Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Approximating Shortest Superstring Problem Using de Bruijn Graphs

Advanced Algorithms Course Presentation

Farshad Barahimi

Department of Computer Science
University of Lethbridge

November 19, 2013

Introduction

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- This presentation is a review of the following paper:

  Title: Approximating Shortest Superstring Problem Using de Bruijn Graphs
  Authors: Alexander Golovnev, Alexander S. Kulikov, and Ivan Mihajlin

- This paper presents some results on approximating the shortest superstring problem.

Problem definition

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

common superstring problem (SCS) :
given: n strings $s_1, \ldots, s_n$
goal: find a shortest string containing each s_i as a substring.

Definition

r-superstring problem (r-SCS) : SCS problem for the special case when all input strings have length exactly r.

Complexity

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- SCS over the binary alphabet and 3-SCS are NP-hard.
- 2-SCS can be solved in linear time.

Previous results for approximating SCS

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

3	Blum, Jiang, Li, Tromp and Yannakakis [4]	1991
$2\frac{8}{9}$	Teng, Yao [23]	1993
$2\frac{5}{6}$	Czumaj, Gasieniec, Piotrow, Rytter [8]	1994
$2\frac{50}{63}$	Kosaraju, Park, Stein [15]	1994
$2\frac{3}{4}$	Armen, Stein [1]	1994
$2\frac{50}{69}$	Armen, Stein [2]	1995
$2\frac{2}{3}$	Armen, Stein [3]	1996
$2\frac{25}{42}$	Breslauer, Jiang, Jiang [5]	1997
$2\frac{1}{2}$	Sweedyk [21]	1999
$2\frac{1}{2}$	Kaplan, Lewenstein, Shafrir, Sviridenko [12]	2005
$2\frac{1}{2}$	Paluch, Elbassioni, van Zuylen [18]	2012
$2\frac{11}{23}$	Mucha [16]	2013

Result presented in this paper

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- In this paper a $\frac{r^2+r-4}{4r-6}$-approximation to r-SCS is presented.
- This is better than the best known approximation ratio ($2\frac{11}{23}$) for $r = 3, \ldots, 7$.

Definitions

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

overlap(s, t): the longest suffix of s that is also a prefix of t.

Definition

prefix(s, t): first $|s| - |overlap(s, t)|$ symbols of s.

Definition

suffix(s, t): last $|t| - |overlap(s, t)|$ symbols of t.

Definitions

Start

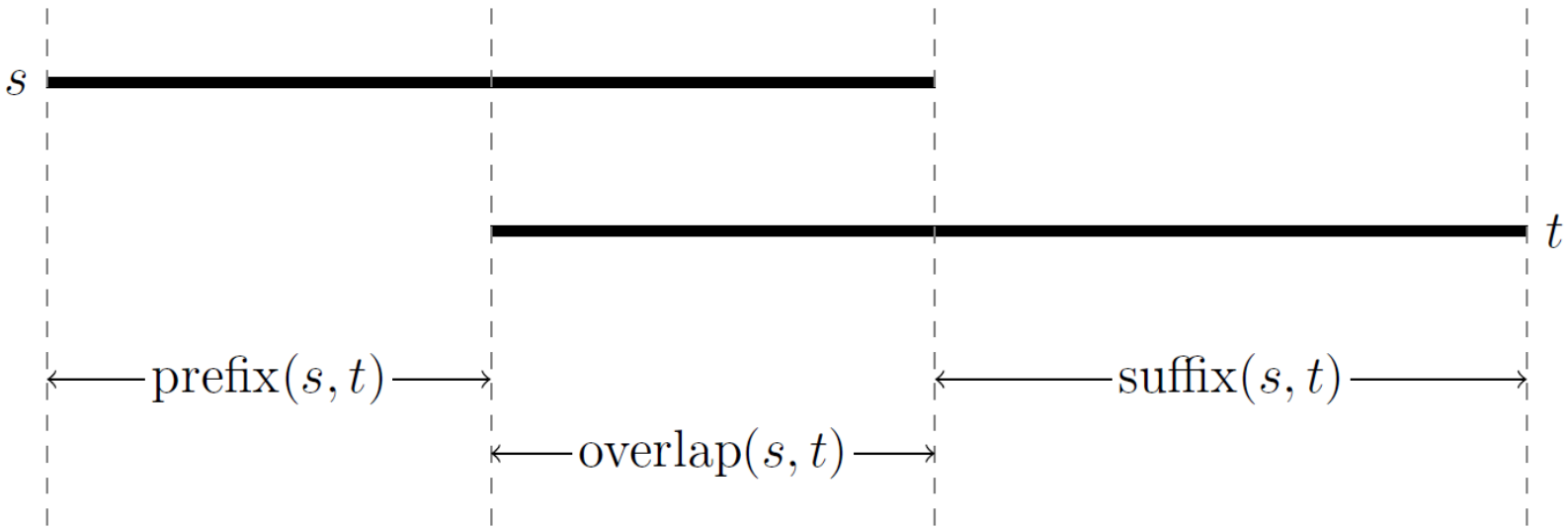
Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof



Definitions

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

prefix(s): string resulting from s by removing the last symbol.

Definition

suffix(s): string resulting from s by removing the first symbol.

Definitions

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

let $S = \{s_1, \ldots, s_n\}$ be a set of strings over an alphabet Σ and s be a superstring of S. The **compression** of s (w.r.t. S) is : $|s_1| + |s_2| + \cdots + |s_n| - |s|$.

- Clearly minimizing the length of a superstring corresponds to maximizing the compression.

SCS as a typical permutation problem

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- If we know the order of the input strings in a shortest superstring then we can recover this superstring by overlapping the strings in this given order.
- For this reason, it will be convenient for us to identify a superstring with the order of input strings in it.

Overlap graph

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

The **overlap graph** OG(S) of the set of strings $S = \{s_1, \ldots, s_n\}$ is a complete weighted directed graph on a set of vertices $V = \{1, \ldots, n\}$. The weight of an edge from i to j equals $|overlap(si, sj)|$.

Start

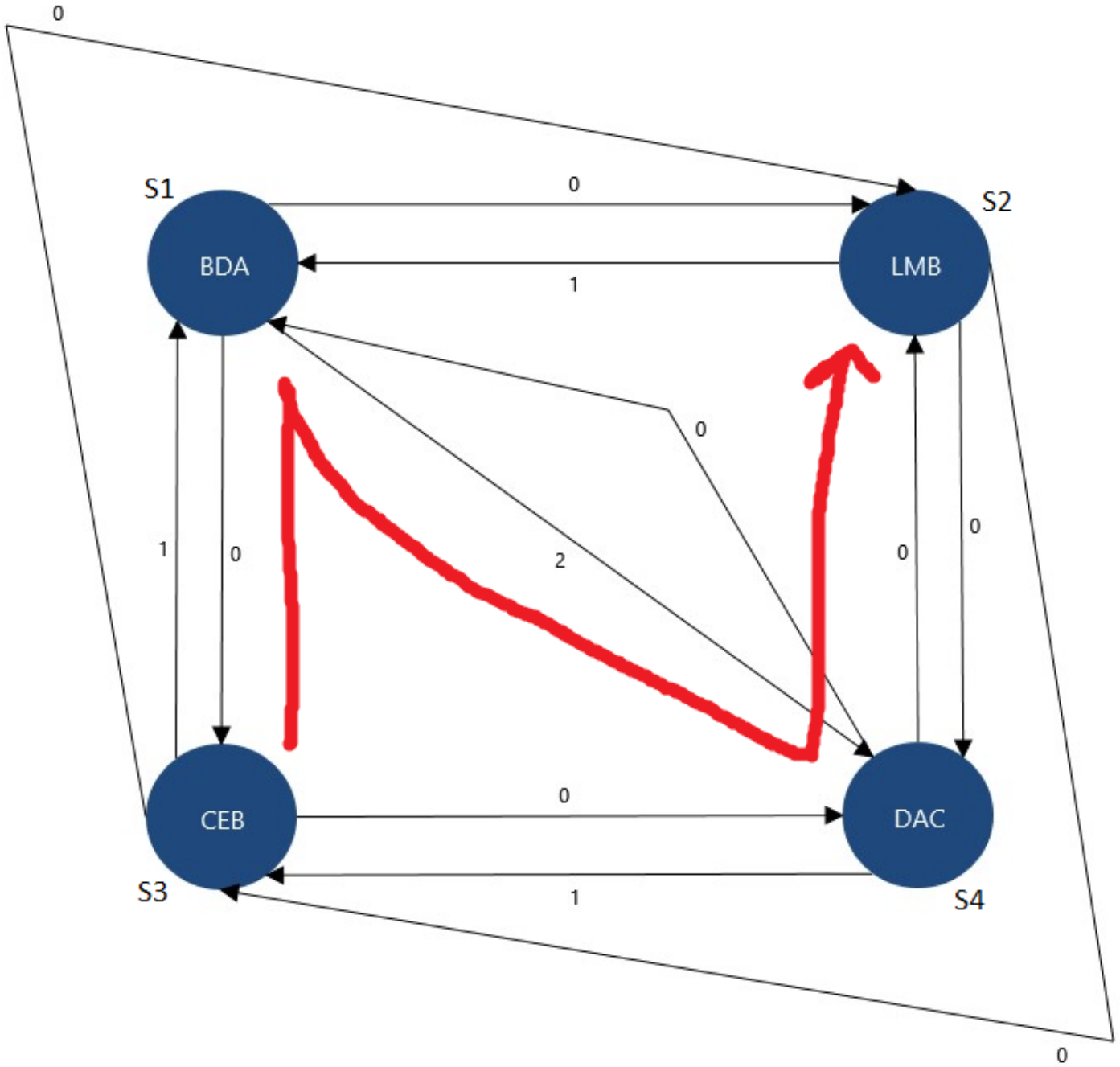
Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof



Overlap graph

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

asymmetric maximum traveling salesman path problem (MAX-ATSP) is to find a longest path visiting each vertex of the graph exactly once (such a path is called Hamiltonian).

Overlap graph

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- Solving SCS corresponds to solving the asymmetric maximum traveling salesman path (MAX-ATSP) problem in overlap graph OG(S).
- The length of any Hamiltonian path in this graph equals the compression of the corresponding superstring.
- The best known approximation ratio for MAX-ATSP is $\frac{2}{3}$.
- This immediately gives a $\frac{2}{3}$-approximation for the compression.
- An α-approximation for MAX-ATSP (compression) implies a $3.5 - 1.5\alpha$ approximation for SCS.
- This means a 2.5-approximation for SCS.

De Bruijn Graphs

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Definition

In **De Bruijn Graph** each input string $s_i \in S$ is represented as a directed (unweighted) edge from prefix(s_i) to suffix(s_i).

2-SCS using De Bruijn Graph and eulerian path

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- 2-SCS can be solved using De Bruijn Graph.
- An eulerian path spells a shortest superstring.
- To make the graph have an eulerian path, for each weakly connected component we add edges between imbalanced vertices (i.e., vertices with non-zero difference of in-degree and out-degree) so that the resulting component contains an Eulerian path. Finally, we add edges between components so that the graph contains an Eulerian path.
- r-SCS cannot be solved with the same technique because in case of 2-SCS, strings from different weakly connected components do not share letters (and hence have empty overlap) so the components can be traversed in any order.

2-SCS using De Bruijn Graph and eulerian path

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

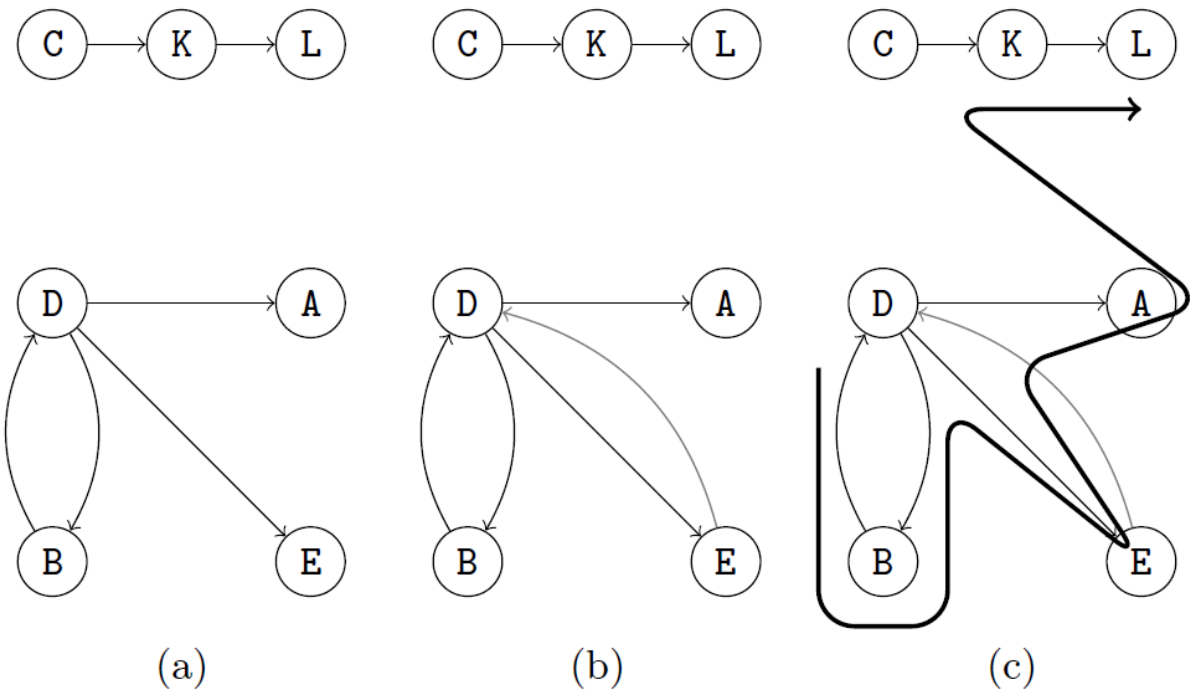


Fig. 2. 2-SCS can be solved in polynomial time. (a) de Bruijn graph of a set of strings {KL, DB, DE, CK, BD, DA}. (b) After adding an edge ED each weakly connected component contains an Eulerian path. (c) The string DBDEDACKL spelled by a path going through all the edges is a shortest superstring.

Algorithm

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Algorithm 3.1. $(r^2 + r - 4)/(4r - 6)$-approximation algorithm r-SCS

Input: $\mathcal{S} = \{s_1, \ldots, s_n\} \subseteq \Sigma^r$.

Output: A superstring of S that is at most $(r^2 + r - 4)/(4r - 6)$ times longer than a shortest superstring.

// first, find a long traveling salesman path in the overlap graph

1: let π be a 2/3-approximate maximum traveling salesman path in $\mathrm{OG}(\mathcal{S})$

// then, find a short rural postman path in the de Bruijn graph

2: let $\mathcal{S}' = \{s'_1, \ldots, s'_n\} \subseteq \Sigma_1^2$ be a set of 2-strings over the alphabet $\Sigma_1 = \Sigma^{r-1}$; s'_i is the 2-string consisting of prefix of s_i of length $r - 1$ and suffix of s_i of length $r - 1$

3: let π_1 be a shortest superstring for the set of 2-strings $\mathcal{S}'$

4: **return** the better one among π and π_1

Algorithm

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

Theorem

Algorithm 3.1 finds an $\alpha(r)$-approximation for r-SCS where $\alpha(r) = \frac{r^2+r-4}{4r-6}$

Proof

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- Let H be a shortest Hamiltonian path in OG(S). Then clearly
  $OPT(S) = rn - w(H)$,
  where w(H) is the weight of H.
- A $\frac{2}{3}$-approximate maximum traveling salesman path has weight at least $\frac{2w(H)}{3}$.
- Thus, the permutation Π gives a superstring of length at most $rn - \frac{2w(H)}{3}$
- The corresponding approximation ratio is:

$$\frac{rn - \frac{2w(H)}{3}}{rn - w(H)}. \qquad (1)$$

Proof

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- Let u denote the number of edges of weight at most (r - 2) in H.
- Then the number of edges of weight exactly (r - 1) in H is (n - 1 - u).
- Then we have

$$w(H) \leq (r-1)(n-1-u) + (r-2)u \qquad \text{(a)}$$
$$w(H) \leq (r-1)(n-1-u) + (r-1)u - u \qquad \text{(b)}$$
$$w(H) \leq (r-1)(n-1) - u \qquad \text{(c)}$$

and hence

$$u \leq (r-1)(n-1) - w(H). \qquad (2)$$

Proof

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- Note that

$$overlap(s'_i, s'_j) = \begin{cases} 1 & \text{if } overlap(s_i, s_j) = r - 1 \\ 0 & \text{otherwise.} \end{cases}$$

- Since S' is a 2-SCS instance, a shortest superstring for S' has the maximal possible number of overlaps of size 1.
- This number is in turn equal to the maximal possible number of overlaps of size r-1 for S.
- Since the number of overlaps of size r-1 in H is (n-1-u), the length of a shortest superstring for S' is at most $2n - (n - 1 - u) = n + u + 1$.

Proof

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- Hence Π_1 gives a superstring of S of length at most
  $rn - (r-1)(n-1-u)$ (a)
- Because of (2), this is at most
  $rn - (r-1)(n-1-(r-1)(n-1)+w(H))$ (b)
  which is less than
  $(r^2-2r+2)n-(r-1)w(H).$ (c)
- The corresponding approximation ratio is

$$\frac{(r^2-2r+2)n-(r-1)w(H)}{rn-w(H)} \quad (3)$$

Proof

Start

Introduction

Overlap graph

De Bruijn Graphs

Algorithm

Proof

- From (1) and (3) and a simple observation that $0 \leq \frac{w(H)}{n} \leq (r-1)$ we conclude that the approximation ratio of the constructed algorithm is

$$\alpha(r) = \max_{0 \leq x \leq r-1} \{\min\{\frac{r-\frac{2x}{3}}{r-x}, \frac{(r^2-2r+2)-(r-1)x}{r-x}\}\}$$

- The maximum of their minimum is attained at x where they meet, namely

$$x = \frac{r^2-3r+2}{r-\frac{5}{3}}.$$

- Plugging in this x gives

$$\alpha(r) = \frac{r^2+r-4}{4r-6}.$$