An Interview with Vladimir Trifonov (2005 Danny Lewin Best Student Paper Award Winner)

Tim Roughgarden*

December 26, 2005

The 2005 Danny Lewin Prize for the best student paper at STOC was awarded to Vladimir Trifonov, of the University of Texas at Austin, for his paper "An $O(\log n \log\log n)$ Space Algorithm for Undirected st-Connectivity" [1]. Hal Gabow recently started the tradition of interviewing the Danny Lewin Prize winner and publishing the interview in SIGACT News. (An interview with the 2003 winner Tom Hayes can be found in the September 2003 issue.) I had the pleasure of (electronically) chatting with Vladimir about his work; a transcript of our conversation follows.

TR: *Congratulations on your award! Could you begin by giving us a brief overview of the contribution of your paper?*

VT: Thank you. On my behalf I would like to thank the STOC program committee for giving me the Danny Lewin award for this year—it is a great honor. My work shows that undirected st-connectivity, the problem of checking whether two given vertices of an undirected graph with n vertices are connected by a path, can be solved using only $O(\log n \log\log n)$ space.

The problem is solved easily with depth-first search if we allow polynomial time and linear space. Achieving $O(\log^2 n)$ space is also not too hard by doing repeated squaring as was shown by Savitch in 1970, but beyond that things get complicated and progress has been very slow, despite the considerable interest. Aleliunas et al. showed in 1979 that if we allow randomness, then undirected st-connectivity can be solved in $O(\log n)$ space by just walking randomly in the graph. Using Nisan's pseudorandom generator for space-bounded computation, Nisan, Szemeredi, and Wigderson showed in 1992 a deterministic algorithm using $O(\log^{3/2} n)$ space. Building on the space-efficient simulation of space-bounded randomized computation due to Saks and Zhou, the bound was improved by Armoni, Ta-Shma, Wigderson, and Zhou to $O(\log^{4/3} n)$ in 1997. Finally, Reingold showed that employing a zig-zag graph product, the optimal $O(\log n)$ space is obtainable by a deterministic algorithm, a marvelous and long anticipated result which appeared simultaneous to my result in this year's STOC.

*Department of Computer Science, Stanford University, 462 Gates Building, Stanford CA 94305. Email: tim@cs.stanford.edu.

Besides being a considerable improvement over the previous best bound and in light of Reingold's result, my result is interesting for the technique used, which although in some sense is another spin on the "repeated squaring done efficiently" theme, differs significantly from what was attempted before. At the bottom of this technique is an approach to solving undirected st-connectivity, suggested by Professor Vijaya Ramachandran, which involves simulating space-efficiently a parallel algorithm for the same problem.

TR: *Could you elaborate some on the similarities and differences between your techniques and those developed simultaneously and independently by Reingold?*

VT: It all begins with the repeated squaring algorithm of Savitch. In this algorithm, the graph is squared, i.e. every path of length two is substituted by an edge, $O(\log n)$ times. The immediate implementation of this idea takes $O(\log n)$ space for each squaring, for a total of $O(\log^2 n)$ space. This works also for directed graphs.

It is not hard to see that the space for squaring is actually $O(\log d)$, where d is the degree of the graph. Unfortunately the squaring graph operation squares the degree of the graph, so even if we start with a constant degree graph, after $O(\log n)$ squarings, the degree will be polynomial in n. Reingold's idea is that for undirected graphs there is another graph operation, called the zig-zag product, which applied to a graph makes its degree some fixed constant, while not increasing its size too much. Besides taking care of the squaring of the degree resulting from the squaring graph operation, the zig-zag product maintains the progress made by the squaring in terms of the expansion of the graph. So the repeated squaring algorithm turns roughly into $O(\log n)$ stages of squaring followed by zig-zagging, which can be done in total space $O(\log n)$ and results in an expander graph which has constant degree and diameter $O(\log n)$.

The relation of our algorithm to the repeated squaring algorithm of Savitch comes from a different point of view. The idea is that if H is a connected subgraph of the given undirected graph, we can choose one of the vertices of H to be its representative and remove the other vertices of H, after they transfer their edges to the representative vertex. Thus, if the whole graph is partitioned into connected subgraphs, which we call neighborhoods, of size at least two, then after the described contraction operation the size of the graph will decrease by a factor of at least two, while the connectivity information is essentially preserved. Thus after $O(\log n)$ contractions, every connected component of the given graph is reduced to a single vertex. Also it is not hard to see that given $O(\log n)$ space we can partition the current graph into neighborhoods of size at least two and then contract each neighborhood. The analogy with the Savitch algorithm is that while after a squaring, an edge represents a path of length two in the old graph, after a contraction, a vertex represents at least two vertices from the old graph.

The algorithm of Nisan-Szemeredi-Wigderson is based on the same idea except that in space $O(\log n)$ they manage to obtain neighborhoods of size $\Omega(2^{\log^{1/2} n})$. This algorithm uses the pseudorandom generator for space-bounded computation of Nisan, which allows it to explore in space $O(\log n)$ a much larger neighborhood of a vertex than just its immediate neighbors.

While our algorithm follows the same contraction paradigm, we do not use Nisan's pseudorandom generator to determine the neighborhoods. Instead we base our algorithm on a parallel algorithm due to Chong and Lam which solves undirected st-connectivity in time $O(\log n \log \log n)$. In fact, our algorithm is just a space-efficient implementation of this parallel algorithm. The idea of this approach was suggested by Prof. Ramachandran. In our initial work in 2000 she conjectured the possibility of simulating space-efficiently the algorithm of Chong and Lam and other efficient parallel algorithms for undirected st-connectivity.

A main component of the Chong-Lam algorithm is determining whether a subtree of a given graph has size at most s, where s is a parameter which changes in the course of the algorithm and could be much smaller than n. Chong and Lam show that this can be done in parallel time $O(\log s)$. We can easily do this in space $O(\log n)$, but unfortunately this bound does not depend on s, which is important for the analysis of the Chong-Lam algorithm, and results in an $O(\log^2 n)$ space algorithm. The major bottleneck is that we have to store a vertex label locally and in my work I show that using the exploration walks on trees defined by Koucký (essentially Euler tours for trees) we can resolve this bottleneck and simulate the Chong-Lam algorithm in space $O(\log n \log \log n)$.

Notice that it is known that space and parallel time are closely related, in particular an algorithm for the EREW PRAM (exclusive-read, exclusive-write, parallel RAM) model (e.g., the Chong-Lam algorithm) running in time $O(T)$ can be simulated in space $O(T^2)$. Thus our space-efficient simulation does much better then the general simulation, but it is specific to the Chong-Lam algorithm.

TR: *Could you talk a bit more about what led you to work on this famous open problem? You say you initially started working on aspects of the problem five years ago?*

VT: I started the Ph.D. program at the University of Texas at Austin in the fall of 1999. In the spring and the summer of 2000 I worked with Prof. Ramachandran on st-connectivity, which she suggested as a research problem. At that time the progress on the space complexity of this problem was lagging significantly behind the progress on its parallel time complexity and her intuition was that we should be able to apply the progress in the parallel time context to the sequential space context. We spent most of the effort on working our her idea about simulating space-efficiently existing parallel algorithms, but although it was clear that we can get $O(\log^2 n)$ space algorithms this way, at the time it was not clear how to get anything which is $o(\log^2 n)$.

After those two semesters with Prof. Ramachandran my interests wandered off in different directions, but never too far from st-connectivity and the somewhat related problem of derandomizing space-bounded computation. In the spring of last year I came back to thinking about st-connectivity. While I was trying a different approach, it struck me that using the exploration walks on trees of Koucký, I can make the idea of Prof. Ramachandran work. As I mentioned, conceptually exploration walks on trees are the same as Euler tours for trees, so the technique is not entirely novel and in some form existed in 2000, but Koucký's observation that exploration walks can be inverted is what helped me make the connection.

TR: *What do you think the future holds for this line of research? Can we expect to have a better understanding of the L vs. RL or even the L vs. NL question a few years from now?*

VT: This year we saw some really exciting developments in understanding the power of logspace bounded computation. I believe that those developments will lead soon to a better understanding of the power of randomness in logspace bounded computation. Concerning the power of non-determinism in such computations, I find the fact that in some sense the distinction between determinism and non-determinism is captured by the distinction between undirected and directed st-connectivity quite intriguing. I strongly believe that a better understanding of logspace bounded computation is one of the central questions of complexity theory and am optimistic about its future. In any case, I am quite happy to have contributed to this understanding.

References

[1] V. Trifonov. An $O(\log n \log \log n)$ Space Algorithm for Undirected st-Connectivity. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 626–633.