Generating Pseudo-Random Permutations and Maximum Flow Algorithms

Noga Alon
IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120 ,USA
and Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, ISRAEL

Abstract

We describe a simple construction of a family of permutations with a certain pseudo-random property. Such a family can be used to derandomize a recent randomized maximum-flow algorithm of Cheriyan and Hagerup for all relatively dense networks. Hence this supplies a deterministic maximum-flow algorithm that works, on a network with n vertices and m edges, in time $O(nm)$ for all $m = \Omega(n^{5/3} \log n)$ (and in time $O(nmlogn)$ for all other values of n and m). This improves the running time of the best known deterministic maximum-flow algorithm, due to Goldberg and Tarjan, whose running time is $O(nmlog(n^2/m))$.

Keywords: maximum-flow algorithms, design of algorithms, derandomization, pseudo-random permutations, longest common ascending subsequence.

1 The main results

Two permutations $\pi = \pi(1), \ldots, \pi(n)$ and $\sigma = \sigma(1), \ldots, \sigma(n)$ of $1, \ldots, n$ have a *common ascending subsequence of length* r if there are $i_1 < \ldots < i_r$ and $j_1 < \ldots < j_r$ such that $\pi(i_l) = \sigma(j_l)$ for all $l = 1, \ldots, r$. Let $\lambda(\pi, \sigma)$ denote the maximum length of a common ascending subsequence of π and σ. (Equivalently, $\lambda(\pi, \sigma)$ is the maximum length of an ascending subsequence of the sequence $\sigma^{-1}\pi(1), \ldots, \sigma^{-1}\pi(n)$).

Theorem 1 *For every two integers k and n, where $k \geq n^{0.2}$, one can construct a sequence $\pi_1, \ldots, \pi_k$ of k permutations of $1, \ldots, n$, such that for every permutation σ of $1, \ldots, n$ the inequality $\sum_{i=1}^{k} \lambda(\sigma, \pi_i) = O(kn^{0.8})$ holds.*
Such a sequence can be constructed (and written) in time $O(kn)$, i.e., in time which is essentially that needed to write these permutations down.

Theorem 2 *For every two integers k and n, where $k \geq n$, one can construct a sequence $\pi_1, \ldots, \pi_k$ of k permutations of $1, \ldots, n$, such that for every permutation σ of $1, \ldots, n$ the inequality*
$\sum_{i=1}^{k} \lambda(\sigma, \pi_i) = O(kn^{2/3})$
holds.
Such a sequence can be constructed (and written) in time $O(kn)$.

We note that the estimate above is not far from being best-possible. In fact for every k and n and for every sequence $\pi_1, \ldots, \pi_k$ of k permutations of $1, \ldots, n$, there is a permutation σ of $1, \ldots, n$ such that
$\sum_{i=1}^{k} \lambda(\sigma, \pi_i) = \Omega(kn^{1/2})$.
This follows from the simple fact that the expecetd length of the maximum ascending subsequence of a random permutation is $\Theta(n^{1/2})$, and hence the expected value of the left hand side of the last inequality, where the permutations π_i are fixed and σ is chosen randomly is $\Theta(kn^{1/2})$. We note also that if the permutations π_i are chosen randomly then one can check that with high probability for every permutation σ
$\sum_{i=1}^{k} \lambda(\sigma, \pi_i) = O(kn^{1/2})$.

Therfore, our explicitly-constructed permutations have a certain pseudo-random property.

As observed by Cheriyan and Hagerup, the permutations constructed above can be used to derandomize their randomized maximum-flow algorithm described in [3] for all relatively dense networks. Hence this supplies a deterministic maximum-flow algorithm that works, on a network with n vertices and m edges, in time $O(nm)$ for all $m \geq \Omega(n^{5/3} \log n)$ (and in time $O(nmlogn)$ for all other values of n and m). This improves the running time of the best known deterministic maximum-flow algorithm, due to Goldberg and Tarjan [5], whose running time is $O(nmlog(n^2/m))$.

It is worth noting that the problem of improving on the $O(nmlogn)$ time bound of the maximum-flow algorithm in [6] has motivated several recent interesting papers; see [4], [5], [1] and [2]. Yet, despite these efforts, before the derandomization given in the present note, for real-valued networks and also for networks with very large integer capacities the algorithm in [6] was still the fastest deterministic algorithm for $m = O(n^{2-\epsilon})$, where $\epsilon > 0$ is fixed.

2 The proofs.

In order to prove the above two theorems we need several simple lemmas.

Lemma 3 *Let $A_1, \ldots, A_s$ be s subsets of an n-element set X , and suppose that the cardinality of the intersection of each two distinct sets A_i does not exceed t. Then $\sum_{i=1}^{s} |A_i| \leq n + \frac{s(s-1)t}{2}$.*

Proof Clearly $n = |X| \geq \sum_{i=1}^{s} |A_i| - \sum_{1 \leq i < j \leq s} |A_i \cap A_j|$, implying the desired estimate. $\square$

Corollary 4 *Let $\pi_1, \ldots, \pi_s$ be s permutations of $1, \ldots, n$, and suppose that $\lambda(\pi_i, \pi_j) \leq t$ for all $1 \leq i < j \leq s$. Then , for every permutation σ of $1, \ldots, n$ $\sum_{i=1}^{s} \lambda(\pi_i, \sigma) \leq n + \frac{s(s-1)t}{2}$.*

Proof Put $X = \{1, \ldots, n\}$. For each i, $1 \leq i \leq s$, fix one maximum-length common ascending subsequence of π_i and σ, and let A_i be the subset of X consisting of the numbers in it. Clearly, $|A_i| = \lambda(\pi_i, \sigma)$, and the cardinality of the intersection of any two distinct sets A_i does not exceed t. The result now follows from Lemma 3. $\square$.

Lemma 5 *Let $n+1 = p$ be a prime and let $s \leq n$ be an integer. Then one can construct a sequence $\pi_1, \ldots, \pi_s$ of s permutations of $1, \ldots, n$, such that for all $1 \leq i < j \leq s$, $\lambda(\pi_i, \pi_j) \leq 2n^{1/2}s^{1/2}$.*

Such a sequence can be constructed (and written) in time $O(sn)$.

Proof The permutations we construct will all be of the form π_a with $1 \leq a \leq n$, where π_a is the permutation $a, 2a, \ldots, na$, in which all numbers are reduced modulo p. The set A of numbers a for which we will take the permuatation π_a will have the following property:

$$\forall a, b \in A, a \neq b \text{ there are no } c, d \text{ with } 1 \leq c, d \leq n^{1/2}/s^{1/2} \text{ such that } ac = bd (\text{modulo } p). \quad (1)$$

Such a set A of cardinality s can be easily constructed greedily. After we have already chosen $k < s$ members we compute all the $kn/s < n$ numbers bd/c (modulo p) where b is such a member and $1 \leq c, d \leq n^{1/2}/s^{1/2}$,and choose a to be different from all those.

Now observe that if j and l are two distinct numbers in $\{1, \ldots, n\}$, then if j appears after l in π_a then the distance between them in π_a is $(j-l)/a$. Similarly, the distance between them in π_b is $(j-l)/b$. (All these operations are modulo p, of course). It is impossible that both these numbers are smaller than $n^{1/2}/s^{1/2}$ for two distinct a, b in A, since in this case $j - l = ac = bd$ where $1 \leq c, d \leq n^{1/2}/s^{1/2}$, contradicting (1). Thus, in any common ascending sequence of π_a and π_b one of the distances between any two corresponding pairs of adjacent elements in the subsequence is at least $n^{1/2}/s^{1/2}$ and hence the size of this sequence cannot exceed $\frac{2n}{n^{1/2}/s^{1/2}} = 2n^{1/2}s^{1/2}$. $\square$.

Proof of Theorem 1 If $n+1$ is a prime then, by Lemma 5 (with $s = \lfloor n^{0.2} \rfloor$) and Corollary 4 there are $k = \lfloor n^{0.2} \rfloor$ permutations for which the assertion of the theorem holds. If k is bigger, we repeat this set of permutations as many times as needed. Finally, if $n+1$ is not a prime we choose a prime larger than $n+1$ and smaller than $2n+2$ (such a prime always exists by Bertrand's postulate and can be found quickly), construct our permutations for that prime and then take their restrictions to $1, \ldots, n$. This completes the proof. $\square$

Proof of Theorem 2 Suppose, first, that $n+1 = p$ is a prime and that $k = n$. In this case we simply take all the permutations π_a for $a \in \{1, \ldots, n\}$. Let σ be an arbitrary permutation of $1, \ldots, n$. Define x by $x = \sum_{i=1}^{n} \lambda(\sigma, \pi_i)$. We must show that $x = O(n^{5/3})$. For each i, $1 \leq i \leq n$ let us fix a common ascending subsequence of π_i and σ of maximum length $\lambda(\pi_i, \sigma)$. Denote this sequence by S_i. For each pair of adjacent elements in S_i define their *distance* to be the distance between them in π_i plus the distance between them in σ. Obviously, the sum of all the distances between all the adjacent pairs of all the sequences S_i (including the cyclic distance between the last element of each S_i and the first element of it) is precisely $2n^2$. Therefore, there are at least $x/2$ adjacent pairs whose distances are all at most $\frac{4n^2}{x}$. Note that we may assume that $\frac{4n^2}{x} \leq n$, since otherwise $x < 4n$ and there is nothing to prove. The number of pairs in the permutation σ whose distance in σ is at most $\frac{4n^2}{x}$ is exactly $n\frac{4n^2}{x} = \frac{4n^3}{x}$. Each such pair appears with all possible distances between its members in the various π_i, and hence there are exactly $4n^2/x$ permutations in which it appears with distance at most $4n^2/x$. Therefore, the number of pairs of adjacent elements of the n subsequences S_i whose distances, as defined above, are at most $4n^2/x$ is certainly at most $\frac{4n^3}{x}\frac{4n^2}{x} = \frac{16n^5}{x^2}$. But this number is at least $x/2$ and hence $x/2 \leq \frac{16n^5}{x^2}$, implying $x \leq 32^{1/3}n^{5/3}$. This completes the proof when $k = n$ and $n+1$ is a prime. The general case follows as in the proof of Theorem 1. $\square$

3 Discussion

In order to derandomize the maximum-flow algorithm of [3] for sparser networks, a more complicated construction is needed. We say that a permutation $\sigma = \sigma(1), \ldots, \sigma(n)$ of $1, \ldots, n$ and a permutation $\pi = \pi(1), \ldots, \pi(q)$ of a subset of cardinality q of $\{1, \ldots, n\}$ have a *common ascending subsequence of length* r if there are $i_1 < \ldots < i_r$ and $j_1 < \ldots < j_r$ such that $\pi(i_l) = \sigma(j_l)$ for all $l = 1, \ldots, r$. Let $\lambda(\sigma, \pi)$ denote the maximum length of a common ascending subsequence of σ and π. (Equivalently, $\lambda(\sigma, \pi)$ is the maximum length of an ascending subsequence of the sequence $\sigma^{-1}\pi(1), \ldots, \sigma^{-1}\pi(q)$). Given a family $F = \{A_1, \ldots, A_n\}$ of n subsets of $\{1, \ldots, n\}$, such that $\sum_{i=1}^{n} |A_i| = m$, we wish to find a family $\{\pi_1, \ldots, \pi_n\}$, where π_i is a permutation of the elements of A_i, such that for every permutation σ of $\{1, \ldots, n\}$, the sum $\sum_{i=1}^{n} \lambda(\sigma, \pi_i)$ does not exceed $O(m/\log n)$. In [3] it is shown, by a simple probabilistic argument, that if $m \geq n(\log n)^2$ such a set of permutations π_i always exists. Moreover, it follows from the analysis in [3] that if, for some n and $m \geq n(\log n)^3$, we can generate such a set of permutations in time $O(nm)$ for any given family of subsets F whose sum of cardinalities is m, then we can obtain a deterministic maximum-flow algorithm that works in time $O(nm)$ for every network with n vertices and m edges. Theorem 2 (with $k = n$) clearly suffices to give the desired permutations in case $m \geq \Omega(n^{5/3} \log n)$. (We simply let π_i be the restriction of the i-th permutation supplied by Theorem 2 to A_i.) This theorem, as well as the somewhat different Theorem 1 do not suffice for smaller values of m. In fact, it is unlikely that a similar method would work for $m = o(n^{3/2})$, since there exist families of n subsets A_i of an n element set, each having cardinality $\Omega(n^{1/2})$, such that no two of these subsets have an intersection of size 2 or more. Since our method depends on the existence of common pairs of elements in the various sets A_i it seems that a new idea is needed for such cases. It is not impossible that some of the known pseudo-random properties of explicitly constructed expander-graphs can be useful here. At the moment we do not see how to use these properties, and the problem of constructing permutations with the desired properties for the cases of small m, as well as the derandomization of the maximum-flow algorithm of [3] for sparser networks, remains open.

References

[1] R. K. Ahuja and J. B. Orlin, *A Fast and Simple Algorithm for the Maximum Flow Problem*, Operations Research, to appear.

[2] A. K. Ahuja, J. B. Orlin and R. E. Tarjan, *Improved Time Bounds for the Maximum Flow Problem*, SIAM J. Comput. 18 (1989), 939-954.

[3] J. Cheriyan and T. Hagerup, *A Randomized Maximum-Flow Algoithm*, Proc. IEEE FOCS (1989), 118-123.

[4] H. N. Gabow, *Scaling Algorithms for Network Problems*, J. Comput. Syst. Sci. 31 (1985), 148-168.

[5] A. V. Goldberg and R. E. Tarjan, *A New Approach to the Maximum-Flow Problem*, J. of the ACM 35 (1988), 921-940.

[6] D. D. Sleator and R. E. Tarjan, *A Data Structure for Dynamic Trees*, J. Comput. Syst. Sci. 26(1983), 362-391.