Analysis of a Greedy Heuristic for the Labeling of a Map with a Time-Window Interface

Annika Bonerath, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Elmar Langetepe, Benjamin Niedermann

September 5, 2022

Abstract

In this paper, we analyze the approximation quality of a greedy heuristic for automatic map labeling. As input, we have a set of events, each associated with a label at a fixed position, a timestamp, and a weight. Let a time-window labeling be a selection of these labels such that all corresponding timestamps lie in a queried time window and no two labels overlap. A solution to the time-window labeling problem consists of a data structure that encodes a time-window labeling for each possible time window; when a user specifies a time window of interest using a slider interface, we query the data structure for the corresponding labeling.

We define the quality of a time-window labeling solution as the sum of the weights of the labels in each time-window labeling, integrated over all time windows. We aim at maximizing the quality under the condition that a label may never disappear when the user shrinks the time window. In this paper, we analyze how well a greedy heuristic approximates the maximum quality that can be realized under this condition.

On the one hand, we present an instance with square labels of equal size and equal weight for which the greedy heuristic fails to find a solution of at least $1/4$ of the quality of an optimal solution. On the other hand, we prove that the greedy heuristic does guarantee a solution with at least $1/8$ of the quality of an optimal solution. In the case of disk-shaped labels of equal size and equal weight, the greedy heuristic gives a solution with at least $1/10$ of the quality of an optimal solution. If the labels are squares or disks of equal size and the maximum weight divided by the minimum weight is at most $b$, then the greedy heuristic has approximation ratio $\Theta(\log b)$.

1 Introduction

For the visualization of spatial data, labeling is a standard technique. Labels are placed in the map over the data points and each label contains information about the corresponding data point, e.g., a name or an icon. Typically, placing all labels leads to unwanted overlaps. Classical map labeling solves this by computing a largest overlap-free selection of labels [22]. Labeling is also often used for interactive maps. Here the user interactively changes the visualization and therewith the labeling must be updated. Recent research concerns stability and consistency conditions for labeling during a change in the visualization [23, 2, 21, 20].

Bonerath et al. [9] look at consistent labeling of maps with a time-slider interface. They introduce a consistency model and describe a data structure that guarantees such consistency criteria. Furthermore, they provide algorithms for the computation of the data structure. In this document, we provide the theoretical analysis of the greedy heuristic presented by Bonerath et al. [9]. In the following, we wrap up their application scenario, model, data structure and algorithm.

Application Scenario In this work, we consider event data as input where each event consists of a label, a timestamp and a positive weight, where the weight of a label reflects its importance; see Figure 1a. Our application scenario consists of a user interface where the user can interactively choose a time window and then an overlap-free selection of labels with corresponding timestamps in the queried time window is visualized; see Figure 1b. We call such a selection of labels a time-window labeling. The users can choose the
time window with a dynamic query interface as introduced by Williamson and Shneiderman [19]. They can perform four basic interactions; see Figure 1c: (1) panning: a continuous translation of the time window; (2) uniform scaling: a continuous change of both boundaries of the time window in opposite directions, such that the center of the time window remains the same; (3+4) right- and left-sided scaling: a continuous change of the time window’s right or left boundary, respectively.

Model As introduced by Bonerath et al. [9], we look at a two-step approach: first, we compute a data structure that encodes a time-window labeling for each possible time window, and then, as the user specifies time windows of interest, we query the data structure for the corresponding labelings. An alternative view of the data structure is the following: a query with a time window \([t', t'']\) can be regarded as a point \((t', t'')\) in a two-dimensional configuration space: the first coordinate of the point specifies the starting time of the time window, the second coordinate specifies the end time. The data structure encodes, for each label \(\ell\), its activity region \(\tau\): the set of points \((t', t'')\) in configuration space such that \(\ell\) is included in the labeling for the time window \([t', t'']\). A query with a time window \([t', t'']\) consists of finding all labels whose activity regions include the point \((t', t'')\).

As for classic map labeling, we aim at transferring as much information as possible for each time-window query. In particular, we want to maximize the sum of the sizes of displayed labels integrated over all time-
window queries. This is inspired by active range maximization as introduced by Been et al. [6]. A naive approach on computing the labelings for this interface would be to apply the classic strategy (find a largest overlap-free selection of labels) independently for each possible time window. This might lead to unwanted flickering effects: a single label may appear and disappear repeatedly even within a single basic interaction; see Figure 2a. One effect of flickering is that the user cannot isolate a single event by systematically shrinking the time window, as the corresponding label may appear and disappear repeatedly without any recognizable systematic. We require that if an event is displayed for a time window \( Q = [t', t'' \rangle \) then it is also displayed for all the time windows that are contained in \( Q \) and contain the timestamp of the event. Figure 2b shows a solution that satisfies this requirement. We call this property Containment.

**Related Work**  
**Map Labeling** is a widely investigated field. For the static case, a common goal is to maximize the number of the displayed labels (or their total weight) while avoiding overlapping labels [10] [22] [18]. For non-interactive animated maps, additional stability constraints are added [5] [17] [7]. For interactive maps, Been et al. [6] introduced the concept of active ranges for labels, considering zooming, panning and rotations of the map which is basically the same concept as our activity regions. They consider the active range to be an interval over, e.g., zoom levels and prove that for such a scenario the maximization of all active ranges is NP-hard.

The data structure that is discussed in this paper, can be classified as a *time-windowed data structure*. This concept from the field of computational geometry, and it summarizes data structures that aim at efficiently answering time-window queries. Our approach is a time-windowed data structure that uses labeling as the underlying visualization technique. Nevertheless, in general, time-windowed data structures do not consider any consistency criteria during interaction. Current research on time-windowed data structures focuses on relational event graphs [4] [14] [16], basic problems from computational geometry [3] [8] [12] [13] [15], and also on event visualization based on α-shapes [11] and density maps [10].

**Approximation algorithms** are efficient algorithms that provide solutions for problems with a guarantee for the quality of the solution with respect to the optimal solution. They are often developed for NP-hard problems. The approximation ratio is a measure for the quality of an algorithm. Let \( \text{OPT}(I) \) denote the optimal solution of a maximization problem for instance \( I \) and \( A(I) \) the solution computed by algorithm \( A \). Then, \( A \) has approximation ratio \( k \) if \( \text{OPT}(I)/A(I) \leq k \) for all instances \( I \) of the problem. For minimization problems, one can define concepts analogously. In this paper, we discuss the approximation ratio of our greedy heuristic.

**Our results** In this paper, we discuss the approximation ratio of the greedy heuristic presented by Bonerath et al. [9]. In Section 2, we give a formalization of the problem and provide a detailed description of the approximation ratio of the greedy heuristic in Theorem 1. Then, the results from Section 3 and Section 4 together prove Theorem 1. In detail, in Section 3 we discuss the lower bound of the approximation ratio of the greedy heuristic. In Section 3.1 we give an exemplary instance for which the greedy heuristic has an approximation ratio above \( 4 \). In Section 3.2 we give a family of instances for which the approximation ratio is above \( n/2 \), where \( n \) is the number of the input events. We provide a deeper analysis of this family of instances in Section 3.3 leading to a more accurate lower bound. In Section 4 we discuss the upper bound of the approximation ratio.

## 2 Problem formalization and algorithm

**Formalization**  
A *label* is a set of points in the plane, for example, a square, a rectangle or a disk with a specific location. Let \( \{\ell_1, \ldots, \ell_n\} \) be a set of labels, \( \{t_1, \ldots, t_n\} \) be a set of timestamps, and \( \{w_1, \ldots, w_n\} \) a set of positive weights. We call the triplet \( e_i = (\ell_i, t_i, w_i) \) for \( 1 \leq i \leq n \) an event. The input data for the algorithm and the data structure is a set of events \( E \). We say that two events and their labels are in conflict if their labels overlap, that is, their interiors have a non-empty intersection. The dynamic query interface introduces two additional input parameters, the minimal and maximal time slider positions \( t_{\min} \) and \( t_{\max} \). We call a range \( Q = [t', t''] \leq [t_{\min}, t_{\max}] \) a *time-window query*. Be aware that depending on the context we interpret \( Q \) either as an interval \( [t', t''] \) or a point \( (t', t'') \) in the plane (configuration space). Due to \( t' \leq t'' \) it holds that \( (t', t'') \) always lies in the triangle \( (t_{\min}, t_{\min}), (t_{\max}, t_{\max}), (t_{\min}, t_{\max}) \). We say that if a label
time-window queries for which \( \tau \) queries, we understand \( \tau \) a time-window labeling. We call \( \nu(\tau) \) the corresponding activity regions do not overlap. Hence, querying an activity

Thus we arrive at the following problem formulation:

Given: A set \( E = \{e_1, \ldots, e_n\} \) of spatio-temporal events with labels; a weighting function \( w : E \rightarrow \mathbb{R}^+ \); the bounds \( t_{\text{min}} \) and \( t_{\text{max}} \) of the activity diagram.

Find: An activity diagram \( T = \{\tau_1, \ldots, \tau_n\} \) of activity regions for \( E \) that maximizes \( \sum_{i=1}^{n} w_i \cdot \text{area}(\tau_i) \) for \( e_i \in E \), where \( \text{area}(\tau_i) \) is the area of \( \tau_i \) in the activity diagram and where \( \tau_i \) is a rectangle with lower right corner \((t_i, t)\).

For an example of an optimal activity diagram, see Figure 3.

Algorithm Next, we present our greedy heuristic for computing a valid activity diagram. For illustration see Figure 4 and Algorithm 1. The greedy heuristic successively selects activity regions that yield the largest gain. While doing so, it maintains for each event that has not yet been placed in the activity diagram its maximal potential activity region. Each time a new event is selected and placed in the diagram, all remaining activity regions that are in conflict with this event are trimmed and their potential contribution is updated accordingly.

More in detail, we initialize for each event \( e_i \in E \) its largest possible activity region \( \tau_i \), i.e., the region that is spanned by \((t_{\text{min}}, t_{\text{max}})\) and \((t_i, t)\) and further, its volume \( \nu(\tau_i) \). We initialize a priority queue \( Q \) of events in descending order by their volumes and the empty solution set \( T \); see step 1 in Figure 4. Then, we remove the first event \( e_i \) from \( Q \) (with largest volume) and add \( \tau_i \) to the solution set \( T \). For each remaining event \( e_j \) in \( Q \) that is in conflict with \( e_i \) we trim \( \tau_j \) to the largest possible activity region \( \tau_j' \subseteq \tau_j \) that does
Figure 3: Valid activity diagram for the data presented in Figure 1. Pairs of conflicting events are marked with an arc, i.e., the pair $e_1$ and $e_3$ and the pair $e_2$ and $e_3$. The time-window query $Q = [t', t'']$ intersects the activity regions of $e_2$ and $e_4$.

Figure 4: Greedy heuristic for the data presented in Figure 1. For each step, the activity region that is chosen is marked with an orange stroke.

not intersect $\tau_i$. Finally, we update the volume of $e_j$ to $w_j \cdot \text{area}(\tau_j')$, possibly changing the position of $e_j$ in the sorting of $Q$. We iterate until $Q$ is empty. Finally, we return the valid activity diagram $T$.

**Approximation Ratio** Let $E$ be a set of events. Let $E_i \subseteq E$ be the set of events that are in conflict with $e_i \in E$. Then, let $a_i$ be the maximum size of a subset of $E_i$ where no two events are in conflict. Let $a$ be the maximum over all $a_i$ with $1 \leq i \leq n$. Let $b \in \mathbb{R}$ such that for any two events $e_i, e_j \in E$, we have $1/b \leq w_i/w_j \leq b$. We call $a$ the degree of interference of $E$ and $b$ the degree of unbalance of $E$. Using the degree of interference and the degree of unbalance, we can describe the approximation ratio of the greedy heuristic as follows.

**Theorem 1.** Let $E$ be a set of events with degree of interference $a$ and degree of unbalance $b$. The approximation ratio of the greedy heuristic is $\Theta(\min(a \log b, n))$. If $b = 1$, that is, all labels have equal weight, then the approximation ratio is at most $2a$.

### 3 Lower bounds on the approximation ratio of the greedy heuristic

#### 3.1 An instance with approximation ratio above 4

In this section, we provide an instance where the approximation ratio of the greedy algorithm is above 4.
The greedy heuristic, however, would first place \( \ell_5 \) with activity region \([0, 8 + 2\epsilon] \times [8 + 2\epsilon, 24] \) of size \( 128 + 16\epsilon - 4\epsilon^2 \). Note that \( \ell_5 \) intersects all labels \( \ell_1, ..., \ell_{11} \). Thus, in the activity diagram:

- \( \ell_1, ..., \ell_4 \) are now confined to the rectangle \([0, 8] \times [8, 8 + 2\epsilon] \) of size \( 16\epsilon \);
- \( \ell_6, ..., \ell_9 \) are now confined to the rectangle \([8 + 2\epsilon, 16] \times [16, 24] \) of size \( 64 - 16\epsilon \);
- \( \ell_{10} \) is now confined to the rectangle \([8 + 2\epsilon, 16 + \epsilon] \times [16 + \epsilon, 24] \) of size \( 64 - 16\epsilon + \epsilon^2 \);
- \( \ell_{11} \) is now confined to the rectangle \([8 + 2\epsilon, 21] \times [21, 24] \) of size \( 39 - 6\epsilon \);
- \( \ell_{12}, ..., \ell_{14} \) may still get active regions \([0, 21] \times [21, 24] \) of size \( 63 \);
- \( \ell_{15} \) may still get an active region \([0, 21 - \epsilon] \times [21 - \epsilon, 24] \) of size \( 63 + 18\epsilon - \epsilon^2 \).

Therefore, the greedy heuristic would now select \( \ell_{10} \), which intersects all other labels except \( \ell_1, ..., \ell_3 \). Thus, the activity regions of the remaining labels are now restricted as follows:

- \( \ell_1, ..., \ell_4 \) are still confined to the rectangle \([0, 8] \times [8, 8 + 2\epsilon] \) of size \( 16\epsilon \);
In the last step, we used $b - 1 \geq \log_2 b = n$.

\[
\frac{b^2}{2} + \sum_{j=2}^{n-1} \frac{b^2 - 2^j}{2} + (b^2 - 2) = \frac{(n + 1)b^2}{2} - 2^{n-1} = \frac{(n + 1)b^2 - b}{2} = \frac{nb^2 + b(b - 1)}{2} > \frac{nb^2 + nb}{2}.
\]

In the last step, we used $b - 1 \geq \log_2 b = n$. 

** Instances** With events of different weights we can even construct input instances that cause the greedy heuristic’s performance to become arbitrarily bad. Choose an interval $[1, b^2]$ from which to pick the weights, such that $b$ is an integral power of two, larger than 1. Let $n$ be $\log_2 b$. Let $t_{\min} = 0$ and $t_{\min} = b^2$. We create $n$ events $e_1, ..., e_n$ in the time window $[0, b^2]$, where $e_i$ has weight $\frac{1}{b^2}$; the events $e_j$, for $j \in \{1, ..., n - 1\}$, have weight $2^{-j}$; each event $e_j$, for $j \in \{1, ..., n\}$, has timestamp $2^j$, and all labels have the same location; see Figure 6 for $b = 16$.

** Optimal Solution** Note that $e_n$ has maximum volume $\frac{1}{b-1}2^n(b^2 - 2^n) = \frac{1}{b^2-1}b(b^2 - b) = b^2$, whereas each other event $e_j$ has maximum volume $2^{-j}2^j(b^2 - 2^j) = b^2 - 2^j$. The optimal solution would contain at least the right half of each event’s maximum possible region (and for $e_1$, also the left half); the total volume will be roughly $\frac{1}{b}nb^2$. More precisely, the total volume of this solution would indeed be:

\[
\frac{b^2}{2} + \sum_{j=2}^{n-1} \frac{b^2 - 2^j}{2} + (b^2 - 2) = \frac{(n + 1)b^2}{2} - 2^{n-1} = \frac{(n + 1)b^2 - b}{2} = \frac{nb^2 + b(b - 1)}{2} > \frac{nb^2 + nb}{2}.
\]
**Greedy Solution**  The greedy heuristic, however, would first give \( e_n \) its maximum possible region. This reduces the maximum height of the activity region of each other event \( e_j \) from \( b^2 - 2^j \) to \( 2^n - 2^j = b - 2^j \); thus its maximum volume is reduced to:

\[
2^{-j}(b - 2^j) = 2^{-j}b - 1 = 2^{n-j} - 1 < 2^{n-j},
\]

and the maximum total volume of all events is reduced to less than:

\[
b^2 + \sum_{j=1}^{n-1} 2^{n-j} < b^2 + 2^n = b^2 + b\]

**Approximation Ratio**  Thus, the greedy heuristic’s solution is worse than the optimal solution by a factor of at least:

\[
\frac{(nb^2 + nb)/2}{b^2 + b} = \frac{n}{2}.
\]

Note that the factor \( n/2 \) is reached under the condition \( n = \log_2 b \), or conversely, \( b = 2^n \). In other words, the construction requires events whose weight differences are exponential in \( n \). Where this is not realistic, the lower bound might better be expressed in terms of \( b \), as we do in the next subsection.

### 3.3 A refined construction of the lower bound

We can extend the construction given above to labels that do not all have the same location. Fix numbers \( a \geq 1 \) and \( b \geq 1 \), where \( b = 2^m \) for some integer \( m \); we will construct a set \( E \) of \( n = (a + 1)m \) events with degree of interference \( a \) and degree of unbalance \( b \). Concretely, let \( E \) consist of \((a + 1) \times m \) events \( e_{i,j} \), for \( i \in \{0, ..., a\} \) and \( j \in \{1, ..., m\} \), with timestamps in the time window \([0,b^2]\). For all \( i \in \{0, ..., a\} \), event \( e_{i,m} \) has weight \( \frac{1}{b-1} \); the events \( e_{i,j} \), for \( j \in \{1, ..., m-1\} \), have weight \( 2^{-j} \); all events \( e_{i,j} \), for \( j \in \{1, ..., m\} \), have timestamp \( 2^j \). For each \( i \), the labels \( \ell_{i,j} \) have the same locations, such that \( \ell_{i,j} \) intersects all other labels \( \ell_{i,j'} \), but these labels do not intersect each other. Thus, two different events \( e_{g,h} \) and \( e_{i,j} \) are in conflict if and only if \( g = 0 \), \( i = 0 \), or \( g = i \). Note that the maximum volume for any event \( e_{i,j} \) with \( j \in \{1, ..., m-1\} \) is \( 2^{-j}2^j(b^2 - 2^j) = b^2 - 2^j \); the maximum volume for any event \( e_{i,m} \) is \( \frac{1}{b-1}2^m(b^2 - 2^m) = \frac{1}{b-1}b(b^2 - b) = b^2 \).

**Optimal Solution**  There is a solution that places, at each point of the diagram, the \( a \) events of highest weight that are in range and are not in conflict with each other. The total volume is thus:

\[
\begin{align*}
a2^{-1}2^1(b^2 - 2^1) + a \left( \sum_{j=2}^{m-1} 2^{-j}(2^j - 2^{j-1})(b^2 - 2^j) \right) + a \frac{1}{b-1}(2^m - 2^{m-1})(b^2 - 2^m) \\
= \frac{(a/2)((m+1)b^2 - b)}{b^2} \\
= \Omega(ab^2 \log b).
\end{align*}
\]

**Greedy Solution**  The greedy heuristic however, could start with giving event \( e_{0,m} \) its maximum region, thus eliminating the events \( e_{i,m} \) for \( i \in \{1, ..., a\} \) completely, and reducing the maximum size of the other labels’ activity regions by a factor at least \( b \). Moreover, in the following steps, the greedy heuristic could always pick an event \( e_{0,j} \), thus eliminating \( e_{i,j} \) for \( i \in \{1, ..., a\} \). In the end, the greedy solution will have total volume at most:

\[
\begin{align*}
2^{-1}2^1(2^m - 2^1) + \left( \sum_{j=2}^{m-1} 2^{-j}(2^j - 2^{j-1})(2^m - 2^j) \right) + b^2 \\
= (1/2)(2b^2 + mb - b) \\
= O(b^2).
\end{align*}
\]
Approximation Ratio  Thus, the approximation ratio of the greedy heuristic is at least $\Omega(a \log b)$ in the worst case, which proves the lower bound stated in [Theorem 1]. Note that with a given number of events $n$, the lower bound construction can only be realized as long as $(a+1) \log b \leq n$, since the construction requires this many labels. If $(a+1) \log b > n$, we can only do the construction for a smaller degree of interference $a'$ and a smaller degree of interference $b'$ such that $(a'+1) \log b' = \Theta(n)$, and the approximation ratio of the greedy heuristic is $\Omega(a' \log b') = \Omega(n)$. Thus, the lower bound on the worst-case approximation ratio is $\Omega(a \log b)$ or $\Omega(n)$, whatever is lower. In the next section we will see that this lower bound is tight up to constant factors.

4 Upper bound on the approximation ratio of the greedy heuristic

In this section, we derive an upper bound on the approximation ratio of the greedy heuristic and hence, together with the results of Section 3, we prove Theorem 1. Without loss of generality, let the time scale run from 0 to 1. Let $T^*$ be an arbitrary optimal solution for $E$. Let $e_i \in E$ and $\tau_i^*$ be the activity region of $e_i$ in $T^*$, let $\tau_i^G$ be the activity region of $e_i$ in the greedy solution, and let $\tau_i^0$ be the maximum possible activity region of $e_i$, that is, the rectangle $[0, t_i] \times [t_i, 1]$. Our goal is now to determine an approximation ratio, that is, to determine a factor $k$ (ideally as low as possible) such that the following holds for any set of events $E$:

$$\sum_{e_i \in E} \frac{v(\tau_i^*)}{v(\tau_i^G)} \leq k.$$  

Charging: In order to prove the upper bound, we need to introduce the concept of charging. Let $e_i$ be an event in $E$. From now on, we define each event to be in conflict with itself. Let $e_j$ be the first-placed event in the greedy solution that is in conflict with $e_i$ and whose active region $\tau_i^G$ intersects $\tau_i^0$, that is, among all events of $E$ that are in conflict with $e_i$, the event $e_j$ is the first to be extracted from the priority queue by the greedy heuristic. Such an event $e_j$ always exists; it might be $e_i$ itself. We say $e_i$ charges $v(\tau_i^*)$ to $e_j$.

By this charging, we model the following circumstances: Before $e_j$ is selected, the event $e_i$ could still get $\tau_i^0$ as its activity region. However, before (or when) the greedy heuristic selects $e_i$, it selects $e_j$, so we know we must have $v(\tau_i^0) \leq v(\tau_i^G)$. After selecting $e_j$, the event $e_i$ cannot get an (other) activity region of size $\tau_i^0$ anymore. We “blame” $e_j$ for that by charging $v(\tau_i^*)$ to $e_j$. Note that the amount charged is only $v(\tau_i^*)$, not $v(\tau_i^0)$.

Bounding the charges: Now consider a given event $e_j$. Let $C_j$ be the set of events that charge to $e_j$. We will calculate an upper bound on $\sum_{e_i \in C_j} v(\tau_i^*)$, that is, the total charge to $e_j$, summed over all events in $C_j$. In fact, we will calculate an upper bound on $\sum_{e_i \in C_j} w_i \cdot \area(\tau_i)$ that holds for any valid solution, and which is therefore also an upper bound on $\sum_{e_i \in C_j} v(\tau_i^*)$.

To derive this bound, we divide the triangle that contains the activity diagram into six regions $Q_1, ..., Q_6$, calculate a bound on $\sum_{e_i \in C_j} w_i \cdot \area(\tau_i \cap C_h)$ for each $h \in \{1, ..., 6\}$, and add up the bounds. The six regions are determined as follows. Let $t_L \leq t_U \leq 1/2$ be such that

$$t_L(1-t_L) = \frac{1}{b} \cdot \area(\tau_i^G),$$
and

$$t_U(1-t_U) = b \cdot \area(\tau_i^G).$$

if such a $t_U$ exists (that is, if $b \cdot \area(\tau_i^G) \leq 1/4$); otherwise $t_U = 1/2$. Note:

$$\frac{t_U}{t_L} = \frac{b \cdot \area(\tau_i^G)/(1-t_U)}{b \cdot \area(\tau_i^G)/(1-t_L)} = b^2 \frac{1-t_L}{1-t_U} \leq b^2 \frac{1}{1/2} = 2b^2,$$

and therefore: $t_U < 2b^2 t_L$. Using $t_L$ and $t_U$, we can now define the six regions $Q_1, ..., Q_6$ as illustrated in Figure 7. The idea of this subdivision in regions is that it distinguishes between three types of regions in the activity diagram. In $Q_4$ and $Q_5$ we find events whose maximum possible activity regions would have so much volume, that they would be selected before $e_i$ and therefore cannot charge to $e_j$. On the other extreme, charges from $Q_3$ and $Q_6$ are possible, but small, because these regions are too narrow to carry much volume. In between there are the regions $Q_2$ and $Q_5$, which cover charges to $e_j$ from events $e_i$ with
maximum possible active regions similar to $e_i$. We will now analyze $\sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i \cap C_h)$ for $h = 1, 2, 3$ in detail; the analysis for $h = 4, 5, 6$ is symmetric.

**Case $h = 1$** First consider a time-window query $q = (t', t'')$ in the activity diagram with $t' + t'' \leq 1$ (therefore, $t' \leq 1/2$) and $t' > t_U$, that is, a point $q \in Q_1$ in Figure 7. Now, we look at an event $e_i$ that is active at $q$. For such an event $e_i$ it holds that $t_U < t_i < 1 - t_U$. Hence, we have $\text{area}(\tau^0_i) = t_i(1 - t_i)$ and it holds that $t_i(1 - t_i) > t_U(1 - t_U)$. Thus we find:

$$\text{area}(\tau^0_i) = t_i(1 - t_i) > t_U(1 - t_U) = b \cdot \text{area}(\tau^G_j)$$

$$\Rightarrow v(\tau^0_i) \geq (w_j/b) \cdot \text{area}(\tau^0_i) > (w_j/b) \cdot b \cdot \text{area}(\tau^G_j) = v(\tau^G_j).$$

Since we assumed that $e_i$ charges to $e_j$ which implies $v(\tau^0_i) \leq v(\tau^G_j)$, such an event $e_i$ cannot be in $C_j$. Otherwise, the greedy heuristic would have given $e_i$ its active region before $e_j$. Thus, the total volume of labels in $C_j$, intersected with the set of points $Q_1 = \{(t', t'') \mid 0 \leq t' \leq t'' \leq 1 \text{ and } t' + t'' \leq 1 \text{ and } t_U \leq t''\}$, is:

$$\sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i \cap Q_1) = 0.$$

**Case $h = 2$** Consider a time-window query $q = (t', t'')$ with $t' + t'' \leq 1$ and $t_L \leq t' \leq t_U$, that is, a point $q \in Q_2$ in Figure 7. In any valid solution, the total number of events in $C_j$ (whose labels all intersect $\ell_j$) that are active at $q$ is, by definition, at most $a$. Furthermore, the weight $v(\tau^G_i)/\text{area}(\tau^0_i)$ of any such event $e_i$ must be at most $v(\tau^G_i)/\text{area}(\tau^0_i) \leq v(\tau^G_i)/(t'(1 - t'))$. Therefore, the total weight of the events active at $q$ is at most $a \cdot v(\tau^G_i)/(t'(1 - t'))$. Thus, the total volume of the activity regions of labels in $C_j$, intersected
with the set of points \( Q_2 = \{(t', t'') \mid 0 \leq t' \leq t'' \leq 1 \text{ and } t' + t'' \leq 1 \text{ and } t_L \leq t' \leq t_U \} \), is at most
\[
\sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i \cap Q_2) = a \cdot v(\tau_j^G) \int_{t_L}^{t_U} \left( \int_{t'}^{1-t'} \frac{1}{t'(1-t')} dt'' \right) dt'.
\]
\[
= a \cdot v(\tau_j^G) \int_{t_L}^{t_U} \frac{1-2t'}{t'(1-t')} dt'
\]
\[
= a \cdot v(\tau_j^G) \int_{t_L}^{t_U} \frac{1}{t'} - \frac{1}{1-t'} dt'
\]
\[
< a \cdot v(\tau_j^G) \int_{t_L}^{t_U} \frac{1}{t'} dt'
\]
\[
= a \cdot v(\tau_j^G) \cdot \ln \frac{t_U}{t_L}
\]
\[
< a \cdot v(\tau_j^G) \cdot (2b^2)
\]
\[
= a \cdot v(\tau_j^G) \cdot (\ln 2 + 2 \ln b).
\]

**Case \( h = 3 \)** Finally, consider a time-window query \( q = (t', t'') \) with \( t' + t'' \leq 1 \) and \( t' < t_L \); that is, a point \( q \in Q_3 \) in Figure 7. We observe that they all lie in the trapezoid with vertices \((0, 0), (t', t_L), (t_L, 1-t_L), (0, 1)\), whose size is \( t_L(1-t_L) = \text{area}(\tau_j^G)/b \), and the weight of any event of \( C_j \) active at \( q \) is at most \( b \cdot w_j = b \cdot v(\tau_j^G) / \text{area}(\tau_j^G) \). Thus, the total volume of the activity regions of the labels in \( C_j \), intersected with the set of points \( Q_3 = \{(t', t'') \mid 0 \leq t' \leq t'' \leq 1 \text{ and } t' + t'' \leq 1 \text{ and } t' < t_L \} \) is at most
\[
\sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i \cap Q_3) = a \cdot \frac{\text{area}(\tau_j^G)}{b} \cdot \frac{b \cdot v(\tau_j^G)}{\text{area}(\tau_j^G)} = a \cdot v(\tau_j^G).
\]

**Adding it up** Together, the sets of points \( Q_h \) considered in cases 1, 2 and 3 and the symmetric cases cover the entire active diagram, that is:
\[
\{ (t', t'') \mid 0 \leq t' \leq t'' \leq 1 \} = \bigcup_{h=1}^{6} Q_h.
\]

Thus, in any valid solution, the total volume of the activity regions of the labels in \( C_j \) is at most
\[
\sum_{e_i \in C_j} v(\tau_i) = \sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i) = \sum_{h=1}^{6} \sum_{e_i \in C_j} w_i \cdot \text{area}(\tau_i \cap Q_h)
\]
\[
\leq 2 \cdot a \cdot v(\tau_j^G) \cdot ( \ln 2 + 2 \ln b ) + 2 \cdot a \cdot v(\tau_j^G)
\]
\[
= a \cdot v(\tau_j^G) \cdot (2 \ln 2 + 4 \ln b + 2).
\]

This holds for any valid solution, so it also holds for the optimal solution \( T^*(E) \) and we get:
\[
\sum_{e_i \in C_j} v(\tau_i^*) \leq a \cdot v(\tau_j^G) \cdot (2 \ln 2 + 4 \ln b + 2).
\]

Note that each event \( e_i \) charges to only one event \( e_j \), and thus occurs in only one set \( C_j \). Thus we find:
\[
\sum_{e_i \in E} v(\tau_i^*) = \sum_{e_j \in E} \sum_{e_i \in C_j} v(\tau_i^*) \leq a \cdot (2 \ln 2 + 4 \ln b + 2) \cdot \sum_{e_j \in E} v(\tau_j^G).
\]

This concludes the proof of approximation ratio \( O(a \log b) \). Thus, the approximation ratio is \( \Theta(a \log b) \) and hence, we prove [Theorem 1] Note that the approximation ratio can never be worse than \( n \), as the first active region chosen by the greedy heuristic has at least as much volume as any active region in the optimal solution. Moreover, if all labels have equal weight \( (b = 1) \), the ratio \( t_U/t_L \) in the above calculation becomes 1, that is, the cases \( h = 2 \) and \( h = 5 \) disappear, and thus, the terms \( (2 \ln 2 + 4 \ln b) \) disappear from the final bound.
Corollary 1. If all labels are unit squares of equal weight, the approximation ratio is at most 8. If all labels are unit disks of equal weight, the approximation ratio is at most 10.

Proof. The maximum number of mutually disjoint unit squares or disks that can intersect a given unit square or disk, respectively, is at most four or five, respectively.

References

[1] P. K. Agarwal, M. J. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry, 11*(3-4):209–218, 1998. doi: 10.1016/S0925-7721(98)00028-5

[2] D. Bahrdt, M. Becher, S. Funke, F. Krumpe, A. Nusser, M. Seybold, and S. Storandt. Growing balls in $\mathbb{R}^d$. In *Proceedings of 19th Workshop on Algorithm Engineering and Experiments (ALENEX’17)*, pp. 247–258. SIAM, 2017. doi: 10.1137/1.9781611974768.20

[3] M. J. Bannister, W. E. Devanny, M. T. Goodrich, J. A. Simons, and L. Trott. Windows into geometric events: Data structures for time-windowed querying of temporal point sets. In *Proceedings of 26th Canadian Conference on Computational Geometry (CCCG 2014)*, 2014.

[4] M. J. Bannister, C. DuBois, D. Eppstein, and P. Smyth. Windows into relational events: Data structures for contiguous subsequences of edges. In *Proceedings of 24th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA’13)*, pp. 856–864. SIAM, 2013. doi: 10.1137/1.9781611973105.61

[5] L. Barth, B. Niedermann, M. Nöllenburg, and D. Strash. Temporal Map Labeling: A New Unified Framework with Experiments. In *Proceedings of 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL’16)*, pp. 23:1–23:10. ACM, 2016. doi: 10.1145/2996913.2996957

[6] K. Been, E. Daiches, and C. Yap. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics, 12*(5):773–780, 2006. doi: 10.1109/TVCG.2006.136

[7] P. Bobák, L. Cmolík, and M. Cadík. Temporally stable boundary labeling for interactive and non-interactive dynamic scenes. *Computers & Graphics, 91*:265–278, 2020. doi: 10.1016/j.cag.2020.08.005

[8] D. Bokal, S. Cabello, and D. Eppstein. Finding All Maximal Subsequences with Hereditary Properties. In *Proceedings of Symposium on Computational Geometry (SoCG’15)*, vol. 34 of LIPIcs, pp. 240–254. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. doi: 10.4230/LIPIcs.SOCG.2015.240

[9] A. Bonerath, A. Driemel, J.-H. Haunert, H. Haverkort, E. Langetepe, and B. Niedermann. Algorithms for consistent dynamic labeling of maps with a time-slider interface. Submitted to IEEE Transactions on Visualization and Computer Graphics, 8 2022.

[10] A. Bonerath, B. Niedermann, J. Diederich, Y. Orgeig, J. Oehrlein, and J. Haunert. A time-windowed data structure for spatial density maps. In *Proceedings of 28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL’20)*, pp. 15–24. ACM, 2020. doi: 10.1145/3397536.3422242

[11] A. Bonerath, B. Niedermann, and J. Haunert. Retrieving alpha-shapes and schematic polygonal approximations for sets of points within queried temporal ranges. In *Proceedings of 27th International Conference on Advances in Geographic Information Systems SIGSPATIAL 2019*, pp. 249–258. ACM, 2019. doi: 10.1145/3347146.3359087

[12] T. M. Chan and S. Pratt. Time-windowed closest pair. In *Proceedings of Canadian Conference on Computational Geometry (CCCG’15)*, 2015.

[13] T. M. Chan and S. Pratt. Two approaches to building time-windowed geometric data structures. In *Proceedings of Symposium on Computational Geometry (SoCG’16)*, vol. 51 of LIPIcs, pp. 28:1–28:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. doi: 10.4230/LIPIcs.SoCG.2016.28
[14] F. Chanchary and A. Maheshwari. Time windowed data structures for graphs. *Journal of Graph Algorithms and Applications*, 23(2):191–226, 2019. doi: 10.7155/jgaa.00489

[15] F. Chanchary, A. Maheshwari, and M. Smid. Window queries for problems on intersecting objects and maximal points*. In *Algorithms and Discrete Applied Mathematics*, pp. 199–213. Springer, 2018. doi: 10.1007/978-3-319-74180-2_17

[16] F. Chanchary, A. Maheshwari, and M. Smid. Querying relational event graphs using colored range searching data structures. *Discrete Applied Mathematics*, 2019. doi: 10.1016/j.dam.2019.03.006

[17] A. Gemsa, B. Niedermann, and M. Nöllenburg. A unified model and algorithms for temporal map labeling. *Algorithmica*, 82(10):2709–2736, 2020. doi: 10.1007/s00453-020-00694-7

[18] J.-H. Haunert and A. Wolff. Beyond maximum independent set: an extended integer programming formulation for point labeling. *ISPRS International Journal of Geo-Information*, 6(11):342, 2017. doi: 10.3390/ijgi6110342

[19] T. Kapler and W. Wright. Geotime information visualization. *Information Visualization*, 4(2):136–146, 2005. doi: 10.1057/palgrave.ivs.9500097

[20] M. Meijers, P. van Oosterom, M. Driel, and R. Šuba. Web-based dissemination of continuously generalized space-scale cube data for smooth user interaction. *International Journal of Cartography*, 6(1):152–176, 2020. doi: 10.1080/23729333.2019.1705144

[21] D. Peng, A. Wolff, and J. Haunert. Finding optimal sequences for area aggregation: A* vs. integer linear programming. *ACM Transactions on Spatial Algorithms and Systems*, 7(1):4:1–4:40, 2020. doi: 10.1145/3409290

[22] P. Yoeli. The logic of automated map lettering. *The Cartographic Journal*, 9(2):99–108, 1972.

[23] X. Zhang, S. Poon, S. Liu, M. Li, and V. C. S. Lee. Consistent dynamic map labeling with fairness and importance. *Computer Aided Geometric Design*, 81:101892, 2020. doi: 10.1016/j.cagd.2020.101892