Shortest Color-Spanning $t$ Intervals

- **Input:** a set of $n$ points on a line, where each point has one of $k$ colors, and an integer $s_i \geq 1$ for each color $i$, $1 \leq i \leq k$.
- **Output:** $t$ intervals to cover at least $s_i$ points of each color $i$, such that the maximum length of the intervals is minimized.

**SCSI-1**

Chen and Misiolek (2013) introduced the problem SCSI-1, and presented an algorithm running in $O(n)$ time if the input points are sorted.

**SCSI-2 with $s_i = 1$**

Khanteimouri et al. (2013) gave an $O(n^2 \log n)$ time algorithm for the special case of SCSI-2 with $s_i = 1$ for all colors $i$.

**SCSI-2 with $s_i \geq 1$**

Our first result is an improved algorithm for SCSI-2 with arbitrary $s_i \geq 1$:

**Theorem 1.** **SCSI-2** admits an exact algorithm running in $O(n^2)$ time.

**Negative Results for SCSI-$t$**

The problems SCSI-1 and SCSI-2 naturally generalize to SCSI-$t$ for $t \geq 1$. Our next theorem shows that SCSI-$t$ is intractable in a very strong sense:

**Theorem 2.** Approximating **SCSI-$t$** within any ratio is NP-hard when $t$ is part of the input, is $W[2]$-hard when $t$ is the parameter, and is $W[1]$-hard with both $t$ and $k$ as parameters. Moreover, the NP-hardness and the $W[2]$-hardness with parameter $t$ hold even if $s_i = 1$ for all $i$. 
Negative Results for SCSI-t

Unless FPT = W[2]:
• no approximation algorithm of any ratio with running time $f(t) \cdot \text{poly}(n)$ for any function $f$.

Unless FPT = W[1]:
• no approximation algorithm of any ratio with running time $g(t, k) \cdot \text{poly}(n)$ for any function $g$.

It is a standard hypothesis in parameterized complexity that

$$\text{FPT} \subset \text{W[1]} \subset \text{W[2]}.$$ 

Positive Result for SCSI-t

In contrast to the very negative result in Theorem 2, our following theorem shows that the special case of SCSI-t with $s_i = 1$ for all $i$ is fixed-parameter tractable when the parameter is the number $k$ of colors:

**Theorem 3.** The special case of SCSI-t with $s_i = 1$ for all $i$ admits an exact algorithm running in $O(2^k n \cdot \max\{k, \log n\})$ time.

In particular, we can solve SCSI-t with $s_i = 1$ for all $i$ in $O(n \log n)$ time if $k$ is a constant, and in polynomial time if $k = O(\log n)$.

Related Work: In The Plane

Abellanas et al. (2001) proposed an $O(n(n - k)\log^2 k)$ time algorithm for computing the smallest (by perimeter or area) axis-parallel rectangle that contains at least one point of each color.

Das et al. (2009) gave an improved algorithm with $O(n(n - k)\log k)$ time for this problem, and moreover gave an $O(n^3 \log k)$ time algorithm for computing the smallest color-spanning rectangle of arbitrary orientation.

Abellanas et al. (2001) and Das et al. (2009) also gave algorithms for computing the smallest color-spanning strips.

Related Work: Color-Spanning Set

Given a set of colored points, a color-spanning set is a subset of the input points including at least one point of each color.

The various color-spanning problems for colored points with $s_i = 1$ for all colors $i$ can be viewed as finding a color-spanning set such that certain geometric property of the set is optimized.

Related Work: In The Plane

Khanteimouri et al. (2013) gave an $O(n \log^2 n)$ time algorithm for computing the smallest color-spanning axis-parallel square.

Barba et al. (2013) considered the related problem of computing a region (e.g., rectangle, square, or disc) that contains exactly (instead of “at least”) $s_i$ points of each color $i$. 

Related Work: In The Plane

Abellanas et al. (2001) proposed an $O(n(n - k)\log^2 k)$ time algorithm for computing the smallest (by perimeter or area) axis-parallel rectangle that contains at least one point of each color.

Das et al. (2009) gave an improved algorithm with $O(n(n - k)\log k)$ time for this problem, and moreover gave an $O(n^3 \log k)$ time algorithm for computing the smallest color-spanning rectangle of arbitrary orientation.

Abellanas et al. (2001) and Das et al. (2009) also gave algorithms for computing the smallest color-spanning strips.

Related Work: In The Plane

Khanteimouri et al. (2013) gave an $O(n \log^2 n)$ time algorithm for computing the smallest color-spanning axis-parallel square.

Barba et al. (2013) considered the related problem of computing a region (e.g., rectangle, square, or disc) that contains exactly (instead of “at least”) $s_i$ points of each color $i$. 

Related Work: Color-Spanning Set

Given a set of colored points, a color-spanning set is a subset of the input points including at least one point of each color.

The various color-spanning problems for colored points with $s_i = 1$ for all colors $i$ can be viewed as finding a color-spanning set such that certain geometric property of the set is optimized.
Related Work: Color-Spanning Set

Fleischer and Xu (2009, 2011) gave polynomial time algorithms for finding a minimum-diameter color-spanning set under the $L_1$ or $L_\infty$ metric, and proved that the problem is NP-hard for all $L_p$ with $1 < p < \infty$.

Ju et al. (2012) gave an efficient algorithm for computing a color-spanning set with the maximum diameter, and proved that several other problems are NP-hard, e.g., finding the color-spanning set with the largest closest-pair distance.

Fan et al. (2013) studied the problem of finding a color-spanning set with the minimum connection radius in the corresponding disk intersection graph.

W[2]-hardness of SCSI-t with Parameter $t$

We show that approximating SCSI-t within any ratio is NP-hard when $t$ is part of the input, and moreover is W[2]-hard when $t$ is the parameter, even for the special case of SCSI-t with $s_i = 1$ for all $i$.

This is achieved by a polynomial FPT reduction from the NP-hard and W[2]-hard problem Colorful Red-Blue Dominating Set.

Given a bipartite graph $G = (R \cup B, E)$ where each vertex in $R$ has one of $\hat{r}$ colors, decide whether there exists a colorful dominating set $D$ of $\hat{r}$ vertices in $R$, including exactly one vertex of each color, such that each vertex in $B$ is adjacent to at least one vertex in $D$.

Put $\hat{n}_r = |R|$, $\hat{n}_b = |B|$, and $\hat{m} = |E|$.

In this example, $\hat{r} = 2$, $\hat{n}_r = 3$, $\hat{n}_b = 5$, and $\hat{m} = 9$.

Our reduction constructs a colored point set with $k = \hat{r} + \hat{n}_b$ colors, including

- one $r$-color $i$ for each color $i$ of the vertices in $R$,
- one $b$-color $v$ for each vertex $v$ in $B$.

Place $n = \hat{n}_r + \hat{m}$ points in $\hat{n}_r$ clusters, one cluster for each vertex in $R$. For each vertex $u$ of color $i$ in $R$, the cluster for $u$ contains

- one point of $r$-color $i$,
- one point of $b$-color $v$ for each vertex $v$ in $B$ that is adjacent to $u$.

Set $t = \hat{r}$.

Set the required number $s_i$ of points to be covered to 1 for each $r$-color and for each $b$-color.
Three clusters: \{1, v, w, x\}, \{1, w, x, y, z\}, \{2, y, z\}.

The two vertices for \{1, v, w, x\} and \{2, y, z\} are a colorful dominating set.

Arrange the clusters of points such that

- the maximum distance between two points in each cluster is 1,
- the minimum distance between two points from different clusters is \(\gamma > 1\).

The reduction can be easily made polynomial; it is also an FPT reduction with parameter \(t\) since \(t\) is a function of \(\hat{n}\) only.

The following two lemmas imply that it is both NP-hard and \(\text{W}[2]\)-hard (with parameter \(t\)) to approximate \(\text{SCSI}-t\) within \(\gamma\) for any approximation ratio \(\gamma > 1\):

**Lemma 1.** There is a colorful dominating set of size \(\hat{n}\) in the graph only if there is a color-spanning set of \(t\) intervals for the colored points with maximum length 1.

Given a colorful dominating set in the graph, the \(\hat{n}\) clusters corresponding to the \(\hat{n}\) vertices in the dominating set clearly span all colors of the points, and each cluster can be covered with an interval of length 1.

**Lemma 2.** There is a colorful dominating set of size \(\hat{n}\) in the graph if there is a color-spanning set of \(t\) intervals for the colored points with maximum length less than \(\gamma\).

Given a color-spanning set of \(t = \hat{n}\) intervals with maximum length less than \(\gamma\), each interval can cover points from at most one cluster due to the restriction on the maximum interval length.

Thus to span all \(\hat{n}\) \(r\)-colors of the points, each of the \(\hat{n}\) intervals must cover a point of a distinct \(r\)-color in a distinct cluster.

Since all \(\hat{n}_b\) \(b\)-colors of the points are spanned by the \(\hat{n}\) intervals, the \(\hat{n}\) vertices in \(R\) corresponding to the \(\hat{n}\) clusters must dominate all \(\hat{n}_b\) vertices in \(B\).
**Inapproximability**

The approximation lower bound $\gamma/1$ can be arbitrarily large: it can even be a function of the instance size for the SCSI-$t$ problem; indeed with suitable (say, binary) encoding of the interval coordinates as rational numbers, the lower bound can be exponential in the size of the problem instance.

$$\gamma/1$$

Moreover, if the colored points are allowed to coincide, or equivalently, if each point can have more than one color, then each cluster in our reduction can be compressed into a single point, and consequently the problem cannot be approximated at all.

$$\gamma/0$$

**Inapproximability**

Put $\hat{n} = |V|$ and $m = |E|$.

In this example, $\hat{k} = 3$, $\hat{n} = 5$, and $m = 6$.

For each $i$, $1 \leq i \leq \hat{k}$, let $\hat{n}_i$ denote the number of vertices of color $i$, and let $v^1_i, \ldots, v^n_i$ denote these vertices.

**$k$ Colors**

Our reduction constructs a colored point set with $k = \hat{k} + \binom{3}{2} + 4 \cdot \binom{2}{2}$ colors of three types:

- one vertex color $i$ for each color $i$ of the vertices in $V$;
- one edge color $ij$ with $i < j$ for each unordered pair of colors $\{i, j\}$ of the edges in $E$;
- four consistency colors $ji_1, ji_2, ij_1, ij_2$ for each unordered pair of colors $\{i, j\}$ of the edges in $E$.

**$n$ Points**

Place $n = \hat{n} + \hat{m} + (\hat{k} - 1)\hat{n} + 2\hat{n} \cdot \hat{m}$ points in $\hat{n} + \hat{m}$ clusters, including exactly one vertex cluster for each vertex in $V$, and exactly one edge cluster for each edge in $E$:

- For each vertex of color $i$ in $V$, put one point of vertex color $i$ in the corresponding vertex cluster.
- For each edge of color pair $\{i, j\}$ in $E$, put one point of edge color $ij$ in the corresponding edge cluster.

**W[1]-hardness of SCSI-$t$ with Parameters $t$ and $k$**

We show that approximating SCSI-$t$ within any ratio is W[1]-hard with parameters $t$ and $k$ by an FPT reduction from the W[1]-hard problem **Colored Clique**:

Given a graph $G = (V, E)$ where each vertex has one of $\hat{k}$ colors, decide whether there exists in $G$ a colored clique of $\hat{k}$ pairwise-adjacent vertices including exactly one vertex of each color.
**n Points**

Place \( n = \hat{n} + \hat{m} + (\hat{k} - 1)\hat{n} \cdot \hat{n} + 2\hat{n} \cdot \hat{m} \) points in \( \hat{n} + \hat{m} \) clusters, including exactly one vertex cluster for each vertex in \( V \), and exactly one edge cluster for each edge in \( E \):

- For each vertex \( v_i^p \) in \( V \), and for each color \( j \neq i \), put \( p \) points of consistency color \( j \), and \( \hat{n} - p \) points of consistency color \( j \) in the vertex cluster for \( v_i^p \).

- For each edge \( e = \{ v_i^p, v_i^q \} \) in \( E \), put \( \hat{n} - q \) points of consistency color \( j \), \( p \) points of consistency color \( j \), \( \hat{n} - q \) points of consistency color \( j \), and \( q \) points of consistency color \( j \) in the edge cluster for \( e \).

**t and s_i**

Set \( t = \hat{k} + \left( \frac{\hat{n}}{2} \right) \).

Set the required number \( s_i \) of points to be covered to

- 1 for each vertex color and for each edge color,
- \( \hat{n} \) for each consistency color.

The colored clique consisting of the three vertices \( v^1, v^2, v^3 \) and the three edges \( \{ v^1, v^2 \}, \{ v^1, v^3 \}, \{ v^2, v^3 \} \) correspond to three vertex clusters and three edge clusters whose union is \( \{ 1, 2, 3, 12, 13, 12^5, 12^5, 12^5, 21^5, 21^5, 23^5, 23^5, 32^5, 32^5 \} \).

The five vertices \( v^1, v^2, v^3, v^2, v^2 \) correspond to the five vertex clusters:

\[
\{ 1, 21^1, 21^2, 21^2, 31^2, 31^2 \}, \\
\{ 2, 121^1, 12^2, 32^1, 32^1, 32^2 \}, \\
\{ 3, 131^1, 13^2, 23^1, 23^1, 23^2 \}, \\
\{ 1, 21^1, 21^2, 31^2, 31^2 \}, \\
\{ 2, 121^1, 12^2, 32^1, 32^1, 32^2 \}.
\]

Again, arrange the clusters of points such that

- the maximum distance between two points in each cluster is 1,
- the minimum distance between two points from different clusters is \( \gamma > 1 \).
\textbf{Lemma 3.} There is a colored clique of size \( k \) in the graph only if there is a color-spanning set of \( t \) intervals for the colored points with maximum length 1.

\textbf{Lemma 4.} There is a colored clique of size \( k \) in the graph if there is a color-spanning set of \( t \) intervals for the colored points with maximum length less than \( \gamma \).

\section*{An FPT Algorithm for SCSI-\( t \)}

We show that the special case of SCSI-\( t \) with \( s_i = 1 \) for all \( i \) is fixed-parameter tractable when the parameter is the number \( k \) of colors, and admits an exact algorithm running in \( O(2^k n \cdot \max\{k, \log n\}) \) time.

\section*{Preprocessing}

Let \( P = \{p_1, p_2, \ldots, p_n\} \) be the set of \( n \) points, sorted from left to right. For each \( i, 1 \leq i \leq n \),

- \( I'_i \) is the interval of length \( d \) with right endpoint at \( p_i \).
- \( g(i) \) is the smallest index \( j \), \( 1 \leq j \leq i \), such that the points \( p_j, p_{j+1}, \ldots, p_i \) are covered by \( I'_i \).
- \( C_i \) is the set of colors of the points covered by the interval \( I'_i \).

\section*{Dynamic Programming}

Let \( C \) be the set of all \( k \) colors.

For each subset \( S \) of \( C \), for any \( 1 \leq i \leq n \), denote by \( N[S, i] \) the minimum number of intervals of length \( d \) for covering at least one point of each color in \( S \) among the points \( p_1, p_2, \ldots, p_i \).

Then \( d \) is feasible if and only if \( N[C, n] \leq t \).

\section*{Decision Problem}

Given a distance value \( d \), decide whether there exists a color-spanning set of \( t \) intervals with uniform length \( d \).

\section*{Dynamic Programming}

The table \( N[S, i] \) can be computed by dynamic programming with the base case \( N[\emptyset, i] = 0 \) and the recurrence

\[ N[S, i] = \min\{N[S, i - 1], N[S \setminus C_i, g(i) - 1] + 1\} \]

which distinguishes two cases: either the point \( p_i \) is not covered, or it is the last point covered by (without loss of generality it is the right endpoint of) an interval of length \( d \), which covers points from \( p_{g(i)} \) to \( p_i \) with color set \( C_i \).
Decision Problem

We can compute the values \( g(i) \) in \( O(n) \) time and the color sets \( C_i \) in \( O(nk) \) time by the standard sweeping technique.

The dynamic programming procedure takes \( O(2^k k n) \) time: the table \( N[S, i] \) has \( 2^k n \) entries; each entry takes \( O(k) \) time with \( g(i) \) and \( C_i \) precomputed.

In summary, we can solve the decision problem in \( O(2^k k n) \) time. With the power of word RAM, we can further improve the running time to \( O(2^k n \lceil k / \log n \rceil) \).

Binary Search

There must exist an optimal solution for the problem SCSI-t that consists of \( t \) intervals such that a longest interval have both left and right endpoints in \( P \).

For any two indices \( i \) and \( j \) with \( 1 \leq i \leq j \leq n \), let \( d_{ij} \) be the distance between the points \( p_i \) and \( p_j \).

Let \( D \) be the set of the distances \( d_{ij} \) for all \( 1 \leq i \leq j \leq n \).

Note that \( |D| = \Theta(n^2) \). To avoid the quadratic time of constructing \( D \), we represent \( D \) implicitly as a set of sorted arrays...

Candidate Interval

Let \( P = \{p_1, p_2, \ldots, p_n\} \) be the set of \( n \) points, sorted from left to right.

For any interval \( I \), let \( d(I) \) denote the length of \( I \).

If two intervals together cover at least \( s_i \) points of each color \( i \) in \( P \), then they form a feasible solution for SCSI-2.

An interval \( I \) is a candidate interval if there is another interval \( I' \) such that

1. \( d(I') \leq d(I) \),
2. \( I \) and \( I' \) form a feasible solution for SCSI-2.

Binary Search

For each \( 1 \leq i \leq n \), let \( D_i = \{d_{ij} | i \leq j \leq n\} \). Then \( D = \bigcup_{i=1}^{n} D_i \).

Note that each set \( D_i \) can be considered as a sorted array of size \( O(n) \) since for any two indices \( j_1 \leq j_2 \), it holds that \( d_{j_1, i} \leq d_{j_2, i} \). Moreover, given any index \( j \), the value \( d_{ij} \) can be obtained in constant time.

By a technique called binary search on sorted arrays, the running time becomes \( O(2^k \min\{k, \log n\} \log n) \), which is \( O(2^k k n) \) if \( k \geq \log n \), and is \( O(2^k n \log n) \) otherwise.

\( O(2^k n \cdot \max\{k, \log n\}) \)

O(n) Time Check

Given any interval \( I \), we can determine whether \( I \) is a candidate interval in \( O(n) \) time:

- First, by scanning all points in \( O(n) \) time, we discard all points from \( P \) whose colors have already been covered by \( I \). Let \( s'_i \) be the number of points covered by \( I \) for each color \( i \).

- Second, using the previous \( O(n) \)-time algorithm for SCSI-1, we find a shortest interval \( I' \) that covers at least \( s_i - s'_i \) points of each color \( i \) among the remaining points of \( P \). Then \( I \) is a candidate interval if and only if \( d(I') \leq d(I) \).
**First Observation**

There must exist an optimal solution for the problem SCSI-2 that consists of two intervals such that the longer interval has both left and right endpoints in \( P \).

The left endpoint of interval \( I_1 \) is not at any point of \( P \). We can obtain another interval \( I_1' \) by moving the left endpoint of \( I_1 \) rightwards for an infinitesimal distance such that \( I_1' \) and \( I_1 \) cover the same subset of points of \( P \).

\[ I_1' \]

\[ L \]

\[ p_i \quad p_{i+1} \quad \cdots \quad p_j \]

**Second Observation**

For each \( 1 \leq i \leq n \), let \( h(i) \) be the smallest index \( j \) such that \( I_{ij} \) is a candidate interval. Then \( I^*_i = I_{ih(i)} \).

Let \( j = h(i) \) and \( j' = h(i') \). If \( i < i' \), then \( j \leq j' \).

Suppose that \( I_{ij} \) and \( I' \) form a feasible solution, where \( d(I') \leq d(I_{ij}) \). Then \( I_{ij} \) and \( I' \) also form a feasible solution, where \( d(I') \leq d(I_{ij}) \).

Monotonicity: \( h(1) \leq h(2) \leq \cdots \leq h(n) \).

**\( O(n^2) \) Time**

For convenience, we let \( h(0) = 1 \). For any \( 0 \leq i \leq n - 1 \), suppose \( h(i) \) has already been computed. Then by the monotonicity of \( h \), to compute \( h(i + 1) \), we only need to check the intervals \( I_{i+1,j} \) in \( I_{i+1} \) for \( j = h(i), h(i) + 1, \ldots \) by a sequential search to find the smallest index \( j \) such that \( I_{i+1,j} \) is a candidate interval.

We can find the \( n \) intervals \( I^*_i \) for \( i = 1, 2, \ldots, n \) by checking only \( O(n) \) intervals of \( I \).

For each such interval, we can check whether it is a candidate interval in \( O(n) \) time. The total running time is thus \( O(n^2) \).

**\( O(n^3) \) Time**

Let \( I \) denote the set of all intervals each of which has its two endpoints in \( P \). Clearly, \( |I| = \Theta(n^2) \).

By the previous observation, we can solve the problem SCSI-2 by finding the shortest candidate interval \( I^* \) in \( I \).

Since \( |I| = \Theta(n^2) \), if we check every interval of \( I \), then we can solve the problem SCSI-2 in \( O(n^3) \) time. In the following we present an \( O(n^2) \) time algorithm.

For any \( 1 \leq i \leq j \leq n \), denote by \( I_{ij} \) the interval with left endpoint at \( p_i \) and right endpoint at \( p_j \).

For any \( 1 \leq i \leq n \), let \( I_i = \{ I_{ij} \mid i \leq j \leq n \} \).

The sets \( I_i \) for \( i = 1, \ldots, n \) form a partition of \( I \).

For each \( 1 \leq i \leq n \), let \( I^*_i \) be the shortest candidate interval in \( I_i \).

\( I^* \) is among \( I^*_1, I^*_2, \ldots, I^*_n \).