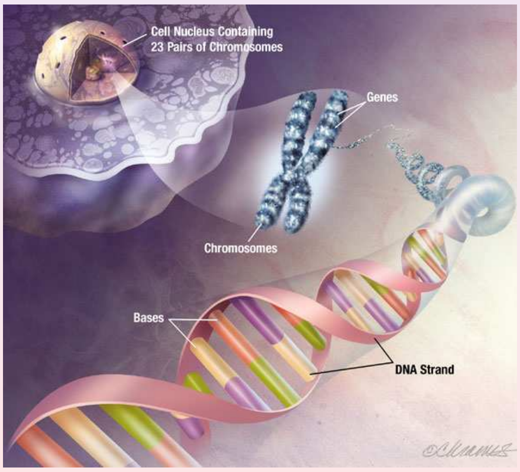
Approximating the Shortest Superstring Problem

Martin Paluszewski
University of Copenhagen

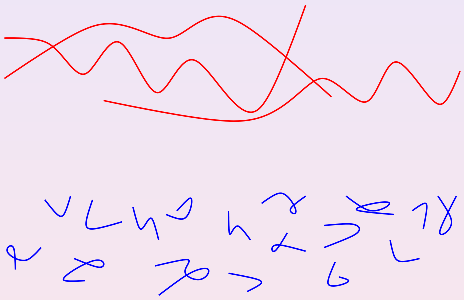
12/2-2008

Sequencing of DNA

- DNA, a string of 4 letters (A, G, C, T)
- Virus DNA 5×10^4 letters (base pairs)
- Human DNA 3×10^9 letters (base pairs)



Shotgun Method




```
ACTGAACTTACGGGCTAAAGCCATACGAATCCTACGAGA
-----------------------------------------
ACTGAACTTACGGG
                           AATCCTAC
         ACGGGCTAAAGCCATA
  TGAACTTACGGGCTAAAG
               TAAAGCCATACGAATCCTAC
ACTGAACTTACGGGCTA
                      ATACGAATCC
                                TACGAGA
```

Sequence Assembly

Problem: How can we automate this?

Data Compression

Input: 00101010111010010101110010101000100111001010100 10

Output: 001010111010010101010010011

	index, length
00101	0,5
0101110100	3,10
1010111	2,7
001010100	12,9
010011	20,6
100101010010	11,12

Output of compression:

- A shortest superstring
- An ordered list of start index and fragment lengths.

Shortest Superstring Problem (SSP)

- Given a finite alphabet Σ, and set of n strings, S=$\{s_1, \ldots, s_n\} \subseteq \Sigma^+$.
- Find a shortest string s that contains each s_i as a substring.
- Without loss of generality, we may assume that no string s_i is a substring of another string s_j, $j \neq i$.

Example

Input (S)

CATGC
CTAAGT
GCTA
TTCA
ATGCATC

$\Sigma = \{A, G, C, T\}$

Output

```
s=GCTAAGTTCATGCATC
 ........CATGC...
 .CTAAGT.........
 GCTA............
 ......TTCA......
 .........ATGCATC
```

NP-hard!

Solution Methods

- Exact Algorithms
- Metaheuristics
- Approximation Algorithms

Outline

1. Greedy, conjecture: ALG $\leq 2\cdot$ OPT
2. Set cover, $\mathrm{ALG} \leq 2 \cdot H_n \cdot \mathrm{OPT}$
   $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$
3. Cycle cover, ALG $\leq 4\cdot$ OPT
4. Cycle cover (greedy), ALG $\leq 3\cdot$ OPT

Best known: ALG $\leq 2.5\cdot$ OPT (Sweedyk)

Greedy Algorithm

Greedy Shortest Superstring
input: A set of strings S.
output: A short superstring of S.
T $\leftarrow$ S
while $|T| > 1$ **do**
    Let a and b be the most overlapping strings of T
    Replace a and b with the string obtained by overlapping a and b
end while
T contains a superstring of S

Example

- S = T = {**CATGC**, CTAAGT, GCTA, TTCA, **ATGCATC**}
- T = {CATGCATC, **CTAAGT**, **GCTA**, TTCA}
- T = {**CATGCATC**, GCTAAGT, **TTCA**}
- T = {**TTCATGCATC**, **GCTAAGT**}
- T = {GCTAAGTTCATGCATC}

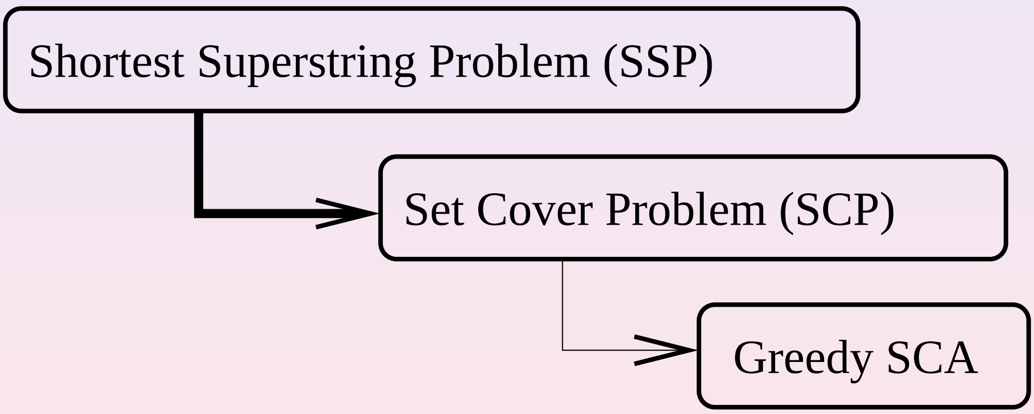
Approximation guarantee

- ALG $\leq 4 \cdot \mathrm{OPT}$ (proved by Blum et. al.)
- ALG $\leq 2 \cdot \mathrm{OPT}$ (conjectured)

Conjectured worst case

$S = \{ab^k, b^k c, b^{k+1}\}$

Approximating SSP Using Set Cover



- **Set Cover Problem:**
  Choose some **sets** that cover all **elements** with least cost
- **Elements**
- **Subsets**
- **Cost of a subset**

- **Set Cover Problem:**
  Choose some **sets** that cover all **elements** with least cost
- **Elements**
  - The input strings
- **Subsets**
  - σ_{ijk} = string obtained by overlapping input strings s_i and s_j, k letters.
  - $\beta = S \cup \sigma_{ijk}$, all i,j,k
  - Let $\pi \in \beta$
  - **set(π)** = {s ∈ S | s is a substr. of π }
- **Cost of a subset**
  - set(π) is $|\pi|$
- A solution to SSP is the concatenation of π obtained from SCP

Example

S = {CATGC, CTAAGT, GCTA, TTCA, ATGCATC}

π	Set	Cost

`CATGC.....`
`....CTAAGT`
`CATGCTAAGT`
	CATGC, CTAAGT, GCTA	10

`CATGC..`
`...GCTA`
`CATGCTA`
	CATGC, GCTA	7

`......CATGC`
`ATGCATC....`
`ATGCATCATGC`
	CATGC, ATGCATC	11

`CTAAGT...`
`.....TTCA`
`CTAAGTTCA`
	CTAAGT, TTCA	9

`ATGCATC.....`
`......CTAAGT`
`ATGCATCTAAGT`
	CTAAGT, ATGCATC	12

Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Strategy
Set Cover
Example
Approximation Guarantee

Example

`GCTA......`
`...ATGCATC`
`GCTATGCATC`
	GCTA, ATGCATC	10

`TTCA......`
`...ATGCATC`
`TTCATGCATC`
	TTCA, ATGCATC, CATGC	10

`GCTA...`
`.CTAAGT`
`GCTAAGT`
	GCTA, CTAAGT	7

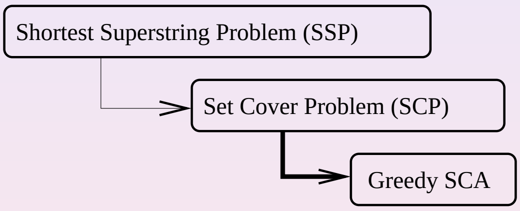
`TTCA...`
`..CATGC`
`TTCATGC`
	CATGC, TTCA	7

`CATGC...`
`.ATGCATC`
`CATGCATC`
	CATGC, ATGCATC	8

`CATGC`	CATGC	5
`CTAAGT`	CTAAGT	6
`GCTA`	GCTA	4
`TTCA`	TTCA	4
`ATGCATC`	ATGCATC	7

Martin Paluszewski University of Copenhagen Approximating the Shortest Superstring Problem

Approximation



Lemma

$$\mathrm{OPT}_{SSP} \leq \mathrm{OPT}_{SCP} \leq 2 \cdot \mathrm{OPT}_{SSP}$$

Approximation

Proof

First inequality: $\mathrm{OPT}_{SSP} \leq \mathrm{OPT}_{SCP}$
Let s be the string obtained from an optimal solution to the set cover problem. Then,

$$\mathrm{OPT}_{SCP} = |s|$$

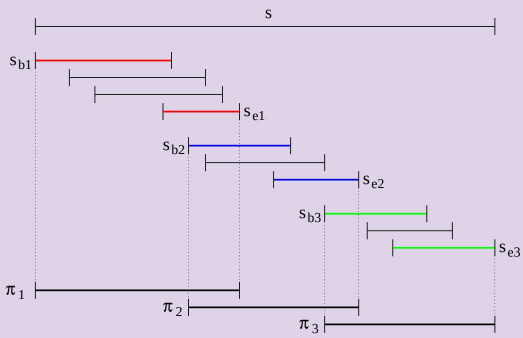
Since s covers all strings, s is a valid solution to the SSP and:

$$|s| \geq \mathrm{OPT}_{SSP}$$

Approximation

Proof, continued

Second inequality: $\mathrm{OPT}_{SCP} \leq 2 \cdot \mathrm{OPT}_{SSP}$
Show that some set cover always can be constructed such that the inequality holds.



Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Strategy
Set Cover
Example
Approximation Guarantee

Approximation

Proof, continued

- set(π_1), set(π_2), ..., set(π_t) is a solution to SCP (not necessarily optimal).
- Each input string is covered by at most two π strings. (π_i cannot overlap with π_{i+2}).
- $\sum_i |\pi_i| \leq 2 \cdot \mathrm{OPT}_{SSP}$
- $\mathrm{OPT}_{SCP} \leq 2 \cdot \mathrm{OPT}_{SSP}$ $\square$

Martin Paluszewski University of Copenhagen
Approximating the Shortest Superstring Problem

Approximation

Proof, continued

- $\mathrm{GREEDY}_{SCP} \leq H_n \mathrm{OPT}_{SCP}$
  $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$
  (Vijay V. Vazirani)

$$\boxed{\mathrm{GREEDY}_{SCP} \leq 2H_n \mathrm{OPT}_{SSP}}$$

$H_{10} \simeq 2.9$, $H_{100} \simeq 5.2$, $H_{1000} \simeq 7.5$

Factor 4 Algorithm (cycle cover)

Some definitions

- **overlap(s_i, s_j):** The maximum overlap between s_i and s_j,

```
i        ACGGCTAT....
j        ....CTATTAGC
overlap  ....CTAT....
```

- **prefix(s_i, s_j):** First letters of s_i where overlap(s_i, s_j) is removed

```
i        ACGGCTAT....
j        ....CTATTAGC
prefix   ACGG........
```

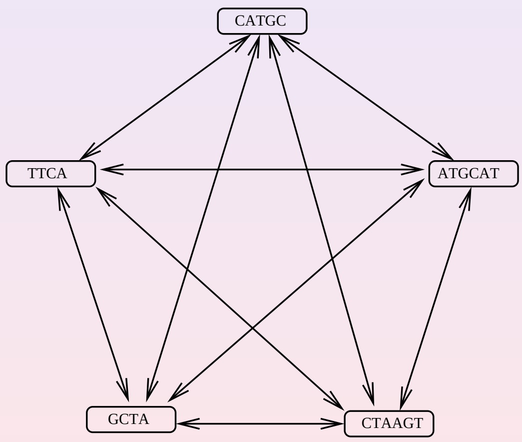
Important property of a shortest superstring s

$$s = \text{prefix}(s_1, s_2) \circ \text{prefix}(s_2, s_3) \circ \cdots \circ \text{prefix}(s_n, s_1) \circ \text{overlap}(s_n, s_1)$$

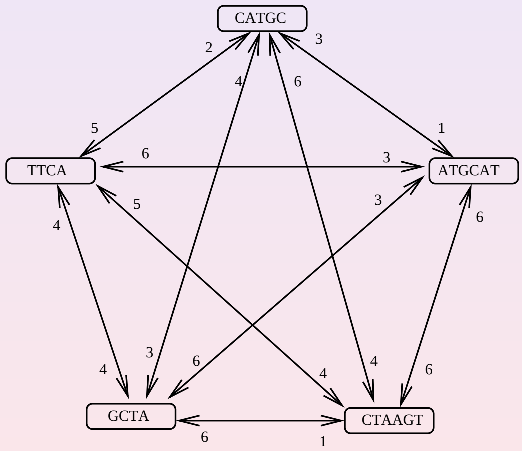
Example

```
s: TCTAAGTTCATGCATC
1  TCTA............
2  .CTAAGT.........
3  ......TTCA......
4  ........CATGC...
5  .........ATGCATC
1  ..............TCTA
```

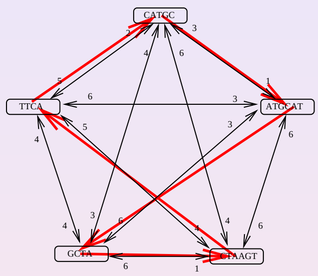

Prefix Graph



Prefix Graph



TSP in Prefix Graph



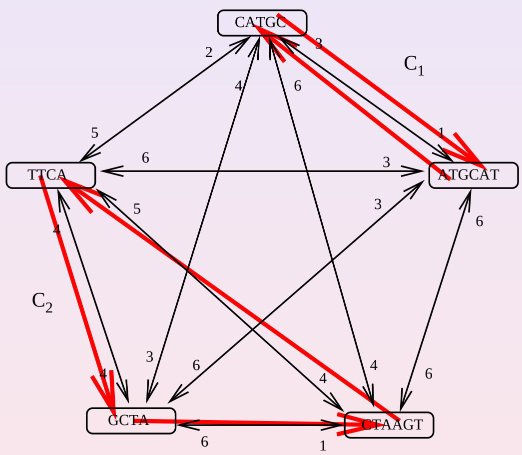
$$\mathrm{OPT}_{TSP} = |\mathrm{prefix}(s_a, s_b)| + |\mathrm{prefix}(s_b, s_c)| + \cdots + |\mathrm{prefix}(s_n, s_a)|$$

$$\mathrm{OPT}_{SSP} = |\mathrm{prefix}(s_1, s_2)| + |\mathrm{prefix}(s_2, s_3)| + \cdots + |\mathrm{prefix}(s_n, s_1)| + |\mathrm{overlap}(s_n, s_1)|$$

Lower bound:

$$\mathrm{OPT}_{\mathrm{TSP}} \leq \mathrm{OPT}_{\mathrm{SSP}}$$

TSP is NP-hard → cycle cover problem

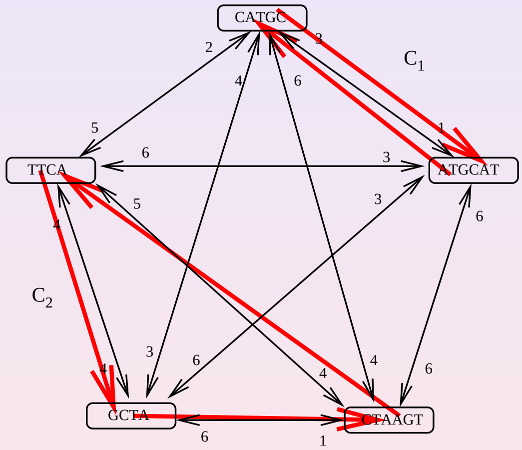


$$\mathrm{OPT}_{\mathrm{CCP}} \leq \mathrm{OPT}_{\mathrm{TSP}} \leq \mathrm{OPT}_{\mathrm{SSP}}$$

Algorithm (factor 4)

1. Construct the prefix graph corresponding to strings in S
2. Find a minimum weight cycle cover of the prefix graph, $C = \{c_1, ..., c_k\}$
3. For each cycle $c_i \in C$, arbitrarily choose a representative string s_{i_1}
4. For cycle $c_i \in C$, construct:
   $\alpha(c_i) = \mathrm{prefix}(s_{i_1}, s_{i_2}) \circ \cdots \circ \mathrm{prefix}(s_{i_{l-1}}, s_{i_l}) \circ \mathrm{prefix}(s_{i_l}, s_{i_1})$
5. For cycle $c_i \in C$, construct:
   $\sigma(c_i) = \alpha(c_i) \circ s_{i_1}$
6. Output $\sigma(c_1) \circ \cdots \circ \sigma(c_k)$

Example



- $\sigma(c_1) =$ C ATG CATGC
- $\sigma(c_2) =$ G CTAAG TTCA GCTA
- $s =$ C ATG CATGC G CTAAG TTCA GCTA

$$\boxed{\mathrm{ALG}_{\mathrm{CC}} \leq 4 \cdot \mathrm{OPT}_{SSP}}$$

Proof (sketch):

$$s = \sigma(c_1) \circ \ldots \circ \sigma(c_k) \quad (1)$$

$$s = \alpha(c_1) \circ s_{1_1} \circ \ldots \circ \alpha(c_k) \circ s_{k_1} \quad (2)$$

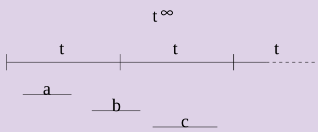
$$|s| = |\alpha(c_1) \circ \ldots \circ \alpha(c_k)| + |s_{1_1} \circ \ldots \circ s_{k_1}| \quad (3)$$

- Since $\mathrm{OPT}_{CCP} = |\alpha(c)|$ the α strings are bounded by OPT_{SSP}.
- $\mathrm{OPT}_{\mathrm{CCP}} \leq \mathrm{OPT}_{\mathrm{TSP}} \leq \mathrm{OPT}_{\mathrm{SSP}}$
- We need to show that concatenation of representative strings is bounded by $3 \cdot \mathrm{OPT}_{SSP}$

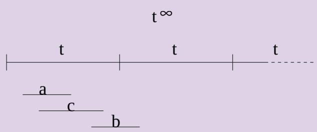
Lemma 1

If each string in $S' \subseteq S$ is a substring of t^∞ for a string t, then there is a cycle of weight at most $|t|$ in the prefix graph covering all vertices corresponding to strings in S'

Proof (sketch)

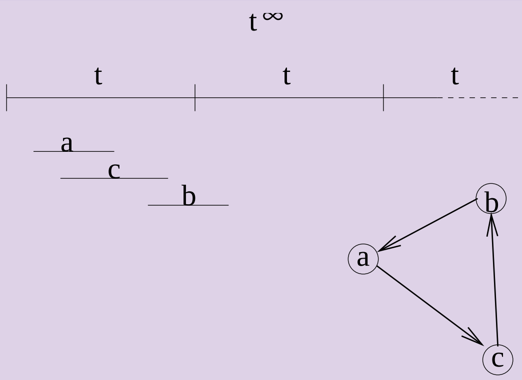


- Sort strings



Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Definitions
Prefix Graph
Cycle Cover
Example
Approximation Guarantee

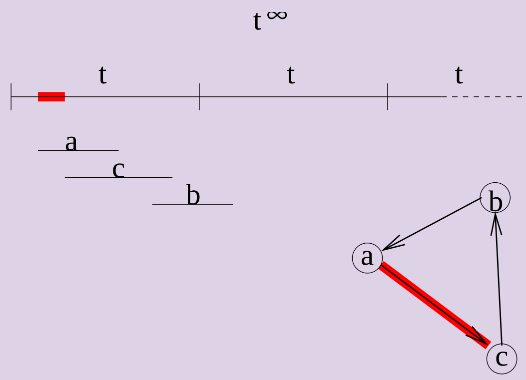
Proof (sketch), continued



Cost$(a \rightarrow c \rightarrow b \rightarrow a) \leq |t|$

Martin Paluszewski University of Copenhagen
Approximating the Shortest Superstring Problem

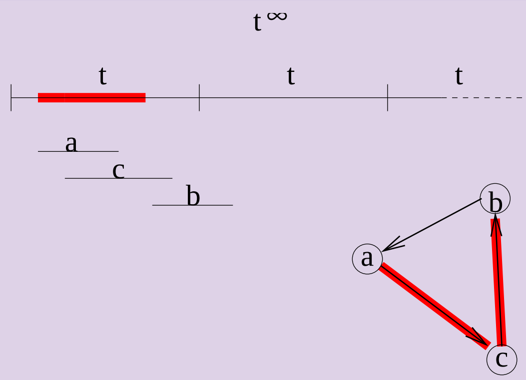
Proof (sketch), continued



$\text{Cost}(a \to c \to b \to a) \leq |t|$

Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Definitions
Prefix Graph
Cycle Cover
Example
Approximation Guarantee

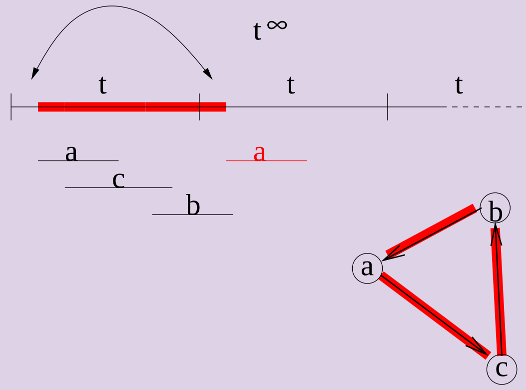
Proof (sketch), continued



$\text{Cost}(a \to c \to b \to a) \leq |t|$

Martin Paluszewski University of Copenhagen
Approximating the Shortest Superstring Problem

Proof (sketch), continued

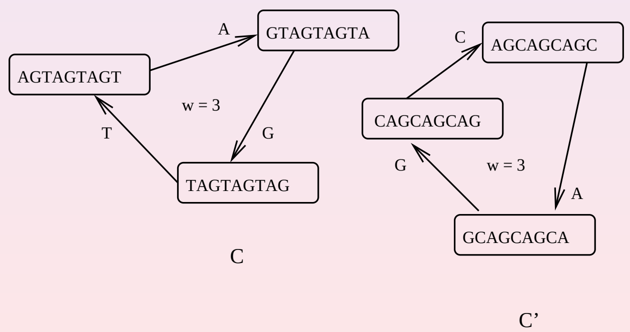


$\text{Cost}(a \to c \to b \to a) \leq |t|$

Lemma 2

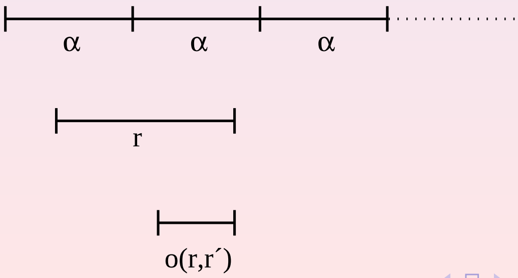
Let C be an optimal cycle cover. Let c and c' be two cycles in C, and let r, r' be strings from each cycle. Then

$$|\text{overlap}(r, r')| < w(c) + w(c')$$



Proof

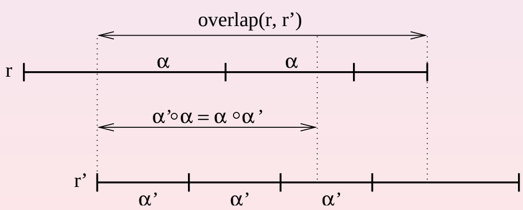
- Suppose $|\text{overlap}(r, r')| \geq w(c) + w(c')$
  $\alpha(c_i) = \text{prefix}(r, s_{i_2}) \circ \cdots \circ \text{prefix}(s_{i_{l-1}}, s_{i_l}) \circ \text{prefix}(s_{i_l}, r)$
- r is a substring of $\alpha(c)^\infty$, so overlap(r, r') is also a substring of $\alpha(c)^\infty$.
- $|\text{overlap}(r, r')| \geq w(c)$, so the overlap consists of repeating strings α(c).



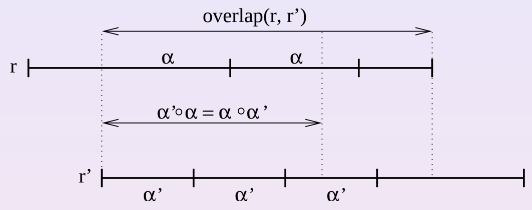
Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Definitions
Prefix Graph
Cycle Cover
Example
Approximation Guarantee

Proof

- Suppose $|\text{overlap}(r, r')| \geq w(c) + w(c')$
  $\alpha(c_i) = \text{prefix}(r, s_{i_2}) \circ \cdots \circ \text{prefix}(s_{i_{l-1}}, s_{i_l}) \circ \text{prefix}(s_{i_l}, r)$
- r is a substring of $\alpha(c)^\infty$, so overlap(r, r') is also a substring of $\alpha(c)^\infty$.
- $|\text{overlap}(r, r')| \geq w(c)$, so the overlap consists of repeating strings α(c).



Martin Paluszewski University of Copenhagen
Approximating the Shortest Superstring Problem



Proof, continued

α and α' commute, so

$$\alpha^k \circ \alpha'^k = \alpha'^k \circ \alpha^k$$

for all $k > 0$, so

$$\alpha^\infty = \alpha'^\infty$$

Applications
Shortest Superstring Problem
Greedy Algorithm
Set Cover Based Algorithm
Factor 4 Algorithm
Factor 3 Algorithm
Definitions
Prefix Graph
Cycle Cover
Example
Approximation Guarantee

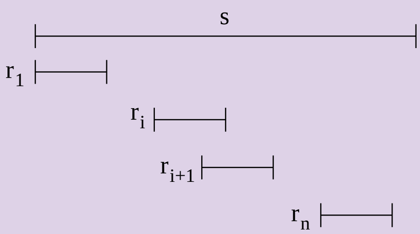
Proof, continued

- All strings from c and c' are substring of α^∞.
- There is a cycle covering all strings in c and c' with weight at most $|\alpha|$ (lemma 1)
- Contradiction.

Martin Paluszewski University of Copenhagen Approximating the Shortest Superstring Problem

Proof of factor 4 algorithm

- r_i are sorted representative strings



$$\mathrm{OPT}_{SSP} \geq \sum_{i=1}^{k} |r_i| - \sum_{i=1}^{k-1} |\mathrm{overlap}(r_i, r_{i+1})|$$

Proof of factor 4 algorithm, continued.

$$\mathrm{OPT}_{SSP} \geq \sum_{i=1}^{k} |r_i| - \sum_{i=1}^{k-1} |\mathrm{overlap}(r_i, r_{i+1})|$$

Lemma 2:

$$|\mathrm{overlap}(r, r')| < w(c) + w(c')$$

$$\mathrm{OPT}_{SSP} \geq \sum_{i=1}^{k} |r_i| - 2\sum_{i=1}^{k} w(c_i)$$

$$\sum_{i=1}^{k} |r_i| \leq \mathrm{OPT}_{SSP} + 2\sum_{i=1}^{k} w(c_i) \leq 3\mathrm{OPT}_{SSP}$$

$$\mathrm{ALG} = \sum_{i=1}^{k} |\sigma(c_i)| = w(C) + \sum_{i=1}^{k} |r_i| \leq 4 \cdot \mathrm{OPT}_{SSP}$$

Factor 3 Algorithm

- Construct the prefix graph corresponding to strings in S
- Find a minimum weight cycle cover of the prefix graph, $C = \{c_1, ..., c_k\}$
- Run the greedy superstring algorithm on $\{\sigma(c_1), \cdots, \sigma(c_k)\}$ and output the resulting string τ

Factor 3 Algorithm

Compression

- $||S|| = \sum_{x \in S} |x|$
- Compression $= ||S|| - |s|$
- Max. compression $\leftrightarrow$ min. superstring
- Lemma: Greedy superstring algorithm achieves at least half the optimal compression:

$$||S|| - \mathrm{GREEDY} \geq \frac{1}{2}(||S|| - \mathrm{OPT})$$

Lemma 1

$$|\tau| \leq \mathrm{OPT}_\sigma + w(C)$$

Proof

- Assume $\sigma(c_1), \cdots, \sigma(c_k)$ appear in this order in the superstring of S_σ
- Maximum compression is:

$$||S_\sigma|| - \mathrm{OPT}_\sigma = \sum_{i=1}^{k-1} |\mathrm{overlap}(\sigma(c_i), \sigma(c_{i+1}))|$$

Proof, continued

$$||S_\sigma|| - \mathrm{OPT}_\sigma = \sum_{i=1}^{k-1} |\mathrm{overlap}(\sigma(c_i), \sigma(c_{i+1}))|$$

- $|\mathrm{overlap}(r, r')| < w(c) + w(c')$ (lemma 2)
- So maximum compression is: $||S_\sigma|| - \mathrm{OPT}_\sigma \leq 2w(C)$
- Greedy algorithm gives: $||S_\sigma|| - |\tau| \geq \frac{1}{2}(||S_\sigma|| - \mathrm{OPT}_\sigma)$
- $2(|\tau| - \mathrm{OPT}_\sigma) \leq ||S_\sigma|| - \mathrm{OPT}_\sigma \leq 2w(C)$

Lemma 2

$$\mathrm{OPT}_\sigma \leq \mathrm{OPT}_{SSP} + w(C)$$

Proof

- Let $S_r = \{r_1, \cdots, r_k\}$ (repr. strings)
- Each $\sigma(c_i)$ begins and ends with r_i, so $||S_\sigma|| - \mathrm{OPT}_\sigma \geq ||S_r|| - \mathrm{OPT}_r$
- $||S_\sigma|| = ||S_r|| + w(C)$
- $\mathrm{OPT}_\sigma \leq \mathrm{OPT}_r + w(C)$

Lemma 1

$$|\tau| \leq \mathrm{OPT}_\sigma + w(C)$$

Lemma 2

$$\mathrm{OPT}_\sigma \leq \mathrm{OPT}_{SSP} + w(C)$$

Combine lemma 1 and lemma 2 to get:

$$|\tau| \leq 3 \cdot \mathrm{OPT}_{SSP}$$