Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cosrev

# Book review

**The Nature of Computation. Cris Moore, Stephen Mertens. Oxford UP, (2011)**

## 1. Introduction

The nature of computation by C.Moore and S. Mertens (Oxford UP) 2011, is a beautiful encyclopedic book, which covers a large range selection of topics from the loosely denoted field of *Theoretical Computer Science* (see the synopsis of the book in the next section). The style favors intuition and clarity over full technical details, which often are left as exercises for the reader, more about this in the conclusions. I believe this style is well reflected in the cite from one of Anne Fadiman's books that heads the Preface: *The familiar essayist did not speak to the millions; he spoke to one reader, as if the two of them were sitting side by side in front of a crackling fire with their cravats loosened, their favorite stimulants at hand, and a long evening of conversation stretching before them. His viewpoint was subjective, his frame of reference concrete, his style digressive, his eccentricities conspicuous, and his laughter usually at his own expense. And though he wrote about himself, he also wrote about a subject, something with which he was so familiar, and about which he was often so enthusiastic, that his words were suffused with a lover's intimacy.* As both authors are physicists by training, a lot of the intuition comes from the physics world, and not only in the chapters on advanced topics, but right from the beginning of the book. Each chapter contains a quite complete collection of historical, bibliographical and anecdotical notes on the technical material, which by themselves make a pleasant historical reading. In my opinion, it would have been better to include in the main text the few notes that sketch proofs of stated results. Quite a few technical details in the chapters are left as exercises, mainly mathematical derivation of formulas or not too difficult proofs of Lemmas. Moreover, at the end of each chapter there is a quite extensive collection of mostly challenging problems. The exercises fill the mathematical manipulation missing from the text, otherwise the book would be too heavy, and the problems aim to complement the material in the chapter. However, as the book progresses and the topics get more advanced, it seems to me that the difference between exercises and problems blurs. The book has several levels of reading, from getting acquainted with the historical evolution of the topics it contains, to learning the technical details of the material presented. In the last level, the reader should try to solve most of the exercises, and in the chapter-by-chapter description below, for each chapter I added my opinion of some of the problems that complements the contents.

## 2. Chapter-by-chapter synopsis and analysis

In the Prologue (**Chapter 1**), the authors introduce in a very intuitive way the concept of algorithm, and succinctly explains the importance of the study of the complexity to solve problems, and the differences between different problems. The problem in this chapter give a clear indication that the authors expect the reader to have a certain maturity in the field of mathematics and computer science.

**Chapter 2** goes in a more formal way into the concepts of algorithms and resources needed to solve a problem (space and time complexity). One of the most challenging parts of this chapter are the problems and the notes at the end of chapter. For instance in problems 2.7 and 2.8 the authors make a subtle (but guided) introduction to the average complexity of algorithms, in this particular case the average complexity of the Euclid algorithm to find the gcd of two integers. Note 2.6 introduces Karatsuba–Ofman's algorithm for multiplication of two integers and note 2.11 introduces the Robertson–Seymour theorem to decide if a graph has a minor-closed property.

In **Chapter 3**, the authors cover in a single stroke the basic algorithmic techniques, *divide and conquer*, including the $n \lg n$ algorithm for the FFT (problems 3.13 to 3.16 deal with further material on the FFT, including how to modify the FFT in the case when $n$ is prime), *dynamic programming*, and *greedy* algorithms, including a brief introduction to *matroid theory*. They also include two sections on flows and cuts in digraphs. The chapter ends with an explanation of the reduction techniques between problems, a concept that will play an important role in the forthcoming complexity chapters. This chapter alone could help to supplement the material for any standard course on algorithmics, at the undergraduate level. Particularly nice are some of the problems in the extended collection (50) at the end of the chapter. For instance, problem 3.8 presents the average analysis of quicksort, problem 3.27 asks for a polynomial time solution of the maximum independent set on a Sierpinski triangle and problem 3.49 asks for a polynomial time algorithm

1574-0137/$ - see front matter
doi:10.1016/j.cosrev.2011.09.001

to minimize the energy of *spin glasses* in some particular case (this problem is almost a prerequisite for Chapters 12 and 13).

**Chapter 4** deals with needles and haystacks, i.e. the class *NP*. It goes into proving that several problems are in the *NP* class, including some known nut shells as *primality*, and also some not so well known as the *unknot problem*: given a non-intersecting curve $\mathcal{C}$ in $\mathbb{R}^3$, decide if $\mathcal{C}$ can be untied without cutting it or passing it through itself. Many of the problems introduced in this chapter will be used throughout the book. For instance, in the chapter it is proved that the *integer partition problem* is in *NPC*, that same problem will play an important role in Chapter 14. In my opinion, Chapter 4 is a chapter that follows the standard material, with a very interesting personal twist in the presentation.

**Chapter 5** introduces the class *NP*-Completeness. To avoid the use of Turing Machine, to be introduced in later, the authors use as base for the existence of a *NPC* problem, the following one: given a program on input $x$ and a witness $w$, and an integer $t$ given in unary, decide if there exists a $w$ with $|w| \leq t$ such that the program returns yes in at most $t$ steps. Although, the outline of the chapter is the classical one, the definition of reduction and the building of the class *NPC* by adding new problems via reductions, have interesting particularities. Section 5.3 is devoted to classify and explain the bag of tricks used to design reductions, and give the feeling for why they work. The following section, presents an array of arithmetic problems, for which their *NP*-Completeness is not obvious. My favorite one is the *cosine integration* problem: given a set of integers $x_1, \ldots, x_n$ decide if $\int_{-\pi}^{\pi} (\cos x_1\theta) \cdots (\cos x_n\theta)\, d\theta \neq 0$. The details are in Section 5.3.5. Section 5.5 deals with the fine edge as why for many problems, the variant with parameter 2 is in P while the variant with parameter $\geq 2$ is *NPC*, for example 2-SAT and 3-SAT, or 2-coloring and 3-coloring. Problem 5.50 asks to prove that the general case for the minimum energy spin glass problem (introduced in Chapter 3 for a particular case) is in *NPC*.

**Chapter 6** deals with the million dollar question of *P* vs. *NP*, and pushes forward the authors' hypothesis that *P* vs. *NP* is the holy grail of all the mathematical challenges between all millennium problems, posed by the *Clay Mathematical Institute*. Chapter 6 is an enjoyable chapter that could be read by a broad spectrum of people outside the field by skipping the formal proofs and definitions, so they can grasp the meaning and importance of the *P* vs *NP* question. As the authors explain, the *P* vs *NP* problem is intrinsically coupled with the survival of the power of human reasoning and creativity (see Section 6.1). Section 6.7 presents a nice survey on the classes defined by nonconstructive proofs: *PPP*, *PPA* and *PPAD* (the Pigeon principle, the polynomial parity argument and Sperner's lemma). The chapter ends with a short section, titled *The Road Ahead*, in which the authors present their intuitions on the improbability that the *P* vs *NP* problem is going to be formally provable in a short time.

**Chapter 7** deals with the different formalism for the notion of *universal computation* and its limitations, i.e. *recursion theory*. I really do not understand why it is placed in between chapters dealing with decidable complexity classes, but it is the choice of the authors and it should be respected. It starts with a beautiful section on the motivation for the chapter, which has the title: *Babbage's Vision and Hilbert Dream*, where the authors present a historical survey of the concurrent efforts to find algorithms for solving problems and the feasibility of the universal computers to implement the algorithms. They cover from the difference engine of Babbage to the Turing machine (more interesting efforts are described in notes 7.1 and 7.2). The following sections in the chapter are devoted to the main three formalisms developed for the concept of computation: *recursive functions*, the *λ-calculus* and the *Turing machine*. After, there is a section with the *Church–Turing thesis* of equivalence between the three formalisms (see also note 7.16). As is done through all the book, the exposition is intuitive but rigorous. At the end of the chapter, the authors extend the Church–Turing thesis with other "modern" models of computation: *Counter Machine* (Minsky, 1967), *Fantastic Fractions* (J.H. Conway, 1987), *Game of Life* (J.H. Conway, 1970), *Tiling the infinite plane with Wang's tiles* (Hao Wang, 1961), and *Iterated maps* from continuous dynamical systems. The exposition of these last material is done in a less formal way than in the core of the chapter.

**Chapter 8** deals with space complexity. The authors follow the canonical methodology and use two-player games for defining the classes. The authors introduce the space classes: *L*, *NL*, *PSPACE* and *NPSPACE*, and relate them to the classes in the Polynomial Hierarchy. They go into detail proving that the reachability problem is *NL*-complete and stress the difference with the *NP* class, by stating the Immerman–Szelepcseny Theorem, that the non-deterministic complexity classes are closed under complement (in particular $NL = \text{co-}NL$). They prove *Savitch's Theorem*: $PSPACE = NPSPACE$ (see also problems 8.2 and 8.3, for variation on the theorem). Section 8.7 presents the proof that generalized geography is *PSPACE*-complete (see further variations of generalized geography in problems 8.16 to 8.20) and they also prove that generalized GO is *PSPACE*-hard, and they state, but not prove, that generalized GO is *EXPSPACE*-complete. Particularly interesting are exercises 8.8.2 and 8.8.30, where the authors deal with the space complexity of 2-SAT and 2-Coloring.

**Chapter 9** is in the line of chapter 3, but here, the authors give an exhaustive tutorial (102 pages) on optimization and approximation (notice much of the basic introductory material has been explained in previous chapters). The chapter could be easily used as a basic material for an advanced course on optimization and approximation. From my point of view, the differences with some of the existing books are the great deal of emphasis on giving intuitive geometrical ideas behind the concepts, and the effort to explain things from a different perspective, for instance in the proofs by the *balloon method*. The chapter starts with a succinct introduction to maximization and minimization problems and their approximation to the optimum (or their non-approximability). It follows a section of *Linear Programming*, which includes a description on the simplex, the Klee–Minty example to show that the simplex could take exponential time, and the section ends on with a brief introduction to *smoothed analysis*. Section 9.5 deals with *duality*, and includes a *different* proof of the strong duality theorem by balloon proving. In Note 9.14, the authors sketch an interesting alternative proof using an idea from economics. Section 9.7 presents the intuitive basis of the

*ellipsoid method* and of go at length in explaining *semidefinite programming*. Problems 9.19 to 9.24 should be considered part of the main text in this section, and some of them are not easy. In the following section, with the nice semantic title of *algorithmic cubism*, the authors present the complexity of *Integer Linear Programming* (ILP), together with some problems with equivalent complexity, i.e. reducible to ILP. Using the concept of unimodularity of the constraints matrix, the authors present an argumentation explaining when instances to the ILP problems are easy to solve (i.e. in *P*). The chapter finishes with a section showing how to cope in practice with *NP*C problems. They take as case the *Traveling salesman problem* (TSP) and present different algorithms and heuristics to solve it for reasonable inputs. They use a concrete example of 42 connecting cities in the US to use as benchmark for the different techniques.

**Chapter 10** deals with an introduction to randomized algorithms and random walks. I believe the reading of this chapter should be compulsory before going into Chapters 12, 13 and 14, as the text and problems in those chapters need some of the materials presented in this chapter. After a brief introduction to randomized algorithms, the authors present the canonical examples of Karger's min-cut algorithm and Papadimitriou's randomize algorithm to find a satisfying assignment for k-SAT. In Section 10.4, the authors sketch the semidefinite programming solution for the Max Cut problem on weighted graphs, in fact to finish the proof the reader should solve problem 10.12. The next section presents the *Minimax Theorem*, again the proof is given as problem 10.13 (with plenty of hints) and in problem 10.14, the reader is asked to provide an alternative proof to Minimax, using Brouwer's fixed point theorem. The section also includes a generalization of the Minimax: *Yao's principle* (problem 10.21 presents a nice example of the power of Yao's principle). The next sections present different types of randomized algorithms; *Hashing*, fingerprint algorithms (Schwartz–Zippel) and *primality* (Miller–Rabin). Then the authors provide a discussion of the deterministic AKS algorithm for primality. It is a bit surprising that a basic technique in computer science as amplification of Monte Carlo algorithm is introduced as problem10.45. The final section in the chapter presents the randomized complexity classes *BPP*, *ZPP* and *RP*, the relations between classes are left as exercises.

**Chapter 11** deals with interaction and pseudorandom generation. The authors first introduce *Arthur Merlin* and *Zero-knowledge proofs*, following the standard presentation around the graph isomorphism problem. I believe that to understand fully the technical details, the reader should work problems 11.1, 11.7 (this one in particular), 11.13 and 11.14. The next section presents *Interactive Proofs* (*IP*) and it is proved that $IP = PSPACE$, giving a nice presentation of the arithmetization of QuantSAT. Section 11.3 deals with *Probabilistic Checkable Proofs*, where the authors first prove a weaker version of the PCP theorem, where every *NP*-problem has an exponential PCP, and the verifier flips polynomial coins and looks to a constant number of bits in the proof. The proof of the weak result uses the usual array of techniques in PCP: error correcting codes, Fourier analysis and quadratic consistency. The authors do a good job in explaining in a clear way, non-easy material. To my taste, they abuse a bit of leaving as exercises some technical points through the proof, which are not particularly difficult to solve, but slows the reading. Finally, from the *weak PCP Theorem*, they sketch Dinur's strategy of gap amplification to obtain the strong PCP Theorem. The final section in the chapter deals with pseudorandomness and derandomization. The section goes all the way up to prove that $BPP = P$ if there are exponentially hard functions in *EXPTIME* (exponential hardness is a similar condition to one-way functions but using non-uniform Boolean functions). The chapter finishes with a very intuitive disquisition on the possibility of the existence of this kind of hard functions. I found some of the problems at the end of this chapter, particularly "challenging".

**Chapters 12 and 13** could be considered as a unity covering random walks, Markov chains sampling and approximated counting. Using the *Ising model* as a motivation to present a condensed course on *Markov chains*. Following the pedagogical line of the book, the authors put a lot of emphasis in the intuition and motivations behind the concepts. Some of the technical details are deferred either to the exercises or to the problems. For example, when presenting the classical Markov chain example of a random walk on the hypercube, the authors give a nice physics interpretation in terms of the Ising model. After the book explains different methods to bound the mixing time of a Markov chain: the basic *equilibrium indicator* method, *coupling*, *coupling from the past* and *conductance*. Each method has their collection of examples. In particular, for coupling and some of its variants as *path coupling* and *coupling with stationarity*, the used example is *random coloring of a graph*. *Coupling from the past*, takes a quite extended section, with a fantastic intuition of the method using the example of counting falling leaves on a square of land. For more formal examples, they count spanning trees, random rhombus tilings and of course coloring and Ising. I have not seen previously, such a explanation of coupling from the past. In the next section, the conductance method is applied to show that a lazy random walk on any undirected graph has mixing time $O(n^4 \log n)$. As a Corollary to that last result, they present a randomized algorithm for the *undirected reachability* problem in an undirected graph. Evidently, the "well known" application of coupling is given in Chapter 13 to approximate the permanent. Chapter 12 ends with two sections; one on the *spectral gap* and random walk on the cycle, and another section on *expanders*. This final section ends with a subsection on the *zig-zag product*, where the authors present an application of the zig-gag product to derandomize the algorithm obtained previously for undirected reachability. Problem 12.46 asks the reader to prove that the Margulis expander defined inside Section 12.9 is indeed an expander. The problem is full of hints and in my opinion, it could have been part of the text. Chapter 13 deals with the complexity and approximation of counting problems. The chapter would be difficult to read, without a previous exposition to complexity theory, for example Chapters 4 to 6 and 8 of the book under review. The chapter begins by giving a $O(n^2)$ algorithm to compute the *determinant* of a $n \times n$ matrix, and showing that it is equivalent to *counting the number of spanning trees* in an undirected graph, so that this counting problem is in *P*. Problem 13.4 uses the Laplacian to give an alternative poly-time solution to counting trees. After,

the chapter presents a full explanation of the equivalence between *counting perfect matching* in a bipartite graph and evaluating the *permanent* of its adjacency matrix, showing the technique used for computing the determinant, would not work for the permanent. Section 13.3 explores the counting classes, #P and #P-complete and goes all the way into proving the #P-completeness of the permanent and a few other counting problems, like counting number of satisfying solutions to an instance of 3-SAT. problems 13.10 and 13.11 ask to show that other counting problems in the class #P-complete, among them the #2-SAT. In the notes at the end of the chapter, the authors provide a sketch of Toda's result separating the counting classes from the polynomial hierarchy. The following section begins with a very nice intuition about the important fact that counting is equivalent to sampling. Then, the authors present a rigorous proof of the fact that for any #P-complete self-reducible problem, approximating in polynomial time the number of its solutions is equivalent to almost uniformly sampling. After, there is a careful description of the classic result of how to approximate #perfect matchings (and the permanent) in polynomial time. Section 13.6 shows that counting perfect can be done in polynomial time if the input graph is planar, because for those graphs the permanent can be computed, in polynomial time, from computing the determinant of the weighted adjacency matrix of the underlying graph. The section also contains how to calculate asymptotic properties such as the number of perfect matchings in a infinite lattice (dimer coverings in the physics argot). In the problems at the end of the chapter, the reader can find many other examples. The section contains an interesting consideration on the different meaning of *what is a problem* for the physics and the computer scientist communities. As a curiosity, problems 13.30 to 13.34 present a tour of different facets of the Tutte polynomial for graphs. In Chapter 12, the 2 dimensional Ising model is used as one of the driving examples to develop the theory of random walks and Markov chain. In Section 13.7, the authors present the *solution* to the 2-Ising problem, i.e. computing the energy and magnetization as a function of temperature. The section boils to a crash mini-course on statistical mechanics, but I believe the reading of this could be very profitable to readers from outside the physics community.

**Chapter 14** presents the basic techniques to investigate the hard-easy *phase transition* of some difficult (NP-hard) problems. This is a field where the authors of the book have important research contributions. The chapter starts by presenting the DPLL algorithm for finding a satisfying assignment for 3-SAT and empirically proving that for a density value of 4.2 (the ratio between clauses and variables), the 3-SAT has a phase transition, which goes from most inputs being satisfiable to be non-satisfiable. After the authors introduce the threshold conjecture and Fridgut's theorem. Problem 14.17 asks for the proof of the theorem for the simpler case of 2-SAT (using methods explained later in the chapter) and note 14.4 goes at length at explaining the significance of the theorem and gives a hint of the proof of the theorem. Section 14.2 presents the first analytical example of phase transition, the giant component in Erdös–Rnyi graphs. A few (not too easy) technical facts are left as exercises. Moreover, notes 14.5 and 14.6 are important to read as well as problems 14.3 and 14.8. The authors also present the existence of a phase transition for the $k$-core in the $G_{n,p}$ graphs. One of the nice features of this chapter, is that the text includes plenty of plots giving a visual evidence of the analytical developments. Section 14.3 presents the methodology to find lower bounds to the phase transition of 3-SAT, analyze concrete algorithms. This shows the details for two concrete algorithms: the unit clause algorithm and the short clause algorithm. The analysis of the algorithms uses the differential equation method, and maybe the reader should start with note 14.7, to understand the general framework of differential equations to the analysis of randomized algorithms. The section ends with a presentation of Achlioptas' card game to model the use of differential equations in analyzing randomize algorithms. Section 14.4, turns into finding tight upper bounds for the phase transition. It starts with the direct application of the first moment, and it continues with a nice exposition of the application of the second moment to find upper bounds for 3-SAT, the failure of the second moment for $k$-SAT and the intuition why it fails (the authors prove that the method works for a variation of $k$-SAT, the NAE-$k$-SAT problem). After, they introduce the *weighted second moment* of Achioptas and Peres to obtain a solution for $k$-SAT. Most of the discussion on different models of random SAT formulas (or random graphs) are presented as problems (14.10 for the configuration model and 14.24 for using $G_{n,m}$ instead the $G_{n,p}$). Section 14.5 presents the full analytical development for obtaining upper and lower bounds to the phase transition of the *integer partition problem*. This chapter gives a very clear exposition on the second moment method. Bounds for phase transition for the colorability of $G_{n,p}$ are given in problems 4.21 and 14.22. From Section 14.6 until the end of the chapter, the authors present the physics approach to phase transition, which yields exact values but the analysis is not fully rigorous: the *message passing methods* and in particular the *belief propagation* (also known as cavity, in the physics community), and its variant *survey propagation*. The last two sections deal with the geometry of solutions for 3-SAT and analytical considerations on the survey propagation method.

**Chapter 15** (the last chapter) is devoted to fundamentals of *quantum computation*. After one introductory section of the plausibility and implications of the *Physical Church–Turing thesis*, namely that a classical computer can simulate any quantum machine with the precision we desire, the authors present a crash introduction on the basics of quantum theory. The text appeals to the concepts described previously about randomized algorithms to give intuitions about quantum phenomena. In Chapter 10, after the presentation of the randomized and the deterministic primality test, I was a bit surprised of not finding a section on the RSA (in Section 11.4.2, there was a brief introduction to cryptography from the point of view of one-way functions and pseudorandom generators). The reason is that in Chapter 15, the authors devote a subsection to RSA and cryptography, as introduction to *Shor's quantum factoring algorithm*, it includes Euler's totient function and the machinery for the RSA. Using the fact that the authors give an extensive description of Shor's polynomial time algorithm for factorize, they define the quantum complexity class *BQP*, and the authors prove that

besides factoring, other cryptography related problems also belong to *BQP*. I was a bit surprised of the little the Chapter 15 deals with complexity issues, which seems a twist in style with respect to the previous chapters. For instance, the $NP \not\subseteq BQP$ hypothesis could have yield some comment about the present and future of hypothetical quantum machines solving problems in *NP*C, or a specific mention (problem) on the relation of *BQP* with *P*, *BPP* and PSPACE. Section 15.6 presents the quantum approach to find a polynomial time algorithm for *NPI* problem (problems that may lie between *P* and *NP*C). The last two chapters present *Grove's algorithm* and *quantum random walks*.

The book ends with an **appendix** where the authors give a brief review of the main mathematical techniques that are used throughout the book: asymptotic notation, inequalities, a quite complete introduction to probability (including a nice subsection on the second moment method), random walks, concentration inequalities (all the way to martingales and Azuma). The last two sections in the appendix deal with Laplace's method to evaluate the asymptotic behavior of integrals and a brief remainder on modular arithmetic. In 28 pages the authors do a good job in presenting those techniques, to the readers that already have a solid mathematical maturity.

## 3. Conclusions

I believe this book should be in the shelf of every researcher working in algorithmics, complexity, discrete mathematics, statistical mechanics and in general of all people interested in recent and future trends of these fields (including professionals of other disciplines such as economists or biologists).

The reader interested in working out the technical details in the book will need a solid knowledge of multivariate calculus, algebra and probability, or have somebody explain him/her the technical stuff. However, as I already mentioned, there is a playful level of reading the book, to get the flavor of the development of an exciting field of research that intersects the disciplines of computer science, physics, mathematics and even economics. Moreover as the book is written in a charming literary English, its reading is a pleasant experience.

Anybody teaching an algorithmics or complexity course, at graduate or undergraduate level, will find the material presented in a very intuitive manner, with nice examples and motivation of the corresponding topic. On the other hand, it would be necessary to present to the students some of the proofs left in the book as exercises. For instance, Chapters 4–8 can be used as a textbook for an undergraduate complexity course. For computer science and mathematics students the book has the great advantage of the examples from the physics world, which in my opinion is an important feature to transmit to the students. The more advance material can be easily used for graduate courses or seminars. For example, Chapters 12, 13 and 14 by themselves could be a perfect basic text for an advanced course in probabilistic methods in computer science and discrete mathematics. I hope future readers enjoy the book as much as I did.

Josep Díaz
*LSI, UPC, Spain*
E-mail address: diaz@lsi.upc.edu.