Contribution ID: **11** Type: **not specified**

New Partitioning Techniques and Faster Algorithms for Approximate Interval Scheduling

Tuesday, July 11, 2023 12:10 PM (20 minutes)

Spencer Compton, Slobodan Mitrović and Ronitt Rubinfeld

Abstract: Interval scheduling is a basic problem in the theory of algorithms and a classical task in combinatorial optimization.
We develop a set of techniques for partitioning and grouping jobs based on their starting and ending times, that enable us to view an instance of interval scheduling on \emph{many}
jobs as a union of
multiple interval scheduling instances, each containing only \emph{a few} jobs. Instantiating these techniques in dynamic and local settings of computation leads to several new results.

For $(1+\epsilon)$-approximation of job scheduling of n jobs on a single machine, we develop a fully dynamic algorithm with $O(\frac{\log n}{\epsilon})$ update and $O(\log n)$ query worst-case time. Further, we design a local computation algorithm that uses only $O(\frac{\log N}{\epsilon})$ queries when all jobs are length at least 1 and have starting/ending times within $[0, N]$.
Our techniques are also applicable in a setting where jobs have rewards/weights. For this case we design a fully dynamic \emph{deterministic} algorithm whose worst-case update and query time are $poly(\log n, \frac{1}{\epsilon})$.
Equivalently, this is \emph{the first} algorithm that maintains a $(1+\epsilon)$-approximation of the maximum independent set of a collection of weighted intervals in $poly(\log n, \frac{1}{\epsilon})$ time updates/queries.
This is an exponential improvement in $1/\epsilon$ over the running time of a randomized algorithm of Henzinger, Neumann, and Wiese ~[SoCG, 2020], while also removing all dependence on the values of the jobs' starting/ending times and rewards, as well as removing the need for any randomness.

We extend our approaches for interval scheduling on a single machine to the setting with M machines. In one method of extension, we achieve similar approximation factors at the expense of a slower update time in the dynamic setting.

Presenter: COMPTON, Spencer

Session Classification: Track A-1