CSC2421: Online and Other Myopic Algorithms Fall 2026

Allan Borodin

September 9, 2026

Lecture 1: What is this course about

While the title may seem very specialized, the course should be accessible to any CS student and also serves as a partial introduction to many of the topics in CSC2420. The course will be based on a two volume text co-authored with Denis Pankratov. We completed Volume 1 at the end of May but the text is not available yet. We are actively working on Volume 2 which we hope will be completed by the summer 2027. It is a "self-contained" text for anyone who has the equivalent of our CSC373 undergraduate text.

The adjective "myopic" implies algorithms where we do not see all the input items in advance; that is, after processing the first i items we have limited knowledge about item $i+1$ and other future items. We will take a broad view of this term and will include (for example) algorithms that have some limited ability to change past decisions, algorithms that are given some advance "advice" about the input, and "greedy-like" algorithms which allow some limited ability to control the order in which input items appear. We will consider both worst case and stochastic inputs.

Some motivation

The topic of online myopic algorithms is of growing interest due to the increasing number of applications. We will consider various applications such auctions, fair division, scheduling, learning theory, online adverstising.

Of course, nothing seems to be as fast growing as the progress in Machine Learning (ML) and Large Language Models. It is impossible to predict what (if any) limits there are on the capabilities of LLMs. And that is maybe additional some motivation for why the study of online algorithms (and, in particular, worst case analysis) may be of particular relevance now that we are in a world where predicting the future seems impossible.

What is clear is that the way we conduct courses and do research has changed. I started using ChatGPT and, in particular, asked about an open problem in a recent paper (with two students from last years course). ChatGPT did not solve the problem but did find our paper (on the arXiv) as the most relevant paper and was able to make some coments of some limited use. Confession: So far, I am cheap (with regard to my grant money) and only purchasing the $25/month version.

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 to discuss whether or not the couuse will be approriate.
- The course is also be appropriate for anyone interested in undertaking research relating to the scope of this course. Online and other myopic algorithms are part of a broader interest I have in "conceptulaly simple algoroithms".
- In addition to our text, there are course notes at other Universities, slides for some of my previous courses, various textbooks, and relatively current research papers. See links posted on the course web page. And, of course, there are LLMs.
- Depending on the size of the class, I will probably have two assignments and (written and oral) presentations. Note: It is a challenge to ask questions that I know how to solve and ChapGPT can't solve.

More organization

- As I said, 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 Chapters).
- There is also quercus. I am not sure if I want to use Piazza, I most prefer in class discussions. There is a link to Markus and Piazza 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/2421f26

About the Title and the Course

In the initial online model, 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 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 extend the competitive ratio measure to other myopic algorithms.

While we intended to create a single text the amount of material was getting too large and we were not seeing an end in sight. So we decided to create two volumes. Volume 1 focucses on the basic online model and competitive analysis. Volume 2 contains more general online models and other myopic models. Here follows a very brief summary of each chapter.

Volume 1 Chapters

Chapter 1 includes for a brief history and three "classic" online algorithms; namely, the ski rental problem, the line search problem, and paing/caching.

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.
Chapter 3 formalizes randomized online algorithms and the different adversaries. We consider some examples where randomization can substantially help and when randomization cannot help. We introduce the Yao Principle which we use to prove negative results about the competitive ratio of randomized algorithms.

Volume 1 Chapters continued

Chapter 4 discusses the potential function method and some other classical applications of online algorithms.

In Chapter 5, we introduce two abstact online 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. Amongst other graph problems we consider the online bipartite matching problem and its relation to online advertising.

Chapter 8 discusses Extensions of some of the Classical Problems previously introduced. Namely, we discuss ski rental; extensions, bin packing extensions, and k-serever extensions.

Volume 1 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 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 suroprising how well simple algorithms can do in comparison to state of the art (but slower) algorithms.

This is the end of Volume 1

Volume 2 Chapters

In Chapter 1, 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 2, we study stochastic online settings.

In the initial works on *competitive analysis*, the input item sequence was completely *adversarial*. But, of course, stochastic input sequences have always been considered. 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 (i.d.).

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).

Volume 2 Chapters continued

In Chapter 3, we use tools from online convex optimization to construct randomized online algorithms. In particular, this methodology may be the best promise for the randomized k server problem.

The next four Chapters concern more general online and other myopic algorithms that extend the basic online model in various ways. .
I am using the term *myopic algorithm* to refer to algorithms that have limited knowledge about the set of inputs. Of course, we already have limited kowledge when we consider stochastic inputs.

In Chapter 4, we encounter *Online algorithms with advice* in which an online algorithm can first obtain a limited amount of information about the input set. In particular, we consider online algorithms with predictions, sometimes refered to online algorithm with ML advice.

Chapter 2 Chapters continued

Chapter 5 introduces the *real-time model*. In the literature of scheduling algorithms, the term online algorithm often refers to what I call a "real-time algorithm" where decisions about an input item arriving at *time* t can be made at any time $t' \geq t$. Real-time algorithms often allow for some form of *preemption/interuption*. We consider preemption with revoking, restarting, and resumming.

In Chapter 6 we study online algorithms with recourse; that is, algorithms that allow limited changing of previous decsiions. This general area is called online algorithms with recourse. We also consider parallel and multi-pass online algorithms.

In Chapter 7, we discuss streaming algorithms and dynamic algorithms, topics that ieach 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

In Chapter 8, we discuss offline greedy algorithms (and some extensions). This is formalized in terms of priority algorithms. Priority algorithms allow for some fixed or adaptive ordering of the input items but still when precessing the i^{th} input item, we do not have complete knowledge of the remaining items.

Chapter 9 discusses alternative measures of performance of an online algorithm. We conclude the text by returning to comments made at the start of Volume 1 about the paging problem. Namely, we observed that the basic competitive analysis model does not account for *locality of reference*. The Markov paging model (in Chapter 2) is one natural approach to model locality of reference. In Chapter 9 we considerthe Working Set model originally proposed in the 1960s by Denning.