Dynamic Sketching for Graph Optimization Problems with Application to Cut-Preserving Sketches

Sepehr Assadi  Sanjeev Khanna  Yang Li  Val Tannen
Department of Computer and Information Science, University of Pennsylvania

1. Dynamic Sketching

The model is defined as follows:

- **Input**: A function \( f \) on \( \{0, 1\}^n \) and a partial input to the function that specifies input value on \( n-k \) positions (static data), but leaves it unspecified for \( k \) positions (dynamic data).
- **Goal**: To compress the static data so that the value of \( f \) can be recovered for any assignment to dynamic variables.

We are interested in the space complexity of the sketch.

**Compact sketches.** Obvious approaches to dynamic sketching require \( \min(n-k, 2^k) \) space. A desired approach is a compact sketch, i.e., a sketch of size \( \text{poly}(k) \).

Not all functions admit compact sketches (follows from a simple counting argument) so what natural problems admit compact sketches?

2. Graph Optimization Problems

A graph \( G(V, E) \) with a set \( T \) of \( k \) terminals.

- **Static data**: the set of edges originally in \( E \).
- **Dynamic data**: the set \( E_T \) of edges between the terminals.
- **Goal**: create a sketch to solve a given problem in \( G(V, E \cup E_T) \).

**Example.** An instance of \( s-t \) shortest path: here, terminals are in red and the query (i.e., dynamic data) is \( \{(v_2, v_3), (s, v_5)\} \).

3. Our Results

- **The minimum spanning tree problem**
  - An \( O(k) \) size dynamic sketch.
- **The maximum matching problem**
  - An \( O(k^2) \) size dynamic sketch (randomized).
  - The sketch size is provably optimal within a logarithmic factor.
- **The \( s-t \) maximum flow problem**
  - Uncapacitated version (i.e., \( s-t \) edge connectivity): \( O(k^2) \) size dynamic sketch (randomized).
  - Capacitated version: \( \min(O(n), 2^{O(k)}) \) lower bound.
- **Application to cut-preserving sketches.**

4. Connection to Existing Models

**Linear sketch.** A linear sketch for a function \( f \) with space \( s \) implies a dynamic sketch for \( f \) with space \( s \).

**Streaming.** A single-pass streaming algorithm for a function \( f \) with space \( s \) implies a dynamic sketch for \( f \) with space \( s \).

**Remark:** Most graph optimization problems do not admit sublinear space (in \( n \)) streaming algorithms or linear sketches.

5. Maximum Matching

An algebraic compression scheme based on the Tutte matrix.

**Example.** A bipartite graph \( G(L, R, E) \)

**Example.** The Tutte matrix:

\[
M = \begin{bmatrix}
0 & x_{1,2} & 0 & x_{1,4} \\
x_{2,1} & 0 & x_{2,3} & 0 \\
0 & 0 & x_{3,3} & 0 \\
x_{4,1} & 0 & 0 & x_{4,4}
\end{bmatrix}
\]

An algorithm for detecting a perfect matching originally introduced by Lovasz:

- Evaluate formal variables in the Tutte matrix by random numbers in \( [1, \ldots, n^2] \).
- If \( G \) has a perfect matching, determinant of this new matrix is non-zero w.p. \( 1 - 1/n \).

**Compression.**

- Consider the Tutte matrix with all edges between terminals present.
- Evaluate non-terminal edges with random numbers in \( [1, \ldots, n^2] \).
- Perform Gaussian elimination to make the matrix lower triangular (except for the symbolic terminal-edges block).

**Extraction.** Zero-out terminal edges not present; evaluate remaining variables with random numbers and compute the determinant.

**Example.** Input graph with terminals in red.

6. Cut-Preserving Sketches

For a graph \( G(V, E) \) and a set \( T \) of terminals, a cut-preserving sketch is a data-structure that answers for each \( A, B \subseteq T \), value of a minimum cut between \( A \) and \( B \).

**Theorem.** Let \( C \) be the total capacity of edges incident on terminals.

- There is a cut-preserving sketch of size \( O(kC) \) for directed graphs.
- Any cut-preserving sketch, even for undirected graphs, requires \( \Omega(C) \) space.

**Connection to dynamic sketching.** A cut-preserving sketch for any graph \( G \)

\[
M = \begin{bmatrix}
x_{1,1} & x_{1,2} & 0 & x_{1,4} \\
x_{2,1} & x_{2,2} & x_{2,3} & 0 \\
0 & 0 & x_{3,3} & 0 \\
x_{4,1} & 0 & 0 & x_{4,4}
\end{bmatrix}
\]

An algorithm for computing a cut-preserving sketch:

- Merge above reduction with an adaption of a well-known reduction from \( s-t \) maximum flow to maximum matching.
- Use dynamic sketching scheme for maximum matching to answer cut queries.

A cut-preserving sketch: can be obtained via a \( s-t \) maximum flow dynamic sketch for a graph \( G(G) \) (here, terminal edges mimic the query \( A = \{v_1, v_3\}, B = \{v_5\} \)).