Katarzyna Paluch

Approximation Algorithms for Rectangle Tiling

Ph.D. Thesis
Supervisor: Prof. Dr. Hab. Krzysztof Loryś

May 18, 2004

Institute of Computer Science
University of Wrocław
ul. Przesmyckiego 20
51-151 Wrocław, Poland

Acknowledgements

First of all I would like to thank my maths and computer science teachers from I Liceum Ogólnokształcące in Głogów: Maria Kajda and Fryderyk Skoczek.

I would like to thank Krzysztof Loryś for encouragement.

There are a lot of people, who read at least parts of this thesis. They are Łukasz Kaiser, Przemka Kanarek, Rafał Kruczek, Piotr Krysta, Domagoj Matijevic, Tibor Różański, Piotrek Wieczorek, Paweł Zalewski. I am very grateful to them.

I would also like to thank Sebastian Bala, Tomek Charatonik, Ola Kaczmarczyk, Marta Kondracka, Marcin Szydzisz, Ania Śledzka, Ania Walczak, Tomek Wierzbicki for all sort of help.

Contents

Chapter 1

Introduction

The RTILE problem belongs to a very wide class of discrete optimization tiling problems. The problems from this class generally require partitioning a multidimensional data set into rectangular partitions so that some optimality criteria are satisfied. In the case of the RTILE problem we are given an array A of nonnegative numbers, a positive number p and the task consists in finding the partition of A into at most p rectangular tiles so as to minimize the maximal weight of the tile. In the partition the tiles are not to overlap and they are to cover the whole array A. By the weight of the tile we mean the sum of all the elements that fall within it. Other problems from this class differ in array dimensions, restrictions on the values of array elements, definitions of metric functions, types of tiles, etc.

These problems find practical applications among others in load balancing in parallel computing environments, data compression, query optimization in database systems. The interested reader may find more details on the applications in the short survey in [5].

As for the types of tiles allowed one can distinguish three main types of tilings: $p \times p$ - where p^2 tiles are induced by choosing p vertical and p horizontal lines ([6], [3]), hierachical - where in the first step of partitioning we choose some lines in one direction and then recursively divide the resulting subarrays [9], and arbitrary - where there are no restrictions on tiles [5].

As far as complexity of the problems is considered, the sharp threshold is imposed by the dimension of the array. For one dimensional arrays several problems are easily solvable in polynomial time. For example, the RTILE problem can be solved by dynamic programming strategy in time $O(np)$. The difficulty of the problems radically changes when we extend them to two dimensions. M. Grigni and F. Manne [3] proved that optimal $p \times p$ tiling is NP-hard and M. Charikar *et.al.* [2] showed that in fact it is NP-hard to approximate it within a factor of 2. For arbitrary tilings S. Khanna, S. Muthukrishnan and M. Paterson [5] showed that the RTILE problem is NP-hard even for the case when the array elements are integers bounded by a constant. Moreover, RTILE remains NP-hard when we relax our demands and look for solutions that are within a factor of $5/4$ of the optimum.

For many years some heuristics were applied in algorithms used in practice to solve

RTILE, however without proved guarantees. The first approximation algorithms with proved ratios appeared in [5]. The authors present a $5/2$-aproximation algorithm for the RTILE problem that works in time $O(n^2 + p\log n)$ and mention that using a similar technique they obtain a $9/4$-approximation algorithm of the same time complexity for binary arrays, i.e. arrays with elements from the set {0, 1}.

Further progress was made by J. Sharp [12] and independently by K. Loryś and K. Paluch [7] who improved the approximation ratio for the RTILE problem to $7/3$. These results were attained using different approaches. In fact, Sharp considers multi-dimensional version of the problem and obtains an algorithm working in time $O(n^d + pd2^d(d + \log n))$ with the appoximation ratio equal to $(d^2 + 2d - 1)/(2d - 1)$. On the other hand, in [7] we lay stress on time efficiency and obtain the algorithm working in linear time. Morever, in [7], we improve several other results from [5]. In particular, we show that our algorithm applied to binary arrays achieves the ratio 2 and is linear, which improves the results from [5] both in approximation ratio and in time. Another advantage over the algorithms from [5] is the simplicity of the produced partitions. Whereas the partitions from [5] are arbitrary, our algorithm gives hierarchical ones with a small depth (at most 2).

Further on approximation for the RTILE problem was successively improved. In particular K. Loryś and K. Paluch [8] gave a $9/4$-approximation algorithm, P. Berman, B. DasGupta, S. Muthukrishnan and S. Ramaswami [1] gave a $11/5$-approximation algorithm and K. Paluch [11] presented an approximation algorithm with the ratio $17/8$. A $17/8$-approximation algorithm is additionally tight with regard to the only known and used lower bound so far.

In the thesis we present two approximation algorithms, with ratios $9/4$ and $17/8$, for the RTILE problem. They, together with a $7/3$-approximation algorithm reflect the gradual development of the approach to the problem. In all of them the array is processed from left to right in a column by column manner. Whenever a currently examined subarray, consisting of a number of columns, satisfies an appropriate criterion, it is partitioned and the algorithm proceeds to the next column. In order to get a $7/3$-approximation, it is enough to consider two-column subarrays. At this stage we identify only two types of columns. To get a $9/4$-approximation, we have to consider subarrays consisting of a greater number of columns. Nevertheless these are still subarrays consisting of a finite number of columns, which is in this case equal to six. Now more classes of columns are identified and based on this we classify subarrays into types. Also, a simple tool is develeped for establishing the upper bound of the weight of the tiles with which we are able to partition a subarray of a given type.

As one can imagine, as we try to further improve the approximation, the situation gets more and more complicated. Also, if we want to obtain a $17/8$-approximation algorithm, it turns out, that we are forced to examine subarrays consisting of arbitrarily many columns. Accordingly, the developed apparatus gets quite involved. Certain properties of subarrays can be expressed using linear programming. Proving them thus amounts to solving corresponding linear programs. However, there are infinitely many such linear programs and they are arbitrarily large. To handle this problem, we devise a special method, which is of independent interest.

The running time of the algorithms is linear in the size of the array, but can be also made to be nearly-linear in the number of non-zero elements in the array. The techniques used in the thesis can be applied to the multidimensional RTILE as well as the binary RTILE.

Chapter 2

Preliminaries

It is clear that any solution to the RTILE problem has to use a tile whose weight is not less than $w(A)/p$, where $w(A)$ denotes the total weight of the input array A and p is the allowed number of tiles. Another value that obviously bounds any solution is the maximal weight of the array element. Therefore let us define

$$W = \max\{\frac{w(A)}{p}, \max\{a_{ij} : 1 \leq i, j \leq n\}\}.$$

This value is used as the lower bound for the optimal solution to the RTILE problem. So far, it is the only known and used lower bound for this problem.

Since we will compare the weights of tiles to the lower bound W, it will be convenient to express the values of array elements in terms of multiplicity of W instead of their original values. Thus we will assume that we deal only with arrays of elements from the interval [0, 1] and the lower bound on the optimal solution is equal to 1.

Let us now define some fundamental notions.

Definition 1 *We say that the array is* f-partitioned *if it is partitioned into rectangles (covered with tiles) of weights not greater than* f*. If we additionally require that the number of tiles used does not exceed* $\lfloor w(A) \rfloor$ *(*$\lceil w(A) \rceil$*, resp.) then we say that the array is* well f-partitioned *(*nearly well f-partitioned*, resp.). The number* f *will be referred to as* a factor.

A simple approach to the solution of the RTILE problem consists in partitioning the input array A into a number of disjoint subarrays $A_1, \ldots, A_l$ and then independent covering each of A_i. It is clear that if each A_i has weight at least 1 and we are able to well f_i-partition it then we get a solution which is at most $\max\{f_i\}$ times worse than the optimum.

Fact 1 *If we partition* A *into subarrays* $A_1, \ldots, A_l$ *with weights at least* 1 *and then well* f_i*-partition each* A_i *then the obtained solution to the RTILE problem has approximation guarantee* $\max\{f_i\}$.

The next fact says that sometimes we can get the same guarantee even if one of the subarrays is not well partitioned.

Fact 2 *Suppose that we partition A into subarrays $A_1, \ldots, A_l, B$ where A_i are as in Fact 1 and B has nonzero weight. Suppose also that we can well f_i-partition each A_i and nearly well f'-partition B. Then,*

we have a solution to the RTILE problem with approximation guarantee $\max\{f_i, f'\}$.

Proof. The only thing that requires explanation is the number of the tiles used. Since $w(A) = \sum_{i=1}^{l} w(A_i) + w(B) \geq \sum_{i=1}^{l} \lfloor w(A_i) \rfloor + w(B)$, we have

$$\lceil w(A) - \sum_{i=1}^{l} \lfloor w(A_i) \rfloor \rceil = \lceil w(A) \rceil - \sum_{i=1}^{l} \lfloor w(A_i) \rfloor \geq w(A) - \sum_{i=1}^{l} \lfloor w(A_i) \rfloor \geq w(B)$$

As $\sum_{i=1}^{l} \lfloor w(A_i) \rfloor$ upper bounds the number of tiles used for covering all A_i's, and $p \geq \lceil w(A) \rceil$ by definition, we conclude that we have still at least $\lceil w(B) \rceil$ tiles for covering B. □

In order to get the first simple $2\frac{1}{3}$-approximation, let us look at the array as a collection of columns and distinguish two classes of them: those with weight at least 1 ($>$*-columns*) and those with weight less than 1 ($<$*-columns*). From [7] we have:

Lemma 1

(a) Any $>$-column can be well 2-partitioned;

(b) any two-column subarray consisting of a $<$-column and a $>$-column can be well 7/3-partitioned;

(c) a group of adjacent $<$-columns of overall weight at least 1 *can be well-*2*-partitioned,*

Therefore we know, that a $2\frac{1}{3}$-approximation is attainable.

By the above we know, that we can assume that in the array there are no adjacent $<$-columns. Thus without loss of generality we can assume that every array consists of alternating $<$- and $>$-columns and begins with a $<$-column. It will sometimes be achieved by inserting artificial columns of weight 0.

Chapter 3

A $2\frac{1}{4}$-approximation algorithm

3.1 Classification into types

We are going to identify more classes of columns.

Definition 2 *For every natural m, a column of type m, also referred to as an m-column, denotes a $>$-column having weight from the interval $[m, m+1)$.*

The type of an array or a subarray is going to be described by the types of its colums given in the order of their occurrence. A column of type m will be represented by a natural number m and a $<$-column by a symbol $\diamond$. The type of a subarray or an array will be given in the form $(\diamond)m_1 \diamond m_2 \ldots \diamond m_n(\diamond)$.

3.2 Two-column subarrays

When we look closer at two-column subarrays, it will turn out, that only one specified kind of them cannot be well-2-partitioned. These, difficult for partitioning, subarrays will be called *complexes*. In Lemma 3 we show that complexes are subarrays of type $\diamond 2$, and then in Lemma 5 we determine their weight more precisely. At the end we introduce the notion of *boundary* and formulate some related properties of the complexes.

Lemma 2 *For each positive integer k, any subarray A of the type $\diamond 1 \diamond 1 \ldots \diamond 1 \diamond$, with k ones and total weight not greater than $2k$, can be well 2-partitioned.*

Proof. (by induction on k) If $k = 1$, then the total weight of A is less than 2 and A can be covered by one tile.

Now let $k \geq 2$ and let $A = B \diamond C$, where B is the subarray consisting of the first two columns of A.

If $w(B) \geq 2$ then we can cover each column of B separately using two tiles. Since the subarray $\diamond C$ contains $k - 1$ ones and has weight at most $2(k - 1)$, we can well 2-partition it by induction.

If $w(B) < 2$, then we consider the subarray $B\diamond$. If it has weight exceeding 2, then we use 2 tiles to cover it: one tile for B and one for the $<$-column . Otherwise we cover $B\diamond$ with one tile and reduce the task to covering C, which is of type $1\diamond \dots 1\diamond$. But this type can be easily well 2-partitioned: simply we separately consider each subarray $1\diamond$ and cover it with one tile if its weight is at most 2 or with two tiles otherwise. $\square$

Lemma 3 *Every subarray A of the type $\diamond m$, where $m \neq 2$, can be well 2-partitioned.*

Proof. If A is of type $\diamond 1$, then its weight is less than 3. If it is greater than 2, then we can divide A vertically, otherwise we can cover A with one tile.

Thus let us assume that $m \geq 3$. Let us look at A^T - the transposition of A. All columns of A^T have weight less than 2, so A^T is of type $\diamond 1 \dots 1\diamond$. Let k be the number of ones in the type.

If the weigth of A does not exceed $2k$ then A can be well 2-partitioned by Lemma 2. Otherwise we cover the $>$-column of A with $k+1$ tiles of weights less than 2, and use an extra tile for covering the $<$-column . What remains is to check if $k+2$ does not exceed the allowed number of tiles. Note that from $m \geq 3$ it follows that $k \geq 2$. Therefore $k+2 \leq 2k$, which is the allowed number of tiles, because we consider the case of $w(A) > 2k$. $\square$

Due to this lemma, two-column subarrays that can be difficult for well 2-partitioning are of type $\diamond 2$. Using the next lemma we shall narrow this class to the subarrays that additionaly have weight between 2.5 and 3.

Lemma 4 *Any column with weight L can be partitioned into n rectangles so that the maximal weight of the tile will not exceed $\frac{L}{n} + \frac{n-1}{n}$.*

Proof. (by induction on n) If $n = 1$ then the lemma is clearly true.

Now, assume that it is true for every $k \leq n$. We want to show the appropriate partition into $(n+1)$ parts. We proceed as follows. Find such an element a_i of the column that $\sum a_i \geq \frac{L}{n+1}$. One can imagine the situation as in Figure 3.1, with the column having length proportionate to its weight and the elements marked accordingly.

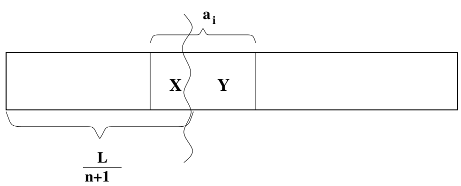

Figure 3.1:

Then we can move the waved line either to the left or to the right, forming this way the first rectangle and partition the rest into n parts. The dividing line is moved to the left if $x \leq \frac{1}{n+1}$ and otherwise to the right. In the first case the remaining part has weight at most $\frac{n}{n+1}L + \frac{1}{n+1}$ and can be by induction partitioned into n parts so that the maximal weight of a tile does not exceed $\frac{\frac{nL+1}{n+1}}{n} + \frac{n-1}{n} = \frac{L}{n+1} + \frac{n}{n+1}$. In the latter case the first rectangle has weight at most $\frac{L}{n+1} + \frac{n}{n+1}$ (because $x + y \leq 1$ and $x \geq \frac{1}{n+1}$) and the rest has weight less than the right part of the column in the previous case, hence can be partitioned even better, which means that the maximal weight of the tile in that part will also not surpass $\frac{L}{n+1} + \frac{n}{n+1}$. □

A simple consequence of Lemma 4 is:

Corollary 1 *Any $>$-column can be well 2-partitioned.*

Lemma 5 *Any $\diamond 2$ array A whose total weight does not belong to the interval $(2.5, 3)$ can be well 2-partitioned.*

Proof. If $w(A) \geq 3$ then we can use three tiles. By Corollary 1, the $>$-column can be well 2-partitioned, so we can use one tile to cover the $<$-column.

Now it suffices to prove the lemma for $w(A)$ equal to 2.5. Let the weight of the $>$-column be equal to $2 + \epsilon$. Using Lemma 4 we can cover this column with two tiles of weight not exceeding $(2+\epsilon)/2+1/2$. Then we can extend the tiles to the $<$-column. Since the total weight of this column is equal to $1/2 - \epsilon$, none of the resulting tiles will have weight exceeding $(2 + \epsilon)/2 + 1/2 + 1/2 - \epsilon = 2 - \epsilon/2 \leq 2$. □

Putting Lemma 3 and Lemma 5 together we conclude that any complex (i.e. two column subarray that cannot be well 2-partitioned) is of type $\diamond 2$ and the total weight between 2.5 and 3. But not all such subarrays are complexes. First note that if the $>$-column has weight equal to 2 then it can be 2-partitioned vertically. As for the horizontal partition it is not hard to observe that the subarray $\diamond 2$ cannot be well 2-partitioned if and only if there exists a row b, that cannot be covered by one tile neither with the part lying above it nor with the part lying below it. Thus the weight of b together with the weight of each of these two parts is greater than 2. Henceforth, referring to such a row b we shall use the term *boundary*.

The following simple properties of the complex will be frequently used:

(a) the weights of the parts above and under the boundary are less than 1,

(b) the weight of the boundary is greater than 1.

Since the boundary together with any part has weight greater than 2, the other part must have weight less than 1, so we get (a). The property (b) is a straightforward consequence of (a).

Sometimes we will also need to partition subarrays of the type $\diamond 2$ into 2 tiles regardless their weight and therefore we will also consider *general complexes*, that is subarrays of the type $\diamond 2$ that cannot be 2-partitioned into 2 tiles. Everything that was said about complexes except for the properties (a) and (b) holds also for general complexes, in particular the existence of the boundary.

3.3 An f-approximation algorithm: the outline

In order to get the approximation ratio better than $2\frac{1}{3}$ we need to consider subarrays consisting of more than two columns. Let *a block* denote a two-column subarray consisting of a $<$-column and a $>$-column. We will process the array A block by block, starting from the leftmost. If the block currently processed can be well-partitioned with a satisfactory factor, we partition it and proceed to the next one. Else we try well-partitioning the subarray consisting of this block and the neighbouring one. If the factor is still unsatisfactory, we add one block more and so on. This process naturally has to end, when there are no more blocks left i.e. we have reached the end of the array or there is one $<$-column left. Then, however, we are allowed to nearly-well-f-partition what is left and nearly-well-partitioning can usually be done with a better factor than well partitioning. Aiming at f- approximation the algorithm can be described as follows:

```
while possible
    subarray S := the first block from the left
    if S can be well-f-partitioned, partition it
    else
        while S cannot be well-f-partitioned and extension possible
            increase S by the next block
        if S can be well-f-partitioned, partition it
        else nearly-well-f−partition S together with what is left
nearly-well-f-partition the remaining (which must be a <-column)
```

It will turn out that to get a $2\frac{1}{4}$- approximation, it is enough to consider subarrays consisting of at most six columns, but in order to get a $2\frac{1}{8}$-approximation we need to consider arbitrarily long subarrays.

3.4 How to establish the factor with which we can well partition a subarray: the method of equaling maximums

As we already know any subarray of the type $\diamond k$ can be well 2-partitioned unless it is a complex. A complex, in turn, can be always well $\frac{7}{3}$-partitioned. However if a complex is followed by a subarray of the type, say, $\diamond 3$ and their total weight is less than 6, then it could turn out more advantegous to partition a complex into 3 tiles and the rest into 2 tiles. What happens in such a situation is that on the one hand there is a tiling in which the heaviest tile (and the only one, whose weight exceeds 2) lies in the complex and on the other we can partition the subarray so that the tile of weight exceeding 2 lies outside the complex. Naturally we can compare these tilings and choose the better. One can see that it would be quite convenient to be able to say what weight the heaviest tile has in the worst case. Below we give a very general description of the way to compare tilings. What it looks like in practice can be seen in the subsequent facts in the following section.

The simple fact that lies at the core of the method is the following

$$max\{min\{S_1, S_2, ..., S_n\} : S_1, S_2, ..., S_n \geq 0, S_1 + S_2 + ... + S_n = S\} = \frac{S}{n}$$

The key property here is that $min\{S_1, S_2, ..., S_n\}$ is greatest when all S_i are equal. A slight generalization of this is

$$max\{min\{f_1(S_1), f_2(S_2), ..., f_n(S_n)\} : S_1, S_2, ..., S_n \geq 0, S_1+S_2+...+S_n = S\} \leq f_1(m_1),$$

where m_1 is such that $f_1(m_1) = f_2(m_2) = ... = f_n(m_n)$ and $m_1 + m_2 + \ldots + m_n = S$ and functions f are increasing and linear, that is have the form $f_i(S_i) = a_i S_i + b_i$ ($a_i > 0$). The existence of $(m_1, m_2, \ldots, m_n)$ follows from the fact that all functions f_i are increasing.

For example if we consider the case of $n = 2$:

$$M = max\{min\{a_1 S_1 + b_1, a_2 S_2 + b_2\} : S_1, S_2 \geq 0, S_1 + S_2 = S\},$$

then

$$M \leq a_1 m + b_1,$$

where m is such that $a_1 m + b_1 = a_2(S - m) + b_2$.

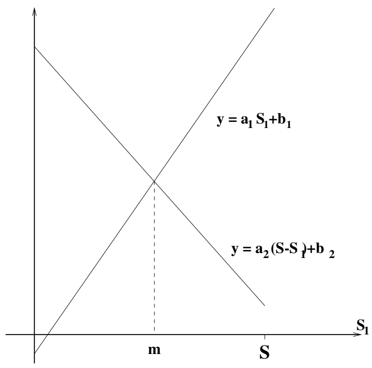

Figure 3.2:

Let us now suppose that we have a subarray B of a certain type with weight S, that is to be partitioned into a certain number of tiles. Then we can benefit from the above observations in the situation when we are able to point out two different tilings T_1 and T_2, such that only one tile (*the critical* one) in each of them has weight exceeding 2 and their weights can be upperbounded by the expressions of the form $a_1 S_1 + b_1$, $a_2 S_2 + b_2$, where S_1 and S_2 denote the weights of the parts of B in which these critical tiles are located.

The same procedure extends easily to any number of tilings over the same subarray.

As far as tilings with critical tiles having no empty common part are considered, the following observation is to prop us up.

Fact 3 *Let us assume that we have two tilings of an array A such that their critical tiles C_1 and C_2 intersect in B. Let a, b, c be nonnegative numbers such that $b = w(B)$, $a = w(C_1 \setminus B)$ and $c = w(C_2 \setminus B)$. If $b \leq v$ then either C_1 or C_2 have weight not greater than $\frac{S+v}{2}$, where $S = w(A)$.*

Proof. We have to estimate $\min\{w(C_1), w(C_2)\}$ for the worst choice of weigths a, b, c. Let $d = w(S \setminus (C_1 \cup C_2))$.

$$max\{min\{a+b, b+c\} : a+b+c+d = S;\ 0 \leq b \leq v, a, c;\ d \geq 0\} =$$

$$max\{min\{a, c\} + b : a+b+c+d = S;\ 0 \leq b \leq v;\ a, c, d \geq 0\} \leq$$

$$max\{\frac{a+c}{2} + b : a+b+c+d = S;\ 0 \leq b \leq v;\ a, c, d \geq 0\} =$$

$$max\{\frac{S-b-d}{2} + b : 0 \leq b \leq v;\ d \geq 0\} = \frac{S+v}{2}.$$

□

In other words, the critical tile in a better tiling is the heaviest under the specified circumstances when the weight of the subarray is concentrated in these tiles, the common part has weight as large as possible and the critical tiles in both tilings have equal weights.

3.5 A $2\frac{1}{4}$-approximation

In this section we examine subarrays beginning with complexes and using the method of equaling maximums gradually gather facts showing that $2\frac{1}{4}$-approximation is attainable.

Fact 4 *If we have a complex $\diamond 2$ such that the weight of the $<$-column is R and the weight of the $>$-column is $2 + \epsilon$, then we can partition it horizontally into 2 tiles so that the weight of the heavier tile is at most $\frac{3}{2} + \frac{\epsilon}{2} + R$.*

Proof. It is sensible to consider the following tilings only: one in which the boundary is included in the upper tile and one in which the boundary is included in the lower tile. The boundary is the common part of the two critical tiles and its weight cannot exceed $1 + R$. Now taking advantage of Fact 3 it is clear, that the weight of the heavier tile in one of these tilings will not surpass $\frac{3}{2} + \frac{\epsilon}{2} + R$. □

Fact 5 *In the subarray $\diamond 2 \diamond$ with two general complexes built-in the boundaries are on the same level. Otherwise we can well 2-partition it.*

Proof. Let us conversely assume that we have a subarray $\diamond 2 \diamond$ with two general complexes built-in and the boundaries are not on the same level. Then the weight of the whole subarray is greater than 4 and we can 2-partition it into 4 tiles. □

In the following we shall formulate several statements about upper bounds on the factors within which one can partition different types of arrays. In these statements we shall use the following abbreviation

$$WM(\diamond k_1 \diamond \ldots \diamond k_l, S \in [p, q), n) \leq f$$

to say that any subarray of the type $\diamond k_1 \diamond \ldots \diamond k_l$ having weight $\sum_{i=1}^{l} k_i + S$ can be f-partitioned into n tiles.

Fact 6

$$WM(\diamond 2 \diamond, S \in [0, 1), 3) \leq 2$$

Proof. If the subarray $\diamond 2 \diamond$ has no complexes or has only one complex, then it clearly can be 2-partitioned into 3 tiles. Let us now consider the case when $\diamond 2 \diamond$ has 2 complexes built-in. We are allowed to say complexes and not general complexes, because any subarray of the type $\diamond 2$ here has weight less than 3. Then by Fact 5 the boundaries are on the same level and since $S \in [0, 1)$, then the row containing the boundaries has weight less than 2, which together with the fact that the parts of the complex under and over the boundary have weights less than 1 means, that the whole array can be 2-partitioned horizontally into 3 tiles (with the row containing the boundaries forming the middle tile). □

The forthcoming facts concern types whose first two columns are $\diamond 2$. In the proofs we shall assume that these columns form a complex. If it were not so, then most of these types could be well 2-partitioned.

Fact 7 *For any integer $k \geq 2$*

$$WM(\diamond 2 \diamond k,\ S \in [0.5, 1),\ 2 + k) \ \leq\ 2 + \frac{2}{2k+1}(S - \frac{1}{2})$$

Proof.

First, we should explain that if $k = 2$, then the subarray does not contain two disjoint complexes, because then the weight of the whole subarray would be greater than $2 \cdot 2.5$, which would contradict the assumption that $S < 1$.

Let us consider three different tilings of the type $\diamond 2 \diamond k$ (see Fig. 3.3):

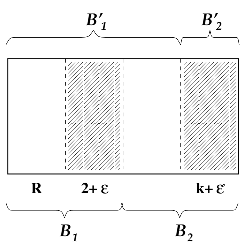

Figure 3.3:

Tiling 1 Partition complex B_1 horizontally into 2 tiles and B_2 into k tiles.
The critical tile:
According to Lemma 3 B_2 can be well 2-partitioned into k parts, so the critical tile lies in B_1 and its weight is by Fact 4 at most $\frac{3}{2} + \frac{\epsilon}{2} + R$.

Tiling 2 Partition complex B_1 vertically into 2 tiles and B_2 into k tiles.
The critical tile:
It lies in B_1 and its weight is equal $2 + \epsilon$.

Tiling 3 Partition B'_1 into 3 tiles and B'_2 into $k - 1$ tiles.
The critical tile:
Using Fact 6 we know that B'_1 can be 2-partitioned into 3 tiles, so the crtitical tile lies in B'_2 and its weight is according to Lemma 4 at most $\frac{k+\epsilon'}{k-1} + \frac{k-2}{k-1} = 2 + \frac{\epsilon'}{k-1}$.

We want to calculate

$$max\{min\{\frac{3}{2} + \frac{\epsilon}{2} + R, 2 + \epsilon, 2 + \frac{\epsilon'}{k-1}\} : \epsilon, \epsilon', R \geq 0, \epsilon + \epsilon' + R \leq S\}.$$

Here we cannot directly take advantage of the facts described in Section 4, as the weight of some critical tiles is the function of two variables.
However, we notice that if $S_1 = \frac{\epsilon}{2} + R$, $S_2 = \epsilon/2$ and $S_3 = \epsilon'$, the above is equivalent to calculating

$$max\{min\{\frac{3}{2} + S_1, 2 + 2S_2, 2 + \frac{S_3}{k-1}\} : S_1, S_2, S_3 \geq 0, S_1 + S_2 + S_3 \leq S\}.$$

Thus $\{min\{\frac{3}{2} + \frac{\epsilon}{2} + R, 2 + \epsilon, 2 + \frac{\epsilon'}{k-1}\}$ is not greater than when the weights of all critical tiles are equal. Equaling them we get

$$R = \frac{1+\epsilon}{2} \qquad \text{and} \qquad \epsilon' = (k-1)\epsilon.$$

Since $\epsilon + \epsilon' + R \leq S$, we have $\epsilon \leq \frac{2}{2k+1}(S - \frac{1}{2})$, which implies the thesis. □

We use the procedure similar to the one described above to estimate the factors of the following types. The general scheme is as follows. Associate a variable denoting weight to each of the columns in the type, find a number of tilings, whose critical tiles have weights expressed in terms of these variables and by using facts from Section 4 arrive at the upper bound on the minimal weight of the critical tiles.

Fact 8 *For every integer $k \geq 2$*

$$WM(\diamond 2 \diamond k, S \in [1, 2), k + 3) \leq 2 + \frac{S-1}{k+1}$$

Proof.
We can assume that B'_1 consists of two general complexes C_1 and C_2 (see Fig. 3.4), because otherwise it could be well 2-partitioned into 3 tiles and then the whole subarray could be well 2-partitioned into $k + 3$ parts.

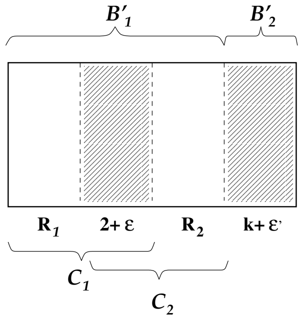

Figure 3.4:

Tiling 1 Partition complex C_1 horizontally into 2 tiles and the rest of the array into $k+1$ parts.

The weight of the critical tile: $\frac{3}{2}+\frac{\epsilon}{2}+R_1$

Tiling 2 Partition complex C_2 horizontally into 2 tiles and the rest into $k+1$ parts.

The weight of the critical tile: $\frac{3}{2}+\frac{\epsilon}{2}+R_2$

Tiling 3 Partition B_1' into 4 tiles and B_2' into $k-1$ tiles.

The weight of the critical tile: $2+\frac{\epsilon'}{k-1}$

Since

$$max\{min\{\frac{3}{2}+\frac{\epsilon}{2}+R_1, \frac{3}{2}+\frac{\epsilon}{2}+R_2, 2+\frac{\epsilon'}{k-1}\} : \epsilon, \epsilon', R_1, R_2 \geq 0, \epsilon+\epsilon'+R_1+R_2 \leq S\},$$

is equivalent to

$$max\{min\{\frac{3}{2}+S_1, \frac{3}{2}+S_2, 2+\frac{S_3}{k-1}\} : S_1, S_2, S_3 \geq 0, S_1+S_2+S_3 \leq S\}$$

if $S_1 = \frac{\epsilon}{2}+R_1$, $S_2 = \frac{\epsilon}{2}+R_2$ and $S_3 = \epsilon'$,

we get that at the worst the minmal weight of the three critical tiles is not greater than when all of them are equal.

Comparing appropriate tilings we obtain:

$S_1 = S_2$, and $S_3 = (k-1)(S_1 - \frac{1}{2})$.

Adding S_1, S_2, S_3 up and comparing to S, we get

$$S_1 = \frac{S}{k+1} + \frac{k-1}{2(k+1)}$$

Hence $\frac{3}{2}+S_1 = \frac{3}{2}+\frac{2S+k-1}{2(k+1)} = 2+\frac{S-1}{k+1}$. □

By the above fact the subarray of the type $\diamond 2 \diamond 2$ having weight from the interval $[5, 6)$ can be well $2\frac{1}{3}$-partitioned. The factor $2\frac{1}{3}$ is clearly not satisfactory and that is why we have to examine this case more thorougly.

Fact 9 $WM(\diamond 2 \diamond 2, S \in [1,2), 5) \leq 2 + \frac{4S-5}{13}$

Proof.

At the beginning we can assume that there are 3 general complexes C_1, C_2 and C_3 (see Figure 3.5) built-in in this case, because otherwise we could well 2-partition the subarray. By Fact 5 the boundaries of complexes C_1 and C_3 are in the same row. If the boundary of C_2 also lies in the same row, then we can 2-partition the whole subarray into 4 tiles in the way shown in Figure 3.5(a). Neither of the tiles under and over the boundaries can have weight greater than 2, because it would mean that the whole subarray $\diamond 2 \diamond 2$ has weight greater than 6.

If the boundary of C_3 is not in the same row, then we consider the following tilings:

Tiling 1 Partition C_1 vertically into 2 tiles and C_2 into 3 tiles.
The critical tile:
It covers the $>$-column of C_1.

Tilings 2 and 3 Partition C_1 horizontally into 2 tiles and C_2 into 3 tiles.
They are formed by the boundary and respectively the part above and under it.

Tiling 4 Partition C_3 horizontally into 2 tiles and the rest into 3 tiles.
The critical tile:
It is formed by the boundary and the part below it.

Tiling 5 (Analogous to Tiling1) Partition C_2 vericaly into 2 tiles and C_1 into 3 tiles.
The critical tile:
It is the $>$-column of C_2.

Tilings 6 and 7 (Analogous to Tilings 2 and 3) Partition C_2 horizontally into 2 tiles and C_1 into 3 tiles.
They are formed by the boundary and respectively the part above and under it.

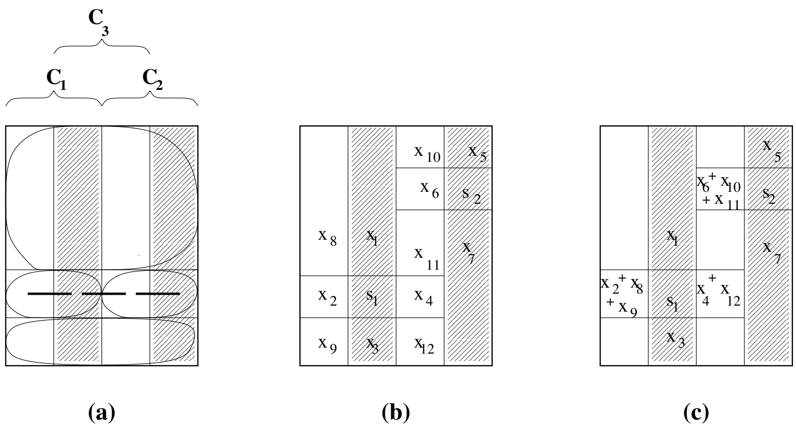

Figure 3.5:

First, let us notice that we can resrict ourselves to the situation, when the weight of $<$-columns is concentrated in the boundaries (i.e. array-elements of $<$-columns outside the boundaries are equal to 0). In Figure 3.5(c) the weight of every critical tile is bigger or equal to the weight of its counterpart in Figure 3.5(b).

Second, we can also assume that s_1 and s_2 have the greatest weight possible, that is, are equal to 1. To see this, observe that by replacing s_1 and s_2 with 1 and by subtracting $\frac{1-s_1}{3}$ from x_1, x_2 and x_3 and $\frac{1-s_2}{3}$ from x_5, x_6 and x_7 we can only increase the weight of the critical tiles while not changing the overall weight of the subarray.

Thus, we are interested in finding $max\{min\{1 + x_1 + x_2, 1 + x_2 + x_3, 1 + x_1 + x_3, 1 + x_3 + x_4, 1 + x_5 + x_6, 1 + x_4 + x_6 + x_7, 1 + x_5 + x_7\} :$ $x_1, x_2, \ldots, x_7 \geq 0, x_1 + x_2 + \ldots + x_7 + 2 = 4 + S\}$. If we make the substitution $S_1 := \frac{3}{4}(x_1+x_2)$, $S_2 := \frac{1}{4}(x_1+x_3)$, $S_3 := \frac{1}{4}(x_2+x_3)$, $S_4 := \frac{1}{2}(x_3+x_4)$, $S_5 := \frac{1}{2}(x_4+x_6+x_7)$, $S_6 := \frac{1}{2}(x_5+x_7)$, $S_7 := \frac{1}{2}(x_5+x_6)$, we can write the above expression equivalently as $max\{min\{1+4S_1, 1+4/3S_2, 1+4S_3, 1+2S_4, 1+2S_5, 1+2S_6, 1+2S_7\} :$ $S_1, S_2, \ldots, S_7 \geq 0, S_1 + S_2 + \ldots + S_7 + 2 = 4 + S\}$. Therefore, the weight of the critical tile is not bigger than $1 + 4S_1$, where $13S_1 + 2 = 4 + S$. $\square$

There is no need to consider $WM(\diamond 2 \diamond k, S, k + 4)$, for $S \geq 2$ because more generally any subarray of the type $\diamond k_1 \diamond k_2 \cdots \diamond k_l$ can be 2-partitioned into $\sum_{i=1}^{l} k_i + l$ tiles by separately well 2-partitioning $>$-columns and forming l rectangles out of $<$-columns.

In the subsequent proofs of Facts 11-14, we arrive at the upper bound of the minimal weight of the critical tiles by equaling their weights. We are justified to do so as in every one of these cases we can utilize facts from Section by expressing the problem equivalently in the form $max\{min\{f_1(S_1), f_2(S_2), \ldots, f_n(S_n)\} : S_1, S_2, \ldots, S_n \geq 0, S_1 + S_2 + \ldots S_n = S\}$, where $f_i(S_i) = a_i S_i + b_i$ $(a_i > 0)$.

Fact 10 $WM(\diamond 2 \diamond 1, S \in [1, 2), 4) \leq 2 + \frac{2}{5}(S - \frac{3}{2})$

Proof.

We can assume that there are two built-in general complexes C_1 and C_2 as shown in Figure 3.6, because if it were not the case then we would be able to well 2-partition the whole subarray.

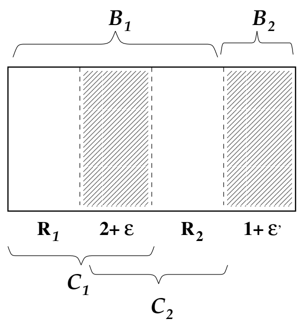

Figure 3.6:

Tiling 1 Partition C_1 horizontally into 2 tiles and the rest of the array also into 2 parts.

The weight of the critical tile: $\frac{3}{2} + \frac{\epsilon}{2} + R_1$ (respectively $\frac{3}{2} + \frac{\epsilon}{2} + R_2$)

Tiling 2 Partition B_1 vertically into 3 tiles and B_2 into 1 tile.

The weight of the critical tile: $2 + \epsilon$

Tiling 3 Partition C_1 into 3 tiles and the rest of the array into 1 tile.

The critical tile:
Since C_1 can be 2-partitioned into 3 parts, the critical tile is simply the tile that covers two last columns. Its weight is equal to $R_2 + 1 + \epsilon'$.

Comparing appropriate tilings we obtain

- $R_1 = \frac{1}{2} + \frac{\epsilon}{2}$ (by comparing tilings 1 and 3),
- $\epsilon' + R_2 = 1 + \epsilon$ (by comparing tilings 3 and 4).

Adding R_1, ϵ and ϵ' up, we get

$$\frac{1}{2} + \frac{\epsilon}{2} + \epsilon + 1 + \epsilon = S.$$

Hence $\epsilon = \frac{2}{5}(S - \frac{3}{2})$. □

If the complex is followed by the subarray $\diamond 1$ and their total weight does not exceed 4, then it may be difficult to show that it can be well partitioned with the factor better than $7/3$. We can omit this problem and consider instead the types $\diamond 2 \diamond 1 \diamond k$ beginning with the subarray $\diamond 2 \diamond 1$ having weight less than 4. For reasons of conveniency we estimate the factors of the type $\diamond 2 \diamond 1 \diamond k$ separately for $k \geq 2$ and $k = 1$.

In the following Facts 11-14 we assume that the beginning subarrays of the type $\diamond 2 \diamond 1$ have weight less than 4.

Fact 11 *For every integer $k \geq 2$*

$$WM(\diamond 2 \diamond 1 \diamond k, S \in [0.5, 1), k + 3) \leq 2 + \frac{2}{2k + 1}(S - \frac{1}{2})$$

Proof. Let us consider three tilings (see Fig. 3.7):

Tiling 1 Partition B_1 horizontally into 2 tiles, B_2 into 1 tile and B_3 into k parts.

The critical tile:
Because $S < 1$, then both B_2 and B_3 can be well 2-partitioned, so the weight of the critical tile is at most $\frac{3}{2} + \frac{\epsilon}{2} + R$.

Tiling 2 Partition B_1 vertically into 2 tiles, B_2 into 1 tile and B_3 into k tiles.

The weight of the critical tile: $2 + \epsilon$

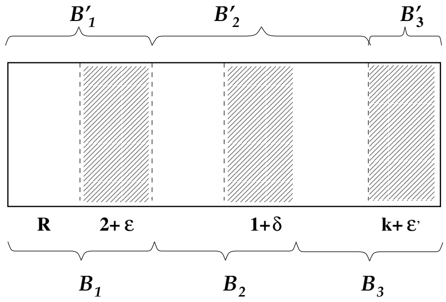

Figure 3.7:

Tiling 3 Partition B_1' into 3 tiles, B_2' into 1 tile and B_3' into $k-1$ tiles.

The critical tile:
Because $S < 1$, then B_2' will have weight less than 2 and the critical tile lies in B_3'. According to Lemma 4 its weight will not surpass $\frac{k+\epsilon'}{k-1} + \frac{k-2}{k-1} = 2 + \frac{\epsilon'}{k-1}$.

Comparing appropriate tilings we obtain

- $R = \frac{1}{2} + \frac{\epsilon}{2}$ (by comparing tilings 1 and 2),
- $\epsilon' = (k-1)\epsilon$ (by comaring tilings 2 and 3).

Since $R + \epsilon + \epsilon' \leq S$, we get

$$\frac{1}{2} + \frac{\epsilon}{2} + \epsilon + (k-1)\epsilon \leq S.$$

Hence $\epsilon \leq \frac{2}{2k+1}(S - \frac{1}{2})$. □

Fact 12 *For every integer $k \geq 2$*

$$WM(\diamond 2 \diamond 1 \diamond k, S \in [1,2), k+4) \leq 2 + \frac{2}{2k+3}(S - \frac{3}{2})$$

Proof. Let us assume notation as in Figure 3.8. We consider the following tilings:

Tiling 1 Partition B_1 horizontally into 2 tiles, B_2 into 1 tile, B_3 into 1 tile and B_4 into k parts.

The weight of the critical tile: $\frac{3}{2} + \frac{\epsilon}{2} + R$.

Tiling 2 Partition B_1 vertically into 2 tiles, B_2 into 1 tile, B_3 into 1 tile and B_4 into k tiles.

The weight of the critical tile: $2 + \epsilon$

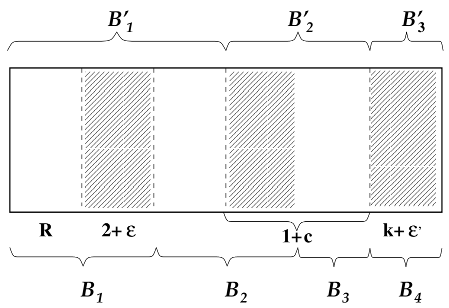

Figure 3.8:

Tiling 3 Partition B_1' into 3 tiles, B_2' into 2 tile and B_3' into $k-1$ tiles.

The critical tile:
By Fact 6, B_1' can be 2-partitioned into 3 parts, so the critical tile lies in B_3' and according to Lemma 4 its weight will not surpass $\frac{k+\epsilon'}{k-1} + \frac{k-2}{k-1} = 2 + \frac{\epsilon'}{k-1}$.

Tiling 4 Partition B_1' into 3 tiles, B_2' into 1 tile and B_3' into k tiles.

The weight of the critical tile: $1 + c$

Comparing appropriate tilings we obtain

- $R = \frac{1}{2} + \frac{\epsilon}{2}$ (by comparing tilings 1 and 2)
- $\epsilon' = (k-1)\epsilon$ (by comparing tilings 2 and 3)
- $c = 1 + \epsilon$(by comparing tilings 2 and 4)

Since $R + \epsilon + c + \epsilon' \leq S$, we get

$$\frac{1}{2} + \frac{\epsilon}{2} + \epsilon + 1 + \epsilon + (k-1)\epsilon \leq S$$

Hence $\epsilon \leq \frac{2}{2k+3}(S - \frac{3}{2})$. □

Fact 13 $WM(\diamond 2 \diamond 1 \diamond 1, S \in [0.5, 1), 4) \leq 2 + \frac{2}{5}(S - \frac{1}{2})$

Proof. Let us assume notation as in Figure 3.10. We consider the following tilings:

Tiling 1 Partition B_1 horizontally into 2 tiles, B_2 into 1 tile and B_3 into 1 part.

The weight of the critical tile: $\frac{3}{2} + \frac{\epsilon}{2} + R$.

Tiling 2 Partition B_1 vertically into 2 tiles, B_2 into 1 tile and B_3 into 1 tile.

The weight of the critical tile: $2 + \epsilon$

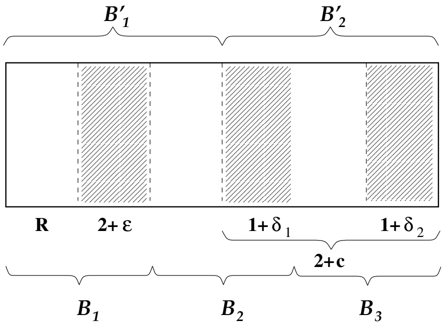

Figure 3.9:

Tiling 3 Partition B_1' into 3 tiles and B_2' into 1 tile.

The critical tile:
By Fact 6, B_1' can be 2-partitioned into 3 parts, so the critical tile is formed by B_2' and its weight is equal $2 + c$.

Comparing appropriate tilings we obtain

- $R = \frac{1}{2} + \frac{\epsilon}{2}$ (by comparing tilings 1 and 2),
- $c = \epsilon$ (by comparing tilings 2 and 3)

Since $R + \epsilon + c \leq S$, we get

$$\frac{1}{2} + \frac{\epsilon}{2} + 2\epsilon \leq S$$

Hence $\epsilon \leq \frac{2}{5}(S - \frac{1}{2})$. □

Fact 14 $WM(\diamond 2 \diamond 1 \diamond 1, S \in [1, 2), 5) \leq 2 + \frac{2}{5}(S - \frac{3}{2})$

Proof. Let us assume notation as in Figure **??**. We consider the following tilings:

Tiling 1 Partition B_1 horizontally into 2 tiles, B_2 into 1 tile and B_3 into 2 parts.
The weight of the critical tile: $\frac{3}{2} + \frac{\epsilon}{2} + R$.

Tiling 2 Partition B_1 vertically into 2 tiles, B_2 into 1 tile and B_3 into 2 tile.
The weight of the critical tile: $2 + \epsilon$

Tiling 3 Partition B_1' into 3 tiles, B_2' into 1 tile and B_3' into 1 part.
The weight of the critical tile: $1 + c$

Comparing appropriate tilings we obtain

- $R = \frac{1}{2} + \frac{\epsilon}{2}$ (by comparing tilings 1 and 2),

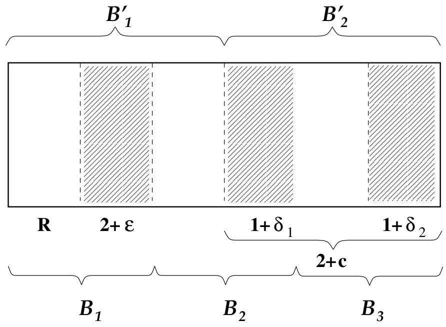

Figure 3.10:

- $c = 1 + \epsilon$ (by comparing tilings 2 and 3).

Since $R + \epsilon + c \leq S$, we get

$$\frac{1}{2} + \frac{\epsilon}{2} + 1 + 2\epsilon \leq S$$

Hence $\epsilon \leq \frac{2}{5}(S - \frac{3}{2})$. □

Fact 15 *Any subarray of the type* $\diamond 2 \diamond 1 \diamond$ *with the beginning subarray* $\diamond 2 \diamond 1$ *having weight less than* 4 *can be nearly well* 2*-partitioned.*

Proof. If the weight of the whole subarray is less than 4, then it can be 2-partitioned into 4 tiles, by partitioning the complex into 3 tiles and the rest into 1 tile. Otherwise, if the weight of the subarray is at least 4, then it can be 2-partitioned into 5 tiles by partitioning the part $\diamond 2 \diamond 1$ into 4 tiles and the last $<$-column into 1 tile. □

Using the facts above we are able to state that they suffice to construct a 9/4-approximation algorithm.

Theorem 1 *There exists a linear-time* 9/4*-approximation algorithm for the RTILE problem.*

Proof. Recall that we are allowed to treat any array as if it consisted of alternating $>$- and $<$-columns and began with a $<$-column.

We proceed from the left to the right and examine subsequent subarrays of the type $\diamond k$ combining them if necessary. If $\diamond k$ is not a complex, then it will be treated separately. According to Lemma 3 and Lemma 5 it can be well 2-partitioned. If $\diamond k$ is a complex, then we look at the neighbouring subarray $\diamond k'$. If $k' \geq 2$, then by Facts 7-9 the type $\diamond 2 \diamond k'$ can be well partitioned with the ratio not greater than 9/4. If the complex is followed by a subarray of the type $\diamond 1$, then we look at the weight of the type $\diamond 2 \diamond 1$ and if it is at least 4, then according to Fact 10, it can be well 11/5-partitioned. Otherwise we are forced to consider one more subarray of the type $\diamond k$. Then using

Facts 11-14 we see that we are able to well partition such a type ($\diamond 2 \diamond 1 \diamond k$ with the beginning subarray $\diamond 2 \diamond 1$ having weight less than 4) with the factor not greater than $11/5$. What remains to be explained are the situations when the subarrays of the type $\diamond 2$ (a complex) and $\diamond 2 \diamond 1$ having weight less than 4 are not followed by a subarray of the type $\diamond k$ (that is they are at the right end of the array). However then by the Facts 6 and 15 they can be nearly well 2-partitioned. Therefore by the Fact 2 we can indeed well $9/4$-partition any array. $\square$

Chapter 4

A $2\frac{1}{8}$-approximation algorithm

If we want to further improve the approximation ratio, we will need to consider subarrays consisting of more than six columns. In fact in order to get a $2\frac{1}{8}$-approximation, we will have to consider subarrays consisting of arbitrarily many columns. In handling such large subarrays we are not going to examine them globally, which might prove a discouraging task as well as result in non-polynomial running time. Rather, we will process them locally, focusing attention on small subarrays i.e. subarrays consisting of a small number of columns. Any time it proves to be impossible to well-f-partition a subarray by the appropriate partitioning of its small subarrays, it will turn out that its weight must be greater than some convenient value. For example, if we could prove that whenever we are unable to well-f-partition a subarray of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$, its weight is at least $\sum_{i=1}^{n} k_i + n - 1$, then we would at least know that it can be nearly-2-partitioned, as we can cover every k_i column by k_i tiles and every $<$-column by one tile. The situation is of course somewhat more complicated. To be able to control it we extend the classification into types.

4.1 Extended classification into types

If we have a block of type $\diamond m$, then we know, its weight is at least m. We also know that it can always be well-2-partitioned if it is not a complex and a complex can be well-$2\frac{1}{3}$-partitioned. Next, we are going to examine when it can be f-partitioned into m tiles.

Definition 3 *A block of type $\_\, m$ denotes a block consisting of a $<$-column and a m-column that can be f-partitioned into m tiles. A block of type $\sqcup m$ denotes a block consisting of two columns as above that cannot be f-partitioned into m tiles.*

Based on this, we describe the type of the whole array by giving the type of every block of type $\diamond m$ contained in it. If the array's last column is a $<$-column, we denote it by $\_$. Examples of types of the array: $\sqcup 2 \sqcup 2 \sqcup 3$ or $\sqcup 2 \_ 3 \_$. Notice, that the type of the array does not contain information about the types of the blocks, in which a $<$-column is on the right of a $>$-column. For example, if the type of the array is $\sqcup 2 \sqcup 3$, we do not know whether the second and a third column are a block of type $2\sqcup$ or 2_{-}.

The subarrays we will consider, will consist of columns of the array. The type of the subarray is given in the same way, except for two things. The subarray may begin with a $>$-column, then we first give the type of this column. If the subarray ends with a $<$-column, then we denote it by $\sqcup$, if it is the part of a block of type $\sqcup m$ in the whole array.

$\diamond$ denotes $\sqcup$ or $_{-}$.

4.2 Blocks

4.2.1 A complex

Definition 4 *A complex, denoted by* $\sqcup 2$ *is a two-column subarray consisting of a* $<$*-column and a* 2*-column, that cannot be* $2\frac{1}{8}$*-partitioned into two rectangles.*

A complex cannot be $2\frac{1}{8}$-partitioned into two rectangles if and only if

1. the vertical partition into two rectangles is impossible, which implies that the column of type 2 has weight greater than $2\frac{1}{8}$
2. the horizontal partition is impossible, which means, that there exists such a row b, that both the weight of b together with the part of the subarray above it is greater than $2\frac{1}{8}$ and so is the weight of b together with the part below it.

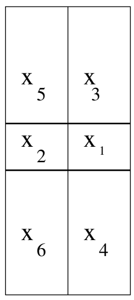

Figure 4.1:

The minimal weight of the complex can be estimated using linear programming. Consider Figure 4.1 and the following linear program:

$$\begin{array}{lrl}
\text{minimize} & \sum_{i=1}^{6} x_i & \\
\text{subject to} & x_2 + x_5 + x_6 & < 1 \\
 & x_1 & \leq 1 \\
 & x_1 + x_3 + x_4 & \geq 2\frac{1}{8} \\
 & x_1 + x_2 + x_3 + x_5 & \geq 2\frac{1}{8} \\
 & x_1 + x_2 + x_4 + x_6 & \geq 2\frac{1}{8}
\end{array}$$

We get that the weight of the above complex is minimal when $x_1 = 1, x_2 = x_3 = x_4 = \frac{9}{16}$ and $x_5 = x_6 = 0$, and is equal to $2\frac{11}{16}$. Therefore, we have

Fact 16 *The weight of the complex is greater than* $2\frac{11}{16}$.

One can notice that in the description of the complex we might use only four variables by assuming x_5 and x_6 equal to 0, because we can make the following observation.

Observation 1 *Assume we have two complexes: one as in Figure 4.1 and the other with the variables related to the variables of the first one as follows:* $x'_1 = x_1,\ x'_4 = x_4$, $x'_3 = x_3$, $x'_5 = x'_6 = 0$ *and* $x'_2 = x_2 + max\{x_5, x_6\}$. *Then the weight of the second complex is not bigger than the weight of the first one while the inequalities describing the first complex remain true for the second.*

The row b in the complex, which prevents its horizontal $2\frac{1}{8}$-partition, will be further on referred to as a $b$oundary. It is the term, which will also occur in connection with subarrays of a few other types.

Definition 5 *The boundary stays for that row in the subarray that prevents its horizontal* $2\frac{1}{8}$*-partition into* 2 *tiles. It will occur in the subarrays of type* $\sqcup 2$, $2\sqcup$, 3 *and* $\sqcup 1 \sqcup$.

4.2.2 A non-complex

For blocks of type $\sqcup m$ we have the following lemma:

Lemma 6 *If a subarray consisting of a* $<$*-column and an* m*-column cannot be* $2\frac{1}{8}$*-partitioned into m tiles, then its weight is greater than, correspondingly:* $m + 1 + \frac{m-1}{8}$, *if* $m \geq 3$ *and greater than* $2\frac{1}{8}$, *if* $m = 1$.

The proof is deferred to the last section of the chapter.

Let us observe that every block of type $\diamond m$ can always be 2-partitioned into $m + 1$ tiles, since every m-column can be well-2-partitioned ([7]), i.e. 2-partitioned into m tiles. Therefore the above lemma implies that all blocks, except for blocks of type $\sqcup 2$, can be well-$2\frac{1}{8}$-partitioned.

4.3 Definitions

We assign every type T a certain natural number $N(T)$ in the way described below.

Definition 6 $N(T)$ *is defined as follows. For types of columns:*

$$(\_\,) = -1$$
$$N(\sqcup) = 0$$
$$N(m) = m - 1$$

For more complex types:

$$N(T_1 T_2) = N(T_1) + N(T_2) + 1.$$

For almost all types $N(T)$ is defined in such a way that it is the smallest number n, for which we are guaranteed from the definition of types of blocks, that every subarray of type T can be $2\frac{1}{8}$-partitioned into $n+1$ tiles. For example $N(\_\, m) = m - 1$ and $N(\sqcup m) = m$ and indeed we know that a block of type $\_\, m$ can be $2\frac{1}{8}$-partitioned into $N(\_\, m) + 1 = m$ tiles and a block of type $\sqcup m$ can be 2-partitioned (hence also $2\frac{1}{8}$-partitioned) into $m+1$ tiles. Also $N(\sqcup 2 \_\, 4) = 7$ and we can $2\frac{1}{8}$-partition every subarray of this type by 2-partitioning the first block into 3 tiles and $2\frac{1}{8}$-partitioning the second one into 4 tiles. The above does not hold for types ended with $\_$. (For example $N(2\_\,) = 1$, however we are not guaranteed that a subarray of this type can be $2\frac{1}{8}$-partitioned into 2 tiles.)

$N(T)$ is also to denote that every subarray of type T, encountered in the course of carrying out the algorithm, will always have weight at least $N(T)$. For some subarrays, it is immediate from Lemma 6 and the definition of types, like every subarray of type $\sqcup 2 \_\, 2$ has weight greater than $2\frac{11}{16} + 2$ (the second 2 comes from the fact that a 2-column has weight at least 2) or every subarray of type $\sqcup 2 \sqcup 2$ has weight greater than $5 + \frac{3}{8} = N(\sqcup 2 \sqcup 2) + \frac{3}{8}$. However, $N(\sqcup 2 \sqcup 2 \sqcup 2 \sqcup 2) = 11$ and from Fact 16, it only implies that every subarray of this type has weight greater than $10\frac{6}{8}$.

The notions of well- and nearly-well-partitioning are related with the actual weight of the subarray. Sometimes, however, it would be convenient to have analogous notions defined with regard to $N(T)$.

Definition 7 *To well-tile a subarray of type T means to $2\frac{1}{8}$-partition it into $N(T)$ tiles. Similarly, to nearly-well-tile a subarray of type T means to $2\frac{1}{8}$-partition it into $N(T) + 1$ tiles.*

We will often take advantage of the following facts:

Fact 17 *If we have a subarray S of type $T = T_1 T_2$, then in order to well-tile S we can well-tile its subarray of type T_1 and nearly-well-tile its subarray of type T_2 or the other way round. Similarly, in order to nearly-well-tile S we can nearly-well-tile both its subarray of type T_1 and its subarray of type T_2.*

Proof. $N(T) = N(T_1) + N(T_2) + 1$ □

Fact 18 *Every subarray of type T, $\diamond T$, $T\sqcup$, $\diamond T \sqcup$, where $T = k_1 \diamond k_2 \ldots \diamond k_n$ can always be nearly-well-tiled.*

If we have a subarray S of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$ and any one of its subarrays of type $\diamond k_i$ can be well-tiled, then by Facts 17 and 18 we can well-tile S, as we can nearly-well-tile the preceding and succeeding part. Similarly, if S is of type $\sqcup k_1 \sqcup k_2 \ldots \sqcup k_n$ and any one of its subarrays of type $k_i \sqcup$ can be well-tiled, then S can be well-tiled. On the other hand if none of the enumerated subarrays can be well-tiled, then the weight of S must surpass some value. For example, if we have a subarray S of type $\sqcup 3 \sqcup$, then if its subarray of type $\sqcup 3$ or of type $3\sqcup$ can be well-tiled, then S can be well-tiled. Otherwise, by Lemma 6, each of them has weight greater than $4\frac{1}{4}$. Also since the weight of a 3-column is from the definition less than 4, we get that S must have weight greater than $2 \cdot 4\frac{1}{4} - 4 = 4\frac{1}{2}$.

We will quite often use this kind of reasoning. In order to make it more clear and concise we introduce

Definition 8 *For a subarray A of type T we will say that*

$$well(A, Z) > w$$

if

1. *if any one of the subarrays from the set Z can be well-tiled, then A can be well-tiled and*

2. *if none of the subarrays from the set Z can be well-tiled, then A has weight greater than $N(T) + w$.*

Similarly we will say that

$$nearly(A) > w$$

if

1. *if any one of the subarrays from the set Z can be well-tiled, then A can be nearly-well-tiled and*

2. *if none of the subarrays from the set Z can be well-tiled, then A has weight greater than $N(T) + w$.*

For example, suppose we have a subarray A of type $\sqcup k_1 \sqcup k_2$ and $k_1, k_2 > 3$. Let Z denote subarrays of A of type $\sqcup k_i$. Then by Lemma 6 and Facts 17,18, we are justified to state $well(A, Z) > 1 + \frac{k_1+k_2-2}{8}$.

Let us also notice, that if for a subarray A of type T and some set Z of its subarrays, it holds $well(A, Z) > w$ and A happens to have weight at most $N(T) + w$, then A can be well-tiled.

Let us notice that if we have a subarray A of type $T = \diamond k_1 \diamond k_2 \ldots \diamond k_n$ having weight at least $N(T)$ and we are able to state that $well(A) > 1$, then we are in a good situation in the sense that A can be well-$2\frac{1}{8}$-partitioned. It is so, because either we are able to well-tile the subarray (i.e $2\frac{1}{8}$-partition it into $N(T)$ tiles) or it has weight at least $N(T) + 1$, which means in turn that for well-partitioning we are allowed to use $N(T) + 1$ tiles and thus able to well-2-partition it (block-wise).

Fact 19 *If a subarray A of type T has weight at least $N(T)$, can be nearly-well-tiled and $well(A) > 1$, then A can be well-$2\frac{1}{8}$-partitioned.*

The knowledge about the subarrays can be appropriately combined.

Lemma 7 *Suppose we have a subarray A of type $T = T_1T_2$ that consists of two subarrays B and C correspondingly of type T_1 and T_2. Suppose also that both B and C*

can be nearly-well-tiled. Let Z_B and Z_C denote some sets of subarrays of respectively B or C. Then

$$well(B, Z_B) > w_1 \text{ and } well(C, Z_C) > w_2 \Rightarrow well(A, Z_B \cup Z_C) > w_1 + w_2 - 1.$$

Suppose now that a subarray A is of type $T = T_1 \_\, T_2$ and consists of three subarrays B, C and D correspondingly of type T_1, $\_$ and T_2.Suppose also that subarrays BC and D can be nearly-well-tiled. Then

$$well(B, Z_B) > w_1 \text{ and } well(C, Z_C) > w_2 \Rightarrow well(A, Z_B \cup Z_C) > w_1 + w_2 - 1.$$

Proof. It follows from Fact 17. □

As it will be usually clear which set Z_A of subarrays we mean, we will write $well(A) > w$ instead of $well(A, Z_A) > w$.

Let us now show how we can apply Lemma 7. Suppose we have a subarray A of type $T = \sqcup 2 \sqcup k$ and $k \geq 4$. Let B denote the beginning complex of type $\sqcup 2$ and C the subarray of type $\sqcup k$. By Fact 16 and Lemma 6 we have $well(B) > \frac{11}{16}$ and $well(C) > 1 + \frac{k-1}{8}$. By Fact 18 we can use Lemma 7 and get that $well(A) > \frac{11}{16} + \frac{k-1}{8}$. Since $k \geq 4$ we know that $well(A) > 1$, which means that A can always be well-$2\frac{1}{8}$-partitioned.

Suppose now we have a subarray A of type $\sqcup 2 \sqcup 1 \sqcup 1 \sqcup 1$. Let B denote a complex $\sqcup 2$ and C_i the i-th subarray of type $\sqcup 1$. We know that $well(B) > \frac{11}{16}$ and that $well(C_i) > \frac{1}{8}$. Using Lemma 7, we will obtain $well(A) > 1\frac{1}{16}$.

We will further prove, that for every subarray A that the algorithm could not well-$2\frac{1}{8}$-partition, it holds $well(A) > \frac{5}{8}$. Thus if A is followed by B such that $well(B) > 1\frac{3}{8}$, then $well(AB) > 1$ and as all the other conditions of Fact 19 will be satisfied, it will mean that AB can be well-$2\frac{1}{8}$-partitioned.

4.4 Two neighbouring complexes and degenerate subarrays

Let us now look at the subarray C of type $T = \sqcup 2 \sqcup 2$. By Fact 16 and Lemma 7, we know that $well(C) > \frac{3}{8}$. We notice that the situation seems to be worse now than in the case of only one complex, because for example, if C is followed by D of type $\sqcup 4$, then $well(CD) > 1\frac{6}{8}$, and we are not guaranteed, that CD can be well-$2\frac{1}{8}$-partitioned. We have to examine the structure of the subarray of type $\sqcup 2 \sqcup 2$ somewhat closer.

Before doing this however, we introduce degenerate subarrays. We later prove that degenerate subarrays can always be well-$2\frac{1}{8}$-partitioned and also that whenever some subarray we could not well-$2\frac{1}{8}$-partition is followed by a degenerate subarray, then the whole can be well-$2\frac{1}{8}$-partitioned. It will turn out that in the non-degenerate and difficult to partition subarrays the boundaries of $\sqcup 2$, $2\sqcup$ or $\sqcup 1 \sqcup$ and $>$-columns (i.e.1- and 2-columns) form ”nonoverlapping crosses”. More precisely one type of a cross arises in the part of type $\sqcup 2 \sqcup$ if the boundaries of the complexes $\sqcup 2$ and $2\sqcup$ are in the same row, it is formed by these boundaries and the 2-column. The second type of a cross is the one in the part of type $\sqcup 1 \sqcup$, which is formed by the boundary of this part and the 1-column. The crosses non-overlap, if the boundaries of two subarrays, whose common part is a $<$-column, fall into different rows, that is, for example, if in the part of type $2 \sqcup 1 \sqcup$ the boundaries of $2\sqcup$ and $\sqcup 1 \sqcup$ are in different rows or if in the part of type $2 \sqcup 2$ the boundaries of $2\sqcup$ and $\sqcup 2$ are also in different rows. Degenerate subarrays are mostly those whose structure contradicts in some way the idea of "non-overlapping crosses".

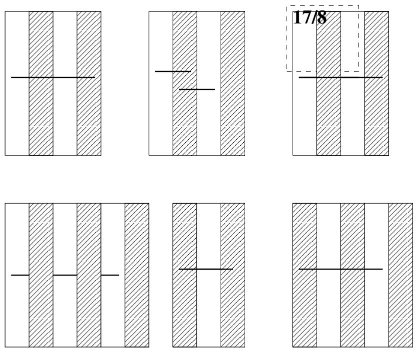

Figure 4.2: Dark columns correspond to $>$-columns and bright ones to $<$-columns. The horizontal line denotes a boundary or boundaries, if they fall in the same row.

Definition 9 *The following kinds of subarrays will be called degenerate (compare Fig 4.2):*

- $\sqcup 1 \sqcup 2$ *with the boundaries of subarrays* $\sqcup 1 \sqcup$ *and* $\sqcup 2$ *in the same row,*
- $\sqcup 2 \sqcup k$ *with the boundaries of the first complex and the "right" complex* $2\sqcup$ *in different rows,*
- $\sqcup 2 \sqcup k$ *with the part above (or under) the boundaries of the complexes* $\sqcup 2$ *and* $2\sqcup$ *greater than* $2\frac{1}{8}$*,*
- $\sqcup 1 \sqcup 1 \sqcup 2$ *with the boundaries of two subarrays of type* $\sqcup 1 \sqcup$ *in the same row.*

Also, those of type $\sqcup T$ *and* $\_\, T$*, where* T *is*

- $2 \sqcup 2$ *with the boundaries of subarrays* $\sqcup 2$*,* $2\sqcup$ *in the same row,*
- $2 \sqcup 1 \sqcup k$ *with the boundaries of* $2\sqcup$ *and* $\sqcup 1 \sqcup$ *in the same row.*

We define also some degenerate subarrays of type $\sqcup 2 \sqcup 3$ *and* $\sqcup 2 \sqcup 3 \sqcup 2$*.*

Lemma 8 *If a degenerate subarray* D *is of type* $\sqcup k_1 \ldots \diamond k_m$*, then*

$$well(D) > 1\frac{3}{8}.$$

Otherwise, if it is of type $\_k_1 \ldots \diamond k_m$*, then for its subarray* D' *of type* $k_1 \ldots k_m$ *we have*

$$well(D') > 1\frac{3}{8}.$$

Proof. First, let us notice that in the subarray of type $\sqcup 1$, the column of type 1 has weight greater than $1\frac{1}{8}$, because the weight of $\sqcup 1$ is greater than $2\frac{1}{8}$ and the weight of a $<$-column is at most 1. Therefore we can in a way write $well(1) > 1\frac{1}{8}$. Also by Lemma 4, for $k \geq 2$, we can write $well(k) > 1 + \frac{k-1}{8}$.

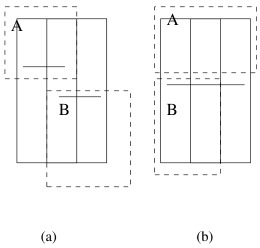

Figure 4.3:

If the boundaries of $\sqcup 2$ and $2\sqcup$ fall in different rows then, the weight of both A and B shown in Figure 4.3(a) is greater than $2\frac{1}{8}$ and thus the weight of $\sqcup 2 \sqcup$ is greater than $N(\sqcup 2 \sqcup) + 1\frac{2}{8}$.

If the weight of A shown in Figure 4.3(b) is greater than $2\frac{1}{8}$, then the weight of $\sqcup 2 \sqcup$ is also greater than $N(\sqcup 2 \sqcup) + 1\frac{2}{8}$ as the weight of B is greater than $2\frac{1}{8}$.

Thus in both cases $well(\sqcup 2 \sqcup) \geq 1\frac{2}{8}$. Since we also have $well(k) > 1\frac{1}{8}$, we get then if either of the above holds, then $well(\sqcup 2 \sqcup k) \geq 1\frac{3}{8}$.

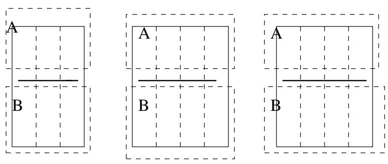

Figure 4.4: The subarrays of type accordingly $2 \sqcup 2$, $\sqcup 2 \sqcup 2$ and $\sqcup 1 \sqcup 2$

If in any one of the cases in Figure 4.4 both A and B have weight at most $2\frac{1}{8}$, then the appropriate subarrays can be well-tiled by taking A and B as two tiles and partitioning the row containing the boundaries into two tiles, each of these two tiles will have weight at most 2, as it will consists of at most two array elements.

Let us now suppose that A (see Figure 4.5) has weight greater than $2\frac{1}{8}$, then each of these subarrays of type T has weight greater than $N(T) + 1\frac{3}{8}$, as A and B have weight greater than $2\frac{1}{8}$ and C has weight greater than $1\frac{1}{8}$ in cases (a) and (c) and greater than $2\frac{1}{8}$ in case (b).

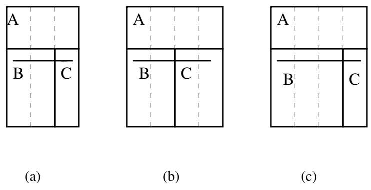

Figure 4.5:

The case when two subarrays of type $\sqcup 1 \sqcup$ overlap and their boundaries fall in the same row is considered at the end of the proof of Lemma 9.

$\square$

Assume now we have a subarray S of type $\sqcup 2 \sqcup 2$, that is not degenerate. We will prove

Fact 20 *For every non-degenarate subarray S of type $\sqcup 2 \sqcup 2$, it holds*

$$well(S) > \frac{21}{32}.$$

Proof. Let us try to give the conditions, under which it cannot be well-tiled. First we can notice that the part $2\sqcup$ should be a complex - a so called "right" complex, because if it were not, then it could be $2\frac{1}{8}$-partitioned into 2 tiles and the whole could be $2\frac{1}{8}$-partitioned into 5 tiles, that is the whole could be well-tiled. Another condition is that the sum of the three elements contained in the boundaries of $\sqcup 2$ and $2\sqcup$ in the part $\sqcup 2 \sqcup$ should be greater than $2\frac{1}{8}$. If it were not, then we could well-tile the subarray, as we know that the part above and under this tile has weight not greater than $2\frac{1}{8}$ (because the subarray is not degenerate), and the rest (the 2-column) can easily be 2-partitioned into 2 tiles.

Let us now estimate the minimal weight of such a subarray. Since it is not degenerate its structure is as shown in Figure 4.6 or it is symmetrical to that in the figure. The boundaries of the first complex $\sqcup 2$, the "right" one $2\sqcup$ and the second one $\sqcup 2$ are represented by variables correspondingly s_1, x_2 and s_1, x_4 and s_2, x_6. Variables x_1, x_3, x_5 and x_7 denote the sums of the values of the elements accordingly above or below the appropriate boundary.

We do not need more variables in $<$-columns and can assume that the whole weight of $<$-columns is concentrated in x_2, x_4, x_6, i.e. in the boundaries. It can be shown using arguments similar to those used in Observation 1. We calculate:

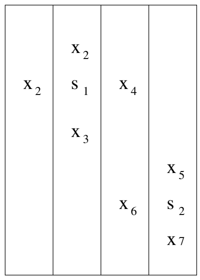

Figure 4.6:

$$
\begin{array}{lrl}
\text{minimize} & s_1 + s_2 + \sum_{i=1}^{7} x_i & \\
\text{subject to} & s_1 & \le 1 \\
& s_1 + x_1 + x_2 & \ge 2\frac{1}{8} \\
& s_1 + x_2 + x_3 & \ge 2\frac{1}{8} \\
& s_1 + x_1 + x_4 & \ge 2\frac{1}{8} \\
& s_1 + x_3 + x_4 + x_6 & \ge 2\frac{1}{8} \\
& s_1 + x_1 + x_3 & \ge 2\frac{1}{8} \\
& s_1 + x_2 + x_4 & \ge 2\frac{1}{8} \\
& s_2 & \le 1 \\
& s_2 + x_4 + x_5 + x_6 & \ge 2\frac{1}{8} \\
& s_2 + x_6 + x_7 & \ge 2\frac{1}{8} \\
& s_2 + x_5 + x_7 & \ge 2\frac{1}{8},
\end{array}
$$

which turns out to be $5\frac{21}{32}$. □

4.5 The algorithm in more detail

In the description of the algorithm in Chapter 3, Section 3.3, we start from the leftmost block and try to well-$2\frac{1}{8}$-partition it. From Lemma 6, we know, that unless it is a complex, its weight is greater than $N(T)+1$ and if it is a complex, its weight is greater than $N(T)+\frac{11}{16}=2\frac{11}{16}$. Therefore in new terminology, if we want to well-$2\frac{1}{8}$-partition it, we in fact want to well- or nearly-tile it, depending on whether its weight is less than $N(T)+1$ or at least $N(T)+1$. Further on we will prove that every subarray encountered in the course of carrying out the algorithm, will have weight at least $N(T)$. Thus we will always be interested in either well- or nearly-tiling it. From Fact 18 we know that nearly-well-tiling is easy. As for well-tiling, we have that it is enough to well-tile some subarray and nearly-tile the preceding and succeeding part. Therefore in order to well-tile some subarray, we will try well-tiling its *small* subarrays i.e. those of type $\diamond k$, $k\diamond$, $\diamond 1 \diamond$, $\diamond 2 \diamond$.

We will call a subarray H *h*eavy, if it is of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$ and $well(H) > 1\frac{3}{8}$ or if it is of type $\_\, k_1 \diamond k_2 \ldots \diamond k_n$ and $well(H') > 1\frac{3}{8}$, where H' is a subarray of H

of type $k_1 \diamond k_2 \ldots \diamond k_n$. Let us notice that every degenerate subarray is heavy. Other examples of heavy subarrays are those of type $\sqcup k$ for $k \geq 4$ or of type $\sqcup 3 \sqcup 1$, $\_2 \sqcup 1 \sqcup 1$.

$S := \emptyset$
while extension of S to the next block possible
  extend S to the next block
  if S of type T has weight at least $N(T) + 1$, nearly-well-tile S and $S := \emptyset$
  else if any small subarray can be well-tiled
  or if S is ended with a heavy subarray,
    well-tile S and $S := \emptyset$
nearly-well-$2\frac{1}{8}$-partition S together with the last $<$-column.

We are going to prove

Theorem 2 *The above algorithm is a $2\frac{1}{8}$-approximation algorithm for the RTILE problem.*

In the following, we first show, that for every subarray S of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$ that appears in the algorithm above, we have $well(S) > \frac{5}{8}$.

We will say that a subarray is *clear* if it does not contain a heavy subarray.

The following lemma is in some sense the most essential and difficult part of the $2\frac{1}{8}$-approximation algorithm. Its proof is given in Chapter 5.2

Lemma 9 *For every clear subarray S of type $T_1 = \sqcup 2(B \sqcup 2)^n (n \geq 0)$, where B denotes either an empty type or types $\sqcup 1$, $\sqcup 1 \sqcup 1$ or $\sqcup 3$, it holds $well(S) > \frac{5}{8}$.*

For every clear subarray S of type $T_2 = 2A \sqcup 2(B \sqcup 2)^n (n \geq 0)$, where B is as above and A denotes either an empty type or a subarray of type $\sqcup 1$, it holds $well(S) > 1$.

We will say that a subarray is *proper*, if it is clear and if it is of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$ or $\diamond k_1 \diamond k_2 \ldots \diamond k_{n\_}$ and its every subarray of type $T_i = \diamond k_1 \ldots \diamond k_i$ consisting of the beginning $2i$ columns has weight less than $N(T_i) + 1$.

The corollary of Lemma 9 is

Lemma 10 *(a) For every proper subarray S*

$$well(S) > \frac{5}{8}.$$

(b) For every proper subarray S of type $\diamond k_1 \diamond k_2 \ldots \diamond k_{n\_}$

$$nearly(S) > 1.$$

Proof.

Claim 1 *For a clear subarray S' of type $T_3 = k_1 \sqcup_2 \sqcup \ldots k_n \sqcup T_2$, where $k_i \neq 2$ for each i between 1 and n and T_2 as in Lemma 9, we have $well(S') > 1$.*

Proof. For every subarray S_1 of type $k\sqcup$, where $k \neq 2$, we have by Lemma 6 $well(S_1) > 1$ and for every clear subarray S_2 of type T_2 we also have $well(S_2) > 1$. Thus by using Lemma 7, we prove the claim. $\square$

Suppose S is proper and is of type $\diamond k_1$, then it must be of type $\sqcup 2$ and by Fact 16, we know its weight is greater than $2\frac{11}{16}$, thus $well(S) > \frac{5}{8}$.

If S is proper and is of type $\diamond k_{1\_}$, then it must be of type $\sqcup 2_\_$. As we can show that a subarray of type $\diamond 2 \diamond$ that cannot be $2\frac{1}{8}$-partitioned into three tiles has minimal weight when it is of the form:

$$\begin{array}{ccc} & \frac{9}{16}^+ & \\ \frac{9}{16}^+ & 1 & \frac{9}{16}^+ \\ & \frac{9}{16}^+ & \end{array}$$

we get that $nearly(S) > 1\frac{1}{4}$.

Suppose now, that the lemma is true for every proper subarray S of type having length at most $2n$ i.e. S has type $\diamond k_1 \diamond k_2 \ldots \diamond k_m$ and $m \leq n$ or S has type $\diamond k_1 \diamond k_2 \ldots \diamond k_{m\_}$ and $m \leq n-1$.

Now suppose we have a subarray S' of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n \diamond k_{n+1}$. Let us denote its subarray of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n$ by S and the ending subarray by S_1.

If S_1 is not a complex, then $well(S_1) > 1$ and thus by Lemma 7 $well(S') > \frac{5}{8}$.

If S_1 is a complex, then either the whole S' is of type T_1,where T_1 is as in Lemma 9 and thus $well(S') > \frac{5}{8}$ or the type of S' has the form $T'\_\ T_3$ and T_3 is as in Claim 1. To see this, recall that S' does not contain a heavy subarray, thus it does not contain subarrays of type $\sqcup k$ for $k \geq 4$ or of types $\sqcup 1 \sqcup 1 \sqcup 1$, $\sqcup 1 \sqcup 3$, $\_2 \sqcup 1 \sqcup 1$.

Let us decompose S' into three subarrays B', C', D' of types correspondingly T', $\_$ and T_3. By Corollary 2, the subarray $B'C'$ can be nearly-well-tiled, by the induction hypothesis we have $well(B') > \frac{5}{8}$ and by Claim 1, $well(D) > 1$. Thus by Lemma 7, we get that $well(S') > \frac{5}{8}$.

Let us now prove part b) of the lemma. Thus we have a subarray S' of type $T = \diamond k_1 \diamond k_2 \ldots \diamond k_{n-1} \diamond k_{n\_}$. First let us notice that if we can well-tile its subarray S of type $T' = \diamond k_1 \diamond k_2 \ldots \diamond k_{n-1}$, then we can nearly-well-tile S', as we can use $N(T'')+2$ tiles for partitioning the subarray of type $T'' = \diamond k_{n\_}$, because $N(T)+1 = N(T') + N(T'') + 2$ and we are always in a position to $2\frac{1}{8}$-partition the subarray of type T'' into $N(T'')+2$ tiles from the definition of types. Let us then assume that S cannot be well-tiled. Therefore its weight is greater than $N(T') + \frac{5}{8}$.

Let us now consider the case when S' is ended with a subarray of type $\_k_{n\_}$. If we can nearly-well-tile the subarray of type $k_{n\_}$, then by Corollary 2 and Fact 17, we can nearly-well-tile S'. Otherwise if it cannot be nearly-well-tiled, than by Lemma 6 its weight is greater than $N(k_n + 1_\_)$ if $k_n \neq 2$ and we get that the weight of S' is greater than $N(T)+1$.

For a subarray S_1 of type $\sqcup 1_\_$ and $\sqcup 3_\_$, we can easily show that $nearly(S_1) > \frac{1}{2}$. Therefore in case S' is ended with a subarray of type $\sqcup 1_\_$ or $\sqcup 3_\_$ we can either nearly-well-tile S and S_1 or if S_1 cannot be nearly-well-tiled, S' has weight greater than $N(T) + \frac{5}{8} + \frac{3}{2} - 1$.

We are left with the case when S' is of type $T = \diamond k_1 \diamond k_2 \ldots \diamond k_{n-1} \sqcup 2_\_$. We can notice similarly as above that if we can well-tile the subarray S of type $T' = \diamond k_1 \ldots \diamond k_{n-1} \sqcup 2$, then we can nearly-well-tile S'. We thus assume that S has weight greater than $N(T') + \frac{5}{8}$. If the weight of the last column of type $\_$ were at least $\frac{3}{8}$ we would have that the weight of S' is greater than $N(T) + 1$. Let us than suppose that its weight is less than $\frac{3}{8}$. Next we calculate the minimal weight of a subarray S_2 of type $\sqcup 2 \_$ that cannot be $2\frac{1}{8}$-partitioned into three tiles subject to the additional condition that the weight of a $\_$-column is at most $\frac{3}{8}$. We get that S_2 has minimal weight when it is of the following form.

$$\begin{array}{ccc} & \frac{6}{8}^{+} & \\ \frac{6}{8}^{+} & 1 & \frac{3}{8} \\ & \frac{6}{8}^{+} & \end{array}$$

Therefore we will also get that either S' can be nearly-well-tiled or has weight greater than $N(T) + 1$.

□

Corollary 2 *If S of type $T = \diamond k_1 \diamond k_2 \ldots \diamond k_n \_ k_{n+1}$ is proper, then its subarray S' of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n \_$ can be nearly-well-tiled.*

Proof. By Lemma 10, we have $nearly(S') > 1$. Thus if S' cannot be nearly-well-tiled, its weight is greater than $N(\diamond k_1 \ldots \diamond k_n \_) + 1$ and the weight of a k_{n+1}-column is from the definition at least $N(k_{n+1}) + 1$. Thus we would get that the weight of S surpasses $N(T) + 1$ contradicting its being proper. □

Let us now deal with subarrays ended with a heavy subarray.

Lemma 11 *Suppose S is of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n \diamond h_1 \ldots \diamond h_m$, and the ending subarray H of type $\diamond h_1 \ldots \diamond h_m$ is heavy. Suppose also that the subarray of type $\diamond k_1 \ldots \diamond k_n \diamond h_1 \ldots \diamond h_{m-1}$ is proper. Then $well(S) > 1$.*

Proof.

Let us denote the beginning subarray of type $\diamond k_1 \ldots \diamond k_n$ by S_1. If H is of type $\sqcup h_1 \ldots \diamond h_m$, then we have $well(H) > 1\frac{3}{8}$ and by Lemma 10 $well(S_1) > \frac{5}{8}$, thus by Lemma 7, we obtain $well(S) > 1$. If H is of type $\_ h_1 \ldots \diamond h_m$, then for its subarray H' of type $h_1 \ldots \diamond h_m$ we have $well(H') > 1\frac{3}{8}$ and by Corollary 2 the subarray of type $\diamond k_1 \diamond k_2 \ldots \diamond k_n \_$ can be nearly-well-tiled. Thus we can use Lemma 7, getting $well(S) > 1$. Let us notice that the lemma deals also with subarrays S that are whole heavy i.e. such that the part $\diamond k_1 \ldots \diamond k_n$ is non-existant. It is so it is obviously true for heavy subarrays of type $\sqcup h_1 \ldots \diamond h_m$ and S cannot be of type $\_ h_1 \ldots \diamond h_m$ as no subarray of type $\_ k$ is proper.

□

From the above we get, that every subarray that the algorithm tries to well-tile, has weight at least $N(T)$. First, it is true for every block. Next, suppose we have a subarray S of type T, that we could not well-tile, then by Lemma 10(a), its weight is greater than $N(T) + \frac{5}{8}$. If we extend it to the next block B of type T', then the weight of SB is greater than $N(T) + \frac{5}{8} + N(T') + \frac{11}{16} > N(T) + N(T') + 1 = N(TT')$.

Another implication of lemma 10(b) is that we are always in a position to nearly-well-$2\frac{1}{8}$-partition the subarray at the end of running the algorithm, as either it can be nearly-well-tiled i.e. nearly-well-$2\frac{1}{8}$-partitioned as it has weight at least $N(T)$ and we use $N(T) + 1$ tiles or it has weight greater than $N(T) + 1$, but then in order to nearly-well-$2\frac{1}{8}$-partition it, we can use $N(T) + 2$ tiles.

Theorem 3 *The approximation with a better factor using the lower bound used in this paper is impossible.*

Proof. Let us consider the subarrays of the type $\sqcup 2(\sqcup 1 \sqcup 2)^n_-$ having weight $N(T)+1$, which for n will be equal to $5n + 3$. Let them have the following structure drawn in Figure 4.7:

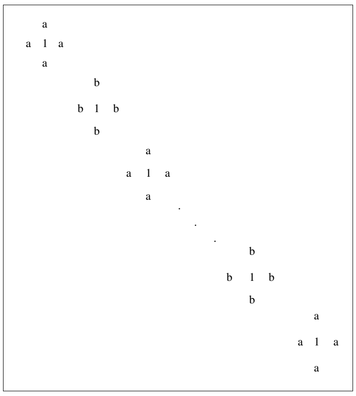

Figure 4.7:

There are as many ones encircled by a as the columns of type 2 and as many ones encircled by b as columns of type 1. Additionally $b = \frac{a}{3}$. Let $a = \frac{9n+6}{16n+12}$. It can be noticed, that if the subarray is to be partitioned into $5n + 3$ tiles, then one tile will have to contain at least one 1 and two a or at least one 1, one a and three b. The weight of such a tile amounts to $2 + \frac{2n}{16n+12}$. $\square$

4.6 Proof of Lemma 6

Let us assume, that we have a subarray S of type $\diamond m$. We want to show that if it cannot be $(2 + x)$-partitioned into m tiles, than its weight is greater than $m + 1 + (m - 1)x$. For abbreviation, we will say that a subarray of type $\diamond m$ is well-tiled, if it is $(2 + x)$-partitioned it into m tiles.

x is a parameter that can have any value greater than 0 and less than 1.

4.6.1 $m = 3$

Obvious.

4.6.2 $m \geq 4$

Let us look at the type of the transposed subarray of the type $\diamond m$. It is not difficult to see, that it can have the form $\diamond 1 \diamond 1 \ldots \diamond 1 \diamond$ (the weight of every column is less than 2, as it consists of two array elements and one of them belongs to a $<$-column). In fact it can be described as $\begin{matrix} \epsilon_1 & 1 & \epsilon_2 & 1 & \ldots & 1 & \epsilon_{d+1} \\ & \delta_1 & & \delta_2 & \ldots & \delta_d & \end{matrix}$, with additionally $\delta_1 + \delta_2 + \ldots \delta_d < 1$.

The number of 1-columns can vary, but cannot surpass $m+1$, as the weight of S of type $\diamond m$ is less than $m+2$. At any rate, whenever the number of 1-columns is $m-2$ or less, we are always able to 2-partition it into m tiles, as we can form $m-2$ tiles, each containing ϵ_i $(i \leq m-2)$ and 1, one tile containing ϵ_{m-1} and one containing the whole $<$-column. If the number of 1-columns is equal to $m-1$ and $m \geq 6$, then we are also always able to 2-partition it into m tiles, because it cannot happen that $\epsilon_1 + \epsilon_2 > 1$ and $\epsilon_3 + \epsilon_4 > 1$ and $\epsilon_5 + \epsilon_6 > 1$, as it would mean that the weight of the whole subarray is greater than $m - 1 + 3$. Thus we can form one tile out of two epsilons and one 1 and $m-2$ tiles out of one epsilon and one 1 and the last mth tile out of the $<$-column.

It remains to examine the situation when the number of 1-columns is $m-1$ and $m = 4$ or $m = 5$.

The number of 1-columns equals 3 and $m = 4$

The type of the subarray can be represented in the following way

$$\begin{matrix} \epsilon_1 & 1 & \epsilon_2 & 1 & \epsilon_3 & 1 & \epsilon_4 \\ & \delta_1 & & \delta_2 & & \delta_3 & \end{matrix}.$$

If any of the sums $\epsilon_1 + \epsilon_2$, $\epsilon_2 + \epsilon_3$ or $\epsilon_3 + \epsilon_4$ were less than $1 + x$, we would be able to well-tile S. Hence, we will further assume all of these sums surpass $1 + x$. Now, let $A = 1 + \epsilon_1 + \delta_1$, $C = 1 + \epsilon_2 + \delta_2$, $D = 1 + \epsilon_3 + \delta_2$ and $B = 1 + \epsilon_4 + \delta_3$. The situation that $A > 2$ and $B > 2$ cannot happen, because since $\epsilon_2 + \epsilon_3 > 1$, it would imply that the weight of the whole subarray is greater than 6.

If both A and B are at most $2 + x$, then both C and D must be greater than $2 + x$. Otherwise we could well-tile the subarray . Now, the minimal weight of the subarray in this situation can be calculated using linear programming:

$$
\begin{array}{ll}
\text{minimize} & \sum_{i=1} 4\epsilon_i + \sum_{i=1} 3\delta_i \\
\text{subject to} & \epsilon_1 < 1 \\
& \epsilon_2 < 1 \\
& \epsilon_3 < 1 \\
& \epsilon_4 < 1 \\
& \epsilon_1 + \epsilon_2 \geq 1 + x \\
& \epsilon_2 + \epsilon_3 \geq 1 + x \\
& \epsilon_3 + \epsilon_4 \geq 1 + x \\
& \epsilon_2 + \delta_2 \geq 1 + x \\
& \epsilon_3 + \delta_3 \geq 1 + x,
\end{array}
$$

which yields

$$
\begin{array}{ccccccc}
x & 1 & 1^- & 1 & 1^- & 1 & x \\
 & & & x & & &
\end{array}.
$$

If $A > 2+x$, then $C < 2$ and $B < 2$, as otherwise the weight of the whole subarray would surpass 6. Again, using linear programming, we get

$$
\begin{array}{ccccccc}
1^- & 1 & x & 1 & 1^- & 1 & x \\
 & x & & & & &
\end{array}.
$$

4.6.3 The number of 1-columns equals 4 and $m = 5$

The subarray we want to examine is of the form

$$
\begin{array}{ccccccccc}
\epsilon_1 & 1 & \epsilon_2 & 1 & \epsilon_3 & 1 & \epsilon_4 & 1 & \epsilon_5 \\
 & \delta_1 & & \delta_2 & & \delta_3 & & \delta_4 &
\end{array}.
$$

As before, we must have $\epsilon_1 + \epsilon_2 > 1 + x$ and $\epsilon_2 + \epsilon_3 > 1 + x$ and $\epsilon_3 + \epsilon_4 > 1 + x$ and $\epsilon_4 + \epsilon_5 > 1 + x$, otherwise we could well-tile the subarray. The expressions $\epsilon_1 + \delta_1$, $\epsilon_3 + \delta_2$, $\epsilon_3 + \delta_3$ and $\epsilon_5 + \delta_4$ all must be less than 1, otherwise the weight of the whole subarray would be greater than 7. Finally, if $\epsilon_2 + \delta_2 \leq 1 + x$ or $\epsilon_4 + \delta_3 \leq 1 + x$, we are able to well-tile the subarray. Thus we will assume that the converse holds and using linear programming we will get, that the subarray has minimal weight when it is of the form:

$$
\begin{array}{ccccccccc}
x & 1 & 1^- & 1 & x & 1 & 1^- & 1 & x \\
 & & & x & & x & & &
\end{array}.
$$

4.6.4 The number of 1-columns equals m

The type is as follows:

$$
\begin{array}{cccccccc}
\epsilon_1 & 1 & \epsilon_2 & 1 & \ldots & \epsilon_m & 1 & \epsilon_{m+1} \\
 & \delta_1 & & \delta_2 & & & \delta_m &
\end{array}.
$$

First, let us consider the situation when $\epsilon_i + \delta_i > 1 + x$ for some $i \neq 2, m$ or $\delta_i + \epsilon_{i+1} > 1 + x$ for some $i \neq 1, m - 1$.

For fixing attention suppose that $\delta_2 + \epsilon_3 > 1 + x$. Now, if any one of the expressions $\epsilon_1 + \epsilon_2, \epsilon_k$ for $k \in \{3, \ldots, m - 1\}$ and $\epsilon_m + \epsilon_{m+1}$ does not surpass x, then we are able to well-tile the subarray, by partitioning ones and epsilons into $m - 1$ tiles and taking the $<$-column as an mth tile. (Note that the sum of two neighbouring epsilons not containing ϵ_3 is less than 1, otherwise the weight of S is greater than $m + 2$.) Therefore all of these expressions are greater than x, which together with the fact that $\delta_2 + \epsilon_3 > 1 + x$, implies that the weight of S is greater than $m + 1 + (m - 1)x$.

Now, let us suppose that the above does not hold. Then if both "corners" $1 + \epsilon_1 + \epsilon_2 + \delta_1$ and $1 + \epsilon_m + \epsilon_{m+1} + \delta_m$ have weight at most $2 + x$, we are able to well-tile S. Of course, only one of them can have weight greater than $2 + x$. Suppose it is $1 + \epsilon_1 + \epsilon_2 + \delta_2$. Next, we can notice that $1 + \delta_2 + \epsilon_2$ must be greater than $2 + x$, otherwise since $1 + \epsilon_1 + \delta_1 \leq 2 + x$, we would be able to well-tile S. Also, all of the expressions ϵ_k for $k \in \{4, \ldots, m - 1$ and $\epsilon_m + \epsilon_{m+1}$ must be greater than x and we get that S has minimal weight, if epsilons and deltas are as follows:

$\epsilon_1 = \epsilon_3 = \epsilon_{m+1} = 0, \epsilon_4 = \epsilon_5 = \ldots = \epsilon_m = \delta_1 = \delta_2 = x$ and $\epsilon_2 = 1^-$ and all other deltas are equal to 0.

4.6.5 The number of 1-columns is equal to $m + 1$

The type is represented in the following way:

$$\begin{array}{ccccccccccc} \epsilon_0 & 1 & \epsilon_1 & 1 & \epsilon_2 & 1 & \ldots & \epsilon_m & 1 & \epsilon_{m+1} \\ & \delta_1 & & \delta_2 & & \delta_3 & & & \delta_{m+1} & \end{array}$$

Let $A_i = 2 + \epsilon_i + \delta_i + \delta_{i+1}, 2 \leq i \leq m - 1$ and $A_1 = 2 + \epsilon_0 + \epsilon_1 + \delta_1 + \delta_2$ and $A_m = 2 + \epsilon_m + \epsilon_{m+1} + \delta_m + \delta_{m+1}$. It is not difficult to see that if S cannot be well-tiled, then for each i, $A_i > 2 + x$.

Therefore we get $\sum_{i=1}^{m} A_i = 2m + mx = \sum_{i=0}^{m+1} \epsilon_i + 2\sum_{i=2}^{m} \delta_i + 2m + \delta_1 + \delta_{m+1}$, which means that $\sum_{i=0}^{m+1} \epsilon_i + \sum_{i=1}^{m+1} \delta_i + \sum_{i=2}^{m} \delta_i \geq mx$.

Let us also consider partitioning the $>$-column, (the one with epsilons) into $m - 1$ tiles. We prove

Claim 2 *A column of weight less than $k + 1$ with k ones, either can be $(2 + x)$-partitioned into $k - 2$ tiles or has weight greater than $k + (k - 3)x$.*

Proof. For $k = 4$, it is obvious. Suppose that for $k \leq i$, the claim is true. Now, let us consider a column of weight less than $i + 2$ with $i + 1$ ones. It can be represented in the following way $\epsilon_0\ 1\ \epsilon_1\ 1\ \ldots \epsilon_i\ 1\ \epsilon_{i+1}$. If both $\epsilon_0 + \epsilon_1 \leq x$ and $\epsilon_i + \epsilon_{i+1} \leq x$, then we can $(2 + x)$-partition the column into $i - 1$ tiles. Thus suppose that $\epsilon_0 + \epsilon_1 > x$. Then if we could $(2 + x)$-partition the part of the column corresponding to $1\ \epsilon_2\ 1 \ldots 1\ \epsilon_{i+1}$, the whole column could still be $(2 + x)$-partitioned into $i - 1$ tiles. If not, then the whole column must have weight greater than $i + 1 + (i - 2)x$. □

Therefore we get that if we cannot $2\frac{1}{8}$-partition the $>$-column into $m - 1$ tiles, then $\sum_{i=0}^{m+1} \epsilon_i \geq (m - 2)x$.

Finally, if we calculate

$$\begin{array}{lr} \text{minimize} & \sum_{i=0}^{m+1} \epsilon_i + \sum_{i=1}^{m+1} \delta_i \\ \text{subject to} & \sum_{i=0}^{m+1} \epsilon_i \geq (m - 2)x \\ & \sum_{i=0}^{m+1} \epsilon_i + \sum_{i=1}^{m+1} \delta_i + \sum_{i=2}^{m} \delta_i \geq mx, \end{array}$$

we will get that $\sum_{i=0}^{m+1} \epsilon_i + \sum_{i=1}^{m+1} \delta_i \geq (m-1)x$.

$\square$

Chapter 5

Proof of Lemma 9

5.1 The main idea

One of the key ideas of proof of Lemma 9 is as follows.
Suppose we have the following linear program

$$\begin{array}{lll} \text{minimize} & cx & \\ \text{subject to} & Ax & \leq b \end{array} \tag{*}$$

and A is of the form as in Figure 5.1, let us call it *stair-like*.

Had it not been for the variables $x_{i_1}, x_{i_2}, \ldots, x_{i_{k-1}}$ we could partition (*) into k smaller linear programs and calculate each one separately. In a way we can still do it, if we treat the variables $x_{i_1}, \ldots, x_{i_{k-1}}$ as constants. Then, in order to find the overall solution we take the minimum over all possible choices of the values for these constants. More precisely, let k functions f_i be defined as follows.

$$\begin{array}{rlrl} f_1(t_1) = & \text{minimize} & \sum_{i=1}^{i} {}_1 c_i x_i & \\ & \text{subject to} & A_1[x_1, \ldots, x_{i_1}] & \leq b_1 \\ & & x_{i_1} & = t_1 \end{array}$$

$$\begin{array}{rlrl} f_j(t_{j-1}, t_j) = & \text{minimize} & \sum_{i=i_{j-1}+1}^{i_j} c_i x_i & \\ & \text{subject to} & A_j[x_{i_{j-1}}, \ldots, x_{i_j}] & \leq b_j \\ & & x_{i_{j-1}} & = t_{j-1} \\ & & x_{i_j} & = t_j \end{array}$$

$$\begin{array}{rlrl} f_k(t_{k-1}) = & \text{minimize} & \sum_{i=i_{k-1}+1}^{n} c_i x_i & \\ & \text{subject to} & A_k[x_{i_{k-1}}, \ldots, x_n] & \leq b_k \\ & & x_{i_{k-1}} & = t_{k-1} \end{array}$$

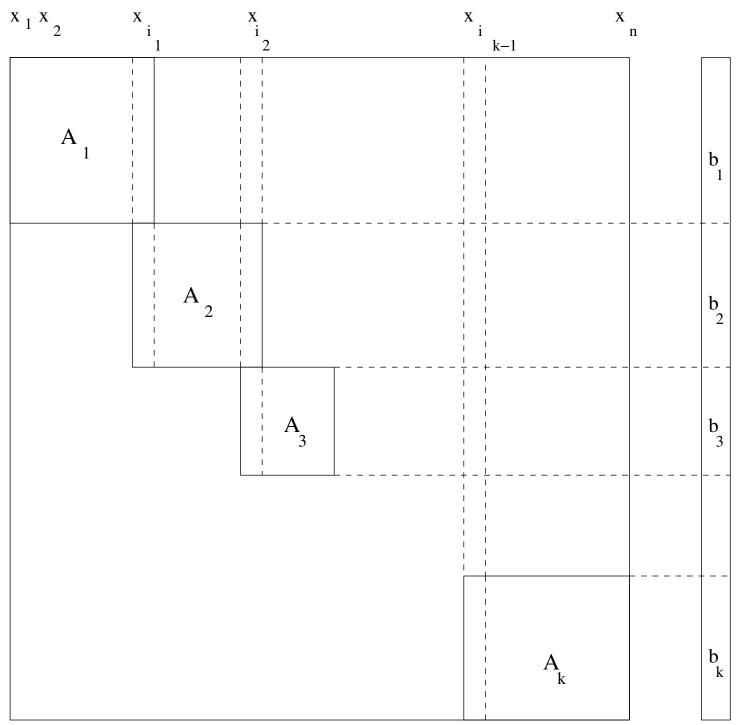

Figure 5.1: Non-zero elements are contained in $A_1, A_2, \ldots, A_k$. Each A_i shares at most one variable with its neighbour.

The task of solving (*) amounts to calculating

$$\min_{t_1, t_2, \ldots, t_{k-1}} \{f_1(t_1) + \sum_{j=2}^{k-1} f_j(t_{j-1}, t_j) + f_k(t_{k-1})\}.$$

5.2 Reduction to the linear program

For now, we will restrict ourselves to those subarrays that do not contain a 3-column.

By Fact 18, if any small subarray of S can be well-tiled, then S can be well-tiled. Therefore we will assume that no small subarray of S can be well-tiled.

Since S does not contain degenerate subarrays, its structure can be represented by a set of nonoverlapping “crosses”. The minimal weight of such subarrays can be calculated by solving linear programs which minimize the sum of the variables in the ”crosses” subject to appropriate inequalities.

The variables in the part of the subarray of type ⊔2⊔ can form themselves into one of the following patterns (or the symmetrical ones):

$$\begin{array}{ccccccc}
 & | & & & & & \\
- & s_{-1} & x_6 & & & & \\
 & | & & & & & \\
 & & & x_1 & & & \\
 & & x_4 & s_0 & x_2 & & \\
 & & & x_3 & & & \\
 & & & & & | & \\
 & & & & x_5 & s_1 & - \\
 & & & & & | &
\end{array}$$

or

$$\begin{array}{ccccccc}
 & | & & & & & \\
- & s_{-1} & x_6 & & & & \\
 & | & & & & & \\
 & & & & & | & \\
 & & & & x_5 & s_1 & - \\
 & & & & & | & \\
 & & & x_1 & & & \\
 & & x_4 & s_0 & x_2 & & \\
 & & & x_3 & & &
\end{array}$$

(The column of type 2 consists of variables s_0, x_1 and x_3, the preceding $<$-column consists of variables x_4 and x_6, the succeeding $<$-column of variables x_2 and x_5 and s_{-1} and s_1 do not belong to the part $\sqcup 2 \sqcup$, they belong to neighbouring $>$-columns.)

The variables satisfy appropriately in the first case

$$\begin{array}{rcl}
s_0 & \leq & 1 \\
s_0 + x_1 + x_3 & > & 2\frac{1}{8} \\
s_0 + x_2 + x_4 & > & 2\frac{1}{8} \\
s_0 + x_1 + x_4 + x_6 & > & 2\frac{1}{8} \\
s_0 + x_1 + x_2 & > & 2\frac{1}{8} \\
s_0 + x_3 + x_4 & > & 2\frac{1}{8} \\
s_0 + x_2 + x_3 + x_5 & > & 2\frac{1}{8}
\end{array}$$

and in the other

$$\begin{array}{rcl}
s_0 & \leq & 1 \\
s_0 + x_1 + x_3 & > & 2\frac{1}{8} \\
s_0 + x_2 + x_4 & > & 2\frac{1}{8} \\
s_0 + x_1 + x_4 + x_6 & > & 2\frac{1}{8} \\
s_0 + x_1 + x_2 + x_5 & > & 2\frac{1}{8} \\
s_0 + x_3 + x_4 & > & 2\frac{1}{8} \\
s_0 + x_2 + x_3 & > & 2\frac{1}{8}.
\end{array}$$

The variables in the part of type $\sqcup 1 \sqcup$ can form themselves into the same patterns as in the part of type $\sqcup 2 \sqcup$, but they satisfy appropriately in the first case

$$\begin{array}{rcl} s_0 & \leq & 1 \\ s_0 + x_1 + x_3 + x_4 + x_6 & > & 2\frac{1}{8} \\ s_0 + x_1 + x_2 + x_4 + x_6 & > & 2\frac{1}{8} \\ s_0 + x_1 + x_2 + x_3 + x_5 & > & 2\frac{1}{8} \\ s_0 + x_2 + x_3 + x_4 + x_5 & > & 2\frac{1}{8} \end{array}$$

and in the other

$$\begin{array}{rcl} s_0 & \leq & 1 \\ s_0 + x_1 + x_3 + x_4 + x_6 & > & 2\frac{1}{8} \\ s_0 + x_1 + x_2 + x_4 + x_5 + x_6 & > & 2\frac{1}{8} \\ s_0 + x_1 + x_2 + x_3 + x_5 & > & 2\frac{1}{8} \\ s_0 + x_2 + x_3 + x_4 & > & 2\frac{1}{8} \end{array}$$

The variables are also required to be non-negative.

5.3 Subarrays of type $\sqcup 2(\sqcup 1 \sqcup 2)^n$

Let us first prove the lemma for subarrays of type $\sqcup 2(\sqcup 1 \sqcup 2)^n$. Let S be of type $\sqcup 2(\sqcup 1 \sqcup 2)^n$. We use linear programming to lower bound the weight of S. The structure of the beginning part corresponding to the subarray of type $\sqcup 2$ will be represented as

$$\begin{array}{ccc} & x_{n+1,1} & \\ x_{n+1,2} & s_{n+1} & x_{n,8} \\ & x_{n+1,3} & \end{array}$$

$x_{n,8}$ belongs to the column of type $\sqcup$ behind the first 2-column.

The structure of $(\sqcup 1 \sqcup 2)^n$ consists of n subarrays of type $\sqcup 1 \sqcup 2$. The subarrays are numerated from right to left.

The structure of the i-th subarray of type $\sqcup 1 \sqcup 2$:

$$\begin{array}{cccccc} s_{i+1,1} & x_{i,8} & & & \\ & & x_{i,5} & & \\ & x_{i,6} & s_{i,2} & x_{i,4} & \\ & & x_{i,7} & & \\ & & & & x_{i,1} \\ & & & x_{i,2} & s_{i,1} \\ & & & & x_{i,3} \end{array}$$

or

$$\begin{array}{cccccc} & & x_{i,5} & & \\ & x_{i,6} & s_{i,2} & x_{i,4} & \\ & & x_{i,7} & & \\ s_{i+1,1} & x_{i,8} & & & \\ & & & & x_{i,1} \\ & & & x_{i,2} & s_{i,1} \\ & & & & x_{i,3} \end{array}.$$

($s_{i+1,i}$ belongs to the $(i+1)$-st subarray of course.)

In order to lower bound the weight of S we will count the minimal value of $W = \sum_{i=1}^{n}(\sum_{k=1}^{8} x_{i,k}+s_{i,1}+s_{i,2})+\sum_{j=1}^{3} x_{n+1,j}+s_{n+1}$ subject to the following inequalities.

Since we are interested in the lower bound on the weight, we can use not sharp inequalities instead of the sharp ones. Also we will not require the variable to be non-negative. We are allowed to do so, because $\min\{cx : Ax \geq b,\ x \geq 0\} \geq \min\{cx : Ax \geq b\}$.

The inequalities that concern subsequent subarrays of type $\sqcup 1 \sqcup 2$ are

$$
\begin{array}{llllllllll}
s_{i,1} & & +x_{i,2} & & & & & & +x_{i-1,8} & \geq 2\frac{1}{8} \\
s_{i,1} & +x_{i,1} & & +x_{i,3} & & & & & & \geq 2\frac{1}{8} \\
s_{i,1} & & +x_{i,2} & +x_{i,3} & & & & & & \geq 2\frac{1}{8} \\
s_{i,1} & +x_{i,1} & +x_{i,2} & & +x_{i,4} & & & & & \geq 2\frac{1}{8} \\
s_{i,2} & & +x_{i,2} & & +x_{i,4} & +x_{i,5} & & +x_{i,7} & & \geq 2\frac{1}{8} \\
s_{i,2} & & +x_{i,2} & & +x_{i,4} & & +x_{i,6} & +x_{i,7} & & \geq 2\frac{1}{8} \\
s_{i,2} & & & & +x_{i,4} & +x_{i,5} & +x_{i,6} & & +x_{i,8} & \geq 2\frac{1}{8} \\
s_{i,2} & & & & +x_{i,5} & & +x_{i,6} & +x_{i,7} & +x_{i,8} & \geq 2\frac{1}{8} \\
-s_{i,1} & & & & & & & & & \geq -1 \\
-s_{i,2} & & & & & & & & & \geq -1
\end{array}
$$

(the first inequality does not occur in the leftmost subarray) or respectively:

$$
\begin{array}{llllllllll}
s_{i,1} & & +x_{i,2} & & & & & & +x_{i-1,8} & \geq 2\frac{1}{8} \\
s_{i,1} & +x_{i,1} & & +x_{i,3} & & & & & & \geq 2\frac{1}{8} \\
s_{i,1} & & +x_{i,2} & +x_{i,3} & & & & & & \geq 2\frac{1}{8} \\
s_{i,1} & +x_{i,1} & +x_{i,2} & & +x_{i,4} & & & & & \geq 2\frac{1}{8} \\
s_{i,2} & & +x_{i,2} & & +x_{i,4} & +x_{i,5} & & +x_{i,7} & & \geq 2\frac{1}{8} \\
s_{i,2} & & & & +x_{i,4} & +x_{i,5} & +x_{i,6} & & & \geq 2\frac{1}{8} \\
s_{i,2} & & +x_{i,2} & & +x_{i,4} & & +x_{i,6} & +x_{i,7} & +x_{i,8} & \geq 2\frac{1}{8} \\
s_{i,2} & & & & & +x_{i,5} & +x_{i,6} & +x_{i,7} & +x_{i,8} & \geq 2\frac{1}{8} \\
-s_{i,1} & & & & & & & & & \geq -1 \\
-s_{i,2} & & & & & & & & & \geq -1
\end{array}
$$

and the inequalities concerning the beginning subarray of type $\sqcup 2$ are

$$
\begin{array}{llllll}
s_{n+1} & & +x_{n+1,2} & & +x_{n,8} & \geq 2\frac{1}{8} \\
s_{n+1} & +x_{n+1,1} & & +x_{n+1,3} & & \geq 2\frac{1}{8} \\
s_{n+1} & & +x_{n+1,2} & +x_{n+1,3} & & \geq 2\frac{1}{8} \\
s_{n+1} & +x_{n+1,1} & +x_{n+1,2} & & & \geq 2\frac{1}{8} \\
-s_{n+1} & & & & & \geq -1
\end{array}
$$

Observe that we have not included inequalities stating that any subarray of type $2\sqcup$ is a complex, for example if we wanted to we should have added inequalities

$$\begin{aligned} {}_{i+1,1} + x_{i+1,1} + x_{i,8} &\geq 2\tfrac{1}{8} \\ s_{i+1,1} + x_{i+1,3} + x_{i,6} + x_{i,8} &\geq 2\tfrac{1}{8} \end{aligned}$$

or

$$\begin{aligned} {}_{i+1,1} + x_{i+1,3} + x_{i,8} &\geq 2\tfrac{1}{8} \\ s_{i+1,1} + x_{i+1,1} + x_{i,6} + x_{i,8} &\geq 2\tfrac{1}{8}, \end{aligned}$$

depending on whether the variable $x_{i,6}$ is above or below the variable $x_{i,8}$. Omitting any inequalities cannot do any harm, as we are interested in the lower bound on the weight of S and this way we can avoid many complications.

We show, that W is minimal, when all the constraints are satisfied with equality. Let us consider the dual of this linear program. For $2 \leq i \leq n$ the dual variable $y_{i,j}$ corresponds to the $(j+1)$st inequality in the ith subarray of type $\sqcup 1 \sqcup 2$ (in other words inequalities are numerated from 0 to 9) and the dual variable $y_{1,j}$ corresponds to the jth inequality in the first subarray of type $\sqcup 1 \sqcup 2$(the inequalities are numerated from 1 to 9, recall also that the first subarray of type $\sqcup 1 \sqcup 2$ is the rightmost) and the variable $y_{n+1,j}$ corresponds to the $(j+1)$th inequality in the beginning subarray of type $\sqcup 2$.

It turns out that regardless of the structure of every subarray of type $\sqcup 1 \sqcup 2$ we get the same equalities. For $2 \leq i \leq n$, these are

$$\begin{aligned} y_{i,0} + y_{i-1,6} + y_{i-1,7} &= 1 \\ y_{i,1} + y_{i,3} &= 1 \\ y_{i,0} + y_{i,2} + y_{i,3} + y_{i,4} + y_{i,5} &= 1 \\ y_{i,1} + y_{i,2} &= 1 \\ y_{i,3} + y_{i,4} + y_{i,5} + y_{i,6} &= 1 \\ y_{i,4} + y_{i,6} + y_{i,7} &= 1 \\ y_{i,5} + y_{i,6} + y_{i,7} &= 1 \\ y_{i,4} + y_{i,5} + y_{i,7} &= 1 \\ y_{i,0} + y_{i,1} + y_{i,2} + y_{i,3} - y_{i,8} &= 1 \\ y_{i,4} + y_{i,5} + y_{i,6} + y_{i,7} - y_{i,9} &= 1 \end{aligned}$$

(They correspond in turn to $x_{i-1,8}, x_{i,1}, x_{i,2}, \ldots, x_{i,7}, s_{i,1}, s_{i,2}$). For $i = 1$, there are eight equalities the same as those above beginning from the second one with that modification that $y_{1,0}$ does not appear in them. For $i = n+1$ the equalities are

$$\begin{aligned} y_{n+1,0} + y_{n,6} + y_{n,7} &= 1 \\ y_{n+1,1} + y_{n+1,3} &= 1 \\ y_{n+1,0} + y_{n+1,2} + y_{n+1,3} &= 1 \\ y_{n+1,1} + y_{n+1,2} &= 1 \\ y_{n+1,0} + y_{n+1,1} + y_{n+1,2} + y_{n+1,3} - y_{n+1,4} &= 1 \end{aligned}$$

(They correspond in turn to $x_{n,8}, x_{n+1,1}, x_{n+1,2}, x_{n+1,3}, s_{n+1}$.)

Since $\min\{cx : Ax \geq b\} = \max\{by : A^T y = c, y \geq 0\}$, it suffices to show, that all $y's$ are nonnegative. Solving the equations "part-wise", we get that for $2 \leq i \leq n$

$$
\begin{aligned}
y_{i,0} &= 1 - y_{i-1,6} - y_{i-1,7} \\
y_{i,1} &= \tfrac{3}{4} + \tfrac{3}{4} y_{i,0} \\
y_{i,2} &= y_{i,3} = \tfrac{1}{4} - \tfrac{3}{4} y_{i,0} \\
y_{i,4} &= y_{i,5} = y_{i,6} = \tfrac{1}{4} + \tfrac{1}{4} y_{i,0} \\
y_{i,7} &= \tfrac{1}{2} - \tfrac{1}{2} y_{i,0} \\
y_{i,8} &= \tfrac{1}{4} y_{i,0} + \tfrac{1}{4} \\
y_{i,9} &= \tfrac{1}{4} + \tfrac{1}{4} y_{i,0}
\end{aligned}
$$

For $i = 1$ we have $y_{1,1} = \frac{3}{10}$, $y_{1,2} = y_{1,3} = y_{1,8} = \frac{3}{10}$, $y_{1,4} = y_{1,5} = y_{1,6} = \frac{1}{5}$, $y_{1,7} = \frac{3}{5}$ and $y_{1,9} = \frac{1}{5}$. For $2 \leq i \leq n$, if all $y's$ are to be nonnegative, it should be $y_{i,0} \leq \frac{1}{3}$ and $y_{i-1,6} + y_{i-1,7} \leq 1$, which in turn is equivalent to $1 \geq y_{i-1,6} + y_{i-1,7} \geq \frac{2}{3}$. Since $y_{1,6}$ and $y_{1,7}$ are equal to $\frac{1}{5}$ and $\frac{3}{5}$, for $i = 2$ the condition holds. On the other hand,$y_{i,6} + y_{i,7} = \frac{3}{4} - \frac{1}{4} y_{i,0}$. Thus $0 \leq y_{i,0} \leq \frac{1}{3}$ implies $\frac{3}{4} \geq y_{i,6} + y_{i,7} \geq \frac{2}{3}$ and we have proved that for $1 \leq i \leq n$ all $y_{i,j}$ are nonnegative.

For $i = n + 1$ we have

$$
\begin{aligned}
y_{n+1,0} &= 1 - y_{n,6} - y_{n,7} \\
y_{n+1,1} &= y_{n+1,3} = \tfrac{1}{2}(1 - y_{n+1,0}) \\
y_{n+1,2} &= y_{n+1,4} = \tfrac{1}{2} + \tfrac{1}{2} y_{n+1,0}.
\end{aligned}
$$

Since from the above $y_{n,6} + y_{n,7} \leq \frac{3}{4}$, $0 \leq y_{n+1,0} \leq 1$ and thus all $y_{n+1,j}$ are nonnegative.

This means, that W is minimal when the values of the variables are as shown below.

The beginning subarray of type $\sqcup 2$:

The ith subarray of type $\sqcup 1 \sqcup 2$ (with subarrays numerated from left to right and from 0 to $n-1$ this time):

$$
\begin{array}{cccccc}
1 & c_i & & & & \\
 & & \frac{9}{16} - \frac{3}{4}c_i & & & \\
 & \frac{9}{16} - \frac{3}{4}c_i & 1 & \frac{c_i}{2} & & \\
 & & \frac{c_i}{2} & & & \\
 & & & & \frac{9}{16} - \frac{c_i}{4} & \\
 & & & \frac{9}{16} - \frac{c_i}{4} & 1 & \frac{9}{16} + \frac{c_i}{4} = c_{i+1} \\
 & & & & \frac{9}{16} + \frac{c_i}{4} &
\end{array}
$$

or

$$
\begin{array}{cccccc}
 & & \frac{9}{16}+\frac{c_i}{4} & & & \\
 & \frac{9}{16}-\frac{3c_i}{4} & 1 & \frac{c_i}{2} & & \\
 & & -\frac{c_i}{2} & & & \\
1 & c_i & & & & \\
 & & & & \frac{9}{16}-\frac{c_i}{4} & \\
 & & & \frac{9}{16}-\frac{c_i}{4} & 1 & \frac{9}{16}+\frac{c_i}{4}=c_{i+1} \\
 & & & & \frac{9}{16}+\frac{c_i}{4} &
\end{array}
$$

Therefore, we get that $W = 1 + 3 \cdot \frac{9}{16} + \sum_{i=1}^{n} 4\frac{13}{16} + \frac{c_{i-1}}{4}$, where $c_0 = \frac{9}{16}$ and $c_{i+1} = \frac{9}{16} + \frac{c_i}{4}$. Further we calculate that $W = 2\frac{11}{16} + \sum_{i=1}^{n} 4\frac{13}{16} + \frac{c_{i-1}}{4} = 2\frac{11}{16}(\frac{b}{3} - \frac{1}{4})(1 - \frac{1}{4^n}) + 5n$, which converges to $N(T) + \frac{5}{8}$, where $T = \sqcup 2(\sqcup 1 \sqcup 2)^n$.

5.4 How to use the main idea

Things get complicated in subarrays of other types and it is not so easy to find the relationship between the type of a subarray and the set of inequalities which are satisfied with equality in an optimal solution. To circumvent this problem we decompose the subarray (and at the same time the linear program describing it) into subarrays of certain types, for which it is easy to calculate their minimal weight and then appropriately piece together the information to estimate the minimal weight of the whole subarray.

We decompose a subarray into subarrays of three *uniform* types. A type T is *uniform*, if it is of the form $T_s T_b^k T_p$ for some types T_s, T_p, T_b, where T_s is a suffix of T_b and T_p a prefix, and a natural positive k. T_b is called the *base* of T. We will consider *clusters, non-clusters* and *long non-clusters* - arrays of a uniform type of bases correspondingly: $\sqcup 2$, $\sqcup 2 \sqcup 1$ and $\sqcup 2 \sqcup 1 \sqcup 1$.

The linear program describing the subarray S is decomposed accordingly. A smaller linear program corresponding to a certain subpart consists of all the inequalities in the original linear program that concern the variables "building" the subpart.

The subarray S is decomposed in such a way that the line of partition always falls in a 2-column and so that firstly clusters are maximally large and subject to this condition non-clusters are maximally large.

For example a subarray of type $\sqcup 2 \sqcup 2 \sqcup 1 \sqcup 1 \sqcup 2 \sqcup 1 \sqcup 2$ would be decomposed into subarrays of type $\sqcup 2 \sqcup 2$ and $\sqcup 1 \sqcup 1 \sqcup$ and $2 \sqcup 1 \sqcup 2$.

As for the partition of the linear program describing S, we may notice that it is not stair-like, as some of the linear programs, into which we decomposed the original one, share more than one variable. We will deal with this problem by throwing away some of the inequalities.

Let us first show how we do it in case of a subarray S of type $2 \sqcup 2 \sqcup 1 \sqcup 2$. Suppose its structure is represented as follows

$$
\begin{array}{ccccccc}
x_2 & & & & & & \\
x_1 & x_3 & & & & & \\
x_4 & & & & & & \\
 & & x_6 & & & & \\
 & x_7 & x_5 & x_9 & & & \\
 & & x_8 & & & & \\
 & & & & x_{11} & & \\
 & & & x_{12} & x_{10} & x_{14} & \\
 & & & & x_{13} & & \\
 & & & & & & x_{16} \\
 & & & & & x_{17} & x_{15} \\
 & & & & & & x_{18}
\end{array}
$$

We decompose S into S_1 of type $2 \sqcup 2$ and S_2 of type $\sqcup 1 \sqcup 2$. The linear program

describing S is decomposed into

$$
\begin{array}{lrl}
\text{minimize} & \sum_{i=1}^{8} x_i & \\
\text{subject to} & x_1 & \leq 1 \\
 & x_1 + x_2 + x_3 & \geq 2\frac{1}{8} \\
 & x_1 + x_2 + x_4 & \geq 2\frac{1}{8} \\
 & x_1 + x_3 + x_4 + x_7 & \geq 2\frac{1}{8} \\
 & x_5 & \leq 1 \\
 & x_3 + x_5 + x_6 + x_7 & \geq 2\frac{1}{8} \\
 & x_5 + x_7 + x_8 & \geq 2\frac{1}{8} \\
 & x_5 + x_6 + x_8 & \geq 2\frac{1}{8} \\
 & x_5 + x_7 + x_9 & \geq 2\frac{1}{8} \\
 & x_5 + x_6 + x_9 & \geq 2\frac{1}{8} \\
 & x_5 + x_8 + x_9 + x_{12} & \geq 2\frac{1}{8}
\end{array}
$$

and

$$
\begin{array}{lrl}
\text{minimize} & \sum_{i=9}^{18} x_i & \\
\text{subject to} & x_5 + x_7 + x_9 & \geq 2\frac{1}{8} \\
 & x_5 + x_6 + x_9 & \geq 2\frac{1}{8} \\
 & x_5 + x_8 + x_9 + x_{12} & \geq 2\frac{1}{8} \\
 & x_10 & \leq 1 \\
 & x_9 + x_{10} + x_{11} + x_{12} + x_{13} & \geq 2\frac{1}{8} \\
 & x_9 + x_{10} + x_{11} + x_{12} + x_{14} & \geq 2\frac{1}{8} \\
 & x_{10} + x_{11} + x_{13} + x_{14} + x_{17} & \geq 2\frac{1}{8} \\
 & x_{10} + x_{12} + x_{13} + x_{14} + x_{17} & \geq 2\frac{1}{8} \\
 & x_15 & \leq 1 \\
 & x_{14} + x_{15} + x_{16} + x_{17} & \geq 2\frac{1}{8} \\
 & x_{15} + x_{17} + x_{18} & \geq 2\frac{1}{8} \\
 & x_{15} + x_{16} + x_{18} & \geq 2\frac{1}{8}
\end{array}
$$

There are three inequalities containing variables both from S_1 and S_2. The one that we will keep is $x_5 + x_7 + x_9 \geq 2\frac{1}{8}$ and it will belong to a linear program corresponding to S_2. Now the partition is stair-like and we define $f_1(b) =$

$$\begin{array}{lrl}
\text{minimize} & \sum_{i=1}^{8} x_i & \\
\text{subject to} & x_1 & \leq 1 \\
& x_1+x_2+x_3 & \geq 2\frac{1}{8} \\
& x_1+x_2+x_4 & \geq 2\frac{1}{8} \\
& x_1+x_3+x_4+x_7 & \geq 2\frac{1}{8} \\
& x_5 & \leq 1 \\
& x_3+x_5+x_6+x_7 & \geq 2\frac{1}{8} \\
& x_5+x_7+x_8 & \geq 2\frac{1}{8} \\
& x_5+x_6+x_8 & \geq 2\frac{1}{8} \\
& x_5+x_7+x_9 & \geq 2\frac{1}{8} \\
& x_9 & = b
\end{array}$$

and $f_2(b) =$

$$\begin{array}{lrl}
\text{minimize} & \sum_{i=9}^{18} x_i & \\
\text{subject to} & x_5+x_7+x_9 & \geq 2\frac{1}{8} \\
& x_10 & \leq 1 \\
& x_9+x_{10}+x_{11}+x_{12}+x_{13} & \geq 2\frac{1}{8} \\
& x_9+x_{10}+x_{11}+x_{12}+x_{14} & \geq 2\frac{1}{8} \\
& x_{10}+x_{11}+x_{13}+x_{14}+x_{17} & \geq 2\frac{1}{8} \\
& x_{10}+x_{12}+x_{13}+x_{14}+x_{17} & \geq 2\frac{1}{8} \\
& x_15 & \leq 1 \\
& x_{14}+x_{15}+x_{16}+x_{17} & \geq 2\frac{1}{8} \\
& x_{15}+x_{17}+x_{18} & \geq 2\frac{1}{8} \\
& x_{15}+x_{16}+x_{18} & \geq 2\frac{1}{8} \\
& x_9 & = b
\end{array}$$

The task of finding the minimal weight of S is now to calculate $\min_b\{f_1(b)+f_2(b)\}$.

In a similar way we will get rid of some inequalities so that the partition is stair like. From all inequalities containing variables from two neighbouring subarrays in the decomposition we will keep the one concerning the boundary of the subarray of type $\sqcup 2\sqcup$ and will include it in the linear program corresponding to that subpart that contains a 2-column.

Accordingly, we introduce clusters, non-clusters and long non-clusters with the left or right parameter or both. Such clusters, etc. with a parameter or parameters are to denote that they are integral parts of the larger subarray and parameters correspond to the shared variables. More formally their definition is

Definition 10 *A* **cluster, non-cluster or long non-cluster with the left parameter** *b is any clear subarray of a corresponding type, which satisfies additionally the following property. In the case it begins with a complex of type $2\sqcup$, then the boundary of this complex has weight at least $2\frac{1}{8}-b$. In the case it begins with a subarray of type $\sqcup 1\sqcup$, then in the leftmost column of type $\sqcup$ there exists an element outside the boundary of $\sqcup 1\sqcup$, that has weight b.*

Similarly we can define clusters, etc. with the right parameter as well as clusters etc. with the left and right parameter.

5.5 Clusters, non-clusters and long non-clusters

In this section we establish the minimal weight of clusters, non-clusters and long non-clusters depending on the values of their parameters.

5.5.1 Non-clusters

Fact 21 *The minimal weight of the non-cluster* $\sqcup 1 \sqcup$ *with parameters* b_1 *and* b_2 *is greater than* $2\frac{1}{2} + \frac{b_1+b_2}{3}$.

Proof.
Assuming that the variables occurring in the non-cluster are s, x_1, x_2, x_3, x_4, we count, using linear programming, the minimal value of the expression $s + x_1 + x_2 + x_3 + x_4 + b_1 + b_2$ subject to appropriate inequalities. $\square$

For larger non-clusters of type $(\sqcup 1(\sqcup 2 \sqcup 1)^k \sqcup$ we have

Fact 22 *The weight of the non-cluster* $\sqcup 1(\sqcup 2 \sqcup 1)^k \sqcup$ *with parameters* b_1 *and* b_2 *is greater than* $2\frac{1}{2} + \frac{b_1+b_2}{3}$ *plus* $5k$.

Proof. Let us assume that we have a non-cluster of type $\sqcup 1 \sqcup 2 \sqcup 1 \sqcup$ with parameters b_1 and b_2. We can look at it as the concatenation of two non-clusters of type $\sqcup 1 \sqcup$ and a 2-column The non-clusters have parameters accordingly b_1 and x, and y and b_2, with $x + y > \frac{9}{8}$. Then the weight of the non-cluster $\sqcup 1 \sqcup 2 \sqcup 1 \sqcup$ is not less than $2\frac{1}{2} + \frac{b_1+x}{3} + 2\frac{1}{2} + \frac{\frac{9}{8}-x+b_2}{3} + 2\frac{1}{8}$. $\square$

In Section 5.3 we have already estimated the weight of the non-cluster $(\sqcup 1 \sqcup 2)^k$ with one parameter:

Fact 23 *The minimal weight of the array* $\sqcup 1 \sqcup 2$ *with parameter* b *is greater than* $4\frac{13}{16} + \frac{b}{4}$. *The minimal weight of the array* $(\sqcup 1 \sqcup 2)^n$ *with parameter b is greater than* $\sum_{i=1}^{n} 4\frac{13}{16} + \frac{c_{i-1}}{4}$, *where* $c_0 = b$ *and* $c_{i+1} = \frac{9}{16} + \frac{c_i}{4}$.

Observation 2 *Let us notice that if* $c_{n+1} = \frac{9}{16} + \frac{c_n}{4}$ *for* $n > 0$, *then* $S_n = \sum_{i=1}^{n} 4\frac{13}{16} + \frac{c_{i-1}}{4} = (\frac{c_0}{3} - \frac{1}{4})(1 - \frac{1}{4^n}) + 5n$. *If we want to lower bound* $min\{S_n - 5n : n > 0\}$, *then we notice that, for* $c_0 \geq \frac{3}{4}$, $min\{S_n - 5n : n > 0\} = S_1 - 5 = -\frac{3}{16} + \frac{c_0}{4}$ *and for* $c_0 \leq \frac{3}{4}$, $min\{S_n - 5n : n > 0\} > \lim_{n\to\infty}(\frac{c_0}{3} - \frac{1}{4})(1 - \frac{1}{4^n}) = (\frac{c_0}{3} - \frac{1}{4})$

Also, $\lim_{n\to\infty} c_n = \frac{3}{4}$ *and for* $n > 0$, $c_n = \frac{3\cdot(4^n-1)}{4^{n+1}+\frac{c_0}{4^n}}$.

5.6 Non-clusters with outer parameters

Fact 24 *Every non-cluster* $2(\sqcup 1 \sqcup 2)^n (n > 0)$ *has weight greater than* $2 + 5n$.

Proof. The weight of a non-cluster of type $2 \sqcup 1 \sqcup 2$ is minimized, when the values of the variables constituting it are as drawn below:

$$\begin{array}{ccccc}
\frac{9}{8}-r & & & & \\
1 & r & & & \\
r & & & & \\
 & & b & & \\
 & b & 1 & b & \\
 & & b & & \\
 & & & & r \\
 & & & r & 1 \\
 & & & & \frac{9}{8}-r
\end{array}\quad \text{, with } r = \frac{9}{20} \text{ and } b = \frac{9}{40}.$$

It can also be proved (similarly as in Fact 23) that the weight of a non-cluster $2(\sqcup 1 \sqcup 2)^n$ for odd n is minimized, when the values of the variables it consists of form themselves into the regular pattern as follows. The middle subarray of type $2 \sqcup 1 \sqcup 2$ is the same as above and the "wings" are as in non-clusters of type $(\sqcup 1 \sqcup 2)^k$ with parameter $\frac{9}{8} - r = \frac{27}{40}$.

As for non-clusters $2(\sqcup 1 \sqcup 2)^n$ for even n, the situation is analogous, with the core of type $2 \sqcup 1 \sqcup 2 \sqcup 1 \sqcup 2$ having the structure:

$$\begin{array}{cccccc}
\ddots & a & & & & \\
a & 1 & a & & & \\
 & a & & & & \\
 & & & \frac{a}{4} & & \\
 & & \frac{a}{4} & 1 & \frac{a}{2} & \\
 & & & \frac{a}{2} & & \\
 & & & & & \frac{3}{4}a \\
 & & & & \frac{3}{4}a & 1 \\
 & & & & & \frac{5}{4}a
\end{array}\quad ,$$

with $a = \frac{9}{16}$ and with the left part of the core filled symmetrically.

Thus by Observation 2, the weight of a non-cluster $2(\sqcup 1 \sqcup 2)^{2k+1}$ is greater than $7\frac{1}{20} + 2(5k - \frac{1}{4} + \frac{1}{3} \cdot \frac{27}{40}) = 2 + (2k+1)5$ and the weight of a non-cluster $2(\sqcup 1 \sqcup 2)^{2k}$ is greater than $2\frac{1}{8} + 2(5k - \frac{1}{4} + \frac{1}{3} \cdot \frac{9}{16}) = 2 + 10k$.

□

Fact 25 *The non-cluster $2(\sqcup 1 \sqcup 2)^n$ ($n > 0$) with outer parameters b_1 and b_2 ($b_1 \leq b_2$) has weight greater than $2 + 5n + \frac{1}{2} - \frac{1}{3}(b_1 + b_2)$ if $\frac{b_1}{4^n} + b_2 \leq \frac{6*4^n+3}{2*4^{n+1}}$ and otherwise if $b_1 \leq \frac{3}{4}$ greater than $2\frac{1}{8} + \sum_{i=1}^{n}(4\frac{13}{16} + \frac{c_{i-1}}{4})$, with $c_0 = \frac{9}{8} - b_1$ and $c_{i+1} = \frac{9}{16} + \frac{c_i}{4}$. If $b_1 \geq \frac{3}{4}$, then its weight surpasses $2 + 5n$.*

Proof.

Let us first consider the non-cluster of type $2 \sqcup 1 \sqcup 2$ with outer parameters b_1 and b_2 ($b_1 \leq b_2$). Its structure can be represented as:

$$
\begin{array}{ccccccc}
 & x_1 & & & & & \\
b_1 & s_1 & x_3 & & & & \\
 & x_2 & & & & & \\
 & & & x_4 & & & \\
 & & x_5 & s_2 & x_7 & & \\
 & & & x_6 & & & \\
 & & & & & x_8 & \\
 & & & & x_9 & s_3 & b_2 \\
 & & & & & x_{10} &
\end{array}
$$

What we seek is, in a way, the solution to

$$
\begin{array}{lrl}
\text{minimize} & \sum_{i=1}^{3} s_i + \sum_{i=1}^{10} x_i & \\
\text{subject to} & & \\
 & b_1 + s_1 + x_3 & \geq 2\frac{1}{8} \\
 & s_1 & \leq 1 \\
 & s_1 + x_1 + x_3 & \geq 2\frac{1}{8} \\
 & s_1 + x_1 + x_2 & \geq 2\frac{1}{8} \\
 & s_1 + x_2 + x_3 + x_5 & \geq 2\frac{1}{8} \\
 & s_2 & \leq 1 \\
 & s_2 + x_3 + x_4 + x_5 + x_6 & \geq 2\frac{1}{8} \quad . \\
 & s_2 + x_3 + x_4 + x_5 + x_7 & \geq 2\frac{1}{8} \\
 & s_2 + x_4 + x_6 + x_7 + x_9 & \geq 2\frac{1}{8} \\
 & s_2 + x_5 + x_6 + x_7 + x_9 & \geq 2\frac{1}{8} \\
 & s_3 & \leq 1 \\
 & s_3 + x_7 + x_8 + x_9 & \geq 2\frac{1}{8} \\
 & s_3 + x_9 + x_{10} & \geq 2\frac{1}{8} \\
 & s_3 + x_8 + x_{10} & \geq 2\frac{1}{8} \\
 & s_3 + x_9 + b_2 & \geq 2\frac{1}{8}
\end{array}
$$

If $\frac{1}{3}b_1 + \frac{4}{3}b_2 \leq \frac{9}{8}$ (which since $b_1 \leq b_2$ implies also $\frac{4}{3}b_1 + \frac{1}{3}b_2 \leq \frac{9}{8}$), then the optimal solution of $(1) - (15)$ is:

$$
\begin{array}{ccccc}
b_1 & & & & \\
1 & \frac{9}{8} - b_1 & & & \\
\frac{9}{8} - b_1 & & & & \\
 & & \frac{2}{3}b_1 - \frac{1}{3}b_2 & & \\
 & \frac{2}{3}b_1 - \frac{1}{3}b_2 & 1 & -\frac{1}{3}b_1 + \frac{2}{3}b_2 & \\
 & & -\frac{1}{3}b_1 + \frac{2}{3}b_2 & & \\
 & & & & \frac{9}{8} - b_2 \\
 & & & \frac{9}{8} - b_2 & 1 \\
 & & & & b_2
\end{array} \quad .
$$

The subarray of type $\sqcup 1 \sqcup$ is as in the non-cluster with inner parameters $\frac{9}{8} - b_1$ and $\frac{9}{8} - b_2$. Let us notice, that the only constraints that could be violated by this solution are (5) and (12). Thus if we ignored them and calculated the minimal value of the sum $\sum_{i=1}^{3} s_i + \sum_{i=1}^{10} x_i$ subject to the remaining constraints, we would get, that the optimal solution is always as above. Taking advantage of this, we get that the lower bounds on the weight of the non-cluster $2 \sqcup 1 \sqcup 2$ with outer parameters b_1 and b_2 is $7\frac{1}{2} - \frac{b_1}{3} - \frac{b_2}{3}$. Using arguments very similar to those used in the proof of Fact 22, we get also that the weight of any non-cluster $2(\sqcup 1 \sqcup 2)^n$ is greater than $2 + 5n + \frac{1}{2} - \frac{b_1}{3} - \frac{b_2}{3}$.

If $\frac{1}{3}b_1 + \frac{4}{3}b_2 \geq \frac{9}{8}$ and additionally $b_1 \leq \frac{27}{40}$, then the optimal solution is

$$\begin{array}{ccccc}
b_1 & & & & \\
1 & \frac{9}{8} - b_1 & & & \\
\frac{9}{8} - b_1 & & & & \\
 & & -\frac{9}{32} + \frac{3}{4}b_1 & & \\
 & -\frac{9}{32} + \frac{3}{4}b_1 & 1 & \frac{9}{16} - \frac{b_1}{2} & \\
 & & \frac{9}{16} - \frac{b_1}{2} & & \\
 & & & & \frac{9}{32} + \frac{b_1}{4} \\
 & & & \frac{9}{32} + \frac{b_1}{4} & 1 \\
 & & & & \frac{27}{32} - \frac{b_1}{4}
\end{array} .$$

Again, if we ignored constraints (5) and (15), this would always be an optimal solution. If we consider a non-cluster $2(\sqcup 1 \sqcup 2)^n$ with outer parameters b_1 and b_2 and calculate its minimal weight subject to all constraints defining it except for $s_1 + x_2 + x_3 + x_5 \geq 2\frac{1}{8}$ (assuming that the variables in the beginning of the non-cluster are called as written above) and the one containing b_2, we will obtain, that it is equal to $2\frac{1}{8} + \sum_{i=1}^{n}(4\frac{13}{16} + \frac{c_{i-1}}{4})$, with $c_0 = \frac{9}{8} - b_1$ and $c_{i+1} = \frac{9}{16} + \frac{c_i}{4}$. One can observe, that the subarray of type $(\sqcup 1 \sqcup 2)^n$ is the same as in the non-cluster $(\sqcup 1 \sqcup 2)^n$ with an "inner" parameter $\frac{9}{8} - b_1$ of minimal weight. This estimation of the weight of a non-cluster $2(\sqcup 1 \sqcup 2)^n$ we will use, however, when the constraint containing b_2 is also satisfied. One can see that the constraint is satisfied if $\frac{9}{16} - \frac{c_{n-1}}{4} + b_2 \geq 2\frac{1}{8}$ (see the figure at the end of the proof of Fact 23), in other words if $\frac{9}{8} - c_n + b_2 \geq \frac{9}{8}$, that is if $b_2 \geq c_{n+1}$, which is equivalent to $\frac{b_1}{4^n} + b_2 \geq \frac{6*4^n+3}{2*4^{n+1}}$.

Finally, if $b_1, b_2 \geq \frac{3}{4}$, then the minimal weight of a non-cluster $2(\sqcup 1 \sqcup 2)^n$ with outer parameters b_1 and b_2 is equal to the minimal weight of a non-cluster $2(\sqcup 1 \sqcup 2)^n$ without parameters. It is so because the solution to the problem of minimizing the weight of a non-cluster $2(\sqcup 1 \sqcup 2)^n$ is also a feasible solution to our problem, if $b_1, b_2 \geq \frac{3}{4}$.

□

Fact 26 *The weight of a non-cluster* $(\sqcup 1 \sqcup 2)^n (n > 0)$ *with an inner and outer parameter* b_1 *and* b_2*, correspondingly is greater than* $5n + \frac{b_1}{3} - \frac{b_2}{3}$ *and if* $b_2 \geq \frac{3(4^n-1)}{4^{n+1}} + \frac{b_1}{4^n}$ *its minimal weight equals the minimal weight of a non-cluster* $(\sqcup 1 \sqcup 2)^n$ *with left parameter* b_1.

Proof. The weight of a non-cluster $\sqcup 1 \sqcup 2$ with an inner parameter b_1 and an outer parameter b_2 is not less than the minimal weight of non-cluster $\sqcup 1 \sqcup$ with parameters b_1 and $\frac{9}{8} - b_2$ plus $2\frac{1}{8}$ (the weight of a 2-column), which by Fact 21 equals to $5 + \frac{b_1}{3} - \frac{b_2}{3}$. Similarly, the minimal weight of a non-cluster $(\sqcup 1 \sqcup 2)^n$ with an inner parameter b_1 and an outer paramaeter b_2 is not less than the minimal weight of a non-cluster $\sqcup 1(\sqcup 2 \sqcup 1)^{n-1} \sqcup$ with parameters b_1 and $\frac{9}{8} - b_2$ plus $2\frac{1}{8}$, which by Fact 22 equals to exactly $5n + \frac{b1}{3} - \frac{b_2}{3}$.

Further, let us notice, that the inequalities describing a non-cluster $(\sqcup 1 \sqcup 2)^n$ with an inner and outer parameters b_1 and b_2 are the inequalities describing a non-cluster $(\sqcup 1 \sqcup 2)^n$ with a left parameter b_1 plus one inequality containing b_2. Thus the solution minimizing the weight of that type of a non-cluster minimizes also the weight of "our" type of a non-cluster, if it is a feasible solution to the system of all inequalities, that is if $b_2 \geq c_n$(see the proofs of Facts 23, 25and Observation 2) $= \frac{3(4^n-1)}{4^{n+1}} + \frac{b_1}{4^n}$, where $c_0 = b_1$ and $c_{n+1} = \frac{9}{16} + \frac{c_i}{4}$. □

Fact 27 *The weight of a non-cluster* $2(\sqcup 1 \sqcup 2)^n (n > 0)$ *with a left parameter* b_1 *is not less than the minimal weight of a non-cluster* $(\sqcup 1 \sqcup 2)^n$ *with a parameter* $\frac{9}{8} - b_1$ *plus* $2\frac{1}{8}$ *and if* $b_1 \geq \frac{3}{4}$ *its minimal weight equals the minimal weight of a non-cluster* $2(\sqcup 1 \sqcup 2)^n$ *without parameters. Therefore by Observation 2 its minimal weight minus* $2 + 5n$ *is greater than* $\frac{7}{32} - \frac{b_1}{4}$ *if* $b_1 \leq \frac{3}{8}$ *and greater than* $-\frac{b_1}{3} + \frac{1}{4}$ *if* $b_1 \geq \frac{3}{8}$.

5.7 Long non-clusters

Fact 28 *The minimal weight of a long non-cluster* $\sqcup 1 \sqcup 1 \sqcup$ *with parameters* b_1 *and* b_2 *is greater than* $4\frac{7}{10} + \frac{b_1+b_2}{5}$.

Proof. The structure of a long non-cluster $\sqcup 1 \sqcup 1 \sqcup$ with parameters b_1 and b_2 can be represented as:

$$\begin{array}{ccccc}
b_1 & & & & \\
 & & & & \\
 & x_1 & & & \\
x_2 & s_1 & x_4 & & \\
 & x_3 & & & \\
 & & & x_5 & \\
 & & x_6 & s_2 & x_8 \\
 & & & x_7 & \\
 & & & & \\
 & & & & b_2
\end{array}.$$

We calculate

$$\begin{array}{lrl}
\text{minimize} & s_1+s_2+b_1+b_2+\sum_{i=1}^{8} x_i & \\
\text{subject to} & s_1 & \le 1 \\
& x_1+x_2+x_3 & \ge 2\frac{1}{8}-b_1 \\
& x_1+x_2+x_4 & \ge 2\frac{1}{8}-b_1 \\
& x_1+x_3+x_4+x_6 & \ge 2\frac{1}{8} \\
& x_2+x_3+x_4+x_6 & \ge 2\frac{1}{8} \\
& s_2 & \le 1 \\
& x_4+x_5+x_6+x_7 & \ge 2\frac{1}{8} \\
& x_4+x_5+x_6+x_8 & \ge 2\frac{1}{8} \\
& x_5+x_7+x_8 & \ge 2\frac{1}{8}-b_2 \\
& x_6+x_7+x_8 & \ge 2\frac{1}{8}-b_2
\end{array}.$$

Whenever the "crosses" and parameters are placed in a different configuration, the minimal weight turns out to be the same (as long as the boundaries of the parts $\sqcup 1 \sqcup$ do not fall in the same level, of course). □

We also have the analogues of Facts 22 and 23:

Fact 29 *The weight of the long non-cluster* $\sqcup 1 \sqcup 1(\sqcup 2 \sqcup 1 \sqcup 1)^n \sqcup$ *is greater than* $4\frac{7}{10}+7n+\frac{b_1+b_2}{5}$.

Fact 30 *The minimal weight of the long non-cluster* $\sqcup 1 \sqcup 1 \sqcup 2$ *with left parameter* b_1 *is greater than* $6\frac{15}{16}+\frac{b_1}{6}$. *The minimal weight of the long non-cluster* $(\sqcup 1 \sqcup 1 \sqcup 2)^n$ $(n>0)$ *is greater than* $\sum_{i=1}^{n} 6\frac{15}{16}+\frac{e_{i-1}}{6}$, *with* $e_0=b_1$ *and* $e_{i+1}=\frac{9}{16}+\frac{e_i}{6}$.

Proof.

The optimal solution has the form:

$$\begin{array}{ccccccccc}
1 & e_i & & & & & & & \\
& & \frac{9}{16}-\frac{5}{6}e_i & & & & & & \\
& \frac{9}{16}-\frac{5}{6}e_i & 1 & \frac{2}{3}e_i & & & & & \\
& & \frac{2}{3}e_i & & & & & & \\
& & & & \frac{9}{16}-\frac{e_i}{2} & & & & \\
& & & \frac{9}{16}-\frac{e_i}{2} & 1 & \frac{e_i}{3} & & & \\
& & & & \frac{e_i}{3} & & & & \\
& & & & & & \frac{9}{16}-\frac{e_i}{6} & & \\
& & & & & \frac{9}{16}-\frac{e_i}{6} & 1 & \frac{9}{16}+\frac{e_i}{6}=e_{i+1} & \\
& & & & & & \frac{9}{16}+\frac{e_i}{6} & &
\end{array}.$$

□

Observation 3 *The minimal weight of the long non-cluster* $\sqcup 1 \sqcup 1 \sqcup 2$ *with left parameter* b_1 *is less than* 7 *only if* $b_1 < \frac{3}{8}$.

5.8 Clusters

First, we give an auxiliary definition and fact, which will be helpful in further proofs.

Definition 11 *The cluster with an inner parameter b_1 is the subarray of type $(\sqcup 2)^n (n > 0)$ having the structure:*

$$\begin{array}{cccccc} b_1 & & & & & \\ & x_1 & & & & \\ x_2 & s_1 & x_4 & & & \\ & x_3 & & & & \\ & & \ddots & & & \\ & & & & x_{4n-3} & \\ & & & x_{4n-2} & s_n & \\ & & & & x_{4n-1} & \end{array},$$

with the variables satisfying the usual inequalities:

$$\begin{array}{rl} s_1 & \leq 1 \\ s_1 + x_1 + x_2 & > 2\frac{1}{8} - b_1 \\ s_1 + x_2 + x_3 & > 2\frac{1}{8} \\ \ldots\ldots & \ldots\ldots \\ s_n + x_{4n-3} + x_{4n-1} & > 2\frac{1}{8} \end{array}.$$

Fact 31 *The minimal weight of the cluster $(\sqcup 2)^n$ with an inner parameter b_1 is greater than $\sum_{i=1}^{n}(2\frac{1}{8} + f_i) = \frac{13}{4}n + (b_1 - \frac{9}{8})(1 - \frac{1}{2^n})$, with $f_0 = b_1$ and $f_{i+1} = \frac{9}{16} + \frac{f_i}{2}$.*

Proof. By the way analogous to the one used in the proof of Fact 23 one can prove, that the weight of the cluster $(\sqcup 2)^n$ with an inner parameter b_1 is minimized, when it has the form:

$$\begin{array}{ccc} f_i & & \\ & \frac{9}{16} - \frac{f_i}{2} & \\ \frac{9}{16} - \frac{f_i}{2} & 1 & \frac{9}{16} + \frac{f_i}{2} = f_{i+1} \\ & \frac{9}{16} + \frac{f_i}{2} & \end{array}.$$

□

Corollary 3 *The weight of the cluster $(\sqcup 2)^n$ is greater than $2 + 3(n-1) + \frac{1}{2} + \frac{1}{8}$.*

Proof. Its minimal weight equals the minimal weight of the cluster $(\sqcup 2)^n$ with an inner parameter 0. □

Fact 32 *The weight of the cluster $2(\sqcup 2)^n$ without parameters is greater than $2 + 3n$.*

Proof. The cluster $2 \sqcup 2$ has minimal weight, when it has the form:

$$\begin{array}{ccc}
\frac{4}{3}a & & \\
1 & \frac{2}{3}a & \\
\frac{2}{3}a & & \\
 & & \frac{2}{3}a \\
 & \frac{2}{3}a & 1 \\
 & & \frac{4}{3}a
\end{array},$$

with $a = \frac{9}{16}$, and the cluster $2 \sqcup 2 \sqcup 2$ has minimal weight, when it has the following form:

$$\begin{array}{ccccc}
\frac{3}{2}a & & & & \\
1 & \frac{a}{2} & & & \\
\frac{a}{2} & & & & \\
 & & a & & \\
 & a & 1 & a & \\
 & & a & & \\
 & & & & \frac{a}{2} \\
 & & & \frac{a}{2} & 1 \\
 & & & & \frac{3}{2}a
\end{array}.$$

Longer clusters have these as their cores depending on whether n is even or odd and the "wings" are as in the clusters with an appropriate inner parameter ($\frac{3}{2}a$ or $\frac{4}{3}a$) of minimal weight.

$\square$

Fact 33 *The weight of the cluster $2 \sqcup 2$ with parameters b_1 and b_2 ($b_1 \leq b_2$) is greater than $6\frac{1}{2} - b_1 - b_2$. If $b_1 + 2b_2 \geq 2\frac{1}{4}$ and $b_1 \leq \frac{3}{4}$ its minimal weight equals the minimal weight of the cluster $\sqcup 2$ with an inner parameter $\frac{9}{8} - b_1$ plus $2\frac{1}{8}$, which equals to $5\frac{3}{8} - \frac{b_1}{2}$, and if $b_1 \geq \frac{3}{4}$ the minimal weight of the cluster $2 \sqcup 2$ equals to 5.*

As before, it turns out that thanks to the subseqent fact and also Fact 31 we can restrict ourselves to the clusters of type $2 \sqcup 2$ and forget the longer ones.

Fact 34 *The weight of the cluster $2(\sqcup 2)^n$ with parameters b_1 and b_2 ($b_1 \leq b_2$) is greater than $\frac{13}{4}(n+1) - b_1 - b_2$. If $b_1 + 2b_2 \geq 2\frac{1}{4}$ and $b_1 \leq \frac{3}{4}$ its minimal weight is greater than $5\frac{3}{8} - \frac{b_1}{2} + 3(n-1)$. If $b_1 \geq \frac{3}{4}$, the minimal weight is greater than $2 + 3n$.*

Proof. To prove the first statement let us notice that every "cross" in the cluster has weight not less than $3\frac{1}{4}$. Since we do not include the weights of parameters, the result follows. To prove the second part, it suffices to calculate the minimal weight of the cluster $(\sqcup 2)^n$ with an inner parameter $\frac{9}{8} - b_1$, add $2\frac{1}{8}$ to it and compare with $5\frac{3}{8} - \frac{b_1}{2} + 3(n-1)$ (not forgetting the assumption about b_1). $\square$

And the final fact in the series, which is the corollary of Facts 31 and 34:

Fact 35 *The minimal weight of the cluster $(\sqcup 2)^n$ with right (outer) parameter b_1 equals the minimal weight of the cluster $(\sqcup 2)^n$ if $b_1 \geq \frac{9}{8}(1 - \frac{1}{2^n})$ and is greater than $\frac{13}{4}n - b_1$.*

5.9 Summing up

Let us sum up our knowledge about clusters, non-clusters and long non-clusters. Instead of just giving their minimal weight, it will be more convenient for us to give their minimal weight minus $N(T)+1$, where T is the type of a respective cluster, non-cluster or long non-cluster. For clusters of type $T = (\sqcup 2)^n$ we will give their minimal weight minus $N(T)$.

The functions in the table below should be interpreted in the following way. $f(b_1, b_2) = v$ (or correspondingly $f(b_1) = v$) if we have calculated that the minimal weight of a corresponding subarray of type T minus $N(T)+1$ (minus $N(T)$ in case of $(\sqcup 2)^n$) with parameter(s) b_1 and b_2 is not less than v.

function	corresponding to	is equal to	if
$c(b1)$	$(\sqcup 2)^n\ (b_1)$	$\frac{1}{4}n + 1 - \frac{9}{8}(1 - \frac{1}{2^n})$ $\frac{1}{4}n + 1 - b_1$	$b_1 \geq \frac{9}{8}(1 - \frac{1}{2^n})$ $b_1 \leq \frac{9}{8}(1 - \frac{1}{2^n})$
$c2(b_1, b_2)$	$(b_1)\ 2(\sqcup 2)^n\ (b_2)\ (b_1 \leq b_2)$	$\frac{3}{2} - b_1 - b_2$ $\frac{3}{8} - \frac{b_1}{2}$ 0	$b_1 + 2b_2 \leq \frac{9}{4}$ $b_1 + 2b_2 \geq \frac{9}{4}$ $b_1 \leq \frac{3}{4}$ $b_1 \geq \frac{3}{4}$
$c1(b_1)$	$2(\sqcup 2)^n\ (b_1)$	$\frac{3}{8} - \frac{b_1}{2}$ 0	$b_1 \leq \frac{3}{4}$ $b_1 \geq \frac{3}{4}$
$n2(b_1, b_2)$	$(b_1)\ \sqcup 1(\sqcup 2 \sqcup 1)^n \sqcup\ (b_2)$	$-\frac{1}{2} + \frac{b_1}{3} + \frac{b_2}{3}$	
$nb2(b_1, b_2)$	$(b_1)\ 2(\sqcup 1 \sqcup 2)^n\ (b_2)\ (b_1 \leq b_2)$	$\frac{1}{2} - \frac{b_1}{3} - \frac{b_2}{3}$ $\frac{1}{8} + (\frac{\frac{9}{8} - b_1}{3} - \frac{1}{4})(1 - \frac{1}{4^n})$ 0	$\frac{b_1}{4^n} + b_2 \leq \frac{3(2*4^n+1)}{2*4^{n+1}}$ $\frac{b_1}{4^n} + b_2 \geq \frac{3(2*4^n+1)}{2*4^{n+1}}$ $b_1 \leq \frac{3}{4}$ $b_1 \geq \frac{3}{4}$
$n1(b_1)$	$(b_1)\ (\sqcup 1 \sqcup 2)^n$	$-\frac{1}{4} + \frac{b_1}{3}$ $-\frac{3}{16} + \frac{b_1}{4}$	$b_1 \leq \frac{3}{4}$ $b_1 \geq \frac{3}{4}$
$np2(b_1, b_2)$	$(b_1)\ (\sqcup 1 \sqcup 2)^n\ (b_2)$	$(\frac{b_1}{3} - \frac{1}{4})(1 - \frac{1}{4^n})$ $\frac{b_1}{3} - \frac{b_2}{3}$	$b_2 \geq \frac{3}{4} + \frac{1}{4^n}(b_1 - \frac{3}{4})$ $b_2 \leq \frac{3}{4} + \frac{1}{4^n}(b_1 - \frac{3}{4})$
$nb1(b_1)$	$(b_1)\ 2(\sqcup 1 \sqcup 2)^n$	$\frac{1}{4} - \frac{b_1}{3}$ $\frac{7}{32} - \frac{b_1}{4}$ 0	$\frac{3}{4} \geq b_1 \geq \frac{3}{8}$ $b_1 \leq \frac{3}{8}$ $b_1 \geq \frac{3}{4}$
$l2(b_1, b_2)$	$(b_1)\ \sqcup 1 \sqcup 1(\sqcup 2 \sqcup 1 \sqcup 1)^n \sqcup\ (b_2)$	$-\frac{3}{10} + \frac{b_1}{5} + \frac{b_2}{5}$	
$l1(b_1)$	$(b_1)\ (\sqcup 1 \sqcup 1 \sqcup 2)^n$	$\frac{-1}{16} + \frac{b_1}{6}$ 0	$b_1 \leq \frac{3}{8}$ $b_1 \geq \frac{3}{8}$

Since we want to prove that the weight of every difficult subarray of type T_1 is not less than $N(T_1) + \frac{5}{8}$ and not less than $N(T_2) + 1$ in case of subarrays of type T_2 and since $N(T_1 T_{2k}) + 1 = N(T_1) + 1 + N(T_2) + 1 + \ldots + N(T_k) + 1$, this task amounts to showing that the appropriate sum of functions above is at least zero or at least $\frac{5}{8}$.

More formally we want to show that if $f_1(b_1) = c(b_1)$,

$$\min_{b_1,b_2,\ldots,b_{k-1}} \{f_1(b_1) + \sum_{i=2}^{k-1} f_i(b_{i-1}, b_i) + f_k(b_k - 1)\} \geq \frac{5}{8}$$

and otherwise that

$$\min_{b_1,b_2,\ldots,b_{k-1}} \{f_1(b_1) + \sum_{i=2}^{k-1} f_i(b_{i-1}, b_i) + f_k(b_k - 1)\} \geq 0.$$

Additionally, functions f_i are functions from the table above. They correspond to subparts of a subarray of type T_1 and T_2 and therefore must occur in a sensible order, i.e. for example a function c can be followed only by functions $n1, n2, np2, l1, l2$ and a function l_2 by functions $c1, c2, nb1, nb2$.

Let us replace functions $n2$ and $l2$ with the sums of the following functions.

Let $n'(b_1) = -\frac{1}{4} + \frac{b_1}{3}$ and $l'(b_1) = -\frac{3}{20} + \frac{b_1}{5}$. Then

$$n2(b_1, b_2) = n'(b_1) + n'(b_2)$$

and

$$l2(b_1, b_2) = l'(b_1) + l'(b_2).$$

After the replacement the original arbitrarily long sums will disintegrate into smaller sums consisting of at most four components. Let g_1 denote any one of the functions $n1, l1, n', l'$ and g_2 any one of the functions $n1, n', l', nb1$. It suffices to prove the following.

- $\min_{b_1}\{c(b_1) + g_1(b_1)\} \geq \frac{5}{8}$
- $\min_{b_1,b_2}\{c(b_1) + np2(b_1, b_2) + l1(b_2)\} \geq \frac{5}{8}$
- $\min_{b_1,b_2}\{c(b_1) + np2(b_1, b_2) + l'(b_2)\} \geq \frac{5}{8}$
- $\min_{b_1}\{c1(b_1) + g_1(b_1)\} \geq 0$
- $\min_{b_1,b_2}\{c1(b_1) + np2(b_1, b_2) + l1(b_2)\} \geq 0$
- $\min_{b_1,b_2}\{c1(b_1) + np2(b_1, b_2) + l'(b_2)\} \geq 0$
- $\min_{b_1}\{nb1(b_1) + l1(b_1)\} \geq 0$
- $\min_{b_1}\{nb1(b_1) + l'(b_1)\} \geq 0$
- $\min_{b_1,b_2}\{l'(b_1) + nb2(b_1, b_2) + l1(b_2)\} \geq 0$
- $\min_{b_1,b_2}\{l'(b_1) + nb2(b_1, b_2) + l'(b_2)\} \geq 0$
- $\min_{b_1,b_2}\{g_2(b_1) + c2(b_1, b_2) + g_1(b_2)\} \geq 0$

- $\min_{b_1,b_2,b_3}\{g_2(b_1)+c2(b_1,b_2)+np2(b_2,b_3)+l1(b_3)\}\geq 0$
- $\min_{b_1,b_2,b_3}\{g_2(b_1)+c2(b_1,b_2)+np2(b_2,b_3)+l'(b_3)\}\geq 0$

First we prove the following fact.

Fact 36 *If $\alpha_2 \leq \frac{1}{3}$ and $\frac{1}{3}-\alpha_1+\frac{3\alpha_2-1}{3\cdot 4^n}\geq 0$, then*

$$\min_{b_1,b_2}\{nb2(b_1,b_2)+\alpha_1 * b_1+\alpha_2 * b_2\}\geq \frac{3}{4}(\alpha_1+\alpha_2)$$

Proof. Let us assume that $b_1 \leq b_2$.

First we prove the fact for the case when $\frac{b_1}{4^n}+b_2 \leq \frac{3(2\cdot 4^n+1)}{2\cdot 4^{n+1}}$. We can notice, that the conditions imply that $b_1 \leq \frac{3}{4}$.

Then
$nb2(b_1,b_2)+\alpha_1 * b_1+\alpha_2 * b_2 = \frac{1}{2}-\frac{b_1}{3}-\frac{b_2}{3}+\alpha_1 b_1+\alpha_2 b_2 = \frac{1}{2}-b_1(\frac{1}{3}-\alpha_1)-b_2(\frac{1}{3}-\alpha_2) = \frac{1}{2}-b_1(\frac{1}{3}-\alpha_1)-(\frac{b_1}{4^n}+b_2)(\frac{1}{3}-\alpha_2)+\frac{b_1}{4^n}(\frac{1}{3}-\alpha_2) = \frac{1}{2}-b_1(\frac{1}{3}-\alpha_1+\frac{3\alpha_2-1}{3\cdot 4^n})-(\frac{b_1}{4^n}+b_2)(\frac{1}{3}-\alpha_2)\geq \frac{1}{2}-\frac{3}{4}(\frac{1}{3}-\alpha_1+\frac{3\alpha_2-1}{3\cdot 4^n})-\frac{3(2\cdot 4^n+1)}{2\cdot 4^{n+1}}(\frac{1}{3}-\alpha_2) = \frac{1-3\alpha_2}{2\cdot 4^{n+1}}+\frac{3}{4}(\alpha_1+\alpha_2)\geq \frac{3}{4}(\alpha_1+\alpha_2)$

Next we prove the fact for the case when $\frac{b_1}{4^n}+b_2 \geq \frac{3(2\cdot 4^n+1)}{2\cdot 4^n+1}$ and $b_1 \leq \frac{3}{4}$.

Then
$nb2(b_1,b_2)+\alpha_1 * b_1+\alpha_2 * b_2 = \frac{1}{8}+(\frac{\frac{9}{8}-b_1}{3}-\frac{1}{4})(1-\frac{1}{4^n})+\alpha_1 b_1+\alpha_2 b_2 = \frac{1}{4}-\frac{1}{8\cdot 4^n}-\frac{b_1}{3}+\frac{b_1}{3\cdot 4^n}+\alpha_1 b_1+\alpha_2 b_2 = \frac{1}{4}-\frac{1}{8\cdot 4^n}-\frac{b_1}{3}+\frac{b_1}{3\cdot 4^n}+\alpha_1 b_1+\alpha_2 b_2+\alpha_2\frac{b_1}{4^n}-\alpha_2\frac{b_1}{4^n} = \frac{1}{4}-\frac{1}{8\cdot 4^n}-b_1(\frac{1}{3}-\alpha_1+\frac{3\alpha_2-1}{3\cdot 4^n})+\alpha_2(\frac{b_1}{4^n}+b_2)\geq \frac{1-3\alpha_2}{2\cdot 4^{n+1}}+\frac{3}{4}(\alpha_1+\alpha_2)\geq \frac{3}{4}(\alpha_1+\alpha_2)$

The last case is $b_1 \geq \frac{3}{4}$, which implies that also $b_2 \geq \frac{3}{4}$. Then
$nb2(b_1,b_2)+\alpha_1 * b_1+\alpha_2 * b_2 = \alpha_1 * b_1+\alpha_2 * b_2 \geq \frac{3}{4}(\alpha_1+\alpha_2)$

□

The corollary of this fact is

Corollary 4

$$\min_{b_1,b_2}\{l'(b_1)+nb2(b_1,b_2)+l1(b_2)\}\geq 0$$

$$\min_{b_1,b_2}\{l'(b_1)+nb2(b_1,b_2)+l'(b_2)\}\geq 0$$

Next in a similar way we prove

Fact 37 *If $\alpha_1 \leq \frac{1}{3}$, then*

$$\min_{b_1}\{nb1(b_1)+\alpha_1 * b_1\}\geq \frac{3}{4}\alpha_1$$

The corollary of this fact is

Corollary 5

$$\min_{b_1}\{nb1(b_1) + l1(b_1)\} \geq 0$$

$$\min_{b_1}\{nb1(b_1) + l'(b_1)\} \geq 0$$

Finally we prove

Fact 38 • – *For* $b_1 \leq \frac{3}{4}$

$$\min_{b_2}\{np2(b_1, b_2) + n'(b_2)\} \geq \frac{b1}{3} - \frac{1}{4}$$

– *For* $b_1 \geq \frac{3}{4}\min_{b_2}\{np2(b_1, b_2) + n'(b_2)\} \geq \frac{3b1}{10} - \frac{9}{40}$

- *For* $\alpha_1 \leq \frac{1}{2}$, $\min_{b_1}\{c1(b_1) + \alpha_1 b_1\} \geq \frac{3}{4}\alpha_1$
- *For* $\alpha_1 \leq \frac{1}{2}$ *and* $2\alpha_1 - \alpha_2 \leq 1$,

$$\min_{b_1,b_2}\{c2(b_1, b_2) + \alpha_1 b_1 + \alpha_2(b_2)\} \geq \frac{3}{4}\alpha_1$$

- *For* $\alpha_1 \geq \frac{1}{9}$,

$$\min_{b_1}\{c(b_1) + \alpha_1 b_1\} \geq \frac{11}{16} + \frac{9}{16}\alpha_1$$

As a corollary we get

Corollary 6 • $\min_{b_1}\{c(b_1) + g_1(b_1)\} \geq \frac{5}{8}$

- $\min_{b_1,b_2}\{c(b_1) + np2(b_1, b_2) + l1(b_2)\} \geq \frac{5}{8}$
- $\min_{b_1,b_2}\{c(b_1) + np2(b_1, b_2) + l'(b_2)\} \geq \frac{5}{8}$
- $\min_{b_1}\{c1(b_1) + g_1(b_1)\} \geq 0$
- $\min_{b_1,b_2}\{c1(b_1) + np2(b_1, b_2) + l1(b_2)\} \geq 0$
- $\min_{b_1,b_2}\{c1(b_1) + np2(b_1, b_2) + l'(b_2)\} \geq 0$
- $\min_{b_1,b_2}\{g_2(b_1) + c2(b_1, b_2) + g_1(b_2)\} \geq 0$
- $\min_{b_1,b_2,b_3}\{g_2(b_1) + c2(b_1, b_2) + np2(b_2, b_3) + l1(b_3)\} \geq 0$
- $\min_{b_1,b_2,b_3}\{g_2(b_1) + c2(b_1, b_2) + np2(b_2, b_3) + l'(b_3)\} \geq 0$

This way we have proved Lemma 9 for subarrays that do not contain a 3-column. In Section 5.10 we deal with 3-columns.

5.10 Appendix to the proof of Lemma 9

First let us notice, that if any one of the subarrays containing $\sqcup 2$ or $2\sqcup$ can be well-tiled or if a subarray of type $\sqcup 3$, $3\sqcup$, $\sqcup 3\sqcup$ can be well-tiled, then the whole subarray (of type appropriately $\sqcup 2 \sqcup 3$ or $\sqcup 2 \sqcup 3 \sqcup 2$) can be well-tiled.

To prove the lemma, we will successively show, that if in the degenerated subarray some collection of subarrays cannot be well-tiled, then the whole subarray has weight greater than $7\frac{3}{8}$ in the case of $\sqcup 2 \sqcup 3$ or in the case of $\sqcup 2 \sqcup 3 \sqcup 2$ that the whole subarray has weight greater than $10\frac{3}{8}$ or that one of its subarrays of type $\sqcup 2 \sqcup 3$ or $\sqcup 3 \sqcup 2$ has weight greater than 7

Let us then look at the subarray of type $\sqcup 3$ that cannot be well-tiled. The obsevation we can make is that the upper line and the lower line cannot fall in the same row, as in such a case we would be easily able to $2\frac{1}{8}$-partition the subarray $\sqcup 3$ into 3 tiles horizontally.

The sign"x" denoting the boundary of a 3-column, i.e. that element which unables the $2\frac{1}{8}$-partition of the column into 2 tiles can fall in any row between the upper line and the lower line, with these two rows including.

The structure of such $\sqcup 3$ is presented in Figure 5.2. In the figure all ovals indicate areas of weight greater than $2\frac{1}{8}$. The area described by the middle oval must also have weight greater than $2\frac{1}{8}$, as otherwise we could $2\frac{1}{8}$-partition the subarray $\sqcup 3$ into 3 tiles horizontally.

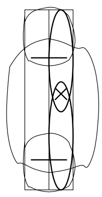

Figure 5.2:

Next let us look at $\sqcup 2 \sqcup 3$.

We can notice that if "x" falls below the upper line, then the weight of the subarray $\sqcup 2 \sqcup 3$ is greater than $3 + \frac{2}{8} + 4 + \frac{2}{8}$, as shown in Figure 5.3. Let us then further assume that "x" falls in the same row as the upper line.

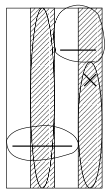

Figure 5.3:

Further, observe that, if the upper line of the subarray $\sqcup 3$ is above the boundaries of $\sqcup 2$ and $2\sqcup$ and the lower line below them, then the weight of $\sqcup 2 \sqcup 3$ is greater than $7 + \frac{4}{8}$, as shown in Figure **??**.

Let us now consider the situation, when the lower line of the subarray $\sqcup 3$ falls in the same row as the boundaries of $\sqcup 2$ and $2\sqcup$. (The case when the upper line falls in the same row as the boundaries is symmetric.)

If region B in Figure 5.4 has weight greater than $2\frac{1}{8}$, then the subarray $\sqcup 2 \sqcup 3$ has weight greater than $7 + \frac{4}{8}$, which is also the case if both regions E and F have weights greater than $2\frac{1}{8}$. Otherwise, that is, if B has weight at most $2\frac{1}{8}$ and E or F has weight at most $2\frac{1}{8}$, we are able to $2\frac{1}{8}$-pa the subarray $\sqcup 2 \sqcup 3$ into 6 tiles as shown in Figure 5.4(c).

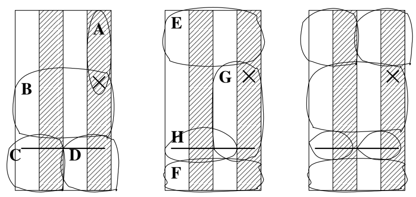

Figure 5.4:

Thus the structure of the subarray $\sqcup 2 \sqcup 3$ that cannot be well-tiled and has weight less than $7 + \frac{3}{8}$ must have the form shown in Figure 5.5 (or the symmetric one with both lines of $\sqcup 3$ below the boundaries of $\sqcup 2$ and $2\sqcup$).

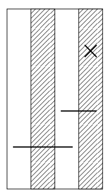

Figure 5.5:

We will now focus on the subarray of type $\sqcup 3 \sqcup$. We will show that in many cases it either can be $2\frac{1}{8}$-partitioned into 4 tiles or has weight greater than $5+\frac{2}{8}$. If the subarray $\sqcup 3 \sqcup$ has weight greater than $5 + \frac{2}{8}$, then the subarray $\sqcup 3 \sqcup 2$ that cannot be well-tiled has weight greater than $7\frac{3}{8}$, as the 2-column must have weight greater than $2\frac{1}{8}$.

Let us look at Figure 5.6. We also draw an upper and a lower line in the part denoting $3\sqcup$. First we consider four cases when the upper line of the subarray $3\sqcup$ is over the sign "x". If the lower line of $3\sqcup$ falls below the lower line of $\sqcup 3$, then it follows that the weight of $\sqcup 3 \sqcup$ surpasses $6 + \frac{3}{8}$. If the lower line of $3\sqcup$ falls in the same row as the lower line of $\sqcup 3$, the weight of $\sqcup 3 \sqcup$ surpasses $6\frac{3}{8} - 1$. The case when the lower line of $3\sqcup$ is above the lower line of $\sqcup 3$, but below the sign"x" will be checked later using linear programming and if the lower line of $3\sqcup$ falls in the same row as the sign "x", then the subarray $\sqcup 2 \sqcup 3 \sqcup 2$ is not degenerated in some cases.

Next, three cases, when the upper line of $3\sqcup$ is in the same row, as the sign ”x”. If the lower line of $3\sqcup$ falls below or above the lower line of $\sqcup 3$, then the weight of $\sqcup 3\sqcup$ surpasses $5\frac{3}{8}$. If the lower lines of $\sqcup 3$ and $3\sqcup$ fall in the same row, then if any one of the regions A,B or C drawn in the last row of Figure 5.6 has weight greater than $2\frac{1}{8}$, then the weight of $\sqcup 3\sqcup$ is greater than $5\frac{3}{8}$. If neither of A,B and C has weight surpassing $2\frac{1}{8}$, then if the row containing the lower lines of $\sqcup 3$ and $3\sqcup$ had weight at most $2\frac{1}{8}$, we could $2\frac{1}{8}$-partition $\sqcup 3\sqcup$ into 4 tiles horizontally and thus would be able to well-tile $\sqcup 2 \sqcup 3 \sqcup 2$. Therefore we further assume, that in such a case the row containing the lower lines has weight greater than $2\frac{1}{8}$.

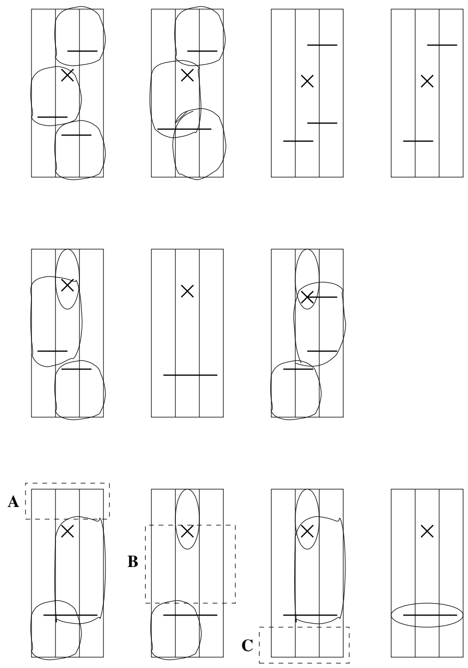

Figure 5.6:

Let us now return to the cases when the upper line of $3\sqcup$ is above the sign ”x” and the lower line of $3\sqcup$ is above the lower line of $\sqcup 3$.

It can be represented as shown in Figure 5.7.

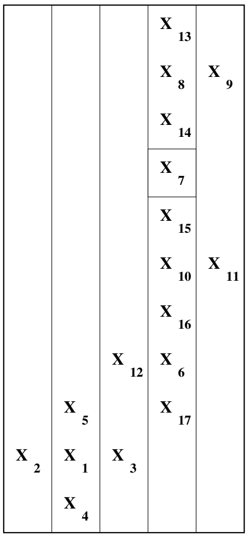

Figure 5.7:

We want to calculate

$$
\begin{array}{lrl}
\text{minimize} & \sum_{i=1}^{16} x_i & \\
\text{subject to} & x_1 + x_2 + x_5 & \geq 2\frac{1}{8} \\
& x_1 + x_2 + x_4 & \geq 2\frac{1}{8} \\
& x_1 + x_3 + x_4 & \geq 2\frac{1}{8} \\
& x_1 + x_3 + x_5 + x_{12} & \geq 2\frac{1}{8} \\
& x_1 + x_4 + x_5 & \geq 2\frac{1}{8} \\
& x_1 + x_2 + x_3 & \geq 2\frac{1}{8} \\
& x_2 & < 1 \\
& x_1 & \leq 1 \\
& x_1 + x_4 + x_5 & < 3 \\
& x_1 + x_2 + x_4 + x_5 & < 3\frac{3}{8} \\
& x_7 + x_8 + x_{13} + x_{14} & \geq 2\frac{1}{8} \\
& x_6 + x_7 + x_{10} + x_{15} + x_{16} + x_{17} & \geq 2\frac{1}{8} \\
& x_6 + x_7 + x_{10} + x_{12} + x_{15} + x_{16} & \geq 2\frac{1}{8} \\
& x_3 + x_6 + x_{12} + x_{17} & \geq 2\frac{1}{8} \\
& x_8 + x_9 + x_{13} & \geq 2\frac{1}{8} \\
& x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{14} + x_{15} & \geq 2\frac{1}{8} \\
& x_6 + x_{10} + x_{11} + x_{16} + x_{17} & \geq 2\frac{1}{8} \\
& x_6,\ x_7,\ x_8,\ x_{10} & \leq 1 \\
& x_3 + x_1 2 & < 1 \\
& x_9 + x_{11} & < 1 \\
& x_6 + x_7 + x_8 + x_{10} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} & < 4 \\
& x_3 + x_6 + x_7 + x_8 + x_{10} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} & < 4\frac{3}{8}
\end{array}
$$

It turns out that the system is infeasible.

Let us now return to the situation, when $\sqcup 3 \sqcup$ has the structure as the last one in Figure 5.6 and neither of A, B or C has weight greater than $2\frac{1}{8}$. If the boundary of the second complex $\sqcup 2$ falls in the same row as the lower lines of $\sqcup 3 \sqcup$ then, if A from Figure 5.8 has weight greater than $2\frac{1}{8}$, then the weight of $\sqcup 3 \sqcup 2$ is greater than $7\frac{4}{8}$.

Otherwise we are able to $2\frac{1}{8}$-partition $\sqcup 3 \sqcup 2$ into 6 tiles as shown in the second drawing of Figure 5.8.

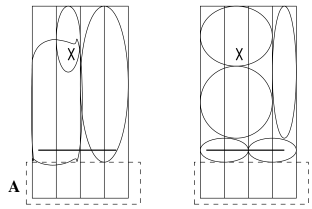

Figure 5.8:

If the boundary of the second complex $\sqcup 2$ falls in the same row as the sign "x", then we are either able to $2\frac{1}{8}$-partition $3 \sqcup 2$ into 5 tiles as shown in Figure 5.9 or the weight of $\sqcup 3 \sqcup 2$ surpasses $7\frac{4}{8}$. It is so, because if A has weight greater than $2\frac{1}{8}$, then the weight of the second complex $\sqcup 2$ has weight greater than $3\frac{2}{8}$ and the weight of $\sqcup 3$ is greater than $4\frac{2}{8}$.

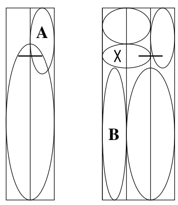

Figure 5.9:

If the boundary of the second complex is between the sign "x" and the lower lines, as shown in Figure 5.10, then we tackle it with linear programming.

We make also two observations. One is, that if region A in Figure 5.11 has weight greater than $2\frac{1}{8}$m then the subarray $\sqcup 2 \sqcup 3$ has weight greater than $7\frac{4}{8}$. Therefore, we assume this is not the case. The second is that if region B has weight at most $2\frac{1}{8}$, then we are able to well-tile the part $3 \sqcup 2$. Thus, further we will assume that B has weight greater than $2\frac{1}{8}$.

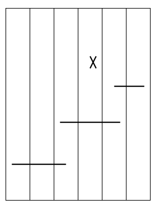

Figure 5.10:

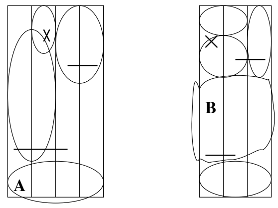

Figure 5.11:

Our case can be represented as in Figure 5.12. We want to calculate the minimal weight of $\sqcup 2 \sqcup 3 \sqcup 2$ in such a situation. To this end we calculate the minimal value of $\sum_{i=1}^{16} x_i$ subject to the constraints describing the type. For example, we bound the variables in a 3-column to have weight less than 4 and the variables of $<$-columns to have weight less than 1. Also we want the parts $\sqcup 2 \sqcup 3$ and $\sqcup 3 \sqcup 2$ to have weight less than $7\frac{3}{8}$, the complexes to have weight less than $3\frac{3}{8}$ and the subarray $\sqcup 3$ to have weight less than $4\frac{3}{8}$. Besides to these we also add the following constraints:

$$
\begin{array}{ll}
x_1 + x_2 + x_4 & > 2\frac{1}{8} \\
x_1 + x_3 + x_4 & > 2\frac{1}{8} \\
x_1 + x_2 + x_5 & > 2\frac{1}{8} \\
x_1 + x_3 + x_5 + x_9 & > 2\frac{1}{8} \\
x_1 + x_4 + x_5 & > 2\frac{1}{8} \\
x_1 + x_2 + x_3 & > 2\frac{1}{8} \\
x_6 + x_7 & > 2\frac{1}{8} \\
x_6 + x_8 + x_{11} + x_{12} & > 2\frac{1}{8} \\
x_6 + x_8 + x_9 + x_{11} & > 2\frac{1}{8} \\
x_3 + x_8 + x_9 + x_{12} & > 2\frac{1}{8} \\
x_6 + x_8 + x_{10} + x_{11} + x_{15} & > 2\frac{1}{8} \\
x_8 + x_{10} + x_{12} & > 2\frac{1}{8} \\
x_8 + x_{10} + x_{11} + x_{16} & > 2\frac{1}{8} \\
x_{13} + x_{14} + x_{15} & > 2\frac{1}{8} \\
x_{10} + x_{13} + x_{15} + x_{16} & > 2\frac{1}{8} \\
x_{13} + x_{14} + x_{16} & > 2\frac{1}{8}
\end{array}
$$

The system however turns out to be infeasible.

Let us now examine subarrays of type $\sqcup 2 \sqcup 1 \sqcup 1 \sqcup 2$ with the boundaries of two subarrays of type $\sqcup 1 \sqcup$ falling in the same row.

Let us first focus attention on the subarray S of type $T = \sqcup 1 \sqcup 1 \sqcup$. If A or B as shown in Figure has weight greater than $2\frac{1}{8}$, then S has weight greater than $N(T) + 1\frac{3}{8}$.

Therefore we will further assume that both A and B have weight at most $2\frac{1}{8}$. We can notice that if C or D had weight at most $2\frac{1}{8}$, we could well-tile S. C and D contain three array elements constituting a boundary of two subarrays of type $\sqcup 1 \sqcup$.

Concluding, we get that clear subarrays of type $\sqcup 1 \sqcup 1 \sqcup$ and $\sqcup 3 \sqcup$ can be described in following way.

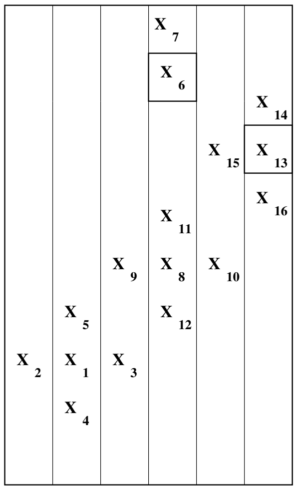

Figure 5.12:

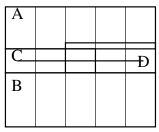

Figure 5.13:

		b_2
	x_3	
	s_2	x_4
	w	
x_1	s_1	
	x_2	
b_1		

$$
\begin{array}{rll}
f(b_1, b_2) = & \text{minimize} & b_1 + b_2 + s_1 + s_2 + w + x_1 + x_2 + x_3 + x_4 \\
& \text{subject to} & s_1 \leq 1 \\
& & s_1 + x_1 + x_2 + b_1 \geq 2\frac{1}{8} \\
& & w + s_1 + x_2 \geq f \\
& & w + s_1 + x_1 \geq 2\frac{1}{8} \\
& & w + s_2 + x_3 \geq 2\frac{1}{8} \\
& & w + s_2 + x_4 \geq 2\frac{1}{8} \\
& & s_2 + x_3 + x_4 \geq 2\frac{1}{8}.
\end{array}
$$

We get that $f(b_1, b_2) = 4\frac{13}{16} + \frac{b_1}{2} = 5 - 316 + \frac{b_1}{2}$. Since $f(b_1, b_2) > l2(b_1, b_2)$ we do not have to include such subarrays to the analysis of the proof of Lemma 9.

Bibliography

[1] P. Berman, B. DasGupta, S. Muthukrishnan, S. Ramaswami, *Efficient Approximation Algorithms for Tiling and Packing Problems with Rectangles*, J. Algorithms 41(2): 443-470 (2001).

[2] M. Charikar, C. Chekuri, R. Motwani, Unpublished.

[3] M. Grigni, F. Manne, *On the complexity of generalized block distribution*, Proc. 3rd intern. workshop on parallel algorithms for irregularly structured problems (IRREGULAR'96), Springer, 1996, LNCS 1117, 319-326.

[4] Y. Han, B. Narahari, H.-A. Choi, *Mapping a chain task to chained processors*, Information Processing Letters 44, 141-148, 1992.

[5] S. Khanna, S. Muthukrishnan, M. Paterson, *On approximating rectangle tiling and packing*, Proc. 19th SODA (1998), 384–393.

[6] S. Khanna, S. Muthukrishnan, S. Skiena, *Efficient array partitioning,* Proc. 24th ICALP, 616-626, 1997.

[7] K. Loryś, K. Paluch, *Rectangle Tiling*, Proc. 3rd APPROX, Springer, 2000, LNCS 1923, 206-213.

[8] K. Loryś, K. Paluch, *A new approximation algorithm for RTILE problem*, Theor. Comput. Sci. 2-3(303): 517-537 (2003).

[9] G. Martin, S. Muthukrishnan, R. Packwood, I. Rhee,*Fast algorithms for variable size block matching motion estimation with minimal error*.

[10] S. Muthukrishnan, V. Poosala, T. Suel, *On rectangular partitioning in two dimensions: algorithms, complexity, and applications*, Proc. 7th ICDT, 1999, 236-256.

[11] K. Paluch, *A* $2\frac{1}{8}$*-approximation algorithm for rectangle tiling*, to appear in ICALP 2004,

[12] J. Sharp, *Tiling Multi-dimensional Arrays*, Proc. 12th FCT, Springer, 1999, LNCS 1684, 500-511.

[13] A. Smith, S. Suri, *Rectangular Tiling in Multidimensional Arrays*, J.Algorithms 37(2): 451-467 (2000).