CS/CE 412/471 - Project Proposal

Musab Kasbati, Meesum Abbas

1 Paper Details

Title: A nearly optimal randomized algorithm for explorable heap selection
Authors: Sander Borst, Daniel Dadush, Sophie Huiberts, and Danish Kashaev
Conference: Integer Programming and Combinatorial Optimzation 2023
Year: 2023
DOI: 10.1007/978-3-031-32726-1_3

2 Summary

This paper introduces a randomized algorithm for the explorable heap selection problem, which seeks the nth smallest element in a binary heap with traversal-based access. The authors achieve an expected time complexity of $O(n \cdot log^3(n))$ using $O(log(n))$ space, significantly improving prior results. The algorithm leverages recursive exploration and random sampling to balance efficiency and memory constraints, particularly relevant for branch-and-bound algorithms in optimization. A lower bound of $\Omega(n \cdot log(n)/log(log(n)))$ is established, demonstrating near-optimality. The work bridges theoretical advancements with practical applications in memory-constrained environments like integer programming solvers.

3 Justification

This paper is theoretically intriguing as it introduces a randomized approach that breaks the long-standing $n \cdot exp(O(log(n)))$ runtime barrier for explorable heap selection, achieving a near-linear $O(n \cdot log^3(n))$ time. It does so by recursively leveraging randomness and locality to minimize backtracking, which is a key bottleneck in memory-constrained search. This directly impacts practical optimization tasks (e.g., MIP solvers) by enabling faster, space-efficient exploration of large decision trees which is crucial for real-world problems like resource allocation or routing.

4 Implementation Feasibility

There are no easily accessible resources for implementation, i.e. no code repositories available on GitHub, authors' websites, nor on arXiv search. However, the paper provides a Pseudocode implementation of the algorithm.

5 Team Responsibilities

The plan is for both teammates to read and discuss the paper for a thorough understanding. Coding tasks will be divided based on preference, including algorithm implementation and testing. A joint brainstorming session will determine an application or visualization, which will be modularized for individual contributions. The report writing task will be divided such that each member writes about their respective work.