Approximation Guarantees for Shortest Superstrings: Simpler and Better

Matthias Englert
University of Warwick, United Kingdom

Nicolaos Matsakis
Charles University, Czech Republic

Pavel Veselý
Charles University, Czech Republic

Abstract

The Shortest Superstring problem is an NP-hard problem, in which given as input a set of strings, we are looking for a string of minimum length that contains all input strings as substrings. The Greedy Conjecture (Tarhio and Ukkonen, 1988) states that the GREEDY algorithm, which repeatedly merges the two strings of maximum overlap, is 2-approximate. We have recently shown (STOC 2022) that the approximation guarantee of GREEDY is at most $\frac{13+\sqrt{57}}{6} \approx 3.425$. Before that, the best established upper bound for this was 3.5 by Kaplan and Shafrir (IPL 2005), which improved upon the upper bound of 4 by Blum et al. (STOC 1991). To derive our previous result, we established two incomparable upper bounds on the overlap sum of all cycle-closing edges in an optimal cycle cover and utilized lemmas of Blum et al.

We improve the more involved one of the two bounds and, at the same time, make its proof more straightforward. This results in an improved approximation guarantee of $\frac{\sqrt{67}+2}{3} \approx 3.396$ for GREEDY. Additionally, our result implies an algorithm for the Shortest Superstring problem having an approximation guarantee of $\frac{\sqrt{67}+14}{9} \approx 2.466$, improving slightly upon the previously best guarantee of $\frac{\sqrt{57}+37}{18} \approx 2.475$ (STOC 2022).

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Shortest Superstring problem, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.26

Funding *Nicolaos Matsakis*: Supported by GA ČR project 22-22997S.
Pavel Veselý: Partially supported by GA ČR project 22-22997S and by Center for Foundations of Modern Computer Science (Charles University project UNCE/SCI/004).

1 Introduction

The shortest superstring problem naturally models a scenario when we have a set of overlapping strings which we need to represent in a compressed form. However, unlike in typical lossless data compression such as Lempel-Ziv schemes, we would like the input strings to be human-readable in the result. That is, the compressed representation of input strings should be a string over the same alphabet that contains all of the strings as substrings. This viewpoint of superstrings as compressed representations has been the crux of their very recent application for representing k-mers, which are k-long substrings of a genomic sequence [19]. These k-mers are typically highly overlapping and in such cases, the shortest superstring of k-mers has length close to the theoretical minimum of the number of distinct k-mers.

Formally, we define the Shortest Superstring problem (SSP) as follows: For a given set of strings S (over a fixed alphabet), compute a minimum-length common *superstring* for the input strings, i.e., a string that contains any $s \in S$ as a substring. SSP is a classical and well-studied problem mentioned in several algorithmic textbooks, e.g., [25, 18, 9, 5]. SSP is APX-hard (i.e., it is NP-hard to obtain a $(1+\varepsilon)$-approximation for some $\varepsilon > 0$) and remains so even when restricted to binary alphabets or input strings having the same length $r \geq 3$ [24].

Therefore, assuming P $\neq$ NP, the best we can hope for are constant-guarantee approximation algorithms. However, determining the best possible constant guarantee is a long-standing open problem, studied for more than three decades. First, Blum et al. [3] designed an algorithm for which they proved an upper bound of 3 on its approximation ratio. Several papers subsequently obtained better approximations using various algorithms [22, 6, 13, 1, 2, 4, 20, 16] and the currently best approximation guarantee is $\frac{37+\sqrt{57}}{18} \approx 2.475$ [7]. In contrast, the hardness result only rules out a 1.003-approximation [12].

Perhaps the most well-known approximation algorithm for SSP is GREEDY which iteratively merges two strings of maximum overlap until only one string remains (if there are more pairs of strings with maximum overlap, we choose arbitrarily). GREEDY is an appealing choice to implement in practice due to its simplicity and close-to-optimal results in experiments [8, 14, 19]. However, the worst-case behavior of GREEDY is far from understood. Blum et al. [3] showed that GREEDY is 4-approximate, an upper bound which was improved to 3.5 by Kaplan and Shafrir [11] and recently, in our previous work, to $\frac{13+\sqrt{57}}{6} \approx 3.425$ [7]. It is easy to see that GREEDY is at least 2-approximate by considering the input $\{c(ab)^k, (ba)^k, (ab)^k c\}$ for $k \to \infty$ [21]. The *Greedy Conjecture* states that this lower bound is tight [21]. Despite an extensive effort to prove or disprove this, the three works [3, 11, 7] comprise the only improvements to the approximation guarantee of GREEDY since the conjecture was first made.

Our results. We make progress on determining the optimal approximation guarantees of GREEDY and of another, more involved algorithm; the latter one improves the best proven approximation guarantee for SSP. In particular, we show the following theorems.

▶ **Theorem 1.** *The approximation guarantee of GREEDY is at most* $\frac{\sqrt{67}+2}{3} \approx 3.396$.

▶ **Theorem 2.** *An algorithm from the literature that combines GREEDY and a Max-ATSP approximation algorithm (outlined in Appendix A.2) computes a superstring of length at most* $\frac{\sqrt{67}+14}{9} \approx 2.466$ *times the optimal.*

Furthermore, our result implies improved approximation guarantees for two algorithms which are variants of GREEDY established in [3], namely TGREEDY and MGREEDY (outlined in Appendix A.2).

As in previous work, all our improved approximation bounds follow from a better inequality that relates certain overlaps between strings to the cost of the optimal solution.

2 The General Setting and Our Technical Contribution

Preliminaries. The set of input strings is denoted by $S = \{s_1, ..., s_{|S|}\}$. Without loss of generality, it is assumed that no string of S is a substring of another string of S. The *length* of a string s is the number of its characters and we denote it by $|s| \in \mathbb{Z}^+$. The concatenation of two strings s and t is denoted by st. A substring of s starting at character i and ending at character $j \geq i$ of s is denoted by $s[i, j]$.

By $\mathsf{ov}(s,t)$ we denote the maximum overlap to merge a string s to the left of a string $t \neq s$, i.e., the longest suffix of s that is a prefix of t. By $\mathsf{ov}(s,s)$ we denote the maximum self-overlap of string s with itself, which is smaller than $|s|$. By $\mathsf{pref}(s,t)$ we denote the prefix of s that remains after removing the overlap with t; thus, $s = \mathsf{pref}(s,t)\mathsf{ov}(s,t)$ and $|\mathsf{pref}(s,t)| = |s| - |\mathsf{ov}(s,t)|$.

2.1 Overlap Graph, Cycle-Closing Edges, and Overlap Inequalities

The overlap graph G_{ov} plays a central role in SSP approximation, including the analysis of GREEDY. It is a complete directed graph with self-loops in which vertices correspond to the input strings, and the weight of each edge (s,t) equals the overlap length $|\mathsf{ov}(s,t)|$.

Note that the optimal solution OPT for a fixed input corresponds to an optimal (maximum overlap) Hamiltonian path in G_{ov}; however, finding such a path is in general a hard problem. On the other hand, finding an optimal cycle cover CC in G_{ov} can be done efficiently. In particular, in a variant of GREEDY, called MGREEDY, such a cycle cover is produced as a by-product. Observe that the total overlap of edges in CC is only larger than that of the optimal Hamiltonian path OPT; indeed, by adding the edge between the endpoints of OPT, we obtain a Hamiltonian cycle, which is a particular cycle cover (not necessarily optimal).

The GREEDY algorithm can be stated as a heuristic for a Hamiltonian path in G_{ov}: Sort the edges of G_{ov} by their overlap lengths non-increasingly, then go over the sorted list and add the i-th edge e_i to the path unless:

(i) there would be a vertex of indegree or outdegree more than one after adding e_i (that is, edge e_i shares a head node or a tail node with an edge picked in a previous step), or
(ii) e_i closes a cycle.

The crucial difference between GREEDY for computing an approximate superstring and MGREEDY for the optimal cycle cover CC is the condition (ii), not present in the latter, i.e., MGREEDY is defined just by condition (i). Call an edge of CC *cycle-closing* if it is the last edge of its cycle added by MGREEDY to CC (i.e., it has the smallest overlap on the cycle, breaking ties arbitrarily).

To obtain a bound on the approximation guarantee of GREEDY, we intuitively need a suitable upper bound on the total overlap of cycle-closing edges, denoted o (strictly speaking, when analyzing GREEDY we consider only the optimal cycle cover of a certain subset of nodes in G_{ov}, but this does not make a difference for our technical contribution; we explain these details in Appendix A.1). Furthermore, the overlap bound should be in terms of the *length* (and not overlap) of OPT.

This intuition was formalized in [3], who proved that $o \le 3 \cdot n$, where n is the length of the optimal solution OPT. Moreover, they show that such a bound is sufficient for a constant upper bound on the approximation ratio of GREEDY. Later works improved the inequality to $o \le 2.5 \cdot n$ [11] and to $o < 2.425 \cdot n$ [7]. Our technical contribution is to show that $o < 2.396 \cdot n$.

In fact, these overlap inequalities are proven and applied in a stronger form of $o < n + \beta \cdot w$, where w is a lower bound on n. To define w, we associate each edge (s,t) of the overlap graph G_{ov} also with a *length* which equals the prefix length $|\mathsf{pref}(s,t)| = |s| - |\mathsf{ov}(s,t)|$. Then w is the total length of all edges in the optimal cycle cover CC.

2.2 Main technical result

We now state our main technical contribution.

▶ **Theorem 3.** *Let S be any input set of strings, and consider an optimal superstring of length n and an optimal cycle cover CC of length w, computed using MGREEDY. Let o be the sum of overlaps of all cycle-closing edges of CC. Then it holds that*

$$o \le n + \beta \cdot w \qquad \textit{for } \beta = (\sqrt{67} - 4)/3 \approx 1.396$$

The proofs of Theorems 1 and 2 using Theorem 3 are the same as in previous work, but we provide an outline for completeness. In Appendix A.1 we describe how Theorem 3 implies the improved upper bound on the approximation guarantees of GREEDY, using another inequality from Blum et al. [3]. Then, in Appendix A.2, we show how to derive better approximation guarantees for a family of SSP algorithms that are based on a Max-ATSP approximation algorithm; the argument is the same as in previous work (e.g., see [4, 15, 16, 7]).

2.3 Overview of the proof of Theorem 3

We build on our previous work [7], where one of the conceptual contributions was in classifying the cycles of CC into three main types. To define them, for a cycle c of CC we let

- $o(c)$ = the overlap of the cycle-closing edge of c, i.e., the smallest overlap on cycle c, and
- $w(c)$ = the total length of edges on c, i.e., the sum of prefixes of the edges of c.

The classification is done according to the $o(c)/w(c)$ ratio.

▶ **Definition 4.** *For parameter β defined in Theorem 3, a cycle c of CC is*

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

The intuition behind the names is that short cycles contain highly periodic strings (e.g., *abcabcabca*), whereas strings in large cycles are not so periodic (e.g., *abcdeabcd*)

In order to prove that $o \leq n + \beta \cdot w$ for $\beta = (\sqrt{67} - 4)/3$, we will assume, without loss of generality, that CC contains no extra large cycle. This follows by the argument in [7, Section 5.1], though for a different overlap to length ratio threshold between large and extra large cycles (which was suitably chosen to match the upper bound $o \leq n + 1.425w$). For completeness, we repeat the proof in Appendix B.

Our analysis in [7] proceeds by showing two incomparable bounds: one better if large cycles have much larger total length than small cycles, and another one for the other case. Namely, letting w_s be the sum of lengths of all small cycles and w_ℓ be the sum of lengths of large cycles, the first upper bound is

$$o \leq n + w_s + 1.5w_\ell \tag{1}$$

and the second upper bound is

$$o \leq n + w_\ell + \frac{31 + 3 \cdot \sqrt{57}}{14} w_s \approx n + w_\ell + 3.832w_s\ . \tag{2}$$

Using the better of (1) and (2) together with $w = w_s + w_\ell$, it follows that $o \leq n + 1.425w$ (recall that the extra large cycles are not taken into account here).

Our improvement and simplification comes from a better version of the second upper bound. Specifically, we show

$$o \leq n + w_\ell + (\gamma - 1) \cdot w_s \approx n + w_\ell + 2.884w_s\ , \tag{3}$$

where $\gamma = (\sqrt{67} + 19)/7 \approx 3.884$. In [7], the bound was shown by first modifying the input in such a way that the overlap graph G_{ov} has the property that all short cycles in the optimal cycle cover only consist of a single edge that is a self-loop. The analysis is then done utilizing this somewhat simpler cycle cover. However, the modification of the input introduces an additional loss that has to be accounted for in the bound. Our analysis is more direct and works with the original optimal cycle cover, which eliminates the need for the input modification and therefore the additional loss. This brings new technical complications because certain properties no longer hold in these more general cycle covers. Nevertheless, we are able to provide a slightly simpler and more straightforward analysis.

Choice of parameters. To combine the two incomparable bounds, $o \leq n + w_s + 1.5 \cdot w_\ell$ and $o \leq n + (\gamma - 1) \cdot w_s + w_\ell$, we set $\lambda = \frac{1}{2\gamma-3}$. As long as $\gamma \geq 2$, this means $\lambda \in [0, 1]$. We then multiply the first bound by $(1 - \lambda)$ and the second bound by λ and add them together. Using $w_s + w_\ell = w$ we get $o \leq n + (\frac{3}{2} - \frac{1}{4\gamma-6}) \cdot w$. In Theorem 3, we want to show that $o \leq n + \beta \cdot w$ and so if

$$\frac{3}{2} - \frac{1}{4\gamma - 6} \leq \beta \tag{4}$$

we are done. We will also need

$$3 \cdot (\beta - \frac{2}{\gamma - 2}) \geq 1 \text{ (for Lemma 6)} \tag{5}$$

or equivalently

$$\gamma \geq 2 + \frac{6}{3\beta - 1} \text{ (for Lemma 12(b)).} \tag{6}$$

The maximum of these two lower bounds (4) and (5) on β is minimized for $\gamma = (\sqrt{67}+19)/7$ and at this point both bounds are equal to $(\sqrt{67} - 4)/3$, which is our choice for β. Apart from this, we will use a number of further inequalities that hold for this choice of parameters (but are not tight). Namely,

$$\frac{5}{2} + \frac{1}{2(\beta - 1)} \leq \gamma \text{ (for Lemma 12(c))}, \tag{7}$$

$$\beta \geq \frac{\gamma}{\gamma - 1} \text{ (for Lemma 12(d))}, \text{ and} \tag{8}$$

$$\gamma \geq 2 \text{ (for Lemma 12(d))}. \tag{9}$$

3 Analysis

In this section we show our improved second bound $o \leq n + w_\ell + (\gamma - 1) \cdot w_s$, following a similar general strategy as in [7].

3.1 Proof Outline

Consider a directed Hamiltonian cycle CC_0 of maximum total overlap in G_{ov}. This cycle is in particular also a (not necessarily maximum) cycle cover. Therefore, the total overlap of CC_0 must be bounded from above by the total overlap of CC. Our goal is to show something stronger than this: that there is a gap between the total overlap of CC_0 and the total overlap of CC that depends in a specific way on the properties of the cycles in CC. Specifically, let $\mathcal{L}$ and $\mathcal{S}$ denote the sets of large and small cycles in CC, respectively, and let $|\mathsf{CC}_i|$ denote the total overlap of a cycle cover CC_i. Then we want to show that the total overlap $|\mathsf{CC}|$ of CC is by at least

$$\sum_{c \in \mathcal{S}} \Big(o(c) - \gamma \cdot w(c)\Big) + \sum_{c \in \mathcal{L}} (o(c) - 2 \cdot w(c)) \tag{10}$$

larger than the total overlap $|\mathsf{CC}_0|$ of CC_0. Showing this is sufficient to establish $o \leq n + w_\ell + (\gamma - 1) \cdot w_s$ because

$$
\begin{aligned}
n &\geq \sum_{\ell=1}^{|S|} |s_\ell| - |\mathsf{CC}_0| \geq \sum_{\ell=1}^{|S|} |s_\ell| - |\mathsf{CC}| + \sum_{c \in \mathcal{S}} (o(c) - \gamma \cdot w(c)) + \sum_{c \in \mathcal{L}} (o(c) - 2 \cdot w(c)) \\
&\geq \sum_{c \in \mathcal{S}} w(c) + \sum_{c \in \mathcal{L}} w(c) + \sum_{c \in \mathcal{S}} (o(c) - \gamma \cdot w(c)) + \sum_{c \in \mathcal{L}} (o(c) - 2 \cdot w(c)) \\
&= \sum_{c \in \mathcal{S}} (o(c) - (\gamma - 1) \cdot w(c)) + \sum_{c \in \mathcal{L}} (o(c) - w(c)) \\
&= o - (\gamma - 1) \cdot \sum_{c \in \mathcal{S}} w(c) - \sum_{c \in \mathcal{L}} w(c) = o - (\gamma - 1) \cdot w_s - w_\ell \,.
\end{aligned}
$$

Related cycles. Before proceeding to describe how we show (10), we borrow the following definition of *related cycles* from [7] that is useful to improve our final bounds slightly. We note that a simpler version of our proof could still be carried out without this additional concept, but at the cost of a slightly weaker bound.

▶ **Definition 5.** *We define a relation R between cycles as follows. A* small *cycle c of CC is related to a* large *cycle c' of CC if $w(c) \leq (\beta/2 - 1/6) \cdot w(c')$ and there exists an edge e in G_{ov} that has one endpoint in cycle c, the other endpoint in cycle c' and satisfies $|\mathsf{ov}(e)| \geq \beta \cdot w(c')$. In this case, we write $(c, c') \in R$.*

In [7], the following lemma is shown. We use different values for β and γ, but the proof of the lemma only requires that $3 \cdot (\beta - 2/(\gamma - 2)) \geq 1$ and this is still satisfied for our new choice of $\beta = (\sqrt{67} - 4)/3$ and $\gamma = (5 - 3\beta)/(3 - 2\beta)$.

▶ **Lemma 6** (Lemma 7.3 in [7]). *For every large cycle c' of CC, at most two different small cycles of CC are related to c'.*

Transforming cycle cover CC_0 into CC in small steps. We analyze the difference of the total overlap between CC_0 and CC in small steps, gradually changing the Hamiltonian cycle CC_0 into a sequence of cycle covers $\mathsf{CC}_0, \mathsf{CC}_1, \mathsf{CC}_2, \ldots$ until we obtain CC. We modify a cycle cover CC_i by removing two edges $f = (v', v)$ and $f' = (u, u')$ from $\mathsf{CC}_i \setminus \mathsf{CC}$ and replace them with the new edges $e = (u, v)$ and $e' = (v', u')$. The resulting set of edges forms a (not necessarily optimal) cycle cover again. Furthermore, if the edges are chosen such that $e \in \mathsf{CC}$ or $e' \in \mathsf{CC}$ (or both), then the resulting cycle cover is closer to the cycle cover CC in the sense that the cardinality of the symmetric difference of the corresponding edge sets decreases.

For a cycle cover CC_i, let $\mathcal{M}(\mathsf{CC}_i)$ be the set of *small* cycles c in CC for which CC_i contains no edge with one endpoint in c and the other endpoint being a string not in c. We define

$$
\begin{aligned}
\phi(i) = \sum_{c \in \mathcal{M}(\mathsf{CC}_i)} \Bigg(&\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\} - \gamma \cdot w(c) \\
&- \sum_{c' : (c, c') \in R} \left(w(c') - \frac{o(c')}{2} \right) \Bigg) \,.
\end{aligned}
$$

The idea is to perform such edge swaps to obtain a sequence $\mathsf{CC}_0, \mathsf{CC}_1, \mathsf{CC}_2, \ldots, \mathsf{CC}_k = \mathsf{CC}$ of cycle covers, such that each cycle cover CC_i is closer to CC than the previous one CC_{i-1}

and such that $|\mathsf{CC}_i| \geq |\mathsf{CC}_0| + \phi(i)$. Then this implies (10) since

$$\begin{aligned}
|\mathsf{CC}| - |\mathsf{CC}_0| &= |\mathsf{CC}_k| - |\mathsf{CC}_0| \geq \phi(k) \\
&= \sum_{c \in \mathcal{M}(\mathsf{CC})} \Big(\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC} \text{ connects two strings of } c\} - \gamma \cdot w(c) \\
&\qquad - \sum_{c' : (c,c') \in R} \Big(w(c') - \frac{o(c')}{2} \Big) \Big) \\
&= \sum_{c \in \mathcal{S}} \Big(\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC} \text{ connects two strings of } c\} - \gamma \cdot w(c) \\
&\qquad - \sum_{c' : (c,c') \in R} \Big(w(c') - \frac{o(c')}{2} \Big) \Big) \\
&= \sum_{c \in \mathcal{S}} \Big(o(c) - \gamma \cdot w(c) - \sum_{c' : (c,c') \in R} \Big(w(c') - \frac{o(c')}{2} \Big) \Big) \\
&= \sum_{c \in \mathcal{S}} (o(c) - \gamma \cdot w(c)) - \sum_{c \in \mathcal{S}} \sum_{c' : (c,c') \in R} \Big(w(c') - \frac{o(c')}{2} \Big) \\
&\geq \sum_{c \in \mathcal{S}} (o(c) - \gamma \cdot w(c)) - \sum_{c \in \mathcal{L}} (2 \cdot w(c) - o(c)) \ ,
\end{aligned}$$

where the last step follows from Lemma 6 and the fact that for large cycles c', by definition, $2w(c') \geq o(c')$.

We use induction to show that it is possible to construct the desired sequence of cycle covers that satisfies $|\mathsf{CC}_i| \geq |\mathsf{CC}_0| + \phi(i)$. The base case is $i = 0$ and we have $\phi(i) = 0$ because $\mathcal{M}(\mathsf{CC}_0) = \emptyset$. (Strictly speaking, it may happen that $\mathcal{M}(\mathsf{CC}_0) \neq \emptyset$; however, in such a case, the optimal Hamiltonian cycle CC_0 is a small cycle of CC, thus $\mathsf{CC}_0 = \mathsf{CC}$. Moreover, in such a case, (1) implies $o < n + w$.)

In the following, we assume that we have a cycle cover CC_i with $|\mathsf{CC}_i| \geq |\mathsf{CC}_0| + \phi(i)$ and we show how to construct CC_{i+1} such that $|\mathsf{CC}_{i+1}| \geq |\mathsf{CC}_0| + \phi(i+1)$ and such that the symmetric difference between CC_{i+1} and CC is smaller than the symmetric difference between CC_i and CC. Specifically, we will identify a swap of four edges as described above to obtain CC_{i+1} from CC_i such that:

- one of the edges that are swapped in belongs to CC, which implies that the symmetric difference between CC_{i+1} and CC will decrease, and
- $|\mathsf{CC}_{i+1}| - |\mathsf{CC}_i| \geq \phi(i+1) - \phi(i)$.

This proves the claim due to the induction hypothesis.

3.2 Important Lemmas

We begin with the following bound on the overlap between two strings from different cycles of CC.

▶ **Lemma 7** (Lemma 9 in [3]). *Let c and $c' \neq c$ be two cycles in CC. It holds that $|\mathsf{ov}(s, s')| < w(c) + w(c')$ for any two strings $s \in c$ and $s' \in c'$.*

When changing cycle cover CC_i into CC_{i+1}, we identify an edge $e = (u, v) \in \mathsf{CC} \setminus \mathsf{CC}_i$ that we add into CC_{i+1}. This triggers removal of edges $f = (v', v)$ and $f' = (u, u')$ from CC_i and addition of one more edge $e' = (v', u')$ that does not belong to CC_i but may or may not be in CC; see Figure 1. In the following, we provide several lower bounds on

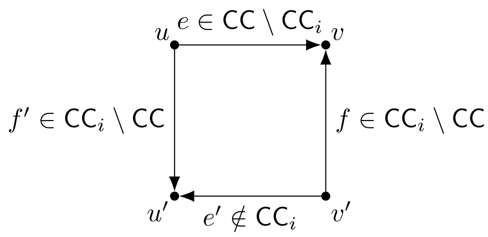

Figure 1 Illustration of the notation used in lemmas in Section 3.2.

$|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')|$, which is the total overlap length difference between CC_i and CC_{i+1}. The first lemma is the well-known *Monge Condition.*

▶ **Lemma 8** (Lemma 7 in [3])**.** *Let $e = (u, v)$, $f = (v', v)$, $f' = (u, u')$, $e' = (v', u')$ be edges in G_{ov}, 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 lemma is shown in [7, Lemma 7.5] for the special case of inputs where each small cycle of CC consists of one string. Below, we generalize it for any input and cycle.

▶ **Lemma 9.** *Let $e = (u, v)$, $f = (v', v)$, $f' = (u, u')$, and $e' = (v', u')$ be edges in G_{ov} such that e is an edge in cycle c in CC. Then,*

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

Before proving Lemma 9, we recall a few definitions from the literature. Consider a cycle c of CC having k nodes $s_1, s_2, \ldots, s_k$. Assuming that the cycle-closing edge of c is (s_k, s_1), we define $s(c)$ as the string $\mathsf{pref}(s_1, s_2)\mathsf{pref}(s_2, s_3) \ldots \mathsf{pref}(s_k, s_1)$.

A *semi-infinite* string is a string obtained by concatenating an infinite number of finite strings. A semi-infinite string s is *periodic* if $s = ts$ for a non-empty string t, that is, $s = t^\infty$.

A string t is a *factor* of a string s if $s = t^i y$ for an integer $i > 0$, where y is a (possibly empty) prefix y of t. By $\mathsf{factor}(s)$ of s, we denote the shortest factor of s and we define $\mathsf{period}(s) = |\mathsf{factor}(s)|$. Finally, we say that a string s has a *periodicity* of length q for $q \leq |s|$ if s is a prefix of the semi-infinite string x^∞ for some string x of length q.

Next, we need a basic observation.

▶ **Observation 10.** *Let s and t be two strings that are substrings of some string z. Then, $|ov(s, t)| > \min\{|s|, |t|\} - period(z)$.*

Proof. We can assume without loss of generality (w.l.o.g.) that $|s| \leq |t|$. This is because, otherwise, let s_R, t_R, and z_R be the reverse of the strings s, t, and z, respectively. We observe that $\mathsf{ov}(t_R, s_R) = \mathsf{ov}(s, t)$ and $\mathsf{period}(z_R) = \mathsf{period}(z)$. Clearly also $|s_R| = |s|$, $|t_R| = |t|$. Therefore, the inequality in the statement of the observation is equivalent to $|\mathsf{ov}(t_R, s_R)| > \min\{|s_R|, |t_R|\} - \mathsf{period}(z_R)$. Hence, if $|s| > |t|$ then $|t_R| \leq |s_R|$ and we can apply the arguments below to the strings t_R, s_R, and z_R instead of s, t, and z (in this order).

Since s and t are substrings of z we can write them as $s = z[i, i + |s| - 1]$ and $t = z[j, j+|t|-1]$ for some i and j. Because of the period of z, we can assume that $i \in [1, \mathsf{period}(z)]$ and $j \in [1, \mathsf{period}(z)]$.

- If $j \geq i$, we have $\mathsf{ov}(s, t) = z[j, i+|s|-1]$ and hence $|\mathsf{ov}(s, t)| = i-j+|s| > |s|-\mathsf{period}(z)$.

- If $j < i$ and $j + \mathsf{period}(z) > |z|$, then $j + \mathsf{period}(z) > |z| \geq i + |s| - 1$ and hence, $|\mathsf{ov}(s,t)| \geq 0 > j - i \geq |s| - \mathsf{period}(z)$.
- If $j < i$ and $j + \mathsf{period}(z) \leq |z|$, we observe that $t = z[j, j+|t|-1]$ also has $z[j + \mathsf{period}(z), \min\{j+|t|-1+\mathsf{period}(z), |z|\}]$ as a prefix (indeed, if $j+|t|-1+\mathsf{period}(z) \leq |z|$ this is not just a prefix of t, but exactly t). Since $i \leq j + \mathsf{period}(z)$ and $|s| \leq |t|$, we have $\mathsf{ov}(s,t) = z[j + \mathsf{period}(z), i + |s| - 1]$ and hence, $|\mathsf{ov}(s,t)| = i - j + |s| - \mathsf{period}(z) > |s| - \mathsf{period}(z)$. ◀

Proof of Lemma 9. Since $\mathsf{ov}(f)$ and $\mathsf{ov}(f')$ are substrings of $s(c)^\infty$, we use Observation 10 to get

$$\begin{aligned}|\mathsf{ov}(e')| &\geq |\mathsf{ov}(\mathsf{ov}(f), \mathsf{ov}(f'))| \\ &> \min\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} - \mathsf{period}(s(c)^\infty) \geq \min\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} - w(c)\,.\end{aligned}$$

It follows that

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

◀

The following lemma is, also, due to [7]. Here, we state it in a slightly different way, but the proof is essentially the same and included in Appendix C for completeness.

▶ **Lemma 11.** *Consider the edges $e = (u,v)$, $f = (v',v)$, $f' = (u,u')$, and $e' = (v',u')$ between (not necessarily different) nodes u, u', v, v' in G_{ov}. Suppose u' and v' are strings in the same cycle c' of CC and that whichever of f or f' has larger overlap connects a string from cycle c and a string from cycle $c' \neq c$ (if $|\mathsf{ov}(f)| = |\mathsf{ov}(f')|$ then it is sufficient if one of them satisfies this). If $|\mathsf{ov}(e)| \geq w(c) + w(c')$, then*

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

The following lemma draws conclusions from the previous ones in a way that will be useful later for our analysis.

▶ **Lemma 12.** *Consider the edges $e = (u,v)$, $f = (v',v)$, $f' = (u,u')$, and $e' = (v',u')$ between (not necessarily different) nodes u, u', v, v' in G_{ov}. Suppose e is an edge in a cycle c of CC. Suppose further that $|\mathsf{ov}(e)| \geq \max\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\}$ and the edge of f and f' that has larger overlap connects a string of cycle c and a string of cycle $c' \neq c$ (if $|\mathsf{ov}(f)| = |\mathsf{ov}(f')|$, then either one of f and f' may satisfy this condition). All of the following statements hold:*

(a) $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq 0$.
(b) *If $w(c) \geq (\beta/2 - 1/6) \cdot w(c')$, then $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq |\mathsf{ov}(e)| - \gamma w(c)$.*
(c) *If $w(c) \geq (\beta - 1) \cdot w(c')$, then $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq |\mathsf{ov}(e)| - \gamma w(c) - w(c')/2 + w(c)/2$.*
(d) *Furthermore, if v' and u' are strings in the same cycle in CC, then also $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq \max\{|\mathsf{ov}(e')| - \gamma w(c'), |\mathsf{ov}(e)| - \gamma w(c) + |\mathsf{ov}(e')| - \gamma w(c')\}$.*

Proof. We show the relevant lower bounds on $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')|$ separately.

(a) Due to Lemma 8, we have $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq 0$.

(b) If $w(c) \geq (\beta/2 - 1/6) \cdot w(c')$, due to Lemma 9, we have

$$\begin{aligned}|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| &\geq |\mathsf{ov}(e)| - \max\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} - w(c) \\ &\geq |\mathsf{ov}(e)| - 2w(c) - w(c') \geq |\mathsf{ov}(e)| - \gamma w(c)\,,\end{aligned}$$

where the second step uses Lemma 7 and the last inequality follows from $2+6/(3\beta-1) = \gamma$.

(c) If $w(c) \geq (\beta - 1) \cdot w(c')$, we have due to Lemma 9 that

$$\begin{aligned}|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| &\geq |\mathsf{ov}(e)| - \max\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} - w(c) \\ &\geq |\mathsf{ov}(e)| - 2w(c) - w(c') \\ &= |\mathsf{ov}(e)| - \frac{5}{2}w(c) - w(c')/2 - w(c')/2 + w(c)/2 \\ &\geq |\mathsf{ov}(e)| - \gamma w(c) - w(c')/2 + w(c)/2\,,\end{aligned}$$

where the second step uses Lemma 7 and the last inequality follows from $5/2 + 1/(2(\beta - 1)) \leq \gamma$.

(d)
- Suppose v' and u' are strings in the same cycle in CC. If $|\mathsf{ov}(e)| \geq w(c)+w(c')$, we apply Lemma 11 to get $|\mathsf{ov}(e)|+|\mathsf{ov}(e')|-|\mathsf{ov}(f)|-|\mathsf{ov}(f')| \geq |\mathsf{ov}(e')|-w(c') \geq |\mathsf{ov}(e')|-\gamma w(c')$. Otherwise, we have $|\mathsf{ov}(e)| < w(c) + w(c')$ and hence,

$$\begin{aligned}|\mathsf{ov}(f)| \leq w(c) + w(c') &= w(c') + \gamma w(c) - (\gamma - 1)w(c) \\ &\leq w(c') + (\gamma - 1)o(c) - (\gamma - 1)w(c) \\ &\leq w(c') + (\gamma - 1)|\mathsf{ov}(e)| - (\gamma - 1)w(c) < \gamma w(c')\,,\end{aligned}$$

since it holds $\beta \geq \frac{\gamma}{\gamma-1}$ and $o(c) > \beta w(c)$ for any large or small cycle c (recall that we assume that CC contains no extra large cycle). We get $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| \geq |\mathsf{ov}(e')| - |\mathsf{ov}(f)| \geq |\mathsf{ov}(e')| - \gamma w(c')$.
- Suppose v' and u' are strings in the same cycle in CC. Due to Lemma 7,

$$\begin{aligned}|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(f)| - |\mathsf{ov}(f')| &\geq |\mathsf{ov}(e)| + |\mathsf{ov}(e')| - 2\max\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} \\ &\geq |\mathsf{ov}(e)| - 2w(c) + |\mathsf{ov}(e')| - 2w(c') \\ &\geq |\mathsf{ov}(e)| - \gamma w(c) + |\mathsf{ov}(e')| - \gamma w(c')\,.\end{aligned}$$

◀

3.3 The Induction Step

We specify how an edge swap is made at a fixed step i in which we obtain cycle cover CC_{i+1} from CC_i. We start by identifying the largest-overlap edge $m = (v_t, w_h)$ in $\mathsf{CC}_i \setminus \mathsf{CC}$, breaking ties arbitrarily. Six further edges will be important. First, let $e_h = (v_h, w_h)$ and $e_t = (v_t, w_t)$ be the edges in CC that share heads and tails with m, respectively. Further, let $f_1' = (v_x, w_t)$ and $f_2' = (v_h, w_x)$ be the two edges in $\mathsf{CC}_i \setminus \mathsf{CC}$ that share heads with e_t and tails with e_h, respectively. Lastly, define $e_1' = (v_x, w_h)$ and $e_2' = (v_t, w_x)$. See Figure 2 for a summary of this notation. It is important to note that the six strings v_h, w_h, v_x, w_x, v_t, and w_t are not necessarily different.

With this, we can define two potential edge swaps. In the first one, we add e_t and e_1' to the cycle cover and instead remove m and f_1'. In the second one, we add e_h and e_2' to the cycle over and instead remove m and f_2'. Which one of these two swaps we will perform depends on a few properties of the edges involved. First of all, we assume that $|\mathsf{ov}(e_h)| \geq |\mathsf{ov}(e_t)|$.

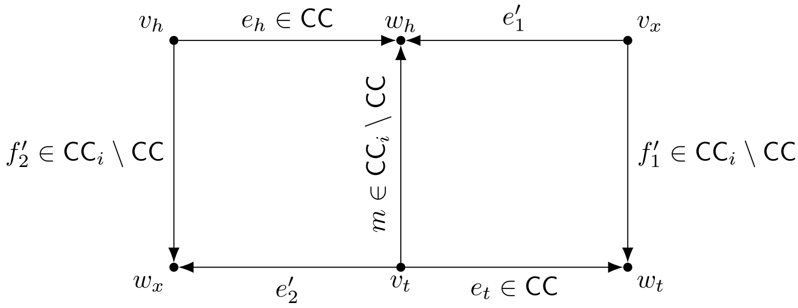

Figure 2 Illustration of the notation. Note that we also allow nodes to be equal to one another here, e.g., it could be that $w_t = w_x$, in which case $e_t = e'_2$, $v_h = v_x$, $e_h = e'_1$, and $f'_1 = f'_2$.

Otherwise, all the remaining arguments follow symmetrically by considering e_t instead of e_h and vice versa. Furthermore, we have that

$$|\mathsf{ov}(e_h)| \geq |\mathsf{ov}(m)|\,, \tag{11}$$

since otherwise $|\mathsf{ov}(m)| > |\mathsf{ov}(e_h)| \geq |\mathsf{ov}(e_t)|$ and m would be added to CC by the greedy algorithm for the optimal cycle cover before e_h and e_t, contradicting the choice of m as an edge of largest overlap in $\mathsf{CC}_i \setminus \mathsf{CC}$.

We observe that there are two reasons why $\phi(i+1)$ may be larger than $\phi(i)$.

- The first potential reason is a difference between the sets $\mathcal{M}(\mathsf{CC}_{i+1})$ and $\mathcal{M}(\mathsf{CC}_i)$. We know that $\mathcal{M}(\mathsf{CC}_{i+1}) \supseteq \mathcal{M}(\mathsf{CC}_i)$, because if a cycle c is in $\mathcal{M}(\mathsf{CC}_i)$, then there is no edge in CC_i connecting a string of c to a string of another cycle. That means that the edges f and f' that we remove from CC_i in the process of constructing CC_{i+1} either have both their endpoints in c or both their endpoints not in c. If both endpoints of both edges f and f' are part of c, then also the two edges that are swapped in to obtain CC_{i+1} from CC_i have their endpoints entirely in c. Therefore, c would still be in $\mathcal{M}(\mathsf{CC}_{i+1})$ after the swap. If both endpoints of both edges f and f' are outside of c, then also the two edges that are swapped in to obtain CC_{i+1} from CC_i have their endpoints entirely outside of c. Again, c would still be in $\mathcal{M}(\mathsf{CC}_{i+1})$ after the swap in this case. Finally, if one of f and f' has both endpoints in c and the other one has both endpoints outside of c, then the two edges that are swapped in both have one endpoint in c and the other endpoint outside of c. However, this is not possible because one of the edges we swap in is e_h or e_t and must therefore be part of the optimal cycle cover CC.

  We can further observe that $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$ must either be equal to $\emptyset$, $\{c\}$, $\{c'\}$, or $\{c, c'\}$, where c and c' are the cycles that e_h and e_t belong to in CC, respectively. (It is possible that $c = c'$.) To see this, observe that one edge being swapped out to obtain CC_{i+1} from CC_i is m and that m has one endpoint (w_h) in c and the other endpoint (v_t) in c'. However, for each cycle of CC, it is clear from a parity argument that the number of edges of CC_i connecting the cycle to other cycles must be even. Hence, for a cycle c'' to be in $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, each of the edges being swapped out must have a string from cycle c'' as an endpoint. This can only be true for c or c' and not for any other cycle.

  Overall, if this reason for the difference between $\phi(i+1)$ and $\phi(i)$ applies, we have that

$$\phi(i+1) - \phi(i) = \sum_{c \in \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)} \Big(\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\} - \gamma \cdot w(c)$$
$$- \sum_{c' : (c,c') \in R} \Big(w(c') - \frac{o(c')}{2} \Big) \Big) .$$

- The second potential reason why $\phi(i+1)$ may be larger than $\phi(i)$ is that for a cycle $c \in \mathcal{M}(\mathsf{CC}_i)$ the term $\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\}$ could change. However, this can only happen if $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \emptyset$ and, furthermore, it can only happen for a cycle c when both edges f and f' that are swapped out have both their endpoints in cycle c. In this case, all four strings involved in the swap (either v_h, w_x, w_h, and v_t or v_x, w_t, w_h, and v_t), must be part of the same cycle in CC. If the value $\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_{i+1} \text{ connects two strings of } c\}$ is larger than the value $\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\}$, then an edge in $\arg\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\}$ must have been swapped out. This means, that if f and f' are the edges being swapped out to obtain CC_{i+1} from CC_i, then $\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_i \text{ connects two strings of } c\} = \min\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\}$. If e and e' are the two edges being swapped in, the new value of $\min\{|\mathsf{ov}(\hat{e})| \mid \hat{e} \in \mathsf{CC}_{i+1} \text{ connects two strings of } c\}$ can be at most $\min\{|\mathsf{ov}(e)|, |\mathsf{ov}(e')|\}$ because e and e' are in CC_{i+1} and satisfy the condition that they connect two strings of c. So overall, in this situation,

$$\phi(i+1) - \phi(i) \leq \min\{|\mathsf{ov}(e)|, |\mathsf{ov}(e')|\} - \min\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} .$$

In summary, we note that only one of the two reasons can apply for any fixed step i. If there is an increase of $\phi(i+1)$ over $\phi(i)$ due to the first reason (a change in the set $\mathcal{M}(\mathsf{CC}_{i+1})$ compared to $\mathcal{M}(\mathsf{CC}_i)$), then there is no increase due to the second reason and vice versa.

We are now ready to complete the proof by showing how to select one of the two identified swap operations such that the total overlap increases by at least $\phi(i+1) - \phi(i)$.

- If m connects two strings of the same cycle in CC, then observe that $\mathcal{M}(\mathsf{CC}_{i+1}) = \mathcal{M}(\mathsf{CC}_i)$. We swap in e_h and e_2' and swap out f_2' and m. Since $|\mathsf{ov}(e_h)| \geq |\mathsf{ov}(m)|$ by (11), we can apply Lemma 8 and establish that the total overlap does not decrease when this swap is performed.

  Furthermore, if v_h, w_h, v_t, and w_x all belong to the same cycle of CC, then the total overlap increases by $|\mathsf{ov}(e_h)| + |\mathsf{ov}(e_2')| - |\mathsf{ov}(f_2')| - |\mathsf{ov}(m)| \geq |\mathsf{ov}(e_2')| - |\mathsf{ov}(f_2')| \geq \min\{|\mathsf{ov}(e_h)|, |\mathsf{ov}(e_2')|\} - \min\{|\mathsf{ov}(f_2')|, |\mathsf{ov}(m)|\}$, where the second inequality uses $|\mathsf{ov}(f_2')| \leq |\mathsf{ov}(m)|$ by the definition of m. This is the only case in which

  $$\min\{|\mathsf{ov}(e)| \mid e \text{ is edge of } \mathsf{CC}_i \text{ connecting two strings of cycle } c\}$$

  can change for a cycle in $c \in \mathcal{M}(\mathsf{CC}_i)$ and the increase is at least $\min\{|\mathsf{ov}(e_h)|, |\mathsf{ov}(e_2')|\} - \min\{|\mathsf{ov}(f_2')|, |\mathsf{ov}(m)|\} \geq \phi(i+1) - \phi(i)$, as required.
- If m connects strings of two different cycles in CC and $|\mathsf{ov}(e_t)| \geq |\mathsf{ov}(m)|$. Let c be the cycle of e_h and c' be the cycle of e_t. If $w(c) \geq w(c')$, we swap in $e = e_h$ and $e' = e_2'$ and swap out $f' = f_2'$ and m. Otherwise, we swap in $e = e_t$ and $e' = e_1'$ and swap out $f' = f_1'$ and m.

  We distinguish between these two cases:
  - Suppose $w(c) \geq w(c')$.

  Then, if $c' \in \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, Lemma 12(d) applies and we know that the increase in total overlap due to the swap is $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(m)| - |\mathsf{ov}(f')| \geq \max\{|\mathsf{ov}(e')| - \gamma w(c'), |\mathsf{ov}(e)| - \gamma w(c) + |\mathsf{ov}(e')| - \gamma w(c')\} \geq \phi(i+1) - \phi(i)$, as required since $\phi(i+1) - \phi(i)$ is either equal to $|\mathsf{ov}(e')| - \gamma w(c')$ or equal to $|\mathsf{ov}(e)| - \gamma w(c) + |\mathsf{ov}(e')| - \gamma w(c')$ depending on whether $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \{c'\}$ or $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \{c', c\}$.
  Otherwise, if $c' \notin \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, Lemma 12(a) and (b) both apply and we know that the increase in total overlap due to the swap is $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(m)| - |\mathsf{ov}(f')| \geq \max\{0, |\mathsf{ov}(e)| - \gamma w(c)\} \geq \phi(i+1) - \phi(i)$, as required since $\phi(i+1) - \phi(i)$ is either equal to 0 or equal to $|\mathsf{ov}(e)| - \gamma w(c)$ depending on whether $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \emptyset$ or $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \{c\}$.
- Suppose $w(c) < w(c')$.
  Then, the same argument as above holds with the only difference being that the roles of e and e' and of c and c' are reversed. Specifically, if $c \in \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, Lemma 12(d) applies with the roles of e and e' and the roles of c and c' reversed. It follows that the increase in total overlap due to the swap is $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(m)| - |\mathsf{ov}(f')| \geq \max\{|\mathsf{ov}(e)| - \gamma w(c), |\mathsf{ov}(e')| - \gamma w(c') + |\mathsf{ov}(e)| - \gamma w(c)\} \geq \phi(i+1) - \phi(i)$, as required.
  Otherwise, if $c \notin \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, Lemma 12(a) and (b) both apply (again with the roles of e and e' and c and c' reversed) and we know that the increase in total overlap due to the swap is $|\mathsf{ov}(e)| + |\mathsf{ov}(e')| - |\mathsf{ov}(m)| - |\mathsf{ov}(f')| \geq \max\{0, |\mathsf{ov}(e')| - \gamma w(c')\} \geq \phi(i+1) - \phi(i)$, as required.

- If m connects strings of two different cycles in CC and $|\mathsf{ov}(e_t)| < |\mathsf{ov}(m)|$, then we swap in e_h and e'_2 and swap out f'_2 and m. Let c be the cycle of e_h and c' be the cycle of e_t.
  - If $\mathcal{M}(\mathsf{CC}_{i+1}) = \mathcal{M}(\mathsf{CC}_i)$, then Lemma 12(a) shows that the total overlap does not decrease, while the potential $\phi(i)$ does not increase.
  - If $c' \in \mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i)$, then w_x and v_t must both be strings in cycle c' as otherwise, v' is a string of cycle c' and w_x is a string of a different cycle and thus e'_2, which is an edge in CC_{i+1}, would connect a string of cycle c' to a string of another cycle. Thus, by Lemma 12(d), $|\mathsf{ov}(e_h)| + |\mathsf{ov}(e'_2)| - |\mathsf{ov}(m)| - |\mathsf{ov}(f'_2)| \geq \max\{|\mathsf{ov}(e'_2)| - \gamma w(c'), |\mathsf{ov}(e_h)| - \gamma w(c) + |\mathsf{ov}(e'_2)| - \gamma w(c')\} \geq \phi(i+1) - \phi(i)$, as required.
  - If $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \{c\}$ and $(c, c') \in R$, we first observe

$$w(c) \geq |\mathsf{ov}(m)| - w(c') > |\mathsf{ov}(e_t)| - w(c') \geq o(c') - w(c') \geq (\beta - 1) \cdot w(c') ,$$

    where the third inequality follows from the fact that e_t is an edge of the cycle c' and the last step follows because c' is not extra large. Therefore, we can apply Lemma 12(c) which is sufficient because $w(c)/2 - w(c')/2 = w(c)/2 + w(c')/2 - w(c') \geq |\mathsf{ov}(m)|/2 - w(c') \geq o(c')/2 - w(c')$ and therefore, $|\mathsf{ov}(e_h)| + |\mathsf{ov}(e'_2)| - |\mathsf{ov}(m)| - |\mathsf{ov}(f'_2)| \geq |\mathsf{ov}(e)| - \gamma w(c) - w(c')/2 + w(c)/2 \geq |\mathsf{ov}(e)| - \gamma w(c) - w(c') + o(c')/2 \geq \phi(i+1) - \phi(i)$, as required.
  - If $\mathcal{M}(\mathsf{CC}_{i+1}) \setminus \mathcal{M}(\mathsf{CC}_i) = \{c\}$ and $(c, c') \notin R$, there are two possibilities.
    1. If c' is a small cycle, then $w(c') \leq o(c') - w(c') \leq |\mathsf{ov}(e_t)| - w(c') < |\mathsf{ov}(m)| - w(c') \leq w(c)$, where the first step uses the definition of a small cycle and the last step uses Lemma 7.
    2. If c' is a large cycle and $(c, c') \notin R$, then, because $|\mathsf{ov}(m)| > |\mathsf{ov}(e_t)| \geq \beta w(c')$ by the definition of related cycles, $w(c) > (\beta/2 - 1/6) \cdot w(c')$.

    Either way $w(c) > (\beta/2 - 1/6) \cdot w(c')$, which means that Lemma 12(b) implies $|\mathsf{ov}(e_h)| + |\mathsf{ov}(e'_2)| - |\mathsf{ov}(m)| - |\mathsf{ov}(f'_2)| \geq |\mathsf{ov}(e_h)| - \gamma w(c) \geq \phi(i+1) - \phi(i)$, as required.

References

1. Chris Armen and Clifford Stein. Improved length bounds for the shortest superstring problem. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures (WADS)*, pages 494–505, 1995. doi:10.1007/3-540-60220-8_88.
2. Chris Armen and Clifford Stein. A 2 2/3 superstring approximation algorithm. *Discret. Appl. Math.*, 88(1-3):29–57, 1998. doi:10.1016/S0166-218X(98)00065-1.
3. Avrim Blum, Tao Jiang, Ming Li, John Tromp, and Mihalis Yannakakis. Linear approximation of shortest superstrings. *Journal of the ACM*, 41(4):630–647, 1994. doi:10.1145/179812.179818.
4. Dany Breslauer, Tao Jiang, and Zhigen Jiang. Rotations of periodic strings and short superstrings. *J. Algorithms*, 24(2):340–353, 1997. doi:10.1006/jagm.1997.0861.
5. M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.
6. Artur Czumaj, Leszek Gasieniec, Marek Piotrów, and Wojciech Rytter. Sequential and parallel approximation of shortest superstrings. *J. Algorithms*, 23(1):74–100, 1997. doi:10.1006/jagm.1996.0823.
7. Matthias Englert, Nicolaos Matsakis, and Pavel Veselý. Improved approximation guarantees for shortest superstrings using cycle classification by overlap to length ratios. In *Proceedings of the 54th ACM Symposium on Theory of Computing (STOC)*, pages 317–330. ACM, 2022. doi:10.1145/3519935.3520001.
8. Alan M. Frieze and Wojciech Szpankowski. Greedy algorithms for the shortest common superstring that are asymptotically optimal. *Algorithmica*, 21(1):21–36, 1998.
9. Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997. doi:10.1017/CBO9780511574931.
10. Haim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *Journal of the ACM*, 52(4):602–626, 2005. doi:10.1145/1082036.1082041.
11. Haim Kaplan and Nira Shafrir. The greedy algorithm for shortest superstrings. *Inf. Process. Lett.*, 93(1):13–17, 2005. doi:10.1016/j.ipl.2004.09.012.
12. Marek Karpinski and Richard Schmied. Improved inapproximability results for the shortest superstring and related problems. In *Proceedings of the 19th Computing: The Australasian Theory Symposium (CATS)*, pages 27–36, 2013.
13. S. Rao Kosaraju, James K. Park, and Clifford Stein. Long tours and short superstrings. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 166–177, 1994. doi:10.1109/SFCS.1994.365696.
14. Bin Ma. Why greed works for shortest common superstring problem. *Theor. Comput. Sci.*, 410(51):5374–5381, 2009.
15. Marcin Mucha. A tutorial on shortest superstring approximation. https://www.mimuw.edu.pl/~mucha/teaching/aa2008/ss.pdf, 2007. [Accessed 15-June-2023].
16. Marcin Mucha. Lyndon words and short superstrings. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 958–972, 2013. doi:10.1137/1.9781611973105.69.
17. Katarzyna Paluch, Khaled Elbassioni, and Anke van Zuylen. Simpler approximation of the maximum asymmetric traveling salesman problem. In *Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 501–506, 2012. doi:10.4230/LIPIcs.STACS.2012.501.
18. Steven Skiena. *The Algorithm Design Manual, Third Edition*. Texts in Computer Science. Springer, 2020.
19. Ondřej Sladký, Pavel Veselý, and Karel Břinda. Masked superstrings as a unified framework for textual k-mer set representations. *bioRxiv*, 2023. URL: https://www.biorxiv.org/content/early/2023/02/03/2023.02.01.526717, doi:10.1101/2023.02.01.526717.
20. Z. Sweedyk. A 2½-approximation algorithm for shortest superstring. *SIAM J. Comput.*, 29(3):954–986, 1999. doi:10.1137/S0097539796324661.

21 Jorma Tarhio and Esko Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theor. Comput. Sci.*, 57:131–145, 1988. doi:10.1016/0304-3975(88)90167-3.

22 Shang-Hua Teng and Frances Yao. Approximating shortest superstrings. *SIAM Journal on Computing*, 26(2):410–417, 1997. doi:10.1137/S0097539794286125.

23 Jonathan S. Turner. Approximation algorithms for the shortest common superstring problem. *Inf. Comput.*, 83(1):1–20, 1989. doi:10.1016/0890-5401(89)90044-8.

24 Virginia Vassilevska. Explicit inapproximability bounds for the shortest superstring problem. In *30th International Symposium, MFCS, Gdansk, Poland*, volume 3618 of *Lecture Notes in Computer Science*, pages 793–800. Springer, 2005.

25 Vijay Vazirani. *Approximation algorithms.* Springer, 2001.

A Deriving Approximation Guarantees from Theorem 3

The technical contribution of the paper is proving Theorem 3 that shows an improved inequality for overlaps of cycle-closing edges in terms of the optimal superstring length n and the length w of the optimal cycle cover CC. In the next two subsections, we explain how our improved approximation guarantees follow, using essentially the same arguments (and algorithms) as in previous work.

A.1 The GREEDY Algorithm for SSP

The $|S|^2$ edges of the overlap graph G_{ov} are assumed to be ordered by non-increasing overlap length. The GREEDY algorithm for SSP chooses edges from this order, unless an edge shares an endpoint with an already chosen edge or closes a cycle. The edges corresponding to the latter case are called *bad back edges*. As proven in [3], bad back edges do not intersect each other, forming a laminar family of edges. Each inner-most bad back edge forms a cycle in the output of GREEDY and each such cycle is called *culprit*. The sum of lengths of all culprit cycles is denoted by w_c and the sum of overlap lengths of the cycle-closing edges of all culprits is denoted by o_c.

Blum et al. have shown the following two inequalities (Section 5 in [3]):

$$|\mathsf{GREEDY}(S)| \leq 2n + o_c - w_c \tag{12}$$

$$o \leq n + 2w \tag{13}$$

Moreover, the application of the GREEDY algorithm for the optimal cycle cover CC on the set of strings comprising the culprit cycles only, outputs the exact same set of culprit cycles (Lemma 15 in [3]). By this and (13) it follows that $o_c \leq n + 2w_c$, which by (12) gives $|\mathsf{GREEDY}(S)| \leq 4n$, completing their proof.

Theorem 3 shows that $o \leq n + \frac{\sqrt{67}-4}{3} w$ which implies that $o_c \leq n + \frac{\sqrt{67}-4}{3} w_c$ using the same syllogism (Lemma 15 in [3]). By this and (12), we have $|\mathsf{GREEDY}(S)| \leq \frac{\sqrt{67}+2}{3} n \approx 3.396 \cdot n$, completing our proof.

A.2 SSP Algorithms Based on Max-ATSP Approximations

Blum et al. proposed the following 4-approximate SSP algorithm, called MGREEDY:

1. Apply GREEDY to find an optimal cycle cover CC.
2. Open all cycle-closing edges in CC to obtain a set of strings called *representatives*.
3. Concatenate the representatives in an arbitrary order.

If instead of concatenating the representatives in the third step, we merge them using a Max-ATSP approximation algorithm (executed on the overlap graph of the representatives), then we will obtain an SSP approximation algorithm which, obviously, cannot perform worse. This is the idea behind the 3-approximate TGREEDY algorithm [3]. The Max-ATSP algorithm utilized as a black-box within TGREEDY is GREEDY, which had been already shown [21, 23] to be a $\frac{1}{2}$-approximate Max-ATSP algorithm for the overlap graphs.

We will need the following theorem from [7], which has already appeared in similar forms in literature (e.g., [3, 4, 15]).

▶ **Theorem 13.** *If MGREEDY is a $(2+\zeta)$-approximate SSP algorithm and there exists a δ-approximate algorithm for Max-ATSP then there exists a $(2+(1-\delta)\cdot\zeta)$-approximate SSP algorithm.*

Showing that $o \leq n + (\sqrt{67}-4)w/3 \approx n + 1.396w$ implies that $\mathsf{MGREEDY}$ is a 3.396-approximate SSP algorithm, since $|\mathsf{MGREEDY}(S)| = w + o \leq w + n + (\sqrt{67}-4)w/3 < 3.396n$. Moreover, the currently best Max-ATSP approximation algorithms are $\frac{2}{3}$-approximate, due to Kaplan et al. [10] or due to Paluch et al. [17]. Setting $\delta = \frac{2}{3}$ and $\zeta = (\sqrt{67}-4)/3 \approx 1.396$ in Theorem 13, we obtain an SSP algorithm with approximation guarantee $\frac{\sqrt{67}+14}{9} \approx 2.466$.

Finally, regarding $\mathsf{TGREEDY}$, setting $\delta = \frac{1}{2}$ and $\zeta = (\sqrt{67}-4)/3 \approx 1.396$ in Theorem 13, we improve the approximation guarantee of $\mathsf{TGREEDY}$ to $(\sqrt{67}+8)/6 \approx 2.698$, from $(25+\sqrt{57})/12 \approx 2.712$ as shown in [7].

B Dealing with extra large cycles (as in [7])

Let $\overline{S} \subseteq S$ be the subset of strings that belong to all small and large cycles of CC. Observation 5.1 in [7] implies that the optimal cycle cover for $\bar{S}$ (in short $\mathsf{CC}(\overline{S})$) consists of all small and large cycles of the optimal cycle cover for S (for simplicity denoted by $\mathsf{CC}(S) = \mathsf{CC}$), while the optimal cycle cover for $S - \overline{S}$ (in short $\mathsf{CC}(S - \overline{S})$) consists of all extra large cycles of $\mathsf{CC}(S)$.

Let $\hat{w}$ denote the sum of lengths of the (extra large) cycles in $\mathsf{CC}(S - \overline{S})$ and let $\hat{o}$ be the sum of overlap lengths of the cycle-closing edges of the cycles in $\mathsf{CC}(S - \overline{S})$. Similarly, let $\overline{o}$ be the sum of overlap lengths of the cycle-closing edges in $\mathsf{CC}(\overline{S})$ and let $\overline{w}$ be the sum of lengths of the cycles in $\mathsf{CC}(\overline{S})$.

Proving $o \leq n + \beta \cdot w$ for input $\overline{S}$ implies that $\overline{o} \leq |\mathsf{OPT}(\overline{S})| + \beta \cdot \overline{w}$, and assuming this, we show $o \leq n + \beta \cdot w$. Indeed, we take the sum of inequality $\overline{o} \leq |\mathsf{OPT}(\overline{S})| + \beta \cdot \overline{w}$ with inequality $\hat{o} \leq \beta \cdot \hat{w}$ (which holds by the definition of extra large cycles) and obtain:

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

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

C Lemma 11 (slightly modified from [7])

For completeness, we include a proof of Lemma 11. The proof is almost identical to the one in [7] with only very minor changes to make it more general.

We start by stating a corollary, a version of which is already stated in [7] and in slight variations has been known already before (e.g. see Lemma 9 in [3] and Lemma 7 in [15]).

▶ **Corollary 14.** *Let c and c' be any two cycles of CC. Any string h, which is a substring of both $s(c)^\infty$ and $s(c')^\infty$,*[1] *satisfies $|h| < w(c) + w(c')$.*

This enables us to restate the proof of Lemma 11.

Proof of Lemma 11. We show that $|\mathsf{ov}(e)| > |\mathsf{ov}(f)| + |\mathsf{ov}(f')| - w(c')$, which implies the lemma. If $\min\{|\mathsf{ov}(f)|, |\mathsf{ov}(f')|\} \leq w(c')$, this inequality holds because by using Lemma 7,

[1] The definitions of $s(c)$ and s^∞ appear below Lemma 9.

we get

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

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

Now, assume for contradiction that $|\mathsf{ov}(e)| \leq |\mathsf{ov}(f)| + |\mathsf{ov}(f')| - w(c')$. We claim that in this case $\mathsf{ov}(e)$ has a periodicity of length $w(c')$, i.e., $\mathsf{ov}(e)$ is a prefix of x^∞ for some string x with $|x| = w(c')$. To show this, first recall that $|\mathsf{ov}(e)| \geq w(c)+w(c') > \max\{|\mathsf{ov}(f')|, |\mathsf{ov}(f)|\}$ by Lemma 7. Since $\mathsf{ov}(f)$ is a prefix of v and a suffix of v' and since $\mathsf{ov}(e)$ is a prefix of v, the first $|\mathsf{ov}(f)|$ characters of $\mathsf{ov}(e)$ are also a suffix of v', i.e.,

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

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

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

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

Next, we show that $\mathsf{ov}(e)$ is a substring of the semi-infinite string $s(c')^\infty$. Because $\mathsf{ov}(e)$ 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 $\mathsf{ov}(e)$ are a substring of $s(c')^\infty$. This is indeed the case since $\mathsf{ov}(e)[1, |\mathsf{ov}(f)|]$ is a substring of v' which is a substring of $s(c')^\infty$ and we assume that $|\mathsf{ov}(f)| > w(c')$.

Since $\mathsf{ov}(e)$ is a substring of $s(c')^\infty$ as well as of $s(c)^\infty$ (because $\mathsf{ov}(e)$ is a substring of a string that is part of c), Corollary 14 implies $|\mathsf{ov}(e)| < w(c) + w(c')$ which contradicts the assumption of the lemma. ◀