

CSC2421: Online and Other Myopic Algorithms

Fall 2025

Allan Borodin

September 3, 2025

Week 1

Course Organization

- This course is appropriate as a foundational course for anyone who has the equivalent of our undergraduate algorithms course CSC373. (If you do not have this prerequisite then please speak to me as I think this course will not be appropriate.)
- In addition, the course would also be appropriate for anyone interested in undertaking research relating to the scope of this course.
- **Sources:** The main source is a text (with the same title as the course title) that I am now writing with Denis Pankratov.
NOTE: For the last 3-4 years, I say that we expect the text to be completed by the end of the academic year.
- In addition, there are course notes at other Universities, slides for this course and my previous courses, various textbooks, and relatively current research papers. See links posted on the course web page.
- Depending on the size of the class, I will have about 3 assignments and possibly some student presentations.

More organization

- This is a very active field which is one of the reasons we haven't finished the text. I will attempt give a sense of the different aspects of this field (according to the current planned Chapters).
- There is also quercus and Piazza page which I encourage as a good way to share comments and questions. I particularly welcome students answering questions. But I most prefer in class discussions. There are links to Piazza and Markus on Quercus. All assignments will be submitted and graded on Markus.
- My contact information : bor@cs.toronto.edu The course web page is www.cs.toronto.edu/~bor/2421f25
- The prerequisite for this course is CSC373 or an equivalent. If you do not have this required background, this is probably not an appropriate course for you.

About the Title and the Course

An **online algorithm** receives its input as a sequence of *input items*. When a new input item arrives, the algorithm needs to make an irrevocable decision about the input item. We think of the inputs arriving in discrete steps; that is, input item I_j arrives at step j . We may or may not also have a real time t associated with each input item.

When considering online algorithms, we initially use the term *competitive ratio* rather than approximation ratio. Informally, this is a “worst case” (over all possible input sequences) ratio for the performance (e.g, profit or cost) for an algorithm’s solution relative to an optimal solution. We will then extend the competitive ratio to models that go beyond the initial online model.

Chapter 1 includes for a brief history and a couple of online algorithms.

In Chapter 2, we formalize *deterministic online algorithms* in terms of *request answer games*. We then formalize the competitive ratio of a deterministic online algorithm and discuss four more basic examples of online algorithms.

More About the Chapters

Chapter 3 formalizes randomized online algorithms and the different adversaries. We then consider some examples where randomization can help and when they can't help.

Chapter 4 discusses a number of classical applications of online algorithms

In Chapter 5, we introduce two abstract formulations, namely the k server problem and the metrical task systems for online problems in the context of metric spaces. In particular, we discuss the k -server conjecture.

Chapter 6 is a very brief introduction to online learning and regret and the relation to online algorithms and the competitive ratio. Online learning is a field of study that has its own texts and history.

We return to competitive analysis in Chapter 7 where we consider online graph algorithms.

Chapter 8 discusses Extensions of some of the Classical Problems previously introduced.

Chapters continued

Chapter 9 is about online mechanism design and online social choice theory (e.g., voting and fair allocation). Both these topics involve agents who have self interests. We have to reconcile these self interests with some global objective such as revenue or “social welfare”.

In Chapter 10, we consider online algorithms based on linear programming and primal dual algorithms. Primal dual algorithms provide a general approach to designing online and offline algorithms.

In Chapter 11, we consider MaxSat and Submodular Maximization. MaxSat is the basic constraint satisfaction problem. Sat and Max-Sat contests are regularly run on real world benchmarks to evaluate the performance of algorithms. It is surprising how well simple algorithms can do in comparison to state of the art (but slower) algorithms.

And yet more chapters

In Chapter 12, we study stochastic online settings.

In the initial works on *competitive analysis*, the input item sequence can be completely *adversarial*. Arguably, the most studied model for stochastic inputs is when each item is drawn independently from some distribution where these distributions can be known or unknown in advance. The distributions for each input may be identical and independent (i.i.d), or independent but not necessarily identical.

Online items can also come from non-identical processes such as Markov and Poisson arrivals.

Another example of *stochastic inputs* is when an adversary provides an input set but the input sequence is a random permutation of the items in the input set. This is called the random order model (ROM).

In Chapter 13, we use tools from online convex optimization to construct randomized online algorithms. In particular, this methodology seems to be the best promise for the randomized k server problem.

Next Four Chapters: myopic algorithms that extend the basic online model

I am using the term *myopic algorithm* to refer to algorithms that have limited knowledge about the set of inputs.

- *Online algorithms with advice* refers to an online algorithm that can first obtain a limited amount of information about the input set. In particular, we consider online algorithms with predictions, sometimes referred to as online algorithms with ML advice. See Chapter 14.
- In the literature of scheduling algorithms, the term online algorithm often refers to what I would call a “real time algorithm” where decisions about an input item arriving at *time* t can be made at any time $t' \geq t$. See Chapter 15.
- In Chapter 16 we study online algorithms with recourse; that is, algorithms that allow limited changing of previous decisions.
- In Chapter 17, we discuss streaming algorithms and dynamic algorithms, topics that deserve their own texts and courses. The emphasis here is on tradeoffs between resources (space for streaming, time for dynamic algorithms) and performance.

Final Chapters

- We consider offline greedy algorithms (and some extensions) to be myopic algorithms. This is formalized in Chapter 18 in terms of priority algorithms. Priority algorithms allow for some fixed or adaptive ordering of the input items but still when processing the i^{th} input item, we do not have complete knowledge of the remaining items.
- Chapter 19 discusses alternative measures of performance of an online algorithm.

End of Week 1

We actually ended at slide 7 but I am including the last two slides to complete the brief discussion of the chapters. We will start week 2 by discussing the contents of Chapters 14-19.