CONSISTENT LABELING OF ROTATING MAPS*

Andreas Gemsa, Martin Nöllenburg,† Ignaz Rutter‡

Abstract. Dynamic maps that allow continuous map rotations, for example, on mobile devices, encounter new geometric labeling issues unseen in static maps before. We study the following dynamic map labeling problem: The input is an abstract map consisting of a set P of points in the plane with attached horizontally aligned rectangular labels. While the map with the point set P is rotated, all labels remain horizontally aligned. We are interested in a *consistent* labeling of P under rotation, i.e., an assignment of a single (possibly empty) *active* interval of angles for each label that determines its visibility under rotations such that visible labels neither intersect each other (soft conflicts) nor occlude points in P at any rotation angle (hard conflicts). Our goal is to find a consistent labeling that maximizes the number of visible labels integrated over all rotation angles.

We first introduce a general model for labeling rotating maps and derive basic geometric properties of consistent solutions. We show NP-hardness of the above optimization problem even for unit-square labels. We then present a constant-factor approximation for this problem based on line stabbing, and refine it further into an efficient polynomial-time approximation scheme (EPTAS).

1 Introduction

Dynamic maps, in which the user can navigate continuously through space, are becoming increasingly important in scientific and commercial GIS applications as well as in personal mapping applications. In particular, GPS-equipped mobile devices offer various new possibilities for interactive, location-aware maps. A common principle in dynamic maps is that users can pan, rotate, and zoom the map view. Despite the popularity of several commercial and free applications, relatively little attention has been paid to provably good labeling algorithms for dynamic maps.

Been et al. [2] identified a set of consistency desiderata for dynamic map labeling. Labels should neither "jump" (suddenly change position or size) nor "pop" (appear and disappear more than once) during monotonous map navigation; moreover, the labeling should be a function of the selected map viewport and not depend on the user's navigation history. Previous work on the topic has focused solely on supporting zooming and/or panning of the

*A preliminary version of this paper was presented at the 12th International Symposium on Algorithms and Data Structures (WADS 2011) [7].

†*Algorithms and Complexity Group, TU Wien, Vienna, Austria*, noellenburg@ac.tuwien.ac.at

‡*Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*, rutter@kit.edu

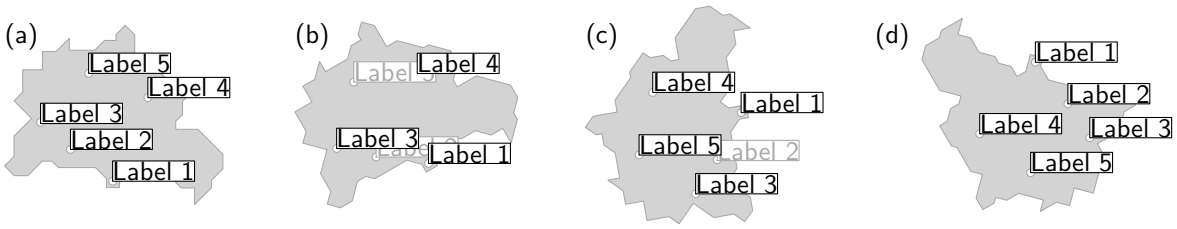


Figure 1: Input map with five points (a) and three rotated views with some partially occluded labels (b)–(d).

map [2, 3, 15, 19], whereas consistent labeling under map rotations has not been considered prior to the conference version of this paper [7].

Most maps come with a natural default orientation (usually the northern direction facing upward), but applications such as car or pedestrian navigation often rotate the map view dynamically to be always forward facing [10]. Still, the labels, usually modeled as rectangles, must remain horizontally aligned for best readability regardless of the actual rotation angle of the map. A basic requirement in static and dynamic label placement is that labels are pairwise disjoint and thus form an independent set of rectangles, which means that in general not all labels can be placed simultaneously. For labeling point features unambiguously, it is further required that each label rectangle contains the labeled point (also called *anchor point*), either in its inside or on its boundary. Further, we do not allow that labels occlude other input points, even if the points are unlabeled at a particular rotation angle. Figure 1 shows a sketch of a map that is rotated and labeled, depending on the rotation angle. The objective in map labeling is usually to place as many labels as possible. Translating this into the context of rotating maps means that, integrated over one full rotation from 0 to 2π, we want to maximize the number of visible labels. The consistency requirements of Been et al. [2] can immediately be applied to rotating maps.

Our Results. Initially, we define a model for rotating maps and show some basic properties of the different types of conflicts that may arise during rotation. Next, we prove that consistently labeling rotating maps is NP-hard, both for the maximization of the total number of visible labels integrated over one full rotation and for the maximization of the smallest visibility range of any label. Under the assumption of a bounded feature density in the input map we present a 1/4-approximation algorithm and an efficient polynomial-time approximation scheme (EPTAS) for unit-height rectangles[1]. We extend both algorithms to the case of rectangular labels with the property that the ratio of the smallest and largest width, the ratio of the smallest and largest height, as well as the aspect ratio of every label is bounded by a constant, even if we allow the anchor point of each label to be an arbitrary point of the label. This applies to most practical scenarios, where labels typically consist of few and relatively short lines of text. Moreover, our algorithmic results also apply to the extensions of the labeling model for rotating maps introduced in our follow-up paper [9].

[1] A PTAS is called *efficient* if its running time is $O(f(\varepsilon) \cdot n^c)$ for some constant c independent of ε.

Related Work. Most previous algorithmic research efforts on automated label placement cover *static* labeling models for point, line, or area features. For static point labeling, fixed-position models and slider models have been introduced [5, 21], in which the label, represented by its bounding box, needs to touch the labeled point with one of its corners or along its boundary. The label number maximization problem is NP-hard even for the simplest labeling models [12, 17, 21], whereas there are efficient algorithms for the decision problem that asks whether all points can be labeled in some of the simpler models. This decision problem is NP-hard in the fixed-position model with four different positions, but is in P if only two positions per label are allowed [5]. Approximation results [1, 4, 21], heuristics [23], and exact approaches [13, 22] are known for many variants of the static label number maximization problem. In addition to the label number maximization problem, there is also the label size maximization problem, where *all* labels need to be placed and the optimization goal is to find the largest factor by which all labels can be scaled such that not two labels overlap each other. As for the label size maximization problem, approximation algorithms are known [5, 23].

In recent years, *dynamic* map labeling has emerged as a new research topic that gives rise to many unsolved algorithmic problems. Petzold et al. [20] used a preprocessing step to generate a reactive conflict graph that represents possible label overlaps for maps of all scales. For any fixed scale and map region, their method computes a conflict-free labeling using static labeling heuristics. Mote [18] presents another fast heuristic method for dynamic conflict resolution in label placement that does not require preprocessing. The consistency desiderata of Been et al. [2] for dynamic labeling (no popping and jumping effects during panning and zooming), however, are not satisfied by either of the methods as they were mainly designed for quickly labeling a static map of arbitrary scale. Been et al. [3] showed NP-hardness of the label number maximization problem in the consistent labeling model and presented several approximation algorithms. Gemsa et al. [8] considered a slider model and presented an FPTAS for labeling a one-dimensional zoomable point set, where for each label not only a consistent visibility range but also an optimal slider position for all zoom levels needs to be found. Nöllenburg et al. [19] studied a dynamic version of the alternative boundary labeling model, in which labels are placed at the sides of the map and connected to their points by leaders. They presented an algorithm to precompute a data structure that represents an optimal one-sided labeling for all possible scales and thus allows continuous zooming and panning. None of the existing dynamic map labeling approaches supports map rotation.

After the publication of the conference version of this paper, rotating maps have been further studied. Gemsa et al. [6] discussed the problem of labeling a rotating and translating map, where only a small part of the map can be seen on screen. This is a problem that manufacturers of modern GPS navigation devices face when implementing the algorithms for displaying the map. They presented an NP-hardness proof as well as an approximation algorithm for unit-square labels. The restricted case that no more than k labels may be shown simultaneously can be solved efficiently. Yokosuka and Imai [24] investigated the label size maximization problem for rotating maps. They presented efficient algorithms for finding anchor points inside or on the boundary of the labels such that the label size is maximized and no conflicts occur during a full rotation. Finally, in a follow-up paper [9],

we have recently experimentally evaluated several labeling strategies for rotating maps and compared the achievable activity levels for different degrees of consistency in the sense of Been et al. [2]. Our results confirmed that the model we propose in this paper offers a good trade-off between consistency and label activity. We further evaluated the performance of the 1/4-approximation algorithms proposed in this paper in comparison to additional heuristics and an exact ILP approach.

2 Model

In this section we describe our labeling model for rotating maps with axis-aligned rectangular labels. Although this paper focuses on a theoretical understanding of one particular model, the one introduced in the conference version of this paper, our presentation is inspired by the more general family of models from the follow-up paper [9].

Let M be an (abstract) map, consisting of a set $P = \{p_1, \ldots, p_n\}$ of points in the plane together with a set $L = \{\ell_1, \ldots, \ell_n\}$ of closed, axis-aligned, and not necessarily disjoint rectangular labels in the plane. Each point p_i must lie inside (or on the boundary of) its corresponding label ℓ_i at an arbitrary but fixed position; we denote p_i as the *anchor* of label ℓ_i and ℓ_i as *anchored* at p_i. Note that we do not allow movement of a label relative to its anchor, i.e., ℓ_i must maintain the same position relative to p_i.

As M rotates, each label ℓ_i in L must remain horizontally aligned and anchored at p_i. Thus, new label intersections form and existing ones disappear during the rotation of M. We take the following alternative perspective on the rotation of M. Rather than rotating the point set P, say clockwise, and keeping the labels horizontally aligned, we may instead rotate each label counterclockwise around its anchor point and keep the set of points fixed. It is easy to see that both rotations are equivalent in the sense that they yield exactly the same intersections of labels and occlusions of points.

We consider all rotation angles modulo 2π. For convenience we introduce the interval notation (a, b) for any two angles $a, b \in [0, 2\pi]$. If $a \leq b$, this corresponds to the standard meaning of an open interval, otherwise, if $a > b$, we define $(a, b) := (a, 2\pi] \cup [0, b)$. For simplicity, we refer to any set of the form (a, b) as an interval. We further define the length of an interval I as $|I| = b - a$ if $I = (a, b)$ with $a \leq b$ and $|I| = 2\pi - a + b$ if $I = (a, b)$ with $a > b$. We further extend the above definitions to the standard interval notation for half-open and closed intervals of the form $(a, b]$, $[a, b)$, and $[a, b]$.

A *rotation* of L is defined by a rotation angle $\alpha \in [0, 2\pi)$. We define the rotated label set $L(\alpha)$ of all labels, each rotated by an angle of α around its anchor point. A *rotation labeling* of M is a function $\phi\colon L \times [0, 2\pi) \to \{0, 1\}$ such that $\phi(\ell, \alpha) = 1$ if label ℓ is visible or *active* in the rotation of L by α, and $\phi(\ell, \alpha) = 0$ otherwise. We call a labeling ϕ *valid* if, for any rotation α, the set of labels $L_\phi(\alpha) = \{\ell \in L(\alpha) \mid \phi(\ell, \alpha) = 1\}$ consists of pairwise disjoint labels. If two labels ℓ and ℓ' in $L(\alpha)$ intersect, we say that they have a *soft conflict* (or a *label-label conflict*) at α, i.e., in a valid labeling at most one of them can be active at α. We define the set $C(\ell, \ell') = \{\alpha \in [0, 2\pi) \mid \ell \text{ and } \ell' \text{ are in conflict at } \alpha\}$ as the *conflict set* of ℓ and ℓ'. Further, we call the begin and end of a maximal contiguous conflict range in $C(\ell, \ell')$ *conflict* or *label events*.

For a label ℓ we call each maximal interval $I \subseteq [0, 2\pi)$ with $\phi(\ell, \alpha) = 1$ for all $\alpha \in I$ an *active range* of label ℓ and define the set $A_\phi(\ell)$ as the set of all active ranges of ℓ in ϕ. We call an active range where both boundaries are conflict events a *regular* active range. Our optimization goal is to find a valid labeling ϕ that maximizes the number of active labels integrated over one full rotation from 0 to 2π. The value of this integral is called the *total activity* $t(\phi)$ and can be computed as $t(\phi) = \sum_{\ell \in L} \sum_{I \in A_\phi(\ell)} |I|$.

A valid labeling is not yet consistent in terms of the definition of Been et al. [2, 3]: for given fixed anchor points, labels clearly do not jump and the labeling is independent of the rotation history, but labels may still *flicker* multiple times during a full rotation from 0 to 2π, depending on how many active ranges they have in ϕ. To reduce flickering effects, we allow each label to have only a single active range. Accordingly, we call such a labeling *consistent*. Since a consistent labeling allows only one active range, this is called the *1R-model*.

We apply another restriction to our consistency model, which is based on the occlusion of anchors. Among the conflicts in set $C(\ell, \ell')$ we further distinguish *hard conflicts* (or *label-point conflicts*), i.e., conflicts where label ℓ intersects the anchor of label ℓ'. In our definition every hard conflict is also a soft conflict as clearly the two labels intersect as well. If a labeling ϕ sets ℓ active during a hard conflict with ℓ', the anchor of ℓ' is occluded. This may be undesirable in some situations in practice, for example, if every point in P carries useful information in the map, even if it is currently unlabeled. Thus we may optionally require that $\phi(\ell, \alpha) = 0$ during any hard conflict of a label ℓ with another label ℓ' at angle α. In that sense, the occluded anchor dictates that the occluding label cannot be active.

The objective in static map labeling is usually to find a maximum subset of pairwise disjoint labels, i. e., to label as many points as possible. Generalizing this objective to rotating maps means that integrated over all rotations $\alpha \in [0, 2\pi)$ we want to display as many labels as possible. This corresponds to finding a valid rotation labeling ϕ with maximum total activity $t(\phi)$ over all valid rotation labelings; we call this optimization problem MaxTotal. An alternative objective is to maximize over all valid rotation labelings ϕ the minimum length $\min\{|I| \in A_\phi(\ell) \mid \ell \in L\}$ of all active ranges; this problem is called MaxMin.

We note that the 1R-model readily generalizes to the *kR-model* that allows at most k active ranges per label. The version without any consistency constraints is called *∞R-model*; it mainly serves as an upper bound for the other models. Finally, a more drastic approach that eliminates all flickering is the 0/1-model, where every label is either active for the full rotation $[0, 2\pi)$ or never at all. Our experimental evaluation [9] showed that the 0/1-model is too restrictive as it often only achieves 50% or less of the label activity of an optimal solution in the ∞R-model, whereas an optimal solution for the 1R-model achieves between 80% and 95% of the maximum activity in the ∞R-model for several real-world instances. Hence the 1R-model is a reasonable choice that offers good solutions while containing sufficient structure for a theoretical treatment.

For this paper we consider the 1R hard-conflict model only; whenever we use the term "optimal rotation labeling", this refers to a valid optimal rotation labeling in this particular model. We first give some complexity results and then develop approximation algorithms, most notably an EPTAS for the case that the rectangular labels have bounded

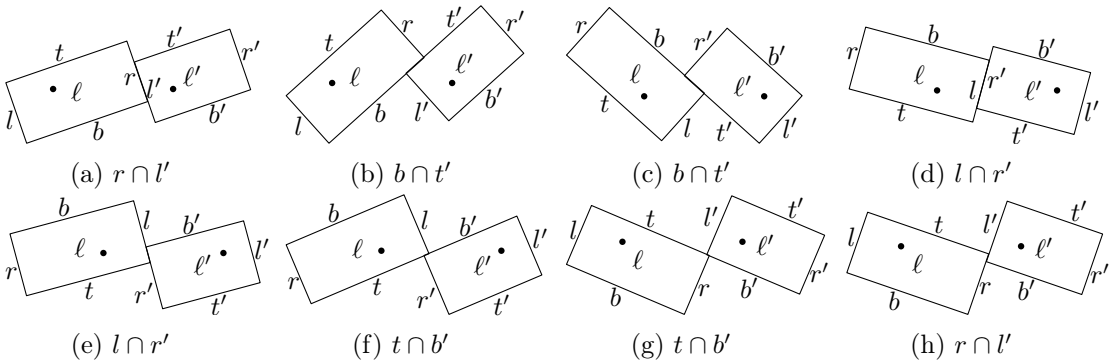


Figure 2: Two labels ℓ and ℓ' and their eight possible boundary intersection events. Anchor points are marked as black dots.

width ratio, bounded height ratio and bounded aspect ratio. It should be noted that, while our hardness results rely on both the hard conflicts and the restriction to a single active range, our algorithmic results do not have this limitation and generalize to the kR-model with or without explicit hard conflicts.

3 Properties of Rotation Labelings

In this section we show basic properties of rotation labelings. If two labels ℓ and ℓ' intersect in a rotation of α, they have a conflict at α, i.e., in a consistent labeling at most one of them can be active at α.

Lemma 1. *For any two labels ℓ and ℓ' with anchor points $p \in \ell$ and $p' \in \ell'$ the set $C(\ell, \ell')$ consists of at most four disjoint contiguous conflict ranges.*

Proof. The first observation is that due to the simultaneous rotation of all initially axis-parallel labels in L, ℓ and ℓ' remain "parallel" at any rotation angle α. Rotation is a continuous movement and hence any maximal contiguous conflict range in $C(\ell, \ell')$ must be a closed "interval" $[\alpha, \beta]$, where $0 \leq \alpha, \beta \leq 2\pi$. At a rotation of α (resp. β) the two labels ℓ and ℓ' intersect only on their boundary. Let l, r, t, b be the left, right, top, and bottom sides of ℓ and let l', r', t', b' be the left, right, top, and bottom sides of ℓ' (defined at a rotation of 0). Since ℓ and ℓ' are parallel, the only possible cases in which they intersect on their boundary but not in their interior are $t \cap b'$, $b \cap t'$, $l \cap r'$, and $r \cap l'$. Each of those four cases may appear twice, once for each pair of opposite corners contained in the intersection. Figure 2 shows all eight boundary intersection events. Each of the conflicts defines a unique rotation angle and obviously at most four disjoint conflict ranges can be defined with these eight rotation angles as their endpoints. □

In the following we look more closely at the conditions under which the boundary intersection events (the *conflict events*) occur and at the rotation angles defining them. Let h_t and h_b be the distances from p to t and b, respectively. Similarly, let w_l and w_r be the distances from p to l and r, respectively; see Figure 3. By h'_t, h'_b, w'_l, and w'_r we

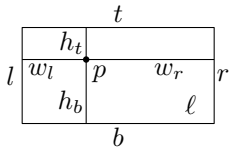


Figure 3: Parameters of label ℓ anchored at p.

(a) rotation of $2\pi - \alpha$ (b) rotation of $\pi + \alpha$

Figure 4: Boundary intersection events for $t \cap b'$.

denote the corresponding values for label ℓ'. Finally, let d be the distance of the two anchor points p and p'. To improve readability of the following lemmas we define two functions $f_d(x) = \arcsin(x/d)$ and $g_d(x) = \arccos(x/d)$.

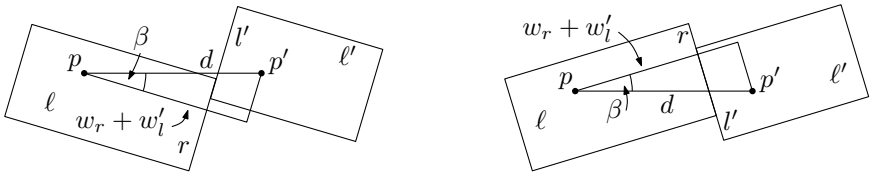
Lemma 2. *Let ℓ and ℓ' be two labels anchored at points p and p' that lie on a horizontal line. Then the conflict events in $C(\ell, \ell')$ are a subset of $\mathcal{C} = \{2\pi - f_d(h_t + h'_b), \pi + f_d(h_t + h'_b), f_d(h_b + h'_t), \pi - f_d(h_b + h'_t), 2\pi - g_d(w_r + w'_l), g_d(w_r + w'_l), \pi - g_d(w_l + w'_r), \pi + g_d(w_l + w'_r)\}$.*

Proof. First we show that the possible conflict events are precisely the rotation angles in $\mathcal{C}$. We start considering the intersection of the two sides t and b'. If there is a rotation angle under which t and b' intersect then we have the situation depicted in Figure 4 and by simple trigonometric reasoning the two rotation angles at which the conflict events occur are $2\pi - \arcsin((h_t + h'_b)/d)$ and $\pi + \arcsin((h_t + h'_b)/d)$. Obviously, we need $d \geq h_t + h'_b$. Furthermore, for the intersection in Figure 4a to be non-empty, we need $d^2 \leq (w_r + w'_l)^2 + (h_t + h'_b)^2$; similarly, for the intersection in Figure 4b, we need $d^2 \leq (w_l + w'_r)^2 + (h_t + h'_b)^2$.

From an analogous argument we obtain that the rotation angles under which b and t' intersect are $\arcsin((h_b + h'_t)/d)$ and $\pi - \arcsin((h_b + h'_t)/d)$. Clearly, we need $d \geq h_b + h'_t$. Furthermore, we need $d^2 \leq (w_r + w'_l)^2 + (h_b + h'_t)^2$ for the first intersection and $d^2 \leq (w_l + w'_r)^2 + (h_b + h'_t)^2$ for the second intersection to be non-empty under the above rotations.

The next case is the intersection of the two sides r and l', depicted in Figure 5. Here the two rotation angles at which the conflict events occur are $2\pi - \arccos((w_r + w'_l)/d)$ and $\arccos((w_r + w'_l)/d)$. For the first conflict event we need $d^2 \leq (w_r + w'_l)^2 + (h_t + h'_b)^2$, and for the second we need $d^2 \leq (w_r + w'_l)^2 + (h_b + h'_t)^2$. For each of the intersections to be non-empty we additionally require that $d \geq w_r + w'_l$.

Similar reasoning for the final conflict events of $l \cap r'$ yields the rotation angles $\pi - \arccos((w_l + w'_r)/d)$ and $\pi + \arccos((w_l + w'_r)/d)$. The additional constraints are $d \geq w_l + w'_r$ for both events and $d^2 \leq (w_l + w'_r)^2 + (h_b + h'_t)^2$ for the first intersection and



(a) rotation of $2\pi - \beta$ (b) rotation of β

Figure 5: Boundary intersection events for $r \cap l'$.

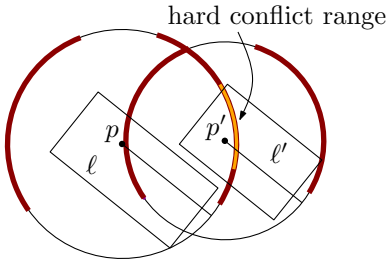


Figure 6: Conflict ranges of two labels ℓ and ℓ' marked in bold on the enclosing circles.

$d^2 \leq (w_l + w'_r)^2 + (h_t + h'_b)^2$ for the second intersection. Thus, $\mathcal{C}$ contains all possible conflict events. □

One of the requirements for a valid labeling in the hard-conflict model is that no label may contain a point in P other than its anchor point. For each label ℓ this gives rise to the special class of *hard* conflict ranges, in which ℓ may never be active. We note that by definition soft conflicts are symmetric, i.e., $C(\ell, \ell') = C(\ell', \ell)$, whereas hard conflicts are not symmetric. The next lemma characterizes the hard conflict ranges.

Lemma 3. *For a label ℓ anchored at point p and a point $q \neq p$ in P such that p and q lie on a horizontal line, the hard conflict events of ℓ and q are a subset of $\mathcal{H} = \{2\pi - f_d(h_t), \pi + f_d(h_t), f_d(h_b), \pi - f_d(h_b), 2\pi - g_d(w_r), g_d(w_r), \pi - g_d(w_l), \pi + g_d(w_l)\}$.*

Proof. We define a label of width and height 0 for q, i.e., we set $h'_t = h'_b = w'_l = w'_r = 0$. Then the result follows immediately from Lemma 2. □

Obviously, Lemma 2 and Lemma 3 can also be used to compute the conflict events when the respective points do not lie on a horizontal line. In this case, we simply rotate the instance by an angle α such that the (anchor) points are horizontally aligned, then we compute the set of conflict events according to the lemma, and finally we shift the computed angles of conflict events by $-\alpha$ to offset the initial rotation.

A simple way to visualize conflict ranges and hard conflict ranges is to mark, for each label ℓ anchored at p and each of its (hard) conflict ranges, its corresponding circular arc on the circle centered at p and enclosing ℓ. Figure 6 shows an example.

In the following we show that the MaxTotal problem can be discretized in the sense that there exists an optimal solution whose active ranges are defined as intervals

whose borders are label events. An active range *border* of a label ℓ is an angle α that is characterized by the property that the labeling ϕ is not constant in any ε-neighborhood of α. Recall that a regular active range is an active range whose borders are both label events.

Lemma 4 (Discretization Lemma)**.** *Given a labeled map M there is an optimal rotation labeling of M consisting of only regular active ranges.*

Proof. Let ϕ be an optimal labeling with a minimum number of active range borders that are not label events. Assume that there is at least one active range border β that is not a label event. Let α and γ be the two adjacent active range borders of β, i.e., $\alpha < \beta < \gamma$, where α and γ are active range borders, but not necessarily label events. Then let L_l be the set of labels whose active ranges have left border β and let L_r be the set of labels whose active ranges have right border β. For ϕ to be optimal L_l and L_r must have the same cardinality since otherwise we could increase the active ranges of the larger set and decrease the active ranges of the smaller set by an $\varepsilon > 0$ and obtain a better labeling.

So define a new labeling ϕ' that is equal to ϕ except for the labels in L_l and L_r: define the left border of the active ranges of all labels in L_l and the right border of the active ranges of all labels in L_r as γ instead of β. Since $|L_l| = |L_r|$ we shrink and grow an equal number of active ranges by the same amount. Thus the two labelings ϕ and ϕ' have the same objective value $\sum_{\ell \in L} \sum_{I \in A_\phi(\ell)} |I| = \sum_{\ell \in L} \sum_{I \in A_{\phi'}(\ell)} |I|$. Because ϕ' uses as active range borders one non-label event less than ϕ this number was not minimum in ϕ—a contradiction. As a consequence ϕ has only label events as active range borders. □

4 Complexity Considerations

In this section we show that finding an optimal solution for MaxTotal (and also MaxMin) is NP-hard in the 1R hard-conflict model even if all labels are unit squares and their anchor points are their lower-left corners. We present a gadget proof reducing from the NP-complete problem Planar 3Sat [16], which is the restriction of 3Sat for planar formulas defined as follows. Let φ be a Boolean 3Sat formula and let G_φ denote the variable–clause graph, which contains a vertex v_x for each variable x of φ and a vertex v_C for each clause C of φ such that v_x and v_C are adjacent if and only if C contains x or $\neg x$. The formula φ is *planar* if and only if G_φ is planar.

Every planar variable–clause graph G_φ admits a *planar grid layout* where all vertices are positioned on the x-axis and the clauses are drawn as three-legged combs above and below the x-axis [14]; see Fig. 7. The layout fits on a polynomially bounded integer grid.

The general idea behind the reduction from an instance φ of Planar 3Sat is to construct an instance I_φ of MaxTotal whose geometric layout mimics the planar grid layout of G_φ and whose optimal rotation labelings are related to the satisfiability of φ in the following sense. There is a value K depending only on φ such that φ is satisfiable if and only if I_φ admits a rotation labeling with total activity at least K.

For this reduction, we describe how to build *gadgets* that correspond to the variables and clauses of the Planar 3Sat formula, called *variable* and *clause gadgets*, respectively,

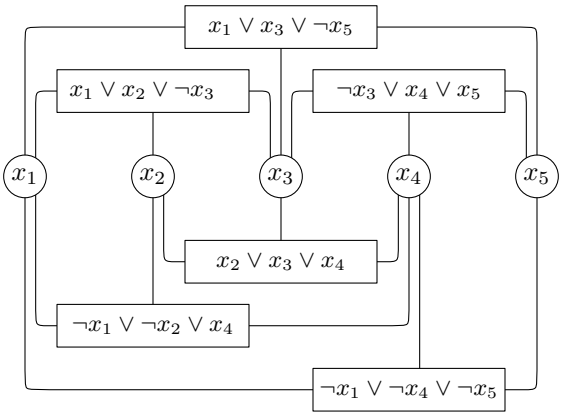


Figure 7: An instance of Planar 3Sat with a grid layout of the variable–clause graph.

as well as *pipes* that model the edges of G_φ. The fact that our layout mimics the planar grid layout of G_φ ensures that the only pairs of labels that interact via conflicts are either part of the same gadget or they are in adjacent gadgets, i.e., one is in a variable gadget and one is in a pipe or one is in a pipe and the other one is in a clause gadget.

Since the gadgets need to properly correspond to Boolean variables, the gadgets can have, in an optimal MaxTotal solution, only two possible states; one which corresponds to `true`, and the other to `false`. Similarly, the clause gadgets should contribute the same value to the total activity of the optimal solution if and only if at least one of the clause's literals is in the state `true`; otherwise, if all three literals are `false`, a clause gadget contributes a smaller value. Obviously, we need also some kind of connection between the variable gadgets and the clause gadgets to transmit the states of the variable gadgets. Those should contribute to the value of the optimal solution of MaxTotal the same value regardless of the state of the connected variable. If this is the case, then we only need to choose the appropriate value K, for which the value of the optimal solution of the instance indicates that the corresponding Planar 3Sat formula has a satisfying truth assignment. Since the construction of the MaxTotal instance can be carried out in polynomial time, it follows that MaxTotal is NP-hard. The same gadgets can also be used to prove NP-hardness of MaxMin as we will see in the end of this section.

4.1 Basic Building Blocks

We start by constructing basic building blocks, called *chain*, *inverter*, and *turn*, from which we construct our gadgets. As a preliminary step we need a special property of unit-square labels.

Lemma 5. *If two unit-square labels ℓ and ℓ' whose anchor points are their lower-left corners have a conflict at a rotation angle α, then they have conflicts at all angles $\alpha + i \cdot \pi/2$ for $i \in \mathbb{Z}$.*

Proof. Similar to the notation used in Section 3, let $f_d = \arcsin(1/d)$ and $g_d = \arccos(1/d)$. From Lemma 2 we obtain the set $\mathcal{C} = \{2\pi - f_d, \pi + f_d, f_d, \pi - f_d, 2\pi - g_d, g_d, \pi - g_d, \pi + g_d\}$

of conflict events for which it is necessary that the distance d between the two anchor points is $1 \leq d \leq \sqrt{2}$. Since $\arccos x = \pi/2 - \arcsin x$ the set $\mathcal{C}$ can be rewritten as $\mathcal{C} = \{f_d, \pi/2 - f_d, \pi/2 + f_d, \pi - f_d, \pi + f_d, 3\pi/2 - f_d, 3\pi/2 + f_d, 2\pi - f_d\}$. This shows that conflicts repeat after every rotation of $\pi/2$. □

For every label ℓ we define the *outer circle* of ℓ as the circle of radius $\sqrt{2}$ centered at the anchor point of ℓ. Since the top-right corner of ℓ traces the outer circle we will use the locus of that corner to visualize active ranges or conflict ranges on the outer circle. Note that due to the fact that at the initial rotation of 0 the diagonal from the anchor point to the top-right corner of ℓ forms an angle of $\pi/4$ all marked ranges are actually offset by $\pi/4$. We now describe the building blocks and prove their basic properties.

Chain. A *chain* consists of at least four labels anchored at collinear points that are evenly spaced with distance $\sqrt{2}$. Hence, each point is placed on the outer circles of its neighbors. We call the first two and last two labels of a chain *terminals* and the remaining part *inner chain*, see Figure 8. A chain is *horizontal* (*vertical*) if its anchor points are horizontally (vertically) aligned. We denote an assignment of active ranges to the labels as the *state* of the chain. The important observation is that in any optimal solution of MaxTotal (and MaxMin) an inner chain has only two different states, whereas terminals have multiple optimal states that are all equivalent for the purpose of MaxTotal; see Figure 8. In particular, in an optimal solution each label of an inner chain has an active range of length π and active ranges alternate between adjacent labels. We will use the two states of chains as a way to encode truth values in our reduction. In an optimal solution of MaxTotal, each terminal contributes $5\pi/2$ to the objective function. One way to achieve this is by assigning active ranges of length π to the inner terminal labels and active ranges of $3\pi/2$ to the outer terminal labels, see the left terminal in Figure 8. Another possibility is to assign an active range of $\pi/2$ to the inner terminal label (starting at the angle $0, \pi/2, \pi$, or $3\pi/2$) and 2π to its neighboring outer terminal label, see the right terminal in Figure 8. For MaxTotal all states behave equivalently, for MaxMin we observe that the first terminal configuration with minimum active range π is always possible.

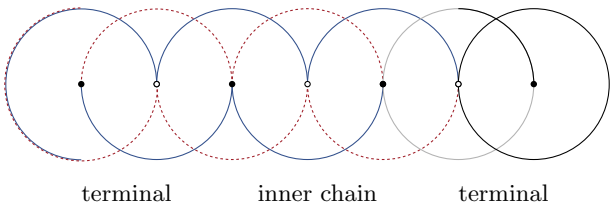


Figure 8: Chain. It has only two states in any optimal solution of MaxTotal. One of the states is marked by the solid blue arcs, while the other is marked by the dotted red arcs. The solid black arcs of the right terminal show an alternative configuration that is independent of the state of the inner chain.

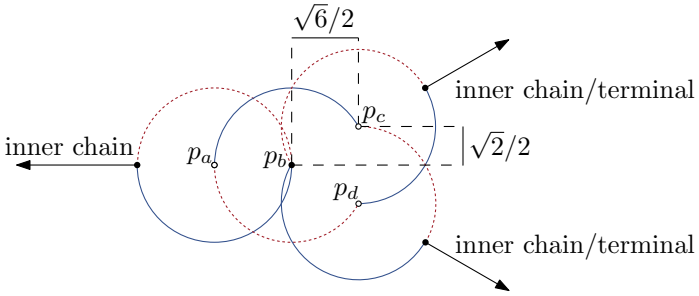


Figure 9: Turn. A turn that splits one inner chain into two inner chains. Like the chain, the turn has only two possible states in an optimal solution. One is marked by the solid blue arc, while the other is marked by the red dotted arc. Connecting gadgets are indicated by labeled arrows.

Lemma 6. *In any optimal solution, any label of an inner chain has an active range of length π. The active ranges of consecutive labels of an inner chain alternate between $(0, \pi)$ and $(\pi, 2\pi)$.*

Proof. By construction every label of an inner chain has two hard conflicts at angles 0 and π, so no active range can have length larger than π. From Lemma 5 we know that every label of an inner chain further has conflicts at $\pi/2$ and $3\pi/2$. These conflicts are soft conflicts and can be resolved by either assigning all odd labels the active range $(0, \pi)$ and all even labels the active range $(\pi, 2\pi)$ or vice versa. Obviously both assignments are optimal and there is no optimal assignment in which two adjacent labels have active ranges on the same side of π. □

For inner chains whose distance between two adjacent points is less than $\sqrt{2}$ the length of the conflict region changes, but the above arguments remain valid for any distance between 1 and $\sqrt{2}$. To ensure that each label has a maximal active range of at least $2\pi/3$, we need at least distance $1/\cos(\pi/12) = \sqrt{6} - \sqrt{2} \approx 1.035$. This implies that chains are *compressible* in the sense that we can move some of their anchors closer together without violating the overall behavior. We will use this freedom later to ensure that some anchors of our constructions end up at specific positions.

Turn. The third building block is a *turn* that consists of four labels; see Figure 9. The anchor points p_a and p_b are at distance $\sqrt{2}$ and the pairwise distances between p_b, p_c, and p_d are also $\sqrt{2}$ such that the whole structure is symmetric with respect to the line through p_a and p_b. The central point p_b is called *turn point*, and the two points p_c and p_d are called *outgoing points*. Due to the hard conflicts created by the four points we observe that the outer circle of p_b is divided into two ranges of length $5\pi/6$ and one range of length $\pi/3$. The outer circles of the outgoing points are divided into ranges of length π, $2\pi/3$, and $\pi/3$. The outer circle of p_a is divided into two ranges of length π. The outgoing points serve as connectors to terminals, inner chains, or further turns. By coupling multiple turns we can divert an inner chain by any multiple of 30°.

Lemma 7. *A turn has only two optimal states and allows an inner chain to be split into two equivalent parts in an optimal solution.*

Proof. We show that the activity in an optimal solution for the turn is $21\pi/6$ and that there are only two different active range assignments that yield this solution. Note that for the label ℓ_a the length of its active range is at most π. For ℓ_b it is at most $5\pi/6$ and for ℓ_c and ℓ_d it is at most π.

We first observe that ℓ_c and ℓ_d cannot both have an active range of length π since by Lemma 5 they have a soft conflict in the intersection of their length-π ranges. Thus at most one of them has an active range of length π and the other has an active range of length at most $5\pi/6$. But in that case the same argumentation shows that the active range of ℓ_b is at most $\pi/2$. Combined with an active range of length π for ℓ_a this yields in total a sum of $20\pi/6$.

On the other hand, if one of ℓ_c and ℓ_d is assigned an active range of length $2\pi/3$ and the other an active range of length π as indicated in Figure 9, the soft conflict of ℓ_b in one of its ranges of length $5\pi/6$ is resolved and ℓ_b can be assigned an active range of maximum length. This also holds for ℓ_a resulting in a total sum of $21\pi/6$.

Since the gadget is symmetric there are only two states that produce an optimal solution for the lengths of the active ranges. By attaching inner chains to the two outgoing points the truth state of the inner chain to the left is transferred into both chains on the right. □

4.2 Gadgets of the Reduction

In the following, we construct the variable, clause, and pipe gadgets. As mentioned before our reduction mimics the planar grid layout of G_φ (recall Figure 7). However, as we have seen (recall Figures 8 and 9), the distances between anchors of our constructions involve distances such as $\sqrt{2}$ and $\sqrt{6}$. Nonetheless, we construct our gadgets in such a way that the coordinates of all anchor points can be expressed in the number field $\mathbb{Q}(\sqrt{2}, \sqrt{3})$. This ensures two crucial properties for the NP-hardness reduction. Firstly, this allows to encode the anchor coordinates using a polynomial number of bits, and thus ensures that the reduction can be carried out in polynomial time. Secondly, all our constructions are exact in the sense that the coordinates of anchors shared by two gadgets can be expressed relative to an arbitrary reference point in each gadget.

Variable Gadget. The variable gadget consists of an alternating sequence of two building blocks: horizontal chains and *literal readers*. A literal reader is a structure that allows us to split the truth value of a variable into one part running via a pipe towards a clause and the part that continues the variable gadget; see Figure 10. The literal reader consists of four turns and three terminals extended with short inner chains on both sides. The first turn connects to a pipe and the other three are dummy turns needed to lead the variable gadget back to the horizontal line. By vertically mirroring the literal reader we can also connect to pipes below the variable gadget.

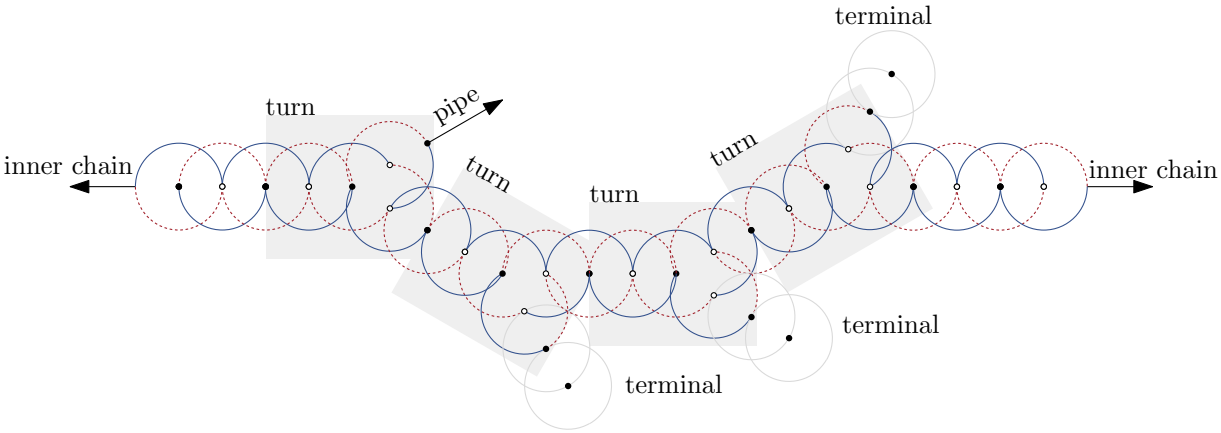
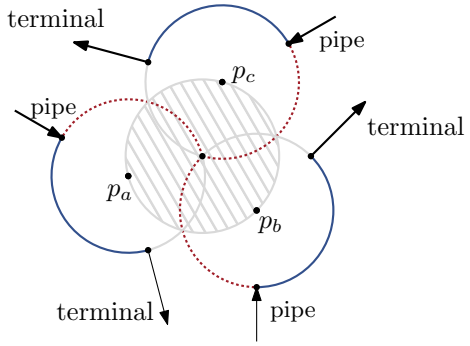


Figure 10: Sketch of a literal reader where connecting gadgets are indicated by labeled arrows. Anchor points at odd (even) positions are marked by black (white) disks.

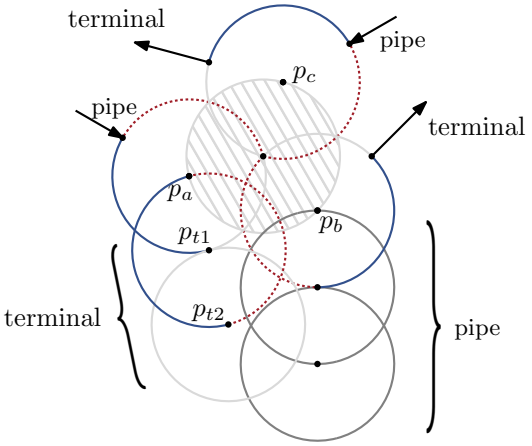
In order to encode truth values we define the state in which the leftmost label (and any other label at an odd position) of the leftmost horizontal chain has active range $(0, \pi)$ as `true` and the state with active range $(\pi, 2\pi)$ as `false`. All horizontal chains and literal readers of the variable gadget have an even number of labels so that the truth values propagate consistently through the variable gadget with respect to the parity of the label positions.

Clause Gadget. The clause gadget consists of one *inner* and three *outer labels*, where the anchor points of the outer labels split the outer circle of the inner label into three equal parts of length $2\pi/3$; see Figures 11 and 12. Each outer label further connects to a pipe and a terminal. These two connector labels are placed so that the outer circle of the outer label is split into two ranges of length $3\pi/4$ and one range of length $\pi/2$. The terminals can be placed such that they do not have a conflict with the other pipes or terminals; see Figure 11b.

The general idea behind the clause gadget is as follows. The inner label obviously cannot have an active range larger than $2\pi/3$. Each outer label is placed in such a way that if it carries the value `false` it has a soft conflict with the inner label in one of the three possible active ranges of length $2\pi/3$; see Figure 12a for an illustration of this (for simplicity only the left outer label and the inner label are depicted). In this example the pipe coming from the left side transmits the truth value `false`. This forces the outer label ℓ_a with anchor p_a to have the active range indicated by the red dotted arc. This splits the arc from p_a to p_c of the inner label's outer circle into two arcs where one arc has length $\pi/2$ and the other $\pi/6$. The functionality of the other outer labels is identical. Hence, if all three pipes transmit the value `false` then every possible active range of the inner label of length $2\pi/3$ is affected by a soft conflict. Consequently, its active range can be at most $\pi/2$. Alternatively, an outer label's active range might get split to allow the inner label to have an active range with length $2\pi/3$, but then this outer label's active range in turn can have length at most $\pi/2$ instead of $3\pi/4$. It is also possible that the active range of a label in the pipe (including the turns), or in a variable might get split, but this also would again reduce

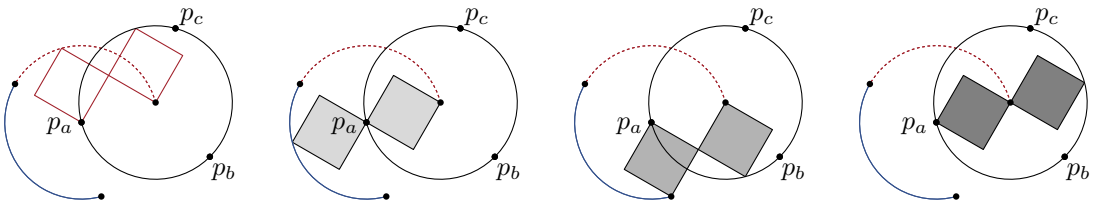


(a) Clause gadget with one inner label (hatched disk) and three outer labels with their anchors p_a, p_b, and p_c placed on the boundary of the inner label such that the intersect at the center of the hatched disk.



(b) Same clause gadget as in a, but with a terminal attached to the left outer label (consisting of two labels with anchors p_{t1} and p_{t2}), and a pipe attached to the lower outer label (anchors at the center of the solid grey circles).

Figure 11: Illustration of the clause gadget.



(a) Soft Conflict. (b) Hard Conflict. (c) Hard Conflict. (d) Hard Conflict.

Figure 12: Illustration of the four conflicts of the left outer label with the inner label. The only soft conflict is depicted in a. All remaining conflicts are hard conflicts.

the active range of that particular label to $\pi/2$. Note that there is an optimal solution of any satisfiable instance, in which any label has an active range of at least $2\pi/3$.

On the other hand, if at least one of the pipes transmits `true`, the inner label can be assigned an active range of length $2\pi/3$. As can be seen in the Figures 12b and 12c, when the outer label ℓ_a with anchor p_a has the active range indicated by the solid blue line (i. e., the attached pipe transmits the truth value `true`), no arc of the inner label is affected. The inner label can have an active range that corresponds to the arc from p_c to p_a of length $2\pi/3$.

Lemma 8. *There must be a label in a clause or in one of the connecting pipes with an active range of length at most $\pi/2$ if and only if all three literals of that clause evaluate to* `false`*.*

Proof. The active range for the lower-right outer label that is equal to the state `false` is $(3\pi/4, 3\pi/2)$. For the two other outer labels the active range corresponding to `false` is rotated by $\pm 2/3\pi$. Note that the outer clause labels can have an active range of at most

$3/4\pi$ and the inner clause label can have an active range of at most $2/3\pi$. For every literal that is `false` one of the possible active ranges of the inner clause label is split by a conflict into two parts of length $\pi/2$ and $\pi/6$. This conflict is either resolved by assigning an active range of length $\pi/2$ to the inner clause label or by propagating the conflict into the pipe or variable where it is eventually resolved by assigning some active range with length at most $\pi/2$.

Otherwise, if at least one pipe transmits `true`, the inner label of the clause can be active for $2\pi/3$ while the outer clause labels have an active range of length $3\pi/4$ and no chain or turn has a label that is visible for less than $2\pi/3$. □

Pipes. Pipes propagate truth values of variable gadgets to clause gadgets. We use three different types of pipes, which we call *left pipe*, *middle pipe*, and *right pipe*, depending on where the pipe attaches to the clause. A pipe is formed by a sequence of turns and chains. Note that since each turn changes the direction by 30°, we can route the pipes at any angle that is an integer multiple of 30°. The construction is illustrated in Fig 13.

One end of each pipe attaches to a variable at the open outgoing label of a literal reader (marked by "pipe" in Figure 10). Initially, the pipe leaves the variable gadget at an angle of 30° with respect to the positive x-axis. We first use two left turns so that the angle is 90°. We then attach a vertical chain to reach the target clause. For a left or right pipe we further extend the construction by a sequence of three right turns (for a left pipe) or a sequence of three left turns (for a right pipe); see Fig 13b. Finally, for left and right pipe, we attach one further turns at the end so that a left pipe enters the clause gadget at an angle of 330°, a middle pipe at an angle of 90°, and a right pipe at an angle of 210° with respect to the positive x-axis. For clauses below the variables the pipes are mirrored.

We position our pipes such that they attach to a literal reader that represents the state of the variable. For transmitting the state of a variable gadget, we use a pipe with an even number of labels along its main path, to transmit its negation, we use a pipe with an odd number of labels on the main path. Due to the alternation property of chains (Lemma 6) this ensure that the correct state is transmitted.

Note that the horizontal position of the clause is determined by the position of the middle pipe. The lengths of the vertical chains of all pipes and the lengths of the horizontal chains of the left and right pipe can be chosen such that we end up sufficiently close to the position of its clause, e.g., by enlarging chains by pairs of anchors so that a pipe gets longer without changing its parity. In this way we can ensure that the last anchor, which attaches to the clause, has distance at most $2\sqrt{2}$ from the target clause in both x- and y-direction. We then compress the horizontal and vertical chains by a small amount to reach the exact target position.

4.3 Reduction

For the final reduction from Planar 3Sat it remains to put together all gadgets according to the grid layout of the given variable–clause graph G_φ. We place all variable gadgets on the same y-coordinate, and all clause gadgets and pipes lie below and above the variables

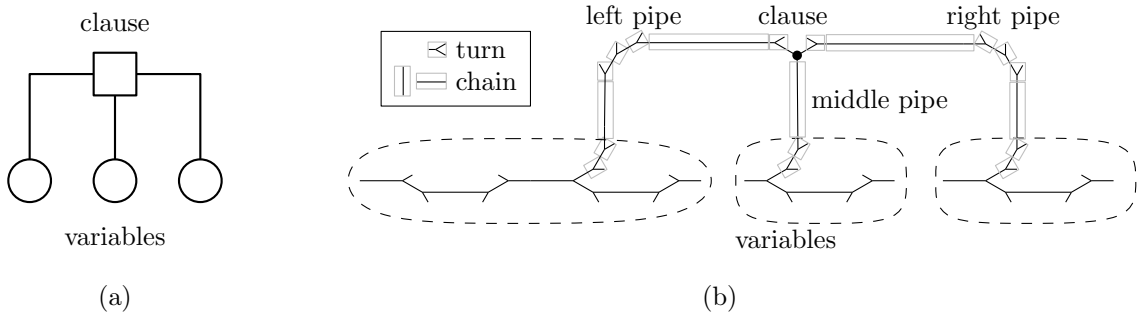


Figure 13: A layout of a part of a variable clause graph (a) and a sketch of the corresponding gadget placement for the reduction (b).

and form three-legged "combs" as in the input layout of G_φ. The overall structure of the gadget arrangement is sketched in Figure 13.

At first glance it seems as if we might have a problem in describing the anchor coordinates with polynomially bounded precision, since, for example, each variable gadget requires us to place anchors with distance $\sqrt{2}$. By encoding the anchor coordinates with algebraic numbers, however, we can avoid this problem. Recall that all our constructed gadgets are either aligned with the coordinate axes directly or they are sloped by 15°, 30°, 45°, or 60° with respect to an axis-parallel line through a reference point. Now it is well known that the sine and cosine values of these angles can all be expressed as algebraic numbers in the number field $\mathbb{Q}(\sqrt{2}, \sqrt{3})$. Hence, if we choose, say, the leftmost anchor of the leftmost variable gadget as the origin, we can express the coordinates of any other anchor with respect to this origin as an algebraic number in $\mathbb{Q}(\sqrt{2}, \sqrt{3})$.

Theorem 1. MaxTotal *is* NP*-hard in the 1R hard-conflict model even if all labels are unit squares and their anchor points are their lower-left corners.*

Proof. For a given planar 3-SAT formula φ we construct the MaxTotal instance as described above. By construction of the hard conflicts in our basic building blocks (chains and turns), we know that in any optimal labeling each label in such a building block has exactly two choices for its active range, both of equal length. Similarly, the inner labels of the clause gadgets have three choices for their active range, all of equal length. Hence we can sum these upper bounds on the lengths of the active ranges for all labels in our entire construction to obtain an upper bound K on the total activity.

It remains to show that φ is satisfiable if and only if this upper bound K can be achieved as the total activity of our construction. In fact, in any basic building block the respective upper bound on the total activity is achieved if and only if all active ranges are assigned correctly, i.e., corresponding to a truth assignment. Hence by Lemma 8 every unsatisfied clause forces its inner label to have an active range of only $\pi/2$ instead of the upper bound $2\pi/3$. Thus we know that φ is satisfiable if and only if the MaxTotal instance has a total activity of at least K (in fact, the total activity must be exactly K since this is also an upper bound). Constructing and placing the gadgets can be done in polynomial time and space, which concludes the proof. □

We note that the same construction as for the NP-hardness of MaxTotal can also be applied to prove NP-hardness of MaxMin. The maximally achievable minimum length of an active range for a satisfiable formula is $2\pi/3$, whereas for an unsatisfiable formula the maximally achievable minimum length is $\pi/2$ due to Lemma 8. This observation also yields that MaxMin cannot be efficiently approximated within a factor of $3/4$, unless P = NP. We summarize this in the following corollary.

Corollary 1. MaxMin *is* NP*-hard in the 1R hard-conflict model even if all labels are unit squares and their anchor points are their lower-left corners. Moreover, there exists no efficient approximation algorithm with an approximation factor larger than* $3/4$ *for this problem, unless* P = NP.

In the conference version of this paper [7] we erroneously claimed that MaxTotal is contained in NP and thus the (corresponding decision) problem is NP-complete. Although it seems that due to the discretization lemma we can simply guess and verify a solution in polynomial time, we encounter the problem that in order to find the value of the guessed solution we potentially need to sum up irrational numbers. This, unfortunately, cannot be done in polynomial time on a Turing machine, and hence, we fail to show membership of MaxTotal in NP.

5 Approximation Algorithms for MaxTotal

In the previous section we have established that MaxTotal is NP-hard. Unless P = NP we cannot hope for an efficient exact algorithm to solve the problem. In the following we devise a 1/4-approximation algorithm for MaxTotal and then refine it to an EPTAS. For both algorithms we initially assume that labels are congruent unit-height rectangles with constant width $w \geq 1$ and that the anchor points are the lower-left corners of the labels. Afterwards we argue how to drop this restriction. Let d be the length of the label's diagonal, i.e., $d = \sqrt{w^2 + 1}$.

Before we describe the algorithms we state one important requirement for our algorithms and three derived properties that apply even to the more general labeling model, where anchor points are arbitrary points within the label or on its boundary, and where the ratio of the smallest and largest width and height, as well as the aspect ratio are bounded by constants:

(i) the number of anchor points contained in any rectangle is at most proportional to its area,
(ii) each label has conflicts with $O(1)$ other labels,
(iii) each label has $O(1)$ conflict events with other labels, and finally,
(iv) there is an optimal MaxTotal solution where all active ranges are bounded by label events.

Property (i) is a fundamental requirement for our algorithms and demands that the point set P is not too dense. In fact, property (i) is usually true in practice, e.g., if we

assume that P is a set of features that admits a conflict-free labeling in some static view of the map M as we show in Lemma 9. Lemma 10 proves property (ii) using a simple packing argument. Property (iii) follows from Property (ii) and Lemma 1. Property (iv) follows immediately from Lemma 4.

Lemma 9. *If M is an input map in which the label set L consists of pairwise disjoint labels (corresponding to a valid labeling of the entire point set P at rotation angle 0) then the number of anchor points in the interior or on the boundary of any rectangle R with width W and height H is at most proportional to the area of R.*

Proof. We consider the non-rotated map M, in which, by assumption, all labels are pairwise disjoint. Let the smallest label height be $h_{\min}$, the smallest label width be $w_{\min}$ and the smallest label area be $a_{\min}$. There can be at most $\lceil 2W/w_{\min} \rceil + \lceil 2H/h_{\min} \rceil$ independent labels intersecting the boundary of R such that their anchor points are contained in R. All remaining labels with an anchor point in R must be completely contained in R, i.e., there can be at most $\lceil W \cdot H/a_{\min} \rceil$ such labels. Hence, the number of anchor points in R is bounded by a constant. □

Lemma 10. *Each label ℓ has conflicts with at most a constant number of other labels.*

Proof. For two labels ℓ and ℓ' to have a conflict their outer circles need to intersect and thus the maximum possible distance between their anchor points is bounded by twice the maximum diameter of all labels in L. By the assumption that the height ratio, width ratio, and aspect ratio of all labels in L is bounded by a constant this diameter is bounded by a constant, too. Hence we can define for each label ℓ a constant-size area around its anchor point containing all relevant anchor points. By Lemma 9 this area contains only a constant number of anchor points. □

5.1 A 1/4-approximation for MaxTotal

The basis for our algorithm is the *line stabbing* or *shifting* technique by Hochbaum and Maass [11], which has been applied before to *static* labeling problems for (non-rotating) unit-height labels [1, 21]. Consider a grid G where each grid cell is a square with side length $2d$. We can address every grid cell by its row and column index. Now we can partition G into four subsets by deleting every other row and every other column with either even or odd parity. Within each of these subsets we have the property that any two grid cells have a distance of at least $2d$. Thus no two labels whose anchor points lie in different cells of the same subset can have a conflict. For an illustration see Figure 14.

For a cell c we call the area that is not contained in c but within distance of at most d to c the *border* of c. To compute an optimal solution for the labels contained in c we must also consider anchor points that lie in this area since during a full rotation a label may intersect such an anchor point. We have marked the border of the cell in row 2 and column 3 in blue in Figure 14.

We say that a grid cell c *covers* a label ℓ if its anchor point lies inside c. By Property (i) only $O(1)$ labels are covered by a single grid cell. Combining this with Properties (ii)

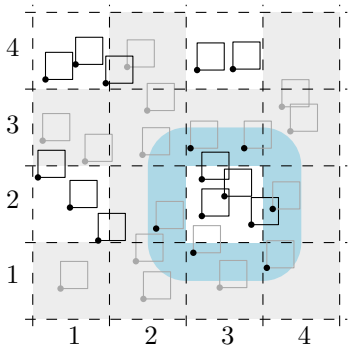


Figure 14: Example for the 1/4-approximation. Depicted are all grid cells and one of the four different subinstances (the even numbered rows and the odd numbered columns are selected). The blue area depicts the area in which the anchor points lie which may be important for the labels in the cell the blue area surrounds.

and (iii) we see that the number of conflict events of the labels covered by the same cell is constant. It remains to determine the number of conflict events that are due to anchors in the border of the cell. It is easy to see that all anchors contained in the border of a cell lie inside a rectangle with dimensions $4d \times 4d$. Again by Property (i) we know that the number of anchor points is bounded by a constant and hence we can conclude that the number of conflict events contributed by these anchor points is also constant. This implies that the total number of conflict events per cell remains constant.

The four different subsets of grid cells divide a MaxTotal instance into four subinstances, each of which decomposes into independent grid cells. If we solve each subset optimally, at least one of the solutions is a 1/4-approximation for the initial instance due to the pigeon-hole principle.

Determining an optimal solution for the labels covered by a grid cell c works as follows. We compute, for the set of labels $L_c \subseteq L$ covered by c, the set E_c of label events (conflict events due to the soft/hard conflicts with labels covered by c and the conflict events due to hard conflicts with the anchors in the border of c). Due to Property (iv) we know that there exists an optimal solution where all borders of active ranges are label events. Thus, to compute an optimal active range assignment for the labels in L_c we need to test all possible combinations of active ranges for all labels $\ell \in L_c$. For a single cell this requires only constant time.

We can precompute the non-empty grid cells by simple arithmetic operations on the coordinates of the anchor points and store those cells in a binary search tree. Since we have n anchor points there are at most n non-empty grid cells in the tree, and each of the cells holds a list of the covered anchor points. Building this data structure takes $O(n \log n)$ time and then optimally solving the active range assignment problem in the non-empty cells takes $O(n)$ time.

Theorem 2. *There exists an $O(n \log n)$-time algorithm that yields a 1/4-approximation of* MaxTotal *for congruent unit-height rectangles with their lower-left corners as anchor points.*

5.2 An Efficient Polynomial-Time Approximation Scheme for MaxTotal

We extend the technique for the 1/4-approximation to achieve a $(1-\varepsilon)$-approximation. Let again G be a grid whose grid cells are squares of side length $2d$. For any integer k we can remove every k-th row and every k-th column of the grid cells, starting at two offsets i and j $(0 \leq i, j \leq k-1)$. This yields collections of meta cells of side length $(k-1) \cdot 2d$ that are pairwise separated by a distance of at least $2d$ and thus independent. In total, we obtain k^2 such collections of meta cells. For an illustration see Figure 15.

For a given $\varepsilon \in (0,1)$ we set $k = \lceil 2/\varepsilon \rceil$. Let c be a meta cell for the given k and let again L_c be the set of labels covered by c, and E_c the set of label events for L_c. Then, by Properties (i) and (ii), the number $|L_c|$ of labels inside c is in $O(1/\varepsilon^2)$ and the number $|E_c|$ of conflict events for the all labels inside c is in $O(1/\varepsilon^4)$. The same upper bound on the number of conflict events holds for anchor points in the border of the meta cell c. By Property (iv) there are for each label $O(1/\varepsilon^8)$ possible combinations of conflict events to define a potential active range. Since we need to test all possible active ranges for all labels in L_c, it takes $O(2^{O(1/\varepsilon^2 \log 1/\varepsilon^8)})$ time to determine an optimal solution for the meta cell c.

For a given collection of disjoint meta cells we determine (as in Section 5.1) all $O(n)$ non-empty meta cells and store them in a binary search tree such that each cell holds a list of its covered anchor points. This requires again $O(n \log n)$ time. So for one collection of meta cells the time complexity for finding an optimal solution is $O(n2^{O(1/\varepsilon^2 \log 1/\varepsilon^8)} + n \log n)$. There are k^2 such collections and, by the pigeon hole principle, the optimal solution for at least one of them is a $(1-\varepsilon)$-approximation of the original instance. This yields the following theorem.

Theorem 3. *There exists an EPTAS that computes a $(1-\varepsilon)$-approximation of* MaxTotal *for congruent unit-height rectangles with their lower-left corners as anchor points. Its time complexity is $O((n2^{O(1/\varepsilon^2 \log 1/\varepsilon)} + n \log n)/\varepsilon^2)$.*

We note that this EPTAS basically relies on properties (i)–(iv) and that there is nothing special about congruent rectangles anchored at their lower-left corners. Hence we can generalize the algorithm to the less restrictive labeling model, in which the ratio of the label heights, the ratio of the label widths, and the aspect ratios of all labels are bounded by constants. Furthermore, the anchor points are not required to be label corners; rather they can be any point on the boundary or in the interior of the labels. Finally, we can even ignore the distinction between hard and soft conflicts, i.e., allow that anchor points of non-active labels are occluded. Properties (i)–(iv) still hold in this general model. The only change in the EPTAS is to set the width and height of the grid cells to twice the maximum diameter of all labels in L.

Corollary 2. *There exists an EPTAS that computes a $(1-\varepsilon)$-approximation of* MaxTotal *in the general labeling model with rectangular labels of bounded height ratio, width ratio, and aspect ratio, where the anchor point of each label is an arbitrary point in that label. The time complexity of the EPTAS is $O((n2^{O(1/\varepsilon^2 \log 1/\varepsilon)} + n \log n)/\varepsilon^2)$.*

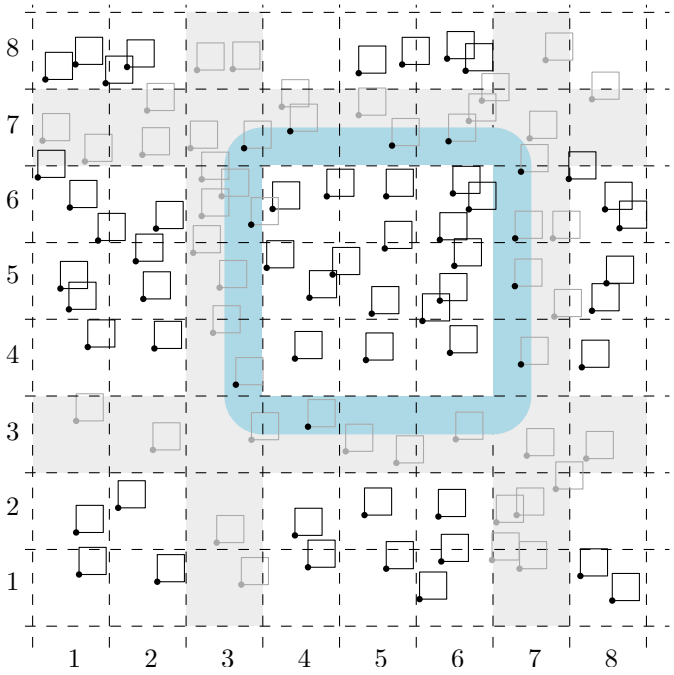


Figure 15: Illustration for our EPTAS for $k = 4$. The blue area is the border of the meta cell spanning columns and rows 4-6.

6 Conclusion

We have introduced a new model for consistent labeling of rotating maps and proved NP-hardness of the active range maximization problem. We could, however, show that there is an EPTAS for the MaxTotal problem that works for rectangular labels with arbitrary anchor points and bounded height ratio, width ratio, and aspect ratio.

We note that our algorithmic techniques can also be applied for the kR-model where k active ranges per label are allowed and, moreover, the special treatment of hard conflicts is not mandatory. Thus, the EPTAS essentially applies to all kR-models with and without hard conflicts. However, our NP-hardness proof heavily relies on the properties of hard conflicts and the presence of only a single active range per label. We cannot make any assertions on the computational complexity of MaxTotal or MaxMin for the kR-model with $k \geq 2$ or even the 1R-model without hard conflicts. We leave this question open for future work. A further open question is to study approximation algorithms for inputs that do not satisfy the property of having a bounded density of point features. Our EPTAS does not extend to this case.

An interesting variation of our problem is to start with a static labeled map (say at rotation angle 0) and optimize the total active ranges such that each active range must contain this angle. While our EPTAS easily extends to this case, the complexity result does not and it is open whether this problem remains NP-hard.

Another interesting open question and an important challenge in practice is to combine map rotation with zooming and panning and to study the arising algorithmic labeling problems in theory and practice. The aspect of temporal consistency must play an important role in this challenge in order to avoid distracting visual effects. Our approach using a single active range per label can in fact be extended to combinations of interaction modes. However, once the space of possible map views becomes multi-dimensional (scale, rotation angle, and possibly more dimensions), a single global pre-computed (multi-dimensional) active range per label is unlikely to be practically relevant as user interactions simply might not pass through these active ranges in most cases. Rather we expect that relatively local consistency concepts for individual interaction sequences will be more useful in practice.

Acknowledgements. A.G. and M.N. were supported by the Concept for the Future of KIT within the framework of the German Excellence Initiative (grant YIG 10-209).

References

[1] Pankaj K. Agarwal, Marc van Kreveld, and Subhash Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11:209–218, 1998.

[2] Ken Been, Eli Daiches, and Chee Yap. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780, 2006.

[3] Ken Been, Martin Nöllenburg, Sheung-Hung Poon, and Alexander Wolff. Optimizing active ranges for consistent dynamic map labeling. *Computational Geometry: Theory and Applications*, 43(3):312–328, 2010.

[4] Parinya Chalermsook and Julia Chuzhoy. Maximum independent set of rectangles. In *Symp. Discrete Algorithms (SODA'09)*, pages 892–901. SIAM, 2009.

[5] Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Symp. Computational Geometry (SoCG'91)*, pages 281–288. ACM, 1991.

[6] Andreas Gemsa, Benjamin Niedermann, and Martin Nöllenburg. Trajectory-Based Dynamic Map Labeling. In *Algorithms and Computation (ISAAC'13)*, volume 8283 of *Lecture Notes Comput. Sci.*, pages 413–423. Springer, 2013.

[7] Andreas Gemsa, Martin Nöllenburg, and Ignaz Rutter. Consistent Labeling of Rotating Maps. In *Algorithms and Data Structures (WADS'11)*, volume 6844 of *Lecture Notes Comput. Sci.*, pages 451–462. Springer, 2011.

[8] Andreas Gemsa, Martin Nöllenburg, and Ignaz Rutter. Sliding labels for dynamic point labeling. In *Canadian Conf. Computational Geometry (CCCG '11)*, pages 205–210. University of Toronto, 2011.

[9] Andreas Gemsa, Martin Nöllenburg, and Ignaz Rutter. Evaluation of labeling strategies for rotating maps. In *Experimental Algorithms (SEA'14)*, volume 8504 of *Lecture Notes Comput. Sci.*, pages 235–346. Springer, 2014.

[10] E. Gervais, D. Nussbaum, and J.-R. Sack. Dynamap: a context aware dynamic map application. In *Proc. GISPlanet, Estoril, Lisbon, Portugal*, 2005.

[11] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[12] Claudia Iturriaga and Anna Lubiw. NP-hardness of some map labeling problems. Technical Report CS-97-18, University of Waterloo, Canada, 1997.

[13] Gunnar W. Klau and Petra Mutzel. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming (Series B)*, pages 435–458, 2003.

[14] Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM J. Discrete Mathematics*, 5(3):422–427, 1992.

[15] Chung-Shou Liao, Chih-Wei Liang, and Sheung-Hung Poon. Approximation algorithms on consistent dynamic map labeling. In *Frontiers in Algorithmics (FAW'14)*, volume 8497 of *Lecture Notes Comput. Sci.*, pages 170–181. Springer, 2014.

[16] David Lichtenstein. Planar formulae and their uses. *SIAM J. Computing*, 11(2):329–343, 1982.

[17] Joe Marks and Stuart Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard CS, 1991.

[18] Kevin D. Mote. Fast point-feature label placement for dynamic visualizations. *Information Visualization*, 6(4):249–260, 2007.

[19] Martin Nöllenburg, Valentin Polishchuk, and Mikko Sysikaski. Dynamic one-sided boundary labeling. In *Advances in Geographic Information Systems (SIGSPATIAL'10)*, pages 310–319. ACM, 2010.

[20] Ingo Petzold, Gerhard Gröger, and Lutz Plümer. Fast screen map labeling—data-structures and algorithms. In *International Cartographic Conf. (ICC'03)*, pages 288–298, Durban, South Africa, 2003.

[21] Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.

[22] Bram Verweij and Karen Aardal. An optimisation algorithm for maximum independent set with applications in map labelling. In *European Symp. Algorithms (ESA'99)*, volume 1643 of *Lecture Notes Comput. Sci.*, pages 426–437. Springer, 1999.

[23] Frank Wagner, Alexander Wolff, Vikas Kapoor, and Tycho Strijk. Three rules suffice for good label placement. *Algorithmica*, 30(2):334–349, 2001.

[24] Yusuke Yokosuka and Keiko Imai. Polynomial time algorithms for label size maximization on rotating maps. In *Canadian Conf. Computational Geometry (CCCG'13)*, pages 187–192, 2013.