Article:
Brucker, P and Shakhlevich, NV (2011) A polynomial-time algorithm for a flow-shop batching problem with equal-length operations. Journal of Scheduling, 14 (4). 371 - 389. ISSN 1094-6136

https://doi.org/10.1007/s10951-009-0150-8

Noname manuscript No.
(will be inserted by the editor)

A Polynomial-Time Algorithm for a Flow-Shop Batching Problem with Equal-Length Operations

Peter Brucker · Natalia V. Shakhlevich

Received: date / Accepted: date

Abstract A flow-shop batching problem with consistent batches is considered in which the processing times of all jobs on each machine are equal to p and all batch set-up times are equal to s. In such a problem one has to partition the set of jobs into batches and to schedule the batches on each machine. The processing time of a batch B_i is the sum of processing times of operations in B_i and the earliest start of B_i on a machine is the finishing time of B_i on the previous machine plus the set-up time s. Cheng et al. [2] provided an $O(n)$ pseudopolynomial-time algorithm for solving the special case of the problem with two machines. Mosheiov & Oron [3] developed an algorithm of the same time complexity for the general case with more than two machines. Ng and Kovalyov [9] improved the pseudopolynomial complexity to $O\left(\sqrt{n}\right)$. In this paper we provide a polynomial-time algorithm of time complexity $O\left(\log^3 n\right)$.

Keywords Batch scheduling · Flow shop · Polynomial-time algorithm

1 Introduction

Ng and Kovalyov [9] investigated batching problems in a flow-shop environment and derived complexity results for several special cases. One result is an $O\left(\sqrt{n}\right)$-time algorithm for the following problem.

There are n identical jobs $j = 1, \ldots, n$. Each job is ready for processing at time zero and it has to be processed on machines $M_1, M_2, \ldots, M_m$ in this order. The processing time of job j is equal to $p > 0$ on each machine and is equal for all jobs. Batches are formed on each machine. They may include any number of jobs. Jobs in a batch are processed on each machine sequentially so that the processing time of batch B_i on one machine is equal to pb_i, where $b_i = |B_i|$ is the number of jobs in B_i. In other words, jobs of batch B_i become available for downstream processing on machine M_{l+1}

P. Brucker
Universität Osnabrück, Fachbereich Mathematik/Informatik, 49069 Osnabrück, Germany
E-mail: pbrucker@uni-osnabrueck.de

N.V. Shakhlevich
School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.
E-mail: NShakhlevich@leeds.ac.uk

after completion of B_i on machine M_l. A setup time $s \geq 0$ immediately precedes the processing of a batch on each machine and it cannot start until processing of a batch has ended on the previous machine. No machine can process a job while performing a setup. s and p are assumed to be integer. On each machine one has to partition the jobs into batches and to sequence the batches so that the makespan is minimized. Using standard three-field notation, the problem can be denoted as $F|p_{ij} = p, s-batch|C_{\max}$, see, e.g., [1].

Ng and Kovalyov [9] show that there exists an optimal permutation schedule, i.e., a schedule in which the jobs (and therefore the batches) are scheduled in the same order on each machine. Furthermore, for the $O(\sqrt{n})$ algorithm they assume that the building of batches is consistent among the machines, i.e., the batch partitioning $B_1, \ldots, B_k$ is the same on each machine. In such a situation one has to partition the set of jobs into batches and to sequence these batches.

Let $B_1, \ldots, B_k$ be a sequence of batches to be scheduled in this order on each machine. Then the makespan of this schedule is equal to

$$C_{\max} = \sum_{i=1}^{k}(s + pb_i) + (m-1)(s + pb_{i^*}), \tag{1}$$

where $b_{i^*} = \max\{b_i | i = 1, \ldots, k\}$. Minimizing the right hand side of (1) for a fixed k is equivalent to solving the integer program

$$\begin{array}{ll} \min & \sum_{i=1}^{k}(s + pb_i) + (m-1)(s + py) \\ \text{s.t.} & \sum_{i=1}^{k} b_i = n \\ & 1 \leq b_i \leq y \text{ and integer}, \qquad 1 \leq i \leq k. \end{array} \tag{2}$$

As the objective function of (2) can be written

$$\sum_{i=1}^{k}(s + pb_i) + (m-1)(s + py) = ks + pn + (m-1)s + p(m-1)y,$$

(2) is equivalent to the integer program

$$\begin{array}{ll} \min & y \\ \text{s.t.} & \sum_{i=1}^{k} b_i = n \\ & 1 \leq b_i \leq y \text{ and integer}, \; 1 \leq i \leq k. \end{array} \tag{3}$$

Summation of the constraints $1 \leq b_i \leq y$ together with $\sum_{i=1}^{k} b_i = n$ provides

$$n = \sum_{i=1}^{k} b_i \leq ky \quad \text{or} \quad \frac{n}{k} \leq y.$$

This implies $\lceil \frac{n}{k} \rceil \leq y$ because y must be integer. Thus $\lceil \frac{n}{k} \rceil$ is a lower bound for the optimal solution value of (3). If $\frac{n}{k}$ is integer, this lower bound is achieved by setting

$$b_i = y := \frac{n}{k} \text{ for } i = 1, \ldots, k.$$

Otherwise we set

$$b_i = \begin{cases} \lfloor \frac{n}{k} \rfloor & \text{for } i = 1, \ldots, k\lceil \frac{n}{k} \rceil - n, \\ \lceil \frac{n}{k} \rceil & \text{for } i = k\lceil \frac{n}{k} \rceil - n + 1, \ldots, k, \end{cases}$$

i.e., we choose $x = k\left\lceil \frac{n}{k} \right\rceil - n$ batches of size $\left\lfloor \frac{n}{k} \right\rfloor$ and $k - x = k - (k\left\lceil \frac{n}{k} \right\rceil - n) = n - k\left\lfloor \frac{n}{k} \right\rfloor$ batches of size $\left\lceil \frac{n}{k} \right\rceil$.

Thus the batching problem reduces to finding an integer $1 \leq k_{opt} \leq n$ which minimizes

$$\varphi(k) = ks + p(m-1)\left\lceil \frac{n}{k} \right\rceil = p(m-1)\left(\frac{s}{p(m-1)}k + \left\lceil \frac{n}{k} \right\rceil\right)$$

or

$$g(k) = ak + \left\lceil \frac{n}{k} \right\rceil \tag{4}$$

with

$$a = \frac{s}{p(m-1)}.$$

In this paper we present a polynomial-time algorithm which calculates an integer $1 \leq k_{opt} \leq n$ which minimizes the function (4) for any fixed positive rational number a. The shape of the graph of the function $g(k)$ is shown in Figure 1.

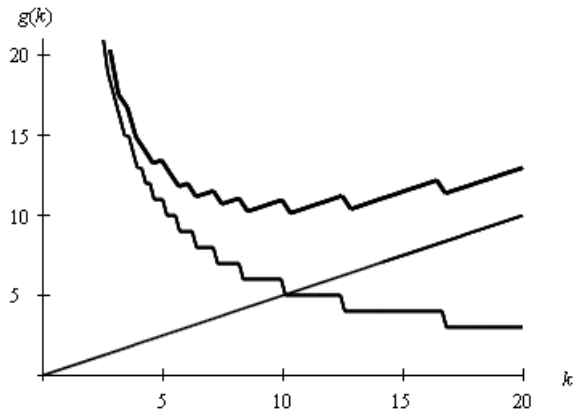

Fig. 1 Function $g(k)$ and its two components ak and $\left\lceil \frac{n}{k} \right\rceil$

The paper is organized as follows. In Sections 2 and 3 properties of the function $g(k) = ak + \left\lceil \frac{n}{k} \right\rceil$ are derived. It is shown in Section 2 that the problem can be solved easily if a is an integer. Additionally some symmetry properties for the values $\left\lceil \frac{n}{k} \right\rceil$ are derived. For non-integer values a the difficulties of calculating a value $k_{opt} \in \{1, \ldots, n\}$ which minimizes $g(k)$ arise because $g(k)$ is oscillating in a neighborhood of k_{opt}. In Section 3 this oscillation behavior is studied in detail. It leads to a polynomial-time algorithm which is derived in Sections 4 and 5. In Section 6 it is shown that the running time of this algorithm is $O(\log^3 n)$. Section 7 contains some concluding remarks.

2 Symmetry and reduction to the case $a > 1$

In this section we first show that it is easy to minimize

$$g(k) = ak + \left\lceil \frac{n}{k} \right\rceil$$

if a is a non-negative integer. Then for non-integer positive values a we reduce the case $a < 1$ to the case $a > 1$ by using some symmetry properties of the values $\lceil \frac{n}{k} \rceil$.

To calculate a value k_{opt} which minimizes the function $g(k)$ if a is integer we consider the continuous function

$$f(k) = ak + \frac{n}{k}.$$

The functions f and g have the following properties:

- $f(k)$ is strictly decreasing (increasing) for $k < \sqrt{\frac{n}{a}}$ $(k > \sqrt{\frac{n}{a}})$ and reaches its minimum if $k = \sqrt{\frac{n}{a}}$,
- $g(k) = ak + \lceil \frac{n}{k} \rceil = \lceil ak + \frac{n}{k} \rceil = \lceil f(k) \rceil$ holds for integer a .

Using these two properties we have for all $k \leq \lfloor \sqrt{\frac{n}{a}} \rfloor$

$$f\left(\left\lfloor \sqrt{\tfrac{n}{a}} \right\rfloor\right) \leq f(k)$$

and therefore

$$g\left(\left\lfloor \sqrt{\tfrac{n}{a}} \right\rfloor\right) = \left\lceil f\left(\left\lfloor \sqrt{\tfrac{n}{a}} \right\rfloor\right)\right\rceil \leq \lceil f(k) \rceil = g(k).$$

Similarly it can be shown that

$$g\left(\left\lceil \sqrt{\tfrac{n}{a}} \right\rceil\right) \leq g(k) \text{ for all } k \geq \left\lceil \sqrt{\tfrac{n}{a}} \right\rceil.$$

Thus,

$$k_{opt} = \begin{cases} \lfloor \sqrt{\frac{n}{a}} \rfloor, \text{ if } g\left(\lfloor \sqrt{\frac{n}{a}} \rfloor\right) \leq g\left(\lceil \sqrt{\frac{n}{a}} \rceil\right), \\ \lceil \sqrt{\frac{n}{a}} \rceil, \text{ otherwise.} \end{cases}$$

From now on let a be a positive non-integer number.

Definition 1 Integer values K_i are defined recursively as follows:

- $K_1 = 1$;
- given K_i, let K_{i+1} be the integer w with $K_i < w \leq n$, such that $\lceil \frac{n}{\nu} \rceil = \lceil \frac{n}{\nu - 1} \rceil$ for $\nu = K_i + 1, \ldots, w-1$, and $\lceil \frac{n}{w} \rceil \neq \lceil \frac{n}{w-1} \rceil$.

Let r be the number of all K_i-values $K_1, \ldots, K_r = n$. Notice, that the function $g(k)$ is strictly increasing in the interval $[K_i, K_{i+1} - 1]$. Therefore, one can restrict to the K_i-values when searching for an integer value k_{opt} minimizing the function g.

The following theorem describes an important symmetry relation between the K_i-values and $\lceil \frac{n}{i} \rceil$-values.

Theorem 1 *There exists a symmetry relation between the values K_i and K_{r-i+1} of the form:*

$$K_i = \left\lceil \frac{n}{K_{r-i+1}} \right\rceil \quad \textit{for } i = 1, \ldots, r. \tag{5}$$

Equivalently,

$$K_{r-i+1} = \left\lceil \frac{n}{K_i} \right\rceil \quad \textit{for } i = 1, \ldots, r \tag{6}$$

holds.

Before proving Theorem 1 we derive some useful properties.

For all positive integers j the inequality

$$\frac{1}{j+1} - \frac{1}{j+2} = \frac{1}{(j+1)(j+2)} \leq \frac{1}{j(j+1)} = \frac{1}{j} - \frac{1}{j+1}$$

holds, which implies the inequality

$$\frac{n}{j+1} - \frac{n}{j+2} \leq \frac{n}{j} - \frac{n}{j+1} \text{ for all } j \geq 1. \tag{7}$$

Lemma 1 *If* $\left\lceil \frac{n}{j} \right\rceil = \left\lceil \frac{n}{j+1} \right\rceil$, *then* $\left\lceil \frac{n}{\nu} \right\rceil - \left\lceil \frac{n}{\nu+1} \right\rceil \leq 1$ *for all* $\nu \geq j$.

Proof Assume that $\left\lceil \frac{n}{\nu} \right\rceil - \left\lceil \frac{n}{\nu+1} \right\rceil > 1$, i.e., $\left\lceil \frac{n}{\nu} \right\rceil - \left\lceil \frac{n}{\nu+1} \right\rceil \geq 2$ for some $\nu > j$. Then

$$\frac{n}{\nu} - 1 \geq \left\lceil \frac{n}{\nu} \right\rceil - 2 \geq \left\lceil \frac{n}{\nu+1} \right\rceil \geq \frac{n}{\nu+1}$$

or

$$\frac{n}{\nu} - \frac{n}{\nu+1} \geq 1.$$

Furthermore $\left\lceil \frac{n}{j} \right\rceil = \left\lceil \frac{n}{j+1} \right\rceil$ implies

$$\frac{n}{j} - \frac{n}{j+1} < 1.$$

Together with (7) we have the contradiction:

$$1 \leq \frac{n}{\nu} - \frac{n}{\nu+1} \leq \frac{n}{\nu-1} - \frac{n}{\nu} \leq \cdots \leq \frac{n}{j} - \frac{n}{j+1} < 1.$$

□

Definition 2 The index t is defined as the largest index such that $K_\nu = K_{\nu-1} + 1$ for $\nu = 2, \ldots, t$.

Observe that

$$K_\nu = \nu \quad \text{for } \nu = 1, \ldots, t$$

and

$$\begin{aligned} \left\lceil \frac{n}{\nu} \right\rceil &< \left\lceil \frac{n}{\nu-1} \right\rceil \text{ for } \nu = 2, \ldots, t, \\ \left\lceil \frac{n}{t} \right\rceil &= \left\lceil \frac{n}{t+1} \right\rceil. \end{aligned} \tag{8}$$

Furthermore, due to Lemma 1

$$\left\lceil \frac{n}{K_\nu} \right\rceil = \left\lceil \frac{n}{K_{\nu+1}} \right\rceil + 1 \quad \text{for all } \nu \geq t. \tag{9}$$

Lemma 2 *For* $i = 1, \ldots, r - t + 1$ *the equality*

$$\left\lceil \frac{n}{K_{r-i+1}} \right\rceil = i \tag{10}$$

holds.

Proof Due to (9) and Definition 1 of the K_ν-values

$$\left\lceil \frac{n}{K_{r-i}} \right\rceil = \left\lceil \frac{n}{K_{r-i+1}} \right\rceil + 1$$

holds for all i with $r - i \geq t$. Thus, by induction we have

$$\left\lceil \frac{n}{K_{r-i+1}} \right\rceil = i \quad \text{for all } i \text{ with } r - i \geq t - 1, \text{ i.e., } i \leq r - t + 1$$

because $\left\lceil \frac{n}{K_r} \right\rceil = 1$. □

Lemma 3 *For $i = 1, \ldots, r - t + 1$ the equality*

$$\left\lceil \frac{n}{i} \right\rceil = K_{r-i+1}$$

is valid.

Proof By Definition 1 of the K_ν-values we have

$$\left\lceil \frac{n}{K_{r-i+1}} \right\rceil \neq \left\lceil \frac{n}{K_{r-i+1} - 1} \right\rceil$$

which together with (10) implies

$$\left\lceil \frac{n}{K_{r-i+1} - 1} \right\rceil > i.$$

With

$$\frac{n}{K_{r-i+1}} \leq \left\lceil \frac{n}{K_{r-i+1}} \right\rceil = i \quad \text{and} \quad \frac{n}{K_{r-i+1} - 1} > i$$

we conclude

$$K_{r-i+1} - 1 < \frac{n}{i} \leq K_{r-i+1} \text{ or } \left\lceil \frac{n}{i} \right\rceil = K_{r-i+1}.$$

□

Lemma 4 *$r - t < t$ and therefore*

$$\left\lceil \frac{r}{2} \right\rceil \leq t.$$

Proof Assume that $r - t \geq t$. Then

$$\left\lceil \frac{n}{K_{r-t+1}} \right\rceil = t = K_t = \left\lceil \frac{n}{r - t + 1} \right\rceil, \tag{11}$$

where the first equality holds due to Lemma 2 taking $i = t$, the second equality holds due to Definition 2 and the third equality holds due to Lemma 3 taking $i = r - t + 1$.

On the other hand, by Definition 2

$$\left\lceil \frac{n}{t} \right\rceil = \left\lceil \frac{n}{t+1} \right\rceil,$$

which together with $r-t+1 \geq t+1$ implies $K_{r-t+1} > r-t+1$ or $K_{r-t+1}-1 \geq r-t+1$. Together with Definition 1 we have:

$$\left\lceil \frac{n}{K_{r-t+1}} \right\rceil < \left\lceil \frac{n}{K_{r-t+1} - 1} \right\rceil \leq \left\lceil \frac{n}{r - t + 1} \right\rceil,$$

which is a contradiction to (11). □

Lemma 5 *For $i = r - t + 1, \dots, t$ the equality*

$$K_i = \left\lceil \frac{n}{K_{r-i+1}} \right\rceil$$

holds.

Proof First we observe that in accordance with Lemma 4

$$r - t + 1 \le t,$$

so that

$$r - t + 1 = K_{r-t+1}.$$

It follows that

$$\left\lceil \frac{n}{K_t} \right\rceil \overset{\text{Lemma 2}}{=} r - t + 1 = K_{r-t+1}$$

and

$$\left\lceil \frac{n}{K_{r-t+1}} \right\rceil = \left\lceil \frac{n}{r-t+1} \right\rceil \overset{\text{Lemma 3}}{=} K_t.$$

The statement of Lemma 5 now follows from the relations:

$$\begin{array}{lllll} K_{r-t+1} & = K_{r-t} + 1, & K_{r-t+2} = K_{r-t+1} + 1, & \dots, & K_t = K_{t-1} + 1, \\ \left\lceil \frac{n}{K_t} \right\rceil & < \left\lceil \frac{n}{K_{t-1}} \right\rceil & < \quad \cdots \quad < & & \left\lceil \frac{n}{K_{r-t+1}} \right\rceil \end{array}$$

together with the equalities

$$\left\lceil \frac{n}{K_t} \right\rceil = K_{r-t+1} \qquad \text{and} \qquad \left\lceil \frac{n}{K_{r-t+1}} \right\rceil = K_t.$$

□

Proof of Theorem 1 The proof is based on the above lemmas and equality

$$K_i = i \quad \text{for } i = 1, \dots, t.$$

By Lemma 4 we have $r - t + 1 \le t$. Therefore, condition (5) holds due to Lemma 2 for $i = 1, \dots, r - t$, due to Lemma 5 for $i = r - t + 1, \dots, t$, and due to Lemma 3 for $i = t + 1, \dots, r$. □

In what follows we demonstrate that if $0 < a < 1$, then the problem of minimizing function g can be reduced to a symmetric problem with $a > 1$. For this purpose, we indicate that the function g depends on the parameter a denoting the function g by g_a. We have to find an index i^* such that K_{i^*} minimizes $g_a(K_i) = aK_i + \left\lceil \frac{n}{K_i} \right\rceil$ for all $i = 1, \dots, r$. Due to Theorem 1

$$K_i = \left\lceil \frac{n}{K_{r-i+1}} \right\rceil \text{ for } i = 1, .., r \text{ or } \left\lceil \frac{n}{K_i} \right\rceil = K_{r-i+1} \text{ for } i = 1, .., r.$$

Therefore for $a > 0$ we have

$$\begin{aligned} g_a(K_{r-i+1}) &= aK_{r-i+1} + \left\lceil \frac{n}{K_{r-i+1}} \right\rceil = a \left\lceil \frac{n}{K_i} \right\rceil + K_i \\ &= a \left(\frac{1}{a} K_i + \left\lceil \frac{n}{K_i} \right\rceil \right) = a g_{\frac{1}{a}}(K_i). \end{aligned} \tag{12}$$

Thus, with (12) the Case $0 < a < 1$ can be reduced to the Case $a > 1$. One has to find K_{i^*} such that $g_{\frac{1}{a}}(K_{i^*})$ (where $\frac{1}{a} > 1$) is minimal. Then $K_{r-i^*+1} = \left\lceil \frac{n}{K_{i^*}} \right\rceil$ provides an optimal solution $g_a\left(\left\lceil \frac{n}{K_{i^*}} \right\rceil\right)$ for g_a.

Now we derive an estimate on the value r of the possible K_i-values.

Lemma 6 *The value of r is either $2s$ or $2s-1$ with a unique integer $s \le t$ defined from the condition:*

$$\sqrt{n+\frac{1}{4}} - \frac{1}{2} \le s < \sqrt{n+\frac{1}{4}} + \frac{1}{2}.$$

Proof Condition $r = 2s$ implies together with Lemma 4 that $s = \frac{r}{2} < t$. Thus

$$s = K_s \overset{\text{Theorem 1}}{=} \left\lceil \frac{n}{K_{r-s+1}} \right\rceil = \left\lceil \frac{n}{K_{2s-s+1}} \right\rceil = \left\lceil \frac{n}{K_{s+1}} \right\rceil \overset{\text{Definition 2}}{=} \left\lceil \frac{n}{s+1} \right\rceil.$$

Therefore $s - 1 < \frac{n}{s+1} \le s$ or

$$\sqrt{n+\frac{1}{4}} - \frac{1}{2} \le s < \sqrt{n+1}.$$

Observe that $\sqrt{n+1} < \sqrt{n+\frac{1}{4}} + \frac{1}{2}$.

If $r = 2s-1$ for some s, then we have $s = K_s$. Indeed, by Lemma 4

$$t \ge \left\lceil \frac{r}{2} \right\rceil = \left\lceil s - \frac{1}{2} \right\rceil \ge s - \frac{1}{2},$$

which implies

$$s \le t$$

since s and t are integer. The latter inequality together with Definition 2 of t implies $s = K_s$. By Theorem 1

$$s = K_s = \left\lceil \frac{n}{K_{r-s+1}} \right\rceil = \left\lceil \frac{n}{K_{2s-1-s+1}} \right\rceil = \left\lceil \frac{n}{K_s} \right\rceil = \left\lceil \frac{n}{s} \right\rceil,$$

which implies $s - 1 < \frac{n}{s} \le s$ or

$$\sqrt{n} \le s < \sqrt{n+\frac{1}{4}} + \frac{1}{2}.$$

Observe that $\sqrt{n+\frac{1}{4}} - \frac{1}{2} \le \sqrt{n}$. □

To find an integer $1 \le i^* \le r$ such that K_{i^*} minimizes g we now assume that a is a non-integer number with $a > 1$. As we show in the following lemma, the search for i^* can be limited to the range $\{1, 2, \ldots, h_1\}$ with $h_1 \le t$ since the sequence $K_{h_1+1} < K_{h_1+2} < \cdots < K_r$ is increasing.

Lemma 7 *If $a > 1$ then $K_{i^*} \in \{1, \ldots, h_1\}$ where*

$$h_1 = \begin{cases} s, & \text{if } \left\lceil \frac{n}{s+1} \right\rceil = \left\lceil \frac{n}{s} \right\rceil, \\ s+1, & \text{otherwise.} \end{cases}$$

Furthermore $h_1 \le t$.

Proof Due to Definition 2 of t we have $K_i = K_{i-1} + 1$ for $i = 2, \ldots, t$. Together with Theorem 1 this implies

$$\left\lceil \frac{n}{K_{r-i+1}} \right\rceil = K_i = K_{i-1} + 1 = \left\lceil \frac{n}{K_{r-i+2}} \right\rceil + 1 \quad \text{for } i = 2, \ldots, t.$$

Because $a > 1$, this implies

$$g(K_{\nu-1}) - g(K_\nu) = aK_{\nu-1} + \left\lceil \frac{n}{K_{\nu-1}} \right\rceil - aK_\nu - \left\lceil \frac{n}{K_\nu} \right\rceil \leq -a + \left(\left\lceil \frac{n}{K_{\nu-1}} \right\rceil - \left\lceil \frac{n}{K_\nu} \right\rceil \right) = \\ = -a + 1 < 0$$

for $\nu = r - t + 2, \ldots, r$, or

$$g(K_{r-t+1}) < g(K_{r-t+2}) < \cdots < g(K_r).$$

Due to Lemma 4 the inequality $t \geq \left\lceil \frac{r}{2} \right\rceil$ holds, which implies

$$r - t \leq r - \left\lceil \frac{r}{2} \right\rceil \leq r - \frac{r}{2} = \frac{r}{2}$$

and therefore

$$r - t + 1 \leq \frac{r}{2} + 1 \overset{\text{Lemma 6}}{\leq} \frac{2s}{2} + 1 = s + 1. \tag{13}$$

We are on the safe side if we look for a value K_{i^*} such that $g(K_{i^*})$ is a minimum of

$$\{g(K_\nu) | \nu = 1, 2, \ldots, s + 1\}.$$

Notice that by Lemma 6

$$s \leq t. \tag{14}$$

Now h_1 can be calculated by the following procedure.
If $\left\lceil \frac{n}{s+1} \right\rceil \neq \left\lceil \frac{n}{s} \right\rceil$, then $s \neq t$ and therefore by (14) $s + 1 \leq t$. Thus, we set

$$h_1 := s + 1.$$

Otherwise $s = t$ and we can set

$$h_1 := s.$$

□

Notice that due to the above lemma we have

$$K_i = i$$

for all relevant i-values. For this reason in what follows we use i instead of K_i.

Lemma 7 immediately implies an $O(\sqrt{n})$-time algorithm just by calculating h_1 and evaluating values $g(i)$ for $i = 1, \ldots, h_1$. Observe that the earlier algorithm due to Ng and Kovalyov [9] has the same complexity bound. A faster $O\left(\log^3 n\right)$ will be formulated in Section 4.

3 Oscillation

The results of the previous section imply that the case of integer a is easily solvable and for the non-integer case one may consider $a > 1$. From now on we assume that a is non-integer and greater than 1. The difficulties to find a positive integer k_{opt} minimizing the function g for non-integer values a arise because g is usually oscillating in the area around the optimal solution value k_{opt}. In this section we first present a general description of the oscillating behavior of function g and then proceed with details and justification.

Consider differences

$$\Delta_i = \left\lceil \frac{n}{i-1} \right\rceil - \left\lceil \frac{n}{i} \right\rceil \tag{15}$$

which characterize the changes in the values of function g:

$$g(i-1) - g(i) = \Delta_i - a.$$

As we show later in this section, Δ-values mainly decrease, but they may occasionally increase by 1:

$$\Delta_j \leq \Delta_i + 1 \quad \text{for all } 1 < i < j.$$

The increase/decrease structure of Δ-values has some important structural properties.

Consider the subsequence $\Delta_2, \ldots, \Delta_{h_1}$ which contains an optimum k_{opt}.

- The first Δ_j values are large and if $\Delta_j > \lceil a \rceil$, then function g is strictly decreasing.
- When Δ_j achieves $\lfloor a \rfloor$ for the first time for some $j = l$, the values of Δ_j start alternating between $\lceil a \rceil$ and $\lfloor a \rfloor$ and function g is oscillating (increasing when $\Delta_j = \lfloor a \rfloor$ and decreasing when $\Delta_j = \lceil a \rceil$).
- The oscillation range of Δ-values ends after Δ_j changes to $\lceil a \rceil$ for the last time for some $j = h - 1$, all subsequent Δ-values are less than or equal to $\lfloor a \rfloor$ and function g is strictly increasing.

This means that the range $\{1, \ldots, h_1\}$ can be narrowed to a smaller range $\{l, \ldots, h\}$ with $1 \leq l \leq h \leq h_1$ and within that range $\Delta_j \in \{\lceil a \rceil\,,\ \lfloor a \rfloor\}$. We use notation $A^1 = \lceil a \rceil$ for the Δ_j-value which leads to a decrease in g and and $B^1 = \lfloor a \rfloor$ for the Δ_j-value which leads to an increase in g.

As we show later in this section, there is a certain regularity in the pattern of A^1 and B^1. In particular,

(i) subsequence $\Delta_l, \ldots, \Delta_h$ starts with $B^1 \underbrace{A^1 \ldots A^1} B^1 \underbrace{A^1 \ldots A^1} \ldots B^1 \underbrace{A^1 \ldots A^1}$ consisting of repeated $A^1 \ldots A^1$ separated by a single B^1;

(ii) the next part is of the form $B^1 A^1 \ldots B^1 A^1$ with single occurrences of B^1 alternating with single occurrences of A^1;

(iii) the final part $\underbrace{B^1 \ldots B^1} A^1 \underbrace{B^1 \ldots B^1} A^1 \ldots \underbrace{B^1 \ldots B^1} A^1$ consists of repeated $B^1 \ldots B^1$ separated by a single A^1.

If $a - \lfloor a \rfloor \geq \lceil a \rceil - a$, i.e. $a \geq \frac{1}{2}$, then the search can be limited to the first subsequence $B^1 \underbrace{A^1 \ldots A^1} B^1 \underbrace{A^1 \ldots A^1} \ldots B^1 \underbrace{A^1 \ldots A^1}$ only. Depending on the number of repetitions of A^1, the patterns $B^1 A^1 \ldots A^1$ are combined into blocks A^2 and B^2, which are called level 2 blocks. They are defined in such a way that the cumulative effect of A^2 (B^2) leads to a decrease in g (increase in g, respectively). Similar to level 1,

the blocks A^2 and B^2 of level 2 are alternating in a manner described by (i)-(iii) with the first part $B^2\underbrace{A^2\dots A^2}B^2\underbrace{A^2\dots A^2}\dots B^2\underbrace{A^2\dots A^2}$ followed by $B^2A^2\dots B^2A^2$ and then by $\underbrace{B^2\dots B^2}A^2\underbrace{B^2\dots B^2}A^2\dots\underbrace{B^2\dots B^2}A^2$. Again, the search can be limited to a smaller subsequence $\Delta_{l'},\dots,\Delta_{h'}$ $(l\le l'\le h'\le h)$ corresponding to a part of $\Delta_l,\dots,\Delta_h$.

The process of grouping several lower level blocks into higher level blocks and narrowing the boundaries of the search continues iteratively until at some level there are no blocks in-between the left and the right boundaries. Since any higher level block consists of at least 2 lower level blocks, the overall number of levels is no more than $\log_2 n$. In what follows we explain the described approach in detail justifying its correctness.

First we derive some properties of the Δ_i-values.

Lemma 8 *For integers $1\le i<j<n$ and $k\ge 1$ with $j+k\le n$*

$$\left\lceil\frac{n}{j}\right\rceil-\left\lceil\frac{n}{j+k}\right\rceil\le\left\lceil\frac{n}{i}\right\rceil-\left\lceil\frac{n}{i+k}\right\rceil+1$$

holds.

Proof For each non-negative integer ν with $j+\nu+1\le n$ due to (7)

$$\frac{n}{j+\nu}-\frac{n}{j+\nu+1}\le\frac{n}{i+\nu}-\frac{n}{i+\nu+1}$$

holds. This implies

$$\begin{aligned}\frac{n}{j}-\left\lceil\frac{n}{j+k}\right\rceil&\le\frac{n}{j}-\frac{n}{j+k}\\&=\left(\frac{n}{j}-\frac{n}{j+1}\right)+\left(\frac{n}{j+1}-\frac{n}{j+2}\right)+\dots+\left(\frac{n}{j+k-1}-\frac{n}{j+k}\right)\\&\le\left(\frac{n}{i}-\frac{n}{i+1}\right)+\left(\frac{n}{i+1}-\frac{n}{i+2}\right)+\dots+\left(\frac{n}{i+k-1}-\frac{n}{i+k}\right)\\&=\frac{n}{i}-\frac{n}{i+k}\le\left\lceil\frac{n}{i}\right\rceil-\frac{n}{i+k}.\end{aligned}$$

Therefore, together with $\left\lceil\frac{n}{j}\right\rceil\le\frac{n}{j}+1$

$$\left\lceil\frac{n}{j}\right\rceil-\left\lceil\frac{n}{j+k}\right\rceil\le\frac{n}{j}+1-\left\lceil\frac{n}{j+k}\right\rceil\le\left\lceil\frac{n}{i}\right\rceil-\frac{n}{i+k}+1,$$

i.e.

$$\left\lceil\frac{n}{j}\right\rceil-\left\lceil\frac{n}{j+k}\right\rceil\le\left\lceil\frac{n}{i}\right\rceil-\frac{n}{i+k}+1.$$

In the previous inequality $\frac{n}{i+k}$ is the only possible non-integer value. Hence it can be replaced by $\left\lceil\frac{n}{i+k}\right\rceil$, which proves the lemma. □

Lemma 9 $\Delta_j\le\ \Delta_i+1$ *for all $1<i<j$.*

Proof Application of Lemma 8 with $k=1$ and i replaced by $i-1$ as well as j replaced by $j-1$ proves the claim. □

Lemma 10 *Let $j \le h_1$ be a positive integer. Then the following statements are correct.*

(a) If $\Delta_j > \lceil a \rceil$ then $g(\nu-1) > g(\nu)$ for all $2 \le \nu \le j$.
(b) If $\Delta_j < \lfloor a \rfloor$ then $g(\nu) > g(\nu-1)$ for all $h_1 \ge \nu \ge j$.
(c) If $\Delta_j = \lceil a \rceil$ then $g(j) = g(j-1) - (\lceil a \rceil - a)$, i.e. g decreases by $d = \lceil a \rceil - a$.
(d) If $\Delta_j = \lfloor a \rfloor$ then $g(j) = g(j-1) + (a - \lfloor a \rfloor)$, i.e. g increases by $e = a - \lfloor a \rfloor$.

Proof (a) By Lemma 9 for all $\nu < j$ we have $\Delta_j \le \Delta_\nu + 1$. Furthermore, $\Delta_j > \lceil a \rceil$ implies $\Delta_j \ge \lceil a \rceil + 1$. Therefore

$$\begin{aligned} g(\nu-1) - g(\nu) = a(\nu-1) + \left\lceil \tfrac{n}{\nu-1} \right\rceil - a\nu - \left\lceil \tfrac{n}{\nu} \right\rceil = \\ = \Delta_\nu - a \ge \Delta_j - 1 - a \ge (\lceil a \rceil + 1) - 1 - a > 0. \end{aligned} \tag{16}$$

(b) By Lemma 9 for all $j \le \nu$ we have $\Delta_\nu \le \Delta_j + 1$. Furthermore, $\Delta_j < \lfloor a \rfloor$ implies $\Delta_j + 1 \le \lfloor a \rfloor$. Therefore

$$\begin{aligned} g(\nu) - g(\nu-1) = a\nu + \left\lceil \tfrac{n}{\nu} \right\rceil - a\,(\nu-1) - \left\lceil \tfrac{n}{\nu-1} \right\rceil = \\ = a - \Delta_\nu \ge a - \Delta_j - 1 \ge a - \lfloor a \rfloor > 0. \end{aligned} \tag{17}$$

(c) For $\nu = j$ (16) provides $g(j-1) - g(j) = \Delta_j - a = \lceil a \rceil - a$.
(d) For $\nu = j$ (17) provides $g(j) - g(j-1) = a - \Delta_j = a - \lfloor a \rfloor$.

□

Notice that the statements of Lemma 10 need not to be true if $j > h_1$ because for $j > t \ge h_1$ the equality $K_{j+1} = K_j + 1$ does not hold. Therefore one has to restrict the search for k_{opt} to the range $1, \ldots, h_1$. Remember, we have no efficient algorithm to calculate t.

Let $l \le h_1$ be the smallest positive integer with $\Delta_l \le \lfloor a \rfloor$. We may assume that such an l always exists; otherwise $\Delta_j > \lfloor a \rfloor$, i.e., $\Delta_j \ge \lceil a \rceil$ for all $j \le h_1$, so that g decreases in $[1, h_1]$ and $k_{opt} = h_1$. Furthermore, $\Delta_j > \lfloor a \rfloor$ for at least one j; otherwise $\Delta_j \le \lfloor a \rfloor$ for all $j \le t$, so that g increases in $[1, h_1]$ and $k_{opt} = 1$.

Let h be the smallest integer with $\Delta_\nu \le \lfloor a \rfloor$ for all ν, $h \le \nu \le h_1$. If such an integer does not exist, we set $h = h_1$. Clearly $l \le h$ and an optimal solution can be easily identified if $l \in \{h-1, h\}$. Therefore it remains to consider the case $l < h - 1$.

By Lemma 9 we have

$$\Delta_\nu \in \{\lfloor a \rfloor, \lceil a \rceil\} \text{ for all } \nu = l, \ldots, h-1.$$

We study the structure of the oscillation area defined by the sequence $\Delta_l, \Delta_{l+1}, \ldots, \Delta_{h-1}$. We represent it as a sequence of repeated blocks of Δ-values of two types A and B. *Level 1* blocks A^1 and B^1 are of the form

$$\begin{aligned} A^1 &= \lceil a \rceil, \\ B^1 &= \lfloor a \rfloor. \end{aligned}$$

More complex *level* λ blocks A^λ and B^λ, $\lambda = 2, 3, ...$, are defined inductively. Each of these blocks A^λ or B^λ consists of several blocks $A^{\lambda-1}$ and $B^{\lambda-1}$.

The following lemma describes the structure of the sequence $\Delta_l, \Delta_{l+1}, \ldots, \Delta_{h-1}$ in terms of A^1 and B^1. The higher level blocks A^λ and B^λ are introduced and studied after it.

Lemma 11 *The oscillation area has the form*

$$\begin{array}{l} \overset{\Delta_l}{B^1}\underbrace{A^1\dots A^1}_{\overline{m}_1}B^1\underbrace{A^1\dots A^1}_{\overline{m}_2}\dots B^1\underbrace{A^1\dots A^1}_{\overline{m}_{\overline{s}}}B^1A^1B^1A^1\dots \\ \dots B^1A^1\underbrace{B^1\dots B^1}_{\underline{m}_{\underline{s}}}\dots A^1\underbrace{B^1\dots B^1}_{\underline{m}_2}A^1\underbrace{B^1\dots B^1}_{\underline{m}_1}\overset{\Delta_{h-1}}{A^1} \end{array} \tag{18}$$

where the block marked with $\overline{m}_{\overline{s}}$ ($\underline{m}_{\underline{s}}$) corresponds to the last (first) occurrence of two or more A's (B's),

$$\overline{m}_v \le \overline{m}_u + 1 \quad \textit{for} \quad 1 \le u < v \le \overline{s},$$

and

$$\underline{m}_v \le \underline{m}_u + 1 \quad \textit{for} \quad 1 \le v < u \le \underline{s}.$$

The structure (18) can be described informally as follows. At the beginning of the oscillation area subsequences of consecutive level 1 blocks A^1 are separated by a single B^1. Symmetrically, at the end of the oscillation area subsequences of consecutive level 1 blocks B^1 are separated by a single A^1. $\overline{m}_i$ and $\underline{m}_i$ denote the number of repetitions of A^1 subsequences and B^1 subsequences, respectively. If the middle part is non-empty, then it consists of alternating single occurrences of A^1- and B^1-blocks. In what follows we prove that (18) correctly represents the oscillation area.

Proof The proof is split into 2 parts.
Part (a). First we claim that a subsequence of the form

$$\begin{array}{cccccc} & \Delta_i & \Delta_{i+1} & & \Delta_j & \Delta_{j+1} \\ \dots & B^1 & B^1 & \dots & A^1 & A^1 & \dots \end{array} \tag{19}$$

is not possible. Indeed, in this case we would have (with Lemma 8)

$$\begin{aligned} 2\lceil a\rceil = \Delta_j + \Delta_{j+1} &= \left(\left\lceil\frac{n}{j-1}\right\rceil - \left\lceil\frac{n}{j}\right\rceil\right) + \left(\left\lceil\frac{n}{j}\right\rceil - \left\lceil\frac{n}{j+1}\right\rceil\right) = \left\lceil\frac{n}{j-1}\right\rceil - \left\lceil\frac{n}{j+1}\right\rceil \\ &\le \left\lceil\frac{n}{i-1}\right\rceil - \left\lceil\frac{n}{i+1}\right\rceil + 1 = \Delta_i + \Delta_{i+1} + 1 = 2\lfloor a\rfloor + 1 = 2\lceil a\rceil - 1 \end{aligned}$$

which leads to the contradiction $0 \le -1$.

Due to the fact that a subsequence (19) is not possible to the left of an $A^1A^1\dots A^1$-block of length at least two, all $A^1\dots A^1$-blocks must be separated by a single B^1 and to the right of a $B^1B^1\dots B^1$-block of length at least two all $B^1\dots B^1$-blocks must be separated by a single A^1.
Part (b). Now assume that $\overline{m}_v \ge \overline{m}_u + 2$ for some $1 \le u < v \le \overline{s}$. Let Δ_i and Δ_{i+k} be the Δ's corresponding to the B-bounds of the A-block of length of $\overline{m}_u$ and let Δ_j correspond to the first A^1 in the A-block of length $\overline{m}_v$, i.e., we have the following situation:

$$\begin{array}{l} \overset{\Delta_i}{\dots B^1}\underbrace{A^1\dots A^1}_{\overline{m}_u}\overset{\Delta_{i+k}}{B^1}\dots B^1\underbrace{\overset{\Delta_j\dots\Delta_{j+k}\dots\Delta_{j+k+z}}{A^1\ \dots\ A^1\ \dots\ A^1}}_{\overline{m}_v}B^1\dots \end{array}$$

where $z \geq 0$. Again Lemma 8 leads to the contradiction:

$$\begin{aligned}(k+1)\lceil a\rceil &= \Delta_j + \Delta_{j+1} + \cdots + \Delta_{j+k} = \left\lceil \frac{n}{j-1} \right\rceil - \left\lceil \frac{n}{j+k} \right\rceil \\ &\leq \left\lceil \frac{n}{i-1} \right\rceil - \left\lceil \frac{n}{i+k} \right\rceil + 1 = \Delta_i + \Delta_{i+1} + \cdots + \Delta_{i+k} + 1 \\ &= (k-1)\lceil a\rceil + 2\lfloor a\rfloor + 1 = (k+1)\lceil a\rceil - 1.\end{aligned}$$

Thus, $\overline{m}_v < \overline{m}_u + 2$, i.e. $\overline{m}_v \leq \overline{m}_u + 1$ for $1 \leq u < v \leq \overline{s}$. Symmetrically, it can be shown that $\underline{m}_v \leq \underline{m}_u + 1$ for $1 \leq v < u \leq \underline{s}$. □

Consider an oscillation area of the form (18) where by A^1 the function g is decreased by the value d and by B^1 the function g is increased by the value e, see Lemma 10.

If $d = e$ then the subsequences

$$B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_i} \text{ with } \overline{m}_i \geq 2$$

in the left part of the oscillation area decrease g.

The subsequences

$$A^1 \underbrace{B^1 \dots B^1}_{\underline{m}_i} \text{ with } \underline{m}_i \geq 2$$

in the right part of the oscillation area increase g. The subsequences A^1B^1 in the middle part do not change g. Therefore an optimal solution is given by the end of the last $B^1\underbrace{A^1 \dots A^1}$ subsequence with $\overline{m}_i \geq 2$ occurrences of A^1.

It remains to consider the case $d < e$ (the case $e < d$ is treated symmetrically). In this case an optimal solution can be found in

$$B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_1} B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_2} \dots B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_{\overline{s}}} \tag{20}$$

Let $l' \leq \overline{s}$ be the smallest positive integer with $\overline{m}_{l'} d \leq e$, i.e., $\overline{m}_{l'} \leq \left\lfloor \frac{e}{d} \right\rfloor$. We may assume that such an l' exists because otherwise g achieves its minimum at the end of sequence (20). Furthermore, $\overline{m}_i > \left\lfloor \frac{e}{d} \right\rfloor$ for at least one i; otherwise the minimum is achieved just before the oscillation area, i.e., it is given by the last position before the oscillation area.

Let h' be the smallest positive integer with $\overline{m}_\nu \leq \left\lfloor \frac{e}{d} \right\rfloor$ for all $\nu \geq h'$. By the definition of l' the function g is strictly decreasing for all $\overline{m}_\nu$ with $\nu < l'$ and non-decreasing for all $\overline{m}_\nu$ with $\nu \geq h'$ (because in the latter case $\overline{m}_\nu \leq \left\lfloor \frac{e}{d} \right\rfloor \leq \frac{e}{d}$ or $e \geq \overline{m}_\nu d$). Clearly, $l' \leq h'$ and an optimal solution can be easily identified if $l' \in \{h'-1, h'\}$. Therefore it remains to consider the case $l' < h' - 1$. With Lemma 11

$$\overline{m}_\nu \in \left\{ \left\lfloor \frac{e}{d} \right\rfloor, \left\lceil \frac{e}{d} \right\rceil \right\} \text{ for all } l' \leq \nu \leq h' - 1. \tag{21}$$

Define

$$A^2 = B^1 \underbrace{A^1 \dots A^1}_{\left\lceil \frac{e}{d} \right\rceil} \text{ and } B^2 = B^1 \underbrace{A^1 \dots A^1}_{\left\lfloor \frac{e}{d} \right\rfloor}. \tag{22}$$

Notice, that by the block $A^2(B^2)$ the function g is decreased (increased) by

$$d' = \left\lceil \frac{e}{d} \right\rceil d - e \geq \frac{e}{d}d - e = 0 \quad (e' = e - \left\lfloor \frac{e}{d} \right\rfloor d \geq e - \frac{e}{d}d = 0). \tag{23}$$

In what follows we will show that the new (considerably smaller) oscillation area

$$B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_{l'}} B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_{l'+1}} \dots B^1 \underbrace{A^1 \dots A^1}_{\overline{m}_{h'-1}}$$

has the form

$$B^2 \underbrace{A^2 \dots A^2}_{\overline{m}'_1} B^2 \underbrace{A^2 \dots A^2}_{\overline{m}'_2} \dots B^2 \underbrace{A^2 \dots A^2}_{\overline{m}'_{\overline{s}'}} B^2A^2B^2A^2 \dots$$

$$\dots B^2A^2 \underbrace{B^2 \dots B^2}_{\underline{m}'_{\underline{s}'}} \dots A^2 \underbrace{B^2 \dots B^2}_{\underline{m}'_2} A^2 \underbrace{B^2 \dots B^2}_{\underline{m}'_1} A^2$$

where $\overline{m}'_{\overline{s}'}$, $\underline{m}'_{\underline{s}'} \geq 2$, $\overline{m}'_j \leq \overline{m}'_i + 1$ for $1 \leq i < j \leq \overline{s}'$, and $\underline{m}'_j \leq \underline{m}'_i + 1$ for $1 \leq j < i \leq \underline{s}'$.

Notice, that level 2 blocks are sequences of level 1 blocks. This process will be iterated considering level 2 blocks A^2 and B^2 instead of A^1 and B^1, etc. More specifically, given level λ blocks A^λ and B^λ, where A^λ (B^λ) decreases (increases) g by d^λ (e^λ), the blocks in level $\lambda + 1$ are defined as

$$A^{\lambda+1} = B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil} \text{ and } B^{\lambda+1} = B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor}.$$

Then by the block $A^{\lambda+1}$ ($B^{\lambda+1}$) the function g is decreased (increased) by

$$d^{\lambda+1} = \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil d^\lambda - e^\lambda \geq \frac{e^\lambda}{d^\lambda}d^\lambda - e^\lambda = 0 \quad (e^{\lambda+1} = e^\lambda - \left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor d^\lambda \geq e^\lambda - \frac{e^\lambda}{d^\lambda}d^\lambda = 0). \tag{24}$$

Later we will show that to identify K_{opt} one has to consider at most $O(\log n)$ levels.

The next theorem generalizes Lemma 11.

Theorem 2 *The oscillation area in each level λ has the form*

$$\begin{array}{l} \overset{\Delta_l}{B^\lambda} \underbrace{A^\lambda \dots A^\lambda}_{\overline{m}^\lambda_1} B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\overline{m}^\lambda_2} \dots B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\overline{m}^\lambda_{\overline{s}^\lambda}} B^\lambda A^\lambda B^\lambda A^\lambda \dots \\ \dots B^\lambda A^\lambda \underbrace{B^\lambda \dots B^\lambda}_{\underline{m}^\lambda_{\underline{s}^\lambda}} \dots A^\lambda \underbrace{B^\lambda \dots B^\lambda}_{\underline{m}^\lambda_2} A^\lambda \underbrace{B^\lambda \dots B^\lambda}_{\underline{m}^\lambda_1} \overset{\Delta_{h-1}}{A^\lambda} \end{array} \tag{25}$$

where

$$\overline{m}^\lambda_{\overline{s}^\lambda}, \underline{m}^\lambda_{\underline{s}^\lambda} \geq 2,$$

$$\begin{array}{l} \overline{m}^\lambda_v \leq \overline{m}^\lambda_u + 1 \quad \textit{for} \quad 1 \leq u < v \leq \overline{s}^\lambda, \\ \underline{m}^\lambda_v \leq \underline{m}^\lambda_u + 1 \quad \textit{for} \quad 1 \leq v < u \leq \underline{s}^\lambda. \end{array}$$

Proof We prove the theorem by induction by λ-values. Due to Lemma 11 the theorem is correct for level $\lambda = 1$. For the induction step we follow the structure of the proof of Lemma 11.

Part (a). Consider the oscillation area at a level $\lambda+1$, $\lambda \geq 1$ consisting of blocks $A^{\lambda+1} = B^\lambda A^\lambda \dots A^\lambda$ with $\left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ repetitions of A^λ and $B^{\lambda+1} = B^\lambda A^\lambda \dots A^\lambda$ with $\left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor$ repetitions of A^λ:

$$A^{\lambda+1} = B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil}, \qquad B^{\lambda+1} = B^\lambda \underbrace{A^\lambda \dots A^\lambda}_{\left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor} \tag{26}$$

Again a subsequence

$$\dots B^{\lambda+1} B^{\lambda+1} \dots A^{\lambda+1} A^{\lambda+1} \dots \tag{27}$$

is not possible. Indeed (27) has the form

$$\underbrace{B^\lambda \underset{i'}{A^\lambda} \dots A^\lambda}_{B^{\lambda+1}} \underbrace{B^\lambda A^\lambda \dots \underset{j'}{A^\lambda}}_{B^{\lambda+1}} B^\lambda \dots \underbrace{B^\lambda A^\lambda \underset{i'+\rho}{A^\lambda} \dots A^\lambda}_{A^{\lambda+1}} \underbrace{B^\lambda A^\lambda \dots \underset{j'+\rho}{A^\lambda} A^\lambda}_{A^{\lambda+1}}. \tag{28}$$

Let i' be the index of the first Δ_ν in the first A^λ of the first $B^{\lambda+1}$, i.e., the first position in this first A^λ. Let j' be the last position of the last A^λ of the second $B^{\lambda+1}$. Furthermore let $i'+\rho$ be the first position of the second A^λ in $A^{\lambda+1}$ in (28). Notice that the sequences $\Delta_{i'} \dots \Delta_{j'}$ and $\Delta_{i'+\rho} \dots \Delta_{j'+\rho}$ are identical. Let $i < i'$ be the largest index with $\Delta_i \neq \Delta_{i+\rho}$ and $\Delta_\nu = \Delta_{\nu+\rho}$ for $\nu = i+1, \dots, i'$. Such an index exists because block B^λ is the block preceding immediately position i' in $B^{\lambda+1}$ and A^λ is the block preceding immediately position $i'+\rho$ in $A^{\lambda+1}$ (see (28)). Furthermore, $\Delta_{i+\rho} = \lceil a \rceil$ and $\Delta_i = \lfloor a \rfloor$. To show this, we decompose A^λ and B^λ into blocks $A^{\lambda-1}$ and $B^{\lambda-1}$:

$$\begin{array}{lllll} B^\lambda = & & B^{\lambda-1} & A^{\lambda-1} \dots A^{\lambda-1}, \\ A^\lambda = & B^{\lambda-1} & A^{\lambda-1} & A^{\lambda-1} \dots A^{\lambda-1}. \end{array}$$

It is easy to see that only the first fragments of the above sequences are different while the final parts are the same. We continue the decomposition into lower level blocks:

$$\begin{array}{lllll} B^\lambda = & & B^{\lambda-2} A^{\lambda-2} \dots A^{\lambda-2} & A^{\lambda-1} \dots A^{\lambda-1}, \\ A^\lambda = & B^{\lambda-1} B^{\lambda-2} & A^{\lambda-2} A^{\lambda-2} \dots A^{\lambda-2} & A^{\lambda-1} \dots A^{\lambda-1}. \end{array}$$

Proceeding in a similar way we obtain:

$$\begin{array}{ll} B^\lambda = & B^1\ A^1 \dots A^1\ A^2 \dots A^2 \dots A^{\lambda-2} \dots A^{\lambda-2} \\ A^\lambda = & B^{\lambda-1} B^{\lambda-2} \dots B^1 \quad A^1\ A^1 \dots A^1\ A^2 \dots A^2 \dots A^{\lambda-2} \dots A^{\lambda-2} \end{array} \tag{29}$$

which proves the claim that $\Delta_{i+\rho} = \lceil a \rceil$ and $\Delta_i = \lfloor a \rfloor$.

Let $j > j'$ be the smallest index with $\Delta_\nu = \Delta_{\nu+\rho}$ for $\nu = j'+1, \dots, j-1$ and $\Delta_j \neq \Delta_{j+\rho}$. Again $\Delta_j = \lfloor a \rfloor$ and $\Delta_{j+\rho} = \lceil a \rceil$. Now we have with Lemma 8

$$\begin{aligned} \Delta_i + \dots + \Delta_j + 2 &= \Delta_{i+1} + \dots + \Delta_{j-1} + 2\lfloor a \rfloor + 2 \\ &= \Delta_{i+1+\rho} + \dots + \Delta_{j-1+\rho} + 2\lceil a \rceil \\ &= \Delta_{i+\rho} + \dots + \Delta_{j+\rho} \overset{\text{Lemma 8}}{\leq} \Delta_i + \dots + \Delta_j + 1 \end{aligned} \tag{30}$$

which provides the contradiction $2 \leq 1$.

Part (b). To prove that $\overline{m}_v^{\lambda+1} \leq \overline{m}_u^{\lambda+1} + 1$ for $1 \leq u < v \leq \overline{s}^{\lambda+1}$ assume that $\overline{m}_v^{\lambda+1} \geq \overline{m}_u^{\lambda+1} + 2$ for some $1 \leq u < v \leq \overline{s}^{\lambda+1}$. Then similar to Part (b) in Lemma 11 we have the situation with the blocks $A^{\lambda+1}$ and $B^{\lambda+1}$ in level $\lambda+1$, $\lambda \geq 1$:

$$\ldots B^{\lambda+1} \underbrace{A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_u^{\lambda+1}} B^{\lambda+1} \ldots B^{\lambda+1} \underbrace{A^{\lambda+1} \underbrace{A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_u^{\lambda+1}} A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_v^{\lambda+1} \geq \overline{m}_u^{\lambda+1}+2} B^{\lambda+1} \ldots$$

Substituting lower level blocks instead of some of $A^{\lambda+1}$ and $B^{\lambda+1}$ we rewrite the above sequence as follows:

$$\begin{aligned}
&\ldots \overbrace{B^\lambda \underset{i'}{A^\lambda} \ldots A^\lambda}^{B^{\lambda+1}} \underbrace{A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_u^{\lambda+1}} \overbrace{B^\lambda A^\lambda \ldots \underset{j'}{A^\lambda}}^{B^{\lambda+1}} \ldots B^{\lambda+1} \overbrace{B^\lambda A^\lambda \underset{i'+\rho}{A^\lambda} \ldots A^\lambda}^{A^{\lambda+1}} \underbrace{A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_u^{\lambda+1}} \\
&\overbrace{B^\lambda A^\lambda \ldots \underset{j'+\rho}{A^\lambda} A^\lambda}^{A^{\lambda+1}} \underbrace{A^{\lambda+1} \ldots A^{\lambda+1}}_{\overline{m}_v^{\lambda+1} - \overline{m}_u^{\lambda+1} - 1} B^{\lambda+1} \ldots
\end{aligned} \tag{31}$$

Definition of indices i', j' and i and j is similar to that in part (a). Namely, let i' be the index of the first Δ_ν in the first A^λ of the first $B^{\lambda+1}$, i.e., the first position in this first A^λ. Let j' be the last position of the last A^λ of the second $B^{\lambda+1}$. Furthermore let $i'+\rho$ be the first position of the second A^λ in $A^{\lambda+1}$ (see (31)). Notice that the sequences $\Delta_{i'} \ldots \Delta_{j'}$ and $\Delta_{i'+\rho} \ldots \Delta_{j'+\rho}$ are identical. Let $i < i'$ be the largest index with $\Delta_i \neq \Delta_{i+\rho}$ and $\Delta_\nu = \Delta_{\nu+\rho}$ for $\nu = i+1, \ldots, i'$. Such an index exists because block B^λ is the block preceding immediately position i' in $B^{\lambda+1}$ and A^λ is the block preceding immediately position $i'+\rho$ in $A^{\lambda+1}$ (see (31)). Furthermore, as shown in Part (a), $\Delta_{i+\rho} = \lceil a \rceil$ and $\Delta_i = \lfloor a \rfloor$. Let $j > j'$ be the smallest index with $\Delta_\nu = \Delta_{\nu+\rho}$ for $\nu = j'+1, \ldots, j-1$ and $\Delta_j \neq \Delta_{j+\rho}$.

We have

$$\begin{array}{l}
\ldots \underset{j'}{A^\lambda} B^\lambda A^\lambda \ldots \\
\ldots \underset{j'+\rho}{A^\lambda} A^\lambda \ldots
\end{array}$$

By expanding $B^\lambda A^\lambda$ and A^λ, which follow $\underset{j'}{A^\lambda}$ and $\underset{j'+\rho}{A^\lambda}$, respectively, we get

$$\begin{aligned}
B^\lambda A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} B^{\lambda-1} A^{\lambda-1} \\
A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} A^{\lambda-1}
\end{aligned}$$

$$\begin{aligned}
B^\lambda A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} B^{\lambda-2} A^{\lambda-2} \ldots A^{\lambda-2} B^{\lambda-2} A^{\lambda-2} \\
A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} B^{\lambda-2} A^{\lambda-2} \ldots A^{\lambda-2} B^{\lambda-2}
\end{aligned}$$

$$\vdots$$

$$\begin{aligned}
B^\lambda A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} B^{\lambda-2} A^{\lambda-2} \ldots A^{\lambda-2} B^{\lambda-3} \ldots B^2 A^2 \ldots A^2 B^1 \\
A^\lambda &= B^{\lambda-1} A^{\lambda-1} \ldots A^{\lambda-1} B^{\lambda-2} A^{\lambda-2} \ldots A^{\lambda-2} B^{\lambda-3} \ldots B^2 A^2 \ldots A^2 A^1
\end{aligned}$$

Again $\Delta_j = \lfloor a \rfloor$ and $\Delta_{j+\rho} = \lceil a \rceil$ and the sequences

$$\Delta_{i+1}, \ldots, \Delta_{j-1} \text{ and } \Delta_{i+\rho+1}, \ldots, \Delta_{j+\rho-1}$$

are identical. Similar to Part (a), relation (30) holds leading to the contradiction $2 \leq 1$.

Symmetrically one can prove $\underline{m}_v^\lambda \leq \underline{m}_u^\lambda + 1$ for $1 \leq v < u \leq \underline{s}^\lambda$. □

We introduce the following notations. A block X^λ ($X^\lambda = A^\lambda$ or $X^\lambda = B^\lambda$) at some arbitrary level λ can be decomposed in a unique way into level 1 blocks $\Delta_\nu \in \{\lceil a \rceil, \lfloor a \rfloor\}$, i.e.,

$$X^\lambda = \Delta_i \Delta_{i+1} \ldots \Delta_j.$$

The cumulative Δ-value of X^λ is defined as

$$\begin{aligned} \Delta_{X^\lambda} = \sum_{\nu=i}^{j} \Delta_\nu = \left(\left\lceil \frac{n}{i-1} \right\rceil - \left\lceil \frac{n}{i} \right\rceil \right) + \left(\left\lceil \frac{n}{i} \right\rceil - \left\lceil \frac{n}{i+1} \right\rceil \right) + \cdots + \\ + \left(\left\lceil \frac{n}{j-1} \right\rceil - \left\lceil \frac{n}{j} \right\rceil \right) = \left\lceil \frac{n}{i-1} \right\rceil - \left\lceil \frac{n}{j} \right\rceil \end{aligned}$$

and its length is given by

$$l_{X^\lambda} = j - i + 1.$$

The values of Δ_{X^λ} and l_{X^λ} for $X^\lambda = B^\lambda$ and $X^\lambda = A^\lambda$ can be calculated level by level using the recursive formulas which follow from (22) and (26):

$$\Delta_{B^\lambda} = \Delta_{B^{\lambda-1}} + \left\lfloor \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rfloor \Delta_{A^{\lambda-1}}, \quad \Delta_{A^\lambda} = \Delta_{B^{\lambda-1}} + \left\lceil \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rceil \Delta_{A^{\lambda-1}} \tag{32}$$

and

$$l_{B^\lambda} = l_{B^{\lambda-1}} + \left\lfloor \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rfloor l_{A^{\lambda-1}}, \quad l_{A^\lambda} = l_{B^{\lambda-1}} + \left\lceil \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rceil l_{A^{\lambda-1}} \tag{33}$$

with initial values

$$\Delta_{B^1} = \lfloor a \rfloor, \quad \Delta_{A^1} = \lceil a \rceil, \quad l_{A^1} = l_{B^1} = 1. \tag{34}$$

Due to Lemma 10 and (24) the values e^λ and d^λ are calculated recursively by

$$e^\lambda = e^{\lambda-1} - \left\lfloor \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rfloor d^{\lambda-1} \quad \text{and} \quad d^\lambda = \left\lceil \frac{e^{\lambda-1}}{d^{\lambda-1}} \right\rceil d^{\lambda-1} - e^{\lambda-1} \tag{35}$$

with initial values

$$e^1 = a - \lfloor a \rfloor \text{ and } d^1 = \lceil a \rceil - a. \tag{36}$$

4 A Polynomial-Time Algorithm

In this section we formulate an algorithm for finding an integer $1 \leq k_{opt} \leq h_1$, which minimizes g under an assumption that $a > 1$. The idea of our algorithm is to calculate for each relevant level the corresponding oscillation area. The levels λ are considered one by one. In our description we assume that in the current level λ condition $e^\lambda \geq d^\lambda$ holds. The case $e^\lambda < d^\lambda$ is symmetric and the corresponding version of the algorithm which takes care of both cases can be easily derived.

The algorithm stops when an oscillation area contains at most two blocks. In this case k_{opt} can be identified easily. The algorithm can be described by a recursive procedure **Optimize**(λ, l, h), where l is a lower bound and h is an upper bound for the oscillation area at level λ.

Procedure Optimize(λ, l, h)
1. **If** $\lambda \geq 2$ **then** $h :=$**Rightboundary**$(\lambda - 1, l, h)$;
2. $l :=$**Updateleft**(λ, l, h);
3. $h :=$**Updateright**(λ, l, h);
4. Calculate the blocks $X_l^{\lambda+1}$ and $X_h^{\lambda+1}$, which can be of type $A^{\lambda+1}$ or $B^{\lambda+1}$ at level $\lambda + 1$ in which l and h are contained;
5. **If** at least one $(\lambda+1)$-block exists between $X_l^{\lambda+1}$ and $X_h^{\lambda+1}$, **then Optimize**$(\lambda+1, l, h)$
6. **else CalculateOptimum**$\left(\lambda + 1, X_l^{\lambda+1}, X_h^{\lambda+1}\right)$

Rightboundary$(\lambda - 1, l, h)$ provides an upper bound for the right boundary of the oscillation area for $\lambda \geq 2$. Such a boundary is provided by the last position of the last $A^{\lambda-1}A^{\lambda-1}$-subsequence in an oscillation area at level $\lambda - 1$. The procedure **Updateleft**(λ, l, h) and **Updateright**(λ, l, h) calculate the left boundary $\widehat{l}$ and right boundary $\widehat{h}$ of the oscillation area at the next higher level, given a lower bound l for $\widehat{l}$ and an upper bound h for $\widehat{h}$. Notice, that **Rightboundary**(λ, l, h) and **Updateright**(λ, l, h) with $\lambda \geq 2$ are different procedures. **Rightboundary**(λ, l, h) identifies the last position of the last $A^\lambda A^\lambda$ -subsequence in an oscillation area at level λ which defines the input h for the procedure **Updateright**(λ, l, h). **Rightboundary**(λ, l, h) is needed to cut off the $B^\lambda A^\lambda B^\lambda A^\lambda \cdots A^\lambda B^\lambda \cdots B^\lambda A^\lambda$ -part which under the assumption $e^\lambda \geq d^\lambda$ is not relevant for identifying an optimal solution.

The procedure **CalculateOptimum**$\left(\lambda, X_l^\lambda, X_h^\lambda\right)$ provides an optimal solution if only one block $X_l^\lambda = X_h^\lambda$ or two adjacent blocks X_l^λ and X_h^λ are left. If X_l^λ and X_h^λ coincide and are equal to B^λ, then $k - 1$ provides an optimal solution where k is the first index in X_l^λ. If X_l^λ and X_h^λ coincide and are equal to A^λ, then the last index in X_l^λ provides an optimal solution. If X_l^λ and X_h^λ are two adjacent blocks, then for $X_l^\lambda X_h^\lambda$ we have the four cases: $A^\lambda A^\lambda$, $A^\lambda B^\lambda$, $B^\lambda A^\lambda$, and $B^\lambda B^\lambda$. In the last two cases again $k-1$ provides an optimal solution where k is the first index in X_l^λ. In Case $A^\lambda A^\lambda$ the last index in X_h^λ provides an optimal solution and in Case $A^\lambda B^\lambda$ the last index in X_l^λ provides an optimal solution.

Minimize g is the main procedure which calculates K_{opt}.

Minimize g
1. $l := 1; h := h_1$;
2. $\lambda := 1$;
3. **Optimize**(λ, l, h)

In the first iteration of **Minimize g** the procedure **Optimize**$(1,1,h_1)$ is called in which l and h are calculated by **Updateleft**$(1,1,h_1)$ and **Updateright**$(11,h_1)$, respectively. Due to Lemma 11, the search can be narrowed to the subsequence $\Delta_l,\ldots,\Delta_{h-1}$, marked by (18), with Δ_l corresponding to the first occurrence of B^1 and Δ_{h-1} corresponding to the last occurrence of A^1. This is depicted in

$$\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{(20)}$$
$$B^1\overbrace{A^1\ldots A^1}\ldots B^1\overbrace{A^1\ldots A^1}B^1A^1\ldots B^1A^1\overbrace{B^1\ldots B^1}A^1\ldots\overbrace{B^1\ldots B^1}A^1$$
$$\updownarrow\qquad\qquad\qquad\updownarrow\qquad\qquad\qquad\qquad\qquad\updownarrow$$
$$\Delta_2\ldots\underbrace{\Delta_l\qquad\cdots\qquad\Delta_{h'}\qquad\qquad\cdots\qquad\qquad\Delta_{h-1}}\Delta_h\ldots\Delta_{h_1}$$

If $l<h-1$ then **Optimize**$(2,l,h)$ is called which first calls **Rightboundary**$(1,l,h)$ to find the last position h' of the last subsequence A^1A^1. Now the level 2 starts which is restricted to the range $\Delta_l,\ldots,\Delta_{h'}$ marked by (20).

Next we describe the procedures **Updateleft**(λ,l,h) and **Updateright**(λ,l,h), which by binary search provides the beginning and end of the oscillation area at level $\lambda+1$; $\left[l',l''\right]$ and $\left[h',h''\right]$ are the corresponding search intervals. The description of **Updateleft**(λ,l,h) and **Updateright**(λ,l,h) for $\lambda=1$ differs slightly from the description for $\lambda>1$.

First we formulate **Updateleft**(λ,l,h) for $\lambda=1$, afterwards we describe **Updateright**(λ,l,h) for $\lambda=1$ and finally the update procedures for $\lambda>1$.

Procedure Updateleft$(1,1,h_1)$
1. $l':=1$; $l'':=h_1$; $\Delta_1:=\infty$;
2. **While** $l''\geq l'+2$ **do**
3. $\quad j:=\left\lfloor\frac{l'+l''}{2}\right\rfloor$;
4. $\quad$ **If** $\Delta_j>\lceil a\rceil$ **then** $l':=j$;
5. $\quad$ **If** $\Delta_j\leq\lfloor a\rfloor$ **then** $l'':=j$;
6. $\quad$ **If** $\Delta_j=\lceil a\rceil$ **then**
7. $\qquad$ Find the maximal $\lceil a\rceil$-block

$$\begin{array}{ccccc}\Delta_{j'} & & \Delta_j & & \Delta_{j''}\\ \lceil a\rceil & \ldots & \lceil a\rceil & \ldots & \lceil a\rceil\end{array}$$

$\qquad$ containing Δ_j;
8. $\qquad$ **If** $\Delta_{j''+1}>\lceil a\rceil$ **then** $l':=j''+1$;
9. $\qquad$ **If** $\Delta_{j''+1}\leq\lfloor a\rfloor$ **then**
10. $\qquad\quad$ **if** $j''+1<l''$ **then** $l'':=j''+1$
11. $\qquad\quad$ **else if** $\Delta_{j'-1}>\lceil a\rceil$ **then return** $j''+1$
12. $\qquad\qquad$ **else** $l'':=j'-1$;
13. **Return** l''

The procedure used in Step 7 for finding the maximal $\lceil a\rceil$-block to which Δ_j belongs will be discussed in Section 5.

If there is no exit from the while-loop 2-12 by the **return** in Line 11, then during the performance of the while-loop always the inequalities $\Delta_{l'}>\lceil a\rceil$ and $\Delta_{l''}\leq\lfloor a\rfloor$ are satisfied. Therefore by Lemma 9 $l'<l''$ is always satisfied. Furthermore in this case the while-loop ends when $l''=l'+1$ is reached. At that point l'' marks the start of the oscillation area. Notice, that $\Delta_{l'}>\lceil a\rceil$ implies that $\Delta_\nu\geq\lceil a\rceil$ for all $\nu\leq l'$.

If on the other hand $\Delta_{j''+1} \le \lfloor a \rfloor$, $l'' \le j'' + 1$ (which implies $l'' = j'' + 1$), and $\Delta_{j'-1} > \lceil a \rceil$, then again l'' marks the start of the oscillation area.

Notice, that **Updateleft**$(1, 1, h_1)$ returns an index l with $\Delta_l \le \lfloor a \rfloor$. If $\Delta_l < \lfloor a \rfloor$ then $l - 1$ provides an optimal solution. Otherwise, $\Delta_l = \lfloor a \rfloor$, i.e., l is the index of a B-block.

The procedure **Updateright**(λ, l, h) for $\lambda = 1$ is shown below. It is symmetric to **Updateleft**(λ, l, h) for $\lambda = 1$.

Procedure Updateright$(1, 1, h_1)$
1. $h' := 1$; $h'' := h_1$; $\Delta_1 := \infty$;
2. **While** $h'' \ge h' + 2$ **do**
3. $\quad j := \left\lfloor \frac{h'+h''}{2} \right\rfloor$;
4. $\quad$ **If** $\Delta_j < \lfloor a \rfloor$ **then** $h'' := j$;
5. $\quad$ **If** $\Delta_j \ge \lceil a \rceil$ **then** $h' := j$;
6. $\quad$ **If** $\Delta_j = \lfloor a \rfloor$ **then**
7. $\qquad$ Find the maximal $\lfloor a \rfloor$-block

$$\underset{\lfloor a \rfloor}{\Delta_{j'}} \ \ldots \ \underset{\lfloor a \rfloor}{\Delta_j} \ \ldots \ \underset{\lfloor a \rfloor}{\Delta_{j''}}$$

$\qquad$ containing Δ_j;
8. $\qquad$ **If** $\Delta_{j'-1} < \lfloor a \rfloor$ **then** $h'' := j' - 1$;
9. $\qquad$ **If** $\Delta_{j'-1} \ge \lceil a \rceil$ **then**
10. $\qquad\quad$ **if** $j' - 1 > h'$ **then** $h' := j' - 1$
11. $\qquad\quad$ **else if** $\Delta_{j''+1} < \lfloor a \rfloor$ **then return** $j' - 1$
12. $\qquad\qquad$ **else** $h' := j'' + 1$;
13. **Return** h'

Updateright$(1, 1, h_1)$ returns an index h with $\Delta_h \ge \lceil a \rceil$. If $\Delta_h > \lceil a \rceil$, then h provides an optimal solution. Otherwise, $\Delta_h = \lceil a \rceil$, i.e. h is the index of an A-block.

Next we describe **Updateleft**$(\lambda + 1, l, h)$ for $\lambda \ge 1$. The corresponding oscillation area has the form (25) where block A^λ decreases the objective function g by d^λ and block B^λ increases g by e^λ. l is the first index in (25) and h is the last index. We assume that $e^\lambda \ge d^\lambda$. The case $e^\lambda \le d^\lambda$ is treated symmetrically. In the case $e^\lambda \ge d^\lambda$ the last occurrence of an $A^\lambda \ldots A^\lambda$-block with at least two A^λ repeated defines a right boundary of the relevant $B^\lambda A^\lambda \ldots A^\lambda$-area. The blocks $X^{\lambda+1}$ considered in **Updateleft**$(\lambda + 1, l, h)$ for $\lambda \ge 1$ have the form

$$X^{\lambda+1} = B^\lambda \underbrace{A^\lambda \ldots A^\lambda}_{m_{X^{\lambda+1}}}$$

and its type depends on the number $m_{X^{\lambda+1}}$ of repetitions of A^λ. **Updateleft**$(\lambda + 1, l, h)$ for $\lambda \ge 1$ is similar to **Updateleft**$(1, 1, h_1)$, see the description below.

Procedure Updateleft$(\lambda + 1, l, h)$
1. $l' := l$; $l'' := h$;
2. Calculate the blocks $X_{l'}^{\lambda+1}$ and $X_{l''}^{\lambda+1}$ in which $\Delta_{l'}$ and $\Delta_{l''}$ are contained;
3. $l' :=$ the last index in $X_{l'}^{\lambda+1}$; $l'' :=$ the first index in $X_{l''}^{\lambda+1}$;
4. **While** a $(\lambda + 1)$-block exists between $X_{l'}^{\lambda+1}$ and $X_{l''}^{\lambda+1}$ **do**

5. $j := \left\lfloor \frac{l'+l''}{2} \right\rfloor$;
6. Let $X_j^{\lambda+1}$ be the block to which Δ_j belongs;
7. **If** $m_{X_j^{\lambda+1}} > \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ **then** $l' :=$ the last index in $X_j^{\lambda+1}$;
8. **If** $m_{X_j^{\lambda+1}} \leq \left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor$ **then** $l'' :=$ the first index in $X_j^{\lambda+1}$;
9. **If** $m_{X_j^{\lambda+1}} = \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ **then**
10. find the maximal sequence of repetitions of the block $X^{\lambda+1} = X_j^{\lambda+1}$:

$$C^{\lambda+1}X^{\lambda+1}\ldots \underset{j}{X^{\lambda+1}} \ldots X^{\lambda+1}D^{\lambda+1} \left(C^{\lambda+1}, D^{\lambda+1} \neq X^{\lambda+1}\right);$$

11. **If** $m_{D^{\lambda+1}} > \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ **then**
replace l' by the last index in $D^{\lambda+1}$ and set $X_{l'}^{\lambda+1} = D^{\lambda+1}$;
12. **If** $m_{D^{\lambda+1}} \leq \left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor$ **then**
13. **if** the first index in $D^{\lambda+1}$ is smaller than l''
then let l'' be the first index in $D^{\lambda+1}$ and set $X_{l''}^{\lambda+1} = D^{\lambda+1}$
14. **else if** $m_{C^{\lambda+1}} > \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ **then return** the first index in $D^{\lambda+1}$
15. **else** replace l'' by the first index in $C^{\lambda+1}$ and set $X_{l''}^{\lambda+1} = C^{\lambda+1}$;
17. **Return** l''

The procedure for identifying block $X_j^{\lambda+1}$ to which Δ_j belongs and for calculating the first and last indices of $X_j^{\lambda+1}$ will be discussed in Section 5.

Notice, that during the performance of the procedure **Updateleft**$(\lambda+1, l, h)$ for $\lambda \geq 1$ the inequalities $m_{X_{l''}^{\lambda+1}} \leq \left\lfloor \frac{e^\lambda}{d^\lambda} \right\rfloor$ and $m_{X_{l'}^{\lambda+1}} > \left\lceil \frac{e^\lambda}{d^\lambda} \right\rceil$ always hold. Therefore $X_{l''}^{\lambda+1}$ is to the right and $X_{l'}^{\lambda+1}$ is to the left of the $X^{\lambda+1}, \ldots, X^{\lambda+1}$-block. This implies that $C^{\lambda+1}$ and $D^{\lambda+1}$ always exist.

The procedure **Updateright**$(\lambda+1, l, h)$ is symmetric to **Updateleft**$(\lambda+1, l, h)$.

It remains to describe the procedure **Rightboundary**(λ, l, h) which calculates the last position of the last $A^\lambda A^\lambda$-subsequence in an oscillation area at level λ, $\lambda \geq 2$, if such a position exists. Again binary search is applied to find this position. More specifically, we calculate a position j in a block in $A^\lambda A^\lambda B^\lambda A^\lambda B^\lambda A^\lambda \ldots$ where $A^\lambda A^\lambda$ is the last occurrence of two consecutive A^λ. In the second step the last position in $A^\lambda A^\lambda$ must be calculated. This is easy if Δ_j belongs to $A^\lambda A^\lambda$. Otherwise, the first occurrence of $B^\lambda A^\lambda$ in $A^\lambda A^\lambda B^\lambda A^\lambda B^\lambda A^\lambda \ldots \underset{j}{B^\lambda} A^\lambda$ must be identified. A corresponding method is discussed in Section 5.3.

We start by calculating the last index h' of the block containing l and the first index h'' of the block containing h. If there exists another block between these two blocks, we calculate the block X_j^λ containing $j := \left\lceil \frac{h'+h''}{2} \right\rceil$. Then we distinguish the two possible cases depending on whether X_j^λ is of type B or A. If X_j^λ is of type B,

then the following three subcases are considered:

Case 1: $\overset{h'}{A^\lambda B^\lambda} \ldots A^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda \overset{j}{B^\lambda}$ (the fragment $\Delta_{h'}, \ldots, \Delta_j$ is covered by $A^\lambda B^\lambda$-sequences)

Case 2: $A^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda$ (nearest repeated A^λ which appear to the left of X_j^λ)

Case 3: $B^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda$ (nearest repeated B^λ which appear to the left of X_j^λ)

If Case 3 holds, then due to Part (a) in the proof of Theorem 2 there is no $A^\lambda A^\lambda$-subsequence to the right of X_j^λ. In this case we set h'' equal to the first index in the j-block X_j^λ. In the other two cases no $B^\lambda B^\lambda$-subsequence occurs to the left of X_j^λ but an $A^\lambda A^\lambda$-subsequence may occur to the right of X_j^λ and we set h' equal the last index in the j-block X_j^λ.

The case $X_j^\lambda = A^\lambda$ is treated symmetrically.

In either case we proceed with the new values h' and h'' and continue until there is no other block between the h'-block and the h''-block. If both blocks are B^λ-blocks then we set h equal to h'. Otherwise we set h equal to h''. Now we have reached a situation in which no $B^\lambda B^\lambda$ occurs to the left of the h-block and we calculate the last position of the last $A^\lambda A^\lambda$ which occurs before the h-block. The algorithm is summarized below.

Procedure Rightboundary(λ, l, h)

1. $h' :=$ last index of the block containing l;
2. $h'' :=$ first index of the block containing h;
3. **While** at least one block exists between the h'-block and the h''-block **do**
4. $\quad j := \left\lceil \frac{h'+h''}{2} \right\rceil$;
5. $\quad X_j^\lambda =$ the block containing Δ_j;
6. $\quad$ **If** X_j^λ is a B-block **then**
7. $\quad\quad$ **if**

$$B^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda \overset{j}{B^\lambda}$$

8. $\quad\quad$ **then** h'' := the first index in the j-block B^λ
9. $\quad\quad$ **else** h' := the last index in the j-block B^λ
10. $\quad$ **else if**

$$B^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda \overset{j}{A^\lambda}$$

11. $\quad\quad$ **then** h'' := the first index in the j-block A^λ
12. $\quad\quad$ **else** h' := the last index in the j-block A^λ;
13. **If** $X_{h'}^\lambda = B^\lambda$ and $X_{h''}^\lambda = B^\lambda$ hold, **then** $h := h'$
14. **else** $h := h''$;
15. **Return** the last position of the last $A^\lambda A^\lambda$-block which does not end later than the h-block

Steps 13 and 14 are such that h belongs to the block in the $B^\lambda A^\lambda B^\lambda A^\lambda \ldots$-area after the last occurrence of $A^\lambda A^\lambda$. Notice that procedure **Rightboundary** always deals with the blocks $X_{h'}^\lambda$ and $X_{h''}^\lambda$ in different positions in the λ-oscillation area so that these two blocks never coincide.

An efficient procedure implementing Steps 7,10, and 15 is presented in the next section.

5 Calculating Block Boundaries

In Step 7 of the procedure **Updateleft**$(1,1,h_1)$ and in a corresponding step in **Updateright**$(1,1,h_1)$ one has to calculate the boundaries of an $AA\ldots A$-block and $BB\ldots B$-block, respectively. Similar calculations are needed for higher level **Updateleft** and **Updateright** procedures. We also have to identify at the current level λ the block X_j^λ in which Δ_j is contained and to find the first and the last index of X_j^λ. This is done by first calculating the boundaries of the level 2 blocks in which Δ_j is contained, then calculating the boundaries of the level 3 blocks in which Δ_j is contained, etc.

In the next two subsections it is shown how to calculate $A^\lambda A^\lambda \ldots A^\lambda$-block boundaries for level $\lambda = 1$ and for higher levels λ. $B^\lambda B^\lambda \ldots B^\lambda$-block boundaries can be calculated in a similar way. We also need to calculate the boundaries of an $A^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda$-block. A corresponding procedure is described in the third subsection.

5.1 Calculating Boundaries for $AA\ldots A$-Blocks at Level 1

To calculate the left boundary of an $AA\ldots A$-block at level 1 we consider the following two situations:

$$\begin{array}{lccccc} \text{Case 1}: & B & A & A & \ldots & A \\ & \Delta_{i-\delta} & \Delta_{i-\delta+1} & \Delta_{i-\delta+2} & \cdots & \Delta_i \end{array}$$

$$\begin{array}{lccccc} \text{Case 2}: & D & A & A & \ldots & A \\ & \Delta_{i-\delta} & \Delta_{i-\delta+1} & \Delta_{i-\delta+2} & \cdots & \Delta_i \end{array}$$

with $A = \lceil a \rceil$, $B = \lfloor a \rfloor$ and $D > \lceil a \rceil$ (cf. Lemma 9). In both cases

$$\begin{gathered} \left\lceil \tfrac{n}{i-\nu} \right\rceil - \left\lceil \tfrac{n}{i} \right\rceil = \left(\left\lceil \tfrac{n}{i-\nu} \right\rceil - \left\lceil \tfrac{n}{i-\nu+1} \right\rceil \right) + \left(\left\lceil \tfrac{n}{i-\nu+1} \right\rceil - \left\lceil \tfrac{n}{i-\nu+2} \right\rceil \right) + \ldots \\ \ldots + \left\lceil \tfrac{n}{i-1} \right\rceil - \left\lceil \tfrac{n}{i} \right\rceil = \lceil a \rceil \, \nu \quad \text{for } \nu = 0, \ldots, \delta. \end{gathered}$$

Thus,

$$\left\lceil \frac{n}{i-\nu} \right\rceil = \xi_i + \lceil a \rceil \, \nu \quad \text{for } \nu = 0, \ldots, \delta \tag{37}$$

with $\xi_i = \left\lceil \frac{n}{i} \right\rceil$ holds.

Additionally,

$$\left\lceil \frac{n}{i-(\delta+1)} \right\rceil = \xi_i + \lceil a \rceil \, \delta + \lfloor a \rfloor = \xi_i + \lceil a \rceil \, (\delta+1) - 1 \tag{38}$$

holds in Case 1 and

$$\left\lceil \frac{n}{i-(\delta+1)} \right\rceil = \xi_i + \lceil a \rceil \, (\delta+1) + x \quad \text{with } x \geq 1 \tag{39}$$

holds in Case 2.

Theorem 3 *(a) Case 1 holds if and only if (37) and (38) are satisfied. (b) Case 2 holds if and only if (37) and (39) are satisfied.*

Proof The necessary part has just been proved. It remains to show that (37) and (38) ((37) and (39)) are sufficient for Case 1 (Case 2).

In Case 1 in which (37) and (38) hold, subtracting $\left\lceil \frac{n}{i-\nu} \right\rceil = \xi_i + \lceil a \rceil \nu$ from $\left\lceil \frac{n}{i-(\nu+1)} \right\rceil = \xi_i + \lceil a \rceil (\nu+1)$ yields $\Delta_{i-\nu} = \lceil a \rceil$ for $\nu = 0, ..., \delta$ and $\Delta_{i-(\delta+1)} = \lfloor a \rfloor$.

In Case 2 in which (37) and (39) hold, a similar subtraction provides $\Delta_{i-\nu} = \lceil a \rceil$ for $\nu = 0, ..., \delta$ and $\Delta_{i-(\delta+1)} = \lceil a \rceil + x$ with $x \geq 1$. □

We give a geometric interpretation of Cases 1 and 2.

Consider first Case 1. Condition (37) is equivalent to

$$\frac{n}{i-\nu} \leq \xi_i + \lceil a \rceil \nu \text{ and } \frac{n}{i-\nu} > \xi_i + \lceil a \rceil \nu - 1. \tag{40}$$

Furthermore, (38) implies

$$\frac{n}{i-(\delta+1)} \leq \left\lceil \frac{n}{i-(\delta+1)} \right\rceil = \xi_i + \lceil a \rceil (\delta+1) - 1. \tag{41}$$

In this case the line $\xi_i + \lceil a \rceil \nu - 1$ intersects the hyperbola $\frac{n}{i-\nu}$ in one point $c' = b'$ or two points $c' < b'$.

Furthermore, the interval $\left[c', b'\right]$ contains the integer point $\delta + 1$ and $c \leq 0 < c'$ where in c the line $\xi_i + \lceil a \rceil \nu$ intersects the hyperbola. The situation with $c' < b'$ is depicted in Figure 2.

To calculate δ one has to find the smallest solution c' of the equation

$$\frac{n}{i-\nu} = \xi_i + \lceil a \rceil \nu - 1$$

which is equivalent to the quadratic equation

$$\lceil a \rceil \nu^2 + (\xi_i - 1 - i \lceil a \rceil)\nu + n - i(\xi_i - 1) = 0.$$

δ is calculated by

$$\delta = \left\lceil c' - 1 \right\rceil \tag{42}$$

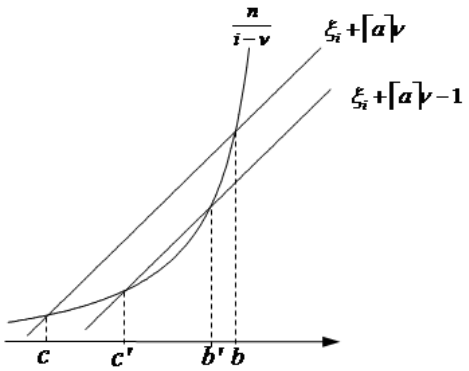

Fig. 2 Calculation of left boundary for A-repetitions

Now consider Case 2 in which (37) and (39) are satisfied. Then besides (40) the inequality

$$\left\lceil \frac{n}{i-(\delta+1)} \right\rceil > \xi_i + \lceil a \rceil (\delta+1)$$

holds. The hyperbola does or does not intersect the lower line $\xi_i + \lceil a \rceil \nu - 1$. In the latter case, due to (40) $c \leq 0 < b$, where c and b are intersection points of the hyperbola and the upper line $\xi_i + \lceil a \rceil \nu$, see Figure 2.

On the other hand, if the hyperbola intersects the lower line $\xi_i + \lceil a \rceil \nu - 1$, then due to (40) either $c \leq 0 < c'$ or $b' < 0 \leq b$, where c' and b' are intersection points of the hyperbola and the lower line $\xi_i + \lceil a \rceil \nu - 1$. Observe that if $c \leq 0 < c'$, then due to (39) and (40), interval $\left[c', b'\right]$ cannot contain an integer point. In both cases case $\delta = \lfloor b \rfloor$ where b is the largest solution of the equation

$$\frac{n}{i-\nu} = \xi_i + \lceil a \rceil \nu.$$

We conclude that in order to find out which case applies and to calculate the corresponding value δ one has to solve two quadratic equations.

The right boundary of an $A...A$-block can be calculated in a similar way.

5.2 Calculating Boundaries for $A^\lambda A^\lambda \dots A^\lambda$-Blocks

We describe how to calculate the left boundary of $A^\lambda A^\lambda \dots A^\lambda$-blocks at some level λ greater than 1. Right boundaries are calculated similarly.

$$
\begin{array}{lll}
\text{Case 1} & : & \\
A^\lambda\text{-sequence} & : & \\
\text{Case 2} & : & B^{\lambda-1}A^{\lambda-1}\dots A^{\lambda-1}
\end{array}
\left|\begin{array}{l}
\\
B^{\lambda-1}\\
A^{\lambda-1}\\
\uparrow\\
{\scriptstyle i-l_{A^\lambda}(\delta+1)}
\end{array}\right|
\begin{array}{l}
B^{\lambda-1}\\
A^{\lambda-1}\\
A^{\lambda-1}\\
\uparrow\\
{\scriptstyle j-l_{A^\lambda}(\delta+1)}
\end{array}
\left|\begin{array}{ll}
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots & \dots A^\lambda\\
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots & \dots A^\lambda\\
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots & \dots A^\lambda
\end{array}\right|
\begin{array}{l}
B^{\lambda-1}\\
B^{\lambda-1}\\
B^{\lambda-1}\\
\uparrow\\
{\scriptstyle i}
\end{array}
\left|\begin{array}{l}
A^{\lambda-1}\\
A^{\lambda-1}\\
A^{\lambda-1}\\
\uparrow\\
{\scriptstyle j}
\end{array}\right|
\begin{array}{l}
A^{\lambda-1}\dots A^{\lambda-1}\\
A^{\lambda-1}\dots A^{\lambda-1}\\
A^{\lambda-1}\dots A^{\lambda-1}
\end{array}
\tag{43}
$$

$$
\begin{array}{lll}
B^{\lambda-1}= & & B\ A\dots A\ A^{(2)}\dots A^{(2)}\dots A^{\lambda-2}\dots A^{\lambda-2}\\
A^{\lambda-1}= & B^{\lambda-2}B^{\lambda-3}\dots B & A\ A\dots A\ A^{(2)}\dots A^{(2)}\dots A^{\lambda-2}\dots A^{\lambda-2}
\end{array}
\tag{44}
$$

$$
\begin{array}{lll}
\text{Case 1} & : & \\
A^\lambda\text{-sequence} & : & \\
\text{Case 2} & : & B^{\lambda-1}A^{\lambda-1}\dots A^{\lambda-1}
\end{array}
\left|\begin{array}{l}
\\
B^{\lambda-2}\dots B
\end{array}\right|
\begin{array}{l}
BA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
AA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
\uparrow\\
{\scriptstyle i-l_{A^\lambda}(\delta+1)}
\end{array}
\left|\begin{array}{l}
B^{\lambda-2}\dots B\\
B^{\lambda-2}\dots B
\end{array}\right|
\begin{array}{l}
BA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
AA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
\uparrow\\
{\scriptstyle j-l_{A^\lambda}(\delta+1)}
\end{array}
\left|\begin{array}{l}
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots A^\lambda\\
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots A^\lambda\\
A^{\lambda-1}\dots A^{\lambda-1}A^\lambda\dots A^\lambda
\end{array}\right|
$$

$$
\begin{array}{lll}
\text{Case 1 (cont.)} & : & BA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
A^\lambda\text{-sequence (cont.)} & : & BA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
\text{Case 2 (cont.)} & : & BA\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
 & & \uparrow\\
 & & {\scriptstyle i}
\end{array}
\left|\begin{array}{ll}
B^{\lambda-2}\dots BA & A\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
B^{\lambda-2}\dots BA & A\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
B^{\lambda-2}\dots BA & A\dots A\dots A^{\lambda-2}\dots A^{\lambda-2}\\
\uparrow & \\
{\scriptstyle j} &
\end{array}\right|
\begin{array}{l}
A^{\lambda-1}\dots A^{\lambda-1}\\
A^{\lambda-1}\dots A^{\lambda-1}\\
A^{\lambda-1}\dots A^{\lambda-1}
\end{array}
\tag{45}
$$

To calculate the block B^λ (Case 1) or D^λ (Case 2) at the left boundary of a sequence $A^\lambda ... A^\lambda$ where

$$\begin{array}{rl} B^\lambda = & B^{\lambda-1}A^{\lambda-1}...A^{\lambda-1} \\ A^\lambda = & B^{\lambda-1}\ A^{\lambda-1}A^{\lambda-1}...A^{\lambda-1} \\ D^\lambda = B^{\lambda-1}A^{\lambda-1}... & A^{\lambda-1}\ A^{\lambda-1}A^{\lambda-1}...A^{\lambda-1} \end{array}$$

we compare the following sequences

$$\begin{array}{ll} \text{Case 1} & : B^\lambda A^\lambda ... A^\lambda A^\lambda \\ A^\lambda\text{-sequence} & : A^\lambda A^\lambda ... A^\lambda A^\lambda \\ \text{Case 2} & : D^\lambda \underbrace{A^\lambda ... A^\lambda}_{\delta} A^\lambda \end{array}$$

which can be written in the form (43). In (43), i (j) is a position of a Δ_i (Δ_j) value contained in the block $B^{\lambda-1}$ ($A^{\lambda-1}$) marked with i (j). Consequently, $i - l_{A^\lambda}(\delta+1)$ ($j - l_{A^\lambda}(\delta+1)$) are contained in the blocks with offset $l_{A^\lambda}(\delta+1)$, as indicated in (43). Recall that l_{A^λ} denotes the length of block A^λ, as introduced at the end of Section 3.

To define position $i - l_{A^\lambda}(\delta+1)$ compare the Case 2 sequence with the A^λ-sequence in representation (43). Scanning these sequences from right to left there is the first column where the blocks in the two sequences are different. This column is marked by $i - l_{A^\lambda}(\delta+1)$. Similarly, if we compare the Case 1 sequence with the A^λ-sequence in representation (43) by scanning these sequences from right to left there, we get the first column where the blocks are different marked by $j - l_{A^\lambda}(\delta+1)$. The precise position of $i - l_{A^\lambda}(\delta+1)$ and $j - l_{A^\lambda}(\delta+1)$ will be explained after a further decomposition of the blocks in the columns marked by i, j.

Notice that $A^{(1)} := A = \lceil a \rceil$ and $B^{(1)} := B = \lfloor a \rfloor$. Also, if $\lambda = 2$ then (43) provides directly the relevant structure and the precise positions $i - l_{A^\lambda}(\delta+1)$ and $j - l_{A^\lambda}(\delta+1)$ and correspondingly i and j.

If $\lambda \geq 3$ consider a further decomposition of $B^{\lambda-1}$ and $A^{\lambda-1}$ of the form (44), which has been derived in Section 3, see (29). Substituting (44) the comparison (43) can be extended to (45).

In representation (45), when scanning from right to left, $i - l_{A^\lambda}(\delta+1)$ marks the first position where the Case 2 sequence and the A^λ-sequence are different. Similarly, $j - l_{A^\lambda}(\delta+1)$ marks the first position where the Case 1 sequence and the A^λ-sequence are different.

To establish the boundaries of the $A^\lambda \dots A^\lambda$ block we need to derive formulas similar to (37)-(39). For Case 1 we compare the first two lines in (45) and conclude that

$$\left\lceil \frac{n}{j - l_{A^\lambda}\nu} \right\rceil = \xi_j + \Delta_{A^\lambda}\nu \quad \text{for } \nu = 0, \dots, \delta, \tag{46}$$

$$\left\lceil \frac{n}{j - l_{A^\lambda}(\delta+1)} \right\rceil = \xi_j + \Delta_{A^\lambda}(\delta+1) - 1, \tag{47}$$

$$\left\lceil \frac{n}{i - l_{A^\lambda}\nu} \right\rceil = \xi_i + \Delta_{A^\lambda}\nu \quad \text{for } \nu = 0, \dots, \delta, \tag{48}$$

where i and j are the positions marked in (45) and Δ_{A^λ} is the cumulative Δ-value of the block A^λ introduced at the end of Section 3. We demonstrate how condition

(47) can be derived; the other conditions are similar. Consider the difference between $\left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)} \right\rceil$ and ξ_j:

$$\begin{aligned}
\left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)} \right\rceil - \xi_j &= \left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)} \right\rceil - \left\lceil \frac{n}{j} \right\rceil \\
&= \left(\left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)} \right\rceil - \left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)+1} \right\rceil \right) + \\
&\quad + \left(\left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)+1} \right\rceil - \left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)+2} \right\rceil \right) + \\
&\quad + \cdots + \left(\left\lceil \frac{n}{j-1} \right\rceil - \left\lceil \frac{n}{j} \right\rceil \right) \\
&= \Delta_{j-l_{A^\lambda}(\delta+1)+1} + \Delta_{j-l_{A^\lambda}(\delta+1)+2} + \cdots + \Delta_j = \\
&= \Delta_{A^\lambda}(\delta+1) - \lceil a \rceil + \lfloor a \rfloor = \Delta_{A^\lambda}(\delta+1) - 1.
\end{aligned}$$

For Case 2 a comparison of the last two lines in (45) leads to

$$\left\lceil \frac{n}{i-l_{A^\lambda}\nu} \right\rceil = \xi_i + \Delta_{A^\lambda}\nu \quad \text{for } \nu = 0, \ldots, \delta \tag{49}$$

$$\left\lceil \frac{n}{i-l_{A^\lambda}(\delta+1)} \right\rceil = \xi_i + \Delta_{A^\lambda}(\delta+1) + 1 \tag{50}$$

$$\left\lceil \frac{n}{j-l_{A^\lambda}\nu} \right\rceil = \xi_j + \Delta_{A^\lambda}\nu \quad \text{for } \nu = 0, \ldots, \delta. \tag{51}$$

Theorem 4 *(a) Case 1 holds if and only if (46) to (48) are satisfied. (b) Case 2 holds if and only if (49) to (51) are satisfied.*

Proof Again it remains to prove that (46) to (48) and (49) to (51) are sufficient for Case 1 and Case 2, respectively. Notice that the sequences in (45) are part of the oscillation area of level $\lambda - 1$, i.e. they consist of blocks of type $A^{\lambda-1}$ and $B^{\lambda-1}$ only. The structure of this oscillation area is described by Theorem 2.
Part (a). Similar to the proof of Theorem 3 it follows from equalities (46) to (48) that

$$\left\lceil \frac{n}{j-l_{A^\lambda}(\nu+1)} \right\rceil - \left\lceil \frac{n}{j-l_{A^\lambda}\nu} \right\rceil = \Delta_{A^\lambda} \quad \text{for } \nu = 0, \ldots, \delta-1 \tag{52}$$

$$\left\lceil \frac{n}{j-l_{A^\lambda}(\delta+1)} \right\rceil - \left\lceil \frac{n}{j-l_{A^\lambda}\delta} \right\rceil = \Delta_{A^\lambda} - 1 \tag{53}$$

$$\left\lceil \frac{n}{i-l_{A^\lambda}(\delta+1)} \right\rceil - \left\lceil \frac{n}{i-l_{A^\lambda}\delta} \right\rceil = \Delta_{A^\lambda} \text{ for } \nu = 0, \ldots, \delta-1. \tag{54}$$

(52) to (54) imply that the first δ blocks to the left of the A^λ-block containing Δ_i and Δ_j must be A^λ-blocks and the next block is a B^λ-block.

The proof of **Part (b)** is similar. □

As discussed in the previous section conditions (46)-(48) and (49)-(51) can be checked and a corresponding δ can be calculated using the technique of the previous section.

5.3 Calculation of Boundaries for $A^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda$-Areas

To calculate the left boundary of an $A^\lambda B^\lambda A^\lambda B^\lambda \ldots A^\lambda B^\lambda$-area we proceed in a similar way as in the previous section.

For $\lambda \geq 3$ consider the sequences given by (55). These sequences can be expanded to (56) by substituting the expressions for B^λ and A^λ.

For $\lambda = 1$ the sequences (55) can be simplified by replacing A^λ by $\lceil a \rceil$ and B^λ by $\lfloor a \rfloor$. Similarly (56) can be simplified for the case $\lambda = 2$.

$$
\begin{array}{lll|l|l|l|l|l}
\text{Case 1} & : & & B^{\lambda} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda} & A^{\lambda} \\
A^{\lambda}B^{\lambda}\text{-sequence} & : & B^{\lambda} & A^{\lambda} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda} & A^{\lambda} \\
\text{Case 2} & : & A^{\lambda} & A^{\lambda} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda} & A^{\lambda} \\
 & & \uparrow & \uparrow & & \uparrow & \uparrow \\
 & & {\scriptstyle i-(l_{A^{\lambda}}+l_{B^{\lambda}})(\delta+1)} & {\scriptstyle j-(l_{A^{\lambda}}+l_{B^{\lambda}})(\delta+1)} & & {\scriptstyle i} & {\scriptstyle j}
\end{array}
\tag{55}
$$

$$
\begin{array}{lll|l|l|l|l|l}
\text{Case 1} & : & & & B^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1}B^{\lambda-1} & A^{\lambda-1}\ldots A^{\lambda-1} \\
A^{\lambda}B^{\lambda}\ldots & : & & B^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1} & A^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1}B^{\lambda-1} & A^{\lambda-1}\ldots A^{\lambda-1} \\
\text{Case 2} & : & B^{\lambda-1} & A^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1} & A^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1} & B^{\lambda}A^{\lambda}\ldots B^{\lambda}A^{\lambda} & B^{\lambda-1}A^{\lambda-1}\ldots A^{\lambda-1}B^{\lambda-1} & A^{\lambda-1}\ldots A^{\lambda-1} \\
 & & & \uparrow & \uparrow & & \uparrow & \uparrow \\
 & & & {\scriptstyle i-(l_{A^{\lambda}}+l_{B^{\lambda}})(\delta+1)} & {\scriptstyle j-(l_{A^{\lambda}}+l_{B^{\lambda}})(\delta+1)} & & {\scriptstyle i} & {\scriptstyle j}
\end{array}
\tag{56}
$$

Sequences (56) lead to the following case dependent equations which are similar to (46)-(48) and (49)-(51) and are necessary and sufficient for the corresponding situations.

Case 1:

$$\left\lceil \frac{n}{j-(l_{A^\lambda}+l_{B^\lambda})\nu} \right\rceil = \xi_j + (\Delta_{A^\lambda}+\Delta_{B^\lambda})\nu \quad \text{for } \nu = 0,\dots,\delta \tag{57}$$

$$\left\lceil \frac{n}{j-(l_{A^\lambda}+l_{B^\lambda})(\delta+1)} \right\rceil = \xi_j + (\Delta_{A^\lambda}+\Delta_{B^\lambda})(\delta+1) - 1 \tag{58}$$

$$\left\lceil \frac{n}{i-(l_{A^\lambda}+l_{B^\lambda})\nu} \right\rceil = \xi_i + (\Delta_{A^\lambda}+\Delta_{B^\lambda})\nu \quad \text{for } \nu = 0,\dots,\delta \tag{59}$$

Case 2:

$$\left\lceil \frac{n}{i-(l_{A^\lambda}+l_{B^\lambda})\nu} \right\rceil = \xi_i + (\Delta_{A^\lambda}+\Delta_{B^\lambda})\nu \quad \text{for } \nu = 0,\dots,\delta \tag{60}$$

$$\left\lceil \frac{n}{i-(l_{A^\lambda}+l_{B^\lambda})(\delta+1)} \right\rceil = \xi_i + (\Delta_{A^\lambda}+\Delta_{B^\lambda})(\delta+1) + 1 \tag{61}$$

$$\left\lceil \frac{n}{j-(l_{A^\lambda}+l_{B^\lambda})\nu} \right\rceil = \xi_j + (\Delta_{A^\lambda}+\Delta_{B^\lambda})\nu \quad \text{for } \nu = 0,\dots,\delta. \tag{62}$$

As before the two cases can be checked and a corresponding δ can be calculated by solving two quadratic equations.

6 Complexity

The complexity of the overall algorithm **Minimize g** which identifies an integer k_{opt} minimizing $g(k)$ can be estimated as follows.

In each level λ the blocks A^λ and B^λ contain one $B^{\lambda-1}$-block and at least one $A^{\lambda-1}$-block of the previous level. Therefore the length of level $\lambda+1$ blocks is bounded from below by 2^λ. On the other hand, the length of the oscillation area is not increasing which implies that the recursive procedure **Optimize** performs no more than $O(\log n)$ recursive calls because the algorithm stops when reaching a level in which the oscillation area has no more than two blocks. Using the recursive formulas (32) to (36) we calculate the values $\Delta_{A^\lambda}, \Delta_{B^\lambda}, l_{A^\lambda}$ and l_{B^λ} also in time $O(\log n)$.

To identify the level λ block in which j is contained, we calculate for each level $\nu = 2,\dots,\lambda$ the first and the last indices of the level ν block in which j is contained. If j is contained in a $B^{\nu-1}$-block or $A^{\nu-1}$-block at level $\nu-1$ with known first and last indices, we can find out in constant time whether j is contained in a B^ν-block B_j^ν or in a A^ν-block A_j^ν at level ν. Furthermore, the first and the last indices of the level ν block can be calculated in time $O(1)$. All this can be accomplished using the techniques described in Section 5. Thus, the first and the last indices of the level $\nu = 2,\dots,\lambda$ blocks containing j can be calculated in $O(\log n)$ time. The overall time complexity of each of the procedures **Updateleft** and **Updateright** is $O(\log^2 n)$ because the while-loops in these procedures are iterated at most $O(\log n)$ times. Similarly, the procedure **Rightboundary** has complexity $O(\log^2 n)$. We conclude that the main procedure **Optimize**$(1,1,h_1)$ which performs at most $\log n$ recursive calls has complexity $O(\log^3 n)$ which is the complexity of the algorithm **Minimize g** as well.

7 Concluding Remarks

In this paper we have resolved an open question posed in [9] for the flow-shop batching problem with equal processing times and equal setup times, which is formulated as an integer non-linear programming problem of one variable k with a solution region given by the natural numbers not exceeding n. The objective function is of the form $g(k) = ak + \left\lceil \frac{n}{k} \right\rceil$. For that problem we have developed a polynomial-time algorithm for finding an integer solution k_{opt}.

A non-integer optimum of the relaxed problem of minimizing the continuous function $f(k) = ak + \frac{n}{k}$ is given by $k = \sqrt{\frac{n}{a}}$ which can be determined easily by standard calculus techniques. On the other hand, an integer optimum of $g(k)$ cannot be obtained by simple rounding since this optimum can be quite far from $\sqrt{\frac{n}{a}}$.

While the earlier algorithms have time complexities $O(n)$ [3,6] and $O(\sqrt{n})$ [9], which are both exponential with respect to the binary encoded input length, the complexity of the new algorithm is $O\left(\log^3 n\right)$. The main challenges of developing a fast solution algorithm involve identification of a complex discrete periodic structure, finding out how it can be described in mathematical terms using a recursive representation and proving that the identified structure and the algorithm based on this structure are correct.

To the best of our knowledge, the technique developed does not have similar counterparts in the optimization literature. However, we believe it has potential to provide solutions to a range of high-multiplicity optimization problems, in particular to various batching problems with equal processing which can be formulated in a form similar to (3):

(i) the single machine batching problem to minimize the sum of completion times [4, 7,8,10,11],
(ii) the open-shop problem to minimize the makespan [5],
(iii) the job-shop problem to minimize the makespan [3].

For all above problems only pseudo-polynomial algorithms are known except for problem (i) for which Shallcross [11] has developed a polynomial-time algorithm with time complexity depending not only on $\log n$, but also on $\log p$ and $\log s$. The approach used in that paper is quite different from ours.

An interesting open question is establishing a link between the periodic structure derived for $g(k)$ and the continuos optimum $\sqrt{\frac{n}{a}}$. We suspect that at least one of the numbers $\left\lfloor \sqrt{\frac{n}{a}} \right\rfloor$ or $\left\lceil \sqrt{\frac{n}{a}} \right\rceil$ is close to the highest level block found by our algorithm. Proving this could lead to a faster algorithm of time complexity $O\left(\log^2 n\right)$.

Acknowledgements This research was partly supported by the EPSRC funded project EP/D059518. The authors are grateful to an anonymous referee for his/her comments that considerably improved the paper.

References

1. Brucker, P. (2004). Scheduling Algorithms, Springer, Berlin.
2. Cheng, T.C.E., Lin, B.M.T, & Toker, A. (2000). Makespan minimization in the two-machine flow-shop batch scheduling problem. Naval Research Logistics, 47, 128-144.
3. Mosheiov, G., & Oron, D., (2005). A note on flow-shop and job-shop batch scheduling with identical processing time jobs. European Journal of Operational Research. 161, 285-291.

4. Mosheiov, G., & Oron, D., (2006). Single machine scheduling with batch-dependent setup times. Information Processing Letters, 98, 73-78.
5. Mosheiov, G., & Oron, D., (2008). Open-shop batch scheduling with identical jobs. European Journal of Operational Research, 187, 1282-1292.
6. Mosheiov, G., Oron, D., & Ritov, Y. (2004). Flow-shop batch scheduling with identical processing time jobs. Naval Research Logistics, 51, 783-799.
7. Mosheiov, G., Oron, D., & Ritov, Y. (2005). Minimizing flow-time on a single machine with integer batch sizes. Operations Research Letters, 23, 497-205.
8. Naddeff, D., & Santos, C. (1988). One-pass batching algorithms for the one-machine batching problem. Discrete Applied Mathematics, 21, 133-145.
9. Ng, C.T., & Kovalyov, M.Y. (2007). Batching and scheduling in a multi-machine flow shop. Journal of Scheduling, 10, 353-364.
10. Santos, C., & Magazine, M. (1985). Batching in single operation manufacturing systems. Operations Research Letters, 4, 99-103.
11. Shallcross, D.F. (1992). A polynomial algorithm for a one machine batching problem. Operations Research Letters, 11, 213-218.