Algorithm Selection using Reinforcement Learning

Michail Lagoudakis and Michael Littman
Computer Science Department
Duke University

Algorithm Selection

Sorting:

- QuickSort is recursive and generally fast for large sizes
- InsertionSort is non-recursive and fast for small sizes

Order Statistic Selection:

- Deterministic Select is recursive and good for indices near the median
- Heap Select is non-recursive and good for indices near the ends

How should the algorithm be selected?

Sorting

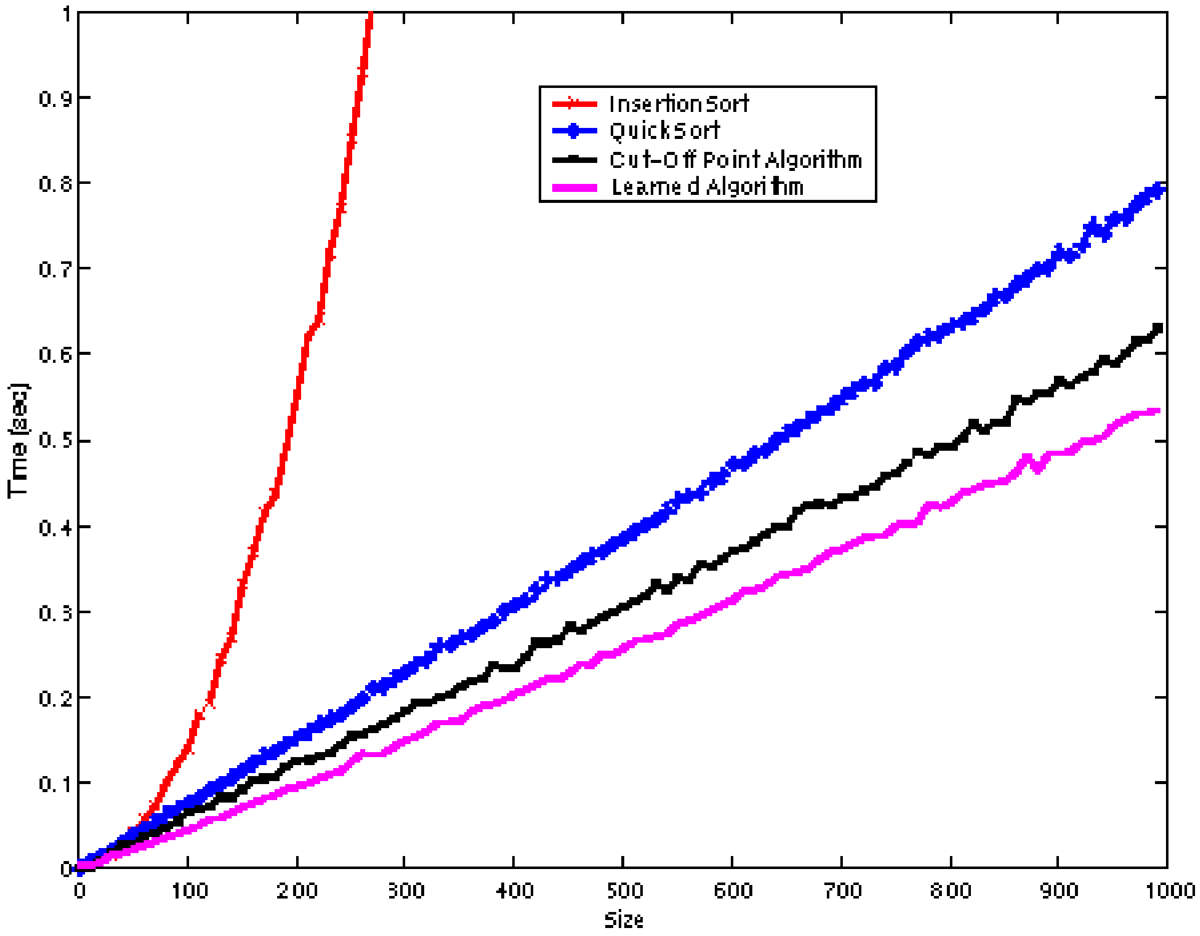

Figure 7. Running times for sorting.

Order Statistic Selection

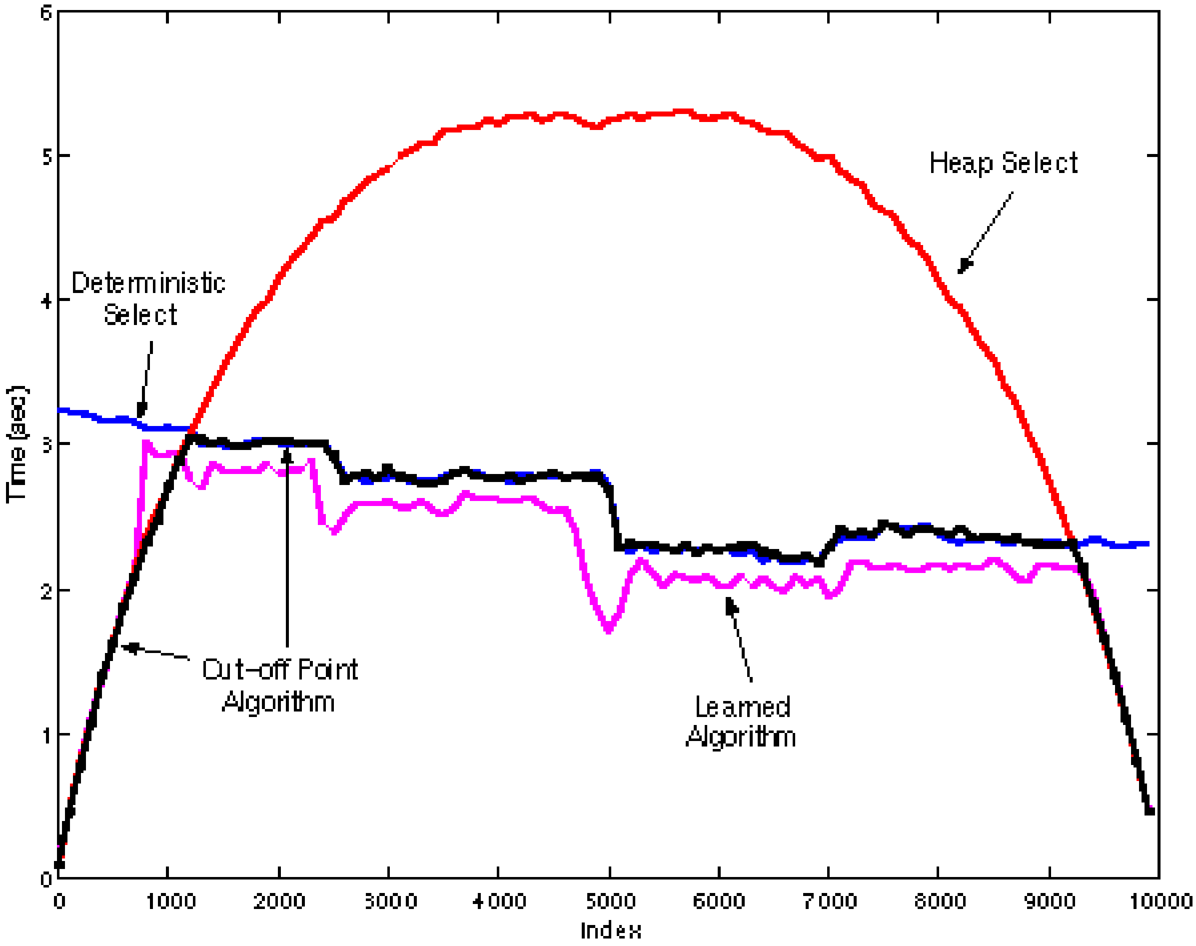

Figure 3. Results for order statistic selection (tabular case).

Use RL!

- State = input parameters to current recursive call
- Action = algorithm to apply
- Immediate Cost = time to pre- and post-process recursive calls or to perform non-recursive algorithm
- State Transition = ?!?   Two next states!

Fix:

- Do no exploration on one subproblem
- Obtain a "Monte-Carlo return" for this subproblem and add this to the Immediate Cost function
- Explore and use Q-learning on other subproblem
- Usually explore on larger subproblem