Theory Comput Syst (2018) 62:162–191
DOI 10.1007/s00224-017-9794-5

# Tighter Bounds and Optimal Algorithms for All Maximal $\alpha$-gapped Repeats and Palindromes

## Finding All Maximal $\alpha$-gapped Repeats and Palindromes in Optimal Worst Case Time on Integer Alphabets

**Paweł Gawrychowski[1] · Tomohiro I[2] · Shunsuke Inenaga[3] · Dominik Köppl[4] · Florin Manea[5]**

Published online: 15 August 2017


**Abstract** An $\alpha$-gapped repeat ($\alpha \geq 1$) in a word $w$ is a factor $uvu$ of $w$ such that $|uv| \leq \alpha\,|u|$; the two occurrences of $u$ are called *arms* of this $\alpha$-gapped repeat. An $\alpha$-gapped repeat is called *maximal* if its arms cannot be extended simultaneously with the same character to the right nor to the left. We show that the number of all maximal $\alpha$-gapped repeats occurring in words of length $n$ is upper bounded by $18\alpha n$. In the

Parts of this work have already been presented at 33rd International Symposium on Theoretical Aspects of Computer Science [13] and at the 20th International Symposium on Fundamentals of Computation Theory [10].

This article is part of the Topical Collection on *Theoretical Aspects of Computer Science*

✉ Paweł Gawrychowski
gawry@mimuw.edu.pl

Tomohiro I
tomohiro@ai.kyutech.ac.jp

Shunsuke Inenaga
inenaga@inf.kyushu-u.ac.jp

Dominik Köppl
dominik.koeppl@tu-dortmund.de

Florin Manea
flm@informatik.uni-kiel.de

[1] Institute of Informatics, University of Warsaw, Warsaw, Poland

[2] Department of Artificial Intelligence, Kyushu Institute of Technology, Fukuoka, Japan

[3] Department of Informatics, Kyushu University, Fukuoka, Japan

[4] Department of Computer Science, TU Dortmund, Dortmund, Germany

[5] Department of Computer Science, Kiel University, Kiel, Germany

case of $\alpha$-gapped palindromes, i.e., factors $uvu^{\mathsf{T}}$ with $|uv| \leq \alpha|u|$, we show that the number of all maximal $\alpha$-gapped palindromes occurring in words of length $n$ is upper bounded by $28\alpha n + 7n$. Both upper bounds allow us to construct algorithms finding all maximal $\alpha$-gapped repeats and/or all maximal $\alpha$-gapped palindromes of a word of length $n$ on an integer alphabet of size $n^{\mathcal{O}(1)}$ in $\mathcal{O}(\alpha n)$ time. The presented running times are optimal since there are words that have $\Theta(\alpha n)$ maximal $\alpha$-gapped repeats/palindromes.

**Keywords** Combinatorics on words · Counting algorithms

## 1 Introduction

Gapped repeats and palindromes are repetitive structures occurring in words that were investigated extensively within theoretical computer science (see, e.g., [3, 5–8, 10, 14, 17–19, 22] and the references therein) with motivation coming especially from the analysis of DNA and RNA structures, modelling different types of tandem and interspersed repeats as well as hairpin structures; such structures are important in analyzing the structural and functional information of the genetic sequences (see, e.g., [3, 14, 18]).

Let $w^{\mathsf{T}}$ denote the ***reversed word*** of a word $w$. An $\alpha$-gapped repeat (respectively, an $\alpha$-gapped palindrome) is a factor of the form $uvu$ (respectively, $uvu^{\mathsf{T}}$) with $|uv| \leq \alpha|u|$, for a real number $\alpha \geq 1$. These are natural generalization of classical repetitive and palindromic structures in a word: 1-gapped repeats and 1-gapped palindromes are respectively equivalent to squares and even palindromes, which are very well known and studied structures. Also, 2-gapped repeats and 2-gapped palindromes are respectively called long-armed pairs and long-armed palindromes.

The problem of searching for gapped repeats and palindromes in words is not so new (see [3, 14, 17]), and different solutions were proposed depending on the type of restrictions imposed on the gap. Kolpakov and Kucherov [18] introduced the notion of long-armed palindromes (equivalently, 2-gapped palindromes), and showed how to compute the set $\mathcal{G}_2^{\mathsf{T}}(w)$ of all maximal 2-gapped palindromes in $\mathcal{O}(n+|\mathcal{G}_2^{\mathsf{T}}(w)|)$ time for an input word $w$ of length $n$ over a constant alphabet. They left the question open how large $\left|\mathcal{G}_2^{\mathsf{T}}(w)\right|$ can actually be. In [19], Kolpakov et al. introduced the notion of $\alpha$-gapped repeats, and showed that the set $\mathcal{G}_\alpha(w)$ of all maximal $\alpha$-gapped repeats can be computed in $\mathcal{O}(\alpha^2 n+|\mathcal{G}_\alpha(w)|)$ time for integer alphabets. They further proved that $|\mathcal{G}_\alpha(w)| = \mathcal{O}(\alpha^2 n)$, and that the number of all maximal factors with exponents in $(1+\delta, 2]$ for a $\delta \in (0, 1]$, so-called $\delta$-subrepetitions, is bounded by the number of all maximal $1/\delta$-gapped repeats. In their article, they posed two open questions concerning the bounds they computed:

- Closing the gap between the upper bound $\mathcal{O}(\alpha^2 n)$ and the lower bound $\Omega(\alpha n)$ for the number of maximal $\alpha$-gapped repeats, and
- Developing a more efficient algorithm.

Problems like how many maximal $\alpha$-gapped repeats or palindromes can a word of length $n$ contain, how efficiently can we compute the set of all maximal $\alpha$-gapped

repeats or palindromes in a word, how efficiently can we compute the $\alpha$-gapped repeat or palindrome with the longest arm, were already investigated [3, 7, 10, 22]: [10] showed how to compute the longest $\alpha$-gapped repeat/palindrome in $\mathcal{O}(\alpha n)$ time, [8] showed how to compute a series of data structures that can give the longest 2-gapped repeat/palindrome that starts at each position (and the results generalize easily to arbitrary $\alpha$), Tanimura et al. [22] gave an $\mathcal{O}(\alpha n + |\mathcal{G}_\alpha(w)|)$-time solution to find all maximal $\alpha$-gapped repeats for an input word over constant alphabets. Finally, in August 2015, the fourth author of this paper announced on the Stringmasters webpage that the bound on the number of all maximal $\alpha$-gapped repeats and palindromes is indeed $\mathcal{O}(\alpha n)$; together with [22], this leads to an optimal algorithm for solving the problem of finding all maximal $\alpha$-gapped repeats in the particular case of constant alphabets. This announcement was followed by Crochemore et al. [7] who confirmed the bound $|\mathcal{G}_\alpha(w)| = \mathcal{O}(\alpha n)$; additionally, they presented an algorithm computing all maximal $\alpha$-gapped repeats for constant alphabets in $\mathcal{O}(\alpha n)$ time.

Our article concludes in big measure this line of research: we give concrete bounds of the number of $\alpha$-gapped repeats and $\alpha$-gapped palindromes, and, building on the approach from [10], we give optimal algorithms for finding them in the usual case of an integer alphabet whose size is polynomial in the input string length. Namely, we obtain the following results:

- The number of all maximal $\alpha$-gapped repeats in a word of length $n$ is at most $18\alpha n$ (Theorem 11).
- The number of all maximal $\alpha$-gapped palindromes in a word of length $n$ is at most $28\alpha n + 7n$ (Theorem 14).
- We can compute the set of all $\alpha$-gapped repeats in $\mathcal{O}(\alpha n)$ time for *integer* alphabets (Theorem 28).
- This algorithm can be adapted to find the number of all maximal $\alpha$-gapped palindromes in $\mathcal{O}(\alpha n)$ time (Corollary 29).

The following example shows that our obtained bounds on the number of all maximal $\alpha$-gapped repeats and palindromes are asymptotically tight:

*Example 1* ([7, Thm. 2]) The word $w_k := (\mathtt{abba})^k$ with $k \in \mathbb{N}$ contains $\Theta(\alpha n)$ maximal $\alpha$-gapped repeats whose arms are of length one. Since an $\alpha$-gapped repeat whose arms have length one is an $\alpha$-gapped palindrome, we get that the number of maximal $\alpha$-gapped repeats and the number of maximal $\alpha$-gapped palindromes in the word $w_k$ is $\Omega(\alpha n)$.

In this sense, we cannot hope for algorithms finding all $\alpha$-gapped repeats or palindromes faster in the worst case. The results above improve those of [19] (as well as those existing in the literature before [19]). Our algorithms require a deeper analysis than the one developed in [10] for finding the longest $\alpha$-gapped repeats. Asides, they use essentially different techniques and data structures than the ones described in [7, 18, 22].

A related problem is the computation of all factors with an exponent less than 2 that are maximal with respect to their exponents. This problem was recently investigated in [1].

## 2 Combinatorics on Words

Let $\Sigma$ be a finite alphabet; an element of $\Sigma$ is called ***character***. $\Sigma^*$ denotes the set of all finite ***words*** over $\Sigma$. The ***length*** of a word $w \in \Sigma^*$ is denoted by $|w|$. For $v = xuy$ with $x, u, y \in \Sigma^*$, we call $x$, $u$ and $y$ a ***prefix***, ***factor***, and ***suffix*** of $v$, respectively. We denote by $w[i]$ the character occurring at position $i$ in $w$, and by $w[i, j]$ the factor of $w$ starting at position $i$ and ending at position $j$, consisting of the catenation of the characters $w[i], \ldots, w[j]$, where $1 \le i \le j \le n$; $w[i, j]$ is the empty word if $i > j$. By $w^{\intercal}$ we denote the ***mirror image*** of $w$.

By $\mathcal{I} = [b, e]$ we represent the set of consecutive integers from $b$ to $e$, for $b \le e$, and call $\mathcal{I}$ an ***interval***. For an interval $\mathcal{I}$, we use the notations $\mathsf{b}(\mathcal{I})$ and $\mathsf{e}(\mathcal{I})$ to denote the beginning and end of $\mathcal{I}$; i.e., $\mathcal{I} = [\mathsf{b}(\mathcal{I}), \mathsf{e}(\mathcal{I})]$. We write $|\mathcal{I}|$ to denote the length of $\mathcal{I}$; i.e., $|\mathcal{I}| = \mathsf{e}(\mathcal{I}) - \mathsf{b}(\mathcal{I}) + 1$.

A ***subword*** $w[b, e]$ of a word $w$ is the occurrence of a factor $f$ equal to $w[b, e]$ in $w$; we say that $f$ ***occurs*** at position $b$ in $w$. While a factor is identified only by a sequence of characters, a subword is also identified by its position in the word. So subwords are always unique, while a word may contain multiple occurrences of the same factor. We use the same notation for defining factors and subwords of a word. For two subwords $u$ and $\overline{u}$ of a word $w$, we write $u = \overline{u}$ if they start at the same position in $w$ and have the same length. We write $u \equiv \overline{u}$ if the factors identifying these subwords are the same (hence $u = \overline{u} \Rightarrow u \equiv \overline{u}$). We implicitly use subwords both like factors of $w$ and as intervals contained in $[1, |w|]$, e.g., we write $u \subseteq \overline{u}$ if two subwords $u := w[b, e]$, $\overline{u} := w[\overline{b}, \overline{e}]$ of $w$ satisfy $[b, e] \subseteq [\overline{b}, \overline{e}]$, i.e., $\mathsf{b}(\overline{u}) \le \mathsf{b}(u) \le \mathsf{e}(u) \le \mathsf{e}(\overline{u})$. Two subwords $u$ and $\overline{u}$ of the same word $w$ are called ***consecutive***, iff $\mathsf{e}(u) + 1 = \mathsf{b}(\overline{u})$. Two occurrences $u$ and $\overline{u}$ with $\mathsf{b}(u) < \mathsf{b}(\overline{u})$ of the same factor $v$ in a word $w$ are called ***subsequent*** if there is no occurrence of $v$ starting between $\mathsf{b}(u) + 1$ and $\mathsf{b}(\overline{u}) - 1$.

A ***period*** of a word $w$ over $\Sigma$ is a positive integer $p < |w|$ such that $w[i] = w[j]$ for all $i$ and $j$ with $1 \le i, j \le |w|$ and $i \equiv j \pmod{p}$; a word that has period $p$ is also called ***p-periodic***. A word $w$ whose smallest period is at most $\lfloor |w| / 2 \rfloor$ is called ***periodic***; otherwise, $w$ is called ***aperiodic***.

For a word $w$, we call a triple of consecutive subwords $(u_\lambda, v, u_\rho)$ a ***gapped repeat*** with ***period*** $|u_\lambda v|$ and ***gap*** $|v|$ iff $u_\rho \equiv u_\lambda$. A triple of consecutive subwords $(u_\lambda, v, u_\rho)$ is called a ***gapped palindrome*** with gap $|v|$ iff $u_\rho \equiv u_\lambda{}^{\intercal}$. The subwords $u_\lambda$ and $u_\rho$ are called left and right ***arm***, respectively. For $\alpha \ge 1$, the gapped repeat (palindrome) $u_\lambda, v, u_\rho$ is called ***α-gapped*** iff $|u_\lambda| + |v| \le \alpha\, |u_\lambda|$ (see Fig. 1). Further,

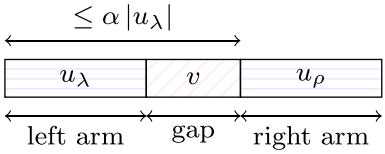


**Fig. 1** A factor of the form $u_\lambda v u_\rho$. It is called an $\alpha$-gapped repeat (palindrome) if $|u_\lambda v| \le \alpha\, |u_\lambda|$ and $u_\lambda \equiv u_\rho$ ($u_\rho \equiv u_\lambda{}^{\intercal}$)

it is called ***maximal*** iff its arms cannot be extended simultaneously to the right nor to the left. This means for a gapped repeat $u_\lambda, v, u_\rho$ that $w[\mathsf{b}(u_\lambda) - 1] \neq w[\mathsf{b}(u_\rho) - 1]$ and $w[\mathsf{e}(u_\lambda) + 1] \neq w[\mathsf{e}(u_\rho) + 1]$, and for a gapped palindrome $u_\lambda, v, u_\rho$ that $w[\mathsf{b}(u_\lambda) - 1] \neq w[\mathsf{e}(u_\rho) + 1]$ and $w[\mathsf{e}(u_\lambda) + 1] \neq w[\mathsf{b}(u_\rho) - 1]$. Let $\mathcal{G}_\alpha(w)$ (resp. $\mathcal{G}_\alpha^\mathsf{T}(w)$) denote the set of all maximal $\alpha$-gapped repeats (resp. palindromes) in $w$. The representation of a maximal gapped repeat (resp. palindrome) by the subword $z := w[u_\lambda]w[v]w[u_\rho]$ is not unique — the same subword $z$ can be composed of gapped repeats (resp. palindromes) with different periods (resp. different gaps). Instead, a maximal gapped repeat (resp. palindrome) is uniquely determined by its left arm $u_\lambda$ and its period (resp. gap). By fixing $w$, we can map $u_\lambda, v, u_\rho$ injectively to the pair of integers $(\mathsf{e}(u_\lambda), |u_\lambda v|)$ in case of gapped repeats, or to $(\mathsf{e}(u_\lambda), |v|)$ in case of gapped palindromes.

A ***repetition*** in a word $w$ is a periodic factor; a ***run*** is a maximal repetition; the ***exponent*** of a run is the (rational) number of times the period fits in that run. Let $E(w)$ denote the sum of the exponents of runs in the word $w$. The exponent of a run $r$ is denoted by $\exp(r)$. We use the following results from the literature:

**Lemma 1** *The length of the overlap between two subsequent occurrences of an aperiodic factor $u$ in a word $w$ is upper bounded by $\lfloor |u| /2 \rfloor$.*

**Lemma 2** ([2]) *For a word $w$, $E(w) < 3\,|w|$, and the number of runs is less than $|w|$.*

**Corollary 3** ([7, Conclusions]) *The number of maximal 1-gapped repeats is less than $|w|$.*

**Lemma 4** ([19]) *Two distinct runs with the same minimal period $p$ cannot have an overlap of length greater than or equal to $p$.*

**Observation 5** *The mirror image of a gapped repeat (resp. palindrome) is a gapped repeat (resp. palindrome) with the same period. Hence, there exist the bijections $\mathcal{G}_\alpha(w) \sim \mathcal{G}_\alpha(w^\mathsf{T})$ and $\mathcal{G}_\alpha^\mathsf{T}(w) \sim \mathcal{G}_\alpha^\mathsf{T}(w^\mathsf{T})$.*

### 2.1 Point Analysis

A pair of positive integers is called a ***point***. We use points to bound the cardinality of a subset of gapped repeats and gapped palindromes by injectively mapping a gapped repeat (resp. palindrome) to a point as stated above. To this end, we show that a certain vicinity of any point generated by a member of this subset does not contain any point that is generated by another member. This vicinity is given by

**Definition 6** Given a real number $\gamma$ with $\gamma \in (0, 1]$, we say that a point $(x, y)$ ***$\gamma$-covers*** a point $(x', y')$ iff $x - \gamma y \leq x' \leq x$ and $y - \gamma y \leq y' \leq y$.

It is crucial that the $\gamma$ factor is always multiplied with the $y$-coordinates. In other words, the number of $\gamma$-covers of a point $(\cdot, y)$ correlates with $\gamma$ and the value $y$. The main property of this definition is given by

**Lemma 7** *Given a real number $\gamma$ with $\gamma \in (0, 1]$, let $S \subset [1, n]^2 \subset \mathbb{N}^2$ be a set of points such that no two distinct points in $S$ $\gamma$-cover the same point. Then $|S| < 3n/\gamma$.*

*Proof* We estimate the maximal number of points that can be placed in $[1, n]^2 \subset \mathbb{N}^2$ such that their covered points are disjoint. First, the number of points $(\cdot, y) \in [1, n]^2$ with $y < 1/\gamma$ is less than $n/\gamma$. Second, if a point $(\cdot, y)$ satisfies $2^\ell/\gamma \leq y < 2^{\ell+1}/\gamma$ for an integer $\ell \geq 0$, the point $(\cdot, y)$ $\gamma$-covers at least $2^\ell \times 2^\ell$ points, or to put it differently, this point $\gamma$-covers at least $2^\ell$ points $(\cdot, y')$ with $y - 2^\ell \leq y' \leq y$. In other words, there are at most $n/(2^\ell\gamma)$ points in $S$ with $2^\ell/\gamma \leq y < 2^{\ell+1}/\gamma$. Hence, $|S| < n/\gamma + \sum_{\ell=0}^{\infty} n/(2^\ell\gamma) = 3n/\gamma$. □

Kolpakov et al. [19] split the set of all maximal $\alpha$-gapped repeats into three subsets, and studied the maximal size of each subset:

- those whose arms are contained in one or two runs,
- those whose arms contain a periodic prefix or suffix larger than half of the size of the arms, and
- those belonging to neither of the two subsets.

They showed that the first two subsets contain at most $\mathcal{O}(\alpha n)$ elements. The point analysis is used as a tool for studying the last subset. By mapping a gapped repeat to a point consisting of the end position of its left arm and its period, they showed that the points created by two different maximal $\alpha$-gapped repeats cannot $\frac{1}{4\alpha}$-cover the same point. By this property, they bounded the size of the last subset by $\mathcal{O}(\alpha^2 n)$. Lemma 7 immediately improves this bound of $\mathcal{O}(\alpha^2 n)$ to $\mathcal{O}(\alpha n)$. Consequently, it shows that the number of maximal $\alpha$-gapped repeats of a word of length $n$ is $\mathcal{O}(\alpha n)$.

### 2.2 Upper Bound on the Number of Periodic Maximal $\alpha$-gapped Repeats and Palindromes

Unlike [7, 18, 19], we partition the maximal $\alpha$-gapped repeats (resp. palindromes) differently. We categorize a gapped repeat (resp. palindrome) depending on whether their left arm contains a periodic prefix or not. The two subsets are treated differently. For the ones having a periodic prefix, we think about the number of runs covering this prefix. The other category is analyzed by using the results of Section 2.1. We begin with a formal definition of both subsets and analyze the former subset.

Let $\beta$ be a real number with $0 < \beta < 1$. A gapped repeat (resp. palindrome) $\sigma = u_\lambda, v, u_\rho$ belongs to $\beta\mathcal{P}_\alpha(w)$ (resp. $\beta\mathcal{P}^{\mathsf{T}}_\alpha(w)$) iff $u_\lambda$ contains a periodic prefix of length at least $\beta\,|u_\lambda|$. We call $\sigma$ ***periodic***. Otherwise $\sigma \in \overline{\beta\mathcal{P}}_\alpha(w)$ (resp. $\sigma \in \overline{\beta\mathcal{P}}^{\mathsf{T}}_\alpha(w)$), where $\overline{\beta\mathcal{P}}_\alpha(w) := \mathcal{G}_\alpha(w) \setminus \beta\mathcal{P}_\alpha(w)$ and $\overline{\beta\mathcal{P}}^{\mathsf{T}}_\alpha(w) := \mathcal{G}^{\mathsf{T}}_\alpha(w) \setminus \beta\mathcal{P}^{\mathsf{T}}_\alpha(w)$; we call $\sigma$ ***aperiodic***.

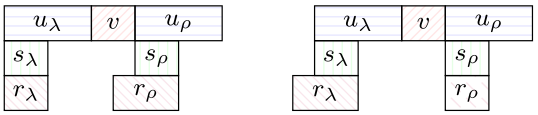


**Fig. 2** Setting of Lemma 8(a). The equation $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$ or $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$ must hold. By the maximality property of runs, $\mathsf{e}(r_\lambda) = \mathsf{e}(s_\lambda)$ and $\mathsf{e}(r_\rho) = \mathsf{e}(s_\rho)$

**Lemma 8** *Let $w$ be a word, $\alpha > 1$ and $0 < \beta < 1$ two real numbers. Then*

(a) $|\beta\mathcal{P}_\alpha(w)|$ *is at most* $2\alpha E(w)/\beta$, *and*
(b) $\left|\beta\mathcal{P}^{\mathsf{T}}_\alpha(w)\right|$ *is at most* $2(\alpha+1)E(w)/\beta$.

*Proof* Let $\sigma = (u_\lambda, v, u_\rho) \in \beta\mathcal{P}_\alpha(w)$ (resp. $\sigma \in \beta\mathcal{P}^{\mathsf{T}}_\alpha(w)$). By definition, the left arm $u_\lambda$ has a periodic prefix $s_\lambda$ of length at least $\beta\,|u_\lambda|$. Let $r_\lambda$ denote the run that generates $s_\lambda$, i.e., $s_\lambda \subseteq r_\lambda$ and they both have the common shortest period $p$. By the definition of gapped repeats (resp. palindromes), there is a right copy $s_\rho$ of $s_\lambda$ contained in $u_\rho$ with

$$s_\rho = \begin{cases} w[\mathsf{b}(s_\lambda) + |u_\lambda v|\,,\, \mathsf{e}(s_\lambda) + |u_\lambda v|] \equiv s_\lambda & \text{if } \sigma \in \beta\mathcal{P}_\alpha(w), \\ w[\mathsf{b}(u_\rho) + (\mathsf{e}(u_\lambda) - \mathsf{e}(s_\lambda)), \mathsf{b}(u_\rho) + (\mathsf{e}(u_\lambda) - \mathsf{e}(s_\lambda)) + |s_\lambda| - 1] \equiv s_\lambda{}^{\mathsf{T}} & \text{if } \sigma \in \beta\mathcal{P}^{\mathsf{T}}_\alpha(w). \end{cases}$$

Let $r_\rho$ be a run generating $s_\rho$ (it is possible that $r_\rho$ and $r_\lambda$ are identical). By definition, $r_\rho$ has the same period $p$ as $r_\lambda$.

Since $\sigma$ is maximal, $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$ or $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$ (resp. $\mathsf{e}(u_\rho) = \mathsf{e}(r_\rho)$) must hold (see Fig. 2, resp. Fig. 3); otherwise we could extend $\sigma$ to the left (resp. outwards).

**(a) Gapped Repeats**

The periodic $\alpha$-gapped repeat $\sigma$ is uniquely determined by its period $q := |u_\lambda v|$, and

– $r_\lambda$ in case $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$, or
– $r_\rho$ in case $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$.

We analyze the case $\mathsf{b}(u_\lambda) = \mathsf{b}(s_\lambda) = \mathsf{b}(r_\lambda)$, the other is treated exactly in the same way by symmetry. The gapped repeat $\sigma$ is identified by $r_\lambda$ and the period $q$. We fix $r_\lambda$ and pose the question how many maximal periodic gapped repeats can be generated by $r_\lambda$. To this end, we count the number of possible values for the period $q$. Given two different gapped repeats $\sigma_1$ and $\sigma_2$ with respective periods $q_1$ and $q_2$ such that the left arms of both are generated by $r_\lambda$, the difference between $q_1$ and $q_2$ must be at least $p$.

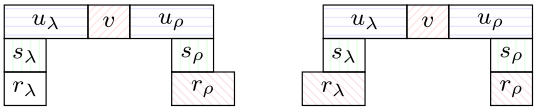


**Fig. 3** Setting of Lemma 8(b). The equation $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$ or $\mathsf{e}(u_\rho) = \mathsf{e}(r_\rho)$ must hold. By the maximality property of runs, $\mathsf{e}(r_\lambda) = \mathsf{e}(s_\lambda)$ and $\mathsf{b}(r_\rho) = \mathsf{b}(s_\rho)$

**Sub-Claim** Given two different gapped repeats $\sigma_1$ and $\sigma_2$ with respective periods $q_1$ and $q_2$ such that the left arms of both are generated by $r_\lambda$, the difference $\delta$ between $q_1$ and $q_2$ must be at least $p$.

**Sub-Proof** We consider two cases:

- If $\mathsf{e}(u_\lambda) = \mathsf{e}(r_\lambda)$, then $u_\lambda = s_\lambda = r_\lambda$, so $u_\lambda$ is $p$-periodic. Since both right arms are $p$-periodic, too, $\delta$ is a multiple of $p$.
- Otherwise, both gapped repeats are generated by two different repeats with period $p$. So by Lemma 4, $\delta$ must be at least $p$.

Since $|u_\lambda| \leq |s_\lambda|/\beta$ and $\sigma$ is $\alpha$-gapped, $1 \leq q \leq |s_\lambda|\alpha/\beta \leq |r_\lambda|\alpha/\beta$. Then the number of possible periods $q$ is bounded by $|r_\lambda|\alpha/(\beta p) = \exp(r_\lambda)\alpha/\beta$. Therefore the number of maximal $\alpha$-gapped repeats is bounded by $\alpha E(w)/\beta$ for the case $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$. Since the case $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$ is symmetric, we get the bound $2\alpha E(w)/\beta$ in total.

**(b) Gapped Palindromes**

The periodic $\alpha$-gapped palindrome $\sigma$ is uniquely determined by its distance $d := \mathsf{b}(s_\rho) - \mathsf{e}(s_\lambda)$, and

- $r_\lambda$ in case $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$, or
- $r_\rho$ in case $\mathsf{e}(u_\rho) = \mathsf{e}(r_\rho)$.

We analyze the case $\mathsf{b}(s_\lambda) = \mathsf{b}(r_\lambda)$, the other is treated exactly in the same way by symmetry. The gapped palindrome $\sigma$ is identified by $r_\lambda$ and $d$. We fix $r_\lambda$ and count the number of possible values for $d$. Given two different periodic $\alpha$-gapped palindromes with the distances $d_1$ and $d_2$, the difference between $d_1$ and $d_2$ must be at least $p$, due to Lemma 4. It follows from $|u_\lambda| \leq |s_\lambda|/\beta$ that $d = |v| + 2(|u_\lambda| - |s_\lambda|) \leq |v| + 2|s_\lambda|/\beta$. Since $\sigma$ is $\alpha$-gapped, $|v| \leq (\alpha - 1)|u_\lambda| \leq (\alpha - 1)|s_\lambda|/\beta$, and hence, $1 \leq d \leq |s_\lambda|(\alpha + 1)/\beta$. Then the number of possible values for the distance $d$ is bounded by $|s_\lambda|(\alpha + 1)/(\beta p) \leq |r_\lambda|(\alpha + 1)/(\beta p) = \exp(r_\lambda)(\alpha + 1)/\beta$. In total, the number of maximal $\alpha$-gapped palindromes in this case is bounded by $(\alpha+1)E(w)/\beta$ for the case $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$. Since the case $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$ is symmetric, we get the bound $2(\alpha + 1)E(w)/\beta$ in total. □

### 2.3 Upper Bound on the Number of Maximal $\alpha$-gapped Repeats

We optimize the proof technique from Kolpakov et al. [19] and improve the upper bound on the number of maximal $\alpha$-gapped repeats in a word of length $n$ from $\mathcal{O}(\alpha n)$ to $18\alpha n$. Remembering the results of Section 2.1, we map gapped repeats to their respective points. By using the period of a gapped repeat as the $y$-coordinate, we can show the following lemma:

**Lemma 9** *Given a word $w$, and two real numbers $\alpha, \beta$ with $\alpha > 1$ and $2/3 \leq \beta < 1$, the points mapped by two different maximal gapped repeats in $\overline{\beta\mathcal{P}}_\alpha(w)$ cannot $\frac{1-\beta}{\alpha}$-cover the same point.*

*Proof* Let $\sigma = u_\lambda, v, u_\rho$ and $\overline{\sigma} = \overline{u_\lambda}, \overline{v}, \overline{u_\rho}$ be two different maximal gapped repeats in $\overline{\beta\mathcal{P}_\alpha(w)}$. Set $u := |u_\lambda| = |u_\rho|$, $\overline{u} := |\overline{u_\lambda}| = |\overline{u_\rho}|$, $q := |u_\lambda v|$ and $\overline{q} := |\overline{u_\lambda}\,\overline{v}|$. We map the maximal gapped repeats $\sigma$ and $\overline{\sigma}$ to the points $(\mathsf{e}(u_\lambda), q)$ and $(\mathsf{e}(\overline{u_\lambda}), \overline{q})$, respectively. Assume, for the sake of contradiction, that both points $\frac{1-\beta}{\alpha}$-cover the same point $(x, y)$.

Let $z := |\mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda})|$ be the difference of the endings of both left arms, and $s_\lambda := w[[\mathsf{b}(u_\lambda), \mathsf{e}(u_\lambda)] \cap [\mathsf{b}(\overline{u_\lambda}), \mathsf{e}(\overline{u_\lambda})]]$ be the overlap of $u_\lambda$ and $\overline{u_\lambda}$. Let $s := |s_\lambda|$, and let $s_\rho$ (resp. $\overline{s_\rho}$) be the right copy of $s_\lambda$ based on $\sigma$ (resp. $\overline{\sigma}$).

**Sub-Claim** The overlap $s_\lambda$ is not empty, and $s_\rho \neq \overline{s_\rho}$

**Sub-Proof** Assume for this sub-proof that $\mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$ (otherwise exchange $\sigma$ with $\overline{\sigma}$, or yield the contradiction $\sigma = \overline{\sigma}$). The latter contradiction ($\sigma = \overline{\sigma}$) is yielded by the following consideration: Since $\mathsf{e}(u_\lambda) = \mathsf{e}(\overline{u_\lambda})$, $s_\lambda$ cannot be empty (it is the intersection of both left arms). Further, both right copies are defined as the right translation of $s_\lambda$ by $q$ and $\overline{q}$, respectively. So if both right copies are identical, then $q = \overline{q}$, which contradicts the fact that the mapping of a maximal gapped repeat to the point consisting of its end point and its period is injective.

Having $\mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$, we can combine the $(1-\beta)/\alpha$-cover property with the fact that $\overline{\sigma}$ is $\alpha$-gapped, and yield $\mathsf{e}(\overline{u_\lambda}) - \overline{u} \leq \mathsf{e}(\overline{u_\lambda}) - \overline{q}/\alpha < \mathsf{e}(\overline{u_\lambda}) - \overline{q}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$. Hence, the subword $w[\mathsf{e}(u_\lambda)]$ is contained in $\overline{u_\lambda}$. If $s_\rho = \overline{s_\rho}$, then we get a contradiction to the maximality of $\sigma$: By the above inequality, $w[\mathsf{e}(u_\lambda) + 1]$ is contained in $\overline{u_\lambda}$, too. Since $\overline{\sigma}$ is a gapped repeat, the character $w[\mathsf{e}(u_\lambda) + 1]$ occurs in $\overline{u_\rho}$, exactly at $w[\mathsf{e}(u_\rho) + 1]$.

This sub-claim shows that $q \neq \overline{q}$. Without loss of generality let $q < \overline{q}$. Then

$$\overline{q} - \frac{\overline{q}(1-\beta)}{\alpha} \leq y \leq q \leq \overline{q}. \tag{1}$$

$$\text{So the difference of both periods is } 0 < \delta := \overline{q} - q \leq \overline{q}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{2}$$

$$\text{Eq. (1) also yields that } u \geq q/\alpha \geq \frac{\overline{q}}{\alpha}(1 - \frac{1-\beta}{\alpha}) \geq \overline{q}\beta/\alpha. \tag{3}$$

Since $s_\rho = [\mathsf{b}(s_\lambda) + q, \mathsf{e}(s_\lambda) + q]$ and $\overline{s_\rho} = [\mathsf{b}(s_\lambda) + \overline{q}, \mathsf{e}(s_\lambda) + \overline{q}]$, we have $\mathsf{b}(\overline{s_\rho}) - \mathsf{b}(s_\rho) = \delta$.

By case analysis, we show that $u_\lambda$ or $\overline{u_\lambda}$ has a periodic prefix, which leads to the contradiction that $\sigma$ or $\overline{\sigma}$ are in $\beta\mathcal{P}_\alpha(w)$.

1. **Case** $\mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(\overline{u_\lambda}) - \overline{q}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$,

$$z = \mathsf{e}(\overline{u_\lambda}) - \mathsf{e}(u_\lambda) \leq \overline{q}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{4}$$

1a. **Sub-Case** $\mathsf{b}(u_\lambda) \leq \mathsf{b}(\overline{u_\lambda})$, see Fig. 4. By (4), we get $s = \overline{u} - z \geq \overline{u}\beta$. It follows from (2) and $2/3 \leq \beta < 1$ that $s/\delta \geq \overline{u}\beta/\overline{u}(1-\beta) = \beta/(1-\beta) \geq 2$, which means that $s_\rho$ and $\overline{s_\rho}$ overlap at least half of their common length, so $s_\lambda$ is periodic. Since $s_\lambda$ is a prefix of $\overline{u_\lambda}$ of length $s \geq \overline{u}\beta$, $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

1b. **Sub-Case** $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$, see Fig. 5. We conclude that $s_\lambda = u_\lambda$. It follows from (2) and (3) and $2/3 \leq \beta < 1$ that $s/\delta \geq \overline{q}\alpha\beta/(\overline{q}\alpha(1-\beta)) = \beta/(1-\beta) \geq 2$, which means that $s_\lambda = u_\lambda$ is periodic. Hence $\sigma$ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

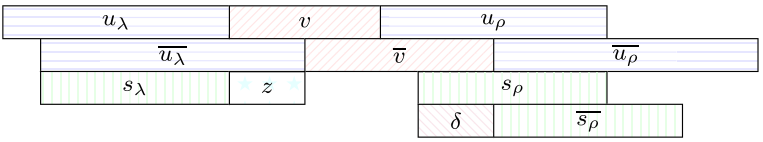


**Fig. 4** Sub-Case 1a

2. **Case** $\mathsf{e}(u_\lambda) > \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(u_\lambda) - q(1-\beta)/\alpha \le x \le \mathsf{e}(\overline{u_\lambda}) \le \mathsf{e}(u_\lambda)$,

$$z = \mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda}) \le q(1-\beta)/\alpha \le \overline{q}(1-\beta)/\alpha \le \overline{u}(1-\beta). \tag{5}$$

2a. **Sub-Case** $\mathsf{b}(u_\lambda) \le \mathsf{b}(\overline{u_\lambda})$, see Fig. 6. We conclude that $s_\lambda = \overline{u_\lambda}$. It follows from (2) and $2/3 \le \beta < 1$ that $s/\delta \ge \overline{u}/(\overline{u}(1-\beta)) = 1/(1-\beta) \ge 3 > 2$, which means that $s_\lambda = \overline{u_\lambda}$ is periodic. Hence $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

2b. **Sub-Case** $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$, see Fig. 7. By (5) we get $z \le q(1-\beta)/\alpha \le u(1-\beta)$ and hence $s = u - z \ge u\beta$. If $\delta \le s/2$, $s_\rho$ and $\overline{s_\rho}$ overlap at least half of their common length, which leads to the contradiction that $u_\lambda$ has a periodic prefix $s_\lambda$ of length at least $u\beta$. Otherwise, let us assume that $s/2 < \delta$. By (2) and (3) we get $u/\delta \ge \overline{q}\alpha\beta/(\overline{q}\alpha(1-\beta)) = \beta/(1-\beta) \ge 2$ with $2/3 \le \beta < 1$. Hence, $\delta$ is upper bounded by $u/2$; so $u_\rho$ has a periodic prefix of length at least $2\delta$ (since $2\delta > s \ge u\beta$), a contradiction. □

The next lemma follows immediately from Lemmas 7 and 9.

**Lemma 10** *Given two real numbers $\alpha, \beta$ with $\alpha > 1$ and $2/3 \le \beta < 1$, the number of all aperiodic maximal $\alpha$-gapped repeats $\left|\overline{\beta\mathcal{P}}_\alpha(w)\right|$ of a word $w$ of length $n$ is less than $3\alpha n/(1-\beta)$.*

**Theorem 11** *Given a word $w$ of length $n$ and a real number $\alpha > 1$, the number of all $\alpha$-gapped repeats $|\mathcal{G}_\alpha(w)|$ is less than $18\alpha n$.*

*Proof* Combining the results of Lemma 8(a) and Lemma 10, $|\mathcal{G}_\alpha(w)| = |\beta\mathcal{P}_\alpha(w)| + \left|\overline{\beta\mathcal{P}}_\alpha(w)\right| < 2\alpha E(w)/\beta + 3\alpha n/(1-\beta)$ for $2/3 \le \beta < 1$. Applying Lemma 2, the term is upper bounded by $6\alpha n/\beta + 3\alpha n/(1-\beta)$. The number is minimal for $\beta = 2/3$, yielding the bound $18\alpha n$. □

With Corollary 3 we obtain the result of Theorem 11 for $\alpha \ge 1$.

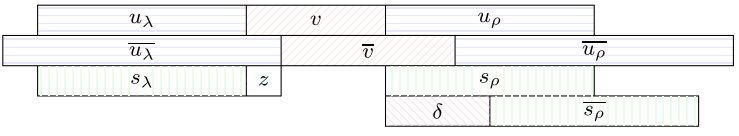


**Fig. 5** Sub-Case 1b

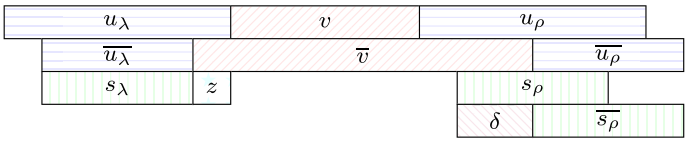


**Fig. 6** Sub-Case 2a

### 2.4 Upper Bound on the Number of Maximal $\alpha$-gapped Palindromes

We can bound the maximum number of maximal $\alpha$-gapped palindromes by similar proofs to $28\alpha n + 7n$. This bound solves an open problem in [18], where Kolpakov and Kucherov conjectured that the number of $\alpha$-gapped palindromes with $\alpha \geq 2$ in a word is linear. We briefly explain the main differences and similarities needed to understand the relationship between gapped repeats and palindromes. Let $\sigma$ be a maximal $\alpha$-gapped repeat (or $\alpha$-gapped palindrome). If $\sigma$ has a periodic prefix $s_\lambda$ generated by a run, then its right arm has a periodic prefix (suffix) $s_\rho$ generated by a run of the same period. Since $\sigma$ is maximal, both runs have to obey constraints that are similar in both cases, considering whether $\sigma$ is a gapped repeat or a gapped palindrome (this is reflected by the fact that large parts for proving the statements of Lemma 8(a) and Lemma 8(b) are identical). Like with aperiodic gapped repeats, we can apply the point analysis to the aperiodic $\alpha$-gapped palindromes, too. Our main idea is to map a gapped palindrome $u_\lambda, v, u_\rho$ injectively to the pair of integers $(\mathsf{e}(u_\lambda), |v|)$, exchanging the period with the size of the gap.

In what follows, we focus on maximal $\alpha$-gapped palindromes with $\alpha > 1$. That is because the case $\alpha = 1$ is already solved in literature. To see this, we observe that 1-gapped palindromes are (plain) palindromes. A ***palindrome*** $u$ is a subword with $u^{\mathsf{T}} = u$. Its ***center*** is the value $(\mathsf{e}(u) + \mathsf{b}(u))/2$. A palindrome is called ***maximal*** if there is no longer palindrome with the same center. So a maximal palindrome is uniquely defined by its center. Hence, the number of maximal palindromes in a word of length $n$ is at most $2n - 1$. We conclude our observation with the fact that maximal 1-gapped palindromes are maximal palindromes. For the algorithmic part, the algorithm of [20] can be used to find all the maximal palindromes in linear time.

**Lemma 12** *Given a word $w$, and two real numbers $\alpha > 1$ and $6/7 \leq \beta < 1$. The points mapped by two different maximal gapped palindromes in $\overline{\beta\mathcal{P}}^{\mathsf{T}}_\alpha(w)$ cannot $\frac{1-\beta}{\alpha}$-cover the same point.*

*Proof* Let $\sigma = u_\lambda, v, u_\rho$ and $\overline{\sigma} = \overline{u_\lambda}, \overline{v}, \overline{u_\rho}$ be two different gapped palindromes in $\overline{\beta\mathcal{P}}^{\mathsf{T}}_\alpha(w)$. Set $u := |u_\lambda| = |u_\rho|$, $\overline{u} := |\overline{u_\lambda}| = |\overline{u_\rho}|$, $g := |v|$ and $\overline{g} := |\overline{v}|$. We map

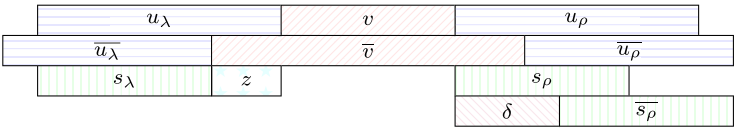


**Fig. 7** Sub-Case 2b

the maximal gapped palindromes $\sigma$ and $\overline{\sigma}$ to the points $(\mathsf{e}(u_\lambda), g)$ and $(\mathsf{e}(\overline{u_\lambda}), \overline{g})$, respectively. Assume, for the sake of contradiction, that both points $\frac{1-\beta}{\alpha}$-cover the same point $(x, y)$. This means for the point $(\mathsf{e}(u_\lambda), g)$ that $\mathsf{e}(u_\lambda) - (1-\beta)g/\alpha \leq x \leq \mathsf{e}(u_\lambda)$ and $g - (1-\beta)g/\alpha \leq y \leq g$ hold. The same inequations hold when exchanging $(\mathsf{e}(u_\lambda), g)$ with $(\mathsf{e}(\overline{u_\lambda}), \overline{g})$.

Let $z := |\mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda})|$ be the difference of the endings of both left arms, and $s_\lambda := w[[\mathsf{b}(u_\lambda), \mathsf{e}(u_\lambda)] \cap [\mathsf{b}(\overline{u_\lambda}), \mathsf{e}(\overline{u_\lambda})]]$ be the overlap of $u_\lambda$ and $\overline{u_\lambda}$. Let $s = |s_\lambda|$, and let $s_\rho$ (resp. $\overline{s_\rho}$) be the reversed copy of $s_\lambda$ based on $\sigma$ (resp. $\overline{\sigma}$).

**Sub-Claim** The overlap $s_\lambda$ is not empty, and $s_\rho \neq \overline{s_\rho}$

**Sub-Proof** Assume for this sub-proof that $\mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$ (otherwise exchange $\sigma$ with $\overline{\sigma}$, or yield the contradiction $\sigma = \overline{\sigma}$). The latter contradiction ($\sigma = \overline{\sigma}$) is yielded by the following consideration: Since $\mathsf{e}(u_\lambda) = \mathsf{e}(\overline{u_\lambda})$, $s_\lambda$ cannot be empty (it is the intersection of both left arms). In particular, it is the longest common suffix of $u_\lambda$ and $\overline{u_\lambda}$. Consequently, both reversed copies $s_\rho$ and $\overline{s_\rho}$ of $s_\lambda$ are prefixes of $u_\rho$ and $\overline{u_\rho}$, respectively. The gap between $s_\rho$ and $\overline{s_\rho}$ to $s_\lambda$ is $g$ and $\overline{g}$, respectively. In other words, if $s_\rho = \overline{s_\rho}$, then $g = \overline{g}$, a contradition to the fact that the mapping of a maximal gapped palindrome to the point consisting of its end point and its gap is injective.

By combining the $(1-\beta)/\alpha$-cover property with the fact that $\overline{\sigma}$ is $\alpha$-gapped, we yield $\mathsf{e}(\overline{u_\lambda}) - \overline{u} \leq \mathsf{e}(\overline{u_\lambda}) - (\overline{u} + \overline{g})/\alpha < \mathsf{e}(\overline{u_\lambda}) - \overline{g}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$. So the subword $w[\mathsf{e}(u_\lambda)]$ is contained in $\overline{u_\lambda}$. If $s_\rho = \overline{s_\rho}$, then we get a contradiction to the maximality of $\sigma$: By the above inequality, $w[\mathsf{e}(u_\lambda)+1]$ is contained in $\overline{u_\lambda}$, too. Since $\overline{\sigma}$ is a gapped palindrome, the character $w[\mathsf{e}(u_\lambda)+1]$ occurs in $\overline{u_\rho}$, exactly at $w[\mathsf{b}(u_\rho)-1]$.

Without loss of generality let $g \leq \overline{g}$. Then

$$\overline{g} - \frac{\overline{g}(1-\beta)}{\alpha} \leq y \leq g \leq \overline{g}. \tag{6}$$

So the difference of both gaps is

$$0 \leq \delta := \overline{g} - g \leq \overline{g}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{7}$$

By case analysis, we show that $s_\lambda$ is periodic, which leads to the contradiction that $\sigma$ or $\overline{\sigma}$ are in $\beta\mathcal{P}^{\mathsf{T}}_\alpha(w)$.

1. **Case** $\mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(\overline{u_\lambda}) - \overline{g}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$,

$$z = \mathsf{e}(\overline{u_\lambda}) - \mathsf{e}(u_\lambda) \leq \overline{g}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{8}$$

Since $s_\lambda$ is a suffix of $u_\lambda$, the reverse copy $s_\rho$ is a prefix of $u_\rho$. The starting positions of both right copies $\overline{s_\rho}$ and $s_\rho$ differ by $\mathsf{b}(\overline{s_\rho}) - \mathsf{b}(s_\rho) = 2z + \delta > 0$. The inequality $2z + \delta > 0$ holds, since $\mathsf{e}(u_\lambda) \neq \mathsf{e}(\overline{u_\lambda})$ or $g \neq \overline{g}$. By (7) and (8), we get

$$2z + \delta \leq 3\overline{g}(1-\beta)/\alpha \leq 3\overline{u}(1-\beta). \tag{9}$$

1a. **Sub-Case** $\mathsf{b}(u_\lambda) \leq \mathsf{b}(\overline{u_\lambda})$, see Fig. 8. By (8), we get $s = \overline{u} - z \geq \overline{u}\beta$. It follows from $6/7 \leq \beta < 1$ and (9) that $s/(2z+\delta) \geq \overline{u}\beta/3\overline{u}(1-\beta) = \beta/(3(1-\beta)) \geq 2$, which means that $s_\rho$ and $\overline{s_\rho}$ overlap by at least half of their common length, and

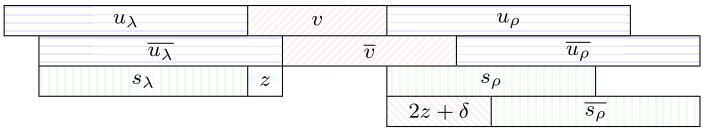


**Fig. 8** Sub-Case 1a

$s_\lambda$ is periodic. Since $s_\lambda$ is a prefix of $\overline{u_\lambda}$ of length $s \geq \overline{u}\beta$, $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)$, a contradiction.

1b. **Sub-Case** $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$, see Fig. 9. We conclude that $s_\lambda = u_\lambda$. By (6) and $\beta < 1$,

$$u \geq g/\alpha \geq \frac{\overline{g}}{\alpha}\left(1 - \frac{1-\beta}{\alpha}\right) \geq \overline{g}\beta/\alpha. \tag{10}$$

It follows from $6/7 \leq \beta < 1$ and (9) that $s/(2z+\delta) \geq \overline{g}\alpha\beta/(3\alpha\overline{g}(1-\beta)) = \beta/(3(1-\beta)) \geq 2$, which means that $s_\lambda = u_\lambda$ is periodic. Hence $\sigma$ is in $\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)$, a contradiction.

2. **Case** $\mathsf{e}(u_\lambda) > \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(u_\lambda) - g(1-\beta)/\alpha \leq x \leq \mathsf{e}(\overline{u_\lambda}) \leq \mathsf{e}(u_\lambda)$,

$$z = \mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda}) \leq g(1-\beta)/\alpha \leq \overline{g}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{11}$$

The starting positions of both right copies differ by $\left|\mathsf{b}(s_\rho) - \mathsf{b}(\overline{s_\rho})\right| = |2z-\delta|$ because $\mathsf{b}(s_\rho) = \mathsf{e}(s_\lambda) + 2z + g$ and $\mathsf{b}(\overline{s_\rho}) = \mathsf{e}(s_\lambda) + \overline{g}$. Since $2z - \delta \leq \max(\delta, 2z)$, we get $|2z-\delta| \leq 2\overline{g}(1-\beta)/\alpha \leq 2\overline{u}(1-\beta)$ by (7) and (11).

2a. **Sub-Case** $\mathsf{b}(u_\lambda) \leq \mathsf{b}(\overline{u_\lambda})$, see Fig. 10. We conclude that $s_\lambda = \overline{u_\lambda}$. It follows from $6/7 \leq \beta < 1$ that $s/|2z-\delta| \geq \overline{u}/(2\overline{u}(1-\beta)) = 1/(2(1-\beta)) \geq 7/2 > 2$, which means that $s_\lambda = \overline{u_\lambda}$ is periodic. Hence $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)$, a contradiction.

2b. **Sub-Case** $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$, see Fig. 11. By $z \leq g(1-\beta)/\alpha$ of (11), we get $z \leq u(1-\beta)$ and thus $s = u - z \geq \beta u$. It follows from (10) and (7), and $2(\sqrt{2}-1) < 6/7 \leq \beta < 1$ that $s/|2z-\delta| \geq \beta u/(2\overline{g}(1-\beta)/\alpha) \geq \overline{g}\beta^2/(2\overline{g}(1-\beta)) = \beta^2/(2(1-\beta)) > 2$, which means that $s_\lambda$ is periodic. Since $s_\lambda$ is a prefix of $u_\lambda$ of length $s \geq u\beta$, $\sigma$ is in $\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)$, a contradiction. □

The next lemma follows immediately from Lemmas 7 and 12.

**Lemma 13** *Given two real numbers $\alpha, \beta$ with $\alpha > 1$ and $6/7 \leq \beta < 1$, the number of all aperiodic $\alpha$-gapped palindromes $\left|\overline{\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)}\right|$ of a word $w$ of length $n$ is less than $3\alpha n/(1-\beta)$.*

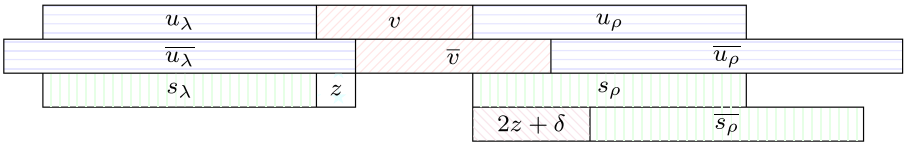


**Fig. 9** Sub-Case 1b


**Fig. 10** Sub-Case 2a. The left and the right gapped repeat show the cases $2z - \delta > 0$ and $2z - \delta < 0$, respectively

**Theorem 14** *Given a word $w$ of length $n$ and a real number $\alpha > 1$, the number of all maximal $\alpha$-gapped palindromes $\left|\mathcal{G}_\alpha^{\mathsf{T}}(w)\right|$ is less than $28\alpha n + 7n$.*

*Proof* By Lemmas 8 and 13, $\left|\mathcal{G}_\alpha^{\mathsf{T}}(w)\right| = \left|\beta\mathcal{P}_\alpha^{\mathsf{T}}(w)\right| + \left|\overline{\beta\mathcal{P}}_\alpha^{\mathsf{T}}(w)\right| < 2(\alpha + 1)E(w)/\beta + 3\alpha n/(1-\beta)$ for every $6/7 \leq \beta < 1$. Applying Lemma 2, the term is upper bounded by $6(\alpha+1)n/\beta + 3\alpha n/(1-\beta)$. This number is minimal when $\beta = 6/7$, yielding the bound $28\alpha n + 7n$. □

## 3 Finding All Maximal $\alpha$-gapped Repeats

For the upcoming algorithmic problems, we fix a word $w$ of length $n$ on an alphabet of size $n^{\mathcal{O}(1)}$. Our computational model is the word RAM model with word size $\Omega(\lg n)$ (where the function lg is the logarithm to base two). Consequently, each character of $w$ fits into a constant number of memory words.

Our algorithm is split into three parts. A part finds all $\alpha$-gapped repeats with

(1) an arm of one character,
(2) an arm of length between two and $\gamma \lg n$ characters, and
(3) arms longer than $\gamma \lg n$ characters.

As a starter, we can find (1) very easily in our target time of $\mathcal{O}(\alpha n)$:

**Lemma 15** *We can compute all maximal $\alpha$-gapped repeats with an arm of one character in a word $w$ of length $n$ in $\mathcal{O}(\alpha n)$ time.*

*Proof* For each position $i$ with $1 \leq i \leq n$, we check whether the characters $w[i]$ and $w[i+j]$ form the arms of an maximal $\alpha$-gapped repeat, for $1 \leq j \leq \alpha$. They form an $\alpha$-gapped repeat if $w[i] = w[i+j]$. If $w[i] = w[i+j]$ we try to prolong the arms $w[i]$ and $w[i+j]$ to check whether the found $\alpha$-gapped repeat is maximal. □

The main ingredient to our algorithms dealing with (2) and (3) is a data structure for finding maximal equal subwords of a word that start or end at some particular positions.


**Fig. 11** Sub-Case 2b. The left and the right gapped repeat show the cases $2z - \delta > 0$ and $2z - \delta < 0$, respectively

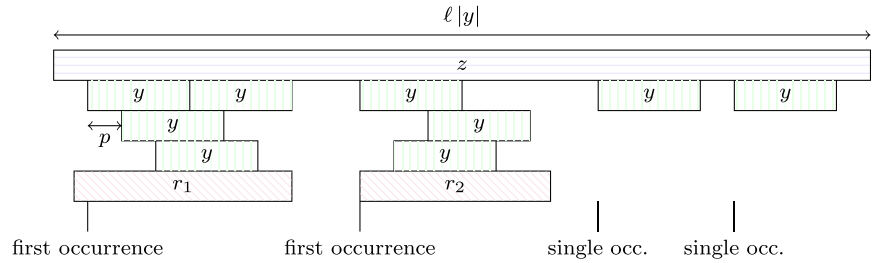


**Fig. 12** Occurrences of $y$ in $z$. The two rightmost subwords are single occurrences. The other subwords are occurrences within a run. We store the starting position of the first occurrences of the subwords in a run, and the starting positions of the subwords that are not part of a run

**Lemma 16** *Given a word $w$ of length $n$, there is a data structure that*

(a) *it can be built in $\mathcal{O}(n)$ time,*
(b) *it can compute the longest common prefix of two suffixes of $w$ in constant time, and*
(c) *it can compute the longest common suffix of two prefixes of $w$ in constant time.*

*Proof* We build the suffix array of $w$ and the longest common prefix (LCP) array of $w$ in $\mathcal{O}(n)$ time [15]; Subsequently we invert the suffix array in $\mathcal{O}(n)$ time. Having a range minimum data structure [9] on the LCP array, we can solve (b) since the longest common prefix of two suffixes $w[s..]$ and $w[t..]$ can be answered with a range minimum query on the LCP array with the range $[\mathsf{SA}^{-1}[s]+1..\mathsf{SA}^{-1}[t]]$ in constant time, where $\mathsf{SA}^{-1}$ denotes the inverse suffix array. By building the same data structure on the mirror image of $w$, we solve (c). □

We call the data structure of Lemma 16 an LCE$^{\leftrightarrow}$ data structure.[1] Subsequently, we provide some tools that show the usefulness of an LCE$^{\leftrightarrow}$ data structure for our problem. We start with a lemma that uses an LCE$^{\leftrightarrow}$ data structure:

**Lemma 17** *Given a word $w$ of length $n$, we can preprocess it in $\mathcal{O}(n)$ time such that we can return the longest factor with the period $p$ starting at position $i$ in $w$ (for $1 \le i, p \le n$ arbitrary), in constant time.*

*Proof* Once we have produced the LCE$^{\leftrightarrow}$ data structure of $w$, we just have to compute the longest common prefix of $w[i, n]$ and $w[i+p, n]$. If this prefix is $w[i+p, \ell]$, then $w[i, \ell]$ is the longest $p$-periodic factor starting at position $i$. □

Let $y$ be a factor of $w$ with period $p$. Further, let $z$ be a subword of length $\ell\,|y|$ of $w$. For an easier presentation of our algorithm, we distinguish between two types of occurrences of $y$ in $z$ (see Fig. 12).

[1] Our abbreviation for a data structure answering longest common extension queries in both directions.

- On the one hand, we have the so-called ***single occurrences***.
  - If $y$ is aperiodic, then all its occurrences in $z$ are single occurrences; there are $\mathcal{O}(\ell)$ such occurrences [16].
  - If $y$ is periodic, then the subword $z[i, i + |y| - 1]$ is a single occurrence if $y$ occurs neither at position $i - p$ nor at position $i + p$ in $z$.
- On the other hand, we call an occurrence of $y$ ***within a run***, if the occurrence is contained in a run whose period is equal to the smallest period of $y$. Let $z[i, i + |y| - 1]$ be an occurrence of $y$. If there is an occurrence of $y$ that shares with $z[i, i + |y| - 1]$ at least $|y|/2$ positions, then both occurrences are occurrences of $y$ within a run. Conversely, given a run $r$ with period $p$ and an occurrence $z[i, i + |y| - 1]$ of $y$, then $z[i, i + |y| - 1]$ is an occurrence within the run $r$ if $y$ occurs either at $i - p$ or at $i + p$. Additionally, we say that $z[i, i + |y| - 1]$ is the ***first occurrence*** of $y$ within a run of period $p$ if $y$ does not occur at $i - p$ but occurs at $i + p$. By Lemma 4, there are at most $\mathcal{O}(\ell)$ runs containing occurrences of $y$ in $z$, i.e., $\mathcal{O}(\ell)$ first occurrences of $y$ in a run in $z$.

**Corollary 18** *Given a subword $y$ of $w$ and a subword $z$ of $w$ with length $\ell\,|y|$, the occurrences of $y$ in $z$ can be represented succinctly in $\mathcal{O}(\ell)$ words.*

*Proof* We only store the starting position of the single and first occurrences, and the period of $y$. This is sufficient, since we can reconstruct the missing information in constant time due to the LCE$^{\leftrightarrow}$ data structure.

Having the starting position of the first occurrence of $y$ in a run $r$ we can compute all further occurrences within the run $r$ by an arithmetic progression with the difference equal to the period of $y$. The number of occurrences within this run can be determined in constant time due to Lemma 17. □

In our approach, we restrict $y$ to be a factor of the length $2^k$ for an integer $k \geq 1$; a factor is called a ***basic factor*** if its length is equal to a power of two. We can find the occurrences of a basic factor $y$ in a subword $z$ of length $\ell\,|y|$ efficiently due to the following lemma:

**Lemma 19** ([4], as well as [10, 16] and the references therein) *For each basic factor $y$ of $w$, we compute an array containing the starting positions of the occurrences of $y$ in ascending order. Computing the arrays can be done in $\mathcal{O}(n \lg n)$ time.*

In order to search in the arrays of Lemma 19 efficiently, we are interested in a data structure built upon a sorted integer array $A$ that can, given an integer $j$,

- retrieve the largest index $i$ with $A[i] \leq j$ (***predecessor query***),
- retrieve the smallest index $i$ with $A[i] \geq j$ (***successor query***), and
- conduct both above operations in $\mathcal{O}(\lg\lg |A|)$ time.

Such a data structure is given in

**Lemma 20** ([21, Observation 2.1]) *Given a sorted array of length $n$ storing integers (represented by $\lg n$ bits), we can build a data structure in $\mathcal{O}(n)$ time that answers predecessor and successor queries in $\mathcal{O}(\lg\lg n)$ time.*

We can use Lemmas 19 and 20 in the following way (see also Fig. 13):

**Corollary 21** *Given a word $w$ of length $n$ and an integer $\ell \geq 2$, we can preprocess $w$ in $\mathcal{O}(n \lg n)$ time such that given a basic factor $y := w[i, i+2^k-1]$ and a subword $z := w[j, j+\ell 2^k-1]$ of $w$ with $k \geq 0$, we can find the occurrences of $y$ in $z$ in $\mathcal{O}(\lg\lg n + \ell)$ time, and compute their representation as described in Corollary 18.*

*Proof* As a preprocessing step, we construct the arrays of Lemma 19 in $\mathcal{O}(n \lg n)$ time. On each such array, we construct the data structure of Lemma 20.

Assume that we get a basic factor and that we want to compute the representation of Corollary 18. In order to find the occurrences of $y$ in $z$, we search the successor of $j$ and the predecessor of $j+\ell 2^k-1$ in the array $A$ storing the starting positions of the occurrences of $y$. Both retrieved values define the range in $A$ where all occurrences of $y$ in $z$ are contained. Within this range, we can compute the representation of Corollary 18 in $\mathcal{O}(\ell)$ time: We linearly process the occurrences in this range from left to right. We return the beginning position of every single occurrence and of every first occurrence. In order to get $\mathcal{O}(\ell)$ running time, we have to omit all occurrences within a run except the first occurrence. To this end, we perform the following procedure using a constant number of longest common prefix queries: Since we scan linearly from left to right, we always access the first two (consecutive) occurrences in the run. First, we compute the length of the overlap of both occurrences; this length is the period of $y$. Subsequently, we determine the length of the run by Lemma 17. Having this length, we skip the remaining occurrences of $y$ in this run (since every occurrence of $y$ is stored in $A$, and we know the run's length and period, we know how many occurrences we have to omit). If the next occurrence of $y$ (starting after this run) starts after the computed predecessor of $j+\ell 2^k-1$, then we terminate.

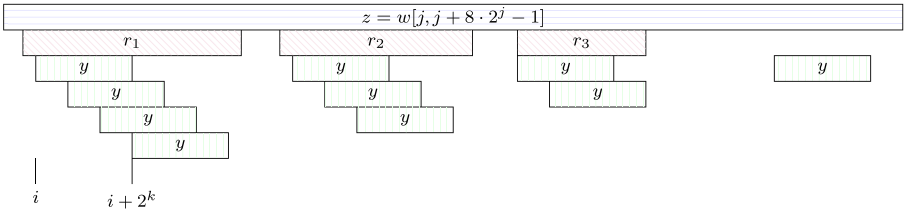


**Fig. 13** Occurrences of the basic factor $y = w[i, i+2^k-1]$ in $z = w[j, j+8\cdot 2^k-1]$ in Corollary 21, with $c = 8$. The overlapping occurrences are part of a run. The occurrences of a run $r$ are represented only by its first (leftmost) occurrence of $y$ in $r$. In total, the representation of the occurrences of $y$ in $z$ is composed of, along with the period of $y$, four starting positions: three first occurrences and one single occurrence

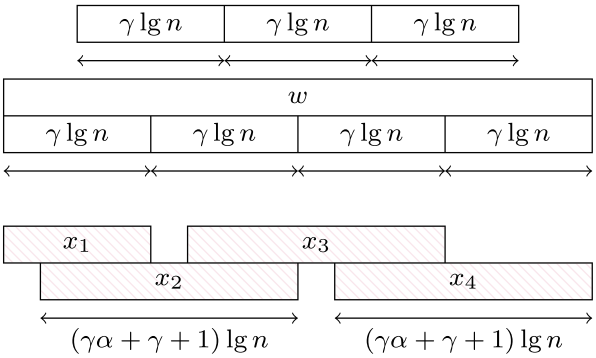


**Fig. 14** Covering the word $w$ with the superblocks $x_m$

Since there are at most $\mathcal{O}(\ell)$ runs and single occurrences of $y$ in $z$, the conclusion follows. □

In order to accelerate the search to $\mathcal{O}(\alpha n)$ time, we first consider $\alpha$-gapped repeats with arms whose lengths are at most $\gamma \lg n$, for an integer constant $\gamma > 0$. Our idea is to first spot the right arm $u_\rho$ and then to apply some techniques to find the left arm $u_\lambda$: If we cover the word $w$ with the set of subwords $\left\{ w[m\lg n + 1, (m + \gamma + 1)\lg n] : 0 \leq m \leq \frac{n}{\lg n} - \gamma - 1 \right\}$ then the right arm has to be contained in at least one of these subwords. The left arm can be at most $\alpha\gamma \lg n$ characters away from the right arm. By stretching every subword of the above cover to the left, the complete gapped repeat is contained in exactly one subword

$$x_m := \begin{cases} w[1, (m + \gamma + 1)\lg n] & \text{if } (m - \gamma\alpha)\lg n + 1 < 1, \\ w[(m - \gamma\alpha)\lg n + 1, (m + \gamma + 1)\lg n] & \text{else,} \end{cases}$$

for an integer $m$ with $\gamma\alpha \leq m \leq \frac{n}{\lg n} - \gamma - 1$. We call each $x_m$ a ***superblock*** (see Fig. 14). Our task is to enhance each superblock with a data structure that allows us to query for possible positions of the left arm. We show that this query can be answered efficiently in the light that the right arm is always contained in the last $\gamma \lg n$ characters of a superblock. The main idea is to use a bit vector marking the starting positions of a basic factor instead of relying on the arrays described in Lemma 19 and used in the proof of Corollary 21. Nevertheless, we need Corollary 21 for finding long-armed gapped repeats (see later Lemma 26).

**Lemma 22** ([10]) *Given a word $x$ and an integer $\beta > \gamma$ such that $|x| = \beta \lg n$, we can process $x$ in $\mathcal{O}(\beta \lg n)$ time such that given a basic factor $y = x[i2^k + 1, (i+1)2^k]$ with $i, k \geq 0$ and $i2^k + 1 > (\beta - \gamma)\lg n$, we can compute a bit vector of length $\beta \lg n$*

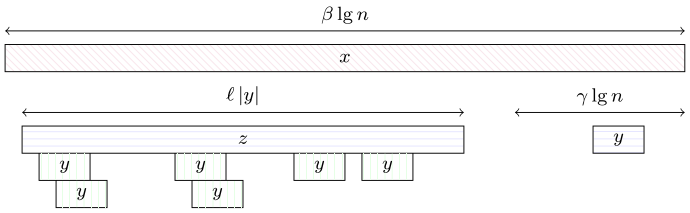


**Fig. 15** The setting of Lemma 23. The difference to Corollary 21 is that $y$ has to appear in the last $\gamma \lg n$ characters of $x$

*marking the beginning positions of the occurrences of $y$ in $x$. The computation of the bit vector takes $\mathcal{O}(\beta)$ time.*

It is easy to use Lemma 22 as a precomputation step in order to find the occurrences of basic factors in small subwords on the precomputed word efficiently (see also Fig. 15):

**Lemma 23** *Let $y$ and $x$ be defined as in Lemma 22, and let $z$ be a subword of $x$ with length $\ell\,|y|$. Given the bit vector of Lemma 22 marking the starting positions of all occurrences of $y$ in $x$, we can represent all occurrences of $y$ in $x$ by the representation described in Corollary 18 in $\mathcal{O}(\ell)$ time.*

*Proof* We assume that our RAM model supports retrieving the location of the most-significant set bit in the binary representation of an integer in constant time.[2] Otherwise, as a preliminary step, we store the mapping $i \mapsto \lfloor \lg i \rfloor + 1$ for every integer $i$ with $1 < i < n$ in a lookup table, in $\mathcal{O}(n)$ time.

By being able to find the location of the most significant set bit of an integer in constant time, we can output all occurrences of $y$ in $z$ in $\mathcal{O}(\ell)$ time. To this end, we scan the bit vector of Lemma 22 in chunks of $\lg n$ bits. By skipping all chunks that represent positions of $x$ before $z$, we only process chunks representing positions of $z$. Given such a chunk, we retrieve the position of the most-significant set bit. This bit represents an occurrence of $y$ that can be retrieved by standard bit-operations. We erase this bit and start to query for the location of the new most-significant set bit. If there is no bit set in the current chunk, we fetch the next chunk.

In order to get the representation of Corollary 18, we need to handle occurrences within a run analogously to the proof of Corollary 21. □

After having described all tools, we start with the presentation of the algorithm finding all maximal $\alpha$-gapped repeats of $w$ with an arm size larger than one. We first deal with the *short* arms:

[2]Commodity computers of the x86 family have an extension instruction set that provides access to the functions `leading zeros count` and `bit scan reverse`, both returning the number of leading zeros of the binary representation.

**Algorithm 1** Scaffold of the proof of Lemma 24 computing all maximal $\alpha$-gapped repeats with an arm length between 1 and $\gamma \lg n$

```
1 foreach 0 ≤ m ≤ n/lgn − γ − 1 do
2     compute data structure of Lemma 22 on superblock x_m
3     foreach 0 ≤ k ≤ lg(γ lgn) do
4         foreach basic factor y_ρ ⊂ x_m[|x_m| − γ lgn + 1, |x_m|] with |y_ρ| = 2^k do
5             Y ← query data structure of Lemma 22 on x_m with pattern y_ρ
6             foreach y_λ ∈ Y do
7                 extend y_λ and y_ρ to a gapped repeat with case analysis (a)–(h)
```

**Lemma 24** *Given a word $w$ and $\alpha \geq 1$, we can find all maximal $\alpha$-gapped repeats $u_\lambda, v, u_\rho$ with $1 < |u_\rho| \leq \gamma \lg n$ occurring in $w$, in $\mathcal{O}(\alpha n)$ time.*

*Proof* A maximal $\alpha$-gapped repeat $u_\lambda, v, u_\rho$ with $|u_\rho| \leq \gamma \lg n$ has a right arm $u_\rho$ that must be contained in a subword $w[m \lg n + 1, (m + \gamma + 1) \lg n]$, for an integer $m$ with $0 \leq m \leq \frac{n}{\lg n} - \gamma - 1$. By fixing the interval where $u_\rho$ may occur (i.e., fix $m$), we know that the entire repeat is contained in $x_m$ (see Fig. 16).

An overview of our algorithm follows (see also Algorithm 1): As a preprocessing step, we equip every superblock with the data structure described in Lemma 22, and create an LCE$^{\leftrightarrow}$ data structure on it. For the actual search, we process each superblock linearly. In each superblock, we search for all maximal $\alpha$-gapped repeats $u_\lambda, v, u_\rho$ that are contained in $x_m$ with $u_\rho$ contained in the suffix of length $\gamma \lg n$ of $x_m$. In order to spot the right arm $u_\rho$ of a possible gapped repeat, we have to iterate over all possible lengths. Since a linear scan over all lengths would take too much time, we first compute a gapped repeat whose right arm is a basic factor, and then try to extend such a gapped repeat to a maximal $\alpha$-gapped repeat. To this end, we iterate over $0 \leq k \leq \lg(\gamma \lg n)$ to find gapped repeats with an arm length between $2^{k+1}$ and $2^{k+2}$ by searching for gapped repeats whose right arms are basic factors of length $2^k$

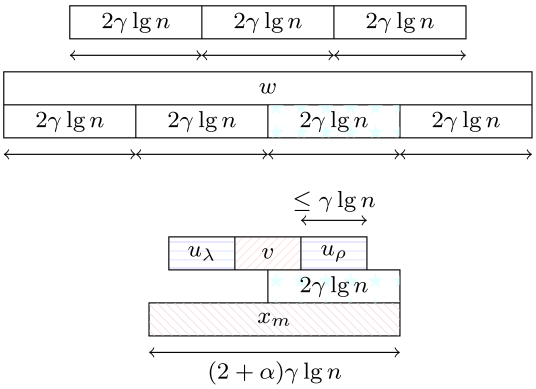


**Fig. 16** Idea of the proof of Lemma 24. The algorithm iterates over $m$, and uses the data structures built on each superblock $x_m$ to spot gapped repeats whose right arms are at the end of $x_m$ while having left arms contained somewhere in $x_m$

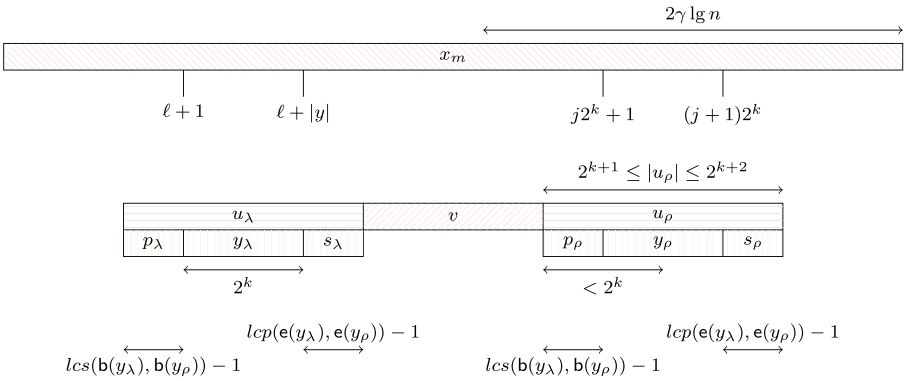


**Fig. 17** Extending a gapped repeat whose right arm is a basic factor. Fixing $x_m$ in the proof of Lemma 24, we try to spot gapped repeats whose arms contain a certain basic factor. If we can extend this gapped repeat to a maximal gapped repeat, we output it. The functions $lcs(i, j)$ and $lcp(i, j)$ denote the longest common prefix and the longest common suffix, respectively, starting at $i$ and $j$

contained in the last $\gamma \lg n$ characters of $x_m$ (since we do not allow the overlapping of those right arms, their number is at most $\gamma \frac{\lg n}{2^k}$).

In more detail, we start with fixing a superblock $x_m$. We want to build maximal $\alpha$-gapped repeats by extending gapped repeats whose arms are basic factors (see Fig. 17). A maximal $\alpha$-gapped repeat $u_\lambda, v, u_\rho$ with $2^{k+1} \le |u_\rho| \le 2^{k+2}$ has a right arm $u_\rho$ that contains at least one subword $y_\rho = w[j2^k + 1, (j+1)2^k]$ starting within the first $2^k$ positions of $u_\rho$ ($\mathsf{b}(u_\rho) \le \mathsf{b}(y_\rho) < \mathsf{b}(u_\rho) + 2^k$). By definition, there is a copy $y_\lambda$ of the subword $y_\rho$ that occurs also within the first $2^k$ positions of $u_\lambda$, namely $y_\lambda = w[\mathsf{b}(u_\lambda) + \mathsf{b}(y_\rho) - \mathsf{b}(u_\rho), \mathsf{b}(u_\lambda) + \mathsf{e}(y_\rho) - \mathsf{b}(u_\rho)]$. Finding the respective copy $y_\lambda$ of $y_\rho$ helps us discovering the location of $u_\lambda$.

Assume that we identified the copy $y_\lambda := w[\ell + 1, \ell + |y_\rho|]$ for an integer $\ell$ with $0 \le \ell < n$; we try to build $u_\lambda$ and $u_\rho$ by extending $y_\lambda$ and $y_\rho$ in both directions, respectively. To this end, we compute the longest factor $p$ of $x_m$ that ends both at $j2^k$ and at $\ell$, and the longest factor $s$ that starts both at $(j+1)2^k + 1$ and at $\ell + |y_\rho| + 1$. If $\ell + |y_\rho| + |s| > j2^k - |p|$, then $y_\lambda$ and $y_\rho$ do not determine a maximal repeat (the gap would have a negative length). Otherwise ($\ell + |y_\rho| + |s| \le j2^k - |p|$), let $s_\lambda$ and $s_\rho$ denote the left and right occurrences of $s$, and let $p_\lambda$ and $p_\rho$ denote the left and right occurrences of $p$, respectively. Then $u_\lambda$ is obtained by concatenating $p_\lambda, x_m[\ell + 1, \ell + |y|]$, and $s_\lambda$, while $u_\rho$ is obtained by concatenating $p_\rho, x_m[j2^k + 1, (j+1)2^k]$, and $s_\rho$. To avoid duplicates, the determined repeat is only reported if its right arm contains the position $j2^k + 1$ of $x_m$ within its first $2^k$ positions.

The algorithm above does not describe how to find the copy $y_\lambda$ (efficiently). We rectify this omission now: Since $|u_\rho| < 2^{k+2}$ and $|y_\rho| = 2^k$, the copy $y_\lambda$ is contained in the subword of $x_m$ of length $\alpha 2^{k+2}$ ending at position $j2^k$. In our preprocessing, we already equipped $x_m$ with the data structure from Lemma 22. We use this data structure as described in Lemma 23: It allows us to retrieve every possible subword $y_\lambda$ inside the subword of length $\alpha 2^{k+2}$ ending at position $j2^k$, in $\mathcal{O}(\alpha)$ time. These occurrences can be single occurrences and occurrences within runs. There are $\mathcal{O}(\alpha)$

single occurrences, and we can process each of them individually to find the maximal $\alpha$-gapped repeat that is determined by $y_\rho$ and this occurrence.

However, it is not efficient to do the same for the occurrences of $y_\rho$ that are within a run (there can be $\Omega(\alpha)$ many occurrences). Instead, we locate the first occurrence in each run (there are at most $\mathcal{O}(\alpha)$ many first occurrences): Let $y$ denote the factor of $y_\rho$. Assume we have a repetition of $y$'s inside the factor of $x_m$ of length $\alpha 2^{k+2}$ ending at position $j2^k$. Let $\ell$ be the starting position of the first occurrence of $y$ in this repetition, and let $p$ be the period of $y$. By Lemma 17, we can determine the maximal $p$-periodic subword (a run of period $p$) $r_\lambda$ of $x_m$ containing this repetition of $y$-occurrences. Similarly, we can determine the maximal $p$-periodic subword (a run of period $p$) $r_\rho$ that contains $y_\rho$. To determine efficiently the $\alpha$-gapped repeats containing $y_\rho$ in the right arm and $y_\lambda$ in the left arm, where $y_\lambda$ is an occurrence in $r_\lambda$, we analyze several cases (see Fig. 18). We group the cases by the fact whether $\mathsf{b}(u_\rho) > \mathsf{b}(r_\rho)$, $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$, or $\mathsf{b}(u_\rho) < \mathsf{b}(r_\rho)$ holds. In each case, we determine the exact location of $u_\lambda$ and $u_\rho$ by querying the $\mathrm{LCE}^{\leftrightarrow}$ data structure on $x_m$:

**(a-c)** Assume $u_\rho$ starts within $r_\rho$, but after $r_\rho$'s first position ($\mathsf{e}(r_\rho) > \mathsf{b}(u_\rho) > \mathsf{b}(r_\rho)$). Then $u_\lambda$ starts at the first position of $r_\lambda$ (otherwise, we could extend both arms to the left, a contradiction to the maximality of the repeat).

(a) If $u_\rho$ ends at a position to the right of $r_\rho$, then $u_\lambda$ ends at a position to the right of $r_\lambda$ (otherwise, it would again contradict to the maximality). Moreover, the suffix of $u_\lambda$ occurring after the end of $r_\lambda$ and the suffix of $u_\rho$ occurring after the end of $r_\rho$ are equal to the longest equal substring starting at positions $\mathsf{e}(r_\lambda) + 1$ and $\mathsf{e}(r_\rho) + 1$, and can be computed by a longest common prefix query on $x_m$.

(b) If $u_\rho$ ends exactly at the same position as $r_\rho$ ($\mathsf{e}(u_\rho) = \mathsf{e}(r_\rho)$), then $u_\rho$ is periodic with the period $p$ as $r_\rho$. We compute the longest $p$-periodic prefix $u'$ of $r_\lambda$ that is a suffix of $r_\rho$. By knowing the period $p$ (determined by two subsequent occurrences of $y_\rho$) and the length of $r_\lambda$ and $r_\rho$, the factor $u'$ can be determined in constant time.

Since $u_\lambda$ is longer than $p$, the $\alpha$-gapped repeats under consideration have the left arm $u_\lambda := r_\lambda[1, |u'| - pi]$ and the right arm $u_\rho := r_\rho[|r_\rho| - (|u'| - pi) + 1, |r_\rho|]$ for $i \geq 0$ such that the gap $v := w[\mathsf{e}(u_\lambda) + 1, \mathsf{b}(u_\rho) - 1]$ respects the condition $|u_\lambda v| \leq \alpha\,|u_\lambda|$.

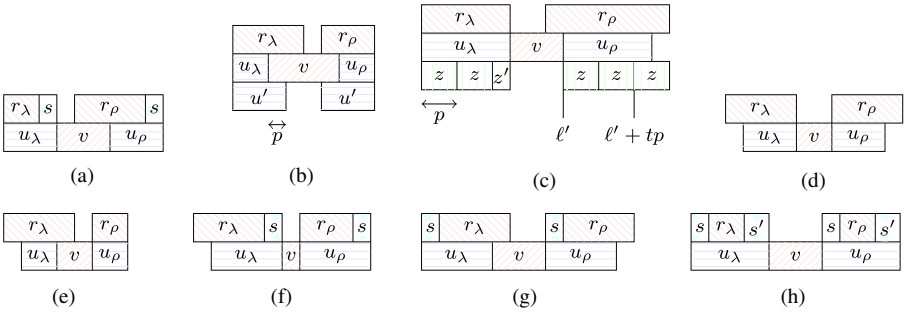


**Fig. 18** Spotting gapped repeats with periodicity. This is done by a case analysis in the proof of Lemma 24. Each case is depicted in order (from *left* to *right*, *top* to *bottom*)

(c) The final case is when $u_\rho$ ends at a position of $r_\rho$, prior to $r_\rho$'s last position ($\mathsf{e}(u_\rho) < \mathsf{e}(r_\rho)$). In that case, we get that $u_\lambda = r_\lambda$ (otherwise, we could extend both arms to the right). The left arm $u_\lambda$ is equal to a factor $z^h z'$ for an integer $h \geq 2$, where $z = r_\lambda[1, p]$, $p$ is the period of $r_\lambda$, and $z'$ is a prefix of $z$.

We can get the position of the first and the last occurrence of $z$ in $r_\rho$. If the first occurrence starts at $\ell'$, then the starting positions of the succeeding occurrences of $z$ form the arithmetic progression $\ell', \ell' + p, \ldots, \ell' + tp$ for an integer $t \geq 1$. For each $0 \leq i \leq t$, we let $u_\rho$ start at position $\ell' + ip$ (and check whether $u_\rho \equiv u_\lambda$ by knowing the length of $u_\rho$ and $r_\rho$).

Finally, additional care has to be taken for the border cases. If $u_\rho$ is a suffix of $r_\lambda$, we have to check that we cannot extend simultaneously $u_\rho$ and $u_\lambda$ to the right. ($u_\rho$ cannot be a prefix of $r_\rho$ since we assumed for the cases (a-c) that $\mathsf{b}(u_\rho) > \mathsf{b}(r_\rho)$.)

**(d-f)** Assume $u_\rho$ starts at the first position of $r_\rho$ ($\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$).

(d) If $u_\rho$ ends at a position inside $r_\rho$, prior to its last position ($\mathsf{b}(r_\rho) = \mathsf{e}(u_\rho) < \mathsf{e}(r_\rho)$), then $u_\lambda$ ends at the last position of $r_\lambda$ (otherwise, both arms could be extended to the right). This means that the gap between the two arms is uniquely determined, and that the arms are periodic for a period $p$. We compute the longest $p$-periodic suffix of $r_\lambda$ that is a prefix of $r_\rho$, and check whether the two occurrences of this factor determine a maximal $\alpha$-gapped repeat.

(e) If $u_\rho$ ends at the last position of $r_\rho$, then we know the exact location of $u_\rho$. We can proceed analogously to case (c) by symmetry.

(f) Finally, if $u_\rho$ ends at a position to the right of $r_\rho$, then $u_\lambda$ ends also after $r_\lambda$ ($\mathsf{e}(u_\rho) > \mathsf{e}(r_\rho) \wedge \mathsf{e}(u_\lambda) > \mathsf{e}(r_\lambda)$), and the suffix of $u_\lambda$ occurring after $r_\lambda$ is equal to the suffix of $u_\rho$ occurring after $r_\rho$. We determine this suffix by a longest common prefix query on $x_m$. With this suffix, we obtain the location of both arms.

**(g-h)** The last case is when $u_\rho$ starts at a position to the left of $r_\rho$ ($\mathsf{b}(u_\rho) < \mathsf{b}(r_\rho)$). Then $u_\lambda$ starts at a position before the first position of $r_\lambda$ ($\mathsf{b}(u_\lambda) < \mathsf{b}(r_\lambda) < \mathsf{e}(u_\lambda)$); the prefix of $u_\rho$ occurring before the beginning of $r_\rho$ is equal to the prefix of $u_\lambda$ occurring before $r_\lambda$. The length of these prefixes can be retrieved with a longest common suffix query.

(g) If $\mathsf{e}(u_\rho) \leq \mathsf{e}(r_\rho)$, then $\mathsf{e}(u_\lambda) \leq \mathsf{e}(r_\lambda)$, and we are done.

(h) Otherwise ($\mathsf{e}(u_\rho) > \mathsf{e}(r_\rho)$), $u_\lambda$ and $u_\rho$ contain $r_\lambda$ and $r_\rho$, respectively. We determine $u_\lambda$ be the longest common substring starting at $\mathsf{e}(r_\lambda)+1$ and $\mathsf{e}(r_\rho)+1$.

To sum up, we can determine the locations of both arms of the repeat in all cases (a-g) by the LCE$^{\leftrightarrow}$ data structure in constant time. For each found pair of arms, we have to check whether

- the arms form a (valid) maximal $\alpha$-gapped repeat,
- their length is between $2^{k+1}$ and $2^{k+2}$, and whether
- the right arm contains position $j2^k + 1$ of $x_m$ within its first $2^k$ positions.
- for the cases (b) and (c) we check that the right arm contains $y_\rho[1]$ in its first $2^k$ positions.

The last two conditions ensure that no gapped repeat is reported twice within the same superblock.

This concludes our analysis for finding all $\alpha$-gapped repeats of $x_m$, for each $m$ *separately*. We can ensure that our algorithm finds and outputs each maximal repeat exactly once when moving from $x_m$ to $x_{m+1}$. To this end, we check that the right arm of each repeat we find is not completely contained in $x_m$ (so it is already found). This condition can be easily imposed in our search: when constructing the arms that are determined by a single occurrence of $y_\rho$, we check the containment condition separately; when constructing the arms determined by a run of $y_\rho$-occurrences, we have to impose the condition that the right arm extends out of $x_m$ when searching the starting positions of the possible arms.

Finally, we compute the complexity of the algorithm. We need $\mathcal{O}(n)$ preprocessing time for $w$, and $\mathcal{O}(|x_m|) = \mathcal{O}(\gamma\alpha\lg n)$ preprocessing time for each $x_m$. For fixed $m$, $k$, and $j$, our process takes $\mathcal{O}(\alpha + \mathrm{occ}_{j,m,k})$ time, where $\mathrm{occ}_{j,m,k}$ is the number of all maximal $\alpha$-gapped repeats determined with the fixed values $m$, $j$, $k$. Iterating over all values $m$, $j$ and $k$ gives the overall time complexity of the algorithm, which is

$$\mathcal{O}\left(\underbrace{n}_{\text{precomp. on } w} + \underbrace{\sum_{m=0}^{\frac{n}{\lg n}}}_{\text{for all } x_m}\left(\underbrace{\gamma\alpha\lg n}_{\text{precomp. on } x_m} + \sum_{k=0}^{\lg(\gamma\lg n)}\left(\underbrace{\sum_{j=0}^{\gamma\frac{\lg n}{2^k}}\left(\alpha + \mathrm{occ}_{j,m,k}\right)}_{\text{for all } y_\rho}\right)\right)\right) \subseteq \mathcal{O}(\alpha n),$$

since the total number of maximal $\alpha$-gapped repeats is $\mathcal{O}(\alpha n)$. □

In the last part of this section, we show how to find all maximal $\alpha$-gapped repeats with longer arms. To this end, we introduce a so-called block-representation of a word $w$: We partition $w$ into subwords of $\lg n$ characters $w[1 + i\lg n, (i + 1)\lg n]$ for every $0 \leq i < n/\lg n$ (we can ensure that every block has the same number of characters by padding $w$ with dummy characters such that $\frac{n}{\lg n}$ is integer). We call these subwords ***blocks*** of $w$. The lexicographic order on $\Sigma$ induces a linear order on the blocks. Since there are at most $n/\lg n$ different blocks, we can enumerate the blocks with numbers from 1 to at most $n/\lg n$ such that the $j$-th smallest block gets the number $j$. For our purpose, an enumeration from 1 to $n$ (possibly omitting some values) is sufficient. Before showing how to compute the enumeration, we start with the definition of the block-representation: A word $w'$ is the ***block-representation*** of $w$ if

- $w'$ is a word of length $n/\lg n$ on the alphabet $\{, \ldots, n\}$, and
- $w'[i] = j$ $(1 \leq i \leq n/\lg n)$ if and only if the block of $w$ with number $j$ is equal (in the sense of $\equiv$) to $w[1 + (i - 1)\lg n, i\lg n]$.

It is easy to provide a linear-time algorithm computing the (larger) enumeration: We start with building the $\mathrm{LCE}^{\leftrightarrow}$ data structure of $w$. Subsequently, we cluster together the suffixes of the suffix array that share a common prefix of length at least $\lg n$. There are at most $n$ different clusters; hence we can enumerate all clusters, starting

from 1 to at most $n$. A block is associated with the number of a cluster if the cluster contains the suffix that starts at the same position as the block.

**Lemma 25** *We can build the block representation $w'$ of a word $w$ of length $n$ in $\mathcal{O}(n)$ time.*

Equipped with Lemma 25, we are ready to present the algorithm finding long-armed maximal $\alpha$-gapped repeats with large values for $\alpha$:

**Lemma 26** *Given a word $w$ of length $n$, and an $\alpha \geq \lg n$, we can find all maximal $\alpha$-gapped repeats $u_\lambda, v, u_\rho$ with $|u_\rho| > \gamma \lg n$ occurring in $w$, in $\mathcal{O}(\alpha n)$ time.*

*Proof* The general approach in proving this lemma is similar to the techniques of the proof of Lemma 24. Essentially, when identifying a new maximal $\alpha$-gapped repeat, we try to fix the place and length of the right arm $u_\rho$ of the respective repeat, which restricts the place where the left arm $u_\lambda$ occurs. This allows us to fix a long enough subword of $w$ as being part of the right arm, detect its occurrences that are possibly contained in the left arm, and, finally, to efficiently identify the actual repeat. The main difference is that we cannot use the result of Lemma 22, as we have to deal with repeats with arms longer than $\gamma \lg n$. Instead, we use the structures constructed in Corollary 21. However, to get the stated complexity, we apply this lemma to the block-representation of $w$, rather than to $w$ itself.

In this sense, the first step is to construct the block-representation $w'$ of $w$. Subsequently, we construct the LCE$^{\leftrightarrow}$ data structures of $w$ and $w'$, as well as the data structure of Corollary 21 for the word $w'$. Every construction step is conducted in $\mathcal{O}(n)$ time.

Like in the proof of Lemma 24, we iterate over all possible arm lengths. For an integer $k$, we search for all maximal $\alpha$-gapped repeats $u_\lambda, v, u_\rho$ in $w$ with $2^{k+1}\lg n \leq |u_\lambda| \leq 2^{k+2}\lg n$.

For the following, we fix $k$. Similar to the block-representation, we partition the word $w$ into subwords, but this time into subwords of length $2^k \lg n$, called ***k*-blocks**. Again (as for blocks), we assume that each $k$-block has the same number of characters.

The idea of this partition is the following: If a maximal $\alpha$-gapped repeat $u_\lambda, v, u_\rho$ with $2^{k+1}\lg n \leq |u_\lambda| \leq 2^{k+2}\lg n$ exists, then it contains a $k$-block within its first $2^k \lg n$ positions. Let $z$ be the *first* $k$-block contained in $u_\rho$. Since $u_\rho$ contains $z$, the left arm $u_\lambda$ also contains an occurrence of $z$. However, this occurrence is *not necessarily* starting at a position $j \lg n + 1$ for an integer $j \geq 0$; this means that it does not have to start with a block. In this sense, we cannot capture this occurrence with our block-representation. Nevertheless, at least one of the subwords of length $2^{k-1}\lg n$ starting within the first $\lg n$ positions of $z$ has an occurrence in $u_\lambda$ that starts with a block (the subword itself in $z$ does *not* have to start with a block, see also Fig. 19). In order to find such a subword, we iterate over all $\lg n$ positions. Let us fix a subword $y_\rho$ of length $2^{k-1}\lg n$ that starts within the first $\lg n$ positions of $z$. As said, $y_\rho$ is not necessarily a sequence of $2^{k-1}$ blocks, i.e., $y_\rho$ is not represented by a subword of $w'$

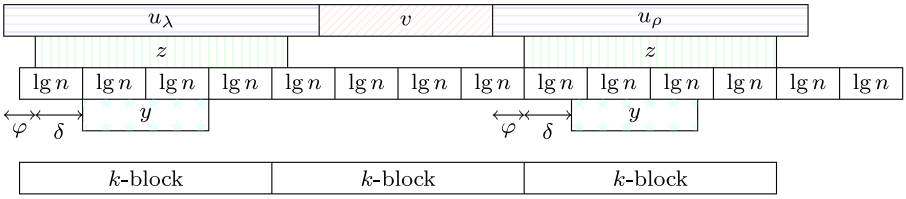


**Fig. 19** Finding occurrences of $y$ starting with a block in proof of Lemma 26. The distance $\delta$ is smaller than lg$n$, and the distance $\varphi$ is smaller than $2^k$lg$n$, since $z$ is the first $k$-block contained in $z$

in general. We look for an occurrence of $y_\rho$ starting at one of the $\alpha 2^{k+2}$lg$n$ positions to the left of $z$. For each such occurrence $y_\lambda$ that correspond to a sequence of blocks, we try to extend both $y_\lambda$ and $y_\rho$ to an $\alpha$-gapped repeat (having the same idea as for Lemma 24 in mind).

Let $y$ denote the factor of $y_\rho$. By binary searching the suffix array of $w'$ (using longest common prefix queries on $w$ to compare the factors of lg$n$ characters of $y$ and the blocks of $w'$, at each step of the search) we try to detect a factor of $w'$ that encodes a word equal to $y$. Assume that we can find such a sequence $y'$ of blocks in $w'$ (otherwise, $y$ cannot correspond to a sequence of blocks from $u_\lambda$, so we choose a new $y_\rho$ by taking the next starting position). By Corollary 21, we can spot the occurrences of $y'$ in the $\alpha 2^{k+2}$ blocks of $w'$ that occur before the blocks of $z$, in $\mathcal{O}(\lg\lg |w'| + \alpha)$ time; this range corresponds to an interval of $w$ with a length of $\alpha 2^{k+2}$lg$n$.

Each of the occurrences of $y'$ fixes a possible left arm $u_\lambda$; this arm, together with the corresponding arm $u_\rho$ can be constructed with the same techniques as in Lemma 24. In the case of a single occurrence $u_\lambda$ (there are at most $\mathcal{O}(\alpha)$ many of that kind), we extend $u_\lambda$ and $u_\rho$ in both directions to obtain two arms, for which we have to check if they define a valid $\alpha$-gapped repeat. In order to avoid duplicates, we check that the length of each arm is between $2^{k+1}$ and $2^{k+2}$, and that $z$ is the first $k$-block of the right arm.

Complications occur when some occurrences of $y'$ are within a run. Given a run of occurrences of $y'$, we cannot determine the period of $y$ in general, but a multiple of this period. More precisely, we know that the period of $y$ is a multiple of the block length lg$n$. However, this is not a problem, since the subword $y$ in $u_\rho$ corresponds to a block sequence from $u_\lambda$, hence definitely to one of the subwords encoded in the run of occurrences of $y'$. Analogously to the analysis in Lemma 24, we can determine the maximal factor containing $y$ such that it has the same period as the repetition of $y'$-occurrences (with the period measured in $w$).

It remains to prove that each maximal gapped repeat is counted only once:

- Assume that there are two factors $y'_1$ and $y'_2$ of $w'$ that correspond to two separate factors $y_1$ and $y_2$, each of length $2^{k-1}$lg$n$, occurring in the first lg$n$ characters of $z$. Since $y'_1$ and $y'_2$ cannot define the same repeat, the distance between $y'_1$ and $y'_2$ is at least one block long, i.e., the distance between $y_1$ and $y_2$ is at least lg$n$, a contradiction.
- Similarly, if we have found a subword $y$ occurring in the first lg$n$ characters of a $k$-block $z$ such that $y$ determines an $\alpha$-gapped maximal repeat, then the same

maximal repeat will not be determined by a subword of another $k$-block, since $z$ is the first $k$-block of $u_\rho$.

Let us evaluate the complexity of the above described algorithm. The preprocessing, i.e., the construction of $w'$ and all of the needed data structures, takes $\mathcal{O}(n)$ time. We have multiple nested iterations:

(a) We iterate over all $0 \leq k \leq \lg \frac{n}{\lg n} - 2$.
(b) For a fixed $k$, we examine every $k$-block $z$, and there are $\frac{n}{2^k \lg n}$ many.
(c) For a fixed $z$, we analyze each subword $y_\rho$ of length $2^{k-1}\lg n$ starting within the first $\lg n$ positions of the chosen $k$-block $z$.
(d) For each such subword $y_\rho$ we find the occurrences of the block encoding the occurrence of $y_\rho$ in $u_\rho$ in $\mathcal{O}\left(\lg \frac{n}{\lg n} + \lg\lg n + \alpha\right)$ time.
(e) For each of the $\mathcal{O}(\alpha)$ single occurrences $u_\lambda$, we check whether it is possible to extend $y_\lambda$ and $y_\rho$ to a maximal $\alpha$-gapped repeat in $\mathcal{O}(1)$ time. We also have $\mathcal{O}(\alpha)$ occurrences of the block encoding $u_\rho$ in runs, all of them are processed in $\mathcal{O}(\alpha + \mathrm{occ}_{z,y})$ time overall, where $\mathrm{occ}_{z,y}$ is the number of maximal $\alpha$-gapped repeats we find for a given $z$ and $y$.

Overall, this adds up to

$$\sum_{k=0}^{\lg \frac{n}{\lg n}-2} \underbrace{\frac{n}{2^k \lg n}}_{\#z} \underbrace{\lg n}_{\#y_\rho} \underbrace{\left(\lg \frac{n}{\lg n} + \lg\lg n + \alpha + \mathrm{occ}_{z,y}\right)}_{\text{process every possible } y_\lambda} \in \mathcal{O}(n \lg n + \alpha n), \tag{12}$$

since the total number of maximal $\alpha$-gapped repeats in $w$ is upper bounded by $\mathcal{O}(\alpha n)$. Since $\alpha \geq \lg n$, the statement of the lemma follows. □

**Lemma 27** *Given a word $w$ of length $n$, and an $\alpha < \lg n$, we can find all maximal $\alpha$-gapped repeats $u_\lambda, v, u_\rho$ with $|u_\rho| > \gamma \lg n$ occurring in $w$, in $\mathcal{O}(\alpha n)$ time.*

*Proof* Initially, we run the algorithm of Lemma 26 only for $k > \lg\lg n$ to find all maximal $\alpha$-gapped repeats with an arm length of at least $2^{\lg\lg n} \lg n$. We see that (12) with $k > \lg\lg n$ yields $\mathcal{O}(\alpha n)$ time.

In the rest of this proof, we search for maximal $\alpha$-gapped repeats whose arms' length is upper bounded by $2^{\lg\lg n+1}\lg n = 2(\lg n)^2$. Setting $\ell := \alpha \cdot 2(\lg n)^2 + 2(\lg n)^2 = 2(\alpha+1)(\lg n)^2$, the lengths of those gapped repeats is at most $\ell$. If we cover $w$ with the set of subwords $\{w[1+m\ell, (m+2)\ell] : 0 \leq m \leq n/\ell - 2\}$, then such an $\alpha$-gapped repeat is contained in (at least) one subword of this cover.

In this sense, we can apply the algorithm of Lemma 26 to each subword in the cover (iterating over all $m$) in order to detect all maximal $\alpha$-gapped repeats with an arm length of at least $2^{\lg\lg(2\ell)+1}\lg(2\ell)$ contained completely in each subword of the cover. Equation (12) with $k \geq \lg\lg(2\ell)$ gives $\mathcal{O}(\alpha\ell + \mathrm{occ}_m)$ running time for the algorithm running on a subword of the cover (each is of length $2\ell$), where $\mathrm{occ}_m$ is the number of occurrences of all maximal $\alpha$-gapped repeats with the above described arm length in the $m$-th subword of the cover. Summing over all subwords of the cover,

we get $\mathcal{O}(\alpha n)$ time in total. By knowing the overlap of two subsequent subwords of the cover, it is easy to adopt the algorithm of Lemma 26 in such a way that no gapped repeat is reported twice.

It is left to find all maximal $\alpha$-gapped repeats with an arm length smaller than $2^{\lg\lg(2\ell)+1}\lg(2\ell)$. For $n$ large enough, it holds that $2^{\lg\lg(2\ell)+1}\lg(2\ell) \leq \gamma\lg n$, since $\alpha \leq \lg n$. But those maximal $\alpha$-gapped repeats are already found by the algorithm of Lemma 24 running in $\mathcal{O}(\alpha n)$ time. □

Putting the results of Lemmas 15, 24, 26 and 27 together, we get the following theorem.

**Theorem 28** *Given a word $w$ and an $\alpha \geq 1$, we can compute $\mathcal{G}_\alpha(w)$ in $\mathcal{O}(\alpha n)$ time.*

Analogously, we can compute $\mathcal{G}^{\mathsf{T}}_\alpha(w)$, generalizing the algorithm of [18]:

**Corollary 29** *Given a word $w$ and $\alpha \geq 1$, we can compute $\mathcal{G}^{\mathsf{T}}_\alpha(w)$ in $\mathcal{O}(\alpha n)$ time.*

*Proof* We construct the LCE$^{\leftrightarrow}$ data structure of $ww^{\mathsf{T}}$ to test in constant time whether a factor $w[i, j]^{\mathsf{T}}$ occurs at a position in $w$. On searching the $\alpha$-gapped palindromes $u_\lambda, v, u_\rho$ (with $u_\rho \equiv u_\lambda{}^{\mathsf{T}}$), we split $w$ into blocks and $k$-blocks (like in Lemma 26) for each $k \leq \lg|w|$, to check whether there exists a gapped palindrome $u_\lambda, v, u_\rho$ with $2^k \leq |u_\lambda| \leq 2^{k+1}$. This search is conducted analogously to the case of gapped repeats, with the difference that when fixing the occurrence of a factor $y$ in $u_\rho$, we have to look for the occurrences of $y^{\mathsf{T}}$ in the subword of length $\mathcal{O}(\alpha\,|u_\rho|)$ preceding it; the LCE$^{\leftrightarrow}$ data structure of $ww^{\mathsf{T}}$ is useful for this task, since it allows us to search the mirror images of factors of $w$ inside $w$ in constant time. □

## 4 Conclusion

We presented two major achievements that shed more light on the combinatorial and computational aspects of $\alpha$-gapped repeats. First, we succeeded in giving concrete bounds for the maximum number of maximal $\alpha$-gapped repeats and maximal $\alpha$-gapped palindromes of a word. Second, we elaborated two algorithms computing the set of all maximal $\alpha$-gapped repeats and the set of all maximal $\alpha$-gapped palindromes, respectively, of a word of length $n$ on an integer alphabet. The achieved combinatorial bounds and the time bounds of the algorithms are asymptotically optimal.

Nevertheless, we deliberately omitted the exact memory consumption of the created data structures (currently $\mathcal{O}(n)$ words). With a more careful analysis of the space, we could give preciser bounds (e.g., measured in bits) of the selected data structures, perhaps yielding an algorithm working on succinct space. It is also interesting to further refine both algorithms to such an extent that their running time is output sensitive, i.e., having $\mathcal{O}(n + |\mathcal{G}_\alpha(w)|)$ and $\mathcal{O}(n + |\mathcal{G}^{\mathsf{T}}_\alpha(w)|)$ worst case running time, respectively, for a word $w$. Additionally, we think that our result can serve as a basis for practical solutions, since most of the used data structures are well studied. In this sense, we also want to get better constants for the combinatorial bounds.

The current constants seem unreasonably large. We think that a more precise analysis allows us to shrink the constants to a smaller number.

The presented bounds are still valid when working with the more general definition of $\alpha$-gapped $\varphi$-repeats or $\alpha$-gapped $\varphi$-palindromes: Let $\varphi : \Sigma^* \to \Sigma^*$ be a word isomorphism, i.e. $\varphi(uv) = \varphi(u)\varphi(v)$, and $\varphi$ is bijective. For instance, $id(v) = v$ is a word isomorphism. A subword of the form $uv\varphi(u)$ $(uv\varphi(u)^\intercal)$ is called $\alpha$-gapped $\varphi$-repeat ($\varphi$-palindrome) iff $uvu$ $(uvu^\intercal)$ is an $\alpha$-gapped repeat (palindrome). It is easy to see that our results are also applicable for $\alpha$-gapped $\varphi$-repeats or $\alpha$-gapped $\varphi$-palindromes. This generalizes the analysis in [18, Sec. 5]; there, $\varphi$ is equal to a function building the base complements of a DNA string. The problem of enumerating all 1-gapped $\varphi$-repeats or all 1-gapped $\varphi$-palindromes was already investigated in [11, 12].

**Acknowledgements** We thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments and suggestions.

The work of Florin Manea was supported by the DFG grant 596676.



## References

1. Badkobeh, G., Crochemore, M.: Computing maximal-exponent factors in an overlap-free word. J. Comput. Syst Sci. **82**(3), 477–487 (2016)
2. Bannai, H., Tomohiro, I., Inenaga, S., Nakashima, Y., Takeda, M., Tsuruta, K.: The runs theorem. arXiv:1406.0263 (2014)
3. Brodal, G.S., Lyngsø, R.B., Pedersen, C.N.S., Stoye, J.: Finding maximal pairs with bounded gap Proc. CPM, volume 1645 of LNCS, pp. 134–149 (1999)
4. Crochemore, M., Rytter, W.: Usefulness of the karp-Miller-Rosenberg algorithm in parallel computations on strings and arrays. Theor. Comput. Sci. **88**(1), 59–82 (1991)
5. Crochemore, M., Tischler, G.: Computing longest previous non-overlapping factors. Inf. Process. Lett. **111**(6), 291–295 (2011)
6. Crochemore, M., Iliopoulos, C.S., Kubica, M., Rytter, W., Walen, T.: Efficient Algorithms for Two Extensions of LPF table: The Power of Suffix Arrays Proc. SOFSEM, volume 5901 of LNCS, pp. 296–307 (2010)
7. Crochemore, M., Kolpakov, R., Kucherov, G.: Optimal Bounds for Computing $\alpha$-Gapped Repeats Proc. LATA, pp. 245–255 (2016)
8. Dumitran, M., Manea, F.: Longest Gapped Repeats and Palindromes Proc. MFCS, volume 9234 of LNCS, pp. 205–217 (2015)
9. Fischer, J., Heun, V.: Space efficient preprocessing schemes for range minimum queries on static arrays. SIAM J. Comput. **40**(2), 465–492 (2011)
10. Gawrychowski, P., Manea, F.: Longest $\alpha$-Gapped Repeat and Palindrome Proc. FCT, volume 9210 of LNCS, pp. 27–40 (2015)
11. Gawrychowski, P., Manea, F., Mercas, R., Nowotka, D., Tiseanu, C.: Finding Pseudo-repetitions Proc. STACS, volume 20 of LIPIcs, pp. 257–268 (2013)
12. Gawrychowski, P., Manea, F., Nowotka, D.: Testing Generalised Freeness of Words Proc. STACS, volume 25 of LIPIcs, pp. 337–349 (2014)
13. Gawrychowski, P., Tomohiro, I., Inenaga, S., Köppl, D., Manea, F.: Efficiently finding all maximal $\alpha$-gapped repeats Proc. STACS, volume 47 of LIPIcs, pp. 39:1–39:14 (2016)

14. Gusfield, D.: Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press, Cambridge (1997)
15. Kärkkäinen, J., Sanders, P., Burkhardt, S.: Linear work suffix array construction. J. ACM **53**, 918–936 (2006)
16. Kociumaka, T., Radoszewski, J., Rytter, W., Walen, T.: Efficient Data Structures for the Factor Periodicity Problem Proc. SPIRE, volume 7608 of LNCS, pp. 284–294 (2012)
17. Kolpakov, R., Kucherov, G.: Finding Repeats with Fixed Gap Proc. SPIRE, pp. 162–168 (2000)
18. Kolpakov, R., Kucherov, G.: Searching for gapped palindromes. Theor. Comput. Sci. **410**(51), 5365–5373 (2009)
19. Kolpakov, R., Podolskiy, M., Posypkin, M., Khrapov, N.: Searching of Gapped Repeats and Subrepetitions in a Word Proc. CPM, volume 8486 of LNCS, pp. 212–221 (2014)
20. Manacher, G.: A new linear-time "on-line" algorithm for finding the smallest initial palindrome of a string. J. ACM **22**(3), 346–351 (1975)
21. Ruzic, M.: Making deterministic signatures quickly. ACM Transactions on Algorithms 5 (3) (2009)
22. Tanimura, Y., Fujishige, Y., Tomohiro, I., Inenaga, S., Bannai, H., Takeda, M.: A faster algorithm for computing maximal $\alpha$-gapped repeats in a string Proc. SPIRE, volume 9309 of LNCS, pp. 124–136 (2015)