The power of branching and learning for online scheduling and bin packing problems

The field of online computation is a major focus of algorithm design and analysis in the areas of scheduling, routing, packing and caching since the 1970s.

In the standard model for theoretical computer science, it is assumed that the entire input for the problem is known in advance and every part of it is fully accessible before an output needs to be produced. To illustrate this, we can imagine that we have a set of n items before us, each with its own non-negative weight in kilograms, and the question posed to us (or our computer) is to pack these items into as few plastic bags as possible, where size is no issue for such bags, but each bag can carry at most 1 kg. (This problem is known as *bin packing*, and it will show up later in our proposal.)

A significant advantage of this standard model is that we, along with our computer, have all the time to analyze the items before us, being able to create elaborate and effective packing strategies, especially if some combination of items combined sum up close to the threshold of 1 kg.

In contrast to this, the *online computation model* assumes that items arrive one by one, quite similarly to how they would arrive to a bagger employee at a North American convenience store. We are tasked again to pack these items into bags of capacity 1 kg, but we learn the weight of an item only once it arrives, and we need to immediately decide if we are to pack it into one of our already used bags (provided it fits) or open a new bag and pack this item into it.

In the online computation model, we compare the objective of the algorithm to the objective of the optimal algorithm that learns the entire input at the start of the instance and can base its decisions on the whole input instance. The computation time of the online algorithm is not restricted.

If we compare one algorithm to all possible instances, the worst-case ratio of the performances is called the *competitive ratio*. For a particular problem, discovering the best-possible ratio is equivalent to discovering the best-possible online algorithm. In the 50 years since the online computation model's inception, several fundamental problems as well as strong algorithms for those problems have manifested, but for many classical problems in the field the optimal competitive ratio is still unknown.

To illustrate the gaps, we can mention the problem Online Scheduling on Parallel Identical Machines, where the interval between the best known lower and upper bounds is $[1.888, 1.920]$ or Online Bin Packing, where the current gap is $[1.542, 1.578]$.

In the last few years, a new approach has proven to be successful for computing lower bounds for specialized settings of the problem Semi-online Scheduling with Known Optimal Makespan, a close relative of the general Online Scheduling on Parallel Identical Machines. This problem has currently a gap of $[4/3, 1.5]$, and a significant open question, standing for over twenty years, is to improve the lower bound of $4/3$.

The idea of the new approach is to reformulate the problem as a finite two-player game, and then apply the Minimax algorithm to evaluate the game. This approach has yielded results for settings with up to 8 machines, but the general lower bound of $4/3$ is still the best known for the general problem.

The central thesis of this proposal is that we can use the notion of *learning* in a two-fold way: First, that learning increases the technical potency of the Minimax search, yielding results for a larger number of machines. Within this proposal, we already provide preliminary progress on this front. Second, we believe the results produced by our learning-augmented computer search will lead to improving the general lower bound of $4/3$. Our hypothesised lower bound of $19/14$ would mean progress on a 20-year-old open problem.

Our secondary thesis is that the results provided by the Minimax search technique are quite *branching* in their nature, in contrast to the best-known lower bounds for Online Scheduling on Parallel Identical Machines and Online Bin Packing, where branching is only applied in a very controlled way. Thus, we propose to investigate how much the power of branching can be applied to improve the state of the art for the aforementioned problems. Towards the success of this theoretical investigation, we wish to combine our insights from the world of Semi-online Scheduling with Known Optimal Makespan with expertise from other research teams in the world that are currently working on closing the gaps in the competitive ratio of these and similar online problems.