

5 minutes with Marc Renault



Marc Renault is currently a CNRS postdoc at the Institut de Recherche en Informatique Fondamentale (IRIF; formerly LIAFA), Université Paris Diderot – Paris 7. He visited our Department in June 2016 to speak at our Seminar on Combinatorics, Games and Optimisation about “The Bijective Ratio of Online Algorithms”. Whilst he was in London, Tom Lidbetter took the opportunity to discover more about Marc’s interests, both in his research and beyond.

What is an online algorithm?

An algorithm is a set of instructions to process the input for a given problem. One classic analogy is to see it as a cooking recipe: the algorithm is the cooking instructions, the input is the ingredients and the resulting dish is the output. In the classical setting, algorithms have access to the entire input and the algorithm is a function applied to this input. The result of the function being the output. In contrast, in the *online* setting, the input is revealed sequentially, piece by piece; these pieces are called requests. Moreover, after receiving each request, the algorithms must take an action before the next request is revealed. That is, the algorithm must make irrevocable decisions based on the input revealed so far without any knowledge of the future input. In other words, at each request, a function is applied to the revealed input. Since the future is unknown, these decisions could prove very costly.

Online problems have many real-world applications, where the input is sequential and each request demands an immediate action. One such example is the problem of managing the fast memory in computers. This is called the *paging problem* and it consists of: a universe U of pages (the data in the slow memory) and a cache of fast memory that can hold k pages. The input is a sequence of requests for pages in U . A page fault occurs when a requested page is not in the cache; in such a case, a page in the cache must be replaced by the requested page. The goal is to minimize the number of page faults over the request sequence.

What is competitive analysis of online algorithms?

The standard technique for evaluating the performance of online algorithms is called *competitive analysis* and it was developed with the aim of understanding the performance of an algorithm with no knowledge of the future against an algorithm with full knowledge of the future. This is done by determining the maximum ratio, called the *competitive ratio*, of the value of the solution produced by the online algorithm as compared to the value of the optimal solution for any fixed-length request sequence. This ratio can be seen as the price for not knowing the future. It is a simple and effective framework for evaluating the performance of online algorithms and has been instrumental in making online computing a well-established field of theoretical computer science.

Even though competitive analysis is the yardstick for analyzing online algorithms, it sometimes fails to distinguish between algorithms for which experimental evidence suggests a significant difference in terms of performance. This was even noted in the seminal work of Sleator and Tarjan. The classic example is the paging problem. Competitive analysis says that the strategies LEAST RECENTLY USED (LRU) (on a page fault, replace the least recently used page) and FLUSH WHEN FULL (FWF) (on a page fault, remove all the pages from the cache) have the same competitive ratio, whereas, in practice, LRU is much better than FWF. Such disconnects between theory and practice have motivated the study of alternatives to the competitive ratio such as the *bijective ratio* (or more generally *approximate stochastic dominance*), the *topic of my talk*.

What is the bijective ratio and how does it enrich the analysis of online algorithms?

Consider two lotteries X and Y . Say that the probability to win £ c is higher in X than in Y for

any value c . That is, you are more likely to win \$10, \$100 or \$1,000,000 in X than you are in Y . Obviously, you would prefer to buy a ticket for X than Y (maximizing profit), and, from the lottery perspective, they would prefer you buy a ticket in Y (minimizing cost). This notion is called *stochastic dominance*. That is, X stochastically dominates Y . This technique is often used in decision theory and microeconomics, and it has been used for evaluating the performance of online algorithms, but less so than competitive analysis. When analyzing online algorithms using stochastic dominance, if the requests are drawn from a uniform distribution, we have *bijjective analysis*. This stipulates a very stringent relationship between the compared algorithms. Suppose now that X and Y are algorithms, and we are interested in minimizing the cost over a request sequence. We say that Y is *bijjectively better* than X if X stochastically dominates Y when the requests are sampled according to a uniform distribution. This relation not only implies that the average cost of Y is no more than the average cost of X , but that Y has at least as many request sequences as X that engender a cost below any threshold c . In previous works, these techniques have been able to provide a clear separation between algorithms where competitive analysis did not; the paging problem being the most notable. However, there are situations in which it may be too difficult to establish this relationship analytically, or it may not even exist.

For this reason, we introduce the *bijjective ratio*, where we extend bijjective analysis in the spirit of the competitive and approximation ratios. That is, we introduce a p -approximate stochastic dominance. Going back to the lottery example, this would be that you are more likely to win $\pounds c$ in X than $\pounds(p \cdot c)$ in Y . This notion maintains the essential aspect of bijjective analysis in that the average cost of Y is no more than p times the average cost of X and that Y has at least as many request sequences that engender a cost below $p \cdot c$ as X has request sequences that engender a cost below c for any c . Moreover, this relaxation makes these analysis techniques amenable to all online problems, and they can be used both to compare two online algorithms, or to compare an online algorithm to the optimal offline algorithm.

What other areas of research are you interested in?

In general, I am interested in the design and analysis of algorithms. In particular, I tend to be interested in computational models that move away from the conventional input- calculation-output scheme. Recently, I've been working on searching and patrolling games, specifically, ones that are inspired by biological systems.

Paris must be an exciting place to live. How do you spend your time when you're not working?

Outside of work I'm busy with family life; my wife and I have two boys, aged 8 and almost 5. I have a passion for cooking, and Paris is a perfect place for this hobby with ready access to great ingredients, amazing cheeses and wines. And, thanks to a great children's cookbook suggestion of [Bernhard von Stengel](#) during my visit to the LSE, my boys are joining me in the kitchen.

September 20th, 2016 | [5 minutes with...](#), [Featured](#), [Game theory](#) | [0 Comments](#)

