Succinct Representation of Separable Graphs

Arash Farzan  
Max-Planck-Institute for Computer Science

Guy E. Blelloch  
Computer Science Department, Carnegie Mellon University
Overview

• Preliminaries:
  – succinctness
  – separability
  – problem formulation
• Motivation
• Related work
• Succinct representation of separable graphs
• Succinct representation of planar maps
• Conclusion and open problems
Succinct Data Structures
Succinct Data Structures

- Highly space-efficient (close to the information-theory minimum):
  - compact data structures: $O(\text{min})$
  - implicit data structures: $\text{min} + O(1)$
  - succinct data structures: $\text{min} + o(\text{min})$
Succinct Data Structures

- Highly space-efficient (close to the information-theory minimum):
  - compact data structures: $O(\min)$
  - implicit data structures: $\min + O(1)$ (different model)
  - succinct data structures: $\min + o(\min)$
Succinct Data Structures

• Highly space-efficient (close to the information-theory minimum):
  – compact data structures: $O(\text{min})$
  – implicit data structures: $\text{min} + O(1)$ (different model)
  – succinct data structures: $\text{min} + o(\text{min})$

• Queries: constant time
Succinct Data Structures

• Highly space-efficient (close to the information-theory minimum):
  - compact data structures: $O(\min)$
  - implicit data structures: $\min + O(1)$ (different model)
  - succinct data structures: $\min + o(\min)$

• Queries: constant time

• $\log(n)$-word RAM model
Separable Graphs
Definition
Separable Graphs
Definition

Separator $S$: 
Separable Graphs

Definition

Separator S:
Separable Graphs

Definition

Separator $S$:

- A family of graphs $G$ is defined as separable if:
A family of graphs $G$ is defined as separable if:

- it is closed under taking subgraphs (monotone), and
Separable Graphs

Definition

Separator $S$:

- A family of graphs $G$ is defined as separable if:
  - it is closed under taking subgraphs (monotone), and
  - satisfies the $n^c$-separator theorem (for some constant $c < 1$):
Separable Graphs
Definition

Separator $S$:

- A family of graphs $\mathcal{G}$ is defined as separable if:
  - it is closed under taking subgraphs (monotone), and
  - satisfies the $n^c$-separator theorem (for some constant $c < 1$):
    - there is a constant $\alpha < 1$ such that each member graph $G \in \mathcal{G}$ with $n$ vertices has a separator $S$ of size $|S| = O(n^c)$ which divides the vertices into parts $A, B$ each of which contains at most $\alpha n$ vertices ($|A| < \alpha n, |B| < \alpha n$).
Separable Graphs
Definition

Separator S:

- A family of graphs $\mathcal{G}$ is defined as separable if:
  - it is closed under taking subgraphs (monotone), and
  - satisfies the $n^c$-separator theorem (for some constant $c < 1$):
    - there is a constant $\alpha < 1$ such that each member graph $G \in \mathcal{G}$ with $n$ vertices has a separator $S$ of size $|S| = O(n^c)$ which divides the vertices into parts $A, B$ each of which contains at most $\alpha n$ vertices ($|A| < \alpha n, |B| < \alpha n$).
- A graph is separable if it belongs to a separable family of graphs.
Succinct Graphs:
Problem Definition
Succinct Graphs: Problem Definition

- Succinctly represent a given an unlabeled and undirected graph answer in constant time:
Succinct Graphs: Problem Definition

- Succinctly represent a given an unlabeled and undirected graph answer in constant time:
  - adjacency queries: is \((v, w)\) an edge?
Succinct Graphs: Problem Definition

- Succinctly represent a given an unlabeled and undirected graph answer in constant time:
  - adjacency queries: is \((v, w)\) an edge?
  - neighborhood queries: report neighbors of vertex \(v\) in constant time per neighbor.
Succinct Graphs: Problem Definition

- Succinctly represent a given an unlabeled and undirected graph answer in constant time:
  - adjacency queries: is \((v, w)\) an edge?
  - neighborhood queries: report neighbors of vertex \(v\) in constant time per neighbor.
  - degree queries: report the degree of a vertex.
Succinct Graphs: Problem Definition

- Succinctly represent a given an unlabeled and undirected graph answer in constant time:
  - adjacency queries: is \((v, w)\) an edge?
  - neighborhood queries: report neighbors of vertex \(v\) in constant time per neighbor.
  - degree queries: report the degree of a vertex.

- The representation has functionality of both an adjacency matrix and an adjacency list representation.
Motivation
Motivation

• Why separable graphs?
Motivation

• Why separable graphs?
  – They include many interesting family of graphs:
Motivation

- Why separable graphs?
  - They include many interesting family of graphs:
    - Bounded-genus and especially planar graphs
Motivation

- Why separable graphs?
  - They include many interesting family of graphs:
    - Bounded-genus and especially planar graphs
    - Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
Motivation

• Why separable graphs?
  - They include many interesting family of graphs:
    ▶ Bounded-genus and especially planar graphs
    ▶ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ▶ Most 3-dimensional meshes
Motivation

• Why separable graphs?
  – They include many interesting family of graphs:
    ‣ Bounded-genus and especially planar graphs
    ‣ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ‣ Most 3-dimensional meshes
    ‣ Most Nearest neighbor graphs in 3-dimensions
Motivation

• Why separable graphs?
  – They include many interesting family of graphs:
    ▶ Bounded-genus and especially planar graphs
    ▶ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ▶ Most 3-dimensional meshes
    ▶ Most Nearest neighbor graphs in 3-dimensions
    ▶ The Web graph
Motivation

• Why separable graphs?
  - They include many interesting family of graphs:
    ‣ Bounded-genus and especially planar graphs
    ‣ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ‣ Most 3-dimensional meshes
    ‣ Most Nearest neighbor graphs in 3-dimensions
    ‣ The Web graph

• Why such succinct representation?
Motivation

• Why separable graphs?
  – They include many interesting family of graphs:
    ▸ Bounded-genus and especially planar graphs
    ▸ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ▸ Most 3-dimensional meshes
    ▸ Most Nearest neighbor graphs in 3-dimensions
    ▸ The Web graph

• Why such succinct representation?
  – Many applications involve representing graphs whose size are increasingly growing.
Motivation

• Why separable graphs?
  - They include many interesting family of graphs:
    ‣ Bounded-genus and especially planar graphs
    ‣ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ‣ Most 3-dimensional meshes
    ‣ Most Nearest neighbor graphs in 3-dimensions
    ‣ The Web graph

• Why such succinct representation?
  - Many applications involve representing graphs whose size are increasingly growing.
  - Space challenge: maintain a compressed representation while supporting dynamic queries efficiently.
Motivation

• Why separable graphs?
  - They include many interesting family of graphs:
    ‣ Bounded-genus and especially planar graphs
    ‣ Some useful non-planar graphs such as the road networks, and utility-distribution networks are separable.
    ‣ Most 3-dimensional meshes
    ‣ Most Nearest neighbor graphs in 3-dimensions
    ‣ The Web graph

• Why such succinct representation?
  - Many applications involve representing graphs whose size are increasingly growing.
  - Space challenge: maintain a compressed representation while supporting dynamic queries efficiently.
  - Adjacency, neighborhood, and degree queries are natural.
Related Work

• Unstructured graphs with $n$ vertices and $m$ edges are hardly compressible as the information theoretic min is $\left\lceil \log \left( \frac{n}{2m} \right) \right\rceil$.
  - Blandford et al. [SODA’03] achieve this by a constant factor.
  - Raman and Rao [SODA’04] improve the constant factor to two.
  - F., Munro [ESA’08] improve it to $(1+\epsilon)$ and show this is optimal.
Related Work

- Unstructured graphs with \( n \) vertices and \( m \) edges are hardly compressible as the information theoretic min is \( \left\lceil \lg \left( \frac{n}{m} \right) \right\rceil \)
  - Blandford et al. [SODA’03] achieve this by a constant factor.
  - Raman and Rao [SODA’04] improve the constant factor to two.
  - F., Munro [ESA’08] improve it to \((1+\epsilon)\) and show this is optimal.

- Therefore, structured graphs having a particular combinatorial property is of interest:
  - limited arboricity
  - c-decomposable
  - planar graphs
  - separable graphs
Related Work

• Unstructured graphs with n vertices and m edges are hardly compressible as the information theoretic min is $\left\lceil \log_2 \left( \frac{n}{m} \right) \right\rceil$
  - Blandford et al. [SODA’03] achieve this by a constant factor.
  - Raman and Rao [SODA’04] improve the constant factor to two.
  - F., Munro [ESA’08] improve it to $(1+\varepsilon)$ and show this is optimal.

• Therefore, structured graphs having a particular combinatorial property is of interest:
  - limited arboricity
  - c-decomposable
  - planar graphs
  - separable graphs

• For planar and separable graphs, the best representations that support the set of queries in constant time require a constant factor more than the optimal space needed.
The Succinct Representation
The Succinct Representation

Based on repeated decomposition of the graph $G$ according to the separator $S$ into smaller subgraphs $G_1, G_2$: 
The Succinct Representation

Based on repeated decomposition of the graph $G$ according to the separator $S$ into smaller subgraphs $G_1, G_2$:
The Succinct Representation
The Succinct Representation

- Where there are $n^c$-separators, we define $\delta = \frac{2}{1+c}$ and repeat the separator-based decomposition till the subgraphs of size at most $(\lg n)^\delta$: mini-graphs.
The Succinct Representation

- Where there are \( n^c \)-separators, we define \( \delta = \frac{2}{1-c} \) and repeat the separator-based decomposition till the subgraphs of size at most \((\lg n)^\delta\): mini-graphs.

- Each mini-graph is decomposed analogously to obtain subgraphs of size at most \( \frac{\lg n}{\lg \lg n} \): micro-graphs.
The Succinct Representation
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small-enough to be catalogued by a look-up table.
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small-enough to be catalogued by a look-up table.
- The representation of a micro-graph is a reference to within this table (takes the dominant term of space: linear in $n$).
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small-enough to be catalogued by a look-up table.
- The representation of a micro-graph is a reference to within this table (takes the dominant term of space: linear in $n$).
- Separator vertices are duplicated by each decomposition.
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small-enough to be catalogued by a look-up table.
- The representation of a micro-graph is a reference to within this table (takes the dominant term of space: linear in $n$).
- Separator vertices are duplicated by each decomposition.
- Each repetition of a vertex receives three labels: micro-graph label, mini-graph label, and a graph label.
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small enough to be catalogued by a look-up table.
- The representation of a micro-graph is a reference to within this table (takes the dominant term of space: linear in $n$).
- Separator vertices are duplicated by each decomposition.
- Each repetition of a vertex receives three labels: micro-graph label, mini-graph label, and a graph label.
- Sophisticated succinct structures are used to translate between these labels.
The Succinct Representation

- The representation of the graph consists of those of mini-graphs which in turn consist of those of micro-graphs.
- Micro-graphs are small-enough to be catalogued by a look-up table.
- The representation of a micro-graph is a reference to within this table (takes the dominant term of space: linear in \( n \)).
- Separator vertices are duplicated by each decomposition.
- Each repetition of a vertex receives three labels: micro-graph label, mini-graph label, and a graph label.
- Sophisticated succinct structures are used to translate between these labels.

**Technical Lemma:**
In the graph, there is a sublinear number of duplicates across mini-graphs and in any mini-graph there is a sublinear number of duplicates across micro-graphs.
Query Support
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
• Descriptions of support for degree and neighborhood queries, we skip.
• Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
  - If a vertex has duplicates across different mini-graphs we explicitly list its out-neighbors (we do the same across micro-graphs)
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
  - If a vertex has duplicates across different mini-graphs we explicitly list its out-neighbors (we do the same across micro-graphs)
  - To determine whether (u,v) is an edge we first check whether u→v is an edge (and then analogously check if v→u is an edge).
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
  - If a vertex has duplicates across different mini-graphs we explicitly list its out-neighbors (we do the same across micro-graphs).
  - To determine whether \((u,v)\) is an edge we first check whether \(u \rightarrow v\) is an edge (and then analogously check if \(v \rightarrow u\) is an edge).
  - If \(u\) is repeated across mini/micro-graphs then its out-neighbors are explicitly listed which we check against \(v\).
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
  - If a vertex has duplicates across different mini-graphs we explicitly list its out-neighbors (we do the same across micro-graphs).
  - To determine whether \((u,v)\) is an edge we first check whether \(u \rightarrow v\) is an edge (and then analogously check if \(v \rightarrow u\) is an edge).
  - If \(u\) is repeated across mini/micro-graphs then its out-neighbors are explicitly listed which we check against \(v\).
  - Otherwise, \(u\) occurs in a single micro-graph. For \(u \rightarrow v\) to be an edge, \(v\) must also occur in the micro-graph. Using the stored structures, we determine the micro-graph label of the proper duplicate of \(v\).
Query Support

- Descriptions of support for degree and neighborhood queries, we skip.
- Adjacency queries:
  - Separable graphs can be oriented such that vertices have constant out-degree.
  - If a vertex has duplicates across different mini-graphs we explicitly list its out-neighbors (we do the same across micro-graphs).
  - To determine whether (u,v) is an edge we first check whether u→v is an edge (and then analogously check if v→u is an edge).
  - If u is repeated across mini/micro-graphs then its out-neighbors are explicitly listed which we check against v.
  - Otherwise, u occurs in a single micro-graph. For u→v to be an edge, v must also occur in the micro-graph. Using the stored structures, we determine the micro-graph label of the proper duplicate of v.
  - We use the look-up table to determine if v→u is an edge.
Result
Result

Theorem:

Any monotone family of separable graphs with entropy $H(n)$, where $n$ is the number of vertices, can be succinctly encoded in $H(n) + o(n)$ bits such that adjacency, neighborhood, and degree queries are supported in constant time.
Representing Planar Maps
Representing Planar Maps

- We can succinctly represent planar graphs.
Representing Planar Maps

- We can succinctly represent planar graphs.
- Planar maps: planar graph + embedding.
Representing Planar Maps

- We can succinctly represent planar graphs.
- Planar maps: planar graph + embedding.
- We can enhance the representation to also reflect the given embedding.
Representing Planar Maps

- We can succinctly represent planar graphs.
- Planar maps: planar graph + embedding.
- We can enhance the representation to also reflect the given embedding.
- We introduce dummy nodes by subdividing edges.
Representing Planar Maps

- We can succinctly represent planar graphs.
- Planar maps: planar graph + embedding.
- We can enhance the representation to also reflect the given embedding.
- We introduce dummy nodes by subdividing edges.
- We cannot afford to store all dummy nodes: only duplicate dummy nodes are stored.
Result
Theorem:

A planar map $G$ with $n$ vertices can be succinctly encoded in $\mathcal{H}_p(n) + o(n)$ bits such that adjacency, degree, and neighborhood queries (according to the combinatorial planar embedding of $G$) are supported in constant time.
Conclusion
Conclusion

• We considered separable graphs that satisfy the $n^c$-separator theorem.
Conclusion

• We considered separable graphs that satisfy the $n^c$-separator theorem.

• We gave a representation with a storage requirement that is only $o(n)$ bits more than the information theory minimum.
Conclusion

• We considered separable graphs that satisfy the $n^c$-separator theorem.
• We gave a representation with a storage requirement that is only $o(n)$ bits more that the information theory minimum.
• The representation supports adjacency, degree, and neighborhood queries in constant time.
Conclusion

• We considered separable graphs that satisfy the $n^c$-separator theorem.
• We gave a representation with a storage requirement that is only $o(n)$ bits more than the information theory minimum.
• The representation supports adjacency, degree, and neighborhood queries in constant time.
• The representation can be enhanced to represent planar maps. The neighborhood queries follow the order dictated by the combinatorial planar embedding.
Conclusion

- We considered separable graphs that satisfy the $n^c$-separator theorem.
- We gave a representation with a storage requirement that is only $o(n)$ bits more than the information theory minimum.
- The representation supports adjacency, degree, and neighborhood queries in constant time.
- The representation can be enhanced to represent planar maps. The neighborhood queries follow the order dictated by the combinatorial planar embedding.
- **Future Work**: dynamic separable graphs