Claire Kenyon's statement of purpose

Research statement

The $P \neq NP$ conjecture may currently be the central open problem of computer science, operations research and mathematics. My work concerns the design and analysis of algorithms, and is largely motivated by the wish to find a way to deal with NP-complete problems. After a PhD on the average-case analysis of dynamic data structures, and some work on computational geometry, I got interested in tiling problems. Then my interest shifted to approximation algorithms, on which I have been working for about five years, and which provide a way to circumvent the NP-hardness of optimization problems by resigning oneself to finding an approximate solution. I am also interested in the probabilistic analysis of algorithms, particularly using Markov chains : resigning oneself to finding an algorithm which is efficient on average, instead of in the worst case, is another way to circumvent NP-hardness.

Tiling algorithms

Given a set of *prototiles*, where a prototile is a connected union of unit squares, and given a rectilinear polygon whose edges all have integer lengths, the problem is to decide whether the polygon can be tiled with translated copies of the prototiles. in 1990, J. Conway and J. Lagarias [13] defined a *tiling group* associated to the prototiles. If the polygon issimply connected and tileable, then the word on the alphabet $\{h, v, h^{-1}, v^{-1}\}$ which is obtained by going around the polygon boundary must be equal to the identity in the tiling group. Unfortunately this condition is neither easy to check (the word problem is undecidable in general) nor sufficient in general. W. Thurston [45] showed how to exploit the geometric structure of the Cayley graph of the tiling group to obtain, in some cases, a necessary and sufficient condition for tiling ; the main example he used is the domino tiling group, when there are two prototiles, the 1×2 and the 2×1 rectangles. In that case, his approach leads to a linear time algorithm, better than all previously known algorithms (which all used the relation between domino tiling and perfect matching in the dual graph).

In 1992, in collaboration with R. Kenyon, I showed how to generalize this approach to the case where there are two rectangular prototiles [32]. In particular, if the prototiles are the 1×3 and the 3×1 bars, then we obtain a linear time algorithm for tiling simply connected polygons, whereas the problem is NP-complete for general polygons [43]. This demonstrated the potential of this approach for algorithmic design. Since then, various papers have suggested tiling algorithms for several different prototiles (see [42] for references).

Such algebraic techniques are best suited to exact tiling algorithms. When a polygon is not tileable, it is natural to aim for a tiling included in the polygon and of maximum surface area, an optimization problem. It turns out that this problem can in general be solved almost optimally, using a variant of the algorithm of Hochbaum and Maass for tiling with squares [26]. The idea is to partition the plane into squares of large enough constant side length ; to translate this partition to a random position ; and to tile the polygon independently inside each of the squares. This is a clear example of an area where exact algorithms are highly sophisticated, must be worked out separately for each case, and only work for particular tiling problems, while the approximation algorithm, based on a few simple ideas, is a general-purpose algorithm suitable for any tiling problem and finds a tiling with relative error ϵ, which can be arbitrarily small. This motivated my recent interest for the design of approximation schemes.

Approximation schemes

Given an optimization problem, an approximation scheme, for any $\epsilon > 0$, gives an algorithm which finds, in time polynomial in the size of the input, a solution whose value V satisfies

$$1 - \epsilon < V/OPT < 1 + \epsilon,$$

where OPT is the value of the optimal solution. One talks of an asymptotic approximation scheme where this inequality only holds for OPT large enough.

Not all optimization problems have approximation schemes, and the PCP theorem is a recent spectacular development in the design of lower bounds [15, 4, 3] for problems such as the maximum clique [25]. In the past ten years, the gist of the effort in algorithms and complexity has been oriented towards a classification of optimization problems depending on how well they can be approximated in polynomial time. My work deals with upper bounds only, i.e. with finding efficient approximation algorithms.

One major paradigm for designing an approximation algorithm consists of formulating the problem as an integer program, relaxing it into a linear program, solving it and using some deterministic or probabilistic rounding to obtain an approximate solution to the original problem. See, for example, Lovász's algorithm for set cover [39].

I first focused on packing problems, and in particular bin-packing type problems. It is well-known that this NP-complete problem [30] has an approximation scheme [21, 31]. Its extension to two dimensions, where rectangular objects must be placed in a strip of unit width an minimum height, only had constant factor approximations. In collaboration with E. Rémila, I designed an approximation scheme for that problem [36], based on a new linear programming formulation and a technical combination of previous approaches. This thorough understanding of the techniques then led to an approximation scheme for "bin-covering", also called dual bin-packing [10], using a well-chosen linear program, and recently to other two dimensional packing problems [8].

Strip-packing is somewhat related to some more applied problems which might be worth studying in detail. In particular, the optical fiber technology uses SONET rings as alternatives to classical queues to deal with local congestion. Placing messages to optimnize the space used, and so to maximize the number of messages which can be put on hold on a given ring, can be modelled by a rectangle packing problem to which I recently provided a partial solution [5], although the general problem remains open.

Packing problems are similar in flavor to scheduling problems, when the tasks are independent, which then led me to study classical scheduling problems.

Perhaps the most basic scheduling problem consists in scheduling n tasks of given processing times onto m identical processors so as to minimize the makespan. This NP-complete problem has an approximation scheme [26]. I have studied the generalization where the tasks are multiprocessor tasks and the processors may be dedicated, and in [2], I proposed an approximation scheme when the number of processors is bounded, based on a new linear programming formulation, where the variables correspond to "snapshots" of the schedule. Although the problem itself is not greatly important, the approach was used in several subsequent papers on scheduling.

Recently, approximation schemes were found for many NP-hard scheduling problems, thanks to the development of an array of tools for designing approximation algorithms. I participated in this construction with [1], a paper merging work from three groups of researchers who found the same result simultaneously and independently. In this problem there is only one processor, but one wants to minimize the average task completion date, when the task are not all available at once but have release dates. Among other results, the paper presents an approximation scheme for this problem.

I am currently working, in collaboration with Howard Karloff and Moses Charikar, on designing an approximation scheme for a basic scheduling problem with general precedence constraints; precedence constraints make a challenging category of algorithmic problems, and insight one of those would be a big step forward in our understanding.

Rounding is a natural concept for problems involving numbers, but is much more difficult to understand for graph problems. In fact, there are few approximation schemes known for general graph problems. On the other hand, when the graph is dense, i.e. when the number of edges is $\geq cn^2$, then randomized algorithms based on random sampling have led to approximation schemes for cuts, bisections, and clustering problems [23, 17]. Can this approach be used for other classes of graphs ?

In collaboration with Fernandez de la Vega, I worked on metric graphs, i.e. complete weighted graphs whose edge weights satisfy the triangular inequality, that is, discrete metric spaces. On these spaces, we were able to design an approximation scheme for the classic MAX-CUT problem (partition the vertices into two subsets so as to maximize the weight of edges accross the cut) [20]. After several simplifications, the final algorithm appreads straightforward. Thus MAX-CUT, which is MAX-SNP-hard for general graphs (hence cannot have an approximation scheme if $P \neq NP$), with a constant factor approximation in that setting [22], is also NP-hard in the metric case, but has an approximation scheme in that setting. Finally, its status in 2-dimensional Euclidian space is currently unknown; it might well be polynomial. Recently, we have successfully combined our approach with to the minimum bisection problem on metric graphs [18]. The problem is to partition the vertices into two equal-size subsets, ao as to minimize the weight of edges going accross the cut. In the general graph case, a lot of understanding has recently been gained on this problem, with an approximation algorithm with a polylogarithmic factor due to Feige and Krauthgamer [16].

A possible area of application of these non-uniform random sampling techniques might be for the design of new algorithms for clustering. We have recently developped an approximation scheme for clustering into a fixed number of clusters [19]. It would be nice to develop this research direction into a better understanding of popular algorithms, for example for the "online clustering problem".

It is frustrating that the status of MAX-CUT is still unknown in the Euclidian case: it is not even clear that the problem is NP-complete. The difficult instances for this problem seem to be the ones where the points are uniformly spread over some finite region, and so one might gain better understanding of the problem by doing a probabilistic analysis and designing an algorithm which finds the exact optimum (with some probability) when the points form a Poisson process in the unit square.

Many of the problems mentioned so far belong to the realm of basic research, but it is also possible to look for approximation algorithms for problems defined in relation to recent applications. Thus, with Neal Young and my ex Phd student Nicolas Schabanel, I worked on the data broadcasting problem: how to repeatedly broadcast messages on a channel (s.a. a radio station) so as to minimize the expected waiting time of the customers, given that each message (news, weather, traffic,...) has a given popularity. With the help of a lower bound obtained by lagrangian relaxation, we designed an approximation scheme for the case when all messages have the same broadcast length, and a constant factor approximation algorithm for the case when the broadcast lengths may be different [37, 38]. I ahve also worked on approximation algorithms in various other contexts [24, 29].

Here is another applied problem where approximation techniques might come in handy. The question is to distribute jobs to heterogeous processors using a master-slave protocol. Here, n identical tasks must be allocated to m processors of speeds $v_1, \ldots, v_m$, who all share a common communication bus; a processor can only start working after having received the tasks allocated to it, and upon completion of its work, must send the results back over the communication bus. The problem is that the bus can only be used by one processor at a time, so that none of the classical scheduling algorithms applies. On the other hand, roudning techniques and others might provide means to design an approximation algorithm. I am planning to work on this problem in collaboration with N. Schabanel and Y. Robert.

Probabilistic analysis

Since NP-completeness is exclusively focused on worst-case analysis, and this may possibly not be representative behaviour, one may attempt to explain the nice behavior of some algorithms in practice by doing a probabilistic analysis on the input distribution.

This approach was pursued very far in the analysis of bin-packing algorithms [6]. Ten years ago, E. Coffman and his coauthors presented results of extensive simulations of the Best-Fit algorithm for various probability distributions, and showed experimentally that the behavior seemed to depend on details of the distribution in a highly non-trivial manner [7, 9]. Based on these simulations, theyr formulated several conjectures. In [34, 33], we prove two of these conjectures. The most significant result is the following [33]: if the number are i.i.d. random variables drawn from the uniform distribution in $[0, a]$, with a close to $2/3$, then Best-Fit is far from optimal, i.e. the ratio of the number of bins used by Best-Fit, to the optimal number of bins, is strictly greater than 1. Until now, such a result had not been shown for value of a. It is worth noting that this result, which is in accordance with numerical simulations, shows that the behavior is very different from when $a = 1$ [44]. Both papers [34] and [33] are based on a detailed study of the specific propertie of the infinite Markoc Chain associated to the algorithm under this distribution, and of the balance between favorable and unfavorable configurations. The second paper is a reduction which groups configurations into a finite but large number of parts (several hundreds), and the last piece of the proof is a dynamic program solved by computer.

The suboptimal performance of Best-Fit for some distributions raise the question of the existence of a *simple* on-line algorithm which is efficient for several distributions. In [11], we propose such an algorithm, which we call *Sum-of-Squares* (it always chooses the bin in which to place the next item so as to minimize the sum of the squares of the number of bins of each residual capacity), and show experimentally that it behaves extremely well. In [12], we analyze its behavior. Our main and most easily explained result is that, if the numbers are i.i.d.r.v.'s drawn from the same discrete distribution, and the optimal packing packs almost all bins to full capacity under this distribution, then Sum-of-Squares will also pack almost all bins to full capacity. In other words, it is optimal up to lower order terms.

Rapidly mixing Monte Carlo Markov chain form a research area which is at the intersection of approximation algorithms and probabilistic analysis of algorithms. This is a typical interdisciplinary area: the problems (random generation, counting) typically come from statistical physics; the Monte Carlo Markov chains usually need to be defined by computer scientists in order to be analyzable; and the analysis of the time to convergence uses advanced combinatorics or probability.

In computer scinec, the founding paper of Jerrum and Sinclair [27] gave an algorithm and analysis technique to calculate the permanent (number of perfect matchings, number of dimer configurations) of a dense bipartite graph and several other cases of a bipartite graph. This result was recently extended to general graphs [28]. In the 12 years separaying the two papers, the area grew and deepened tremendously, thanks to interactions between researchers from the different disciplines involved.

My first result in this area dates back to 1993, when, in collaboration with Dana Randall and Alistair Sinclair, I showed how to estimate the number of dimer configurations on the toroidal grid $\{0, 1, \ldots, n-1\}^k$, as well as the number of monomer-dimer configurations on the square grid [35]. The heart of the paper is a result in bijective combinatorics which implies that the algorithm from [27] has polynomial mixing time for those graphs.

One of the important questions in this area currently is to better understand the relationship between the mixing time of the Markov chain and the existence of a phase transition. I got interested in phase transitions in [14], where we study a process to propagate a random bit on a noisy tree (this process models the Gibbs distribution for the Ising model on a tree). A random bit (or spin) is placed at the root of the tree and transmitted along the edges, with error probability ϵ along each edge (this correspond to the interaction between the spins). How much information do the bits at the nth level of the tree possess about the root ? Our result caracterizes the critical probability ϵ_c beyond which information is lost (i.e. the mutual information tends to 0 as n tends to infinity).

Recently, we have worked on the link between mixing time of the Markov chain and existence of a phase transition, in the simple context of the Ising model on trees. We studied the Glauber dynamics (Gibbs sampler) for the Ising model on a balanced binary tree of height n, as temperature varies [40]. Our analysis of relaxation time is complete in this context. In spite of the existence of a phase transition, the mixing time is polynomial at every temperature. Moreover, there are temperatures at which the relaxation time is linear even though the infinite tree has several Gibbs measures. The analysis uses a variety of techniques: canonical paths, spectral gap and Dirichlet forms, matching. Each technique has its own range of application, and some results extend to slightly more general graphs. One may hope that this paper, by providing a detailed analysis of a simple model, will help to clarify a diverse area.

One reason for the current dynamism of research in algorithms is the emergence of new problems related to the development of the Internet. One particularly interesting questions consists of modeling the behavior of the Internet, particularly the world wide web. Classical random graph models are not well-suited to this (in the classical theory of random graphs, most parameters have a normal distribution, whereas parameters related to the Internet seem to have a heavy-tailed distribution: i.e., rare events do sometimes happen !). A few new models are emerging. I would be interested in exploring the algebraic properties of those models (link with the spectral properties of the adjacency matrix). Reliability and fault-tolerance, connectivity, degree distribution, are the parameters currently under study.

Another popular example of these new problems: we are just beginning to understanf fomr a mathematical viewpoint the "small world phenomenon", according to which any one can communicate with anyone else using a small number of intermediaries, each communication going through two persons who know one another personnally. This also has applications to the web. I have started thinking about an general version of the celebrated small world model from [41].

The most beautiful area in the probabilistic analysis of algorithms, and one of the most difficult, concerns the analysis of Monte-Carlo Markov chains. There is a long and well-defined list of major open problems.

Among these, the random generation of contingency tables looks particularly appealing to me. This problem finds its original motivation in statistics and the tabulation of results from surveys. The issue is to reconstruct a non-negative n by m matrix, given the n row sums and the m column sums. This problem can be solved by dynamic programming if m is constant and the entries are small integers, and can be solved approximately by a volumn estimation when the row and column sums are all large. One originality of the problem is that it seems to combine common quasi-uniform generation techniques (Monte Carlo Markov chains) with techniques from optimization (rounding, dynamic programming). It is conceivable that a suitable combination of those techniques may solve the problem.

References

[1] F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein and M. Sviridenko, Approximation Schemes for Scheduling to Minimize Average Weighted Completion Time with Release Dates, Proceedings of the 1999 *Symposium on Foundations of Computer Science (FOCS)* (plenary session), New York City, NY, October 17-19, 1999, pp. 32-44. Cet article a été sélectionné par le comité de programme de FOCS pour être parmi les 16 articles présentés en session pleinière (sur les 68 articles acceptés cette année-là).

[2] A.K. Amoura, E.Bampis, C.Kenyon and Y. Manoussakis, Scheduling Multiprocessor Tasks on Dedicated Processors, European Symposium on Algorithms (ESA), 1-10, 1997. To appear in Algorithmica.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In Proc. 33rd IEEE Annual Symposium on Foundations of Computer Science, pp 13-22, 1992.

[4] S. Arora and S. Safra. Probabilistic checking of proofs : a new characterization of NP. In Proc. 33rd IEEE Annual Symposium on Foundations of Computer Science, pp 2-13, 1992.

[5] OPT versus LOAD in Dynamic Storage Allocation, Adam L. Buchsbaum, Howard Karloff, Claire Kenyon, Nick Reingold, and Mikkel Thorup. Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC) 2003, 556–564. SIAM J. on Computing, to appear.

[6] Coffman, E. G., Jr., and Lueker, G. S., Probabilistic Analysis of Packing and Partitioning Algorithms, Wiley & Sons, 1991.

[7] Coffman, E. G. Jr., Courcoubetis, C. A., Garey, M. R., Johnson, D. S., McGeoch, L. A., Shor, P. W., Weber, R.R., and Yannakakis, M., Average-Case Performance of One-Dimensional Bin Packing Algorithms under Discrete Uniform Distributions, Proc. ACM Symp. Th. Comput. (STOC), ACM Press, 1991, 230-240.

[8] Asymptotic Approximation Schemes for Two-Dimensional Packing, J. Correa and Claire Kenyon, Proceedings of the *Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004, to appear.

[9] Coffman, E.G., Jr., Johnson, D.S., Shor, P.W., and Weber, R.R., Markov Chains, Computer Proofs, and Best-Fit Bin Packing, Proc. ACM Symp. Th. Comput. (STOC), 1993, 412-421.

[10] Janos Csirik, David Johnson and Claire Kenyon, Better Approximation Algorithms for Bin Covering, Proceedings of the *Twelth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2001, pp. 557-566.

[11] Janos Csirik, David S. Johnson, Claire Kenyon, Peter W. Shor, and Richard Weber, A self-organizing bin packing heuristic, Workshop on Algorithm Engineering and Experimentation (ALENEX), 1999. LNCS 1619, 246-265.

[12] J. Csirik, D. S. Johnson, C. Kenyon, J. B. Orlin, P. W. Shor, and R. R. Weber. On the Sum-of-Squares Algorithm for Bin Packing, Thirty-Second Annual ACM Symposium on Theory of Computing (STOC), 208-217, 2000.

[13] J.H. Conway and J.C. Lagarias, Tilings with polyominoes and combinatorial group theory, J. Combin. Theory Ser. A 53 (1990): 183-206.

[14] Will Evans, Claire Kenyon, Yuval Peres and Leonard Schulman. A Critical Phenomenon in a Broadcast Process. Annals of applied probability, Vol 10, No. 2, May 2000, 410-433.

[15] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In Proc. 32rd IEEE Annual Symposium on Foundations of Computer Science (FOCS), pp 2-12, 1991.

[16] Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. Proceedings of 41st FOCS, 2000.

[17] W. Fernandez de la Vega, MAX-CUT has a Randomized Approximation Scheme in Dense Graphs, Random Structures and Algorithms, 8, 3 (1996), 187-198.

[18] Approximation Schemes for Metric Bisection and Partitioning, W. Fernandez de la Vega, Marek Karpinski, and Claire Kenyon, Proceedings of the *Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004, to appear.

[19] Approximation Schemes for Clustering Problems, W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC), 2003, 50–58.

[20] W. Fernandez de la Vega and Claire Kenyon, A Randomized Approximation Scheme for Metric MAX-CUT, Thirty-Ninth Annual IEEE Symposium on Foundations of Computer Science (FOCS), 468-471, 1998. Cet article a été choisi pour publication dans un numéro de JCSS consacré à des articles sélectionnées de FOCS'98 (to appear).

[21] W. Fernandez de la Vega and G. S. Lueker, Bin packing can be solved within $1 + \epsilon$ in linear time, Combinatorica 1(4), 1981, 349-355.

[22] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using demi-definite programming, Journal of the ACM, 42(6):115-1145, 1995.

[23] O. Goldreich, S. Goldwasser and D. Ron, Property Testing and its Connection to Learning and Approximation, IEEE Annual Symposium on Foundations of Computer Science (FOCS), 1998.

[24] Mordecai J. Golin, Claire Kenyon and Neal E. Young, Huffman Coding with Unequal Letter Costs, Proceedings of the *Thirty-Fourth Annual ACM Symposium on Theory of Computing (STOC)*, 2002.

[25] H. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In Proc. 37rd IEEE Annual Symposium on Foundations of Computer Science, pp 627-636, 1996.

[26] D.S. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, Journal of the ACM, 32, 1985, 130-136.

[27] M. Jerrum and A. Sinclair, Approximating the permanent, SIAM J. on Computing, 18:1149-1178, 1989.

[28] Mark Jerrum, Alistair Sinclair, Eric Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries, Thirty-Third Annual ACM Symposium on Theory of Computing (STOC), 2001, 712-721.

[29] Anna R. Karlin and Claire Kenyon and Dana Randall, Dynamic TCP acknowledgement and other stories about e/(e-1), Thirty-Third Annual ACM Symposium on Theory of Computing (STOC), 2001.

[30] R.M.Karp. Reducibility among combinatorial problems. in R.E. Miller and J.W.Thatcher, editors, Complexity of Computer Communications, pp. 85-103. Plenum Press, New York, NY, 1972.

[31] N. Karmakar and R. M. Karp, An Efficient Approximation Scheme for the One-Dimensional Bin-Packing Problem, FOCS 23 (1982), 312-320.

[32] C. Kenyon and R. Kenyon, Tiling a polygon with rectangles, 33^{rd} Annual IEEE Symposium on Foundations of Computer Science (FOCS), Pittsburgh, PA, Oct. 1992, 610-619.

[33] Claire Kenyon and Michael Mitzenmacher, Linear Waste of Best-Fit Bin Packing on Skewed Distributions, Proceedings of the *41st Annual Symposium on Foundations of Computer Science (FOCS)*, Redondo Beach, CA, November 12-14, 2000, pp. 582-589.

[34] Claire Kenyon, Yuval Rabani and Alistair Sinclair, Biased Random Walks, Lyapunov Functions, and Stochastic Analysis of Best Fit Bin Packing , Journal of Algorithms, Vol. 27, No. 2, 218-235, 1998. A preliminary version appeared in the proceedings of the Seventh Annual ACM-SIAM Symposium On Discrete Algorithms (SODA), 351-358, 1996

[35] Claire Kenyon, Dana Randall and Alistair Sinclair, Approximating the Number of Monomer-Dimer Coverings of a Lattice , Journal of Statistical Physics 83, 1996, 637-659. A preliminary version appeared in the proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory Of Computing (STOC), 738-746, 1993.

[36] Claire Kenyon and Eric Rémila, A near-optimal solution to a two-dimensional cutting stock problem, *Mathematics of Operations Research*, 25, 4, November 2000, 645-656. A preliminary version appeared in the proceedings of the 37*th Symposium on Foundations of Computer Science (FOCS)*, 1996, 31-36.

[37] Claire Kenyon and Nicolas Schabanel, The Data Broadcast Problem with Non-Uniform Transmission Times, Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) , 547-556, 1999.

[38] Claire Kenyon, Nicolas Schabanel et Neal E. Young. Polynomial-Time Approximation Scheme for Data Broadcast, Thirty-Second Annual ACM Symposium on Theory of Computing (STOC), 659-666, 2000.

[39] L. Lovász, On the ratio of optimal integral and fractional covers, Discrete Mathematics, 13, 383-390, 1975.

[40] Claire Kenyon, Elchanan Mossel and Yuval Peres, Glauber Dynamics on Trees and Hyperbolic Graphs, Forty-Second Annual Symposium on Foundations of Computer Science (FOCS), 2001.

[41] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. Proc. 32nd ACM Symposium on Theory of Computing, 2000.

[42] E. Rémila, Habilitation à dirriger des recherches, LIP, ENS Lyon, 1997.

[43] Mike Robson, Sur le pavage de Figures du Plan par des dominos, Rapport 91-18 du Laboratoire de l'Informatique du Paralllisme, Ecole Normale Suprieure de Lyon, aussi comptes rendus des 'Journes Polyominos et Pavages' Universit Paris XII-Val de Marne, 1991.

[44] P.W. Shor. The average case analysis of some on-line algorithms for bin packing. *Combinatorica* **6** (1986), pp. 179–200.

[45] W. P Thurston, Conway's Tiling Groups, Am. Math. Monthly, Oct. 1990, 757-773.

Teaching Statement

I strongly believe in "the pleasure of becoming an educated person, the enormous interest it adds to life. To be able to be caught up into the world of thought – that is to be educated" (Edith Hamilton). In computer science, I particularly try to teach critical thinking (for example, when evaluating the pertinence of theoretical models), problem solving (for example, when creating a new algorithm), logical reasoning (typically when putting together a rigorous proof), and to promote intellectual curiosity in general.

What I enjoy in teaching is seeing the students learn. My style is inspired by remembering, from my own experiences as a student, what courses were most inspiring for me in the long run: not necessarily courses where the students learn a lot quantitatively, but courses which provide a feeling of discovery and intellectual excitement. Thus, my style in lectures is highly interactive; I like to try out the students' ideas, even though they may fail, and challenge them to find counter-examples or improvements to one another's suggestions. Once in a while, the students engage in a discussion in the classroom, and it is a thrill for me to stand back silently and watch them resolve their problems by exchanging ideas back and forth. While some students greatly enjoy this, some of the struggling students find it difficult, since they are never sure, when I start writing a suggested algorithm or conjecture, whether it is correct or whether it will end up being discarded a few minutes later. The lectures sometimes wander (of course, this is only possible for topics with which I am comfortable enough to be able to deal with new algorithms or proof attempts in real time.)

While a professor at Orsay, I had some courses with students who were not comfortable enough with the topic for that kind of teaching, and I resorted to spending part of my lectures giving rather simplistic summaries of class content. The students like it, and at least it makes them memorize a few basic facts, but this is rather frustrating to me. I try to console myself with the thought that the knowledge of certain basic facts is probably a prerequisite before the student can develop the ability to think about them.

When I started at Orsay, my somewhat idealistic viewpoint was that teaching is a rewarding activity, irrespective of the student public (it does not matter if the students are not very strong: what is gratifying is to watch them progress) and of the topic taught (every topic should be amenable to teaching the students how to think). Sadly but perhaps not too surprisingly, it turns out that I have now joined the rank-and-file: what I like best is to teach my research area (Algorithms), to a public of students who are interested a priori (i.e. in an elective course).

In exercice sessions ("TD"), I ask the students to work in small groups and solve exercises together; after a week or two of grumbling, the students adapt and always end up really enjoying that opportunity to work in collaboration with their peers. Occasionally, when the group is small enough, I do not hand out solutions not present solutions on the board, but only go from grup to group to provide help and feedback on their work; thus, a student who passively waits for the solution to the exercise will leave the class without having received anything: it is a great incentive for the students to work in class!

In summary, I admit that I currently enjoy research better than teaching, but nevertheless, teaching is an essential and gratifying part of my work, and I like to think that by teaching I am participating as well as possible in the development of the students' education.

Teaching experience before 1997

Lower level undergraduates
Pascal Programmation, greshman level, Université Claude Bernard, 20 heures (1996-97).
Math weekly oral exams (Louis-le-Grand, 1984-85, Saint-Louis, 1990-91).

Upper level undergraduates
Algorithms, junior level, Université Claude Bernard (1996-97).
Graph algorithms, Summer Math Institute, Berkeley (1996).
Graphs and Combinatorics, junior level elective cours, ENS-Lyon (1995-96).
Discrete Structures, junior level elective cours, ENS-Lyon (1995-96).
Algorithms, Berkeley (1994-95).
Algebra and combinatorics, Ecole Normale Supérieure (ENS), 1985-86.

TA in Algorithms, Princeton (1986-87).
TA in Computational Geometry, Princeton (1986-87).

Graduate level
Berkeley : 1992-93, *Group theory and computer science.*
ENS-Lyon : 1991-92 and 1996-97, *Randomized algorithms*, 1993-94 and 1995-96, *Tiling problems.*
ENS, 1990-91, *Boolean circuits*, 10 heures.

Teaching from 1997 to 2002

While a Professor of Computer Science at Paris-Sud University, from 1997 to 2002, I taught courses at the freshman, sophomore, junior, and graduate levels, as well as courses targeted to junior level coop students, and courses for Mathematics majors. I helped put together a new "Introduction to Computer Science" course for freshmen, participated in the setup of a new Java programmming junior level course, and taught Algorithms to coop students, who tend not to think abstractly, hence a syllabus more oriented towards concrete implementations.

The official teaching load in France is 192 TD-equivalent hours per year. Faculty teach lecture, "TD" sessions (exercice sessions), and lab sessions. One TD hour is equivalent to 2/3 lecture hour and to 4/3 lab hours. The tables below indicate, for each topic, the number of real hours and, in parenthesis, the number of students.

1997-98

	Math for CS	Java programming	Algorithms	Random graphs
Level	freshman/soph.	junior	coop.	graduate
Lecture	21h (12)	11h (120)	25h (70)	10h (3)
TD	21h (12)	32h (30)	35h (30)	
Lab	12h (12)			

Number of TD-equivalent hours : 196.5.

1998-99

	Intro to CS	Algorithms	Complexity and randomized algorithms
Level	freshman	coop	graduate
Lecture	9h (150)	25h (70)	20h (20)
TD	26h (30)	35h (25)	
Lab		10h (25)	

Service :
Organization of freshman and sophomore levels CS courses (30 heqTD).

Number of TD-equivalent hours: 180.2.

1999-2000

	Introduction to CS	Algorithms	Algorithms and data structures	Complexity and randomized algorithms
Level	freshman/soph.	coop	junior	graduate
Lecture	12h (150)	25h (70)		24h (35)
TD	26h (30)		32h (25)	
Lab	24h (15)			

Service :
Organization of freshman and sophomore levels CS courses (30 heqTD).
Tutoring David Gross (4 heqTD).

Number of TD-equivalent hours: 203.5.

2000-01

	Functional programming	Algorithms and data structures	Programming in C	Mixing rates of Markov chains
Level	freshman/soph.	junior	Business	graduate
Lecture	19.5h (150)	23h (160)		10h (6)
TD	24h (30)	32h (27)		
Lab	18h (15)		24h (15)	

Service :
Organization of freshman and sophomore levels CS courses (30 heqTD).
Tutoring Julien Forest (4 heqTD).

Number of TD-equivalent hours: 204.7.

2001-02

	Introduction to CS	Algorithms and data structures	Programming in C	Computer science	Approximation Algorithms
Level	freshman/soph.	junior	junior	math	graduate
Lecture	9h (150)	23h (160)		25h (35)	20h (12)
TD	16h (30)				
Lab			20h (15)		

Service :
Organization of freshman and sophomore levels CS courses (30 heqTD).
Tutoring Antoine Kremer (4 heqTD).

Number of TD-equivalent hours: 173.7.

Service

My main administrative task related to teaching consisted of organizing the freshman and sophomore levels CS courses since 1998. It consists of being the interface between the computer science department and the rest of the university for matters concerning lower level undergraduate courses; of recruiting lecturers and instrcutors for exercice sessions and lab session; and of reading the regulations coming from the Ministry of Research and trying to apply each reform bafore the next reform arrives.

Teaching since 2002

While a professor at Ecole Polytechnique, I have interacted primarily with Computer Science majors. The lecture notes for my Algorithms course are available on line.

2002-03

	Automata, Complexity	Algorithms	Approximation Algorithms
Level	senior	senior	graduate
Lecture	0	12h (65)	20h (15)
TD	18h (30)	18h (30)	0

2003-04

	Algorithms	Intro to CS
Level	senior	junior
Lecture	12h (65)	0
TD	28h (30)	0
Ranking Exam		Exam (180 students)