Relationship between superstring and compression measures: New insights on the greedy conjecture

Bastien Cazaux, Eric Rivals

Discrete Applied Mathematics 245 (2018) 59–64

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Relationship between superstring and compression measures: New insights on the greedy conjecture

Bastien Cazaux, Eric Rivals *

LIRMM, CNRS and Université de Montpellier, 161 rue Ada, 34095 Montpellier Cedex 5, France
Institut Biologie Computationnelle, CNRS and Université de Montpellier, 860 rue Saint Priest, 34095 Montpellier Cedex 5, France

ARTICLE INFO

Article history:
Received 25 December 2015
Received in revised form 31 March 2017
Accepted 19 April 2017
Available online 19 May 2017

Keywords:
Approximation algorithm
Shortest Common Superstring Problem
Stringology
Data compression
Assembly
Greedy conjecture

ABSTRACT

A superstring of a set of words is a string that contains each input word as a substring. Given such a set, the Shortest Superstring Problem (SSP) asks for a superstring of minimum length. SSP is an important theoretical problem related to the Asymmetric Travelling Salesman Problem, and also has practical applications in data compression and in bioinformatics. Indeed, it models the question of assembling a genome from a set of sequencing reads. Unfortunately, SSP is known to be NP-hard even on a binary alphabet and also hard to approximate with respect to the superstring length or to the compression achieved by the superstring. Even the variant in which all words share the same length r, called r-SSP, is NP-hard whenever $r > 2$. Numerous involved approximation algorithms achieve approximation ratio above 2 for the superstring, but remain difficult to implement in practice. In contrast the greedy conjecture asked in 1988 whether a simple greedy algorithm achieves ratio of 2 for SSP. Here, we present a novel approach to bound the superstring approximation ratio with the compression ratio, which, when applied to the greedy algorithm, shows a 2 approximation ratio for 3-SSP, and also that greedy achieves ratios smaller than 2. This leads to a new version of the greedy conjecture.

1. Introduction

Given a set of p words $P := \{s_1, s_2, \ldots, s_p\}$ over a finite alphabet Σ, a superstring of P is a string containing each s_i for $1 \leq i \leq p$ as a substring. The **Shortest Superstring Problem (SSP)** asks for a superstring of P of minimal length. SSP is a well studied problem (alias Shortest Common Superstring), with a strong relation to the Asymmetric Travelling Salesman Problem, and is known to be NP-hard even on a binary alphabet [7]. The restriction to instances where all input strings share the same length, say $r > 1$, is denoted r-SSP, becomes polynomial if $r \leq 2$, but remains NP-hard as soon as the strings are of length at least 3 [1]. Two approximation measures can be optimised for SSP: either the length of the superstring is minimised, or the compression is maximised (i.e., the sum of the lengths of the input strings minus that of the superstring). For a word x, $|x|$ denotes the *length* of x. Let $\|P\|$ denote $\sum_{s_i \in P} |s_i|$ and let t be the output superstring, then the compression equals $\|P\| - |t|$. With both measures SSP is hard to approximate (MAX-SNP-hard, see [1]). Since 1991, a long series of elaborated algorithms have improved the approximation ratio for both measures culminating in $2\frac{11}{23}$ for the superstring [13] and in 3/4 for the compression measure [14]. A recent table listing these ratio and the literature, as well as known inapproximability bounds appears in [9]. A detailed survey gives an overview of the numerous application contexts of SSP [8].

* Corresponding author at: LIRMM, CNRS and Université de Montpellier, 161 rue Ada, 34095 Montpellier Cedex 5, France.
E-mail addresses: bastien.cazaux@lirmm.fr (B. Cazaux), rivals@lirmm.fr (E. Rivals).

http://dx.doi.org/10.1016/j.dam.2017.04.017

In 1988, a seminal paper introduced a simple greedy algorithm, consisting in repeatedly merging two words that exhibit the largest (prefix–suffix) overlap until only one string remains [16]. With $P := \{abba, bbaa, aaba\}$ for example, *abba* is first merged with *bbaa* yielding *abbaa* (they share a 3-letter overlap), then, *abbaa* is merged with *aaba* resulting in the superstring *abbaaba* of length 7; as $\|P\| = 12$, the compression obtained equals $\|P\| - |t| = 12 - 7 = 5$. Note that their greedy algorithm, denoted by GREEDY, can be seen as the greedy algorithm of a specific hereditary system [4]. Tarhio and Ukkonen proved in [16] that GREEDY achieves a compression ratio of $1/2$ and formulated the *greedy conjecture*: the greedy algorithm yields a superstring ratio of 2. Despite a lot of research dedicated to SSP, this conjecture has remained open since 1988. A weaker form of this conjecture asks to prove this ratio for r-SSP and some values of r. Blum et al. have shown for GREEDY a superstring ratio of 4 [1], which was later improved to 3.5 in [10]. The greedy conjecture is supported by simulated experiments [18,15]. Moreover, the superstring approximation ratio obtained by the greedy algorithm remains a crucial question, especially since other approximation algorithms are usually less efficient than GREEDY [10].

Recently, it has been proven that in the case where all input words have length 4 (for 4-SSP) the greedy algorithm achieves a superstring ratio of at most 2, as stated by the conjecture [11]. This proof is valid only for words of length 4 and cannot be adapted to words of length 3, for instance. Kulikov and colleagues [11] suggest that the conjecture for 3-SSP follows from the fact that GREEDY achieves 2-approximation of the compression measure, citing [16]. To our knowledge, no proof for the greedy conjecture for words of length 3 has ever been published and there are no mention of it in a recent survey [8]. Here, we study the relationship between the compression ratio and the superstring ratio of an approximation algorithm in general, and derive a bound of the superstring ratio in function of the compression ratio. When applied to GREEDY on words of fixed length (r-SSP), we obtain a superstring approximation ratio of 2 for 3-SSP, and this ratio increases with r to reach for $r = 6$ a value of $7/2$, which is the best known ratio for the greedy algorithm [10]. But we also get a tight superstring ratio of $3/2$ for 2-SSP, thereby demonstrating that the greedy algorithm can achieve a ratio strictly smaller than 2. This shows first that the general relationship between the superstring and compression measures is important and can serve for future research. Second, the ratio smaller than 2 does not contradict known bounds or instances. Indeed, the known examples give a bound that converges towards 2 from below when the length of the input words tends to infinity. Thus, we propose a more precise conjecture for r-SSP, in which the superstring ratio equals $2 - \frac{1}{r}$ instead of 2.

Notation: An *alphabet* Σ is a finite set of *letters*. A linear *word* or *string* over Σ is a finite sequence of elements of Σ. The set of all finite words over Σ is denoted by $\Sigma^\star$, and Σ^r denotes the subset of $\Sigma^\star$ of words of length r for any positive integer r. Given two words x and y, we denote by xy the *concatenation* of x and y.

2. Relation between maximum compression and shortest superstring approximation ratios for SSP

Here, we exhibit for SSP an upper bound of the superstring approximation ratio of an algorithm in function of its compression ratio.

Let $\mathcal{A}$ be a polynomial-time approximation algorithm for SSP. As all approximation algorithms considered here take polynomial time in the input size, we simply omit this characteristic in the sequel. We denote by $s_{\mathcal{A}}(P)$ the output of algorithm $\mathcal{A}$ with input P, and by $s_{opt}(P)$ an optimal superstring for this input. Note that $s_{opt}(P)$ also achieves a maximum compression for P. We only consider approximation algorithms that return a superstring whose length is bounded by $\|P\|$. In other words, we disregard algorithms that insert additional symbols beyond those required by the words of the instance. Without this restriction, the approximation ratio super($\mathcal{A}$) would not be defined for any algorithm $\mathcal{A}$, and the ratio comp($\mathcal{A}$) could be negative; both ratios are defined a few lines below. Instances where the optimal superstring is the concatenation of all the words of the instance satisfy $|s_{opt}(P)| = \|P\|$. In such cases, for any approximation algorithm $\mathcal{A}$, one has $\|P\| = |s_{opt}(P)| = |s_{\mathcal{A}}(P)| = \|P\|$. Such instances are excluded from Theorem 1. Let us define the superstring approximation ratio of algorithm $\mathcal{A}$, denoted super($\mathcal{A}$), as the smallest real value such that for any input P:

$$1 \leq \frac{|s_{\mathcal{A}}(P)|}{|s_{opt}(P)|} \leq \text{super}(\mathcal{A}).$$

Similarly, we define the compression ratio comp($\mathcal{A}$) as the largest real value such that, for any input P satisfying $\|P\| \neq |s_{opt}(P)|$, we have

$$0 \leq \text{comp}(\mathcal{A}) \leq \frac{\|P\| - |s_{\mathcal{A}}(P)|}{\|P\| - |s_{opt}(P)|}.$$

Instances where the optimal superstring is the concatenation of all the words of the instance satisfy $|s_{opt}(P)| = \|P\|$. In such cases, for any approximation algorithm $\mathcal{A}$ one has $\|P\| = |s_{opt}(P)| = |s_{\mathcal{A}}(P)| = \|P\|$. Such instances are excluded from Theorem 1.

Theorem 1. *Let P be a set of words satisfying $|s_{opt}(P)| \neq \|P\|$. Let γ be a real such that $0 < \gamma \leq \frac{|s_{opt}(P)|}{\|P\|}$, and let $\mathcal{A}$ be an approximation algorithm for SSP. We have:*

$$super(\mathcal{A}) \leq \frac{(\gamma - 1) \times comp(\mathcal{A}) + 1}{\gamma}.$$

Proof. Let $\alpha = \frac{(\gamma-1)\times\text{comp}(\mathcal{A})+1}{\gamma}$ and the function $f : x \mapsto \frac{(x-1)\times\text{comp}(\mathcal{A})+1}{x}$. Its derivative is $f' : x \mapsto \frac{\text{comp}(\mathcal{A})-1}{x^2}$, which is negative since $0 < \text{comp}(\mathcal{A}) \leq 1$. Moreover, f is decreasing, and as $\gamma < 1$, we get $\alpha = f(\gamma) > f(1) = 1$. We obtain that $\gamma = \frac{1-\text{comp}(\mathcal{A})}{\alpha-\text{comp}(\mathcal{A})}$. It follows that:

$$\begin{array}{rrcl} & \gamma \times \|P\| & \leq & |s_{opt}(P)| \\ \Leftrightarrow & \frac{1-\text{comp}(\mathcal{A})}{\alpha-\text{comp}(\mathcal{A})} \times \|P\| & \leq & |s_{opt}(P)| \\ \Leftrightarrow & (1-\text{comp}(\mathcal{A})) \times \|P\| & \leq & (\alpha-\text{comp}(\mathcal{A})) \times |s_{opt}(P)| \\ \Leftrightarrow & \text{comp}(\mathcal{A}) \times |s_{opt}(P)| + (1-\text{comp}(\mathcal{A})) \times \|P\| & \leq & \alpha \times |s_{opt}(P)|. \end{array}$$

By definition $\mathcal{A}$ achieves the compression ratio $\text{comp}(\mathcal{A})$, so using the previous inequality we get

$$\begin{array}{rrcl} & \text{comp}(\mathcal{A}) \times (\|P\| - |s_{opt}(P)|) & \leq & \|P\| - |s_{\mathcal{A}}(P)| \\ \Rightarrow \alpha \times |s_{opt}(P)| \geq & \text{comp}(\mathcal{A}) \times |s_{opt}(P)| + (1-\text{comp}(\mathcal{A})) \times \|P\| & \geq & |s_{\mathcal{A}}(P)| \\ \Rightarrow & \alpha & \geq & \frac{|s_{\mathcal{A}}(P)|}{|s_{opt}(P)|}. \end{array}$$

As for any set P of input words, $\text{super}(\mathcal{A})$ is the smallest value larger than $\frac{|s_{\mathcal{A}}(P)|}{|s_{opt}(P)|}$, and as α does not depend on P, we get:

$$\begin{aligned} \text{super}(\mathcal{A}) &\leq \alpha \\ &= \frac{(\gamma-1)\times\text{comp}(\mathcal{A})+1}{\gamma}. \quad \square \end{aligned}$$

3. Approximation of r-SSP

Let r be an integer satisfying $r > 1$. Now, let us study the superstring approximation for the restriction of SSP to instances in which all input words have the same length r. First we show a theorem bounding the superstring ratio in function of the compression ratio for r-SSP for any algorithm. Then, we derive an upper bound and prove a lower bound for the superstring ratio of the greedy algorithm. Finally, applying this theorem improves the superstring ratio for $r < 6$ compared to the $7/2$ bound of [10], and solves the greedy conjecture for 3-SSP.

Since the instance P is a subset of Σ^r, we have $\|P\| = r \times p$. As all words of P are different, any word differs from the other by at least one symbol and any two words overlap by at most $r-1$ positions, which implies the following property.

Proposition 1. *Let t be a superstring of P. Then $|t| \geq r + p - 1$.*

We derive the following theorem.

Theorem 2. *Let r be an integer such that $r > 1$ and let P be a subset of Σ^r. For any approximation algorithm $\mathcal{A}$, we have:*

$$\frac{|s_{\mathcal{A}}(P)|}{|s_{opt}(P)|} \leq r - (r-1) \times comp(\mathcal{A}).$$

Proof. From Proposition 1, we know that $|s_{opt}(P)| \geq r + p - 1$, which implies

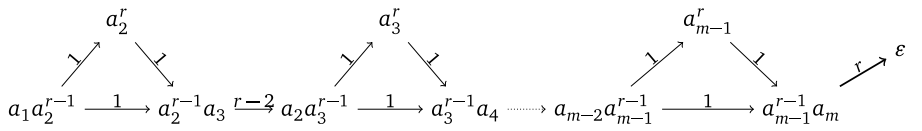
$$\begin{aligned} \frac{|s_{opt}(P)|}{\|P\|} &\geq \frac{r+p-1}{\|P\|} \\ &\geq \frac{p}{r \times p} = \frac{1}{r}. \end{aligned}$$

Using Theorem 1 with $\gamma = 1/r$, we obtain

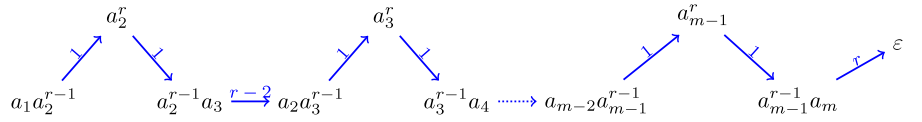
$$\begin{aligned} \frac{|s_{\mathcal{A}}(P)|}{|s_{opt}(P)|} &\leq \frac{\left(\frac{1}{r}-1\right)\times\text{comp}(\mathcal{A})+1}{\frac{1}{r}} \\ &= r \times \left(\left(\frac{1-r}{r}\right) \times \text{comp}(\mathcal{A}) + 1\right) \\ &= r - (r-1) \times \text{comp}(\mathcal{A}). \quad \square \end{aligned}$$

Theorem 2 bounds the ratio of an algorithm $\mathcal{A}$ for any instance of r-SSP. Consequently, $\text{super}(\mathcal{A})$ also satisfies the same inequation.

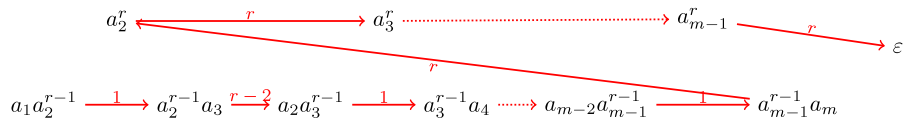
We can now provide a bound on the approximation ratio of the greedy algorithm for r-SSP, knowing that its compression ratio is $1/2$ [16,4,3].



(a) Prefix graph of the instance.



(b) Path of the optimal superstring in the prefix graph.



(c) Path of the greedy superstring in the prefix graph.

Fig. 1. Illustration of the instance considered in the proof of Proposition 2, which gives the lower bound of GREEDY superstring ratio for r-SSP. The prefix graph for this instance is shown in (a), the path corresponding to the optimal solution in (b), and the path of the greedy solution in (c). The prefix graph is complete digraph in which each input word is a node, and the weight of an arc (x, y) equals the length of x minus the length of the overlap between x and y.

Proposition 2. GREEDY *approximates r-SSP with a ratio of at least* $2 - \frac{1}{r}$.

Proof. Theorem 2 gives an upper bound on the approximation ratio of GREEDY. To obtain the desired lower bound, we exhibit an instance where $\frac{|s_{\text{GREEDY}}(P)|}{|s_{opt}(P)|} = 2 - \frac{1}{r}$ (see Fig. 1).

Consider $P := \{a_1a_2^{r-1}, a_2^r, a_2^{r-1}a_3, a_2a_3^{r-1}, \ldots, a_{m-1}^r, a_{m-1}^{r-1}a_m\}$ on the alphabet $\Sigma = \{a_1, a_2, \ldots, a_m\}$. Then in the worst case, the greedy solution is

$$s_{\text{GREEDY}}(P) = a_1a_2^{r-1}a_3^{r-1}\ldots a_{m-1}^{r-1}a_m a_2^r a_4^r \ldots a_{m-1}^r$$

while an optimum superstring is $s_{opt}(P) = a_1a_2^r a_3^r \ldots a_{m-1}^r a_m$. Thus, we get

$$\frac{|s_{\text{GREEDY}}(P)|}{|s_{opt}(P)|} = \frac{2 + (r-1)(m-2) + r(m-2)}{2 + r(m-2)} = 2 - \frac{m}{2 + r(m-2)} \xrightarrow[m\to\infty]{} 2 - \frac{1}{r}. \quad \square$$

In the instance of the proof above, because of the fixed word length, both the alphabet cardinality and the number of words go to infinity to reach the bound. Thanks to Proposition 2 and to Theorem 2, and by using the compression ratio of GREEDY, which equals 1/2, we obtain new bounds on the approximation ratio of GREEDY for r-SSP. The known greedy superstring ratio of 3.5 [10] allows us to precise the upper bound of super(GREEDY) for r-SSP.

Theorem 3. *The superstring approximation ratio of* GREEDY *for r-SSP is bounded by*

$$2 - \frac{1}{r} \leq \mathit{super}(\text{GREEDY}) \leq \min\left(\frac{r+1}{2}, \frac{7}{2}\right).$$

Note that the lower and upper bounds meet for $r = 2$. Theorem 3 suggests a more precise version of the greedy conjecture: the superstring approximation ratio of GREEDY on r-SSP is $2 - \frac{1}{r}$. Note that this ratio has been proven, but for a subset of instances corresponding to a restricted class of orders in which strings are merged, known as "linear greedy orders" [17].

Table 1 shows the actual bounds for small values of r. One observes that GREEDY achieves a superstring ratio that increases from 3/2 for words of length 2 until 7/2 for $r = 6$. It reaches a ratio of 2 for 3-SSP, which solves the classical greedy conjecture for 3-SSP. As the previously known bound on the approximation ratio of GREEDY for r-SSP is 7/2 [10], our theorem improves on this bound for all values of r below 6. Surprisingly for 2-SSP, GREEDY achieves a ratio of 3/2, which is tight. This shows that GREEDY can do better than the ratio of 2 stated by the classical greedy conjecture.

Note that other approximation algorithms (which are more complex than greedy) yield better approximation ratios for small values of r. For instance, an algorithm that combines a de Bruijn graph and an overlap graph approaches yields a ratio $(r^2 + r - 4)/(4r - 6)$, which is 4/3 for 3-SSP [9]. The greedy conjecture remains open for $r > 5$ and in general for SSP.

Table 1
Bounds on the approximation ratio of the greedy algorithm for r-SSP for $r < 7$. It achieves a bound of 2 for 3-SSP. Ratio 7/2 is the currently best known ratio for r-SSP in general. It also gives a tight ratio of 3/2 for 2-SSP, which is polynomial.

	r	1	2	3	4	5	6
Lower bound	$2-\frac{1}{r}$	1	**3/2**	5/3	7/4	9/5	11/6
Upper bound	$\min\left(\frac{r+1}{2}, \frac{7}{2}\right)$	1	**3/2**	**2**	5/2	3	**7/2**

4. Conclusion

The *Shortest Superstring Problem* is a crucial problem in computer science and has many practical applications in data compression, and in bioinformatics where it models genome assembly [8]. In this context, the case of r-SSP is realistic since sequencers often produce sequencing reads of the same length. Because it is simple and more efficient than other methods [10], and because it yields very good solutions in practice [12,15], the greedy algorithm is important. More generally, we exploit the relationship between the two approximation measures, the superstring length and the compression, to bound the superstring ratio in function of the compression ratio, which to our knowledge is new. This bound applies to SSP in general, and our results could prove useful for variants of SSP, like SSP for DNA strings, SSP with flippings, or for cyclic superstrings [2,6].

Maximising the compression or minimising the superstring length are dual problems (known as Maximum Compression and SSP, respectively). An optimal solution for one is also optimal for the other, while good approximate solutions differ for both. To solve this artificial asymmetry, another definition of approximation ratio has been proposed: the differential approximation ratio [5], which incorporates the size of the worst solution. For SSP in general, there is no longest superstring (no worst solution). With the natural restriction we considered for Theorem 1, the superstring ratio is the classical ratio, while the compression ratio is the differential approximation ratio for SSP. For the Maximum Compression problem, the compression ratio is both the classical and the differential approximation ratio. As we conjecture that computing a longest superstring obtained from a permutation of the input words is NP-hard, the study of the differential approximability of SSP appears as an appealing future line of research.

In addition, the greedy algorithm also gives an exact solution for finding the Shortest Cyclic Cover of Strings [4,3]. Proving the greedy conjecture in general remains a challenging open question. Here, we prove the greedy conjecture of a 2 superstring approximation ratio for 3-SSP, a restriction of SSP known to be NP-hard. Our proof also implies better superstring ratios for $r < 6$ (except 4). In addition, we show that greedy has a tight approximation bound of 3/2 on 2-SSP, meaning that it can yield ratios strictly smaller than 2, which was unknown. It suggests that the ratio depends on the length of input words. Hence, we propose to revise the greedy conjecture for input words of fixed length: is the superstring ratio of greedy equal to $2 - 1/r$?

Acknowledgements

This work is supported by Défi MASTODONS C3G, ANR Colib'read (ANR-12-BS02-0008) and the Institut de Biologie Computationnelle (ANR-11-BINF-0002).

References

[1] A. Blum, T. Jiang, M. Li, J. Tromp, M. Yannakakis, Linear approximation of shortest superstrings, J. ACM 41 (4) (1994) 630–647.
[2] B. Cazaux, R. Cánovas, E. Rivals, Shortest DNA cyclic cover in compressed space, in: Data Compression Conference, DCC, IEEE Computer Society Press, 2016, pp. 536–545.
[3] B. Cazaux, E. Rivals, Approximation of greedy algorithms for Max-ATSP, maximal compression, maximal cycle cover, and shortest cyclic cover of strings, in: Proc. of Prague Stringology Conference, (PSC), Czech Technical Univ. Prague, 2014, pp. 148–161. http://www.stringology.org/event/2014/p14.html.
[4] B. Cazaux, E. Rivals, The power of greedy algorithms for approximating Max-ATSP, Cyclic Cover, and superstrings, Discrete Appl. Math. 212 (2016) 48–60. http://dx.doi.org/10.1016/j.dam.2015.06.003, Stringology Algorithms.
[5] M. Demange, V.T. Paschos, On an approximation measure founded on the links between optimization and polynomial approximation theory, Theoret. Comput. Sci. 158 (1&2) (1996) 117–141. http://dx.doi.org/10.1016/0304-3975(95)00060-7.
[6] G. Fici, T. Kociumaka, J. Radoszewski, W. Rytter, T. Walen, On the greedy algorithm for the Shortest Common Superstring problem with reversals, Inform. Process. Lett. 116 (3) (2016) 245–251.
[7] J. Gallant, D. Maier, J.A. Storer, On finding minimal length superstrings, J. Comput. System Sci. 20 (1980) 50–58.
[8] T.P. Gevezes, L.S. Pitsoulis, Optimization in science and engineering, in: Honor of the 60th Birthday of Panos M. Pardalos, Springer New York, New York, NY, 2014, pp. 189–227. http://dx.doi.org/10.1007/978-1-4939-0808-0_10, (Chapter). The Shortest Superstring Problem.
[9] A. Golovnev, A. Kulikov, I. Mihajlin, Approximating shortest superstring problem using de Bruijn graphs, in: Combinatorial Pattern Matching, in: LNCS, Vol. 7922, Springer Verlag, 2013, pp. 120–129.
[10] H. Kaplan, N. Shafrir, The greedy algorithm for shortest superstrings, Inform. Process. Lett. 93 (1) (2005) 13–17.
[11] A.S. Kulikov, S. Savinov, E. Sluzhaev, Greedy conjecture for strings of length 4, in: Combinatorial Pattern Matching - 26th Annual Symposium, CPM 2015, Ischia Island, Italy, June 29 - July 1, 2015, Proceedings, 2015, pp. 307–315.
[12] B. Ma, Why greed works for shortest common superstring problem, Theor. Comput. Sci. 410 (51) (2009) 5374–5381.
[13] M. Mucha, Lyndon words and short superstrings, in: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2013, pp. 958–972.

[14] K.E. Paluch, Better Approximation Algorithms for Maximum Asymmetric Traveling Salesman and Shortest Superstring, CoRR abs/1401.3670.
[15] H.J. Romero, C.A. Brizuela, A. Tchernykh, An experimental comparison of approximation algorithms for the shortest common superstring problem, in: 5th Mexican International Conference on Computer Science, 2004, pp. 27–34.
[16] J. Tarhio, E. Ukkonen, A greedy approximation algorithm for constructing shortest common superstrings, Theor. Comput. Sci. 57 (1988) 131–145.
[17] M. Weinard, G. Schnitger, On the greedy superstring conjecture, SIAM J. Discrete Math. 20 (2) (2006) 502–522.
[18] A. Zaritsky, M. Sipper, The preservation of favored building blocks in the struggle for fitness: the puzzle algorithm, IEEE Trans. Evol. Comput. 8 (5) (2004) 443–455.