Linear Algebra is the Right Way to Think About Graphs
Carl Yang\textsuperscript{1,2}, Aydin Buluç\textsuperscript{2,3} and John D. Owens\textsuperscript{1}
\textsuperscript{1} Department of Electrical and Computer Engineering, UC Davis
\textsuperscript{2} Computational Research Division, LLNL
\textsuperscript{3} Department of Electrical Engineering and Computer Sciences, UC Berkeley

Problem
- High-performance graph processing is an important problem
- Increasing graph sizes require accelerators (e.g. GPUs)
- Mapping irregular graph problems onto the GPU is hard
- What is the “best” interface for graph processing?

Matrix-graph duality
\begin{figure}
\centering
\includegraphics[width=\textwidth]{matrix_graph_duality.png}
\caption{We rely on this connection first noted by König.}
\end{figure}

3 Types of Sparsity
\begin{figure}
\centering
\includegraphics[width=\textwidth]{3_types_of_sparsity.png}
\caption{We take advantage of 3 types of sparsity in our work: (a) matrix sparsity or lefthand-side sparsity, (b) input sparsity or righthand-side sparsity, (c) output sparsity or mask sparsity. Their complexity results are given below.}
\end{figure}

Complexity Results
\begin{table}
\centering
\begin{tabular}{|c|c|}
\hline
Operation & Expected Results \\
\hline
SpMV & no mask \(O(d m n)\) \\
SpM & masked \(O(d n m)\) \\
SpMSpV & no mask \(O(d n m \log n m)\) \\
SpGEMM & masked \(O(d n m \log n m f)\) \\
\hline
\end{tabular}
\caption{Four sparse matvec variants and their associated cost, measured in terms of number of expected memory accesses into the \(M \times b + M\) sparse matrix \(A\) required. We assume the sparse matrix is Erdős-Rényi with mean degree \(d\). The \(m\) and \(f\) refer to the mask and righthand-side vector respectively.}
\end{table}

Direction-optimized BFS
\begin{figure}
\centering
\includegraphics[width=\textwidth]{direction_optimized_bfs.png}
\caption{3 optimizations making up direction-optimized BFS.}
\end{figure}

Ingredient I: Operations
\begin{figure}
\centering
\includegraphics[width=\textwidth]{ingredient_i_operations.png}
\caption{Operations + Operators = GraphBLAS}
\end{figure}

Ingredient II: Operators
\begin{table}
\centering
\begin{tabular}{|c|c|}
\hline
Semiring & Application \\
\hline
Real field \{+, \times, \mathbb{R}\} & Classical linear algebra \\
Boolean \{\{0, 1\}\} & Graph connectivity \\
Tropical \{min, +, \mathbb{R} \cup \{\infty\}\} & Shortest path \\
Max-plus \{max, +, \mathbb{R}\} & Graph matching \\
Min-times \{min, \times, \mathbb{R}\} & Maximal independent set \\
\hline
\end{tabular}
\caption{Semiring notation: (Add, Multiply, Domain)}
\end{table}

Conclusion
- By its nature, linear algebra can be considered very concise
- GraphBLAS is an open standard, so there is consensus on a standard C API and it is possible to build backend-agnostic graph framework
- Our work focuses on the other two metrics:
  - We show that linear algebra is expressible enough to express a fairly complicated optimization: direction-optimized BFS
  - We also demonstrate that our implementation is performant compared to traditional, vertex-centric graph frameworks

Contact Information
- Web: http://www.ece.ucdavis.edu/~ctcyang/
- Email: ctcyang@ucdavis.edu
- Phone: +1 (916) 802-8178