

Efficient Algorithms and Data Structures

Mikkel Thorup

May 19, 2022

Abstract

The proposed project will address some of the fundamental issues in efficient algorithms and data structures, ranging from pseudo-random hashing, to the existence of deterministic dictionaries with constant update and look-up time, to graph algorithms. The applicant is a leading figure in the area of efficient algorithms and data structures. For the last 14 years he has been in the USA where the area has its stronghold. The grant will facilitate his return to the University of Copenhagen and will enable him to create a mini-center of excellence: a new focal point that will boost the research in efficient algorithms and data structures in Europe.

Project description

Efficient Algorithms and Data Structures (with references to some related work of mine) Algorithms is one of the main areas of computer science, both in terms of teaching and research. It is relevant to the processing of data whenever we need scalable solutions, hence for a lot of science and industry. Suppose, for example, that we want to solve a problem involving n variables. A trivial exhaustive search algorithm might try all combinations. However, even if each variable has only two options, there are $2n$ combinations, which is exponential, and if $n = 70$, then this is too much even if all the world's computers worked on it for a year. In algorithms we try to find solutions in time polynomial if not linear in the number of variables. Since any general enough problem is NP-hard or worse, we will have to understand and exploit the special nature of the problem at hand. Take a classic problem like that of sorting n numbers. Sorting is commonly needed in the processing of data. A naive algorithm like Insertion Sort takes one number at the time, scanning for its position among those already sorted, in $O(n^2)$ total time. The ' O ' represents an unspecified constant depending on the concrete computer used. Our focus in algorithms is asymptotic performance with $n \rightarrow \infty$. The more sophisticated Quick Sort solves the problem in $O(n \log n)$ expected time, allowing us to handle much larger data sets. I myself have the record with an algorithm sorting n integer or floating point numbers in $O(n \log \log n)$ expected time [25]. It is still a major open problem if integers can be sorted in linear time. Generally speaking, in algorithms we try to understand how well we can solve combinatorial problems making the most efficient use of computational resources such as time and space. The discipline is theoretical in that we try to prove that our solutions are efficient for all possible inputs, or alternatively, in expectation for a given input distribution. Many algorithms researchers have their main focus on deciding which kind of problems can be solved in polynomial time (see e.g., [49]). In particular, for NP-hard optimization problems, a lot of effort is put into understanding how good an approximation we can guarantee in polynomial time (my papers [4, 11, 27] follow this line of work). However, as in the above sorting example, I am very interested in more efficient algorithms where polynomial time is not enough, but where we try to get the running time closer to linear. The underlying motivation is to understand how to deal with large data sets. Very concretely, it is not uncommon to deal with data sets with, say, $n = 10^8$ elements. On today's computer, simple instructions like a memory lookup take about 10⁷ seconds. A quadratic algorithm using n^2 instructions would take more than 30 years. The issue is not going to be resolved by more powerful computers, it is only going to get worse: the size of the problems considered tends to infinity.

Efficient Algorithms and Data Structures For my kind of research, novelty and feasibility are not to be found in the initial plans and approaches. This is basic theoretical research looking for novel, surprising, simple, and powerful techniques. The feasibility of this objective is to be found in the

general approach and skills, as documented by my track record. The issue is nicely illustrated by the following quote from the expert referee report on my paper “The minimum k-way cut of bounded size is fixed-parameter tractable” with Kawarabayashi [29]: The techniques used in the algorithm are very ingenious and were never before used for parameterized cut problems. The paper is completely elementary and self-contained (in particular, no tools from graph minor theory is used). In fact, the paper essentially contains no proofs: once they explain the algorithm, it is all obvious! This does not mean that the algorithm is trivial: the ideas are quite unnatural and it is not obvious in the beginning why this approach should work at all, but apparently they do. Having such a “simple” proof should be considered as big plus. The point here is that the novel successful approach was unnatural to the expert (I was not an expert but had the outsiders fresh inspiration), so even if I had put it in a project plan, then it would not have been feasible-looking. In fact, since the technique is simple, if it had been natural, then it would have been found long ago. Moreover, we tried lots of novel approaches before we got this one to work. More generally, the basic requirements for STOC/FOCS level research is that the problem should be important and the technique surprising. Thus we are looking for surprisingly powerful ideas, but those you cannot plan for. Many ideas are developed, tried, and failed as work with the problem. When first you have the right idea, it often does not take so long to check if it works. Before you have proved that an idea works, you do not know if you are on the right track. Finally, with ambitious targets, there is a very good chance that the problem will turn out too hard, if not be impossible. Staying focused on a single ambitious target is thus very risky.

A feasible flexible general approach

My research strategy is not unique but shared by many successful researchers in theory. Striving for excellence, I often aim at ambitious targets with a significant risk of failure. To make it statistically feasible, I work on multiple ambitious targets. This is possible because there are no real investments associated with this research on a particular target (no expensive experimental setup): if you are stuck on one target, and another is more promising, then there is no real cost involved in switching. The targets should be unrelated so that the failure of 5

0.1 write an equation

$$S_n = \frac{P_1 + P_2 + \cdots + P_n}{n} = \frac{1}{n} \sum_i^n P_i$$

0.2 add a table

student	roll	
Abu rayhan	MUH2011001	An example table.
Albid hosain	MUH2011048	
sourav day	Muh2011036	

0.3 add a figure



friends pic was uploaded via the file-tree menu.

References