Base-Object Location Problems for Base-Monotone Regions

Jinhee Chun[a], Takashi Horiyama[b], Takehiro Ito[a], Natsuda Kaothanthong[a], Hirotaka Ono[c], Yota Otachi[d], Takeshi Tokuyama[a], Ryuhei Uehara[d], Takeaki Uno[e]

[a] *Graduate School of Information Sciences, Tohoku University. Sendai 980-8579, Japan.*
[b] *Graduate School of Science and Engineering, Saitama University. Saitama 338-8570, Japan.*
[c] *Department of Economic Engineering, Kyushu University. 6-19-1 Hakozaki Higashi-ku, Fukuoka 812-8581,Japan.*
[d] *School of Information Science, Japan Advanced Institute of Science and Technology. Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan.*
[e] *National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.*

Abstract

A base-monotone region with a base is a subset of the cells in a pixel grid such that if a cell is contained in the region then so are the ones on a shortest path from the cell to the base. The problem of decomposing a pixel grid into disjoint base-monotone regions was first studied in the context of image segmentation. It is known that for a given pixel grid and base-lines, one can compute in polynomial time a maximum-weight region that can be decomposed into disjoint base-monotone regions with respect to the given base-lines [Chun et al., Comput. Vis. Image Und. (2012)]. We continue this line of research and show the NP-hardness of the problem of optimally locating k base-lines in a given $n \times n$ pixel grid. We then present an $O(n^3)$-time 2-approximation algorithm for this problem. We also study two related problems, the k base-segment problem and the quad-decomposition problem, and present some complexity results for them.

Keywords: Base-monotone region, Room-Edge Problem, Computational complexity, Image segmentation

1. Introduction

Let P be an $n \times n$ pixel grid. A *pixel* (i, j) of P is the unit square whose top-right corner is the grid point $(i, j) \in \mathbb{Z}^2$. For example the bottom-left cell of P

Email addresses: jinhee@dais.is.tohoku.ac.jp (Jinhee Chun), horiyama@al.ics.saitama-u.ac.jp (Takashi Horiyama), takehiro@ecei.tohoku.ac.jp (Takehiro Ito), natsuda@dais.is.tohoku.ac.jp (Natsuda Kaothanthong), hirotaka@en.kyushu-u.ac.jp (Hirotaka Ono), otachi@jaist.ac.jp (Yota Otachi), takeshi@dais.is.tohoku.ac.jp (Takeshi Tokuyama), uehara@jaist.ac.jp (Ryuhei Uehara), uno@nii.ac.jp (Takeaki Uno)

is $(1,1)$ and the top-right cell is (n,n). Each pixel $p=(i,j)$, where $1 \le i,j \le n$, has its *weight* $w(p) \in \mathbb{Z}$. Now we define the following general problem.

Problem: Maximum Weight Region Problem (MWRP)

Instance: An $n \times n$ pixel grid P.

Objective: Find a region $R \in \mathcal{F}$ maximizing the weight $w(R) = \sum_{p \in R} w(p)$, where $\mathcal{F} \subseteq 2^P$ is a family of pixel regions.

The general problem MWRP has been studied for several families $\mathcal{F}$ that are related to practical problems (see [2, 4] and the references therein). Observe that if $\mathcal{F} = 2^P$, then R can be arbitrarily chosen, and thus the answer is the set of all positive cells. On the other hand, if $\mathcal{F}$ is the family of connected regions (in the usual 4-neighbor topology), then the problem becomes NP-hard [2]. For the complexity of MWRP for other families, see the paper by Chun et al. [4] and the references therein.

Motivated by the image segmentation problem, Chun et al. [4] studied a more complicated family of pixel regions for MWRP (see Figure 1). A *base-line* of an $n \times n$ pixel grid P is a vertical line $x = b$ or horizontal line $y = b$, where $0 \le b \le n$. A pixel region R is a *based x-monotone region* if there is a horizontal base-line $y = b$ such that if a cell is contained in R, then so are the ones on a shortest path (in the usual 4-neighborhood graph) from the cell to the base. *Based y-monotone regions* are analogously defined. Based x-monotone regions and based y-monotone regions are *base-monotone regions.* Given a set of k base-lines, a region R is *base-monotone feasible* if it can be decomposed into pairwise disjoint base-monotone regions with respect to the base-lines. The *k base-line MWRP* is MWRP in which we are given k base-lines, and we find a maximum-weight base-monotone feasible region with respect to the base-lines.

Chun et al. [4] observed that the complement of a maximum-weight base-monotone feasible region represents an object in a picture nicely if the base-lines are located reasonably (see Figs. 1 and 3). They showed that the k base-line MWRP can be solved in polynomial time. In [5], they also studied the *k base-segment MWRP*, in which we are given k segments and find a region decomposable into base-monotone regions with respect to the given base-segments. (We define this problem more precisely in the next section.) They showed some partial results on the complexity of this problem. For other approaches for formulating image segmentation as optimization problems, see e.g. [6, 9].

In the setting of the k base-line MWRP, we are given k base-lines. Thus a natural question would be "*What if base-lines are not given?*" In other words, "*How can we divide the pixel grid into subgrids with vertical and horizontal lines?*" We study this problem and show that the problem of optimally locating k base-lines is NP-hard but it can be approximated within factor 2. Next we study the k base-segment MWRP and present sharp contrasts of its computational complexity. Finally, we propose another way for dividing the pixel grid into subgrids, and show that this variant can be solved in polynomial time.

Figure 1: Image segmentation via k base-line MWRP proposed in [4]. We first convert a picture to a gray scale image. Next, with some suitable function, we construct a pixel grid in which each dark pixel has positive weight and each light pixel has negative weight. Finally we solve the k base-line MWRP to segment the background from the objects. In this example, the boundary edges of the picture are used as base-lines (thus $k = 4$). For example, the red region in the third figure has the top edge of the image as its base-line.

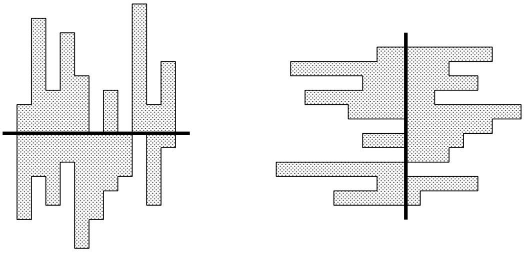

Figure 2: A based x-monotone region (left) and a based y-monotone region (right).

2. Definitions of the Problems and Our Results

In this paper we study three different but well related problems. This section introduces these three problems, briefly explains our results, and then discusses what do the results mean in the context of applications.

2.1. Base-Line Location

To complement the result of Chun et al. [4], who showed that the k base-line MWRP can be solved in $O(n^3)$ time, we study the computational complexity of the following problem.

Problem: Base-Line Location

Instance: An $n \times n$ pixel grid P and positive integers k and w.

Question: Are there k base-lines in P such that a maximum-weight base-monotone feasible region has weight at least w?

There are only $\binom{2n+2}{k}$ possible allocations of k base-lines. Thus Base-Line Location can be solved in $O(2^k n^{k+3})$ time. However, this is impractical if k is a part of the input. We want to solve this problem in $O(\text{poly}(k+n))$ time or in $O(f(k) \cdot \text{poly}(n))$ time, where $f(k)$ is a computable function that depends only on k. Unfortunately, the former, $O(\text{poly}(k+n))$ time, is very unlikely as

Figure 3: The complement of a base-monotone feasible region may represent an object in a picture nicely. By additional base-lines, the result may be improved.

we prove the problem to be NP-hard if k is a part of the input. The latter, $O(f(k) \cdot \mathrm{poly}(n))$ time, remains unsettled in this paper. We also show that this problem allows an $O(n^3)$-time factor-2 approximation.

2.2. The k base-segment MWRP

Consider a segment s contained in a base-line ℓ. If a base-monotone region R with base-line ℓ intersects ℓ only in s, then R has s as its *base-segment*. In [5], Chun et al. studied *k base-segment MWRP*, in which k base-segments are given, and one wants to find a region that can be decomposed into disjoint monotone regions with respect to the given base-segments. They also studied two-directional version of this problem in which the region can be built only on the right side of each vertical base-segment and on the upper side of each horizontal base-segment. They showed the following results.

Theorem 2.1 ([5]). *The k base-segment MWRP can be solved in $O(n^{O(k)})$ time. The two-directional version can be solved in $O(k^{O(k)} n^4)$ time.*

The first statement says that the original problem can be solved in polynomial time if k is not a part of the input. The second statement says that the two-directional version is fixed parameter tractable when parameterized by k. We in this paper complement the first result by showing that the problem is NP-hard when k is a part of the input, and improve the second result by showing that the two-directional version can be solved in polynomial time both in n and k.

2.3. Quad Decomposition

To show that k base-line MWRP can be solved in $O(n^3)$ time, Chun et al. [4] showed that solving the k base-line MWRP is equivalent to solving the following problem for each subgrid obtained by the given base-lines.

Problem: Room-Edge Problem

Instance: An $m \times n$ pixel grid P.

Objective: Find a maximum-weight base-monotone feasible region with the four base-lines $x = 0$, $x = m$, $y = 0$, and $y = n$.

They presented an $O(mn^2)$-time algorithm for the problem above [4]. They solve the Room-Edge Problem for each subgrid, and then return the disjoint union of the solutions as a solution for the k base-line MWRP. From this point

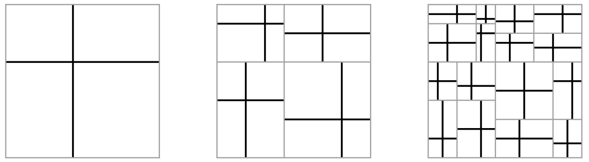

Figure 4: Quad decompositions of depth 1, 2, and 3.

of view, we propose another problem Quad Decomposition. For an $n \times m$ pixel grid P and a point $p = (i, j)$, we can divide P naturally into four subgrids: the bottom-left, bottom-right, upper-left, and upper-right parts with respect to the point p. We call the resultant set of subgrids the *quad decomposition* of P at p. If we recursively apply this decomposition d times (at arbitrarily chosen points), then we will have 4^d subgrids of P (see Figure 4). We call the resultant set of subgrids a *depth d quad decomposition* of P. Now our problem is defined as follows.

Problem: Quad Decomposition

Instance: An $n \times n$ pixel grid P and positive integers d.

Objective: Find a depth d quad decomposition of P that maximizes the total sum of the weight of the optimum solution of Room-Edge Problem for the subgrids in the decomposition.

Note that we can assume $d \leq \log_2 n$ since otherwise the problem becomes trivial (we can take all the positive cells). We will show that this problem can be solved in polynomial time.

2.4. Discussion about the Results

Following the work of Chun et al. [4, 5], we study the image segmentation problem formulated as combinatorial optimization problems. As explained above, the complexity varies with each problem. For Quad Decomposition, an optimal decomposition can be found in polynomial time. For Base-Line Location, an optimal decomposition is hard to find, but given a decomposition, we can find optimal base-monotone regions in polynomial time. For the k base-segment MWRP, even finding optimal base-monotone regions with given base-segments is hard. Also, the hardness proof for Base-Line Location implicitly implies the hardness of the k base-segment location. (We can just use the same reduction.)

In the context of image segmentation, we may expect that quad decompositions work well compared to k base-line decompositions. This is because, by using quad decompositions, we can place many bases in complicated parts of the image. However, the running time of our algorithm for Quad Decomposition is $O(n^7)$, which is quite high. Therefore, in some setting, using approximations or heuristics for Base-Line Location would be much more practical. Indeed, in a sister paper [3], we proposed some heuristic algorithms which run much faster in practice. See [3, 4] for some experimental experiments. Finally, the hardness of the k base-segment MWRP would imply that formulating the image segmentation as this general problem is not a good idea, unless we can do it in the two-directional setting.

3. NP-Hardness of Base-Line Location

Here we prove the following theorem.

Theorem 3.1. Base-Line Location *is NP-complete in the strong sense.*

The problem is clearly in NP. We prove its strong NP-hardness by reducing Independent Set to this problem. An *independent set* of a graph is a set of pairwise non-adjacent vertices. The following problem is known to be NP-complete [7].

Problem: Independent Set

Instance: A graph G and a positive integer s.

Question: Does G have an independent set of size at least s?

Note that Independent Set is NP-complete even if $|V(G)| = |E(G)|$. If $|V(G)| > |E(G)|$, then we increment s by 1 and add a sufficiently large complete component to G to make $|V(G)| \leq |E(G)|$. If $|V(G)| < |E(G)|$, then we increment s by $|E(G)| - |V(G)|$ and add $|E(G)| - |V(G)|$ new isolated vertices to G to make $|V(G)| = |E(G)|$. It is easy to see that the obtained instance is equivalent to the original one.

Proposition 3.2. Independent Set *is NP-complete even if* $|V(G)| = |E(G)|$.

3.1. Gadgets

We first define two small gadgets for forcing base-lines into restricted zones. Throughout this paper, each black $\bullet$ represents a positive weight. Also, each orange $\times$ in a pixel grid represents a huge (but polynomially bounded) negative weight whose absolute value is equal to the sum of all the positive weights in the grid. All the other cells have weight 0. Therefore, we cannot take any orange $\times$ into our solution.

Our first gadget is the 3×3 grid depicted in Figure 5. If we want to take the positive cell at the center into base-monotone regions, we need one base-line as in the figure. Since we cannot take any huge negative cell into solutions, the possible locations of the base-lines are restricted to the four positions in the figure. We call this gadget a *base-line forcer*. The *weight* of a base-line forcer is the weight of the positive cell, and the *position* of a base-line forcer is the position of its bottom-left cell.

Next we consider a similar gadget depicted in Figure 6. In order to take all the positive cells and not to take any negative cell into base-monotone regions, we need either one vertical base-line or two horizontal base-lines. Therefore, if

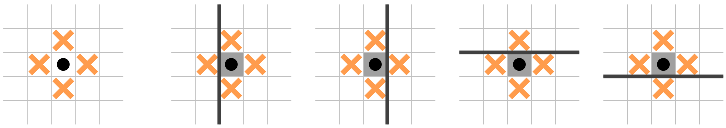

Figure 5: A base-line forcer: forcing one base-line.

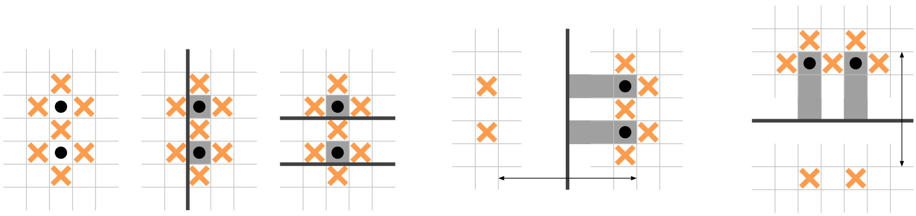

Figure 6: (Left) A vertical base-line forcer: forcing one vertical base-line or two horizontal base-lines. (Right) A generalized vertical base-line forcer and a generalized horizontal base-line forcer: to take the positive cells into a solution, we have to put a base-line in the area depicted in the figure.

we need to minimize the number of base-lines, then we have to use one vertical base-line. We call this gadget a *vertical base-line forcer*. By rotating this gadget, we can also obtain a gadget for forcing two vertical base-lines or one horizontal base-line. We call it a *horizontal base-line forcer*. The two positive cells in this gadget have the same weight, and their weight is the *weight* of the vertical or horizontal base-line forcer. The *position* of a vertical or horizontal base-line forcer is the position of its bottom-left cell.

Vertical and horizontal base-line forcers work even if we insert some space between columns or rows as in Figure 6. The location of the base-line is restricted to the area depicted in the figure. We say that a vertical (horizontal) base-line forcer *intersects* a vertical (horizontal resp.) base-line if the base-line is in the restricted area; that is, a base monotone shape with the vertical or horizontal base-line can contain the positive cells in the vertical or horizontal base-line forcer. The number of the columns used by a vertical base-line forcer is its *width*, and the number of rows used by a horizontal base-line forcer is its *height*. For example, the original vertical base-line forcer in Figure 6 is of width 3.

3.2. Reduction

Given an instance (G, s) of Independent Set, we construct an instance (P, k, w) of Base-Line Location as follows. It is easy to see that the reduction below can be done in polynomial time, and the absolute values of the weights are bounded by a polynomial of the input size.

By Proposition 3.2, we may assume $|V(G)| = |E(G)|$ to simplify the proof. Let $V(G) = \{v_1, \ldots, v_m\}$ and $E(G) = \{e_1, \ldots, e_m\}$. We set the number of base-lines $k = 2m$ and the required weight $w = 8m^3 + 8m^2 + s$. The grid P is the $(20m+20) \times (20m+20)$ pixel grid with the following entries (see Figure 7). Note that we put the following gadgets in such a way that they do not intersect each other, since otherwise they do not work.

Vertex gadgets. For each vertex v_i, we put a vertical base-line forcer of width 5 and weight $2m^2+m$, denoted VF_i, at the position $(10i, 5i)$. We also put a base-line forcer of weight 1, denoted BF_i, at the position $(10i-1, 20m+15)$.

Edge gadgets. Let $e_h = \{v_i, v_j\} \in E(G)$ be an edge with $i < j$. We put a horizontal base-line forcer of height 10 and weight $2m^2+m$, denoted HF_h, at the position $(10m+5h, 5m+15h)$. Next we put two horizontal base-line forcers $HF_{h,i}$ and $HF_{h,j}$ of height 3 and weight m at the positions $(10i-3, 5m+15h-1)$ and $(10j-3, 5m+15h+8)$, respectively. Also, we put two base-line forcers $BF_{h,i}$ and $BF_{h,j}$ of weight m at the positions $(10i+3, 5m+15h+2)$ and $(10j+3, 5m+15h+5)$, respectively.

The weight of negative cells. We have the following positive cells in the grid:

- $4m$ cells of weight $2m^2+m$,
- $6m$ cells of weight m, and
- m cells of weight 1.

The total weight of the positive cells is $W = 4m(2m^2+m) + 6m^2 + m = 8m^3 + 10m^2 + m$. We set the weight of the negative cells to $-W$ so that these cells cannot be taken in any solution with a positive total weight.

3.3. Equivalence

Now we show the equivalence between (G, s) and (P, k, w) constructed above. That is, we show that (G, s) is a yes-instance of Independent Set if and only if (P, k, w) is a yes-instance of Base-Line Location. This shows the NP-hardness of Base-Line Location. Moreover, since the weight of each cell is polynomially bounded, the problem is NP-hard in the strong sense.

Lemma 3.3. *If (G, s) is a yes-instance of* Independent Set*, then (P, k, w) is a yes-instance of* Base-Line Location*.*

Proof. Let S be an independent set of G with $|S| \geq s$. We use m vertical base-lines for vertices and m horizontal base-lines for edges.

For each vertex $v_i \in S$, we set a vertical base-line at $x = 10i$. For each vertex $v_i \in V(G) \setminus S$, we set a vertical base-line at $x = 10i+3$. Let $e_h = \{v_i, v_j\} \in E(G)$ be an edge with $i < j$. If $v_i \in S$, then we set a horizontal base-line at $y = 5m+15h+8$. Otherwise, we set a horizontal base-line at $y = 5m+15h$. For example, see Figure 8 for the case of $e_h = \{v_i, v_j\} \in E(G)$ and $v_i \in S$. Note that these facts imply $v_j \notin S$ from the definition of independent sets.

Each vertical base-line corresponding to a vertex v_i can take two cells of weight $2m^2+m$ in VF_i and $\deg_G(v_i)$ cells of weight m in $\{HF_{h,i} \mid v_i \in e_h\}$ if

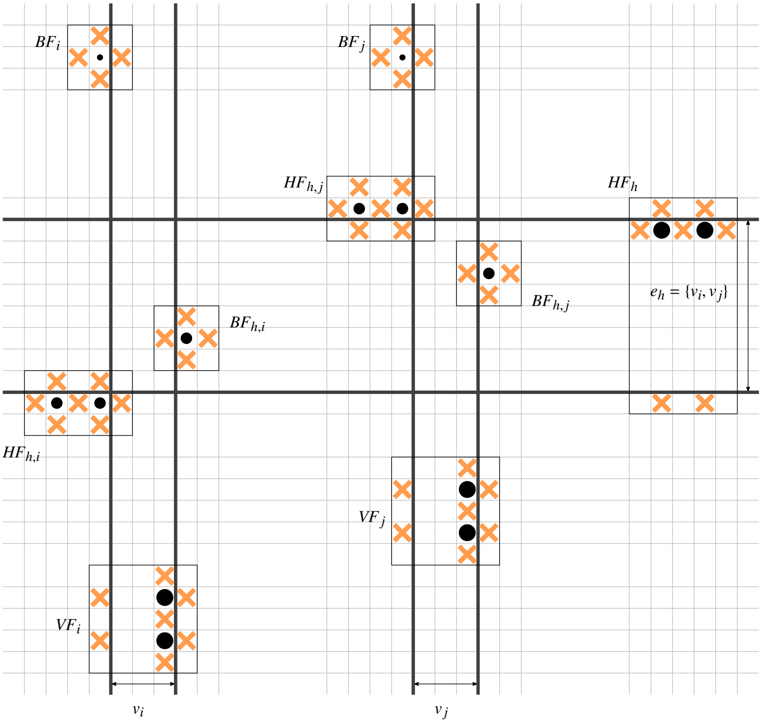

Figure 7: Gadgets for an edge $\{v_i, v_j\}$: black thick lines are the candidates of required base-lines, two vertical and one horizontal.

$v_i \in S$, or $\{BF_{h,i} \mid v_i \in e_h\}$ if $v_i \notin S$. Also, if $v_i \in S$, then the vertical base-line can take one cell of weight 1 in BF_i.

Let $e_h = \{v_i, v_j\}$ and assume $v_j \notin S$ without loss of generality. The horizontal base-line corresponding to e_h can take two cells of weight $2m^2 + m$ in HF_h. If $v_i \in S$, then we have the base-line at $y = 5m + 15h + 8$. Since $v_j \notin S$, the positive cells of weight m in $HF_{h,j}$ are not taken by any vertical base-line, and hence they can be taken by the horizontal base-line. If $v_i \notin S$, then we have the horizontal base-line at $y = 5m + 15h$. The positive cells of weight m in $HF_{h,i}$ are not taken by any vertical base-line, and can be taken by the horizontal base-line.

From the above observation, we can take $4m$ cells of weight $2m^2 + m$, $2m + 2|E| = 4m$ cells of weight m, and $|S|$ cells of weight 1. The total weight of these cells is $4m(2m^2 + m) + 4m^2 + |S| = 8m^3 + 8m^2 + |S| \geq w$. This completes the proof. $\square$

Lemma 3.4. *If* (P, k, w) *is a yes-instance of* Base-Line Location, *then* (G, s) *is a yes-instance of* Independent Set.

To prove this lemma, we need the following propositions.

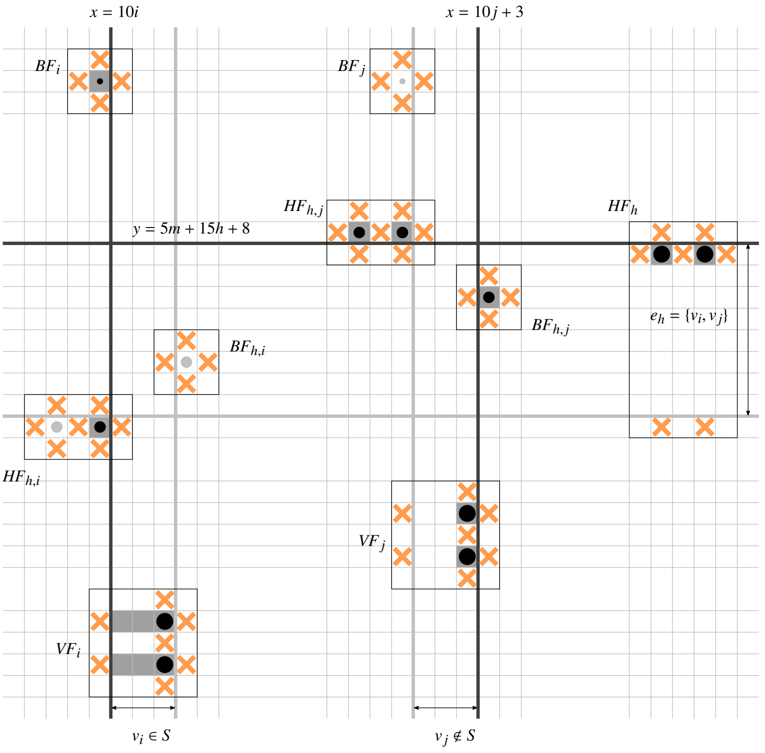

Figure 8: The case of $\{v_i, v_j\} \in E(G)$ and $v_i \in S$. Black thick lines are the selected base-lines.

Proposition 3.5. *To take the total weight at least $w = 8m^3+8m^2+s$, we must take all the $4m$ cells of weight $2m^2+m$ and at least $4m$ cells of weight m.*

Proof. Recall that the sum of the weights of all positive cells is $W = 8m^3 + 10m^2+m$. If we do not take all the cells of weight $2m^2+m$, then the total weight of taken cells will be at most $W - (2m^2 + m) = 8m^3 + 8m^2 < w$. Similarly, if we do not take at least $4m$ cells out of the $6m$ cells of weight m, then the total weight of taken cells will be at most $W - (2m + 1)m = 8m^3 + 8m^2 < w$. $\square$

Corollary 3.6. *If (P, k, w) is a yes-instance, then each VF_i intersects exactly one vertical base-line, and each HF_h intersects exactly one horizontal base-line.*

Proof. Otherwise, we cannot take all $4m$ cells of weight $2m^2 + m$. $\square$

Note that, from the construction, no vertical (horizontal) base-line can intersect two or more vertical (horizontal, resp.) base-line forcers of weight $2m^2+m$. Thus we denote the vertical base-line that intersects VF_i by VL_i, and the horizontal base-line that intersects HF_h by HL_h.

Proposition 3.7. *If (P, k, w) is a yes-instance, then for each $e_h = \{v_i, v_j\} \in E(G)$,*

- *HL_h must take the two positive cells in either $HF_{h,i}$ or $HF_{h,j}$,*
- *VL_i must take one positive cell in $HF_{h,i}$ or $BF_{h,i}$, and*
- *VL_j must take one positive cell in $HF_{h,j}$ or $BF_{h,j}$.*

PROOF. By Corollary 3.6, only HL_h, VL_i, and VL_j can take cells of weight m in $HF_{h,i}$, $HF_{h,j}$, $BF_{h,i}$, and $BF_{h,j}$. It is easy to see that VL_i can take only one positive cell in $HF_{h,i}$ or $BF_{h,i}$ and VL_i can take only one positive cell in $HF_{h,j}$ or $BF_{h,j}$. Also, it is not difficult to see that HL_h can take either two positive cells in $HF_{h,i}$ or two positive cells in $HF_{h,j}$. On the other hand, by Proposition 3.5, we must take at least four cells of weight m in $HF_{h,i}$, $HF_{h,j}$, $BF_{h,i}$, and $BF_{h,j}$. This completes the proof. □

A vertex v_i is *left* if VL_i is a vertical base-line $x = 10i$, and v_i is *right* if VL_i is a vertical base-line $x = 10i + 3$. An edge e_h is *top* if HL_h is a horizontal base-line $y = 5m + 15h + 8$, and e_h is *bottom* if HL_h is a horizontal base-line $y = 5m + 15h$. It is easy to see that if (P, k, w) is a yes-instance, then each vertex is left or right, and each edge is top or bottom, by Proposition 3.7 (see Figure 7 and Figure 8).

The following proposition relates BASE-LINE LOCATION to INDEPENDENT SET.

Proposition 3.8. *If (P, k, w) is a yes-instance, then the set of left vertices is an independent set of G.*

PROOF. It suffices to show that for each $e_h = \{v_i, v_j\} \in E(G)$ with $i < j$, at least one of v_i and v_j must be a right vertex.

Suppose that both v_i and v_j are left. In this case, VL_i can take only one cell of weight m in $HF_{h,i}$, and VL_j can take only one cell of weight m in $HF_{h,j}$. Also HL_h can take only two cell of weight m in either $HF_{h,i}$ or $HF_{h,j}$, but one of them is already taken by VL_i or VL_j. By Proposition 3.7, (P, k, w) is not a yes-instance. □

Now we are ready to prove Lemma 3.4.

PROOF (LEMMA 3.4). Let (P, k, w) be a yes-instance of BASE-LINE LOCATION. By the discussion in this section, each vertex is left or right, and each edge is top or bottom. That is, there is a vertical base-line $x = 10i$ or $x = 10i + 3$ for each vertex v_i, and there is a horizontal base-line $y = 5m + 15h + 8$ or $y = 5m + 15h$ for each edge e_h. These base-lines take all the $4m$ cells of weight $2m^2 + m$ and exactly $4m$ cells of weight m. Additionally for each left vertex v_i, the corresponding vertical base-line $x = 10i$ can take the positive cell of weight 1 in BF_i. No other positive cells can be taken.

Let L be the set of left vertices. Then the total weight of the positive cells taken is

$$4m(2m^2 + m) + 4m^2 + |L| = 8m^3 + 8m^2 + |L|.$$

Since this value is at least $w = 8m^3 + 8m^2 + s$, it follows that $|L| \geq s$. By Proposition 3.8, (G, s) is a yes-instance of INDEPENDENT SET. □

4. A 2-Approximation Algorithm for Base-Line Location

Our approximability result is based on the polynomial-time solvability of the following problem.

Problem: Vertical Base-Line Location

Instance: An $n \times n$ pixel grid P and a positive integer k.

Objective: Find k vertical base-lines in P that maximize the weight of an optimal base-monotone feasible region with respect to these base-lines.

The problem Horizontal Base-Line Location is defined analogously. We show that these problems can be solved in $O(n^3)$ time.

Theorem 4.1. Vertical Base-Line Location *and* Horizontal Base-Line Location *can be solved in $O(n^3)$ time.*

Proof. By symmetry, it suffices to show the result only for Vertical Base-Line Location. Without loss of generality, we assume that we can use the vertical lines $x = 0$ and $x = n$ as base-lines for free by adding new first and last columns to the grid and setting huge negative weights to the new cells.

For $1 \leq r \leq n$ and $0 \leq i \leq j \leq n$, let $P^r_{i,j}$ be the subgrid of P consisting of the pixels (c, r) for $i < c \leq j$. Let $A^r_{i,j}$ be the maximum weight of base-monotone regions in $P^r_{i,j}$ with the base-lines $x = i$ and $x = j$. Similarly, let $B^r_{i,j}$ be the maximum weight of base-monotone regions in $P^r_{i,j}$ with the base-line $x = i$ only. Clearly $A^r_{i,i} = B^r_{i,i} = 0$. For $i < j$, we have

$$A^r_{i,j} = \max\left\{B^r_{i,j-1}, A^r_{i,j-1} + w(j,r)\right\}$$

and

$$B^r_{i,j} = \max\left\{B^r_{i,j-1}, \sum_{j'=i+1}^{j} w(j',r)\right\}.$$

These facts imply that for fixed r and i, we can compute $A^r_{i,j}$ and $B^r_{i,j}$ for $i \leq j \leq n$ in $O(n)$ time. Therefore, we can compute $A^r_{i,j}$ for all r, i, j in $O(n^3)$ time.

For $0 \leq i \leq j \leq n$, let $P_{i,j}$ be the subgrid $[i,j] \times [0,n]$ of P. Let $C_{i,j}$ be the maximum weight of base-monotone regions in $P_{i,j}$ with the base-lines $x = i$ and $x = j$. It is easy to see that $C_{i,j} = \sum_{r=1}^{n} A^r_{i,j}$. Hence we can compute $C_{i,j}$ for all i, j in $O(n^3)$ time by using the entries $A^r_{i,j}$, $1 \leq r \leq n$.

Let $D_{h,j}$ be the optimal value of the h base-line MWRP in $P_{0,j}$ with respect to the vertical base-line $y = j$, and other $h - 1$ base-lines in $P_{0,j}$. It is easy to see that $D_{1,j} = C_{0,j}$ and that for $h \geq 2$,

$$D_{h,j} = \max_{h-1<i<j} \left(D_{h-1,i} + C_{i,j}\right).$$

Using the entries $D_{h-1,i}$ and $C_{i,j}$ for $h - 1 < i < j$, we can compute $D_{h,j}$ in $O(n)$ time. Thus we can compute $D_{h,j}$ for all h, j in $O(n^3)$ time. Now clearly

$$\max_{k<i<n} \left(D_{k,i} + C_{i,n}\right)$$

is the optimal value. Furthermore, by slightly modifying the algorithm, we can output the positions of k vertical base-lines in the same running time. $\square$

We solve Vertical Base-Line Location with k vertical base-lines and Horizontal Base-Line Location with k horizontal base-lines in $O(n^3)$ time, independently. We output the better one of these solutions. We can show that the output is a 2-approximation solution.

Theorem 4.2. *There is an $O(n^3)$-time 2-approximation algorithm for locating k base-lines to maximize the weight of optimum base-monotone feasible region.*

Proof. We first solve Vertical Base-Line Location with k vertical base-lines and Horizontal Base-Line Location with k horizontal base-lines in $O(n^3)$ time, independently. We output the better one of these solutions. We now show that one of these two solutions has weight at least half of the best solution of Base-Line Location with k base-lines.

Assume that an optimal solution of Base-Line Location is attained with k_v vertical and k_h horizontal base-lines, where $k_v + k_h = k$. Let P_v and P_h be the sets of cells taken by vertical and horizontal lines, respectively, in the optimal solution of Base-Line Location. Note that partition into P_v and P_h is not unique. We just select one partition arbitrarily. Let $W_v = \sum_{p \in P_v} w(p)$ and $W_h = \sum_{p \in P_h} w(p)$. Now $W_v + W_h$ is the maximum weight for Base-Line Location. Assume without loss of generality that $W_v \geq W/2$.

Observe that P_v can be taken by k vertical base-lines. This is because, additions of base-lines never violate the feasibility of base-monotone regions. Therefore, the optimum value of Vertical Base-Line Location is at least $W_v \geq W/2$. $\square$

Now we show the tightness of the analysis of the approximation ratio. To this end, we use *generalized horizontal* (*vertical*) *base-line forcers* depicted in Figure 9. The length of a generalized (horizontal or vertical) base-line forcer is the number of positive cells in it. To take all the positive cells in a generalized horizontal base-line forcer of length ℓ (and not to take any negative cell), we need either one horizontal base-line or ℓ vertical base-lines. We construct a tight example by putting one generalized horizontal base-line forcer of length ℓ and one generalized vertical base-line forcer of length ℓ so that no row nor column intersects both of the two base-line forcers. We set the weight 1 to every positive cell. Clearly, we can take all positive cells with one horizontal and one vertical base-lines, and thus the optimal solution is of weight 2ℓ. On the other hand, if we use only horizontal base-lines (or only vertical base-lines), then we can take only $\ell+1$ positive cells using two base-lines. Therefore, the approximation ratio is $2\ell/(\ell+1) = 2 - o(1)$.

The tight example above can be beaten by a heuristic idea: if we guess the number of horizontal base-lines (and thus the number of vertical ones), then we can obtain the optimal solution for the example.

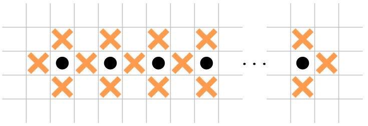

Figure 9: A generalized horizontal base-line forcer. A generalized vertical base-line forcer can be obtained by rotating this gadget.

5. The k Base-Segment MWRP

Here we extend the results of Chun et al. [5] (see Theorem 2.1). We first reduce the two-directional version to Weighted Independent Set in bipartite graphs. We next reduce Independent Set in planar graphs to the original k base-segment MWRP. The first reduction implies that the two-directional version can be solved in polynomial time, and the second implies that the original problem is NP-hard, since Independent Set can be solved in polynomial time for bipartite graphs [11], and is NP-hard for planar graphs [8].

In what follows, we may assume without loss of generality that no base-monotone shape with respect to a base-segment contains another base-segment properly (in such a case, we can partition the base-monotone shape). We may also assume that two vertical (or two horizontal) base-segments may have intersection only at their end-points.

5.1. Two-directional version

To show that 2-directional version can be solved in polynomial time, we reduce the problem to Weighted Independent Set for bipartite graphs.

We first divide each base-segment of length ℓ into ℓ unit base-segments. This refinement does not change the optimum value. Now we have $O(kn)$ base-segments of length 1. We identify a base-segment s with (i, j) if s is the left or bottom edge of a pixel (i, j).

For each vertical base-segment $s = (i, j)$, we define its *range* as follows: if there is no vertical base-segment $s' = (i', j)$ with $i' > i$, then the range of s is $[i, n]$; otherwise the range of s is $[i, i' - 1]$, where i' is the smallest index for which such a segment exists (see Figure 10). We define the *range* of a horizontal base-segment analogously.

Let $s = (i, j)$ be a vertical base-segment with range $[i, i']$. Let $a_s(0) = i - 1$, and for $p \geq 1$, let $a_s(p)$ be the minimum index h such that $a_s(p - 1) < h \leq i'$ and $\sum_{a_s(p-1)<q\leq h} w(q, j)$ is positive. If there is no such index, then $a_s(p)$ is undefined. If $a_s(p)$ is defined for some $p \geq 1$, then let $w_s(p) = \sum_{a_s(p-1)<q\leq a_s(p)} w(q, j)$. See Figure 10. For each horizontal base-segment s', we also define the sequence $a_{s'}(\cdot)$ and $w_{s'}(\cdot)$ analogously.

Now we construct a vertex-weighted bipartite graph $G = (U, V; E)$. Let $s = (i, j)$ be a vertical base-segment. Assume that ℓ is the largest index such that $a_s(\ell)$ is defined. All $a_s(0), \ldots, a_s(\ell)$ are defined by the definition. If $\ell = 0$, then this segment s is useless and ignored. Otherwise, we put vertices $u_s(p)$, $1 \leq$

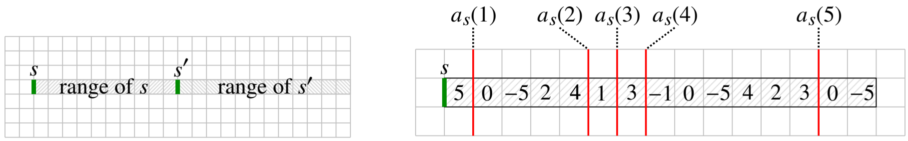

Figure 10: (Left) The ranges of vertical base-segments s and s'. (Right) Example of $a_s(p)$. The corresponding weights $w_s(1), \ldots, w_s(5) = 5, 1, 1, 3, 3$.

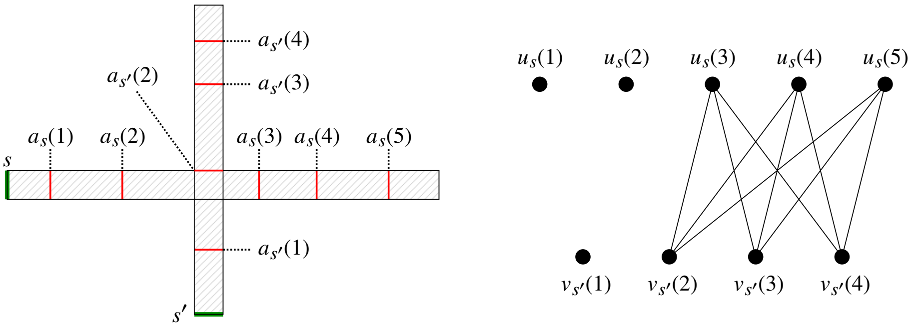

Figure 11: The bipartite graph construction. The vertices corresponding to the crossing ranges of two base-segments induce the disjoint union of an independent set and a complete bipartite graph.

$p \leq \ell$, with weight $w_s(p)$ into U. For each horizontal base-segment $s' = (i', j')$, we put vertices $v_{s'}(p')$ into V in the same way. Next we define the edge set E. Two vertices $u_s(p) \in U$ and $v_{s'}(p') \in V$ are adjacent if and only if two base-monotone regions with base-segments s and s' have nonzero area intersection if they contain $(a_s(p), j)$ and $(i', a_{s'}(p'))$, respectively. More precisely, this can be stated as: $i \leq i' \leq a_s(p)$ and $j' \leq j \leq a_{s'}(p')$. See Figure 11 for an example.

From the definition of $a_s(p)$ and $w_s(p)$, the following fact follows.

Remark 5.1. *In each inclusion-wise minimal optimal solution, the base-monotone region with the vertical base-segment $s = (i, j)$ is either empty or consists of the consecutive cells $(i, j), \ldots, (a_s(p), j)$ for some $p \geq 1$. In the latter case, the weight of the base-monotone region with s is $\sum_{1 \leq q \leq p} w_s(q)$.*

The horizontal counterpart of the above remark also holds.

From the conditions $i \leq i' \leq a_s(p)$ and $j' \leq j \leq a_{s'}(p')$, it is easy to see that if $a_s(p+1)$ is defined and $\{u_s(p), v_{s'}(p')\} \in E$, then $\{u_s(p+1), v_{s'}(p')\} \in E$ also holds. Thus we have $N_G(u_s(1)) \subseteq N_G(u_s(2)) \subseteq \cdots$ for any vertical base-segment s, and $N_G(v_{s'}(1)) \subseteq N_G(v_{s'}(2)) \subseteq \cdots$ for any horizontal base-segment s', where $N_G(v)$ denote the neighborhood of $v \in V(G)$ in G. Thus we have the following lemma.

Lemma 5.2. *Each maximal independent set of G is of the form*

$$\bigcup_{s \in C} \{u_s(1), \ldots, u_s(p_s)\} \cup \bigcup_{s' \in C'} \{v_{s'}(1), \ldots, v_{s'}(p'_{s'})\},$$

where C and C' are the sets of vertical and horizontal base-segments, respectively.

Proof. Let s be a vertical base-segment and I be a maximal independent set of G. Assume $u_s(p) \in I$ for some $p > 1$. Now the neighborhood $N_G(u_s(p))$ of $u_s(p)$ cannot be in I. Thus we remove $N_G(u_s(p))$ from G. In the obtained graph, each $u_s(p')$ with $p' < p$ is an isolated vertex since $N_G(u_s(p')) \subseteq N_G(u_s(p))$. This implies that $\{u_s(1), \dots, u_s(p)\} \subseteq I$. The proof for horizontal base-segments is almost the same. $\square$

Now we prove the main result of this section.

Lemma 5.3. *An optimum solution of an instance of the two-directional k base-segment MWRP has weight at least W if and only if the corresponding bipartite graph G has an independent set of weight at least W.*

Proof. For the only-if part, let R be an inclusion-wise minimal maximum-weight base-monotone region. Let $W = \sum_{(i,j)\in R} w(i,j)$. We shall find an independent set I of G with weight W. For each vertical base-segment $s = (i,j)$, either R has empty intersection with the range of s or R contains the consecutive cells $(i,j), \dots, (a_s(p), j)$ for some $p \geq 1$ (see Remark 5.1). In the latter case, we put the vertices $v_s(1), \dots, v_s(p)$ into I. We do the same thing for each horizontal base-segment. Clearly, I is of weight W. Furthermore, I is indeed an independent set from the construction of G.

For the if part, let I be a maximum-weight independent set of G. Assume that the weight of I is W. Let $s = (i,j)$ be a vertical base-segment. By Lemma 5.2, either I contains no vertex $u_s(\cdot)$ or I contains consecutive vertices $u_s(1), \dots, u_s(p)$ for some $p \geq 1$. In the latter case, we take the consecutive cells $(i,j), \dots, (a_s(p), j)$ as the base-monotone region with the base-segment s. We do the same thing for each horizontal base-segment. The total weight of the taken cells is W. Since I is an independent set of G, the base-monotone regions taken are pairwise disjoint. $\square$

Theorem 5.4. *The two-directional k base-segment MWRP can be solved in $O(k^3 n^6 \log kn)$ time.*

Proof. Given an instance of the two-directional k base-segment MWRP, we first refine each base-segment and construct the corresponding bipartite graph G as described in this section. Clearly, $|U \cup V| = O(kn^2)$, and thus the construction can be done in $O(k^2 n^4)$ time. Next we find the maximum-weight independent set I in G. Since G is bipartite, I can be found in $O(|U \cup V| \cdot |E| \log |U \cup V|) = O(k^3 n^6 \log kn)$ time [11]. By Lemma 5.3, we can construct from I a maximum-weight base-monotone feasible region with respect to the given k base-segments in $O(kn^2)$ time. $\square$

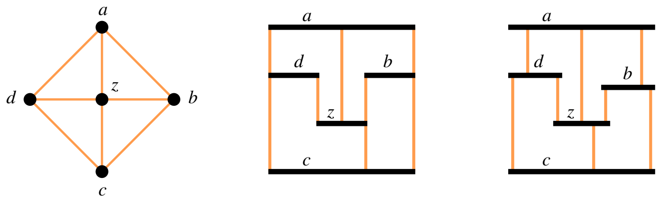

Figure 12: A planar graph. Its visibility and nice visibility representations.

5.2. NP-hardness of the k base-segment MWRP

We now show the following theorem.

Theorem 5.5. *The k base-segment MWRP is NP-complete in the strong sense.*

The problem is clearly in NP, and thus it suffices to show the strong NP-hardness. We reduce INDEPENDENT SET for planar graphs to the k base-segment MWRP. A graph is *planar* if it can be drawn in the plane without edge crossings. It is known that INDEPENDENT SET is NP-hard even for planar graphs [8].

Nice visibility representations. Planar graphs have several geometric representations. We use one of them here. A $w \times h$ *grid* is the subset $\{1, 2, \ldots, w\} \times \{1, 2, \ldots, h\}$ of the plane. A *visibility representation* of a planar graph G maps each vertex of G to a horizontal segment with endpoints in a grid and each edge of G to a vertical segment with endpoints in a grid such that

1. no segments of two distinct vertices intersect,
2. segments of two distinct edges intersect only at their endpoints, and
3. the segment of an edge $\{u, v\}$ touches the segments of u and v.

See Figure 12 for an example. Otten and van Wijk [12] showed that every planar graph has a visibility representation. It is known that a visibility representation of a planar graph in an $O(n) \times O(n)$ grid can be found in linear time (see [13–15]). Here we need the following additional conditions for representations:

4. no two vertical segments have the same x-coordinate,
5. no two horizontal segments have the same y-coordinate, and
6. no two endpoints of segments have the same position.

We call a visibility representation satisfying the three additional conditions a *nice visibility representation.* Given a visibility representation of a planar graph, we can obtain a nice visibility representation of the graph in polynomial time by refining each cell of the grid into a $3n \times 3n$ subgrid, extending each horizontal segment to both directions by one pixel, and shifting each vertical segment to break the same x-coordinate. The shifting of each vertical segment can be done since there are at most $3n - 6$ edges and there are $3n$ new x-coordinates for the original x-coordinate. Note that each segment in this representation has length at least $6n$.

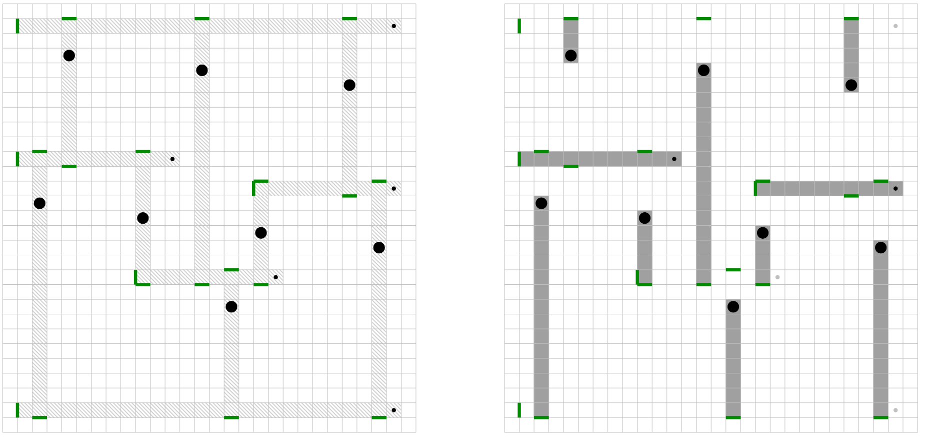

Figure 13: Each green thick segment is a base-segment. In the right figure $S = \{b, d\}$.

Reduction. Let (G, s) be an instance of Independent Set, where G is a planar graph with n vertices and m edges. Note that we do not assume $n = m$ here. We first construct a nice visibility representation $R = (A, B)$ of G in polynomial time, where A is the set of horizontal segments and B is the set of vertical segments. We construct a pixel grid P from R as follows (see Figure 13).

Vertex gadget: For each vertex $u \in V$ with the corresponding horizontal segment $a_u = [x_1, x_2] \times \{y\} \in A$, we put a vertical base-segment (x_1, y) and set the weight 1 to the cell (x_2, y).

Edge gadget: For each edge $e = \{v, w\} \in E$ with the corresponding vertical segment $b_u = \{x\} \times [y_1, y_2] \in B$, we put horizontal base-segments (x, y_1) and $(x, y_2 + 1)$ and set the weight n to the cell (x, y_e), where the y-coordinate y_e is not used by any vertical base-segment and $y_1 < y_e < y_2$. Such a coordinate can be chosen since each segment has length at least yn.

Weight of gadgets: Note that the weight of a cell is at most n and there is no negative-weight cell.

Remark 5.6. *For each base-segment in the construction above, there is only one cell with positive weight that can be taken by the base-segment.*

Equivalence. We now show that (G, s) is a yes-instance if and only if the optimum value of k base-segment MWRP on P is at least $mn + s$. Since the weight of each cell is polynomially bounded, the problem is NP-hard in the strong sense.

For the only-if part, let S be an independent set of G with $|S| \geq s$. We first take $|S|$ positive cells of weight 1 by the vertical base-segments of vertices in S. For each edge $e = \{u, v\} \in E$, P contains two horizontal base-segments. Since S is an independent set, at least one of them can be used to take the corresponding positive cell of weight n (see Figure 13). Therefore, we can take the cells of total weight at least $mn + |S| \geq mn + s$.

For the if part, first observe that we must take all positive cells of weight n since otherwise the total sum is at most $mn < mn+s$. Thus we use one of the two horizontal base-segments for each edge. This implies that for each edge $\{u, v\}$, we can take at most one positive cell of weight 1 using the corresponding vertical base-segment of u or v. Let S be the set of vertices such that the corresponding vertical base-segments are used to take their positive cells of weight 1. By the observation above, S is an independent set of size at least s. This completes the proof.

The three-directional version. In the reduction above, we may assume without loss of generality that the region can be built only on the right side of each vertical base-segment, on the upper sides of some horizontal base-segments, and on the lower sides of the remaining horizontal base-segments. We call this version the three-directional k base-segment MWRP.

Corollary 5.7. *The three-directional k base-segment MWRP is NP-complete in the strong sense.*

6. Polynomial-Time Algorithm for Quad Decomposition

Recall that Quad Decomposition is the problem of finding a depth d quad decomposition of P that maximizes the total sum of the weight of the optimum solution of Room-Edge Problem for the subgrids in the decomposition.

A dynamic programming approach allows us to have the following result.

Theorem 6.1. Quad Decomposition *can be solved in $O(n^7)$ time.*

Proof. For $0 \le i \le j \le n$ and $0 \le s \le t \le n$, let $P_{(i,s),(j,t)}$ be the submatrix of P with the bottom-left point (i, s) and the top-right point (j, t). Let $A^{(0)}_{(i,s),(j,t)}$ be the weight of an optimum solution of Room-Edge Problem in $P_{(i,s),(j,t)}$. The entries $A^{(0)}_{(i,s),(j,t)}$ for all i, j, s, t can be computed in $O(n^7)$ time by using the $O(n^3)$-time algorithm in [4].

For $\delta \ge 1$, let $A^{(\delta)}_{(i,s),(j,t)}$ be the weight of an optimum solution of the depth δ Quad Decomposition in $P_{(i,s),(j,t)}$. It is not difficult to see that

$$A^{(\delta)}_{(i,s),(j,t)} = \max_{i<p<j,\ s<q<t} \left(A^{(\delta-1)}_{(i,s),(p,q)} + A^{(\delta-1)}_{(p,s),(j,q)} + A^{(\delta-1)}_{(i,q),(p,t)} + A^{(\delta-1)}_{(p,q),(j,t)} \right).$$

See Figure 14. Hence each entry $A^{(\delta)}_{(i,s),(j,t)}$ can be computed in $O(n^2)$ time with precomputed entries $A^{(\delta-1)}_{(i',s'),(j',t')}$ for all i', j', s', t', and thus all the entries $A^{(\delta)}_{(i,s),(j,t)}$ can be computed in $O(n^6)$ time in total. Clearly, the weight of an optimal solution for the depth d Quad Decomposition is $A^{(d)}_{(0,n),(0,n)}$. This entry will be computed in $O(n^7 + d \cdot n^6)$ time. Since $d \in O(\log n)$, the theorem holds. □

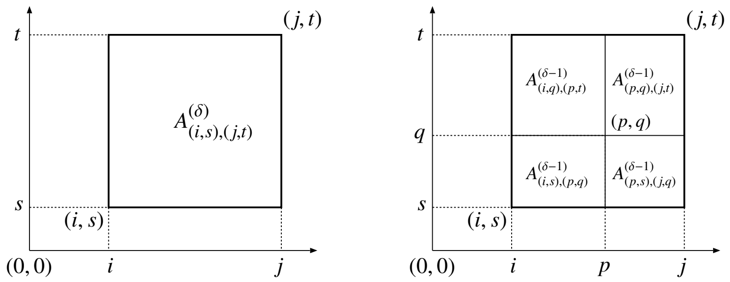

Figure 14: Computing $A^{(\delta)}_{(i,s),(j,t)}$ from the entries of $A^{(\delta-1)}$.

The bottleneck of the running time above is the first phase of solving Room-Edge Problem for all the possible $O(n^4)$ subgrids. Using techniques developed in the study of the all-pairs shortest path problem, we can slightly improve the running time of the first phase as follows.

Given $s \times t$ and $t \times r$ real matrices $A = (a_{i,j})$ and $B = (b_{i,j})$, the *funny product* (or the *distance product*) $A \odot B$ is the $s \times r$ matrix $C = (c_{i,j})$ with $c_{i,j} = \max_{1 \le k \le n}(a_{i,k} + b_{k,j})$. It is known that the computational complexity of funny matrix multiplication is equivalent to that of all-pairs shortest path problem in weighted directed graphs (see [1, Section 5.9]). We can show that the first phase involves funny matrix multiplication. Using the current best algorithm for funny matrix multiplication by Han and Takaoka [10], we can present an $O(n^7 \log\log n / \log^2 n)$-time algorithm for Quad Decomposition.

7. Concluding Remarks

Base-Line Location and related problems are studied as formulations of image segmentation problems. In this paper, although we believe that these problems can arise in practical settings, we focused on their theoretical aspects and studied their computational complexity. Experimental results of k base-line MWRP and Quad Decomposition for image segmentation can be found in [3, 4].

From a computational-complexity point of view, it would be interesting to ask the fixed-parameter tractability of Base-Line Location with parameter k, the number of base-lines.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Efficient algorithms for optimization-based image segmentation. *Internat. J. Comput. Geom. Appl.*, 11:145–166, 2001.

[3] J. Chun, T. Horiyama, T. Ito, N. Kaothanthong, H. Ono, Y. Otachi, T. Tokuyama, R. Uehara, and T. Uno. Algorithms for computing optimal image segmentation using quadtree decomposition. In *TJJCCGG*, pages 23–24, 2012.

[4] J. Chun, N. Kaothanthong, R. Kasai, M. Korman, M. Nöllenburg, and T. Tokuyama. Algorithms for computing the maximum weight region decomposable into elementary shapes. *Comput. Vis. Image Und.*, 116:803–814, 2012.

[5] J. Chun, R. Kasai, M. Korman, and T. Tokuyama. Algorithms for computing the maximum weight region decomposable into elementary shapes. In *20th International Symposium on Algorithms and Computation (ISAAC '09)*, volume 5878 of *Lecture Notes in Comput. Sci.*, pages 1166–1174, 2009.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 59:167–181, 2004.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[8] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1:237–267, 1976.

[9] M. Gibson, D. Han, M. Sonka, and X. Wu. Maximum weight digital regions decomposable into digital star-shaped regions. In *22nd International Symposium on Algorithms and Computation (ISAAC 2011)*, volume 7074 of *Lecture Notes in Comput. Sci.*, pages 724–733, 2011.

[10] Y. Han and T. Takaoka. An $O(n^3 \log\log n/\log^2 n)$ time algorithm for all-pairs shortest paths. In *13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2012)*, volume 7357 of *Lecture Notes in Comput. Sci.*, pages 131–141, 2012.

[11] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Appl. Math.*, 6:243–254, 1983.

[12] R. H. J. M. Otten and J. G. van Wijk. Graph representations in interactive layout design. In *IEEE Internat. Symp. on Circuits and Systems*, pages 914–918, 1978.

[13] P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.*, 1:343–353, 1986.

[14] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1:321–341, 1986.

[15] J.-J. Wang and X. He. Compact visibility representation of plane graphs. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, volume 9 of *LIPIcs*, pages 141–152, 2011.