Improved Approximation Guarantees for Shortest Superstrings using Cycle Classification by Overlap to Length Ratios

Matthias Englert
M.Englert@warwick.ac.uk
University of Warwick
Coventry, United Kingdom

Nicolaos Matsakis
nickmatsakis@gmail.com
Unaffiliated
Athens, Greece

Pavel Veselý*
vesely@iuuk.mff.cuni.cz
Charles University
Prague, Czech Republic

ABSTRACT

In the Shortest Superstring problem, we are given a set of strings and we are asking for a common superstring, which has the minimum number of characters. The Shortest Superstring problem is NP-hard and several constant-factor approximation algorithms are known for it. Of particular interest is the GREEDY algorithm, which repeatedly merges two strings of maximum overlap until a single string remains. The GREEDY algorithm, being simpler than other well-performing approximation algorithms for this problem, has attracted attention since the 1980s and is commonly used in practical applications.

Tarhio and Ukkonen (TCS 1988) conjectured that GREEDY gives a 2-approximation. In a seminal work, Blum, Jiang, Li, Tromp, and Yannakakis (STOC 1991) proved that the superstring computed by GREEDY is a 4-approximation, and this upper bound was improved to 3.5 by Kaplan and Shafrir (IPL 2005).

We show that the approximation guarantee of GREEDY is at most $(13+\sqrt{57})/6 \approx 3.425$. Furthermore, we prove that the Shortest Superstring can be approximated within a factor of $(37+\sqrt{57})/18 \approx 2.475$, improving slightly upon the currently best $2\frac{11}{23}$-approximation algorithm by Mucha (SODA 2013).

CCS CONCEPTS

• **Theory of computation → Approximation algorithms analysis**.

KEYWORDS

shortest common superstring, approximation algorithms

ACM Reference Format:
Matthias Englert, Nicolaos Matsakis, and Pavel Veselý. 2022. Improved Approximation Guarantees for Shortest Superstrings using Cycle Classification by Overlap to Length Ratios. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3520001

*Part of this work was done when the author was at the University of Warwick. Partially supported by European Research Council grant ERC-2014-CoG 647557, by GA ČR project 19-27871X, and by Center for Foundations of Modern Computer Science (Charles University project UNCE/SCI/004).

STOC '22, June 20–24, 2022, Rome, Italy

1 INTRODUCTION

In the Shortest Superstring problem (SSP), we are given a set S of strings over a finite alphabet, and we are asking for a string of minimum length, which contains each member of S as a substring. SSP has found important applications in various scientific domains [9]. One of the early uses was for DNA sequencing [16, 21], where a DNA molecule consisting of four different nucleotides (Adenine, Thymine, Guanine, and Cytosine) is gradually assembled from DNA fragments, which can be viewed as an instance of SSP over a quaternary alphabet. Interestingly, SSP has been used to study how effectively viruses compress their genome by overlapping genes [11].

SSP is NP-hard, even when the alphabet is binary [8]. Moreover, SSP is APX-hard [3] as it is not $(\frac{333}{332} - \epsilon)$-approximable for any constant $\epsilon > 0$ unless P = NP [14]. There exists a plethora of constant-factor SSP approximation algorithms, the currently best of which has an approximation ratio upper bound of $2\frac{11}{23} = \frac{57}{23} \approx 2.478$ [20]. Blum, Jiang, Li, Tromp, and Yannakakis [3] showed that the GREEDY algorithm, which repeatedly merges two strings of maximum overlap (breaking ties arbitrarily) until a single string remains, computes a 4-approximate superstring. Additionally, Blum et al. gave two simple variants of GREEDY, namely TGREEDY with approximation ratio at most 3 and MGREEDY with ratio at most 4. A series of improved approximation algorithms followed, most of which were published in the 1990s [1, 2, 4, 5, 15, 20, 24, 26]. It is worth noting that several of these algorithms are significantly more involved than the natural GREEDY algorithm.

The GREEDY algorithm for SSP was proposed by Gallant, Maier, and Storer [7]. Tarhio and Ukkonen [25] and independently Turner [27] showed that GREEDY gives a $\frac{1}{2}$-approximation for the maximum string compression. The string compression equals the number of characters that a superstring algorithm saves from the total length of all strings in S, i.e., it is the total overlap between all pairs of adjacent strings across the superstring. This result, however, does not imply a constant approximation ratio upper bound for GREEDY, for the length metric.

Moreover, Tarhio and Ukkonen showed that the approximation ratio of GREEDY is at least 2, by considering the input $S = \{ab^k, b^{k+1}, b^k a\}$, for which, depending on the tie-breaking choice, GREEDY will either output the shortest superstring or a superstring of length twice the minimum, when $k \to \infty$ (for input $\{c(ab)^k, (ba)^k, (ab)^k c\}$, the ratio also tends to 2, but no tie-breaking is involved). Finally, Tarhio and Ukkonen conjectured that GREEDY is a 2-approximation algorithm, forming the long-standing *Greedy Conjecture*. By utilizing the Overlap Rotation Lemma of [4] in the proof of Blum et al. [3], Kaplan and Shafrir [13] showed that GREEDY gives a 3.5-approximation.

The GREEDY algorithm has been commonly used in practical applications when it becomes infeasible to compute an optimal solution [11, 17, 21]. Also, the good performance of GREEDY in practice has been documented within a probabilistic framework [6, 18]. In this paper, we make the first progress on the approximation guarantee of GREEDY since 2005.

Theorem 1.1. *The approximation ratio of GREEDY is at most* $(13+\sqrt{57})/6 \approx 3.425$.

Furthermore, we obtain a better approximation guarantee for SSP, improving slightly upon the algorithm by Mucha [20].

Theorem 1.2. *The Shortest Superstring problem can be approximated within a factor of* $(37+\sqrt{57})/18 \approx 2.475$.

Finally, our techniques also imply better approximation guarantees for TGREEDY and MGREEDY; see Section 3.2.

2 DEFINITIONS

We start by reviewing useful notation and concepts from previous works [3, 4, 13] that are necessary to explain our contribution in more detail.

By $S = \{s_1, \ldots, s_m\}$ we denote the input consisting of $m \geq 2$ finite strings. Without loss of generality (w.l.o.g.), we assume that no string in S is a substring of another string in S. This is because the addition of any substring of a string in S to the input cannot modify the superstring that any algorithm considered here outputs.

By $|s|$ we denote the length (i.e., number of characters) of a string s. By $s[i,j]$ we denote the substring of s starting at its i-th character and ending at its j-th character, where $j \in [i, |s|]$. For any two strings s and t, st will denote their concatenation.

Overlaps and distances. By $\text{ov}(s,t)$ we denote the longest (maximum) overlap to merge a string s with a different string t, i.e., $\text{ov}(s,t) = s[|s|-i+1, |s|]$, where i is the largest integer for which $s[|s|-i+1, |s|] = t[1,i]$ holds. For instance, for s='bababa' and t='ababab', we have $\text{ov}(s,t)$='ababa'. By $\text{ov}(s,s)$ we denote the longest self-overlap of string s which has length smaller than $|s|$; for instance, $\text{ov}(s,s)$='baba' for s='bababa'.

By $\text{pref}(s,t)$ we denote the prefix of maximally merging string s with string t, i.e., assuming that $s = uv$ and $t = vz$ for strings u, $v = \text{ov}(s,t)$, and z, it holds that $\text{pref}(s,t) = u$. In the same way, we define $\text{pref}(s,s)$ so that $s = \text{pref}(s,s)\text{ov}(s,s)$. The *distance* $\text{dist}(s,t) = |\text{pref}(s,t)|$ is the number of characters of the prefix; possibly $\text{dist}(s,t) \neq \text{dist}(t,s)$.

Distance and overlap graphs. The *distance graph* is a complete directed graph with self-loops, written as $G_{\text{dist}}(S) = (V, E, \text{dist}(,))$, where $|V| = m$, $|E| = m^2$. Each node corresponds to a string in S and the weight of a directed edge (s,t) equals $\text{dist}(s,t)$, the distance to merge string s with the (not necessarily distinct) string t. Note that the edge lengths satisfy the triangle inequality $\text{dist}(s,t) \leq \text{dist}(s,t') + \text{dist}(t',t)$ as one always obtains the longest overlap by directly merging s to t.

Similarly, the *overlap graph* $G_{\text{ov}}(S)$ is a complete directed graph $(V, E, |\text{ov}(,)|)$ with self-loops, where $|V| = m$, $|E| = m^2$ and the profit of each directed edge (s,t) equals $|\text{ov}(s,t)|$, i.e., the longest overlap to merge string s with the (not necessarily distinct) string t. We will also write $\text{ov}(s,t)$ as $\text{ov}(e)$, where $e = (s,t)$ is a directed edge of the overlap graph.

We can identify an edge $e = (s,t)$ in G_{dist} or G_{ov} with the new string $\text{pref}(s,t)t$ which we obtain by merging s and t. Repeating this argument, we see that a simple directed path $s_0 \to s_1 \to \cdots \to s_k$ corresponds to a new string $\text{pref}(s_0,s_1)\ldots\text{pref}(s_{k-1},s_k)s_k$ which contains all strings represented by nodes on the path as substrings in the same order. Accordingly, a superstring of S simply corresponds to a directed Hamiltonian path in the graph. If two strings s and t appear in adjacent positions and in this order (i.e., s precedes t) across a superstring, we say that s and t are *merged* in the superstring.

Cycle Covers. A cycle cover in a complete directed weighted graph G with self-loops is a set of directed cycles such that the inner degree and the outer degree of each node of G are both unit. An x-cycle, where $x \in [1,m]$, is a directed cycle consisting of x nodes. If s and t are in the same cycle of a cycle cover containing edge (s,t), we say that s and t are *merged* in the cycle cover.

By w we denote the minimum length of a cycle cover in $G_{\text{dist}}(S)$, i.e., w is the minimum sum of distances of edges in a cycle cover in $G_{\text{dist}}(S)$. A minimum-length cycle cover in $G_{\text{dist}}(S)$ is a maximum overlap cycle cover in $G_{\text{ov}}(S)$, since for any edge (s,t), it holds that $|\text{ov}(s,t)| = |s| - \text{dist}(s,t)$. Note that we may have more than one cycle cover with the same length w; to see that, consider the input $S = \{ab^k, b^{k+1}, b^k a\}$, for which the 3-cycle consisting of strings $ab^k, b^{k+1}, b^k a$ has length $k+2$, which equals the length of the 2-cycle for strings $ab^k, b^k a$ plus the length of the 1-cycle for string b^{k+1}.

A maximum overlap cycle cover in $G_{\text{ov}}(S)$ is computed efficiently in the second step of the MGREEDY algorithm of Blum et al. [3, Theorem 10]. In a nutshell, MGREEDY computes an optimal cycle cover by sorting the edges of the overlap graph non-increasingly by their overlap lengths (breaking ties arbitrarily), and adding an edge (s,t) to the cycle cover if and only if no edge (s,t') or (s',t) has been chosen before (s,t). Fixing some arbitrary tie-breaking, we denote the resulting maximum overlap cycle cover by $\text{CC}(S)$. For any cycle c of $\text{CC}(S)$, the last edge of c added by MGREEDY to the solution is called the *cycle-closing edge*. We will frequently use the fact that the overlap length of every edge in a cycle c is at least as large as the overlap length of the cycle-closing edge of c. The sum of overlap lengths of all cycle-closing edges of $\text{CC}(S)$ will be denoted by o.

By $|\text{ALG}(S)|$ we denote the length of a superstring $\text{ALG}(S)$ produced by an algorithm ALG for input S. We use $n = |\text{OPT}(S)|$, where OPT is an optimal Shortest Superstring algorithm. Since merging the last string of a superstring with the first string of this superstring gives a cycle cover in the distance graph (namely, a Hamiltonian cycle), it follows that $w \leq n$.

Representative strings. By $s_{c_0} \to s_{c_1} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$ we denote the cycle $c \in \text{CC}(S)$ consisting of $r \geq 1$ strings, where the last edge $s_{c_{r-1}} \to s_{c_0}$ always denotes the cycle-closing edge. By R_c we denote the string $\text{pref}(s_{c_0}, s_{c_1})\text{pref}(s_{c_1}, s_{c_2})\ldots\text{pref}(s_{c_{r-2}}, s_{c_{r-1}})s_{c_{r-1}}$, i.e., the string obtained by opening the cycle-closing edge $s_{c_{r-1}} \to s_{c_0}$ of cycle c. String R_c will be called the *representative* string of cycle c; note that R_c contains all strings of c as substrings. As $\mathcal{R}$

we denote the set of all representative strings. It follows that a superstring of the strings in $\mathcal{R}$ is, also, a superstring of the strings in S.

3 OUR CONTRIBUTION

Our technical result is the following upper bound on o, the total overlap length of cycle-closing edges, in terms of the shortest superstring length n and w, the total length of all cycles of the minimum-length cycle cover $\mathrm{CC}(S)$:

$$o \leq n + \beta \cdot w \qquad \text{for } \beta = \frac{1+\sqrt{57}}{6} \approx 1.425\,. \tag{1}$$

This improves upon similar bounds on o in [3, 13], which we outline below. In the following two subsections, we explain how this inequality implies Theorems 1.1 and 1.2. The remaining part of the paper is devoted to proving (1).

3.1 Improved Approximation Guarantee of GREEDY

Assuming that all $|E| = m^2$ edges of $G_{\mathrm{ov}}(S)$ are ordered by non-increasing overlap, breaking ties arbitrarily, GREEDY works by going down this list and picking edge e if:

- e does not share a head or tail with an edge e' that GREEDY picked in a previous step (such e' precedes e in the ordered list of edges) and
- e is not a cycle-closing edge.

Otherwise, GREEDY moves to the next edge in the order. Clearly, GREEDY outputs a directed path of $m-1$ edges which gives a superstring by merging adjacent strings. Note that the computation of $\mathrm{CC}(S)$ by MGREEDY only differs from GREEDY by not using the second condition.

Blum et al. [3] call the edges rejected by GREEDY for not satisfying the second condition (but satisfying the first condition) *bad back edges*. The reason that they are called "back edges" is that one can number the input strings $S = \{s_1, \ldots, s_m\}$ so that the superstring GREEDY(S) contains the strings in the same order, i.e., s_i appears before s_j in GREEDY(S) if and only if $i < j$. In this subsection, we assume that the input strings are numbered in this way.

We say that a bad back edge e *spans interval* $[i, j]$ (for $i \leq j$) if $e = (s_j, s_i)$. Blum et al. show that the intervals spanned by two bad back edges are either disjoint or one is contained in the other, i.e., these intervals form a laminar family [3, Lemma 13]. A *culprit* is a bad back edge e such that the interval spanned by e is minimal in this laminar family (i.e., there is no bad back edge e' such that the interval spanned by e' is properly contained in the interval spanned by e). See Figure 1 for an illustration. A cycle is called *culprit* if its cycle-closing edge is a culprit.

Let w_c denote the sum of the lengths of culprit cycles and let o_c be the sum of overlap lengths of culprit edges. Blum et al. showed the following two inequalities (paragraph after the proof of Lemma 17 in [3]):

$$|\mathrm{GREEDY}(S)| \leq 2n + o_c - w_c \tag{2}$$

$$o_c \leq n + 2w_c \tag{3}$$

Plugging (3) into (2), we have $|\mathrm{GREEDY}(S)| \leq 2n + o_c - w_c \leq 3n + w_c \leq 4n$, since $w_c \leq w \leq n$. By using the Overlap Rotation Lemma of [4], Kaplan and Shafrir [13] improved (3) to $o_c \leq n + 1.5w_c$ and, hence, the upper bound on the approximation ratio of GREEDY to 3.5 since $|\mathrm{GREEDY}(S)| \leq 2n + o_c - w_c \leq 3n + 0.5 \cdot w_c \leq 3.5n$.

Let $S_c \subseteq S$ be the set of input strings which lie on culprit cycles. Blum et al. show that the application of MGREEDY on S_c outputs exactly the culprit cycles [3, Lemma 15] (see also Observation 5.1). Therefore, our technical result in (1) applied to input S_c implies $o_c \leq n_c + \beta \cdot w_c$ where $n_c \leq n$ equals the length of the shortest superstring for S_c. Plugging this into (2), we have: $|\mathrm{GREEDY}(S)| \leq 2n + n_c + (\beta - 1) \cdot w_c \leq 3n + (\beta - 1) \cdot w_c \leq (2 + \beta) \cdot n \approx 3.425n$.

3.2 Improved Approximation Guarantee for SSP

As discussed before, the algorithm MGREEDY computes $\mathrm{CC}(S)$ or, more specifically, the set of representative strings $\mathcal{R}$ for all cycles. It then outputs the superstring that is obtained by concatenating all representative strings in an arbitrary order. The total length of the representative strings is $w + o$, i.e., the minimum length of a cycle cover in $G_{\mathrm{dist}}(S)$ plus the sum of overlaps of all cycle-closing edges of the cycle cover. Our main result in (1) states that $o \leq n + \beta \cdot w$. Therefore, the superstring computed by MGREEDY has length $w + o \leq n + (1 + \beta) \cdot w \leq (2 + \beta) \cdot n$. Hence, just as for GREEDY, we get that MGREEDY is a $(2 + \beta)$-approximation algorithm, which improves upon the upper bound of 3.5 implied in [13].

Instead of just concatenating the representative strings, we can also attempt to overlap them, i.e., to compute a shorter superstring of the representative strings. One possibility is to use an approximation algorithm for Maximum Asymmetric TSP (MaxATSP) for this in order to find a superstring that aims to maximize the total overlap between the representative strings.

The following theorem is adopted from the literature [4, 19, 20] (for this particular version we are following [19, Theorem 21]) and, combined with our new result for MGREEDY, results in an improved approximation guarantee for SSP. A proof is included in the appendix of the full version of the paper.

Theorem 3.1. *If MGREEDY is a $(2+\beta)$-approximation algorithm and there exists a δ-approximation algorithm for MaxATSP (for $\delta \leq 1$), then there exists a $(2 + (1 - \delta) \cdot \beta)$-approximation algorithm for SSP.*

Using the $\frac{2}{3}$-approximation algorithm for MaxATSP of [12] or the more recent and simpler $\frac{2}{3}$-approximation algorithm of [23], Theorem 3.1 with $\delta = \frac{2}{3}$ implies that we get an approximation guarantee of $\frac{37+\sqrt{57}}{18} \approx 2.475$. This improves slightly upon the approximation guarantee of $2\frac{11}{23} \approx 2.478$ of the currently best SSP algorithm [20]. The use of a better than $\frac{2}{3}$-approximation algorithm for MaxATSP as a black-box will give an even smaller approximation guarantee for SSP. (For example, there is a recent preprint by Paluch [22] claiming a $\frac{7}{10}$-approximation for MaxATSP. Combining our result with a $\frac{7}{10}$-approximation for MaxATSP would result in a ≈ 2.427-approximation for SSP.)

TGREEDY. The TGREEDY algorithm of Blum et al. works by first computing the representative strings $\mathcal{R}$ and then, rather than using a possibly complicated approximation algorithm for MaxATSP, applying GREEDY to this set of representative strings. As GREEDY

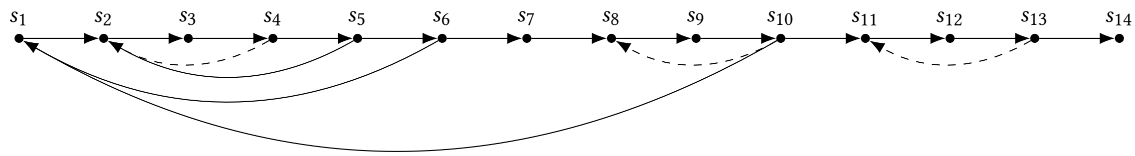

Figure 1: Illustration of *culprits*. The superstring returned by GREEDY merges the strings s_1 to s_{14} in this order as indicated by the path (however, the order in which GREEDY picks edges (s_{i-1}, s_i) is different). The bend edges are the *bad back edges*. Out of the bad back edges, the dashed edges are the *culprits*.

gives a $\frac{1}{2}$-approximation for such instances of MaxATSP [25, 27] (more precisely, for the longest Hamiltonian path, which is sufficient), using $\delta = \frac{1}{2}$ in Theorem 3.1, we get that TGREEDY is a $\frac{25+\sqrt{57}}{12} \approx 2.7125$-approximation algorithm, which improves upon the upper bound of 2.75 (implied in [4, 19]).

4 THE BIG PICTURE

Small, large, and extra large cycles. Our key idea is to partition cycles into a few types according to the ratio between their length and the overlap length of their cycle-closing edge, and treat these types differently in the analysis. To this end, let $w(c)$ denote the length of a cycle c of $\mathrm{CC}(S)$, and let $o(c)$ denote the overlap length of the cycle-closing edge of c, i.e., $o(c) = |\mathrm{ov}(s_{c_{r-1}}, s_{c_0})|$, where $(s_{c_{r-1}}, s_{c_0})$ is the cycle-closing edge. A cycle c of $\mathrm{CC}(S)$ is

- a *small* cycle if $o(c) > 2w(c)$,
- a *large* cycle if $\beta \cdot w(c) < o(c) \leq 2w(c)$, and
- an *extra large* cycle if $o(c) \leq \beta \cdot w(c)$,

where β is the parameter defined in (1). The set of extra large cycles of $\mathrm{CC}(S)$ will be denoted by $\mathcal{X}(S)$, the set of large cycles of $\mathrm{CC}(S)$ will be denoted by $\mathcal{L}(S)$, and the set of small cycles of $\mathrm{CC}(S)$ will be denoted by $\mathcal{S}(S)$.

In Section 5.1, we show that we can assume w.l.o.g. that $\mathrm{CC}(S)$ contains no extra large cycle. For this, we exploit the slack in the right-hand side of $o(c) \leq \beta \cdot w(c)$ for an extra large cycle c, compared to the right-hand side of $o \leq n + \beta \cdot w$ that we want to show.

Outline. To get our technical result in (1), we prove two independent upper bounds on o. In Section 6, we improve $o \leq n + 1.5w = n + 1.5 \cdot \sum_{c \in \mathcal{S}(S)} w(c) + 1.5 \cdot \sum_{c \in \mathcal{L}(S)} w(c)$ of [13] to

$$o \leq n + \sum_{c \in \mathcal{S}(S)} w(c) + \frac{3}{2} \cdot \sum_{c \in \mathcal{L}(S)} w(c)\,. \tag{4}$$

On its own, the improvement by $\frac{1}{2} \cdot \sum_{c \in \mathcal{S}(S)} w(c)$ over [13] is insignificant because the total length of the small cycles may be very small compared to the total length of the large cycles. However, we show a different upper bound which is better when small cycles contribute only very little to w. Namely, in Section 7, we prove that

$$o \leq n + \gamma \cdot \sum_{c \in \mathcal{S}(S)} w(c) + \sum_{c \in \mathcal{L}(S)} w(c) \tag{5}$$

for a positive constant γ, and this is sufficient to obtain $o \leq n + (1.5 - \epsilon) \cdot w$ for a positive constant ϵ, when combined with the first upper bound on o. Naturally, the smaller γ we get, the smaller the resulting upper bound. We will require that γ and the aforementioned parameter β satisfy the following four constraints:

$$(3 - 2\beta) \cdot \gamma = 2 - \beta \tag{6}$$

$$3 \cdot \left(\beta - \frac{2}{\gamma - 2}\right) \geq 1 \tag{7}$$

$$\frac{5}{2} + \frac{1}{2(\beta - 1)} \leq \gamma \tag{8}$$

$$\gamma \leq (\gamma - 1) \cdot \beta \tag{9}$$

Solving this system of inequalities, while minimizing β, yields

$$\beta = \frac{1 + \sqrt{57}}{6} \approx 1.425 \qquad \text{and} \qquad \gamma = \frac{31 + 3\sqrt{57}}{14} \approx 3.832\,.$$

Note that (8) and (9) are not tight, i.e., β and γ are determined by (6) and (7).

Multiplying (4) by $(2\beta - 2)$ and (5) by $(3 - 2\beta)$ and adding the two resulting inequalities we get

$$\begin{aligned} o &\leq n + \big((2\beta - 2) + (3 - 2\beta) \cdot \gamma\big) \cdot \sum_{c \in \mathcal{S}(S)} w(c) \\ &\quad + \big((3\beta - 3) + (3 - 2\beta)\big) \cdot \sum_{c \in \mathcal{L}(S)} w(c) \\ &= n + \beta \sum_{c \in \mathcal{S}(S)} w(c) \; + \; \beta \sum_{c \in \mathcal{L}(S)} w(c) = n + \beta \cdot w\,, \end{aligned}$$

where we use (6) in the second step. This shows (1), as desired.

Intuition. Before we start with formal proofs, we give some intuition and explain the main ideas behind our technical contribution. First, we observe in Section 5.1 that we can assume that there are no extra large cycles (as they can be handled separately), which will be useful in the proof of the second upper bound. Note that if all (remaining) cycles are large, then our proof is complete as summing over all cycles gives $o \leq 2 \cdot w \leq n + w$. On the other hand, if all (remaining) cycles are small, the first bound (4) gives $o \leq n + w$, again implying a better bound than in (1). This means that it is the presence of both small and large cycles that makes the analysis challenging.

To facilitate the analysis of small cycles, we show in Section 5.3 that we can make the following assumption: If an optimal superstring merges two strings from one small cycle c, then these two strings must be merged in the small cycle c as well. This essentially follows from the large amount of overlap length (relatively to $w(c)$) in small cycles.

We obtain the first bound by proving a lower bound on n, the optimal superstring length. Roughly speaking, we show that each small cycle c must contribute at least $o(c) - w(c)$ to n, for which we use that strings of small cycles must be relatively long (longer than $o(c) > 2w(c)$) together with a bound from [3] on the overlap between two strings from different cycles. For a large cycle, we use a generalization of the Overlap Rotation Lemma from [4] to carefully pick a single string from this cycle that is suitable for obtaining the lower bound on n.

It is the second upper bound that constitutes our main technical contribution. Recall that w, the length of the optimal cycle cover CC, is a lower bound on the length of the shortest Hamiltonian cycle CC_0 in G_{dist}, which is itself a lower bound on n. In proving the second upper bound, we make use of the difference between w and the length of CC_0 and therefore, we derive a stronger lower bound on n. Namely, we construct a careful sequence of edge swaps transforming CC_0 into CC such that each step decreases the length of the current cycle cover by at least a certain suitable amount. In a nutshell, when an edge swap in the constructed sequence results in adding a small cycle $c \in \mathrm{CC}$ to the current cycle cover, we show that this must decrease the length of the cycle cover by at least $o(c) - \gamma \cdot w(c)$ minus a term for certain large cycles affected by the swap. Summing up over all steps will give us the desired lower bound on the length of CC_0.

Outline. Before proving the two bounds using the ideas outlined above, we review useful lemmas from previous work in Section 5.2 and derive several properties of strings belonging to small cycles in Section 5.3. We remark that Sections 6 and 7 are independent of each other and can be read in any order.

5 PRELIMINARIES FOR THE ANALYSIS

We start by observing that MGREEDY executed on the strings belonging to a subset of cycles of the minimum cycle cover $\mathrm{CC}(S)$ produces exactly the same subset of cycles.

Observation 5.1. *Let $\overline{CC} \subseteq CC(S)$ be a set of cycles and let $\overline{S} \subseteq S$ be the set of input strings that belong to cycles in $\overline{CC}$. Then MGREEDY on input $\overline{S}$ (with the same tie-breaking rule) outputs $\overline{CC}$, which is thus the minimum-length cycle cover of $\overline{S}$, i.e., $CC(\overline{S}) = \overline{CC}$.*

Proof. Note that MGREEDY on input S rejects any edge (s,t) between $\overline{S}$ and $S \setminus \overline{S}$ because there is an incident edge (s',t) or (s,t') with larger (or equal) overlap that precedes (s,t) in the list of edges sorted by their overlap length. Thus, when we run MGREEDY on input $\overline{S}$, it selects exactly the same edges among vertices in $\overline{S}$ as when we run MGREEDY on input S. □

5.1 Dealing with Extra Large Cycles

Let $\overline{S} \subseteq S$ be the subset of strings that belong to all small and large cycles of $\mathrm{CC}(S)$. Observation 5.1 implies that $\mathrm{CC}(\overline{S})$ consists of all small and large cycles of $\mathrm{CC}(S)$, while $\mathrm{CC}(S-\overline{S})$ consists of all extra large cycles of $\mathrm{CC}(S)$. Let $\hat{w}$ denote the sum of lengths of the (extra large) cycles in $\mathrm{CC}(S-\overline{S})$ and let $\hat{o}$ be the sum of overlap lengths of the cycle-closing edges of the cycles in $\mathrm{CC}(S-\overline{S})$. Similarly, let $\overline{o}$ be the sum of overlap lengths of the cycle-closing edges in $\mathrm{CC}(\overline{S})$ and let $\overline{w}$ be the sum of lengths of the cycles in $\mathrm{CC}(\overline{S})$. Proving (1) for input $\overline{S}$ implies that $\overline{o} \leq |\mathrm{OPT}(\overline{S})| + \beta \cdot \overline{w}$, and assuming this, we show $o \leq n + \beta \cdot w$. Indeed, we take the sum of inequality $\overline{o} \leq |\mathrm{OPT}(\overline{S})| + \beta \cdot \overline{w}$ with inequality $\hat{o} \leq \beta \cdot \hat{w}$ (which holds by the definition of extra large cycles) and obtain:

$$o = \overline{o} + \hat{o} \leq |\mathrm{OPT}(\overline{S})| + \beta \cdot \overline{w} + \beta \cdot \hat{w} = |\mathrm{OPT}(\overline{S})| + \beta \cdot w \leq n + \beta \cdot w$$

where the penultimate step uses $w = \overline{w} + \hat{w}$ and the last inequality uses $|\mathrm{OPT}(\overline{S})| \leq |\mathrm{OPT}(S)| = n$, which follows from $\overline{S} \subseteq S$. Therefore, for proving (1), we assume w.l.o.g. that $\mathrm{CC}(S)$ has no extra large cycle.

5.2 Useful Lemmas from Previous Work

We start with describing further concepts from the literature. A *semi-infinite* string is defined as the concatenation of an infinite number of finite non-empty strings. If these strings are the same string x, then the semi-infinite string will be denoted by x^∞ and called *periodic*. For a semi-infinite string α and integer $k \geq 1$, we denote by $\alpha[k]$ its (semi-infinite) substring which starts at its k-th character.

We say that a string s has a *periodicity* of length a for $a \leq |s|$ if s is a prefix of x^∞ for some string x of length a. Note that $\mathrm{pref}(s,s)$ is the shortest string x such that s is a prefix of x^∞. The length of $\mathrm{pref}(s,s)$ is denoted as $\mathrm{period}(s) = |\mathrm{pref}(s,s)| = \mathrm{dist}(s,s)$. In other words, $\mathrm{period}(s)$ is the smallest periodicity of a string. We will need the following property of periodicity; see e.g. [10, Theorem 16.17.1].

Lemma 5.2. *Any string s with periodicities a and b such that $|s| \geq a+b$ has periodicity $\gcd(a,b)$, where $\gcd(a,b)$ is the greatest common divisor of a and b. Consequently, any periodicity a of s with $a \leq |s|/2$ (if any) is an integer multiple of period(s).*

String z is a *rotation* of string q if $q = uv$ and $z = vu$ for some strings v and u (string z is a rotation of itself if one of them is empty). Two strings s and t are *equivalent* if $\mathrm{pref}(t,t)$ is a rotation of $\mathrm{pref}(s,s)$, i.e., there exist strings x and y (possibly empty) such that $\mathrm{pref}(s,s) = xy$ and $\mathrm{pref}(t,t) = yx$. Two strings that are not equivalent will be called *inequivalent*.

For any cycle $c = s_{c_0} \to s_{c_1} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$ in $G_{\text{dist}}(S)$, we define $s(c)$ as the string $\mathrm{pref}(s_{c_0}, s_{c_1})\mathrm{pref}(s_{c_1}, s_{c_2}) \ldots \mathrm{pref}(s_{c_{r-1}}, s_{c_0})$, which has length $w(c)$. Observe that $R_c = s(c)\mathrm{ov}(s_{c_{r-1}}, s_{c_0})$ and that R_c is a prefix of $s(c)^\infty$. We define as $\mathrm{strings}(c, s_{c_l})$ the string $\mathrm{pref}(s_{c_l}, s_{c_{l+1}}) \ldots \mathrm{pref}(s_{c_{l-1}}, s_{c_l})$, where subscript arithmetic is modulo r and $0 \leq l \leq r-1$. In other words, $\mathrm{strings}(c, s_{c_l})$ is a rotation of $s(c)$ such that string s_{c_l} is a prefix of $\mathrm{strings}(c, s_{c_l})^\infty$.

The following three lemmas appear in previous works:

Lemma 5.3 (Claim 2 in [3]). *For any cycle c in the distance graph for S, every string of c is a substring of $s(c)^\infty$.*

Lemma 5.4 (Claim 3 in [3]). *If all strings of a subset of S are substrings of a semi-infinite string t^∞, then there exists a cycle of length $|t|$ in the distance graph $G_{dist}(S)$ that contains all these strings.*

Lemma 5.5 (Lemma 13 in [19]). *It holds that period$(R_c) = w(c)$ for any cycle c of $CC(S)$.*

As a corollary of these lemmas, we obtain:

Observation 5.6. *The representative strings R_c and $R_{c'}$ for any two cycles c and c' in $CC(S)$ are inequivalent. Moreover, any string*

$\hat{R}_{c'}$ that contains all strings of cycle c' as substrings is inequivalent to $s(c)^\infty$.

Proof. Recall that R_c is a prefix of $s(c)^\infty$, which contains all strings of c by Lemma 5.3. Lemma 5.5 implies that $\mathrm{pref}(R_c, R_c) = s(c)$ and $\mathrm{pref}(R_{c'}, R_{c'}) = s(c')$. If R_c and $R_{c'}$ were equivalent, then $s(c')$ is a rotation of $s(c)$ and thus, any string of both cycles appears as a substring of $s(c)^\infty$. Therefore, by Lemma 5.4, all strings of both c and c' are contained in a single cycle of length $w(c)$, contradicting the minimality of $\mathrm{CC}(S)$.

The second claim follows similarly. If $\mathrm{pref}(\hat{R}_{c'}, \hat{R}_{c'})$ is a rotation of $f := \mathrm{pref}(s(c)^\infty, s(c)^\infty)$, then $f^\infty = s(c)^\infty$ contains all strings of both c and c', so we again obtain a contradiction with the minimality of $\mathrm{CC}(S)$ by using Lemma 5.4. □

Since the representative string R_c contains any string s of the cycle c it belongs to, the period of s cannot be larger than $\mathrm{period}(R_c)$ and thus, by Lemma 5.5, we obtain:

Observation 5.7. *For any string s of a cycle $c \in CC(S)$, it holds that $period(s) \leq w(c)$.*

Next, we need the following upper bound for the overlap length between inequivalent strings:

Lemma 5.8 (Lemma 2.3 in [13]). *For any two inequivalent strings s and t, it holds that $|ov(s,t)| < period(s) + period(t)$.*

In the case that these two inequivalent strings belong to two different cycles c and c' of $\mathrm{CC}(S)$, we have $|\mathrm{ov}(s,t)| < w(c) + w(c')$ by Observation 5.7, and more generally:

Lemma 5.9 (Lemma 9 in [3]). *Let c and c' be any two cycles of $CC(S)$. It holds that $|ov(s,t)| < w(c) + w(c')$, where s is any string of c and t is any string of c'.*

We will need an even more general corollary that follows from the same argument as in Lemma 9 in [3] (see also Lemma 7 in [19]), but we provide a proof for completeness.

Corollary 5.10. *Let c and c' be any two cycles of $CC(S)$. Any string h, which is a substring of both $s(c)^\infty$ and $s(c')^\infty$, satisfies $|h| < w(c) + w(c')$. In particular, it holds that $|ov(s,t)| < w(c) + w(c')$, where s is any substring of $s(c)^\infty$ and t is any substring of $s(c')^\infty$.*

Proof. Assume for a contradiction that $|h| \geq w(c) + w(c')$. Since h is a substring of $s(c)^\infty$, it is a prefix of x_1^∞ for a string x_1 with $|x_1| = w(c)$, which is a rotation of $s(c)$. Similarly, h is a prefix of x_2^∞ for x_2 with $|x_2| = w(c')$, which is a rotation of $s(c')$. Using $|h| \geq w(c) + w(c')$, we get that $x_1x_2 = x_2x_1$ and by a simple induction, it holds that $x_1^k x_2^k = x_2^k x_1^k$ for any $k \geq 1$, which implies $x_1^\infty = x_2^\infty$. Since any string in cycle c is a substring of $s(c)^\infty$, it is also a substring of $x_1^\infty = x_2^\infty$, and similarly for c'. Thus, using Lemma 5.4 gives a contradiction with the fact that c and c' are two cycles of the minimum-length cycle cover $\mathrm{CC}(S)$. □

5.3 Properties of Strings of Small Cycles

In this section, we prove several properties of small cycles. Consider a small cycle c. Recall that the MGREEDY algorithm picks edges in non-increasing order of overlap length when producing $\mathrm{CC}(S)$. Therefore, $o(c)$ is no larger than any other overlap length between two merged strings in cycle c. By this and since the length of any string s in c is greater than the length of any of its two (i.e., left and right) overlaps (or the self-overlap if c is a 1-cycle), we have $|s| > o(c)$. Further, by the definition of a small cycle, it is $o(c) > 2 \cdot w(c)$ and thus, for any string s of c, we get:

$$|s| > 2 \cdot w(c) \tag{10}$$

Note that the representative string R_c is even longer as $|R_c| = w(c) + o(c) > 3 \cdot w(c)$, since string R_c is formed by opening cycle c at the cycle-closing edge.

While a string of a cycle c is not necessarily equivalent to string R_c (cf. Lemma 2.1 in [13]), we prove that this property actually holds for small cycles.

Lemma 5.11. *Consider any small cycle c of $CC(S)$. All strings of c and R_c are equivalent and in particular, $period(s) = w(c)$ for any string s of cycle c.*

Proof. Recall that R_c is a prefix of $s(c)^\infty$. From Lemma 5.5 it follows that $\mathrm{pref}(R_c, R_c) = s(c)$. Hence, it suffices to show that $\mathrm{pref}(s,s)$ is a rotation of $s(c)$ for any string s of the small cycle c. We first prove that $\mathrm{period}(s) = w(c)$. By Observation 5.7, we have $\mathrm{period}(s) \leq w(c)$. Assume for a contradiction that $\mathrm{period}(s) < w(c)$. Since s has periodicity $w(c)$ and, by (10), $|s| > 2w(c)$, we have that $w(c)$ must be a multiple of $\mathrm{period}(s)$ by Lemma 5.2. So there exists an integer $k \geq 2$ such that $k \cdot |\mathrm{pref}(s,s)| = k \cdot \mathrm{period}(s) = w(c)$. Recall that $\mathrm{strings}(c,s)$ is a rotation of $s(c)$ that is a prefix of s and has length $w(c)$. We thus have that $\mathrm{strings}(c,s) = \mathrm{pref}(s,s)^k$, which implies $\mathrm{strings}(c,s)^\infty = \mathrm{pref}(s,s)^\infty$. Note that every substring of $s(c)^\infty$ is also a substring of $\mathrm{strings}(c,s)^\infty = \mathrm{pref}(s,s)^\infty$. By Lemmas 5.3 and 5.4, it follows that all strings of c belong to a cycle (in $G_{\mathrm{dist}}(S)$) of length $|\mathrm{pref}(s,s)| = \mathrm{period}(s) < w(c)$, which contradicts the minimality of $\mathrm{CC}(S)$. Hence, $\mathrm{period}(s) = w(c)$ and thus, $\mathrm{pref}(s,s) = \mathrm{strings}(c,s)$. This concludes the proof as $\mathrm{strings}(c,s)$ is a rotation of $s(c) = \mathrm{pref}(R_c, R_c)$. □

As a corollary, we obtain that for small cycles, the triangle inequality in $G_{\mathrm{dist}}(S)$ becomes equality.

Lemma 5.12. *Consider two strings $s \in S$ and $t \in S$ both belonging to a small cycle $c \in CC(S)$ and assume that s is not merged with t across cycle c. Then, for any string t' that lies on cycle c between s and t (in this order), it holds that $dist(s,t) = dist(s,t') + dist(t',t)$.*

Proof. First, it is $\mathrm{dist}(s,t) \leq \mathrm{dist}(s,t') + \mathrm{dist}(t',t)$ by the triangle inequality in $G_{\mathrm{dist}}(S)$. Next, assume for a contradiction that $\mathrm{dist}(s,t) < \mathrm{dist}(s,t') + \mathrm{dist}(t',t)$. Consider the semi-infinite string $R' = \mathrm{pref}(s,t)\mathrm{strings}(c,t)^\infty$. Let $t_0 = t, t_1, \ldots, t_\ell = s$ be the strings on the directed path from t to s on cycle c. Observe that s is a prefix of R' (as t is a prefix of $\mathrm{strings}(c,t)^\infty$) and a substring of $\mathrm{strings}(c,t)^\infty$, starting at position $\sum_{j=0}^{\ell-1} \mathrm{dist}(t_j, t_{j+1})$. It follows that

$$\begin{aligned}\mathrm{dist}(s,s) &\leq \mathrm{dist}(s,t) + \sum_{j=0}^{\ell-1} \mathrm{dist}(t_j, t_{j+1}) \\ &< \mathrm{dist}(s,t') + \mathrm{dist}(t',t) + \sum_{j=0}^{\ell-1} \mathrm{dist}(t_j, t_{j+1}) \leq w(c)\,,\end{aligned}$$

where the penultimate inequality holds due to our assumption that $\mathrm{dist}(s,t) < \mathrm{dist}(s,t') + \mathrm{dist}(t',t)$ and the last inequality follows by using the triangle inequality in $G_{\text{dist}}(S)$ for the edges between s and t' and for those between t' and t. Thus, we have that $\text{period}(s) = \text{dist}(s, s) < w(c)$, which contradicts Lemma 5.11. ◻

Lemma 5.12 implies the following useful property:

Observation 5.13. *If two strings that belong to the same small cycle $c \in CC(S)$ are* not *merged in c, then there is an optimal superstring in which they are* not *merged.*

Proof. Suppose that strings s, t belonging to $c \in \text{CC}(S)$ are not merged in c, and let $t_1, \ldots, t_\ell$ (for $\ell \geq 1$) be the strings on the directed s-t-path in c. Let σ be any superstring in which s and t are merged. Consider string $\hat{\sigma}$ obtained by removing strings $t_1, \ldots, t_\ell$ from σ, which may only decrease its length, i.e., $|\hat{\sigma}| \leq |\sigma|$. From $\hat{\sigma}$, we create a superstring σ' by inserting strings $t_1, \ldots, t_\ell$ between s and t in $\hat{\sigma}$. Crucially, by Lemma 5.12, it holds that $|\sigma'| = |\hat{\sigma}| \leq |\sigma|$. Thus, if σ is optimal, then σ' is also optimal. ◻

By Observation 5.13, we obtain the following remark:

Remark 5.14. *If a superstring σ merges all r strings belonging to the same small cycle $c = s_{c_0} \to s_{c_1} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$ (i.e., they all appear in adjacent positions across the superstring σ), then we can transform σ into a superstring σ' with $|\sigma'| \leq |\sigma|$ where the order of these strings across σ' is a rotation of the ordered set $\{s_{c_0}, s_{c_1}, \ldots, s_{c_{r-1}}\}$. In this case, each of the r edges of $c \in CC(S)$ coincides with an edge of σ' except for one edge, which is not necessarily the cycle-closing edge $s_{c_{r-1}} \to s_{c_0}$ of c.*

6 THE FIRST UPPER BOUND

In this section, we prove (4), which is our first bound on o.

We consider a partition of strings of all small cycles such that no two strings from two different cycles are in one part and moreover, due to Observation 5.13, if strings s and t from a small cycle c are in one part, then all strings between s and t on c are in that part as well. In other words, this partition consists of directed paths and single nodes that remain after removing a subset of edges from small cycles. The particular partition that we consider below is induced by an optimal superstring for a certain subset of the input S containing all strings of small cycles and one (carefully chosen) string of each large cycle.

Consider a small cycle c. Let r' be the number of parts with strings from cycle c, and for $j = 0, \ldots, r'$, denote by $\bar{s}_j$ the string obtained by merging strings in the j-th part (in the same order as they appear on the small cycle c). In the next technical lemma, we lower-bound the sum of lengths of the strings $\bar{s}_j$.

Lemma 6.1. *It holds that $\sum_{j=0}^{r'-1}(|\bar{s}_j| - 2 \cdot w(c)) \geq o(c) - w(c)$ for any small cycle $c = s_{c_0} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$, where $r' \leq r$.*

Proof. Consider string $\bar{s}_j$, and let $t_j^0, t_j^1, \ldots, t_j^{\ell_j-1}$ for $\ell_j \geq 1$ be the strings that are merged into $\bar{s}_j$. Assuming that the parts are numbered in the order in which they appear on the cycle, t_{j+1}^0 is the string to which $t_j^{\ell_j-1}$ is merged on cycle c, with the subscript arithmetic modulo r'. (In the special case of a 1-cycle, we have $r' = r = 1$, $\ell_0 = 1$, t_0^0 is the only string of that cycle, and we use $t_1^0 = t_0^0$.) It holds that:

$$
\begin{aligned}
|\bar{s}_j| &= \sum_{k=0}^{\ell_j-2} \text{dist}(t_j^k, t_j^{k+1}) + |t_j^{\ell_j-1}| \\
&= \sum_{k=0}^{\ell_j-2} \text{dist}(t_j^k, t_j^{k+1}) + \text{dist}(t_j^{\ell_j-1}, t_{j+1}^0) + |\text{ov}(t_j^{\ell_j-1}, t_{j+1}^0)|\,,
\end{aligned}
$$

since $|s| = \text{dist}(s, t) + |\text{ov}(s, t)|$ for any two strings s and t. Summing over all r' strings $\bar{s}_j$, we get

$$
\begin{aligned}
\sum_{j=0}^{r'-1}(|\bar{s}_j| - 2w(c)) &= \sum_{j=0}^{r'-1}\Bigg(\sum_{k=0}^{\ell_j-2} \text{dist}(t_j^k, t_j^{k+1}) + \text{dist}(t_j^{\ell_j-1}, t_{j+1}^0) \\
&\qquad + |\text{ov}(t_j^{\ell_j-1}, t_{j+1}^0)| - 2w(c)\Bigg) \\
&= w(c) + \left(|\text{ov}(t_0^{\ell_0-1}, t_1^0)| - 2w(c)\right) \\
&\qquad + \sum_{j=1}^{r'-1}\left(|\text{ov}(t_j^{\ell_j-1}, t_{j+1}^0)| - 2w(c)\right) \\
&\geq w(c) + (o(c) - 2w(c)) + 0 = o(c) - w(c)\,,
\end{aligned}
$$

where the second equality uses that each edge of cycle c either "lies inside a string $\bar{s}_j$", i.e., is an edge (t_j^k, t_j^{k+1}) for some j and $0 \leq k \leq \ell_j - 2$, or "leads from string $\bar{s}_j$ to $\bar{s}_{j+1}$", i.e., is an edge $(t_j^{\ell_j-1}, t_{j+1}^0)$ for some j, and the inequality follows from the fact that $o(c)$ is the smallest overlap on cycle c and that $o(c) > 2w(c)$ as the cycle is small. ◻

We will need the Overlap Rotation Lemma from [4]:

Lemma 6.2 (Lemma 3.3 in [4]). *Let α be a periodic semi-infinite string. There exists an integer $k \in [1, period(\alpha)]$ such that $period(s) + \frac{1}{2}period(\alpha) > |ov(s, \alpha[k])|$ for any (finite) string s inequivalent to α.*

Note that the index k is universal for all strings inequivalent to α. We now generalize Lemma 6.2:

Lemma 6.3. *Let α and k be as in Lemma 6.2. For any $k' \in [0, k)$ and any (finite) string s inequivalent to α, the string $\alpha[k-k']$ satisfies $|ov(s, \alpha[k-k'])| < period(s) + \frac{1}{2}period(\alpha) + k'$.*

Proof. For $k' = 0$ the statement of the lemma coincides with Lemma 6.2. It remains to show the lemma for $k' > 0$. We have

$$
\begin{aligned}
|\text{ov}(s, \alpha[k-k'])| &= |s| - \text{dist}(s, \alpha[k-k']) \\
&\leq |s| - \text{dist}(s, \alpha[k]) + \text{dist}(\alpha[k-k'], \alpha[k]) \\
&= |\text{ov}(s, \alpha[k])| + \text{dist}(\alpha[k-k'], \alpha[k]) \\
&\leq |\text{ov}(s, \alpha[k])| + k' \\
&< \text{period}(s) + \frac{\text{period}(\alpha)}{2} + k'\,,
\end{aligned}
$$

where in the second line, we applied the triangle inequality in $G_{\text{dist}}(S)$ and the last step follows from Lemma 6.2. ◻

In Lemma 6.4, we prove the first upper bound on o, i.e., inequality (4).

Lemma 6.4. *We have $o \leq n + \sum_{c \in \mathcal{S}(S)} w(c) + 1.5 \cdot \sum_{c \in \mathcal{L}(S)} w(c)$.*

Proof. First, for each large cycle c, we apply Lemma 6.2 for the semi-infinite string $\alpha_c = s(c)^\infty$ to get an integer $k_c \geq 1$. We also let k'_c be the smallest integer $k' \geq 0$ such that $\alpha_c[k_c - k']$ starts with a string f_c from cycle c. By the minimality of k'_c, it follows that $k'_c < \mathrm{dist}(f_c, t_c)$, which is the prefix length between f_c and string t_c that f_c is merged to across the large cycle c (in case c is a 1-cycle, we assume that $t_c = f_c$).

If c consists of at least two strings, then since $|f_c| = \mathrm{dist}(f_c, t_c) + |\mathrm{ov}(f_c, t_c)|$ and $|\mathrm{ov}(f_c, t_c)| \geq o(c)$, we get that $k'_c < \mathrm{dist}(f_c, t_c) = |f_c| - |\mathrm{ov}(f_c, t_c)| \leq |f_c| - o(c)$. If c is a 1-cycle consisting of string $f_c = t_c$, we again obtain $k'_c < \mathrm{dist}(f_c, f_c) = |f_c| - |\mathrm{ov}(f_c, f_c)| = |f_c| - o(c)$.

Fix input $S_r \subseteq S$, which contains all strings of S belonging to small cycles and only the single string f_c from each large cycle c. Consider $\mathrm{OPT}(S_r)$, the optimal superstring of S_r, and let $n_r = |\mathrm{OPT}(S_r)|$. Our aim is to derive a lower bound on $n_r \leq n$.

Superstring $\mathrm{OPT}(S_r)$ induces a partition of the strings in each small cycle c such that strings in each part are merged together in $\mathrm{OPT}(S_r)$, while strings from different parts are separated by a string from a different cycle; this is the partition for which we apply Lemma 6.1. By Observation 5.13, we may assume that the order in which strings of the same small cycle c are merged in $\mathrm{OPT}(S_r)$ is the same as the order in which they appear on c. For a small cycle c, let r'_c be the size of this partition of strings in c, and for $j = 0, \ldots, r'_c - 1$, denote by $\bar{s}_{c,j}$ the string obtained by merging strings in the j-th part (in the same order as they appear on c).

The key step towards lower-bounding n_r is to obtain suitable upper bounds on the overlap length of two strings merged in $\mathrm{OPT}(S_r)$ after we merge strings of small cycles c to obtain strings $\bar{s}_{c,j}$. First, consider string f_c of a large cycle c and string s' from a cycle c' for $c' \neq c$ such that s' is either $f_{c'}$ or $\bar{s}_{c',j}$ (depending on whether c' is large or small) and s' and f_c are merged in $\mathrm{OPT}(S_r)$ in this order. Consider string $\hat{R}_{c'} := \mathrm{strings}(c', s')s'$[1]. Note that $\mathrm{period}(\hat{R}_{c'}) \leq w(c')$ as $\hat{R}_{c'} = \mathrm{strings}(c', s')s'$, s' is a prefix of $\mathrm{strings}(c', s')^\infty$, and $|\mathrm{strings}(c', s')| = w(c')$. Furthermore, $\hat{R}_{c'}$ contains all strings of cycle c' as substrings, and thus, $\hat{R}_{c'}$ is inequivalent to α_c by Observation 5.6. Since f_c is a prefix of $\alpha_c[k_c - k'_c]$ and s' is a suffix of $\hat{R}_{c'}$, we have $|\mathrm{ov}(s', f_c)| \leq |\mathrm{ov}(\hat{R}_{c'}, \alpha_c[k_c - k'_c])|$. Using this together with Lemma 6.3 for α_c, k'_c, and $\hat{R}_{c'}$, it holds that

$$\begin{aligned} |\mathrm{ov}(s', f_c)| &\leq |\mathrm{ov}(\hat{R}_{c'}, \alpha_c[k_c - k'_c])| \\ &< \mathrm{period}(\hat{R}_{c'}) + \frac{\mathrm{period}(\alpha_c)}{2} + k'_c \\ &< w(c') + \frac{1}{2} w(c) + |f_c| - o(c)\,, \end{aligned} \tag{11}$$

where the third inequality uses $\mathrm{period}(\hat{R}_{c'}) \leq w(c')$, $\mathrm{period}(\alpha_c) \leq w(c)$ (by the definition of $\alpha_c = s(c)^\infty$ and $|s(c)| = w(c)$), and $k'_c < |f_c| - o(c)$.

Second, consider string $\bar{s}_{c,j}$ for a small cycle c (recall that $\bar{s}_{c,j}$ may be the result of merging several strings appearing consecutively on c). Let s' be the string merged to $\bar{s}_{c,j}$ in $\mathrm{OPT}(S_r)$ in this order, and let c' be the (large or small) cycle of string s'. From Corollary 5.10 we get

$$|\mathrm{ov}(s', \bar{s}_{c,j})| < w(c') + w(c)\,. \tag{12}$$

Observe that $n_r \geq \sum_s (|s| - |\mathrm{ov}(s', s)|)$, where the sum is over strings f_c and $\bar{s}_{c,j}$ as defined above and s' is the string merged to s in $\mathrm{OPT}(S_r)$ (s' is empty for the first string in $\mathrm{OPT}(S_r)$). Next, we use (11) or (12) to bound $|\mathrm{ov}(s', s)|$ for all such strings s. In particular, since each such string appears once as string s' (except for the last one), we get that

$$\begin{aligned} n_r \geq & \sum_{c \in \mathcal{L}(S)} (|f_c| - 1.5 \cdot w(c) - (|f_c| - o(c))) \\ & + \sum_{c \in \mathcal{S}(S)} \sum_{j=0}^{r'_c - 1} \left(|\bar{s}_{c,j}| - 2 \cdot w(c)\right). \end{aligned} \tag{13}$$

Using Lemma 6.1, we lower-bound the second term in the right-hand side of (13) and obtain

$$n_r \geq \sum_{c \in \mathcal{L}(S)} \left(o(c) - 1.5 \cdot w(c)\right) + \sum_{c \in \mathcal{S}(S)} \left(o(c) - w(c)\right)$$

Using that $n = |\mathrm{OPT}(S)| \geq |\mathrm{OPT}(S_r)| = n_r$ as $S_r \subseteq S$, and that $o = \sum_{c \in \mathcal{L}(S)} o(c) + \sum_{c \in \mathcal{S}(S)} o(c)$, we obtain

$$n \geq o - 1.5 \cdot \sum_{c \in \mathcal{L}(S)} w(c) - \sum_{c \in \mathcal{S}(S)} w(c)\,,$$

which completes the proof by rearranging. □

7 THE SECOND UPPER BOUND

In this section we show (5). The first ingredient of our analysis is a suitable modification of the input set of strings S.

7.1 Modifying the Input

For each small cycle $c = s_{c_0} \to s_{c_1} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$ in $\mathrm{CC}(S)$, we remove all strings belonging to this cycle from S and instead add the string

$$R'_c := \mathrm{pref}(s_{c_0}, s_{c_1})\mathrm{pref}(s_{c_1}, s_{c_2})\ldots\mathrm{pref}(s_{c_{r-2}}, s_{c_{r-1}})\mathrm{pref}(s_{c_{r-1}}, s_{c_0})s_{c_0}$$

to S. Note that the representative string R_c is a prefix of R'_c and thus, R'_c contains all strings of the small cycle c. We denote the new set of strings obtained this way by S'.

The length of $\mathrm{CC}(S')$ is the same as the length of $\mathrm{CC}(S)$. Indeed, due to Lemma 5.5, the generated optimal cycle cover remains the same except that whenever we had a small cycle c involving nodes $s_{c_0}, s_{c_1}, \ldots, s_{c_{r-1}}$ before, we now only have a single node (corresponding to the string R'_c) and a self-loop at that node. In addition, the length of small cycles does not change, i.e., $\sum_{c \in \mathcal{S}(S')} w(c) = \sum_{c \in \mathcal{S}(S)} w(c)$, again by Lemma 5.5.

However, the length $n' = |\mathrm{OPT}(S')|$ of the shortest superstring of S' could increase compared to the length $n = |\mathrm{OPT}(S)|$ of the optimal shortest superstring of S. The following lemma gives a bound on the increase.

Lemma 7.1. *The shortest superstring for S' is longer by at most $\sum_{c \in \mathcal{S}(S)} w(c)$ characters than the shortest superstring for S.*

Proof. We show how to transform any superstring σ for S into a superstring σ' for S' (which is also a superstring for S as R'_c contains all strings of the small cycle c) while only increasing the length of the superstring by $\sum_{c \in \mathcal{S}(S)} w(c)$, i.e., $|\sigma'| \leq |\sigma| + \sum_{c \in \mathcal{S}(S)} w(c)$.

[1] Strictly speaking, $\mathrm{strings}(c', s')$ is only defined for a string s' of cycle c'. If c' is a small cycle and $s' = \bar{s}_{c',j}$ is a result of merging strings $t_j^0, t_j^1, \ldots, t_j^{\ell_j - 1}$ from cycle c', then we let $\mathrm{strings}(c', s') := \mathrm{strings}(c', t_j^0)$ so that $\hat{R}_{c'} = \mathrm{strings}(c', t_j^0)s'$.

Namely, for every small cycle $s_{c_0} \to s_{c_1} \to \cdots \to s_{c_{r-1}} \to s_{c_0}$ in $\mathrm{CC}(S)$, we replace the first occurrence of s_{c_0} in σ by R'_c. The resulting superstring is our new string σ', which by construction, contains all strings of S' as required.

For a small cycle c, the length of R'_c is equal to $|s_{c_0}| + w(c)$. Therefore, $|\sigma'| \leq |\sigma| + \sum_{c\in\mathcal{S}(S)} w(c)$ as claimed. □

Corollary 7.2. *Let $CC_0(S')$ be a directed Hamiltonian cycle of minimum length in the distance graph $G_{dist}(S')$. The length n of the shortest superstring for S is at least $|CC_0(S')| - \sum_{c\in\mathcal{S}(S')} w(c)$.*

Proof. The length n' of the shortest superstring for S' is at least $|\mathrm{CC}_0(S')|$, since we can form a Hamiltonian cycle of length at most n' by merging the first and last string of the shortest superstring. With this, the corollary follows from Lemma 7.1. □

Since the sum of overlap lengths of cycle-closing edges in $\mathrm{CC}(S')$, denoted o', cannot be smaller than o, the sum of overlap lengths of cycle-closing edges in $\mathrm{CC}(S)$, showing the following inequality

$$o' \leq |\mathrm{CC}_0(S')| + (\gamma - 1) \cdot \sum_{c\in\mathcal{S}(S')} w(c) + \sum_{c\in\mathcal{L}(S')} w(c) \qquad (14)$$

implies (5), due to Corollary 7.2.

7.2 Overview of the Proof

Before proceeding, we note that our goal is to show (14) and from now on we will only be concerned with the modified input S'. Therefore, for the sake of simplicity, we omit the set S' from the cycle cover notation from this point onward (for instance, we shall indicate $\mathrm{CC}(S')$ as CC and $\mathrm{CC}_0(S')$ as CC_0).

Consider a maximum directed Hamiltonian cycle CC_0 in $G_{\mathrm{ov}}(S')$ and note that CC_0 is, in particular, also a (not necessarily maximum) cycle cover in $G_{\mathrm{ov}}(S')$. We call the total profit of all the edges of a cycle cover in $G_{\mathrm{ov}}(S')$ the *total overlap* of the cycle cover. Our goal is to show that the total overlap of CC_0 is by at least

$$\sum_{c\in\mathcal{S}(S')} (o(c) - \gamma \cdot w(c)) + \sum_{c\in\mathcal{L}(S')} (o(c) - 2 \cdot w(c)) \qquad (15)$$

smaller than the total overlap of the optimal cycle cover CC. In terms of the distance graph, this implies that CC_0 has a length which is by at least $\sum_{c\in\mathcal{S}(S')}(o(c) - \gamma \cdot w(c)) + \sum_{c\in\mathcal{L}(S')}(o(c) - 2 \cdot w(c))$ larger than the length of CC. The length of CC is $\sum_{c\in\mathcal{S}(S')} w(c) + \sum_{c\in\mathcal{L}(S')} w(c)$. Therefore, (14) is then implied by the following sequence of calculations:

$$\begin{aligned}
|\mathrm{CC}_0| &\geq \sum_{c\in\mathcal{S}(S')} (o(c) - \gamma \cdot w(c)) + \sum_{c\in\mathcal{L}(S')} (o(c) - 2 \cdot w(c)) \\
&\quad + \sum_{c\in\mathcal{S}(S')} w(c) + \sum_{c\in\mathcal{L}(S')} w(c) \\
&= \sum_{c\in\mathcal{S}(S')} (o(c) - (\gamma - 1) \cdot w(c)) + \sum_{c\in\mathcal{L}(S')} (o(c) - w(c)) \\
&= o' - (\gamma - 1) \cdot \sum_{c\in\mathcal{S}(S')} w(c) - \sum_{c\in\mathcal{L}(S')} w(c),
\end{aligned}$$

and this implies (5), as noted above.

To show the desired lower bound on the difference of total overlap between CC and CC_0, we slowly "transform" CC_0 into CC and track how each step of the transformation increases the total overlap. Next, we describe these individual transformation steps in more detail.

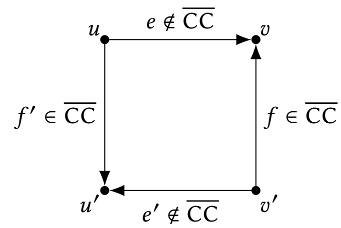

Figure 2: Illustration of the notation for $\mathrm{swap}(\overline{\mathrm{CC}}, e)$. Note that we also allow nodes to be equal to one another here, e.g., it could be that $u = v$, in which case e is a self-loop.

Consider any cycle cover $\overline{\mathrm{CC}}$ and a directed edge $e = (u, v)$ which is not contained in $\overline{\mathrm{CC}}$ (note that $u = v$ is possible because the graphs contain self-loops). Then we can modify $\overline{\mathrm{CC}}$ slightly such that it does contain e. Specifically, let $f = (v', v)$ be the incoming edge of v in $\overline{\mathrm{CC}}$ and $f' = (u, u')$ be the outgoing edge of u in $\overline{\mathrm{CC}}$. Then, we can add e and $e' = (v', u')$ to $\overline{\mathrm{CC}}$ and instead remove f and f' from $\overline{\mathrm{CC}}$. The resulting set of edges forms a cycle cover $\overline{\mathrm{CC}}'$ which now includes the edge e. We call this operation an *edge swap*. Note that the edge swap is completely determined by the given cycle cover $\overline{\mathrm{CC}}$ and the edge e. We refer to this unique swap as $\mathrm{swap}(\overline{\mathrm{CC}}, e)$ and always refer to the edges that are added to the cycle cover as e and e' and to the edges which are removed as f and f'; see Figure 2 for an illustration of the notation.

Given a cycle cover CC_0 (in our case the maximum Hamiltonian cycle) and the cycle cover CC, we can transform CC_0 into CC by a sequence of edge swaps. Specifically, if CC_i is a cycle cover, we can take any edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$, i.e., any edge in CC that is not in CC_i, and obtain a new cycle cover CC_{i+1} from CC_i by performing $\mathrm{swap}(\mathrm{CC}_i, e)$. Note that because $e \in \mathrm{CC}$, the edges f and f' which are swapped out in $\mathrm{swap}(\mathrm{CC}_i, e)$ cannot be part of CC. If e' belongs to CC, the symmetric difference between CC_{i+1} and CC contains four fewer edges than the symmetric difference between CC_i and CC (namely all four edges e, e', f, and f'). If e' is not part of CC, the symmetric difference between CC_{i+1} and CC contains two fewer edges than the symmetric difference between CC_i and CC (it no longer contains e, f, and f', but it now contains e'). In either case, the number of edges in the symmetric difference always decreases and therefore, after a finite number of such edge swap operations, we obtain a cycle cover CC_ℓ which is identical to CC.

If we obtain CC_{i+1} from CC_i by swapping in the edges e and e' and swapping out the edges f and f', then the total overlap of CC_{i+1} is larger than the total overlap of CC_i by $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')|$.

For a cycle cover CC_i, let $\mathcal{M}(\mathrm{CC}_i)$ be the set of small cycles of CC which are also part of CC_i. In other words, if CC_i contains a self-loop (s, s) and the string s corresponds to a small cycle c in CC, then (and only then) $c \in \mathcal{M}(\mathrm{CC}_i)$. Note that since $\mathrm{swap}(\mathrm{CC}_i, e)$ for $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ only removes edges $f, f' \in \mathrm{CC}_i \setminus \mathrm{CC}$ from CC_i, it holds that $\mathcal{M}(\mathrm{CC}_{i+1}) \supseteq \mathcal{M}(\mathrm{CC}_i)$.

Ideally, we would want to show that we can always choose an edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that the total overlap increase from CC_i to CC_{i+1} is at least $\sum_{c \in \mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i)} (o(c) - \gamma \cdot w(c))$. It would not be difficult to see that summing over all i would then imply the desired result, i.e., inequality (15), even without the sum over large cycles. Unfortunately, this appears difficult and in some cases we have to allow for slightly smaller increases. To address this, we relate some small cycles and some large cycles to one another.

We define a relation T between small cycles and a large cycle as follows. A small cycle c of CC and a large cycle c' of CC are *related* if $(\gamma - 2) \cdot w(c) \leq w(c')$ and the large cycle has a string s' such that $|\mathrm{ov}(s, s')| \geq \beta \cdot w(c')$ or $|\mathrm{ov}(s', s)| \geq \beta \cdot w(c')$, where s is the only string corresponding to the small cycle, by the input modification in Section 7.1. In this case, and only in this case, we have $(c, c') \in T$.

Lemma 7.3. *For every large cycle c' of CC, at most two different small cycles of CC are related to c'.*

Proof. Suppose for a contradiction that there are three small cycles c_1, c_2, and c_3 related to cycle c'. For $j \in \{1, 2, 3\}$, let s_j be the only string of cycle c_j and let o_j be the overlap from the definition of the relation satisfying $|o_j| \geq \beta \cdot w(c')$, i.e., either $o_j = \mathrm{ov}(s_j, s'_j)$ or $o_j = \mathrm{ov}(s'_j, s_j)$ for some string s'_j from c'. Note that since o_j is a suffix or prefix of s_j (depending on whether $o_j = \mathrm{ov}(s_j, s'_j)$ or $o_j = \mathrm{ov}(s'_j, s_j)$), Corollary 5.10 implies

$$|\mathrm{ov}(o_1, o_2)| < w(c_1) + w(c_2) \leq \frac{2}{\gamma - 2} \cdot w(c'), \tag{16}$$

where the second inequality holds as both c_1 and c_2 are related to c'. Using the same argument, both $|\mathrm{ov}(o_2, o_3)|$ and $|\mathrm{ov}(o_3, o_1)|$ are also strictly smaller than $\frac{2}{\gamma-2} \cdot w(c')$.

Each overlap string o_j appears as substring in the semi-infinite string $s(c')^\infty$ for the large cycle c', since each s'_j is a substring of $s(c')^\infty$ by Lemma 5.3. For $j \in \{1, 2, 3\}$, let $i_j \in [1, w(c')]$ be the smallest index such that o_j is a prefix of $s(c')^\infty[i_j]$. W.l.o.g., suppose that $i_1 \leq i_2 \leq i_3$ (by reordering indexes of c_1, c_2, and c_3). Observe that

$$i_2 - i_1 > \left(\beta - \frac{2}{\gamma - 2}\right) w(c'),$$

since otherwise, o_1 and o_2 would overlap by at least $\frac{2}{\gamma-2} w(c')$ (using that o_1 and o_2 have length at least $\beta \cdot w(c')$), contradicting (16). Similarly, it holds that $i_3 - i_2 > \left(\beta - \frac{2}{\gamma-2}\right) w(c')$ and $i_1 + w(c') - i_3 > \left(\beta - \frac{2}{\gamma-2}\right) w(c')$; for the latter, we use that o_1 is also a prefix of $s(c')^\infty[i_1 + w(c')]$ as $|s(c')| = w(c')$ is the length of the smallest periodicity of $s(c')^\infty$. Finally, we get a contradiction as follows:

$$w(c') = (i_2 - i_1) + (i_3 - i_2) + (i_1 + w(c') - i_3) > 3 \cdot \left(\beta - \frac{2}{\gamma - 2}\right) \cdot w(c') \geq w(c'),$$

where the last step uses (7). □

With this we define

$$\Delta_i = \sum_{c \in \mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i)} \left(o(c) - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c' : (c, c') \in T} (2 \cdot w(c') - o(c')) \right).$$

We will show that, for every i, we can choose $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that the total overlap increase from CC_i to CC_{i+1} is at least Δ_i when we obtain CC_{i+1} from CC_i by performing $\mathrm{swap}(\mathrm{CC}_i, e)$. Note that the value of Δ_i does depend on CC_{i+1} and therefore on the edge e that we choose.

Summing over all i gives the desired result since then the total overlap increase is at least

$$\begin{aligned}
\sum_{i=0}^{\ell-1} \Delta_i &= \sum_{c \in \mathcal{M}(\mathrm{CC}_\ell) \setminus \mathcal{M}(\mathrm{CC}_0)} \left(o(c) - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c' : (c, c') \in T} (2 \cdot w(c') - o(c')) \right) \\
&= \sum_{c \in \mathcal{S}(S')} \left(o(c) - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c' : (c, c') \in T} (2 \cdot w(c') - o(c')) \right) \\
&\geq \sum_{c \in \mathcal{S}(S')} \Big(o(c) - \gamma \cdot w(c) \Big) - \sum_{c' \in \mathcal{L}(S')} \Big(2 \cdot w(c') - o(c') \Big) \\
&= \sum_{c \in \mathcal{S}(S')} \Big(o(c) - \gamma \cdot w(c) \Big) + \sum_{c' \in \mathcal{L}(S')} \Big(o(c') - 2 \cdot w(c') \Big)
\end{aligned}$$

and this is what we wanted in (15). Here, the first line follows because $\mathcal{M}(\mathrm{CC}_{i+1}) \supseteq \mathcal{M}(\mathrm{CC}_i)$ for all i as noted above, the second line follows because $\mathrm{CC}_\ell = \mathrm{CC}$ and $\mathcal{M}(\mathrm{CC}_0) = \emptyset$, and the third line follows from Lemma 7.3. Strictly speaking, it is possible that $\mathcal{M}(\mathrm{CC}_0) \neq \emptyset$. However, CC_0 is a Hamiltonian cycle, and therefore, the only case in which this happens is if this Hamiltonian cycle is in fact a single small cycle c, in which case, by Observation 5.13, GREEDY computes an optimal solution.

We will sometimes use the fact that the term $2w(c') - o(c')$ is non-negative for every large cycle c'. Therefore, the part of the definition of Δ_i that sums over large cycles c' such that c is related to c' can only decrease the value of Δ_i (and makes it easier to find a suitable edge e in some cases), i.e.,

$$\Delta_i \leq \sum_{c \in \mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i)} \Big(o(c) - \gamma \cdot w(c) \Big). \tag{17}$$

In Section 7.4, we will show that for any cycle cover $\mathrm{CC}_i \neq \mathrm{CC}$, it is always possible to find an edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that if we obtain CC_{i+1} by performing the $\mathrm{swap}(\mathrm{CC}_i, e)$, the total overlap increase is at least Δ_i. Before that, we present three useful lemmas.

7.3 Useful Lemmas

Tarhio and Ukkonen [25] and Turner [27] show the following lemma.

Lemma 7.4. *Let $e = (u, v)$, $f = (v', v)$, $f' = (u, u')$, and $e' = (v', u')$ be edges in $G_{ov}(S')$ such that*

$$\max\{|ov(e)|, ov(e')|\} \geq \max\{|ov(f)|, |ov(f')|\}.$$

Then $|ov(e)| + |ov(e')| - |ov(f)| - |ov(f')| \geq 0$.

The following is a slightly different, but somewhat related inequality which gives us better bounds when e is the edge of a small cycle in CC. Another difference to Lemma 7.4 is that the following lemma can also be applied if $\max\{|\mathrm{ov}(e)|, |\mathrm{ov}(e')|\} < \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$.

Lemma 7.5. *Let $e = (u, v)$, $f = (v', v)$, $f' = (u, u')$, and $e' = (v', u')$ be edges in $G_{ov}(S')$ such that e is an edge in a small cycle c*

in CC. Then

$$|ov(e)| + |ov(e')| - |ov(f)| - |ov(f')| > |ov(e)| - \max\{|ov(f)|, |ov(f')|\} - w(c) \ .$$

Proof. If $\min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} < w(c)$, then trivially $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| \geq |\mathrm{ov}(e)| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| > |\mathrm{ov}(e)| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c)$ and we are done. So now assume $\min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} \geq w(c)$.

First note that since e is an edge of a small cycle in CC, e is a self-loop in $G_{\mathrm{ov}}(S')$ and $u = v$. Since $\mathrm{ov}(f)$ is a prefix of $u = v$, we observe that $\mathrm{ov}(f) = u[1, |\mathrm{ov}(f)|]$. Because u has period $w(c)$ by Lemma 5.11, this also implies $\mathrm{ov}(f) = u[1 + k \cdot w(c), |\mathrm{ov}(f)| + k \cdot w(c)]$, where we choose $k \geq 0$ as the largest integer for which $k \cdot w(c) \leq |u| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$. For this choice of k, we have $k \cdot w(c) > |u| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c)$.

Furthermore, $\mathrm{ov}(f') = u[|u| - |\mathrm{ov}(f')| + 1, |u|]$ because $\mathrm{ov}(f')$ is a suffix of u. Hence, the string $u[|u| - |\mathrm{ov}(f')| + 1, |\mathrm{ov}(f)| + k \cdot w(c)]$ is a suffix of $\mathrm{ov}(f)$ as well as a prefix of $\mathrm{ov}(f')$. This string has length

$$\begin{aligned} &|\mathrm{ov}(f)| + k \cdot w(c) - (|u| - |\mathrm{ov}(f')|) \\ &\quad > |\mathrm{ov}(f)| + |u| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c) \\ &\qquad - (|u| - |\mathrm{ov}(f')|) \\ &\quad = \min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c) \ , \end{aligned}$$

which is non-negative by the assumption above.

Every suffix of $\mathrm{ov}(f)$ is also a suffix of v' and every prefix of $\mathrm{ov}(f')$ is also a prefix of u'. Hence, v' has a suffix of length larger than $\min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c)$ which is identical to a prefix of u'. Therefore, $|\mathrm{ov}(e')| > \min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c)$, which implies the lemma. □

Under a certain condition, we can further strengthen the inequality of the previous lemma.

Lemma 7.6. *Consider the edges $e = (u, v)$, $f = (v', v)$, $f' = (u, u')$, and $e' = (v', u')$. Suppose e is an edge in a (large or small) cycle c of CC, e' is an edge in a (large or small) cycle c' of CC, and $|ov(e')| \geq w(c) + w(c')$. Then*

$$|ov(e)| + |ov(e')| - |ov(f)| - |ov(f')| > |ov(e)| - w(c) \ .$$

Proof. We show that $|\mathrm{ov}(e')| > |\mathrm{ov}(f)| + |\mathrm{ov}(f')| - w(c)$, which implies the lemma. If $\min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} \leq w(c)$, this inequality holds because by using Lemma 5.9, we get

$$\begin{aligned} |\mathrm{ov}(e')| &\geq w(c) + w(c') \\ &> \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} \\ &\geq \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} + \min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c) \\ &= |\mathrm{ov}(f)| + |\mathrm{ov}(f')| - w(c) \ . \end{aligned}$$

Hence, for the remainder of the proof, we assume that we have $\min\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} > w(c)$.

Now, assume for contradiction that $|\mathrm{ov}(e')| \leq |\mathrm{ov}(f)| + |\mathrm{ov}(f')| - w(c)$. We claim that in this case $\mathrm{ov}(e')$ has a periodicity of length $w(c)$, i.e., $\mathrm{ov}(e')$ is a prefix of x^∞ for some string x with $|x| = w(c)$. To show this, first recall that $|\mathrm{ov}(e')| \geq w(c) + w(c') > \max\{|\mathrm{ov}(f')|, |\mathrm{ov}(f)|\}$ by Lemma 5.9. Since $\mathrm{ov}(f')$ is a prefix of u' and a suffix of u and since $\mathrm{ov}(e')$ is a prefix of u', the first $|\mathrm{ov}(f')|$ characters of $\mathrm{ov}(e')$ are also a suffix of u, i.e.,

$$\mathrm{ov}(e')[1, |\mathrm{ov}(f')|] = \mathrm{ov}(f') = u[|u| - |\mathrm{ov}(f')| + 1, |u|] \ .$$

Similarly, since $\mathrm{ov}(f)$ is a prefix of v and a suffix of v' and since $\mathrm{ov}(e')$ is a suffix of v', we get that

$$\mathrm{ov}(e')[|\mathrm{ov}(e')| - |\mathrm{ov}(f)| + 1, |\mathrm{ov}(e')|] = \mathrm{ov}(f) = v[1, |\mathrm{ov}(f)|] \ .$$

Observe that for all $1 \leq i \leq |\mathrm{ov}(e')| - w(c)$, a character at position i of $\mathrm{ov}(e')$ must be the same as the character at position $i + w(c)$ of $\mathrm{ov}(e')$. Indeed, if $i + w(c) \leq |\mathrm{ov}(f')|$, this is true as u has a periodicity of length $w(c)$. If $i > |\mathrm{ov}(e')| - |\mathrm{ov}(f)|$, it is true because v has a periodicity of length $w(c)$. One of these two cases must apply because otherwise, $i + w(c) > |\mathrm{ov}(f')|$ and $i \leq |\mathrm{ov}(e')| - |\mathrm{ov}(f)|$, which implies $|\mathrm{ov}(f')| - w(c) < i \leq |\mathrm{ov}(e')| - |\mathrm{ov}(f)|$, contradicting our assumption that $|\mathrm{ov}(f')| + |\mathrm{ov}(f)| \geq |\mathrm{ov}(e')| + w(c)$. Hence, $\mathrm{ov}(e')$ has a periodicity of length $w(c)$ (in particular, $\mathrm{period}(\mathrm{ov}(e')) \leq w(c)$).

Next, we show that $\mathrm{ov}(e')$ is a substring of the semi-infinite string $s(c)^\infty$. Because $\mathrm{ov}(e')$ has a periodicity of length $w(c)$ and $s(c)^\infty$ has period $w(c)$, it is sufficient to argue that the first $w(c)$ characters of $\mathrm{ov}(e')$ are a substring of $s(c)^\infty$. This is indeed the case since $\mathrm{ov}(e')[1, |\mathrm{ov}(f')|]$ is a substring of u which is a substring of $s(c)^\infty$ and we assume that $|\mathrm{ov}(f')| > w(c)$.

Since $\mathrm{ov}(e')$ is a substring of $s(c)^\infty$ as well as of $s(c')^\infty$ (because e' lies on cycle c'), Corollary 5.10 implies $|\mathrm{ov}(e')| < w(c) + w(c')$ which contradicts the assumption of the lemma. □

7.4 Analysis

In this section, we will show that for any cycle cover $\mathrm{CC}_i \neq \mathrm{CC}$, it is always possible to find an edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that if we obtain CC_{i+1} by performing the $\mathrm{swap}(\mathrm{CC}_i, e)$, the total overlap increase is at least

$$\Delta_i = \sum_{c \in \mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i)} \left(o(c) - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c' : (c, c') \in T} (2 \cdot w(c') - o(c')) \right).$$

The following defines the concept of a *good edge*. It is a slightly technical definition, but it is useful in the sense that (a) we will be able to show that a good edge e is always a suitable choice for $\mathrm{swap}(\mathrm{CC}_i, e)$ and (b) in many cases we can find a good edge. For the remaining cases (i.e., when it is not obvious whether a good edge exists), we will have separate arguments that show that an appropriate swap is possible.

Definition 7.7. We call an edge $e = (u, v) \in \mathrm{CC} \setminus \mathrm{CC}_i$ a *good* edge if the following statements hold for $\mathrm{swap}(\mathrm{CC}_i, e)$ which swaps out edges $f = (v', v) \in \mathrm{CC}_i \setminus \mathrm{CC}$ and $f' = (u, u') \in \mathrm{CC}_i \setminus \mathrm{CC}$ and swaps in edges $e = (u, v)$ and $e' = (v', u')$:

- e belongs to a small cycle c of CC and e' does not belong to a small cycle of CC.
- If $|\mathrm{ov}(f)| \geq |\mathrm{ov}(f')|$, then for the cycle c' in CC that contains the string v', it holds that either $|\mathrm{ov}(f)| \geq o(c')$ or c' is a small cycle with $w(c') \leq w(c)$.
- If $|\mathrm{ov}(f')| > |\mathrm{ov}(f)|$, then for the cycle c' in CC that contains the string u', it holds that either $|\mathrm{ov}(f')| \geq o(c')$ or c' is a small cycle with $w(c') \leq w(c)$.

The following lemma shows that if there is a good edge e, performing $\mathrm{swap}(\mathrm{CC}_i, e)$ results in a sufficient increase of the total overlap.

Lemma 7.8. *If e is a good edge, then after performing $\mathrm{swap}(CC_i, e)$, the resulting cycle cover CC_{i+1} has by at least Δ_i larger total overlap than CC_i.*

Proof. By definition of a good edge, e is the edge of a small cycle c. Due to Lemma 7.5, $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| > |\mathrm{ov}(e)| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c)$.

Suppose $|\mathrm{ov}(f)| \ge |\mathrm{ov}(f')|$ (the other case is analogous) and let c' be the cycle containing the string v'. Then

$$\begin{aligned}
&|\mathrm{ov}(e)| - \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} - w(c) \\
&\quad = |\mathrm{ov}(e)| - |\mathrm{ov}(f)| - w(c) \\
&\quad > |\mathrm{ov}(e)| - w(c) - w(c') - w(c) = |\mathrm{ov}(e)| - 2w(c) - w(c')\,,
\end{aligned}$$

where the inequality follows from Lemma 5.9. Hence, it is sufficient to show that $|\mathrm{ov}(e)| - 2w(c) - w(c') \ge \Delta_i$.

Since e is the only edge of a small cycle in CC and e' is not an edge of a small cycle in CC (by the definition of a good edge), if we obtain CC_{i+1} from CC_i by performing $\mathrm{swap}(\mathrm{CC}_i, e)$, then $\mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i) = \{c\}$. In this case,

$$\begin{aligned}
\Delta_i &= o(c) - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c'':(c,c'')\in T} (2 \cdot w(c'') - o(c'')) \\
&= |\mathrm{ov}(e)| - \gamma \cdot w(c) - \frac{1}{2} \cdot \sum_{c'':(c,c'')\in T} (2 \cdot w(c'') - o(c'')) \\
&\le \begin{cases} |\mathrm{ov}(e)| - \gamma \cdot w(c) - \frac{1}{2} \cdot (2 \cdot w(c') - o(c')) & \text{if } (c, c') \in T \\ |\mathrm{ov}(e)| - \gamma \cdot w(c) & \text{otherwise} \end{cases} \\
&\le \begin{cases} |\mathrm{ov}(e)| - \gamma \cdot w(c) - \frac{1}{2} \cdot (w(c') - w(c)) & \text{if } (c, c') \in T \\ |\mathrm{ov}(e)| - \gamma \cdot w(c) & \text{otherwise} \end{cases} .
\end{aligned} \tag{18}$$

The last step follows since if $(c, c') \in T$, then c' is a large cycle and therefore, $o(c') \le |\mathrm{ov}(f)| < w(c)+w(c')$, where the first inequality follows from the definition of a good edge and the last inequality follows from Lemma 5.9.

The following fact establishes an upper bound on $w(c')$ by a function of $w(c)$.

Fact 7.9.

- *If c' is a large cycle and $(c, c') \in T$, then $w(c') < \frac{1}{\beta-1} w(c)$.*
- *Otherwise, $w(c') < (\gamma - 2) \cdot w(c)$ holds.*

Proof. If c' is a large cycle, then $w(c) + w(c') > |\mathrm{ov}(f)| \ge o(c') > \beta \cdot w(c')$, where the first step follows from Lemma 5.9, the second step follows from the definition of a good edge, and the last step follows from the definition of a large cycle. Rearranging this inequality gives $w(c') < \frac{1}{\beta-1} w(c)$.

Now, to show the second claim, there are two cases. If c' is a large cycle, but $(c, c') \notin T$, then we again recall that $|\mathrm{ov}(f)| \ge \beta \cdot w(c')$. Since $(c, c') \notin T$, this implies that $w(c') < (\gamma - 2) \cdot w(c)$ as claimed. On the other hand, if c' is a small cycle, then, due to the definition of a good edge, either $w(c') \le w(c)$ or $|\mathrm{ov}(f)| \ge o(c')$. In the former case, we are already done as $\gamma > 3$. In the latter case, $|\mathrm{ov}(f)| \ge o(c') > 2w(c')$ and hence $w(c) > |\mathrm{ov}(f)| - w(c') > w(c')$, where the first inequality follows from Lemma 5.9. Again, this implies the second claim as $\gamma > 3$. □

Finally, to show that $|\mathrm{ov}(e)| - 2w(c) - w(c') \ge \Delta_i$, we distinguish two cases and utilize the upper bound on Δ_i derived in (18).

- If c' is large cycle and $(c, c') \in T$, then using the first claim in Fact 7.9,

$$\begin{aligned}
&|\mathrm{ov}(e)| - 2w(c) - w(c') \\
&\quad = |\mathrm{ov}(e)| - 2w(c) - \frac{1}{2} w(c') - \frac{1}{2} w(c') \\
&\quad > |\mathrm{ov}(e)| - 2w(c) - \frac{1}{2} w(c') - \frac{1}{2(\beta-1)} w(c) \\
&\quad = |\mathrm{ov}(e)| - \left(2 + \frac{1}{2(\beta-1)}\right) \cdot w(c) - \frac{1}{2} w(c') \\
&\quad = |\mathrm{ov}(e)| - \left(\frac{5}{2} + \frac{1}{2(\beta-1)}\right) \cdot w(c) - \frac{1}{2} w(c') + \frac{1}{2} w(c) \\
&\quad \ge |\mathrm{ov}(e)| - \gamma \cdot w(c) - \frac{1}{2} w(c') + \frac{1}{2} w(c) \ge \Delta_i\,,
\end{aligned}$$

  where the last line uses (8).
- Otherwise, $|\mathrm{ov}(e)| - 2w(c) - w(c') \ge |\mathrm{ov}(e)| - \gamma \cdot w(c) \ge \Delta_i$. □

There may be cases where $\mathrm{CC} \setminus \mathrm{CC}_i$ does not necessarily have a good edge. In such cases, we can use other arguments. The following lemma is an example of this.

Lemma 7.10. *If there exists an edge $e \in CC \setminus CC_i$ such that*

(i) $\mathrm{swap}(CC_i, e)$ swaps in edges e and e',
(ii) neither e nor e' are edges of a small cycle in CC, and
(iii) $\max\{|ov(e)|, |ov(e')|\} \ge \max\{|ov(f)|, |ov(f')|\}$,

then after performing $\mathrm{swap}(CC_i, e)$, the resulting cycle cover CC_{i+1} has by at least Δ_i larger total overlap than CC_i.

Proof. If neither e nor e' are edges of a small cycle in CC, then performing $\mathrm{swap}(\mathrm{CC}_i, e)$ results in a cycle cover CC_{i+1} for which $\mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i) = \emptyset$. Therefore, $\Delta_i = 0$. Since we have $\max\{|\mathrm{ov}(e)|, |\mathrm{ov}(e')|\} \ge \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$, Lemma 7.4 implies that $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| \ge |\mathrm{ov}(f)| + |\mathrm{ov}(f')|$. Hence, $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| \ge 0 = \Delta_i$. □

If there exists an edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that performing $\mathrm{swap}(\mathrm{CC}_i, e)$ reduces the symmetric difference between CC and CC_i by four, then we show that $\mathrm{swap}(\mathrm{CC}_i, e)$ increases the total overlap by at least Δ_i.

Lemma 7.11. *If there exists an edge $e \in CC \setminus CC_i$ such that performing $\mathrm{swap}(CC_i, e)$ reduces the symmetric difference between the cycle cover CC_i and CC by four edges, then after performing $\mathrm{swap}(CC_i, e)$, the resulting cycle cover CC_{i+1} has by at least Δ_i larger total overlap than CC_i.*

Proof. Recall that $\mathrm{swap}(\mathrm{CC}_i, e)$ adds the edges e and e' to the cycle cover CC_i and removes the edges f and f'. Thus, if the symmetric difference to CC decreases by four edges, then it must be the case that $e, e' \in \mathrm{CC} \setminus \mathrm{CC}_i$ and $f, f' \in \mathrm{CC}_i \setminus \mathrm{CC}$.

We have $\max\{|\mathrm{ov}(e)|, |\mathrm{ov}(e')|\} \ge \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$, since otherwise, MGREEDY would have picked the edge of greater overlap between f and f' for inclusion in CC, before picking either one of e or e'. First, suppose that both e and e' belong to the same cycle, which can only be large, since all small cycles consist of a single edge. In this case, Lemma 7.10 applies and the proof is complete. Now, suppose that e and e' belong to two different cycles of CC. We consider the following four cases:

- Suppose e and e' both belong to large cycles in CC. Then Lemma 7.10 applies and we are done.
- Suppose e and e' both belong to small cycles in CC. Let these two small cycles be c and c', respectively. If we obtain CC_{i+1} from CC_i by performing $\mathrm{swap}(\mathrm{CC}_i, e)$, then $\mathcal{M}(\mathrm{CC}_{i+1}) \setminus \mathcal{M}(\mathrm{CC}_i) = \{c, c'\}$. Thus, using (17) together with $|\mathrm{ov}(e)| = o(c)$, $|\mathrm{ov}(e')| = o(c')$, and $\gamma > 2$, we obtain
$$\Delta_i < |\mathrm{ov}(e)| - 2w(c) + |\mathrm{ov}(e')| - 2w(c') .$$
Due to Lemma 5.9, $\max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} < w(c) + w(c')$. Therefore, $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| \geq |\mathrm{ov}(e)| - 2w(c) + |\mathrm{ov}(e')| - 2w(c') > \Delta_i$ as claimed.
- Suppose e belongs to a small cycle c and e' belongs to a large cycle c' in CC.
We distinguish between three cases:
  - If $|\mathrm{ov}(e')| \leq \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$, then e is a good edge (note that $o(c') \leq |\mathrm{ov}(e')|$ because e' belongs to the cycle c') and we apply Lemma 7.8.
  - If $w(c) + w(c') \geq |\mathrm{ov}(e')| > \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$, then using Lemma 5.9,
$$\begin{aligned}\max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\} &< w(c) + w(c') \\ &= w(c) + \gamma \cdot w(c') - (\gamma - 1) \cdot w(c') \\ &\leq w(c) + (\gamma - 1) \cdot \beta \cdot w(c') - (\gamma - 1) \cdot w(c') \\ &\leq w(c) + (\gamma - 1) \cdot o(c') - (\gamma - 1) \cdot w(c') \\ &\leq w(c) + (\gamma - 1) \cdot |\mathrm{ov}(e')| - (\gamma - 1) \cdot w(c') \\ &\leq w(c) + (\gamma - 1) \cdot w(c) = \gamma \cdot w(c) ,\end{aligned}$$
where the second line uses (9), the third line follows from c' being large, the fourth one from that $o(c')$ is the smallest overlap on cycle c', and the fifth line uses the case condition. Now, the increase in the total overlap when performing $\mathrm{swap}(\mathrm{CC}_i, e)$ is at least $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| \geq |\mathrm{ov}(e)| - |\mathrm{ov}(f)| > o(c) - \gamma \cdot w(c) \geq \Delta_i$, where we use (17) together with $|\mathrm{ov}(e)| = o(c)$ and $\gamma > 1$.
  - Otherwise, we have $|\mathrm{ov}(e')| > w(c) + w(c')$. It follows, by Lemma 7.6, that $|\mathrm{ov}(e)| + |\mathrm{ov}(e')| - |\mathrm{ov}(f)| - |\mathrm{ov}(f')| \geq |\mathrm{ov}(e)| - w(c) \geq \Delta_i$.
- Suppose e belongs to a large cycle and e' belongs to a small cycle in CC. Observe that $\mathrm{swap}(\mathrm{CC}_i, e')$ results in exactly the same cycle cover CC_{i+1} as $\mathrm{swap}(\mathrm{CC}_i, e)$. Therefore, we just apply the previous argument to $\mathrm{swap}(\mathrm{CC}_i, e')$, and we are done. □

Lastly, if neither of the previous two lemmas applies, we can find a good edge for sure:

Lemma 7.12. *Suppose Lemmas 7.10 and 7.11 do not apply, i.e., no edge with the corresponding properties exists. Then there exists a good edge in $CC \setminus CC_i$.*

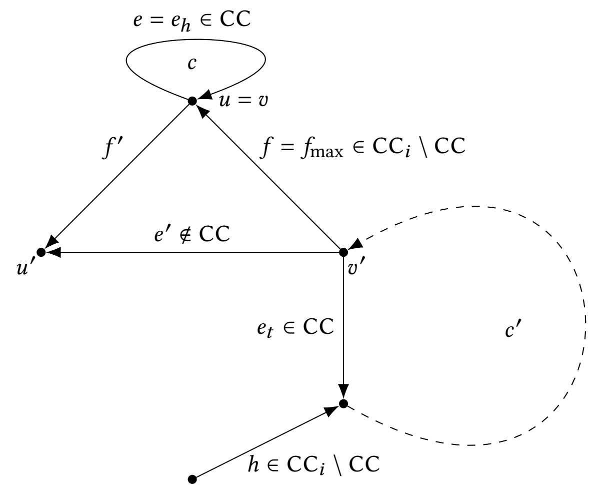

Figure 3: Illustration of Case A in the proof of Lemma 7.12.

Proof. Let $f_{\max}$ be an edge of $\mathrm{CC}_i \setminus \mathrm{CC}$ that has the maximum overlap among the edges of $\mathrm{CC}_i \setminus \mathrm{CC}$. We will show that $f_{\max}$ is a candidate for either f or f'.

Let e_h be the edge of CC that has the same head node as $f_{\max}$ and let e_t be the edge of CC that has the same tail node as $f_{\max}$. We will later pick one of these as our edge e. We have $|\mathrm{ov}(f_{\max})| \leq \max\{|\mathrm{ov}(e_h)|, |\mathrm{ov}(e_t)|\}$ as otherwise, MGREEDY would have picked edge $f_{\max}$ for inclusion in CC before picking either one of e_h or e_t.

We first show that e_h or e_t satisfies the first condition of a good edge in Definition 7.7.

Fact 7.13.

- *If $|ov(e_h)| \geq |ov(f_{\max})|$, then $e = e_h$ satisfies the first condition of a good edge.*
- *Similarly, if $|ov(e_t)| \geq |ov(f_{\max})|$, then $e = e_t$ satisfies the first condition of a good edge.*

Proof.

- To see that $e = e_h$ satisfies the first condition of a good edge if $|\mathrm{ov}(e_h)| \geq |\mathrm{ov}(f_{\max})|$, consider $\mathrm{swap}(\mathrm{CC}_i, e_h)$ and use the same notation as in Figure 2.
First of all, in this case, $f = f_{\max} = (v', v)$ and because $f_{\max}$ was chosen to have the maximum overlap in $\mathrm{CC}_i \setminus \mathrm{CC}$, $|\mathrm{ov}(f)| \geq |\mathrm{ov}(f')|$. From this we conclude that $|\mathrm{ov}(e)| \geq \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$. Lemma 7.11 applies if e and e' both belong to CC. Since we assume that the lemma does not apply and since we know that $e \in \mathrm{CC}$, it follows that $e' \notin \mathrm{CC}$. If e belongs to a large cycle in CC, Lemma 7.10 applies because $e' \notin \mathrm{CC}$ and $|\mathrm{ov}(e)| \geq \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(f')|\}$. Because we assume that the lemma does not apply, we conclude that e must belong to a small cycle. Together with $e' \notin \mathrm{CC}$, this satisfies the first condition of a good edge.
- By symmetric arguments to the above, it also follows that if $|\mathrm{ov}(e_t)| \geq |\mathrm{ov}(f_{\max})|$, then $e = e_t$ satisfies the first condition of a good edge. □

To show that we can also satisfy the second or the third condition (for an edge that satisfies the first), we distinguish three cases:

Case A: Suppose $|\mathrm{ov}(e_h)| \geq |\mathrm{ov}(f_{\max})|$ and $|\mathrm{ov}(e_t)| \geq |\mathrm{ov}(f_{\max})|$.

Let c be the cycle of CC to which e_h belongs and let c' be the cycle of CC to which e_t belongs; see Figure 3 for an illustration. We assume that $w(c) \geq w(c')$ as the arguments for the other case are completely symmetric with the roles of e_h and e_t reversed.

We claim that $e = e_h$ is a good edge. It follows from Fact 7.13 that e satisfies the first condition of a good edge. Since $|\mathrm{ov}(f)| \geq |\mathrm{ov}(f')|$ as $f = f_{\max}$, it only remains to show the second condition. Since e_t is an edge in the cycle c' in CC, we have $|\mathrm{ov}(e_t)| \geq o(c')$. If $o(c') \leq |\mathrm{ov}(f)|$, the second condition of a good edge is already satisfied. So suppose $o(c') > |\mathrm{ov}(f)|$.

Assume for a contradiction that c' is a large cycle in CC. Then consider the edge h in $\mathrm{CC}_i \setminus \mathrm{CC}$ that has the same head node as e_t. We know that $|\mathrm{ov}(f)| \geq |\mathrm{ov}(h)|$ because $f_{\max} = f$ was chosen to have the maximum overlap among all edges in $\mathrm{CC}_i \setminus \mathrm{CC}$. Hence, $|\mathrm{ov}(e_t)| \geq o(c') > |\mathrm{ov}(f)| \geq |\mathrm{ov}(h)|$ and thus $|\mathrm{ov}(e_t)| > \max\{|\mathrm{ov}(f)|, |\mathrm{ov}(h)|\}$. Consider $\mathrm{swap}(\mathrm{CC}_i, e_t)$, i.e., with edge e_t acting as edge e in the operation. If $\mathrm{swap}(\mathrm{CC}_i, e_t)$ reduces the symmetric difference between CC_i and CC by four edges, then Lemma 7.11 applies. Otherwise, $e' \notin \mathrm{CC}$, so Lemma 7.10 applies as the cycle c' containing $e = e_t$ is large. This is a contradiction to our assumption that neither Lemma 7.11 nor Lemma 7.10 can be applied.

Thus, c' must be a small cycle. Since we initially assumed that $w(c) \geq w(c')$, the second condition in Definition 7.7 follows and thus, e is a good edge.

Case B: Suppose that $|\mathrm{ov}(e_h)| \geq |\mathrm{ov}(f_{\max})| > |\mathrm{ov}(e_t)|$. We claim that $e = e_h$ is a good edge. It follows from Fact 7.13 that e satisfies the first condition of a good edge. Since $|\mathrm{ov}(f)| \geq |\mathrm{ov}(f')|$, it only remains to show the second condition.

Let c' be the cycle containing the string v'. Observe that e_t is an edge in the cycle c' and recall that $f = f_{\max}$ and $|\mathrm{ov}(e_t)| < |\mathrm{ov}(f_{\max})|$. We conclude that $o(c') \leq |\mathrm{ov}(e_t)| < |\mathrm{ov}(f)|$, so the second condition in Definition 7.7 is satisfied and e is good edge.

Case C: Otherwise, since $\max\{|\mathrm{ov}(e_h)|, |\mathrm{ov}(e_t)|\} \geq |\mathrm{ov}(f_{\max})|$, we have $|\mathrm{ov}(e_t)| \geq |\mathrm{ov}(f_{\max})| > |\mathrm{ov}(e_h)|$. This case is symmetric to the previous one with the roles of e_t and e_h swapped. □

To summarize, for any arbitrary cycle cover CC_i, there exists an edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$ such that if we obtain the cycle cover CC_{i+1} from CC_i by performing $\mathrm{swap}(\mathrm{CC}_i, e)$, then the total overlap of CC_{i+1} is by at least Δ_i larger than the total overlap of CC_i. This follows because either one of Lemmas 7.10 and 7.11 directly applies or, if that is not the case, Lemma 7.12 guarantees the existence of a good edge $e \in \mathrm{CC} \setminus \mathrm{CC}_i$. For such a good edge, $\mathrm{swap}(\mathrm{CC}_i, e)$ provides the claimed increase of the total overlap due to Lemma 7.8.

REFERENCES

[1] Chris Armen and Clifford Stein. 1995. Improved Length Bounds for the Shortest Superstring Problem. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures (WADS)*. 494–505. https://doi.org/10.1007/3-540-60220-8_88

[2] Chris Armen and Clifford Stein. 1998. A 2 2/3 Superstring Approximation Algorithm. *Discret. Appl. Math.* 88, 1-3 (1998), 29–57. https://doi.org/10.1016/S0166-218X(98)00065-1

[3] Avrim Blum, Tao Jiang, Ming Li, John Tromp, and Mihalis Yannakakis. 1994. Linear Approximation of Shortest Superstrings. *J. ACM* 41, 4 (1994), 630–647. https://doi.org/10.1145/179812.179818

[4] Dany Breslauer, Tao Jiang, and Zhigen Jiang. 1997. Rotations of Periodic Strings and Short Superstrings. *J. Algorithms* 24, 2 (1997), 340–353. https://doi.org/10.1006/jagm.1997.0861

[5] Artur Czumaj, Leszek Gasieniec, Marek Piotrów, and Wojciech Rytter. 1997. Sequential and Parallel Approximation of Shortest Superstrings. *J. Algorithms* 23, 1 (1997), 74–100. https://doi.org/10.1006/jagm.1996.0823

[6] Alan M. Frieze and Wojciech Szpankowski. 1998. Greedy Algorithms for the Shortest Common Superstring That Are Asymptotically Optimal. *Algorithmica* 21, 1 (1998), 21–36. https://doi.org/10.1007/PL00009207

[7] John Gallant, David Maier, and James A. Storer. 1980. On Finding Minimal Length Superstrings. *J. Comput. Syst. Sci.* 20 (1980), 50–58. https://doi.org/10.1016/0022-0000(80)90004-5

[8] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[9] Theodoros P. Gevezes and Leonidas S. Pitsoulis. 2014. *The Shortest Superstring Problem.* Springer, New York, NY, 189–227. https://doi.org/10.1007/978-1-4939-0808-0_10

[10] Dan Gusfield. 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press. https://doi.org/10.1017/CBO9780511574931

[11] Lucian Ilie and Cristian Popescu. 2006. The Shortest Common Superstring Problem and Viral Genome Compression. *Fundamenta Informaticae* 73, 1-2 (2006), 153–164.

[12] Haim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. 2005. Approximation Algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs. *J. ACM* 52, 4 (2005), 602–626. https://doi.org/10.1145/1082036.1082041

[13] Haim Kaplan and Nira Shafrir. 2005. The Greedy Algorithm for Shortest Superstrings. *Inf. Process. Lett.* 93, 1 (2005), 13–17. https://doi.org/10.1016/j.ipl.2004.09.012

[14] Marek Karpinski and Richard Schmied. 2013. Improved Inapproximability Results for the Shortest Superstring and Related Problems. In *Proceedings of the 19th Computing: The Australasian Theory Symposium (CATS).* 27–36.

[15] S. Rao Kosaraju, James K. Park, and Clifford Stein. 1994. Long Tours and Short Superstrings. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science (FOCS).* 166–177. https://doi.org/10.1109/SFCS.1994.365696

[16] Arthur M. Lesk. 1988. *Computational Molecular Biology: Sources and Methods for Sequence Analysis.* Oxford University Press.

[17] Ming Li. 1990. Towards a DNA Sequencing Theory (Learning a String). In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (FOCS).* 125–134. https://doi.org/10.1109/FSCS.1990.89531

[18] Bin Ma. 2009. Why Greed Works for Shortest Common Superstring Problem. *Theor. Comput. Sci.* 410, 51 (2009), 5374–5381. https://doi.org/10.1016/j.tcs.2009.09.014

[19] Marcin Mucha. 2007. A Tutorial on Shortest Superstring Approximation. https://www.mimuw.edu.pl/~mucha/teaching/aa2008/ss.pdf. [Accessed 28-October-2021].

[20] Marcin Mucha. 2013. Lyndon Words and Short Superstrings. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA).* 958–972. https://doi.org/10.1137/1.9781611973105.69

[21] Eugene W. Myers Jr. 2016. A history of DNA sequence assembly. *It-Information Technology* 58, 3 (2016), 126–132. https://doi.org/10.1515/itit-2015-0047

[22] Katarzyna Paluch. 2020. New Approximation Algorithms for Maximum Asymmetric Traveling Salesman and Shortest Superstring. https://arxiv.org/abs/2005.10800.

[23] Katarzyna Paluch, Khaled Elbassioni, and Anke van Zuylen. 2012. Simpler Approximation of the Maximum Asymmetric Traveling Salesman Problem. In *Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science (STACS).* 501–506. https://doi.org/10.4230/LIPIcs.STACS.2012.501

[24] Z. Sweedyk. 1999. A 2½-Approximation Algorithm for Shortest Superstring. *SIAM J. Comput.* 29, 3 (1999), 954–986. https://doi.org/10.1137/S0097539796324661

[25] Jorma Tarhio and Esko Ukkonen. 1988. A Greedy Approximation Algorithm for Constructing Shortest Common Superstrings. *Theor. Comput. Sci.* 57 (1988), 131–145. https://doi.org/10.1016/0304-3975(88)90167-3

[26] Shang-Hua Teng and Frances Yao. 1997. Approximating Shortest Superstrings. *SIAM J. Comput.* 26, 2 (1997), 410–417. https://doi.org/10.1137/S0097539794286125

[27] Jonathan S. Turner. 1989. Approximation Algorithms for the Shortest Common Superstring Problem. *Inf. Comput.* 83, 1 (1989), 1–20. https://doi.org/10.1016/0890-5401(89)90044-8