered sequentially 1 to 4 in counter clockwise direction where the side 1 contains the points a and b. We first generate all possible CSR $\mathcal{R}_{a,b}$ with side 2 being defined by all possible segments with same (say blue) color only, and then we compute the minimum area rectangle among them. Similarly, we use the remaining colors for side 2 to generate all possible $\mathcal{R}_{a,b}$. Finally, we do this for all pairs of $\ell_a(\in \mathcal{L})$ and $\ell_b(\in \mathcal{L})$ to compute the overall minimum area CSR for $\mathcal{L}$. This entire process is done through the duality transformations of the segments in $\mathcal{L}$ and line sweeping technique.

Two vertical lines λ_1 and λ_2 sweep from left to right through the set of event points ξ generated by the intersections of the members in $\mathcal{L}^*$ as defined in Problem 1. The λ_1 and λ_2 have the same set of event points; however, if the λ_1 reaches at its event point $e \in \xi$, then we move λ_2 to its event point $e' \in \xi$ whose position is at $x = \frac{-1}{x(e)}$ (see the Observation 4). However, if no such event point exists at $x = \frac{-1}{x(e)}$, then we choose $e' \in \xi$ that occurs immediately to the left of $x = \frac{-1}{x(e)}$.

Consider a minimal width CSS $\mathcal{V}_1$ with its lower boundary passing through the endpoints a and b of the segments ℓ_a and ℓ_b, respectively. Let c, the endpoint of a segment ℓ_c, lies on the upper boundary of $\mathcal{V}_1$, so that $col(\ell_a) \neq col(\ell_b) \neq col(\ell_c)$ (see the Lemma 3). The double wedges ℓ_a^*, ℓ_b^* and ℓ_c^* represent the duals of ℓ_a, ℓ_b and ℓ_c, respectively. In dual, the point of intersection of $\mathcal{UT}(\ell_a^*)$ and $\mathcal{UT}(\ell_b^*)$ is the top endpoint of the $CS\_segment$ $\mathcal{V}_1^*$, and its bottom endpoint will lie vertically below its top endpoint and on the $\mathcal{LT}(\ell_c^*)$. Let $\mathcal{L}_1^* \subseteq \mathcal{L}^*$ be the set of double wedges intersecting with $\mathcal{V}_1^*$ and $\mathcal{L}_2^* \subset \mathcal{L}^*$ be the set of double wedges lying completely below $\mathcal{V}_1^*$. At each event point e of sweep line λ_1, we compute the $CS\_segments$ $\mathcal{V}_1^*$. For each such segment $\mathcal{V}_1^*$, we determine the corresponding $\mathcal{V}_2^*$ at $x = \frac{-1}{x(e)}$ which lies immediately after the event point, say e' of the sweep line λ_2. The order of the double wedges of $\mathcal{L}_1^*$ at $x = \frac{-1}{x(e)}$, can be obtained from the status of the sweep line λ_2 at $x = x(e')$ in linear time. The $\mathcal{V}_2^*$ to be determined, must intersect with ℓ_a^*, ℓ_b^* and ℓ_c^* in order to obtain a CSR made by the intersection of $\mathcal{V}_1$ and $\mathcal{V}_2$. We take two pointers top_1 (resp. top_2) and bot_1 (resp. bot_2) that are initialized to point to double wedges having top endpoint and bottom endpoint of $\mathcal{V}_1^*$ (resp. $\mathcal{V}_2^*$), respectively. At $x = \frac{-1}{x(e)}$, suppose $\mathcal{L}_{up}^* \subset \mathcal{L}_1^*$ be the set of double wedges with their upper trace lying above both $\mathcal{UT}(\ell_a^*)$ and $\mathcal{UT}(\ell_b^*)$. We choose a distinct colored (say blue) double wedge $w^* \in \mathcal{L}_{up}^*$ at $x = \frac{-1}{x(e)}$ which is closest to and above both the $\mathcal{UT}(\ell_a^*)$ and $\mathcal{UT}(\ell_b^*)$, and compute the $CS\_segment$ $\mathcal{V}_2^*$ (with top endpoint on $\mathcal{UT}(w^*)$) for the double wedges in $\mathcal{L}_1^*$ by maintaining a color array in linear time. Actually this w defines the side 2 of CSR (discussed above). The bottom endpoint of $\mathcal{V}_2^*$ will lie on the lower trace of a double wedge, say t^* and bot_2 will point to t^*. Note that w^* and t^* must be the essential[3] double wedges in $\mathcal{V}_2^*$. We determine the area of the rectangle whose sides are given by $\mathcal{V}_1^*$ and $\mathcal{V}_2^*$ in dual plane. Next we process each double wedge $d_i^* \in \mathcal{L}_2^*$ lying below ℓ_c^* at $x = x(e)$, as follows.

$\boldsymbol{col(d_i^*) = col(w^*)}$**:** Here,

- if $\mathcal{UT}(d_i^*)$ lies below $\mathcal{UT}(\ell_a^*)$ or $\mathcal{UT}(\ell_b^*)$ and above $\mathcal{LT}(\ell_a^*)$ or $\mathcal{LT}(\ell_b^*)$ at $x = -\frac{1}{x(e)}$, then we stop, since no CSR $\mathcal{R}_{a,b}$ with $col(w^*)$ in side 2 is possible with such d_i^*.

[3] Essential color occurs exactly once inside the color spanning region.

- if $\mathcal{UT}(d_i^*)$ lies above both $\mathcal{UT}(\ell_a^*)$ and $\mathcal{UT}(\ell_b^*)$ at $x = \frac{-1}{x(e)}$, and below the double wedge pointed by top_2 pointer, then we reject all the double wedges lying above d_i^* (since same colored segment w as that of d_i cannot occur in $CSR\ \mathcal{R}_{a,b}$) by updating color array. We also update the top_2 pointer to point to d_i^*.
- if d_i^* lies below $\mathcal{LT}(\ell_a^*)$ and $\mathcal{LT}(\ell_b^*)$ at $x = -\frac{1}{x(e)}$, and above bot_2, then $CSR\ \mathcal{R}_{a,b}$ cannot include d_i^* and hence we move the bot_2 pointer to the double wedge lying immediately above d_i^*. Note that, now the vertical segment defined by the top_2 and bot_2 may not be the color spanning, and we should compute the $CSR\ \mathcal{R}_{a,b}$ whenever we get a $\mathcal{V}_2^*$.
- if d_i^* lies above top_2 at $x = -\frac{1}{x(e)}$, then we reject d_i^* (since $col(d_i^*)$ and $col(w^*)$ are same).

$\boldsymbol{col(d_i^*) \neq col(w^*)}$**:** If d_i^* lies between the double wedges pointed by top_2 and bot_2, then we insert d_i^* in $\mathcal{V}_2^*$ and update the color array for $\mathcal{V}_2^*$. If the color of the double wedge pointed by bot_2 is same as that of d_i^* then we move bot_2 upwards to point to an essential wedge. If $\mathcal{V}_2^*$ becomes color spanning after insertion of d_i^*, then we update the bot_1 pointer to point to d_i^* and compute $\mathcal{V}_2^*$ and the corresponding CSR. Otherwise we reject d_i^*.

Since each segment is inserted and/or deleted in $\mathcal{V}_2^*$ at most once, the above procedure needs amortized linear time. We repeat this procedure for each distinct colored double wedge w^* to obtain the minimum area $CSR\ \mathcal{R}_{a,b}$. Finally, we execute this process at each event point of λ_1 to determine the overall minimum area CSR for $\mathcal{L}$. There are $O(n^2)$ event points for λ_1. Similar to the Problem 2, we also need the linear space to solve this problem. Since there are at most m distinct colors, and at each event point of λ_1, it takes amortized linear amount of time to compute all possible $CSS\ \mathcal{V}_2^*$ with each distinct color of the upper trace, we have the following result.

Theorem 7. *The minimum sized (area) color spanning rectangle of arbitrary orientation for a given set of* n *colored line segments in* $\mathbb{R}^2$ *can be determined in* $O(n^3m)$ *time and* $O(n)$ *space.*

References

1. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristán, V.: Smallest color-spanning objects. Algorithms - ESA 2001. In: Proceedings of 9th Annual European Symposium, Aarhus, Denmark, 28–31 August 2001, vol. 2161, pp. 278–289 (2001)
2. Acharyya, A., Maheshwari, A., Nandy, S.C.: Color-spanning localized query. Theor. Comput. Sci. **861**, 85–101 (2021)
3. Asano, T., Bhattacharya, B.K., Keil, J.M., Yao, F.F.: Clustering algorithms based on minimum and maximum spanning trees. Proceedings of the Fourth Annual Symposium on Computational Geometry, Urbana-Champaign, IL, USA, 6–8 June 1988, pp. 252–257 (1988)
4. Bae, S.W.: An algorithm for computing a minimum-width color-spanning rectangular annulus. J. Korean Inst. Inf. Sci. Eng. **44**, 246–252 (2017)

5. de Berg, M., Cheong, O., van Kreveld, M.J., Overmars, M.H.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Berlin (2008)
6. Das, S., Goswami, P.P., Nandy, S.C.: Smallest color-spanning object revisited. Int. J. Comput. Geom. Appl. **19**(5), 457–478 (2009)
7. Freeman, H., Shapira, R.: Determining the minimum-area encasing rectangle for an arbitrary closed curve. Commun. ACM **18**, 409–413 (1975)
8. Hasheminejad, J., Khanteimouri, P., Mohades, A.: Computing the smallest color-spanning equaliteral triangle. In: 31st European Workshop on Computational Geometry. pp. 32–35 (2015)
9. Huttenlocher, D.P., Kedem, K., Sharir, M.: The upper envelope of Voronoi surfaces and its applications. In: Proceedings of the Seventh Annual Symposium on Computational Geometry, North Conway, NH, USA, 10–12 June 1991. pp. 194–203. ACM (1991)
10. Khanteimouri, P., Mohades, A., Abam, M.A., Kazemi, M.R.: Computing the smallest color-spanning axis-parallel square. In: Cai, L., Cheng, S.-W., Lam, T.-W. (eds.) ISAAC 2013. LNCS, vol. 8283, pp. 634–643. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45030-3_59
11. Lee, D.T., Ching, Y.: The power of geometric duality revisited. Inf. Process. Lett. **21**, 117–122 (1985)
12. Pruente, J.: Minimum diameter color-spanning sets revisited. Discret. Optim. **34**, 100550 (2019)