Base location problems for base-monotone regions

Jinhee Chun[1], Takashi Horiyama[2], Takehiro Ito[1], Natsuda Kaothanthong[1], Hirotaka Ono[3], Yota Otachi[4], Takeshi Tokuyama[1], Ryuhei Uehara[4], and Takeaki Uno[5]

[1] Graduate School of Information Sciences, Tohoku University. Sendai 980-8579, Japan. takehiro@ecei.tohoku.ac.jp, {jinhee,natsuda,tokuyama}@dais.is.tohoku.ac.jp

[2] Graduate School of Science and Engineering, Saitama University. Saitama 338-8570, Japan. horiyama@al.ics.saitama-u.ac.jp

[3] Department of Economic Engineering, Kyushu University. 6-19-1 Hakozaki Higashi-ku, Fukuoka 812-8581, Japan. hirotaka@en.kyushu-u.ac.jp

[4] School of Information Science, Japan Advanced Institute of Science and Technology. Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. {otachi,uehara}@jaist.ac.jp

[5] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan. uno@nii.ac.jp

Abstract. The problem of decomposing a pixel grid into base-monotone regions was first studied in the context of image segmentation. It is known that for a given $n \times n$ pixel grid and baselines, one can compute in $O(n^3)$ time a maximum-weight region that can be decomposed into disjoint base-monotone regions [Chun et al. ISAAC 2009]. To complement this fact, we first show the NP-hardness of the problem of optimally locating k baselines in a given pixel grid. Next we present an $O(n^3)$-time 2-approximation algorithm for this problem. We also study some polynomial-time solvable cases, and variants of the problem.

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 is $(1, 1)$ and the top-right cell is (n, n). Each pixel $p = (i, j)$, where $1 \leq i, j \leq n$, 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 $\mathcal{F} \subseteq 2^P$ be a fixed family of pixel regions.

The general problem MWRP has been studied for several families $\mathcal{F}$ that are related to practical problems. 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

Fig. 1. Image segmentation via k baseline MWRP. In this example, the edges of the picture is used as baselines ($k = 4$). For example, the red region in the third figure (from left) uses the top edge as its baseline.

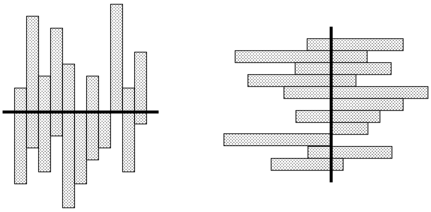

Fig. 2. A based x-monotone region (left) and a based y-monotone region (right).

problem becomes NP-hard [2]. For the complexity of MWRP for other families, see the paper by Chun et al. [5, 4] and the references therein.

Motivated by the image segmentation problem, Chun et al. [5] studied more complicated family of pixel regions for MWRP (see Fig. 1). A *baseline* of an $n \times n$ pixel grid P is a vertical line $x = b$ or horizontal line $y = b$, where $0 \leq b \leq n$. A pixel region R is a *based x-monotone region* if there is a horizontal baseline $y = b$ such that $(i, j) \in R$ implies $(i, j-1) \in R$ for $j \geq b + 1$, and $(i, j) \in R$ implies $(i, j+1) \in R$ for $j < b$ (see Fig. 2). *Based y-monotone regions* are analogously defined. Based x- and y-monotone regions are *base-monotone regions*. Given a set of k baselines, a region R is *base-monotone feasible* if it can be decomposed into pairwise disjoint base-monotone regions with respect to the baselines. The *k baseline MWRP* is MWRP in which we are given k (vertical or horizontal) baselines, and we find a maximum-weight base-monotone feasible region respect to the baselines.

Chun et al. [5] observed that the complement of a maximum-weight base-monotone feasible region represents an object in a picture nicely if the baselines are located reasonably (see Fig. 3). They showed that the k baseline MWRP can be solved in polynomial time. They also studied the *k base-segment MWRP*, in which we are given k segments and find a region decomposable into base-monotone regions respect to the given base-segments. (We will define this problem more precisely in the next section.) They showed some partial results on this problem. For other formulations, as optimization problems, of the image segmentation problem, see the recent work by Gibson et al. [8].

In the setting of the k baseline MWRP, we are given k baselines. Thus a natural question would be "*What if baselines are not given?*" In other words,

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

"*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 baselines is NP-hard but can be approximated within factor 2. Next we propose another way to divide the pixel grid into subgrids, and show that this variant can be solved in polynomial time. Finally, we study the k base-segment MWRP and present sharp contrasts of its computational complexity.

Due to space limitation all proofs are omitted. (They are included in the appendices in the submission version.)

2 Definitions of the three problems

2.1 Baseline Location

To complement the result by Chun et al. [5], who showed that the k baseline MWRP can be solved in $O(n^3)$ time, we study the computational complexity of the following problem.

Problem: Baseline Location
Instance: An $n \times n$ pixel grid P and positive integers k and w.
Question: Is there k baselines 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 baselines. Thus Baseline 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 $O(f(k) \cdot \text{poly}(n))$ time or even in $O(\text{poly}(k+n))$ time. Unfortunately, the latter case very unlikely happens as we will prove the problem is NP-hard if k is a part of the input. The possibility of the former case remains unsettled in this paper.

2.2 The k base-segment MWRP

Consider a segment s in a baseline ℓ. If a monotone region R with baseline ℓ intersects ℓ only in s, then R has s as its *base-segment*. Chun et al. [5] also 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.

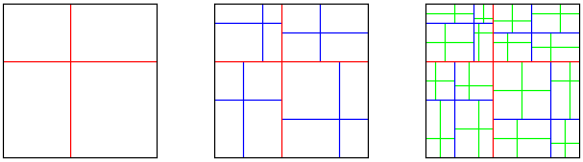

Fig. 4. Quad decompositions of depth 1, 2, and 3.

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.*

It was not known whether the problem is NP-hard when k is a part of the input and whether the two-directional version can be solved in polynomial time with both n and k. We will present affirmative answers to these questions.

2.3 Quad Decomposition

Chun et al. showed that solving the k baseline MWRP is equivalent to solving the following problem for each subgrid obtained by the given baselines. They actually showed that the following problem can be solved in $O(mn^2)$ time.

Problem: Room-Edge Problem
Instance: An $m \times n$ pixel grid P.
Objective: Find a maximum-weight base-monotone feasible region with the four baselines $x = 0$, $x = m$, $y = 0$, and $y = n$.

We solve the Room-Edge Problem for each subgrid, and then answer their total weight as one for the baseline MWRP. From this point 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 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 Fig. 4) We call the resultant set of subgrids a *depth d quad decomposition* of P. Now our problem can be defined as follows.

Problem: Quad Decomposition
Instance: An $n \times n$ pixel grid P and positive integers d and w.
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 $d \in O(\log n)$ since otherwise the problem becomes trivial (we can take all positive cells). We will show that this problem can be solved in polynomial time. In the context of image segmentation, we may expect that the quad decomposition works well compared with k baseline decomposition. This is because, by using quad decompositions, we can place many bases in complicated parts of the image.

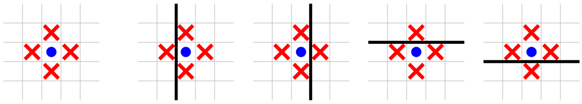

Fig. 5. A baseline forcer: forcing one baseline.

3 NP-hardness of Baseline Location

Here we prove the following theorem.

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

The problem is clearly in NP. We prove its 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 [6].

Problem: Independent Set
Instance: A graph G and a positive integer s.
Question: Does G have an independent set of size at least s?

3.1 Gadgets

We first define two small gadgets for forcing baselines into restricted zones. Throughout this paper, each red $\times$ in a pixel grid represents a huge negative weight whose absolute value is equal to the sum of all the positive weights in the grid. Also, each blue $\bullet$ represents a (not necessarily large) positive weight. All the other cells have weight 0.

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

Next we consider a similar gadget depicted in Fig. 6. To take all the positive cells and not to take any negative cell, we need either one vertical baseline or two horizontal baselines. Therefore, if we need to minimize the number of baselines, then we have to use one vertical baseline. We call this gadget a *vertical baseline forcer*. By rotating this gadget, we can also obtain a gadget for forcing two vertical baselines or one horizontal baseline. We call it a *horizontal baseline forcer*. Two positive cells in this gadget have the same weight, and their weight is the *weight* of the vertical or horizontal baseline forcer. The *position* of a vertical or horizontal baseline forcer is the position of its bottom-left cell.

Vertical and horizontal baseline forcers work even if we insert some space between columns or rows as in Fig. 6. The location of the baseline is restricted to the area depicted in the figure. We say that a vertical (horizontal) baseline forcer *intersects* a vertical (horizontal resp.) baseline if the baseline is in the

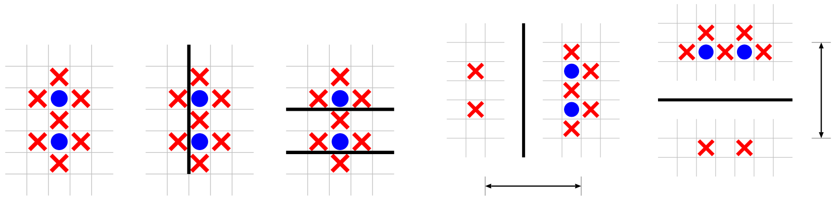

Fig. 6. (Left) A vertical baseline forcer: forcing one vertical baseline. (Right) Forced baselines are restricted to the area indicated by double headed arrows.

restricted area; that is, a base monotone shape with the vertical or horizontal baseline can contain the positive cells in the vertical or horizontal baseline forcer. The number of the columns used by a vertical baseline forcer is its *width*, and the number of rows used by a horizontal baseline forcer is its *height*. For example, the original vertical baseline forcer in Fig. 6 is of width 3.

3.2 Reduction

Given an instance (G, s) of Independent Set, we construct an instance (P, k, w) of Baseline 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.

In the following, we assume $|V(G)| = |E(G)|$ for notational convenience. (It is easy to see that Independent Set is NP-hard even if $|V(G)| = |E(G)|$.) Let $V(G) = \{v_1, \ldots, v_m\}$ and $E(G) = \{e_1, \ldots, e_m\}$. We set the number of baselines $k = 2m$ 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 Fig. 7).

Vertex gadgets For each vertex v_i, we put a vertical baseline forcer of width 5 and weight $2m^2 + m$, denoted VF_i, at the position $(10i, 5i)$. We also put a baseline 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 baseline forcer of height 10 and weight $2m^2 + m$, denoted HF_h, at the position $(10m + 5h, 5m + 15h)$. Next we put two horizontal baseline 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 baseline 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 grids:

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

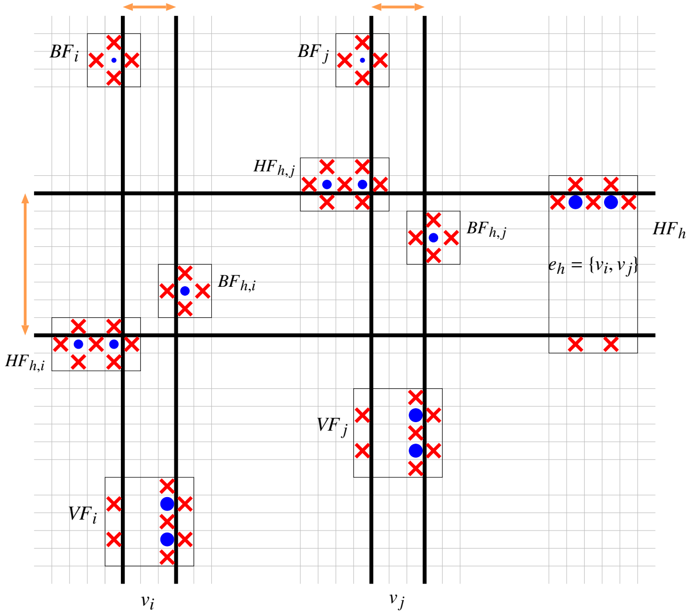

Fig. 7. Gadgets for an edge $\{v_i, v_j\}$: black thick lines are the candidates of required baselines, two vertical and one horizontal.

The total weight of the positive cells is $W = 4m(2m^2 + n) + 6m^2 + n = 8m^3 + 10m^2 + m$. 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

Lemma 3.2. *(G, s) is a yes-instance of* Independent Set *if and only if (P, k, w) is a yes-instance of* Baseline Location*.*

4 A 2-approximation algorithm for Baseline Location

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

Problem: Vertical Baseline Location
Instance: An $n \times n$ pixel grid P and a positive integer k.
Objective: Find k vertical baselines in P that maximize the weight of an optimal base-monotone feasible region respect to these baselines.

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

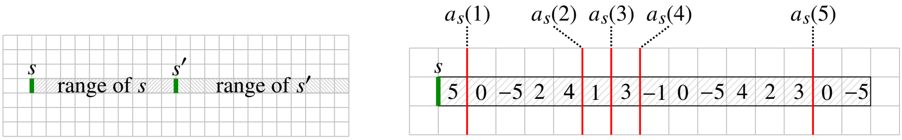

Fig. 8. (Left) The ranges of vertical base-segments s and s'. (Right) Example of $a_s(p)$. The corresponding weights $w_s(1), \ldots, w_s(5) = 5, 1, 1, 3, 3$.

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

We solve Vertical Baseline Location with k vertical baselines and Horizontal Baseline Location with k horizontal baselines 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 *baselines to maximize the weight of optimum base-monotone feasible region.*

5 The k base-segment MWRP

We extend the results of Chun et al. [5] (Theorem 2.1). We first reduce the two-directional version to Weighted Independent Set in bipartite graphs, which can be solve in polynomial time [10]. We next reduce the Independent Set in planar graphs to the original problem. This implies the NP-hardness of the original problem, since Independent Set is NP-hard for planar graphs [7].

5.1 Two-directional version

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 $[i, i']$, where i' is the smallest index for which such a segment exists (see Fig. 8). 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 Fig. 8. For each horizontal base-segment s', we also define the sequence $a_{s'}(\cdot)$ analogously.

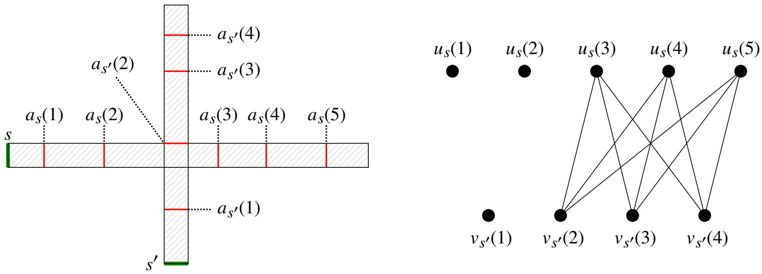

Fig. 9. The bipartite graph construction. The vertices corresponding to the crossing ranges of two base-segments induce the disjoint union of an independent set and a complete bipartite graph.

Now we construct a bipartite graph $G = (U, V; E)$. Let $s = (i, j)$ be a vertical base-segment. Assume that r is the largest index such that $a_s(r)$ is defined. Now all $a_s(0), \ldots, a_s(r)$ are defined by the definition. If $r = 0$, then this segment s is useless and ignored. Otherwise, we put vertices $u_s(p)$, $1 \leq p \leq r$, 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 Fig. 9 for example.

Lemma 5.1. *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.*

Theorem 5.2. *The two-directional k base-segment MWRP can be solved in $O(k^3 n^6 \log kn)$ time.*

5.2 NP-hardness of the k base-segment MWRP

We now show the following theorem.

Theorem 5.3. *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 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 [7].

Nice visibility representations 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

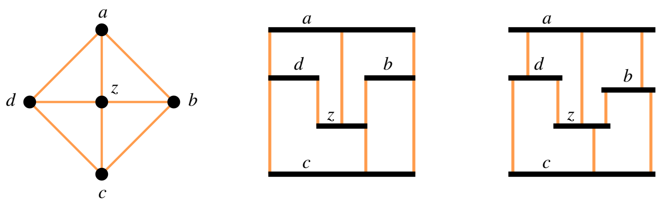

Fig. 10. A planar graph. Its visibility and nice visibility representations.

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 Fig. 10 for example. Otten and van Wijk [11] 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 [12–14]. Additionally, we need the following 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 to an $O(n) \times O(n)$ subgrid, slightly extending each horizontal segment, and slightly shifting each vertical segment.

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 Fig. 11).

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). 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 by the refinement of the grid. Note that the weight of a cell is at most n and there is no negative-weight cell.

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 $mn + s$. (The proof is omitted.)

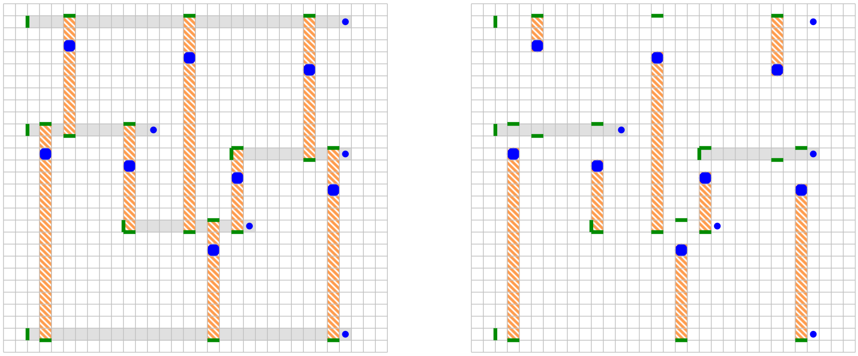

Fig. 11. Each green thick segment is a base-segment. In the right figure $S = \{b, d\}$.

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 5.4. *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.*

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 used in the study of the all-pairs shortest path problem, we can slightly improve the running time of the first phase.

Given $s \times t$ and $t \times r$ real matrices $A = (a_{i,j})$ and $B = (b_{i,j})$, the *funny* matrix product $A \odot B$ is the $s \times r$ matrix $C = (c_{i,j})$ with $c_{i,j} = \max_{1 \leq k \leq n}(a_{i,k} + b_{k,j})$. 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 [9], we have the following result.

Theorem 6.2. Quad Decomposition *can be solved in $O(n^7 \log\log n / \log^2 n)$ time.*

7 Concluding remarks

Baseline Location and related problems are introduced as formulations of image segmentation problems. However, in this paper, we focused on their theoretical aspects and studied their computational complexity. We believe that these problems can be arisen in practical settings. Experimental results of using k-baseline MWRP for image segmentation can be found in [3, 4].

It would be interesting to ask the fixed parameter tractability of Baseline Location with parameter k, the number of baselines.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)
2. Asano, T., Chen, D.Z., Katoh, N., Tokuyama, T.: Efficient algorithms for optimization-based image segmentation. Internat. J. Comput. Geom. Appl. 11, 145–166 (2001)
3. Chun, J., Horiyama, T., Ito, T., Kaothanthong, N., Ono, H., Otachi, Y., Tokuyama, T., Uehara, R., Uno, T.: Algorithms for computing optimal image segmentation using quadtree decomposition, in preparation
4. Chun, J., Kaothanthong, N., Kasai, R., Korman, M., Nöllenburg, M., Tokuyama, T.: Algorithms for computing the maximum weight region decomposable into elementary shapes. Comput. Vis. Image Und. 116, 803–814 (2012)
5. Chun, J., Kasai, R., Korman, M., Tokuyama, T.: Algorithms for computing the maximum weight region decomposable into elementary shapes. In: 20th International Symposium on Algorithms and Computation (ISAAC '09). Lecture Notes in Comput. Sci., vol. 5878, pp. 1166–1174 (2009)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman (1979)
7. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoret. Comput. Sci. 1, 237–267 (1976)
8. Gibson, M., Han, D., Sonka, M., Wu, X.: Maximum weight digital regions decomposable into digital star-shaped regions. In: 22nd International Symposium on Algorithms and Computation (ISAAC 2011). Lecture Notes in Comput. Sci., vol. 7074, pp. 724–733 (2011)
9. Han, Y., Takaoka, T.: 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). Lecture Notes in Comput. Sci., vol. 7357, pp. 131–141 (2012)
10. Hochbaum, D.S.: Efficient bounds for the stable set, vertex cover and set packing problems. Discrete Appl. Math. 6, 243–254 (1983)
11. Otten, R.H.J.M., van Wijk, J.G.: Graph representations in interactive layout design. In: IEEE Internat. Symp. on Circuits and Systems. pp. 914–918 (1978)
12. Rosenstiehl, P., Tarjan, R.E.: Rectilinear planar layouts and bipolar orientations of planar graphs. Discrete Comput. Geom. 1, 343–353 (1986)
13. Tamassia, R., Tollis, I.G.: A unified approach to visibility representations of planar graphs. Discrete Comput. Geom. 1, 321–341 (1986)
14. Wang, J.J., He, X.: Compact visibility representation of plane graphs. In: 28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011). LIPIcs, vol. 9, pp. 141–152 (2011)

A Omitted proofs

A.1 Proofs in Section 3

Here we provide the proof of Lemma 3.2. See Fig. 16 for an example of the reduction.

Lemma A.1. *If* (G, s) *is a yes-instance of* Independent Set*, then* (P, k, w) *is a yes-instance of* Baseline Location*.*

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

For each vertex $v_i \in S$, we set a vertical baseline at $x = 10i$. For each vertex $v_i \in V(G) \setminus S$, we set a vertical baseline 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 baseline at $y = 5m + 15h + 8$. Otherwise, we set a horizontal baseline at $y = 5m + 15h$. For example, see Fig. 12 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 baseline 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 $v_i \in S$, or $\{BF_{h,i} \mid v_i \in e_h\}$ if $v_i \notin S$. If $v_i \in S$, then the vertical baseline can take one cell of weight 1 in BF_i also.

Let $e_h = \{v_i, v_j\}$ and assume $v_j \notin S$ without loss of generality. The horizontal baseline corresponding to e_h can take two cells of weight $2m^2 + m$ in HF_h. Since $v_j \notin S$, the positive cells of weight m in $HF_{h,j}$ are not taken by any vertical baseline. Hence these two cells can be taken by the horizontal baseline.

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. □

Lemma A.2. *If* (P, k, w) *is a yes-instance of* Baseline Location*, then* (G, s) *is a yes-instance of* Independent Set*.*

To prove this lemma, we need to prove the following propositions.

Proposition A.3. *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 take at most $4m - 1$ cells of weight $2m^2 + m$ or at most $4m - 1$ cells of weight m, then we miss one cell of weight $2m^2 + m$ or $2m + 1$ cells of weight m. Thus the sum of the weights of the cells taken is at most $W - (2m^2 + m) = 8m^3 + 8m^2 < w$. □

Corollary A.4. *If* (P, k, w) *is a yes-instance, then each* VF_i *intersects exactly one vertical baseline, and each* HF_i *intersects exactly one horizontal baseline.*

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

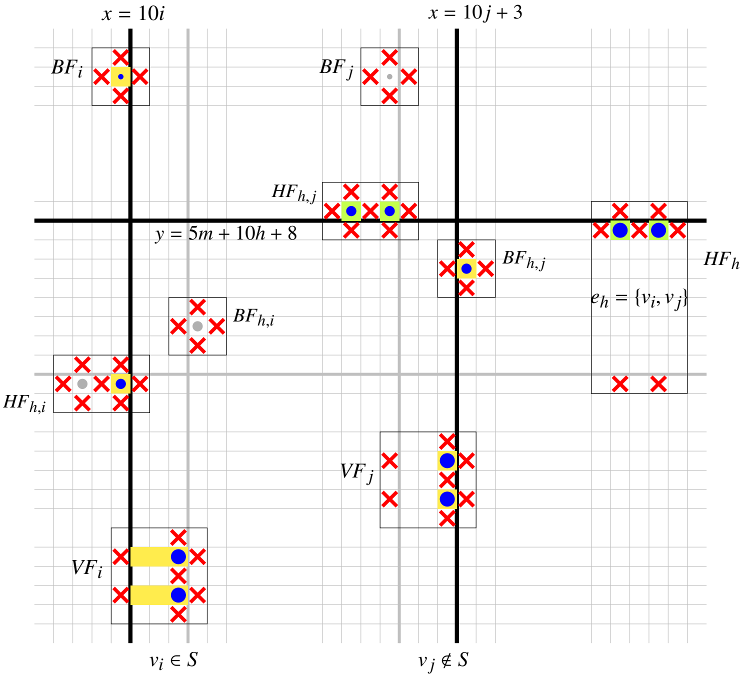

Fig. 12. The case of $\{v_i, v_j\} \in E(G)$ and $v_i \in S$: black thick lines are the selected baselines. Yellow cells are taken by the vertical baselines, and green cells are taken by the horizontal baseline.

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

Proposition A.5. *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 A.4, 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_j 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 A.3, 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. $\square$

A vertex v_i is *left* if VL_i is a vertical baseline $x = 10i$, and v_i is *right* if VL_i is a vertical baseline $x = 10i + 3$. An edge e_h is *top* if HL_h is a horizontal baseline $y = 5m + 15h + 8$, and e_h is *bottom* if HL_h is a horizontal baseline $y = 5m + 15h$. 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 A.5 (see Fig. 7 and Fig. 12).

The following proposition relates BASELINE LOCATION to INDEPENDENT SET.

Proposition A.6. *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 A.5, (P, k, w) is not a yes-instance. □

Now we are ready to prove Lemma A.2.

Proof (Lemma A.2). Let (P, k, w) is a yes-instance of BASELINE 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 baseline $x = 10i$ or $x = 10i+3$ for each vertex v_i, and there is a horizontal baseline $y = 5m + 15h + 8$ or $y = 5m + 15h$ for each edge e_h. These baselines 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 baseline $x = 10i$ 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 A.6, (G, s) is a yes-instance of INDEPENDENT SET. □

A.2 Proofs in Section 4

Theorem A.7. VERTICAL BASELINE LOCATION *and* HORIZONTAL BASELINE LOCATION *can be solved in $O(n^3)$ time.*

Proof (Theorem 4.1). By symmetry, it suffices to show the result only for VERTICAL BASELINE LOCATION. To simplify the presentation, we assume that we can use the vertical lines $x = 0$ and $x = n$ as baselines for free. This can be justified by adding the new first and last columns to the grid and setting huge negative values to the new entries.

For $1 \leq r \leq n$ and $0 \leq i \leq j \leq n$, let $P^r_{i,j}$ be the subgrid $[i, j] \times [r, r + 1]$ of P. Let $A^r_{i,j}$ be the maximum weight of base-monotone regions in $P^r_{i,j}$ with the baselines $x = i$ and $x = j$. Similarly, let $B^r_{i,j}$ be the maximum weight of base-monotone regions in $P^r_{i,j}$ with the baseline $x = i$ only. Clearly $A^r_{i,i} = B^r_{i,i} = 0$. For $i < j$, we have

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

and

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

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

For $0 \le i \le j \le n$, 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 baselines $x = i$ and $x = j$. It is easy to see that $C_{i,j} = \sum_{r=1}^{n} A^r_{i,j}$. Hence we can compute all entries of C in $O(n^3)$ time.

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

$$D_{h,j} = \max_{h-2 \le i < j} \left(D_{h-1,i} + C_{i,j}\right).$$

Using the table C and the entries of D with smaller indices, we can compute $D_{h,j}$ in $O(n)$ time. Thus we can compute all entries of D in $O(n^3)$ time. Now clearly

$$\max_{k-1 \le j \le n} (D_{k,j} + A_{j,n})$$

is the optimal value. Furthermore, by slightly modifying the algorithm, we can easily compute the actual positions of k vertical baselines in the same running time. □

Theorem A.8. *There is an $O(n^3)$-time 2-approximation algorithm for locating k baselines to maximize the weight of optimum base-monotone feasible region.*

Proof (Theorem 4.2). We optimally solve VERTICAL BASELINE LOCATION with k vertical baselines and HORIZONTAL BASELINE LOCATION with k horizontal baselines 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 the half of the best solution of BASELINE LOCATION with k baselines.

Assume that an optimal solution of BASELINE LOCATION is attained with k_v vertical and k_h horizontal baselines, 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 BASELINE 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 BASELINE LOCATION. Assume without loss of generality that $W_v \ge W_h$.

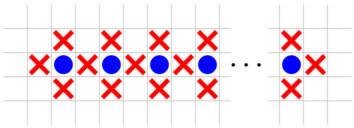

Fig. 13. A generalized horizontal baseline forcer. A generalized vertical baseline forcer can be obtained by rotating this gadget.

Observe that P_v is also a feasible solution of VERTICAL BASELINE LOCATION with k vertical baselines. This is because, additions of baselines never violate the feasibility of base-monotone regions. Therefore, the optimum value of VERTICAL BASELINE LOCATION is at least $W_v \geq (W_v + W_h)/2$. This completes the proof. □

Now we show the tightness of the analysis of the approximation ratio. To this end, we use *generalized horizontal* (*vertical*) *baseline forcers* depicted in Fig. 13. The length of a generalized (horizontal or vertical) baseline forcer is the number of positive cells in it. To take all the positive cells in a generalized horizontal baseline forcer of length ℓ (and not to take any negative cell), we need either one horizontal baseline or ℓ vertical baselines. We construct a tight example by putting one generalized horizontal baseline forcer of length ℓ and one generalized vertical baseline forcer of length ℓ so that no row nor column intersects the baseline forcers. We set the weight 1 to every positive cell. Clearly, we can take all positive cells with one horizontal and one vertical baselines, and thus the optimal solution is of weight 2ℓ. On the other hand, if we use only horizontal baselines (or only vertical baselines), then we can take only $\ell + 1$ positive cells. Therefore, the approximation ratio is $(\ell + 1)/(2\ell) = 1/2 + o(1)$.

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

A.3 Proofs in Section 5.1

To prove Lemma 5.1 and Theorem 5.2, we need the following discussion.

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

Remark A.9. In each minimal optimal solution, the base-monotone region with the vertical base-segment $s = (i, j)$ is either empty or 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'. Thus we have the following lemma.

Lemma A.10. *Each maximal independent set of G is of the form*

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

where C and C' are 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. □

Now we prove Lemma 5.1 and Theorem 5.2.

Lemma A.11. *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 (Lemma 5.1). For the only-if part, let R be a 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 A.9). 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 A.10, 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. □

Theorem A.12. *The two-directional k base-segment MWRP can be solved in $O(k^3 n^6 \log kn)$ time.*

Proof (Theorem 5.2). 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 [10]. From the set I, we can construct a maximum-weight base-monotone feasible region with respect to the given k base-segments in $O(kn^2)$ time. See the if-part proof of Lemma 5.1. □

A.4 Proofs in Section 5.2

Remark A.13. For each base-segment in the construction, there is only one cell with positive weight that can be taken by the base-segment.

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 $mn + s$.

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 Fig. 11). Therefore, we can take the cell 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 some 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-segment 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.

A.5 Proofs in Section 6

Theorem A.14. Quad Decomposition *can be solved in $O(n^7)$ time.*

Proof (Theorem 6.1). For $0 \leq i \leq j \leq n$ and $0 \leq s \leq t \leq n$, 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 $P_{(i,s),(j,t)}$. All $O(n^4)$ entries of $A^{(0)}$ can be computed in $O(n^7)$ time by using the $O(n^3)$-time algorithm in [5].

For $\delta \geq 1$, 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 Fig. 14. Hence each entry of $A^{(\delta)}$ can be computed in $O(n^2)$ time with precomputed matrix $A^{(\delta-1)}$, and thus all entries of $A^{(\delta)}$ 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. □

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 used in the study of the all-pairs shortest path problem, we can slightly improve the running time of the first phase.

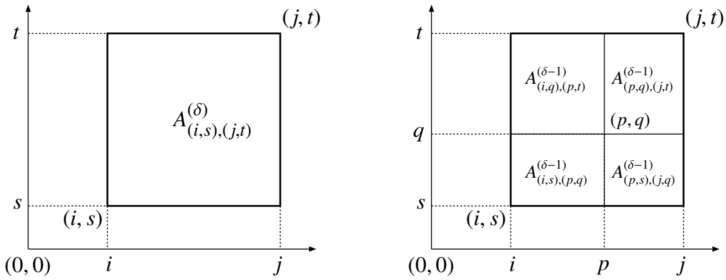

Fig. 14. Computing $A^{(\delta)}_{(i,s),(j,t)}$ from the entries of $A^{(\delta-1)}$.

Theorem A.15. QUAD DECOMPOSITION *can be solved in* $O(n^7 \log\log n/\log^2 n)$ *time.*

Proof (Theorem 6.2). To solve ROOM-EDGE PROBLEM, Chun et al. [5] compute optimal 3-colorings for all L-shaped regions (see Fig. 15), where a 3-coloring consists of base-monotone regions using only three boundaries. They showed that any optimal solution of ROOM-EDGE PROBLEM consists of two optimal 3-colorings of L-shaped regions. In an $n \times n$ pixel grid, there are $O(n^6)$ L-shaped regions. By the dynamic programming algorithm presented by Chun et al. [5], we can compute optimal solutions for all L-shaped regions in $O(n^6)$ total time.

Let $L^{(s,t)}_{i,(a,b,c)}$ be the value of an optimal 3-coloring of the L-shape region depicted in Fig. 15, and let $R^{s,t}_{(a,b,c),j}$ be the value of an optimal 3-coloring of the (rotated) L-shape region depicted in Fig. 15. Note that these six parameters determine the regions. Using these expression, $A^{(0)}_{(i,s),(j,t)}$ can be described as follows [5]:

$$A^{(0)}_{(i,s),(j,t)} = \max_{(a,b,c)} \left\{ L^{(s,t)}_{i,(a,b,c)} + R^{(s,t)}_{(a,b,c),j} \right\}. \tag{1}$$

If we fix the indices s and t, then the right-hand side of Eq. (1) can be seen as the *funny product* (or the *distance product*) of $n \times n^3$ and $n^3 \times n$ matrices. The current fastest algorithm for funny matrix multiplication by Han and Takaoka [9], which runs in $O(N^3 \log\log N/\log^2 N)$ for two $N \times N$ matrices, implies that the funny product of $M \times N$ and $N \times M$ matrices can be computed in $O(NM^2 \log\log M/\log^2 M)$ time. In our case $M = n$ and $N = n^3$, thus we can compute $A^{(0)}_{(i,s),(j,t)}$ in $O(n^5 \log\log n/\log^2 n)$ time for a fixed pair (s,t) and for all pairs (i,j). Hence we can compute all entries of $A^{(0)}_{(i,s),(j,t)}$ in $O(n^7 \log\log n/\log^2 n)$ time for all pairs (s,t) and (i,j). □

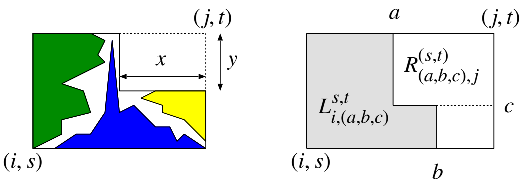

Fig. 15. (Left) A 3-coloring of a L-shaped region. (Right) Combining two L-shaped regions.

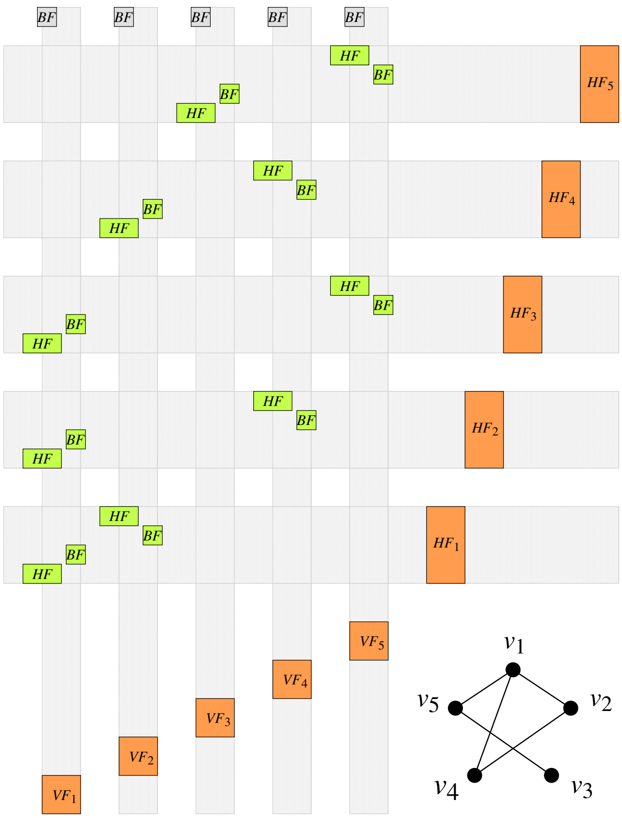

Fig. 16. The reduction for a graph with five vertices $\{v_1, \ldots, v_5\}$ and five edges $\{e_1 = \{v_1, v_2\}, e_2 = \{v_1, v_4\}, e_3 = \{v_1, v_5\}, e_4 = \{v_2, v_4\}, e_5 = \{v_3, v_5\}\}$. Orange gadgets have weight $2m^2 + m$, green gadgets have weight m, and gray gadgets have weight 1.