3-Shortest Superstring is 2-approximable by a greedy algorithm

Bastien Cazaux, Eric Rivals

3-Shortest Superstring is 2-approximable by a greedy algorithm

Bastien Cazaux and Eric Rivals
L.I.R.M.M. & Institut Biologie Computationnelle,
University of Montpellier II, CNRS U.M.R. 5506
161 rue Ada, F-34392 Montpellier Cedex 5, France
cazaux,rivals@lirmm.fr

27th June 2014

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 agglomeration algorithm achieves ratio of 2 for SSP. Here, we present a novel approach to bound the superstring approximation ratio with the compression ratio, which leads to a first proof of the greedy conjecture for 3-SSP.

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 [3]. 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). 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 elaborate algorithms have improved the approximation ratio for both measures culminating in $2\frac{11}{23}$ for the superstring [6] and in 3/4 for the compression measure [7]. A recent table listing these ratio and the literature, as well as known inapproximability bounds appears in [4].

In 1988, a seminal paper introduced a simple greedy algorithm, consisting in repeatedly agglomerating two words that exhibit the largest (prefix - suffix) overlap until only one string remains [8]. For example with

$P := \{abba, bbaa, aaba\}$, first, *abba* is agglomerated with *bbaa* yielding *abbaa* (they share an 3-letter overlap), then, *abbaa* is agglomerated with *aaba* resulting in the superstring *abbaaba* of length 7; as $\|P\| = 12$, the compression obtained equals $\|P\| - |t| = 12 - 7 = 5$. Tarhio and Ukkonen proved in the same article that the greedy algorithm achieves a compression ratio of 1/2 and formulated the *greedy conjecture*: the greedy algorithm yields a superstring ratio of 2. Despite the many 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. [1] have shown for the greedy algorithm a superstring ratio of 4, which was later improved to 3.5 in [5]. The greedy conjecture is supported by simulated experiments [9]. Moreover, the approximation ratio obtained by a simple greedy algorithm remains a crucial question, especially since other approximation algorithms are usually less efficient than a greedy one [5].

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. For a word x, $|x|$ denotes the *length* of x. 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.

Here, we exhibit 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 of P. Let us define the the approximation ratio of algorithm $\mathcal{A}$, denoted super($\mathcal{A}$), as the smallest real value such that for any input P:

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

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

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

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

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

Proof. Let $\alpha = \frac{1+(\gamma-1)\times\text{comp}(\mathcal{A})}{\gamma}$ and the function $f : x \mapsto \frac{1+(x-1)\times\text{comp}(\mathcal{A})}{x}$. 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:

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

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

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

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:

$$\begin{array}{rcl} \text{super}(\mathcal{A}) & \leq & \alpha \\ & \leq & \dfrac{(\gamma - 1) \times \text{comp}(\mathcal{A}) + 1}{\gamma}. \end{array}$$

□

3 Approximation of r-SSP

Let r be an integer satisfying $r > 1$. Here we 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. 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 $r = 2, \ldots, 6$, 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:*

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

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

$$\begin{array}{rcl} \dfrac{|s_{opt,P}|}{\|P\|} & \geq & \dfrac{r + p - 1}{\|P\|} \\ & \geq & \dfrac{r + p - 1}{r \times p} \\ & \geq & \dfrac{p}{r \times p} = \dfrac{1}{r} \end{array}$$

Using Theorem 1, we obtain

$$\begin{array}{rcl} \dfrac{|s_{\mathcal{A},P}|}{|s_{opt}|} & \leq & \dfrac{1 + (\frac{1}{r} - 1) \times \text{comp}(\mathcal{A})}{\frac{1}{r}} \\ & \leq & r \times (1 + (\dfrac{1 - r}{r}) \times \text{comp}(\mathcal{A})) \\ & \leq & r - (r - 1) \times \text{comp}(\mathcal{A}). \end{array}$$

□

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 [8, 2].

Proposition 2. *The greedy algorithm, denoted $\mathcal{G}$ 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 $\mathcal{G}$. To obtain the desired lower bound, we exhibit an instance where $\frac{|s_{\mathcal{G},P}|}{|s_{opt,P}|} = 2-\frac{1}{r}$.

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

$$s_{\mathcal{G},P} = a_1a_2^{r-1}a_3^{r-1}\ldots a_{m-1}^{r-1}a_ma_2^ra_3^r\ldots a_{m-1}^r$$

while an optimum superstring is $s_{opt,P} = a_1a_2^ra_3^r\ldots a_{m-1}^ra_m$. Thus, we get

$$\begin{aligned}\frac{|s_{\mathcal{G},P}|}{|s_{opt,P}|} &= \frac{2+(r-1)(m-2)+r(m-2)}{2+r(m-2)}\\ &= 2-\frac{m}{2+r(m-2)}\\ &\xrightarrow[m\to\infty]{} 2-\frac{1}{r}.\end{aligned}$$

□

Thanks to Proposition 2 and Theorem 2, and by using the compression ratio of $\mathcal{G}$, which equals 1/2, we obtain bounds on the approximation ratio of the greedy algorithm $\mathcal{G}$ for r-SSP.

Theorem 3. *The approximation ratio of the greedy algorithm $\mathcal{G}$ for r-SSP is bounded by*

$$2-\frac{1}{r} \leq \mathit{super}(\mathcal{G}) \leq \frac{r+1}{2}.$$

	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	$\frac{r+1}{2}$	1	3/2	2	5/2	3	7/2

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.

Table 1 shows the actual bounds for small values of r. One observes that GREEDY achieves a ratio of 3/2 for 2-SSP and a ratio of 2 for 3-SSP. This solves the greedy conjecture for 3-SSP. The goal of this work was to solve this long standing conjecture for 3-SSP. As the previously known bound on the approximation ratio of the greedy algorithm for r-SSP is 7/2 [5], our theorem improves on this bound for all values of r up to 5. Note that other approximation algorithms (which are more complex than greedy) yield better approximation ratios for small values of r; see a recent table appears in [4]). 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 [4]. The greedy conjecture remains open for $r \geq 4$ and in general for SSP.

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. In this context, the case of r-SSP is realistic since sequencers often produce sequencing reads of the same length. For the first time, we demonstrate the greedy conjecture of a 2 superstring approximation ratio for 3-SSP, a restriction of SSP known to be NP-hard. More generally we exploit the fact that the two approximation measures, the superstring length and the compression, are related for bounding the ratio of former by the one of the latter. This bound applies to SSP in general and our results can be used in other contexts. Moreover, the same greedy algorithm also gives an exact solution for finding the Shortest Cyclic Cover of Strings [2]. Proving the greedy conjecture in general remains a thrilling and challenging open question.

Acknowledgements: This work is supported by ANR Colib'read (ANR-12-BS02-0008) and Défi MASTODONS SePhHaDe from CNRS.

References

[1] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. In *ACM Symposium on the Theory of Computing*, pages 328–336, 1991.

[2] B. Cazaux and E. Rivals. Approximation of greedy algorithms for max-atsp, maximal compression, maximal cycle cover, and shortest cyclic cover of strings. In *Proceedings of the Prague Stringology Conference 2014, Prague, Czech Republic, September 1-3, 2014*, pages 148–161, 2014.

[3] J. Gallant, D. Maier, and J. A. Storer. On finding minimal length superstrings. *J. Comput. Syst. Sci.*, 20:50–58, 1980.

[4] A. Golovnev, A.S. Kulikov, and I Mihajlin. Approximating shortest superstring problem using de bruijn graphs. In Johannes Fischer and Peter Sanders, editors, *Combinatorial Pattern Matching*, volume 7922 of *Lecture Notes in Computer Science*, pages 120–129. Springer Berlin Heidelberg, 2013.

[5] H. Kaplan and N. Shafrir. The greedy algorithm for shortest superstrings. *Inf. Process. Lett.*, 93(1):13–17, 2005.

[6] M. Mucha. Lyndon words and short superstrings. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 958–972, 2013.

[7] K. E. Paluch. Better approximation algorithms for maximum asymmetric traveling salesman and shortest superstring. *CoRR*, abs/1401.3670, 2014.

[8] J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theor. Comput. Sci.*, 57:131–145, 1988.

[9] A. Zaritsky and M. Sipper. The preservation of favored building blocks in the struggle for fitness: The puzzle algorithm. *Trans. Evol. Comp*, 8(5):443–455, October 2004.