Making de Bruijn Graphs Eulerian

Giulia Bernardini[1,2], Huiping Chen[3], Grigorios Loukides[3],
Solon P. Pissis[2,4], Leen Stougie[2,4] and **Michelle Sweering[2]**

[1]University of Trieste, Trieste,
[2]CWI, Amsterdam
[3]King's College London
[4]Vrije Universiteit, Amsterdam

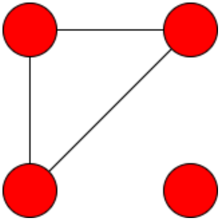
27th June 2022

Outline

Graphs

A **graph** G consists of

- a finite set of nodes V
- a finite set of edges E connecting the nodes

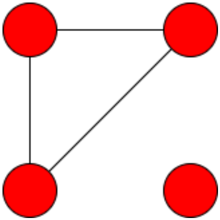


Graphs

A **graph** G consists of

- a finite set of nodes V
- a finite set of edges E connecting the nodes

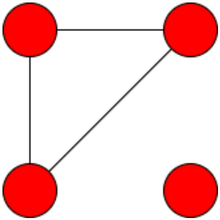
In a **directed graph** the edges $E \subseteq \{(u, v) : u, v \in V\}$ have a direction associated with them.



Graphs

A **graph** G consists of

- a finite set of nodes V
- a finite set of edges E connecting the nodes



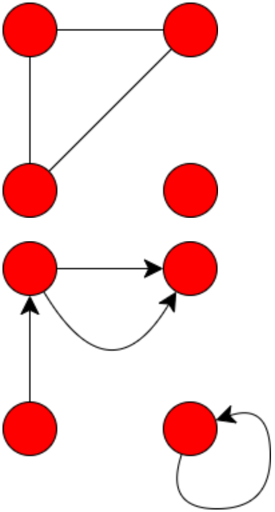
In a **directed graph** the edges $E \subseteq \{(u, v) : u, v \in V\}$ have a direction associated with them.

In a **multigraph** we can have multiple copies of each edge.

Graphs

A **graph** G consists of

- a finite set of nodes V
- a finite set of edges E connecting the nodes



In a **directed graph** the edges $E \subseteq \{(u, v) : u, v \in V\}$ have a direction associated with them.

In a **multigraph** we can have multiple copies of each edge.

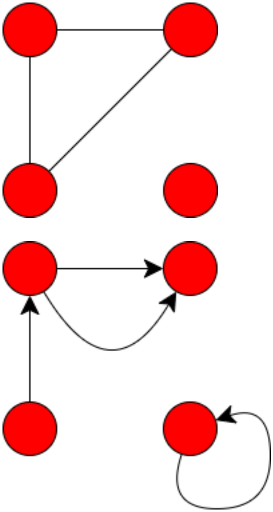
Graphs

A **graph** G consists of

- a finite set of nodes V
- a finite set of edges E connecting the nodes

In a **directed graph** the edges $E \subseteq \{(u, v) : u, v \in V\}$ have a direction associated with them.

In a **multigraph** we can have multiple copies of each edge.

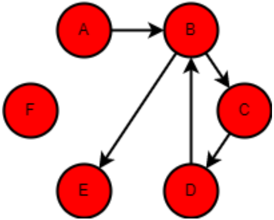


Our problem

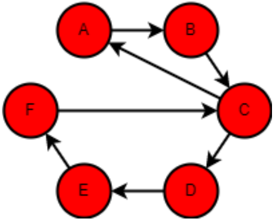
We work with directed multigraphs.

More Graph Definitions

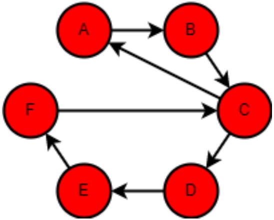
Walk
Sequence of nodes connected by edges



Circuit
Walk with the same first and last node



Graph Problems



Eulerian walk/ciruit

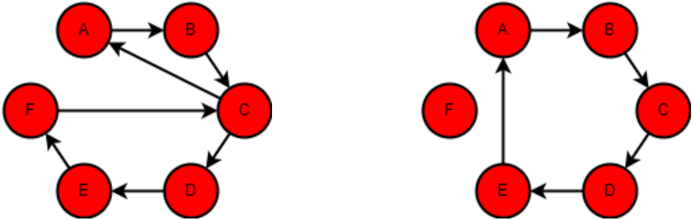
A walk/circuit which visits every edge exactly once.

Euler's Theorem

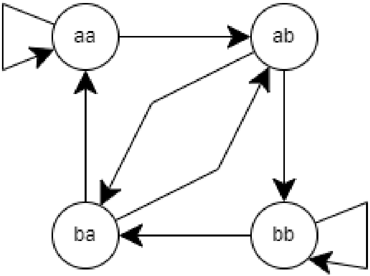
Theorem

A graph contains a Eulerian circuit if and only if

- *the edges are connected and*
- *the nodes are balanced.*

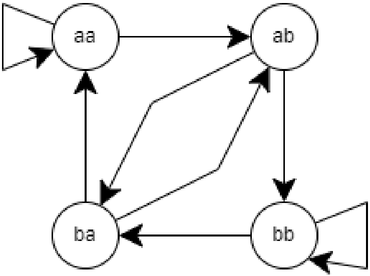


De Bruijn Graphs

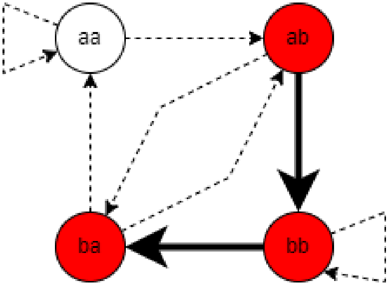


Complete order 3 de Bruijn graph
over alphabet $\Sigma = \{a, b\}$

De Bruijn Graphs



Complete order 3 de Bruijn graph over alphabet $\Sigma = \{a, b\}$



Order 3 de Bruijn graph of *abba*

Outline

Problem Definition

Problem (Eulerian Extension)

We are given a multigraph $G = (V, E)$ *where* $V \subseteq \mathcal{V}$ *and a set of forbidden edges* $F \subseteq \mathcal{V} \times \mathcal{V}$*. Find a minimum multiset of feasible edges* $A \subseteq (\mathcal{V} \times \mathcal{V}) \setminus F$ *and a set of nodes* $B \subseteq \mathcal{V}$ *such that*

- $(V \cup B, E \cup A)$ *is connected and*
- $(V \cup B, E \cup A)$ *is balanced.*

CWI

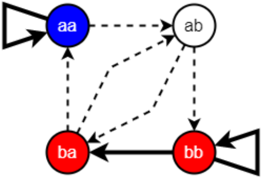
Our Setting

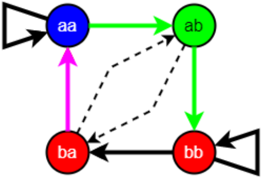
$G = (V, E)$ is a de Bruijn graph of strings

Our Setting

$G = (V, E)$ is a de Bruijn graph of strings

Extend-DBG $\mathcal{V} = \Sigma^{k-1}$ and F is all edges not in the complete de Bruijn graph.

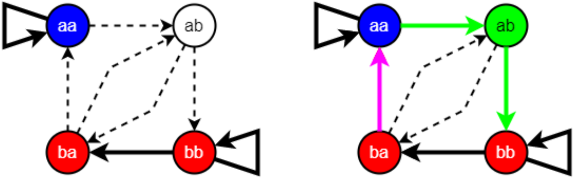




Our Setting

$G = (V, E)$ is a de Bruijn graph of strings

Extend-DBG $\mathcal{V} = \Sigma^{k-1}$ and F is all edges not in the complete de Bruijn graph.

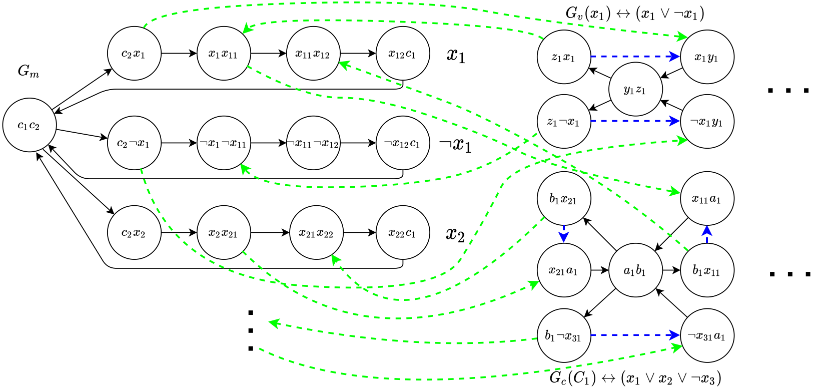


R-Extend-DBG $\mathcal{V} = V$ and F is all edges not in the complete de Bruijn graph.

Outline

Theorem

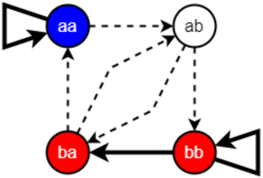
Restricted Eulerian Extension is NP-hard (even if the graph is a de Bruijn graph).



Outline

Connect and Balance (CAB)

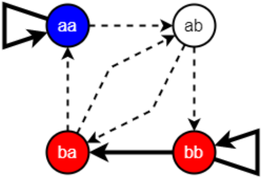
Idea

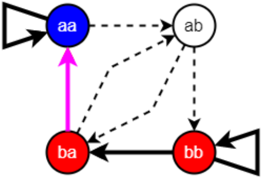


Connect and Balance (CAB)

Idea

1. Connect the graph

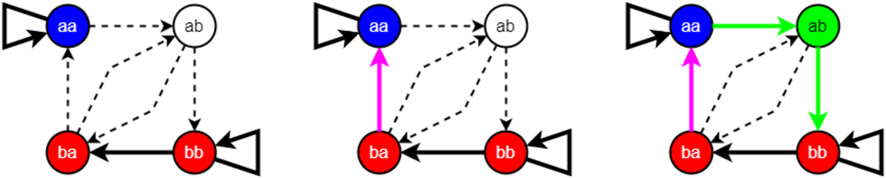




Connect and Balance (CAB)

Idea

1. Connect the graph
2. Balance the graph

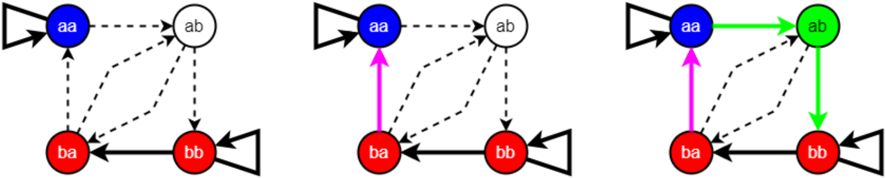


CWI

Connect and Balance (CAB)

Idea

1. Connect the graph
2. Balance the graph



Remark

We do not solve the Eulerian Extension problem optimally.

Connecting

Kruskal's algorithm:

- Connect closest components
- Optimal for minimum spanning tree

Connecting

Idea 1

Use graph algorithms.

- Complete de Bruijn graph has $|\Sigma|^{k-1}$ nodes

Connecting

Idea 1

Use graph algorithms.

- Complete de Bruijn graph has $|\Sigma|^{k-1}$ nodes

Idea 2

Use string algorithms to find the minimum distance between all pairs of nodes in $G = (V, E)$.

- $O(k|V|^2)$

Connecting

Idea 1

Use graph algorithms.

– Complete de Bruijn graph has $|\Sigma|^{k-1}$ nodes

Idea 2

Use string algorithms to find the minimum distance between all pairs of nodes in $G = (V, E)$.

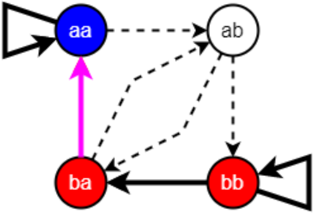
– $O(k|V|^2)$

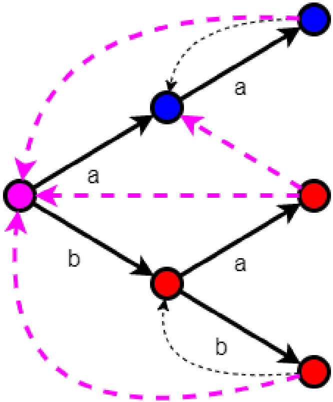
Idea 3

Use an automaton to simultaneously compute all overlaps and than go through them from longest to shortest.

+ $O(k|V|\log|V| + |E|)$

Aho-Corasick Automaton





Balancing

$d^+(v)$ = number of outgoing edges
$d^-(v)$ = number of incoming edges

Balancing

$d^+(v)$ = number of outgoing edges
$d^-(v)$ = number of incoming edges

Two types of unbalanced nodes:

- $Z^+ = \{v \mid d^+(v) > d^-(v)\}$
- $Z^- = \{v \mid d^-(v) > d^+(v)\}$

Balancing

$d^+(v)$ = number of outgoing edges
$d^-(v)$ = number of incoming edges

Two types of unbalanced nodes:

- $Z^+ = \{v \mid d^+(v) > d^-(v)\}$
- $Z^- = \{v \mid d^-(v) > d^+(v)\}$

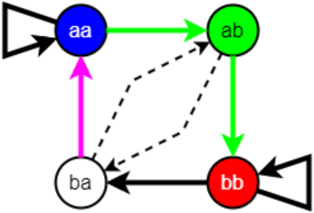
Idea
Use a similar automaton with only links from Z^- to Z^+.

Balancing

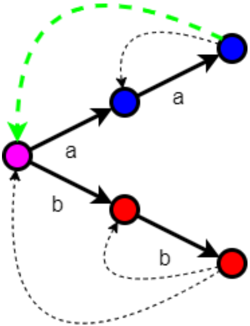
$d^+(v)$ = number of outgoing edges
$d^-(v)$ = number of incoming edges

Two types of unbalanced nodes:

- $Z^+ = \{v \mid d^+(v) > d^-(v)\}$
- $Z^- = \{v \mid d^-(v) > d^+(v)\}$



Idea
Use a similar automaton with only links from Z^- to Z^+.



Time Complexity:
$O(|V|k + |E|)$

Outline

Results

Algorithm: CAB (connect and balance)

Results

Algorithm: CAB (connect and balance)

Benchmarks:

Results

Algorithm: CAB (connect and balance)

Benchmarks:

- MGR (multi-SCS greedy)

Results

Algorithm: CAB (connect and balance)

Benchmarks:

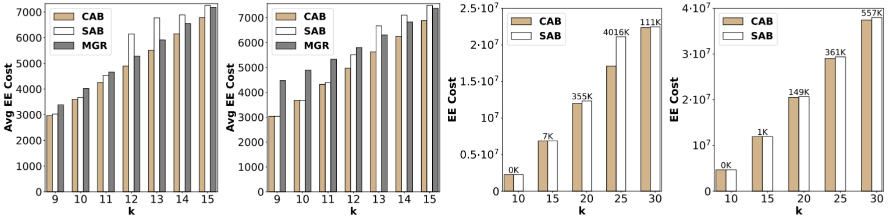
- MGR (multi-SCS greedy)
- SAB (SCS and balance)

Results

Algorithm: CAB (connect and balance)

Benchmarks:

- MGR (multi-SCS greedy)
- SAB (SCS and balance)



(a) STA samples (b) RHO samples (c) STA (d) RHO