The Complexity of Parallel Computation on Matroids

Richard Karp
Eli Upfal
Avi Wigderson

Abstract

In [KUW1] we have proposed the setting of independence systems to study the relation between the computational complexity of search and decision problems. The universal problem that captures this relation, which we termed the S-search problem, is "Given an oracle for the input system, find a maximal independent subset in it."

Many interesting and important search problems can be described by an special class of independence systems, called *matroids.* This paper is devoted to the complexity of the S-search problem for matroids.

Our main result is a lower bound on any probabilistic algorithm for the S-search problem that acquires information about the input system by interrogating an independence oracle. We prove that the expected time of any such probabilistic algorithm that uses a sub-exponential number of processors is $\Omega(n^{1/3-\varepsilon})$. This is one of the first nontrivial, super-logarithmic lower bounds on a randomized parallel computation. It implies that in our model of computation Random-NC is strictly contained in P. Another consequence of the lower bound is that the $O(\sqrt{n})$ time probabilistic upper bound for arbitrary independence systems, presented in [KUW1], is close to optimal and cannot be significantly improved even for matroids.

However, this $O(\sqrt n})$ upper bound can be improved in a different sense for matroids – it can be made deterministic, still with polynomially many processors.

Finally, we show that the lower bound can be better for the special case of graphic matroids. Here, the S-search problem is simply to find a spanning forest of a graph, when the algorithm cannot see the graph, but can only ask whether subsets of edges are forests or not. We give an $O(\log n)$ time deterministic parallel algorithm that uses $n^{O(\log n)}$ processors.

From the upper bounds on parallel time above we decide similar bounds (up to a polylog factor) on the sequential space required by a deterministic Turing machine with an independence oracle to solve the S-search problem.