Electronic Colloquium on Computational Complexity, Report No. 8 (2014)

Space Complexity of the Directed Reachability Problem over Surface-Embedded Graphs

N. V. Vinodchandran

Abstract. The graph reachability problem, the computational task of deciding whether there is a path between two given nodes in a graph is the canonical problem for studying space bounded computations. Three central open questions regarding the space complexity of the reachability problem over directed graphs are: (1) improving Savitch's $O(\log^2 n)$ space bound, (2) designing a polynomial-time algorithm with $O(n^{1-\epsilon})$ space bound, and (3) designing an *unambiguous* non-deterministic log-space (UL) algorithm. These are well-known open questions in complexity theory, and solving any one of them will be a major breakthrough. We will discuss some of the recent progress reported on these questions for certain subclasses of surface-embedded directed graphs.

Mathematics Subject Classification (2010). Primary 68Q15.

Keywords. graph reachability, space-bounded computation, computational complexity.

1. Introduction

The graph reachability problem, the problem of deciding whether there is a path from a given vertex s to a vertex t in a given graph, is central to space-bounded computations. Various versions of this problem characterize several important space complexity classes. Over directed graphs, it is the canonical complete problem for non-deterministic log-space (NL). The breakthrough result of Reingold implies that the undirected reachability problem characterizes the complexity of deterministic log-space (L) [Rei08]. It is also known that a certain restricted promise version of the reachability problem over directed graphs characterizes randomized log-space computations (RL) [RTV06]. Clearly, progress in space complexity studies is directly related to progress in understanding graph reachability problems. We refer the readers to an excellent (albeit two decades old) survey by Avi Wigderson [Wig92]

Supported in part by the NSF grant CCF-0916525.

ISSN 1433-8092

to further understand the significance of reachability problems in complexity theory.

This article is far from an exhaustive survey on the space complexity of the graph reachability problem. In particular, some of the major progress (such as Reingold's algorithm for undirected graph reachability and Saks and Zhou's deterministic simulation of randomized log-space) will not be discussed here. Instead, we will limit our discussion to some recent progress that the author and his collaborators reported on these questions for certain subclasses of surface-embedded directed graphs. It is mostly an elaboration of the talk that the author gave on Prof. Somenath Biswas's 60th birthday celebration at IIT Kanpur in August of 2012.

Three Central Questions

We first discuss three central questions concerning the space complexity of the directed graph reachability problem. These are well-known and difficult open questions in the area, and progress on any of these is a step towards the much bigger NL vs L question (the first two problems are discussed in Wigderson's 1992 survey [Wig92]). However, the author feels that the known barriers for attacking these problems are much less severe than those known for many difficult open problems in time-bounded complexity classes and circuit lower bounds, and believes that breakthrough progress on these problems can be made in the near future.

Problem 1: Improving Savitch's Bound. About 4 decades ago Savitch proved that the reachability problem over directed graphs with n vertices can be solved in space $O(\log^2 n)$ deterministically [Sav70]. This implies that problems that can be solved nondeterministically in space $s(n)$ have *deterministic* algorithms with $O(s^2(n))$ space bound. Thus, for polynomial space bounds, nondeterminism does not add any additional power to determinism. For the important case when the space bound is $O(\log n)$, Savitch's theorem implies that all problems in NL can be solved deterministically in $O(\log^2 n)$ space. *Is this quadratic blow-up necessary?* This is one of the most important open problems in this topic.

Problem 2: Improving the BBRS bound. The time complexity of Savitch's algorithm is $\Theta(n^{\log n})$ - a super-polynomial bound. The standard breadth first search algorithm (BFS) is another fundamental algorithm for solving graph reachability. BFS can be implemented in linear time but requires linear space. BFS is efficient in time but not in space, and Savitch's algorithm is efficient in space but takes super-polynomial time. Hence a natural and significant question that researchers have considered is whether we can design an algorithm for the directed graph reachability problem that is efficient in both time and space. In particular, can we design a polynomial-time algorithm for the directed graph reachability problem that uses only $O(n^{1-\epsilon})$ space for some constant ϵ? The best known result in this direction is the two decades old bound due to Barnes, Buss, Ruzzo, and Schieber [BBRS98] (which we call the BBRS bound). By cleverly combining BFS and Savitch's algorithm, Barnes, Buss, Ruzzo, and Schieber designed a polynomial-time algorithm for reachability that uses space $O(n/2^{\sqrt{\log n}})$ - a slightly sub-linear function. Improving the BBRS bound remains another significant open question regarding the space complexity of the directed reachability problem.

Problem 3: NL vs UL Problem. UL denotes an unambiguous subclass of NL. A decision problem L is in UL if and only if there exists a nondeterministic log-space machine M deciding L such that, for every instance x, M has at most one accepting computation on input x [BJLR91, ÀJ93]. Thus UL is a complexity class that is sandwiched between NL and L. Is NL=UL? While typically such collapse results are unlikely in complexity theory (and even if they are likely, they are nearly impossible to prove), there is an increasing body of evidence that in this specific case the answer is yes, and the author believes that we might be able to prove this equality using known techniques. Reinhardt and Allender showed using the isolation lemma that in the nonuniform setting NL coincides with UL; that is $\mathsf{NL/poly} = \mathsf{UL/poly}$ [RA00]. Further, in a subsequent paper, Allender, Reinhardt, and Zhou showed that, under a certain hardness assumption the construction given in [RA00] can be *derandomized* to show that $\mathsf{NL} = \mathsf{UL}$ [ARZ99]. Thus it is very likely (at least to the author) that $\mathsf{NL} = \mathsf{UL}$, though we do not yet have a proof. Can we show $\mathsf{NL} = \mathsf{UL}$ unconditionally?

Outline

In the next two sections we will discuss some progress that we have made towards these three open questions - Section 2 on problems 1 and 2, and Section 3 on the NL vs UL problem. All the results discussed in these sections are for directed graphs embedded on topological surfaces. As an aside, in Section 4 we reproduce the proof of the BBRS bound from [BBRS98], partly to bring more attention to this nice algorithm.

2. Space efficient reachability algorithms for graphs with topological structure

An important sub-case of Problem 1 (and Problem 2) is to design reachability algorithms that beat Savitch's bound (respectively, the BBRS bound) for directed graphs with some topological structure (graphs that are embedded on topological surfaces). We will discuss some recent progress reported along this direction. The main results are (1) algorithms that beat both Savitch's bound and the BBRS bound for a subclass of directed *acyclic* graphs parameterized by the number of sources and the genus of the embedding [SBV10, SV12] (2) an algorithm for directed planar reachability that improves on the BBRS bound [INP$^+$13]. The main approach in both these results is that of space-efficient "kernelization".

Kernelization is a known preprocessing technique in designing algorithms (for example in the area of fixed parameter tractability). Kernelization algorithms are reductions from a problem to itself so that the easy part of the instance is abstracted out and the core part is retained in the reduced instance. The hope is that the core part will be of smaller size and hence known algorithms can be applied to this compressed instance yielding algorithms with better complexity. We will first illustrate this in a simple scenario.

Consider a reachability instance $\langle G, s, t\rangle$ where $G = (V, E)$ is a n-vertex graph with the guarantee that it has at most k directed edges (the remaining edges are undirected). Let G_{un} be the undirected graph we get by removing all the directed edges from G. For a directed edge $e = (u, v)$ let $tail(e) = u$ and $head(e) = v$. We will show a simple log-space reduction that takes $\langle G, s, t\rangle$ and produces a reachability instance $\langle G', s', t'\rangle$ where G' is a directed graph with $O(k)$ vertices.

In the reduced graph $G' = (V', E')$, $V' = \{s', t'\} \cup \{v_e \mid e \text{ is a directed edge in } G\}$. The pair $(v_{e_1}, v_{e_2}) \in E'$ if $tail(e_2)$ is in the same connected component as $head(e_1)$ in G_{un}. For a $v_e \in V'$, $(s', v_e) \in E'$ if $tail(e)$ is in the same connected component of s in G_{un} and (v_e, t') if $head(e)$ is in the same connected component of t in G_{un}. Notice that this reduction is log-space since for checking whether two vertices u, v are in the same connected component of G_{un}, we can use Reingold's log-space algorithm for undirected reachability. It is clear that there is a s-t path in G if and only if there is a s'-t' path in G'. Using this reduction together with Savitch's algorithm we get that reachability in graphs with $n^{o(1)}$ directed edges can be solved in $o(\log^2 n)$. Also, by applying BFS to the reduced graph, we get that for any $\epsilon > 0$, reachability in graphs with $O(n^{1-\epsilon})$ directed edges can be solved in polynomial time and $O(n^{1-\epsilon})$ space.

We will now describe the the main kernelization result of [SBV10, SV12] and its application. Let $\mathcal{G}(m, g)$ denote the class of DAGs with at most $m = m(n)$ source vertices embedded on a surface (orientable or non-orientable) of genus at most $g = g(n)$, where n is the number of vertices. Building on [SBV10], in [SV12] we show the following reduction for graphs in $\mathcal{G}(m, g)$.

Theorem 1 ([SV12])**.** *There is a log-space reduction that, given an instance $\langle G, s, t\rangle$ (presented as a combinatorial embedding) where $G \in \mathcal{G}(m, g)$ and s, t are vertices of G, outputs an instance $\langle G', s', t'\rangle$ where G' is a directed graph and s', t' vertices of G', so that (a) there is a directed path from s to t in G if and only if there is a directed path from s' to t' in G', (b) G' has $O(m + g)$ vertices.*

By combining the above reduction with Savitch's theorem (with $m = g = 2^{O(\sqrt{\log n})}$) we get that reachability over graphs with $2^{O(\sqrt{\log n})}$ sources embedded on a surface of genus $2^{O(\sqrt{\log n})}$ can be decided in deterministic log-space. For $m = g = n^{o(1)}$ we get $o(\log^2 n)$ space algorithm for reachability that beats Savitch's bound. For $m = g = O(n^{1-\epsilon})$, we get $O(n^{1-\epsilon})$ space algorithm with polynomial running time for reachability, for any small constant ϵ, improving the BBRS bound.

One of the motivations for investigating the reachability problem for this class of sufrace-embedded graphs comes from earlier work due to Allender, Barrington, Chakraborthy, Datta, and Roy [ABC+09]. In [ABC+09], the authors considered reachability in planar DAGs with a *single* source vertex. They called this class of graphs SMPD (abbreviation for Single source Multiple sink Planar Dags). SMPD generalizes SSPD - Single source Single sink Planar Dags. SSPDs are interesting since they generalize *series parallel graphs* which is a well-studied restriction of directed acyclic graphs. Allender et al. presented a log-space algorithm for reachability in SMPD and left open whether reachability can be solved using logarithmic space over planar DAGs with multiple source nodes. In [SBV10], building on the SMPD algorithm, we present a log-space algorithm for planar dags with logarithmic number of sources. In the subsequent paper [SV12], via a careful use of techniques developed in [SBV10], we proved the log-space kernelization theorem that in particular implied a log-space algorithm for reachability in graphs with $2^{O(\sqrt{\log n})}$ sources, embedded on a surface of genus $2^{O(\sqrt{\log n})}$. The proof of this theorem is technically involved and we will not discuss it here. It remains a significant open question whether reachability for planar Dags (without any restriction on the number of sources) can be solved deterministically in $o(\log^2 n)$ space.

While improving Savitch's bound even for planar graphs remains open, the question of improving the BBRS bound for planar graphs was settled recently. Using a kernelization approach, in [INP$^+$13], we showed that the directed planar reachability problem can be solved in polynomial time using roughly $O(n^{1/2})$ space. This result extends a similar bound for the reachability problem over *grid graphs* due to Asano and Doerr [AD11].

Theorem 2 ([INP$^+$13])**.** *For any constant $0 < \epsilon < 1/2$, there is an algorithm that, given a directed planar graph G and two vertices s and t, decides whether there is a path from s to t. This algorithm runs in time $n^{O(1/\epsilon)}$ and uses $O(n^{1/2+\epsilon})$ space, where n is the number of vertices of G.*

For showing this result, we first design a polynomial-time and $\tilde{O}(\sqrt{n})$-space algorithm for computing a "separator" of $O(\sqrt{n})$ size for an undirected planar graph. (For any undirected graph G and for any constant ρ, $0 < \rho < 1$, a *ρ-separator* of G is a a subset of vertices S whose removal disconnects G into two subgraphs A and B, such that $|A|$ and $|B|$ is at most ρn). This algorithm is based on a parallel algorithm for constructing a planar separator due to Gazit and Miller [GM87].

Theorem 3 ([INP$^+$13])**.** *There is an algorithm that takes an undirected planar graph G with n vertices as input and outputs a (8/9)-separator of G of size $O(\sqrt{n})$. This algorithm runs in polynomial time and uses $\tilde{O}(\sqrt{n})$ space. (Here $\tilde{O}(s(n))$ denotes $O(s(n)(\log n)^{O(1)})$).*

Proof Sketch. While for obtaining the $O(n^{1/2+\epsilon})$ space bound we need a recursive approach, it is easy to illustrate the idea for the case when the space bound is $O(n^{2/3})$. Let $G = (V, E)$ be the input directed planar graph. Let G_u be the underlying undirected graph. The first step is to apply the planar separator algorithm repeatedly k times on the connected components of G_u that are bigger than $n^{2/3}$ to further partition the graph until every component is of size $\leq n^{2/3}$. It is easy to see that after $k = \lceil \frac{2}{3} \times \frac{\log n}{\log(9/8)} \rceil$ applications we get a collection $\mathcal{S}$ of separators with total size $O(n^{2/3})$ so that removing $\mathcal{S}$ partitions the graph into disconnected components where each component is of size $\leq n^{2/3}$. (This is a standard trick used in many separator-based algorithms). Let $C_1, C_2, \ldots, C_l$ be the set of vertices in these components.

Now consider the kernel graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{S} \cup \{s, t\}$. For any two nodes x and y in $\mathcal{V}$, (x, y) is a directed edge if and only if there is a directed path from x to y in the subgraph of G that is induced by $\mathcal{V} \cup C_i$ (call this G_i), for some connected component C_i in the partition. Clearly the number of nodes in $\mathcal{G}$ is $O(n^{2/3})$. Now consider the algorithm that decides whether there is a directed path from s to t in G by performing a BFS on $\mathcal{G}$ starting at s. Whenever BFS queries $(x, y) \in \mathcal{E}$?, the algorithm performs BFSs for each of the graphs G_i starting at x looking for a path from x to y, and returns YES if for some G_i this inner BFS returns true. Notice that since $|\mathcal{V} \cup C_i|$ is at most $O(n^{2/3})$, each of this BFSs can be implemented in $O(n^{2/3})$ space and polynomial time. Hence overall the algorithm takes $O(n^{2/3})$ space and polynomial time.

For extending this proof to the $O(n^{1/2+\epsilon})$ space bound, we need $|\mathcal{S}| = O(n^{1/2+\epsilon})$. But that will result in large components and a simple application of the inner BFSs will not give the required space bound. Instead, we can apply the algorithm recursively. By limiting the number of recursive applications to a constant, we can make sure that the running time remains polynomial. We omit the details. □

Before we move on to the next section we would like to mention that there is a certain computational model known as NNJAG model in which it is possible to prove lower bounds those match both Savitch's bound and the BBRS bound [Poo93, CR80, EPA99]. The NNJAG model is a branching program model tailored towards the reachability problem with restricted access to the input graph. While all the known algorithms for general reachability (such as BFS, DFS, Savitch's algorithm, BBRS algorithm) can be implemented in the NNJAG without substantial increase in time and space (in comparison to implementations on a random access machine), it is not clear to the author how a general approach such as kernelization can be handled in these models. It is conceivable that algorithms based on kernelization can overcome NNJAG lower bounds and help solve these open problems.

3. NL vs UL problem

The main progress on this problem also has been on graphs with some topological structure. We first discuss a technique developed by Reinhardt and Allender [RA00] since all the known proofs on this problem use their technique.

A positively and polynomially weighted graph is said to be *min-unique* if, between any two nodes the minimum weight path (if it exists) is unique.

Here the weight of a path is the sum of the weights of its edges. Reinhardt and Allender [RA00] showed, using an adaptation of the inductive counting technique of Immerman [Imm88] and Szelepcsényi [Sze88], that the reachability question in min-unique graphs can be decided in UL. They combine this construction with an observation due to Wigderson [Wig94] that the isolation lemma of Mulmuley, Vazirani and Vazirani [MVV87] can be used to non-uniformly assign weights to make a given graph min-unique. These two steps imply the collapse result that NL is in non-uniform UL.

Thus a space-efficient derandomization of the isolation lemma will show that $\mathsf{NL} = \mathsf{UL}$. However, derandomizing isolation lemma in its generality will have much deeper consequences and is a well known and difficult open problem [AM08]. Instead, a viable and concrete approach for showing $\mathsf{NL} = \mathsf{UL}$ is to first consider a class of graphs over which the reachability problem is NL-complete, and prescribe a *deterministic* log-space computable weight function which will make graphs in this class min-unique.

In [ABC$^+$09], the authors solve this min-unique weight assignment problem for the class of *layered grid graphs*. Layered grid graphs are graphs with vertices on a $n \times n$ grid and the edges that go west-to-east and south-to-north. Subsequently in [BTV09], we show how to extend this weight function to general grid graphs (without restriction on the direction of edges). This implied that directed planar reachability is in UL since the directed planar reachability problem is known to be reducible to the grid graph reachability problem [ADR05]. In fact this even implied that the reachability problem for graphs embedded on constant genus surfaces and graphs that are $K_{3,3}$-free and K_5-free are in UL since the reachability problem for these classes of graphs reduces to the directed planar reachability problem [KV10, TW09] in log-space.

While, in [BTV09] we showed that directed planar reachability is in UL it was not clear then how to solve the min-unique weight assignment problem for planar graphs. In a subsequent paper, we solve this problem using *Green's Theorem*, a celebrated result from multivariate calculus [TV12]. Since it is a slightly non-standard approach to use an analytical result to solve discrete problems, we believe this approach has the potential to solve the general NL vs UL problem. We next present the proof of the min-unique weight assignment problem for directed planar graphs based on Green's theorem.

Green's theorem, stated below, relates a certain curve integral over a closed curve on the plane to a related double integral over the region enclosed by this curve.

Theorem 4 (Green's Theorem). *Let C be a closed, piece-wise smooth, simple curve on the plane which is oriented counterclockwise. Let R_C be the region bounded by C. Let P and Q be functions of (x, y) defined on a region containing R_C that have continuous partial derivatives in the region. Then*

$$\oint_C (P\,dx + Q\,dy) = \iint_{R_C} \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA$$

We use the following corollary that we get if we substitute $Q(x, y) = x$ and $P(x, y) = 0$ in Green's theorem.

Corollary 5 (Area by line integrals). *Let C be a closed, piece-wise smooth, simple curve on the plane that is oriented counterclockwise. Let R_C be the region bounded by C. Then,*

$$\text{Area}(R_C) = \oint_C x\, dy$$

Theorem 6 **([TV12]).** *There is a log-space algorithm that, given any planar graph G, assigns weights to the edges so that the resulting weighted graph is min-unique.*

Let us assume that the planar graph $G = (V, E)$ is presented as a straight-line drawing. That is, each vertex v is represented as a point (x_v, y_v) in the plane so that an edge (u, v) is a line between points (x_u, y_u) and (x_v, y_v). In addition, no such lines intersect other than at the vertices. Moreover, we will assume that the coordinates are integer points with values bounded by poly(n) (n is the number of vertices). Typically, planar graphs are presented as a combinatorial embedding and it is not clear how such line drawings can be computed in log-space from a combinatorial embedding. However, this is not critical and in [TV12] we show how to handle this presentation issue.

Let $e = (u, v)$ be a directed edge directed from u to v where u is identified with the point (x_u, y_u) and v is identified with (x_v, y_v). For such a directed edge, define a weight function w as follows:

$$w_{gt}(e) = 2 \times \oint_e x\, dy = (y_v - y_u)(x_v + x_u)$$

The required isolation property of the weight function is proved using the following crucial lemma.

Lemma 7. *Let G be a directed planar graph and let C be any directed simple cycle in G. Let R_C be the region enclosed by C. Then the weight of the cycle C, $|w_{gt}(C)| = 2 \times \mathrm{Area(R_c)}$. In particular, $w_{gt}(C)$ is non-zero.*

Proof. Let $C = (e_1, e_2, \ldots, e_l)$ be a directed cycle oriented counterclockwise. Then we have

$$w_{gt}(C) = \sum_i w_{gt}(e_i) = 2 \times \sum_i \oint_{e_i} x\, dy = 2 \times \oint_C x\, dy = 2 \times \text{Area}(R_C)$$

The third equality follows from the linearity of integrals and the last equality follows from Corollary 7. If C is oriented clockwise, we get that $w_{gt}(C) = -2 \times \text{Area}(R_C)$. Hence the lemma.

□

The following lemma establishes Theorem 6.

Lemma 8. *Let G be a directed planar graph. Then with respect to the weight function w_{gt}, for every pair of nodes u and v, if there is a directed path from u to v, then there is a* unique *path from u to v of minimum weight.*

Proof. Suppose there are u, v so that there are two u to v paths P_1 and P_2 of minimum weight. We will assume that the paths do not intersect on vertices other than the end points (otherwise we can find two vertices u' and v' along these paths that satisfies this property using a standard cut-and-paste argument and use these vertices instead). We have $w_{gt}(P_1) = w_{gt}(P_2)$. Now consider the graph G' that is same as G except that the path P_2 is reversed so that the set of edges $(P_1, -P_2)$ becomes a simple cycle in G' ($-P_2$ denotes the reversed path). Let C denote this cycle. Then $w_{gt}(C) = w_{gt}(P_1) + w_{gt}(-P_2) = w_{gt}(P_1) - w_{gt}(P_2) = 0$. The second equality because of the skew-symmetry of the weight function. This contradicts Lemma 7. □

It is clear that we can use Green's Theorem to design a class of min-unique weight functions. In fact any "nice" solution to the differential equation $\left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right) = 1$ will yield such a weight function. For example, setting $P(x, y) = \frac{-y}{2}$ and $Q(x, y) = \frac{x}{2}$ to the left hand side of Green's theorem yields the weight function $w(e) = (x_u y_v - x_v y_u)$ which is also min-unique.

Can we use such geometric techniques to design min-unique weight functions for larger classes of graphs? In [BTV09] it is observed that reachability in layered grid graphs over 3 dimensions is complete for NL. It might be possible to use generalizations of Green's theorem (such as Stokes' theorem) to design a min-unique weight function for 3 dimensional layered grid graphs.

4. The BBRS bound

We will present the algorithm due to Barnes, Buss, Ruzzo, and Schieber [BBRS98] that solves the directed graph reachability problem in sub-linear space and polynomial time.

Theorem 9 ([BBRS98])**.** *For any k, there is a polynomial-time algorithm that given a directed graph G and two nodes s and t, decides whether there is a path from s to t in space $O(\frac{n}{2^{k\sqrt{\log n}}})$, where n is the number of vertices of G.*

Proof. The algorithm uses a combination of BFS and Savitch's algorithm. For a parameter λ (this will be set to $2^{k\sqrt{\log n}}$ to get the desired bound), it constructs the levels of BFS tree that are at λ distance apart. Divide the vertex set into levels according to distance from s. That is, the level i vertex set is defined as:

$$V_i = \{v \mid d(s, v) = i\}, \text{where } d \text{ is the distance function.}$$

Partition the set of vertices into λ equivalence classes $C_0, C_1, \ldots, C_{\lambda-1}$ where $C_j = \bigcup_{i=0}^{\lfloor n/\lambda \rfloor} V_{j+i\lambda}$. Since the C_is partition the vertex set, we have the following fact.

Fact 10. $\exists j^*$ so that $|C_{j^*}| \leq \lceil \frac{n}{\lambda} \rceil$

The Partial-BFS algorithm (described below) constructs C_{j^*} level by level. Since we do not explicitly know which C_j has $\leq \frac{n}{\lambda}$ nodes, the algorithm will keep a counter to count the number of vertices and try from $j = 0$. At any point of the construction, if $|C_j| > \frac{n}{\lambda}$, it will abandon that j and try the next value for j. The algorithm will succeed for the first such j. This will only increase the space by an additive $O(\log n)$ factor and the time by a multiplicative factor of λ. Hence we will assume that the algorithm knows j^*. Following is the description of the Partial-BFS algorithm.

```
Partial-BFS($G, s$) /* Outputs $C_{j^*}$*/
$V_0 = \{s\}$
$V_{j^*}$ = Construct($G, V_0, j^*$)
For $i = 1$ to $\lfloor \frac{n}{\lambda} \rfloor$
    $V_{i\lambda+j^*}$ = Construct($G, V_{(i-1)\lambda+j^*}, \lambda$)
    Add $V_{i\lambda+j^*}$ to $C_{j^*}$
End-For
Output $C_{j^*}$
```

In general, the procedure Constrct takes G and a set of nodes S and a parameter λ and returns the set of nodes that are at distance λ from some node in S. Construct will use the bounded version of the reachability problem (Barnes et al. calls it *short path problem*) as subroutine.

$$\text{SPATH}(u, v, \lambda) = \text{true} \Leftrightarrow \text{there is a path of length} \leq \lambda \text{ from } u \text{ to } v \text{ in } G.$$

We can use an algorithm for SPATH as subroutine to solve Construct as follows. Given (G, S, λ), to check whether $v \in V$ is at distance λ from some vertex in S, first check whether SPATH(u, v, λ) is true for some $u \in S$ and check for all $u \in S$, SPATH$(u, v, \lambda - 1)$ is false.

For a given algorithm for SPATH, let $T(n, \lambda)$ be its time complexity and $S(n, \lambda)$ be its space complexity. Then the time complexity of Construct is $O(n^3)T(n, \lambda)$ and its space complexity is $O(\frac{n}{\lambda}) + S(n, \lambda)$. Moreover, once C_{j^*} is constructed, reachability can be solved by making $\lceil \frac{n}{\lambda} \rceil$ calls to SPATH(u, t, λ) (for all $u \in C_{j^*}$). Thus the total running time for the reachability algorithm will be $O(n^4)T(n, \lambda)$ and the space bound will be $O(\frac{n}{\lambda}) + S(n, \lambda)$.

We will now focus on SPATH. We will use a divide and conquer approach as in Savitch's algorithm to design an algorithm for SPATH. The problem with a direct application of Savitch's algorithm is its running time: at each level of recursion it cycles through all n nodes as a candidate for the middle node. This results in $O(n^{\log n})$ time. Since we are interested in keeping the time polynomial, we can not afford to cycle through all n nodes. Instead, we will divide the set of nodes into μ equivalence classes and use a Savitch-like divide and conquer on these equivalence classes (instead of the vertices). For $\mu = 2^{O(\sqrt{\log n})}$ the depth of recursion will be $O(\sqrt{\log n})$ and this approach will result in polynomial time.

For a parameter μ, partition the vertex set into μ equivalence classes $[1], [2], \ldots [\mu]$ where vertex $x \in [a] \Leftrightarrow x \equiv a \pmod{\mu}$. Each equivalence class has $\lceil \frac{n}{\mu} \rceil$ elements (except for the last one whose cardinality may be smaller). We will use $[a], [b], [c]$ etc to denote these equivalence classes of vertices. Although this is not a very standard notation, the i^{th} vertex of the equivalence class $[a]$ (according to some fixed ordering) will be denoted by $[a](i)$.

Consider the procedure Modified-Savitch$(G, [a], [b], X, l)$ where $[a]$ and $[b]$ are equivalence classes of vertices, X is an $\lceil \frac{n}{\mu} \rceil$ binary array, and l is a length parameter. This procedure returns a binary vector Y of size $\lceil \frac{n}{\mu} \rceil$, where

$$Y[j] = 1 \quad \Leftrightarrow \quad \exists i \text{ so that } X[i] = 1 \text{ and there is a path of length } \leq 2^l \text{ from } [a](i) \text{ to } [b](j)$$

SPATH(u, v, λ) can be solved by one call to Modified-Savitch with parameter $([a], [b], X_u, \lceil \log_2 \lambda \rceil)$ where $[a]$ = the equivalence class containing u, $[b]$ = the equivalence class containing v, and X_u is the vector with 1 in the index corresponding to u and 0 otherwise. There is a path from u to v if and only if there is a 1 in the index corresponding to v in the output vector Y. Below is a recursive version of the algorithm Modified-Savitch.

```
Modified-Savitch($G, [a], [b], X, l$)
If $l = 0$ then
 If $[a] = [b]$ then $Y \leftarrow X$
 Else $Y[j] = 1$ iff $\exists i$ such that there is an edge from $[a](i)$ to $[b](j)$
Else
    $Y \leftarrow \overrightarrow{0}$
    For $c = 1$ to $\mu$
        $Z \leftarrow$ Modified-Savitch($G, [a], [c], X, l-1$)
        $Y_c \leftarrow$ Modified-Savitch($G, [c], [b], Z, l-1$)
        $Y \leftarrow Y \vee Y_c$
Return $Y$
```

Correctness of Modified-Savitch is easy to prove. Its time and space bounds can be estimated using the following recurrence.

$$\begin{aligned} S(l) &= O(\frac{n}{\mu}) + S(l-1) \\ &= O(\frac{n}{\mu}) \times l \\ T(l) &= \mu \times 2 \times T(l-1) + O(n) \\ &= (2\mu)^{l+1} \times O(n) \end{aligned}$$

Setting $\mu = 2^{(k+1)\sqrt{\log n}}$ and $l = \lceil \log_2 \lambda \rceil$, we get an algorithm for SPATH with time complexity $T(n, \lambda) = O(2^{\log \lambda} \times 2^{(k+1)\sqrt{\log n}(\log \lambda + 1)} \times n)$ and space complexity $S(n, \lambda) = O(\frac{n}{2^{(k+1)\sqrt{\log n}}} \times \log \lambda)$.

For $\lambda = 2^{k\sqrt{\log n}}$, this results in polynomial time and space $O(\frac{n}{2^{k\sqrt{\log n}}})$ giving an algorithm for the reachability problem with polynomial running time and $O(\frac{n}{2^{k\sqrt{\log n}}})$ space bound.

□

Acknowledgements

I would like to thank Pavan Aduri for extended collaboration and discussions on the reachability problem, and for his valuable comments on an earlier draft of this article. I would like to thank the organizers, V. Arvind and Manindra Agrawal, of the Complexity and Logic Workshop at IIT Kanpur (in celebration of the 60th birthday of Somenath Biswas) for inviting me to give a talk at the workshop.

References

[ABC$^+$09] E. Allender, D. A. Barrington, T. Chakraborty, S. Datta, and S. Roy. Planar and grid graph reachability problems. *Theory of Computing Systems*, 45(4):675–723, 2009.

[AD11] T. Asano and B. Doerr. Memory-constrained algorithms for shortest path problem. In *CCCG*, pages 135–138, 2011.

[ADR05] E. Allender, S. Datta, and S. Roy. The directed planar reachability problem. In *FSTTCS*, pages 238–249, 2005.

[ÀJ93] C. Àlvarez and B. Jenner. A very hard log-space counting class. *Theoret. Comput. Sci.*, 107:3–30, 1993.

[AM08] V. Arvind and P. Mukhopadhyay. Derandomizing the isolation lemma and lower bounds for circuit size. In *APPROX-RANDOM*, pages 276–289, 2008.

[ARZ99] E. Allender, K. Reinhardt, and S. Zhou. Isolation, matching, and counting uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59(2):164–181, 1999.

[BBRS98] G. Barnes, J. F. Buss, W. L. Ruzzo, and B. Schieber. A sublinear space, polynomial time algorithm for directed *s*-*t* connectivity. *SIAM J. Comput.*, 27(5):1273–1282, 1998.

[BJLR91] G. Buntrock, B. Jenner, K. Lange, and P. Rossmanith. Unambiguity and fewness for logarithmic space. In *8th International Symposium on Fundamentals of Computation Theory*, pages 168–179, 1991.

[BTV09] C. Bourke, R. Tewari, and N. V. Vinodchandran. Directed planar reachability is in unambiguous logarithmic space. *ACM Transactions on Computation Theory*, 1(1):1–17, 2009.

[CR80] S. A. Cook and C. Rackoff. Space lower bounds for maze threadability on restricted machines. *SIAM J. Comput.*, 9(3):636–652, 1980.

[EPA99] J. Edmonds, C. K. Poon, and D. Achlioptas. Tight lower bounds for st-connectivity on the NNJAG model. *SIAM J. Comput.*, 28(6), 1999.

[GM87] H. Gazit and G. L. Miller. A parallel algorithm for finding a separator in planar graphs. In *FOCS*, pages 238–248, 1987.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM J. Comput.*, 17(5):935–938, 1988.

[INP$^+$13] T. Imai, K Nakagawa, A. Pavan, N. V. Vinodchandran, and O. Watanabe. A $n^{1/2+\epsilon}$-space and polynomial-time algorithm for directed planar reachability. In *Conference on Computational Complexity*, pages 277–286, 2013.

[KV10] J. Kynčl and T. Vyskočil. Logspace reduction of directed reachability for bounded genus graphs to the planar case. *ACM Transactions on Computation Theory*, 1(3):1–11, 2010.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Cominatorica*, 7(1):105–113, 1987.

[Poo93] C. K. Poon. Space bounds for graph connectivity problems on node-named jags and node-ordered jags. In *FOCS*, pages 218–227, 1993.

[RA00] K. Reinhardt and E. Allender. Making nondeterminism unambiguous. *SIAM J. Comput.*, 29(4):1118–1131, 2000.

[Rei08] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008.

[RTV06] O. Reingold, L. Trevisan, and S. Vadhan. Pseudorandom walks on regular digraphs and the RL vs. L problem. In *STOC '06: Proceedings of the thirty-eighth annual ACM Symposium on Theory of Computing*, pages 457–466, New York, NY, USA, 2006. ACM.

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[SBV10] D. Stolee, C. Bourke, and N. V. Vinodchandran. A log-space algorithm for reachability in planar dags with few sources. In *Proceedings of 25th IEEE Conference on Computational Complexity*, pages 131–138, 2010.

[SV12] D. Stolee and N. V. Vinodchandran. Space-efficient algorithms for reachability in surface-embedded graphs. In *IEEE Conference on Computational Complexity*, pages 326–333, 2012.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

[TV12] R. Tewari and N. V. Vinodchandran. Green's theorem and isolation in planar graphs. *Inf. Comput.*, 215:1–7, 2012.

[TW09] T. Thierauf and F. Wagner. Reachability in $K_{3,3}$-free graphs and K_5-free graphs is in unambiguous log-space. In *FCT*, pages 323–334, 2009.

[Wig92] A. Wigderson. The complexity of graph connectivity. *Mathematical Foundations of Computer Science 1992*, pages 112–132, 1992.

[Wig94] A. Wigderson. NL/poly $\subseteq$ $\oplus$L/poly. In *Proceedings of the 9th Structures in Complexity conference*, pages 59–62, 1994.

N. V. Vinodchandran
Dept. of Computer Science and Engineering
University of Nebraska-Lincoln, USA
e-mail: `vinod@cse.unl.edu`