Approximation of Greedy Algorithms for Max-ATSP, Maximal Compression, Maximal Cycle Cover, and Shortest Cyclic Cover of Strings

Bastien Cazaux and Eric Rivals

Prague Stringology Conference 2014
Tuesday, September 02, 2014

Shortest Superstring and Shortest Cyclic Cover of linear strings

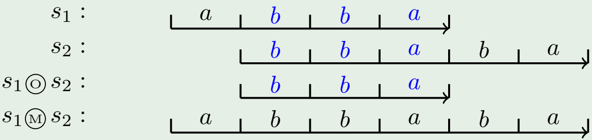
- Two problems related to assembly of string from overlaps of shorter strings.
- A basic step in DNA assembly
- Shortest superstring is a model for DNA assembly
- well studied hard problem, with approximation algorithms using Cyclic Covers.
- Question: what is the compression achieved by a greedy algorithm?
- Result: A new proof of $1/2$ compression ratio using subset systems.

Strings and maximum overlaps

- We consider finite strings over an alphabet Σ
- and denote by $|v|$ the length of a string v.

Example (Maximum overlap between two strings)

Let strings $s_1 := \text{abba}$ and $s_2 := \text{bbaba}$.

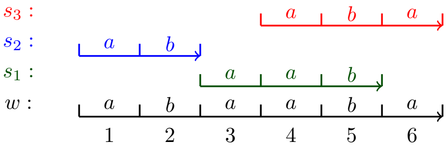


s_1 overlaps s_2 by two characters

overlaps are not symmetric

Definition

Let $P = \{s_1, s_2, \ldots, s_p\}$ be a set of strings. A *superstring* of P is a string w such that any s_i is a substring of w.



Problem: Shortest Superstring Problem (SSP)

Input: P a set of strings over Σ
Output: w a superstring of P of minimal length.

Known results on Shortest Superstring

State of the art

1. Problem is NP-hard [Gallant 1980]
2. and difficult to approximate [Blum et al. 1991]
3. Many variations of this problem: e.g. with fixed length input strings [Gusfield 1997]
4. Many approximation algorithms, most use a similar approach
   best known superstring ratio $2\frac{11}{30}$ [Paluch 2014]
   & conjecture optimum ratio equals 2 [Gallant 1980]

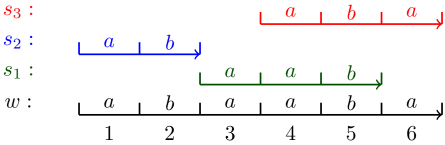
Applications

1. DNA Assembly in bioinformatics
2. Data compression
3. Natural language processing, translation, inference

Approximation measures

Two possible approximation measures:

- the length of the obtained superstring
- the compression of the input strings: $\sum_{i=1..p} |s_i| - |s'|$



Output superstring has length 6
Compression of 2 symbols;

Tarhio & Ukkonen seminal work

Theoretical Computer Science 57 (1988) 131–145
North-Holland 131

A GREEDY APPROXIMATION ALGORITHM FOR CONSTRUCTING SHORTEST COMMON SUPERSTRINGS *

Jorma TARHIO and Esko UKKONEN

Department of Computer Science, University of Helsinki, Teollisuuskatu 23, SF-00510 Helsinki, Finland

Theorem 3.2. *Let H be the approximate longest Hamiltonian path constructed by the greedy heuristic for the overlap graph of R, and let* H_{max} *be a longest Hamiltonian path. Then* $|H| \geq \frac{1}{2}|H_{max}|$.

Subset systems

Definition

A *subset system* is a pair $(E, \mathcal{L})$ comprising

- a finite set of elements E, and
- $\mathcal{L}$ a familly of subsets of E

satisfying two conditions:

(SS1) $\mathcal{L} \neq \emptyset$,

(SS2) If $A' \subseteq A$ and $A \in \mathcal{L}$, then $A' \in \mathcal{L}$.

Greedy algorithm for a subset system

Input : $(E, \mathcal{L})$

The elements e_i of E sorted by increasing weight:

$p(e_1) \leq p(e_2) \leq \ldots \leq p(e_n)$

$F \leftarrow \emptyset$

for $i = 1$ *to* n **do**

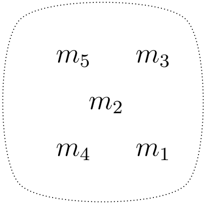
  if $F \cup \{e_i\} \in \mathcal{L}$ **then** $F \leftarrow F \cup \{e_i\}$;

return F

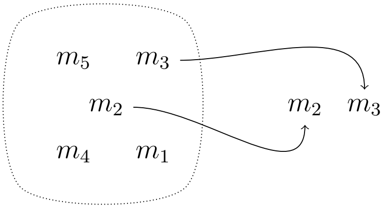
Output: A set F of $\mathcal{L}$ that is maximal for inclusion.

In our case, e_i is a maximum overlap, its weight is its length.

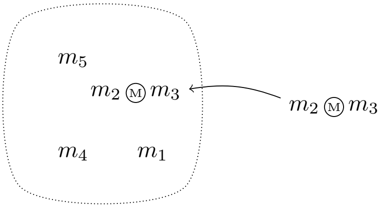
Greedy algorithm for Maximum Compression [Gallant 1980]



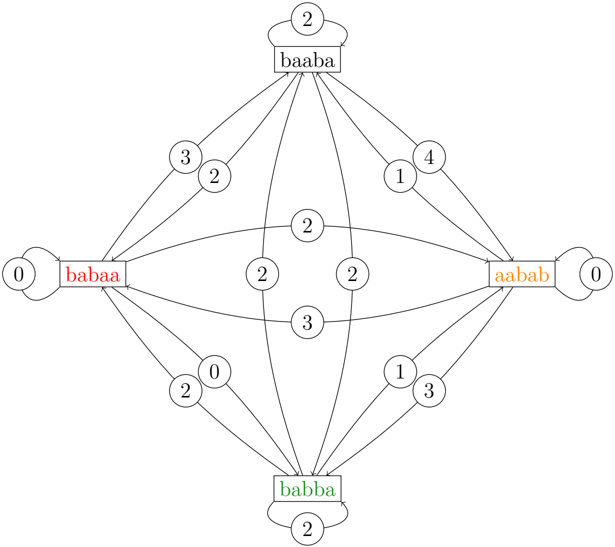
Greedy algorithm for Maximum Compression [Gallant 1980]



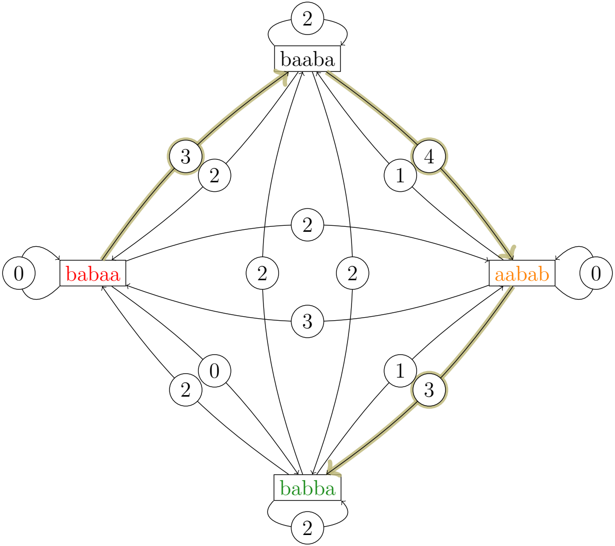
Greedy algorithm for Maximum Compression [Gallant 1980]



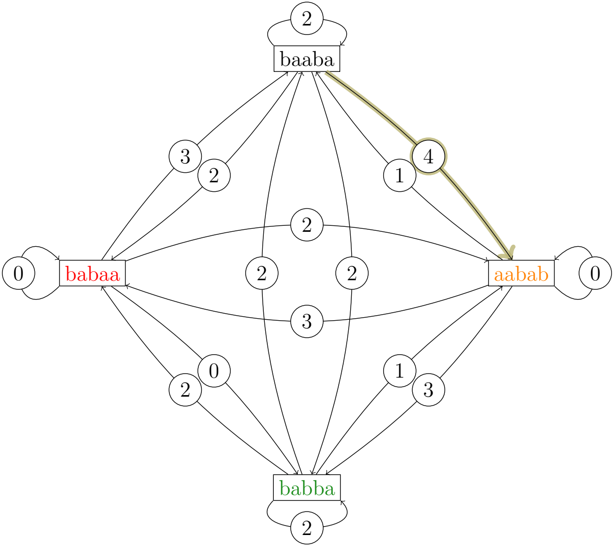
Overlap Graph



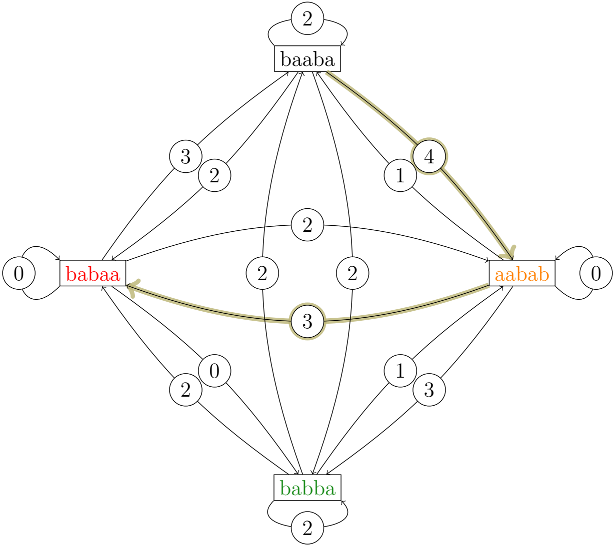
Superstring on the overlap graph



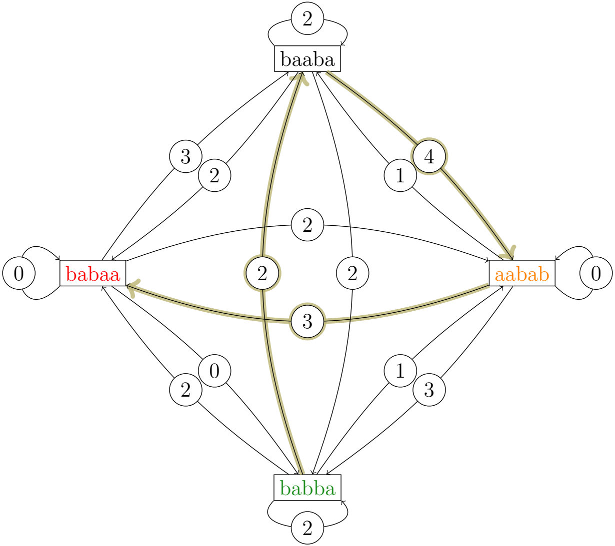
a compression of 10 symbols


2
baaba
3
2
4
1
2
0
babaa
2
2
aabab
0
3
0
2
1
3
babba
2

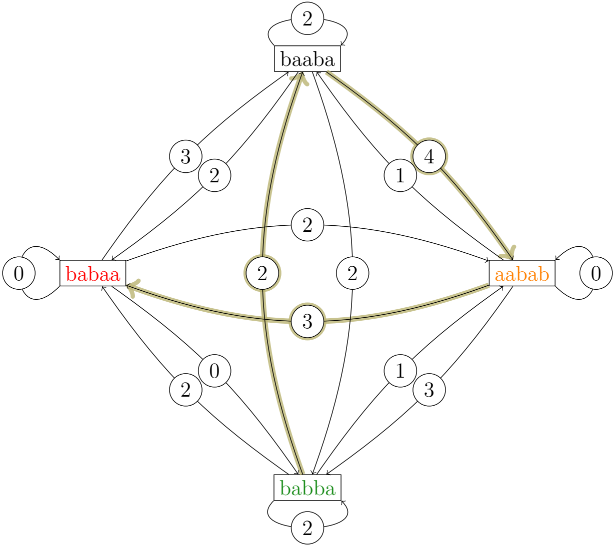
Greedy on the overlap graph



Greedy on the overlap graph



Greedy on the overlap graph



a compression of only 9 symbols

Notation

- $s \circledcirc t$: the maximum overlap between s and t
- E_S: the set of maximum overlaps between words of S
  $E_S := \{s_i \circledcirc s_j \mid s_i \text{ and } s_j \in S\}$.

Definition (Subset system for Maximum Compression)

We define $\mathcal{L}_S$ as the set of $F \subseteq E_S$ such that:

(L1) for each string, there is only one overlap to the left

(L2) and only one overlap to the right

(L3) there exists no cycle $(s_{i_1} \circledcirc s_{i_2}, \ldots, s_{i_{r-1}} \circledcirc s_{i_r}, s_{i_r} \circledcirc s_{i_1})$ in F, such that $\forall k \in \{1, \ldots, r\}, s_{i_k} \in S$.

Subset system for Maximum Compression

Notation

- $s \circledcirc t$: the maximum overlap between s and t
- E_S: the set of maximum overlaps between words of S
  $E_S := \{s_i \circledcirc s_j \mid s_i \text{ and } s_j \in S\}$.

Definition (Subset system for Maximum Compression)

We define $\mathcal{L}_S$ as the set of $F \subseteq E_S$ such that:

(L1) $\forall s_i,\ s_j$ and $s_k \in S$, $s_i \circledcirc s_k$ and $s_j \circledcirc s_k \in F \Rightarrow i = j$,

(L2) $\forall s_i,\ s_j$ and $s_k \in S$, $s_k \circledcirc s_i$ and $s_k \circledcirc s_j \in F \Rightarrow i = j$,

(L3) there exists no cycle $(s_{i_1} \circledcirc s_{i_2}, \ldots, s_{i_{r-1}} \circledcirc s_{i_r}, s_{i_r} \circledcirc s_{i_1})$ in F, such that $\forall k \in \{1, \ldots, r\}, s_{i_k} \in S$.

Extension and extensibility

Definition (Extension)

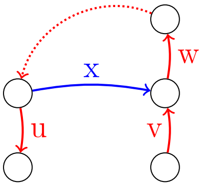
Let $A, B \in \mathcal{L}_{\mathcal{P}}$. B is an *extension of* A if $A \subseteq B$ and $B \in \mathcal{L}_{\mathcal{P}}$.

Definition (k-Extensibility)

Let $k \geq 1$ be an integer.
A subset system $(E, \mathcal{L})$ is said to be *k-extensible* if
for all $C \in \mathcal{L}$ and $x \notin C$ such that $C \cup \{x\} \in \mathcal{L}$, and
for any extension D of C,
there exists a subset $Y \subseteq D \setminus C$ with $\#(Y) \leq k$ satisfying

$$D \setminus Y \cup \{x\} \in \mathcal{L}.$$

$D \setminus C$ contains the red egdes and satisfies SS conditions
we wish to add x to the set
Question: which edges do we need to remove?



Answer: at most $\{u, v, w\}$.

Mestre's theorem

Theorem ([Mestre06])

Let $(E, \mathcal{L})$ be a subset system that is k-extensible. The greedy algorithm defined for $(E, \mathcal{L})$ with weight p yields an approximation ratio of $\frac{1}{k}$.

Theorem (1/3 approximation for Maximum Compression)

The approximation ratio of greedy algorithm for the maximum compression equals $\frac{1}{3}$.

Proof

Follows from the 3-extensibility of $(E_S, \mathcal{L}_S)$.

Greedy is not 2-extensible

The system $(E_S, \mathcal{L}_S)$ isn't 2-extensible.

Example (Non 2-extensible)

Let $P := \{s_1, \ldots, s_5\}$,

$C := \emptyset$, $x := s_1 \circledcirc s_2$ and

$D := \{s_1 \circledcirc s_3,\ s_4 \circledcirc s_2,\ s_5 \circledcirc s_1,\ s_2 \circledcirc s_5\}$, then

$D \setminus C = D$. For any $Y_S \subseteq D$ such that $D \setminus Y_S \cup \{x\} \in \mathcal{L}_S$

we have $\#(Y_S) \geq 3$ because $\{s_1 \circledcirc s_3,\ s_5 \circledcirc s_1,\ s_2 \circledcirc s_5\} \subseteq Y_s$.

Monge's inequality

Lemma Monge's inequality

Let s_1, s_2, s_3 and s_4 be four different words satisfying

1. $|s_1 \circledcirc s_2| \geq |s_1 \circledcirc s_4|$
2. and $|s_1 \circledcirc s_2| \geq |s_3 \circledcirc s_2|$.

Then:

$$|s_1 \circledcirc s_2| + |s_3 \circledcirc s_4| \geq |s_1 \circledcirc s_4| + |s_3 \circledcirc s_2|$$

Main result

Theorem (1/2 approximation)

The approximation ratio of greedy algorithm for the maximum compression equals $\frac{1}{2}$.

Proof

Detail the case of 3-extensibility following Mestre's idea.

combine with Monge's inequality

Shortest Cyclic Cover (SCC)

- Variant of SSP in which cycles are allowed
- The system looses the third "no cycle" condition
- Adapt the proof of 3-extensibility for SSP gives 2-extensibility for SCC
- Adapt the proof of 1/2-ratio of SSP gives a perfect ratio for SCC

Conclusion

- A simple proof of $1/2$ compression ratio for Shortest Superstring
- The approach does not work as such when the approximation measure is the length of the output superstring.
- A proof that greedy algorithm solves exactly the Shortest Cyclic Cover

Funding and acknowledgments

Thanks for your attention
Questions ?