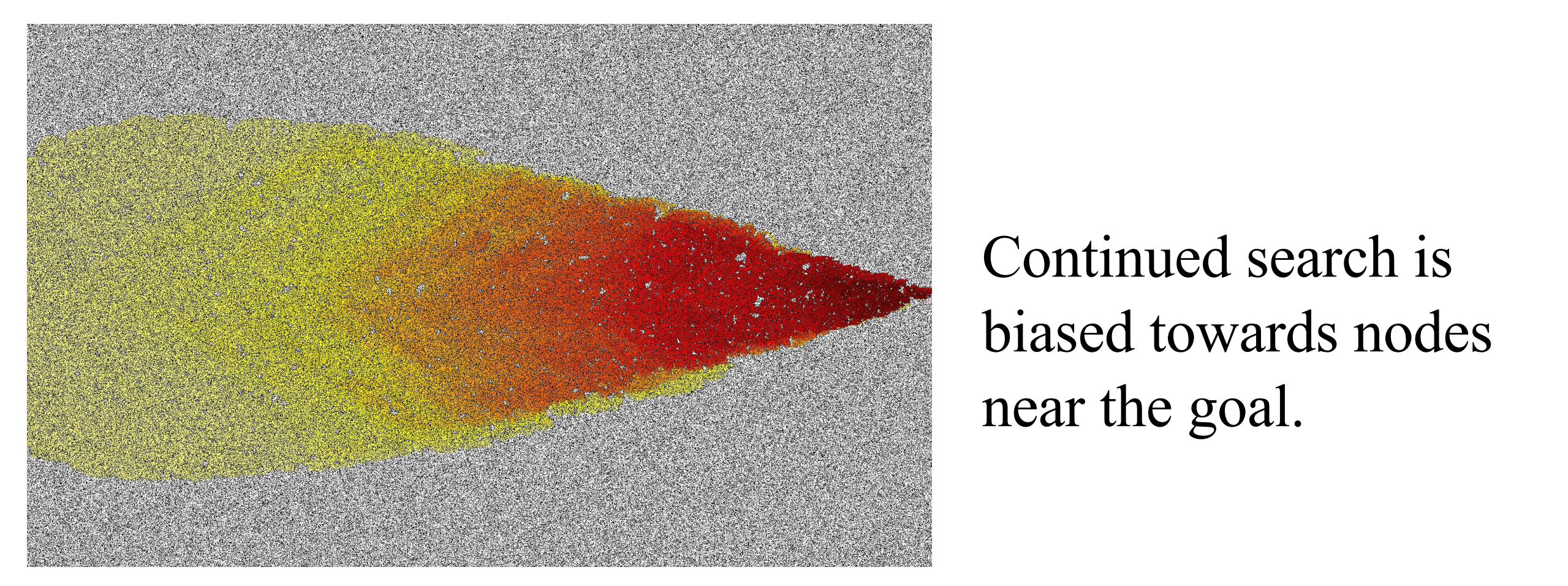
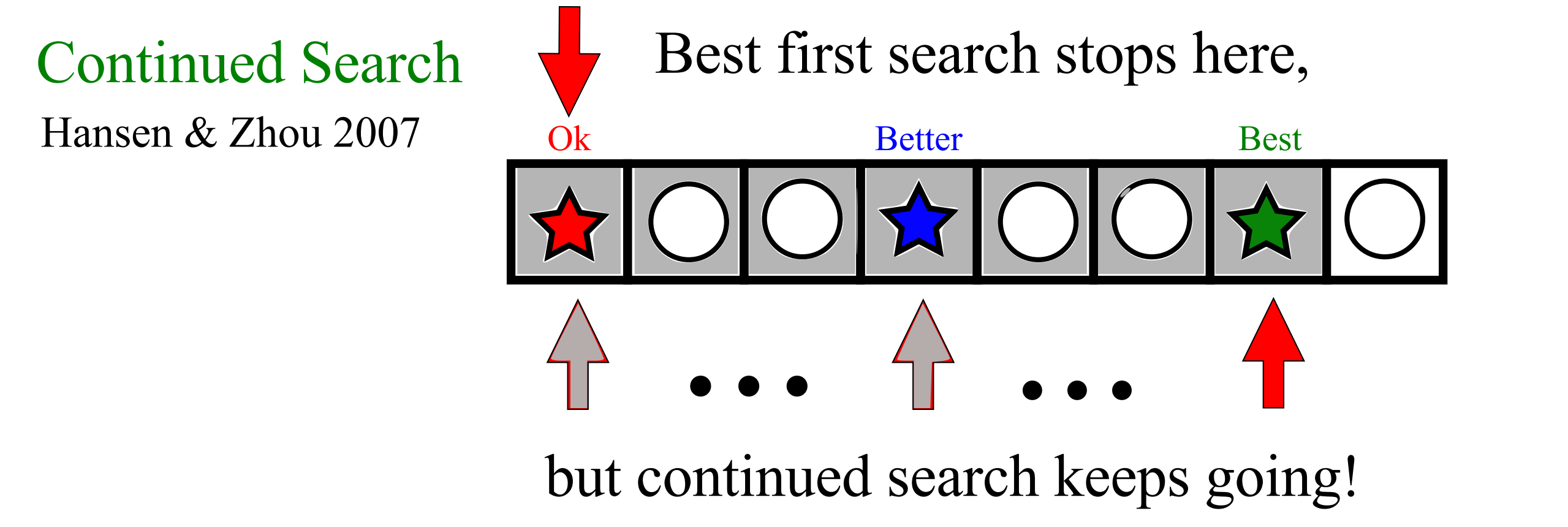


Anytime Heuristic Search: Frameworks and Algorithms

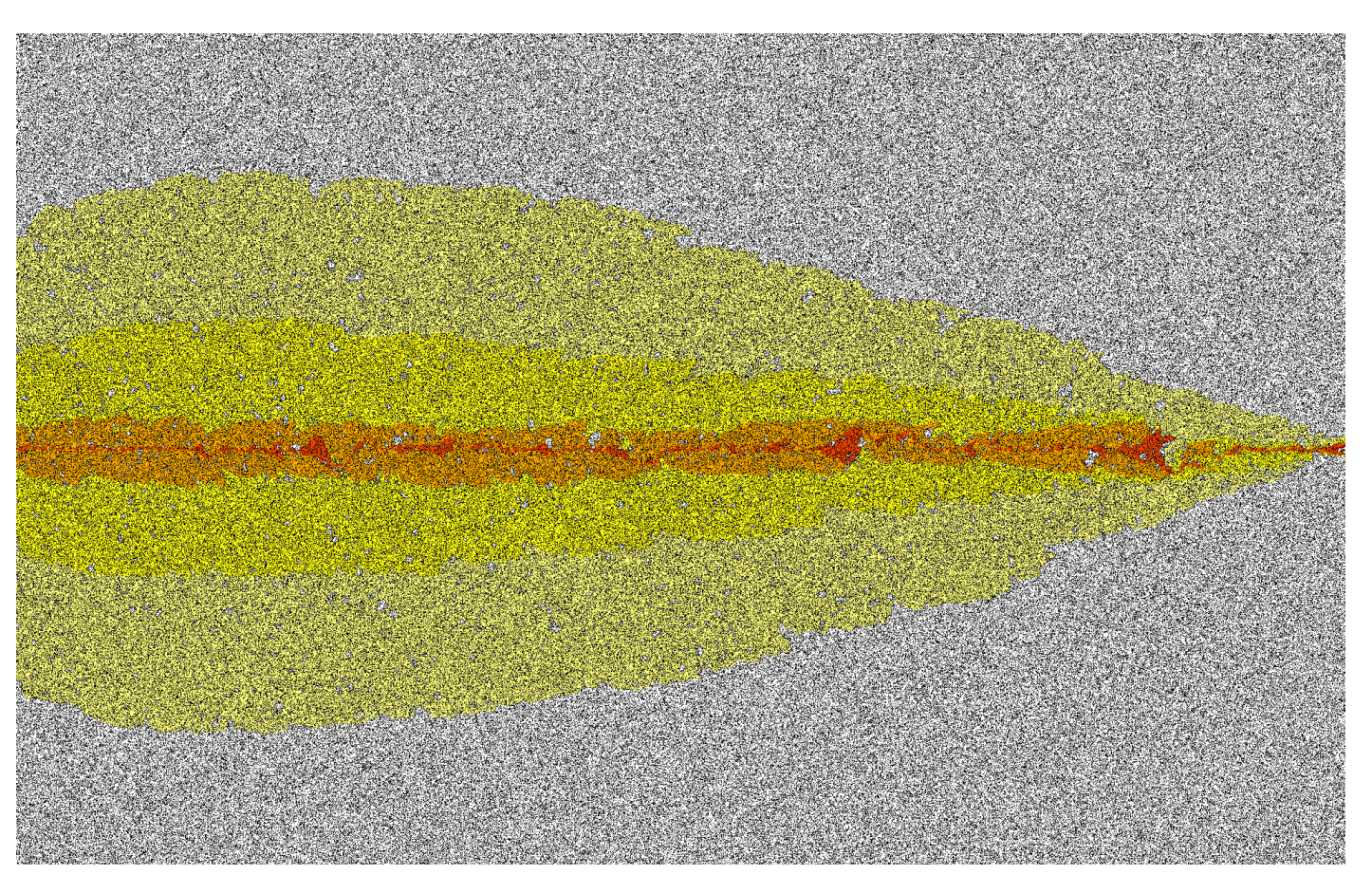
Which bounded suboptimal search performs best when converted to an anytime search?

Previously Proposed Frameworks



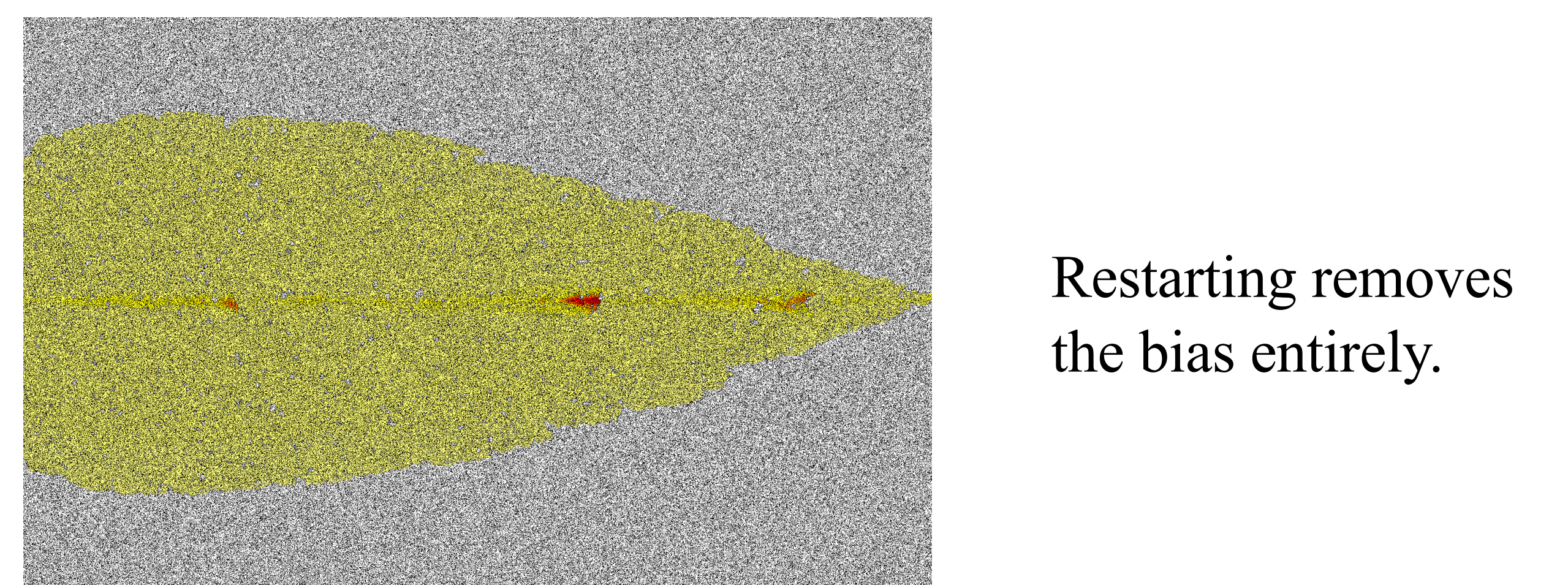
Repairing Search
Likhachev, Gordon, & Thrun 2003

We change the algorithm parameters between goals and resort.



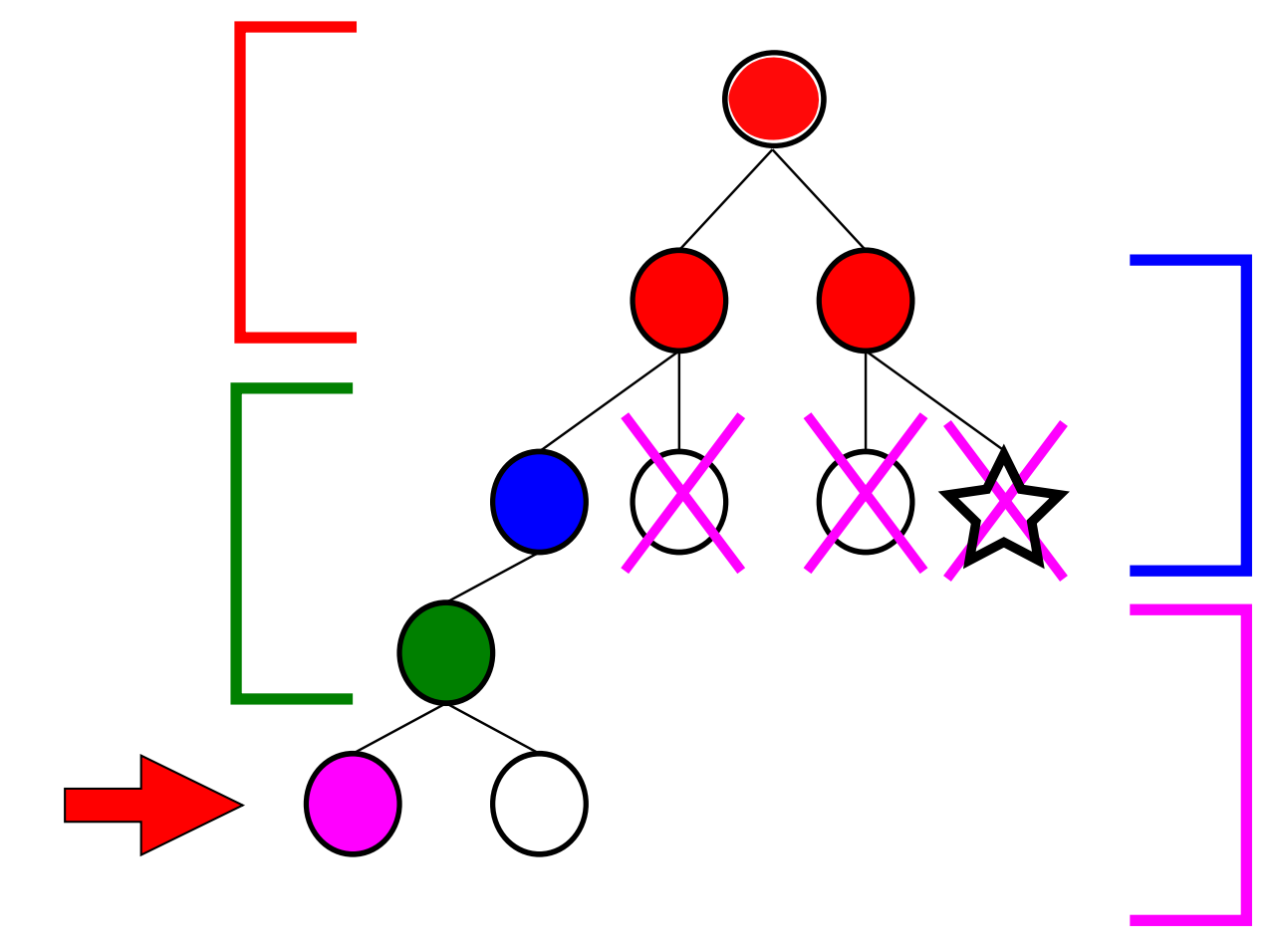
Restarting Search
Richter, Thayer, & Ruml 2010

Run a series of bounded suboptimal searches ending with an optimal search.

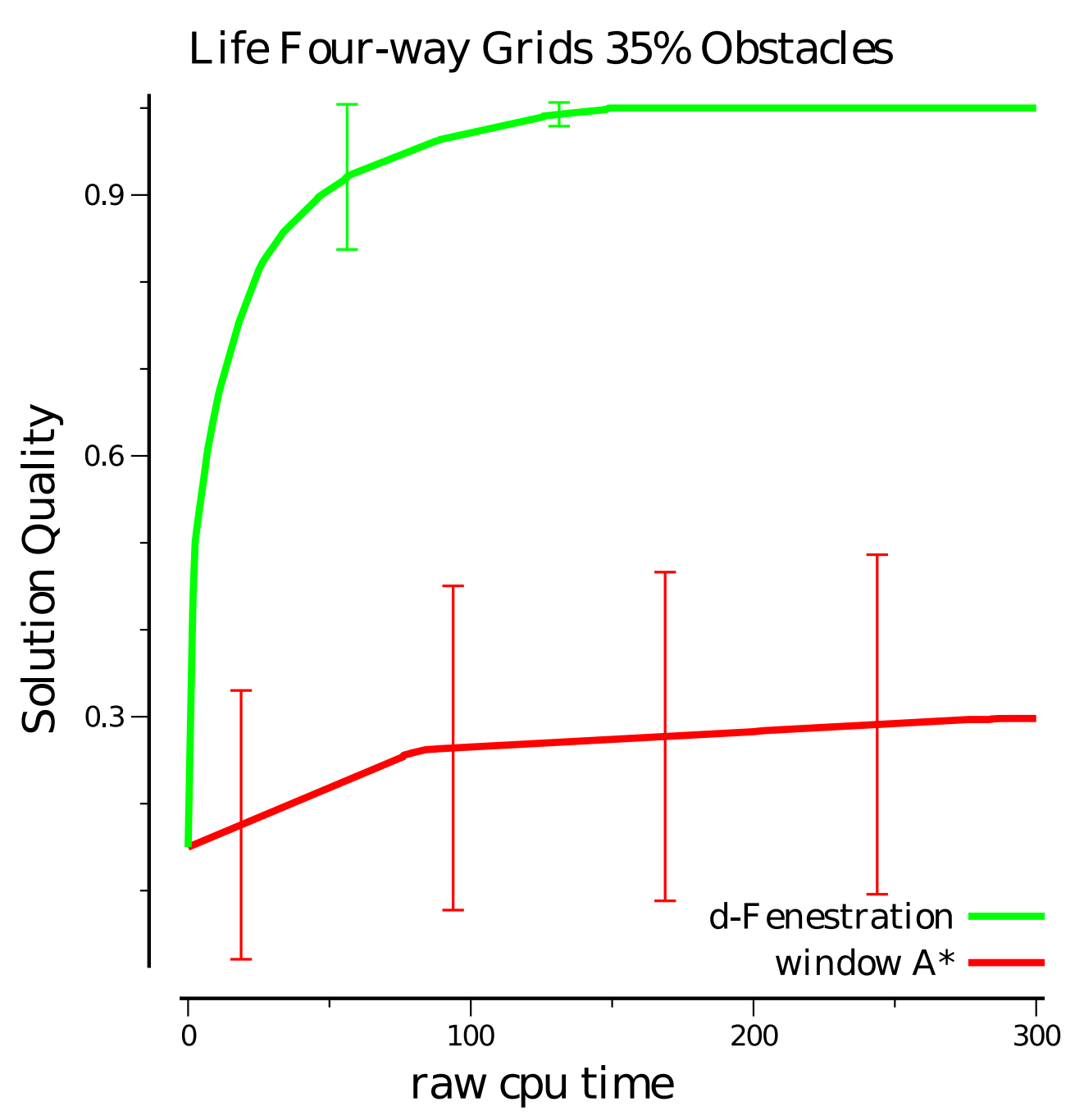


Contribution: Improved Anytime WindowA*

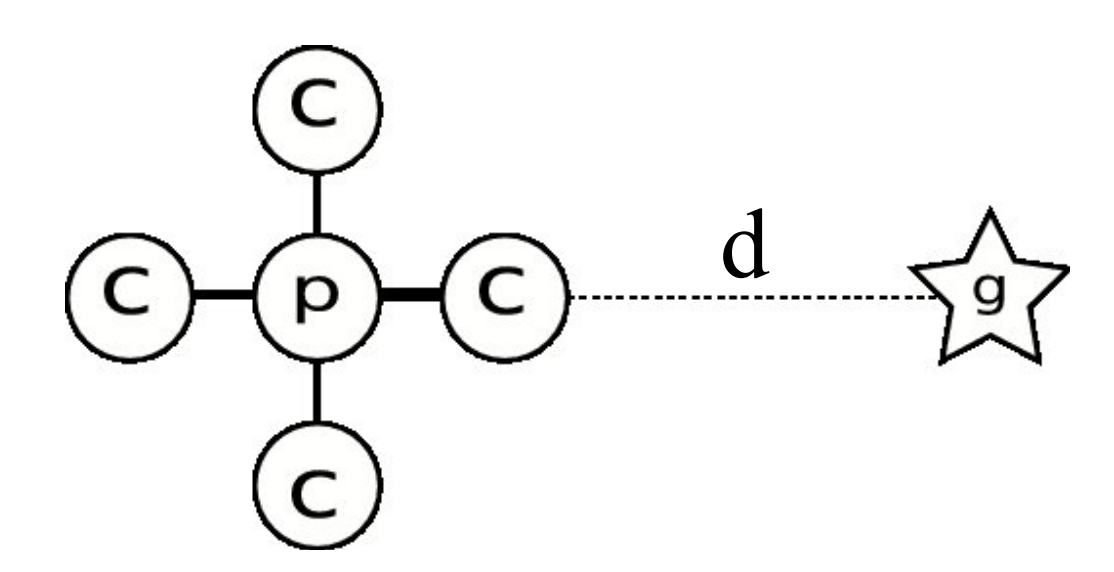
Window A* expands nodes within a fixed distance of the deepest node expanded. It assumes nodes at similar depths are the same distance from the goal and similarly informed.



d-Fenestration

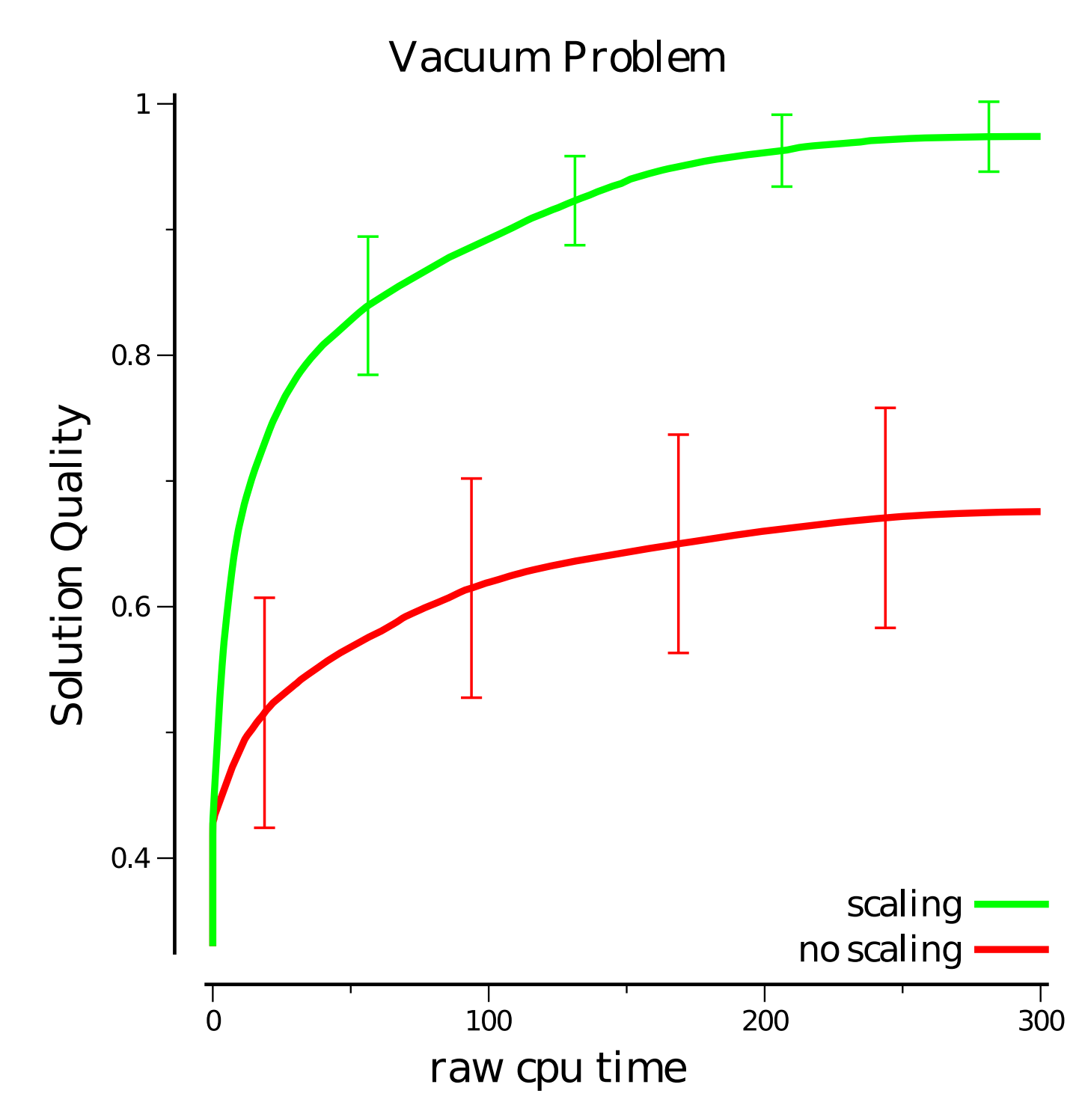


Not all nodes progress towards the goal at the same rate. We can estimate their distance using d.



Basing the windows on d rather than depth leads to large speedups!

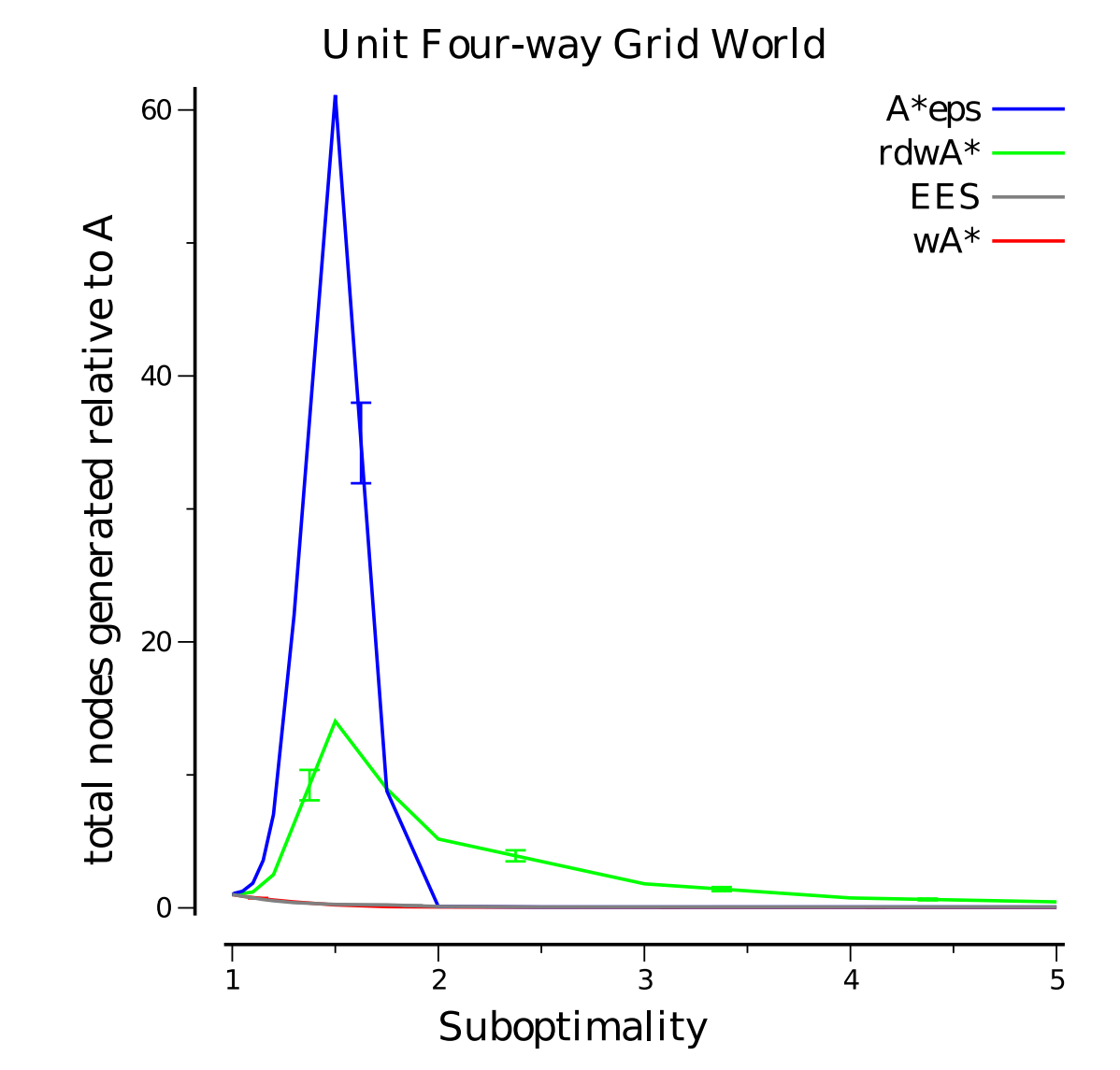
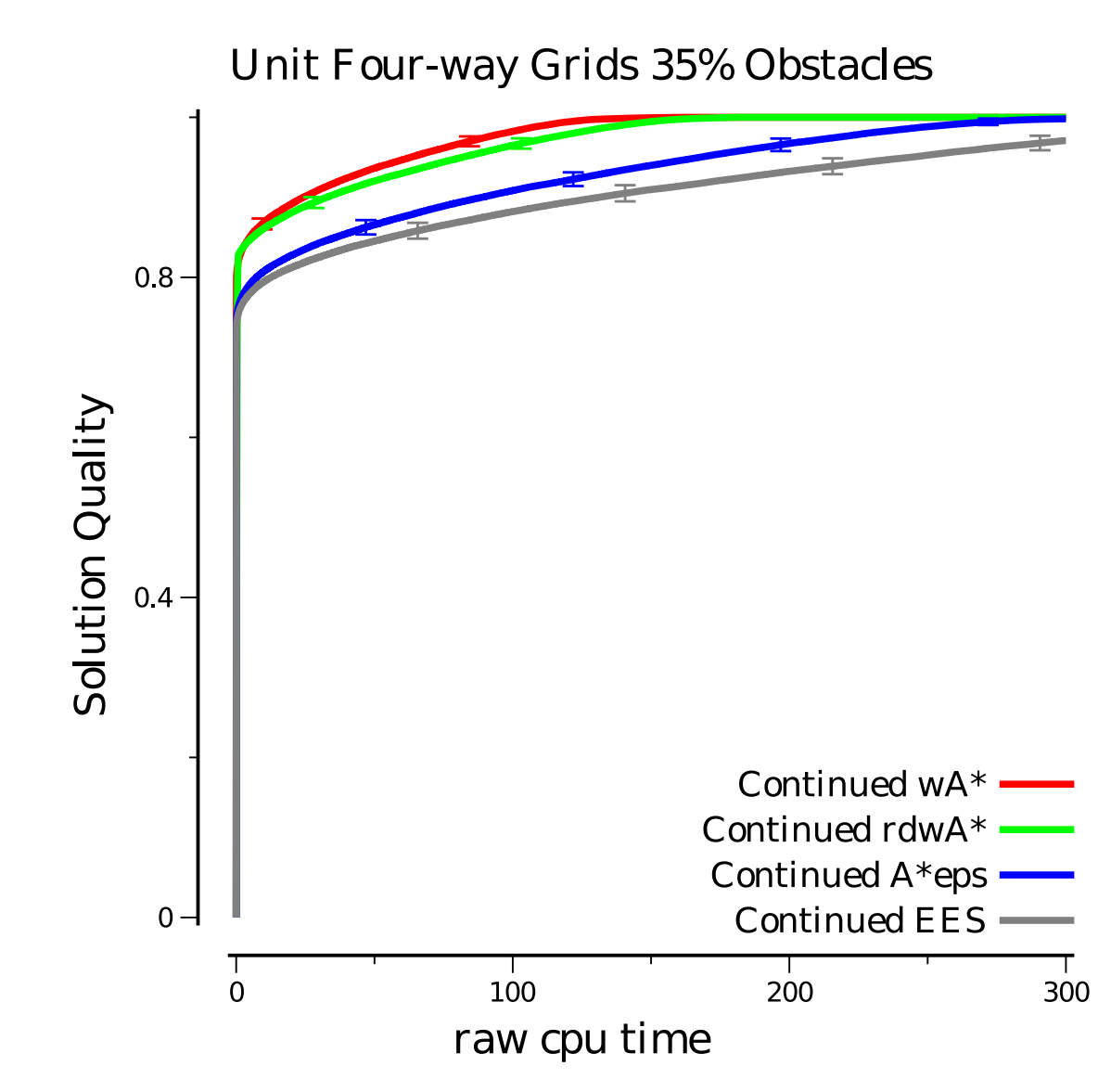
scaling window size



The window size of an iteration of anytime window A* greatly impacts performance. We grow the window quickly when no solution is found, and slowly otherwise.

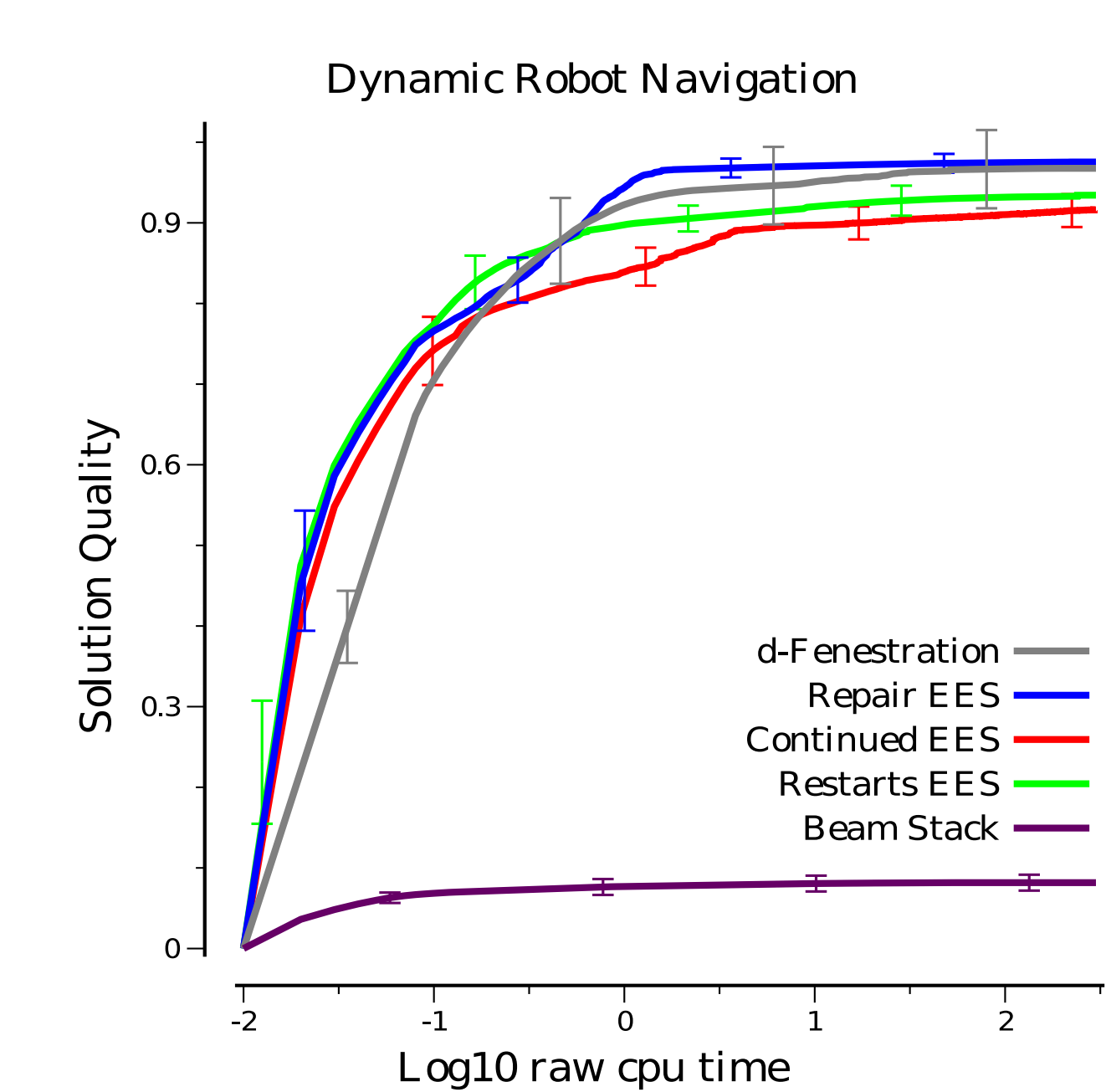
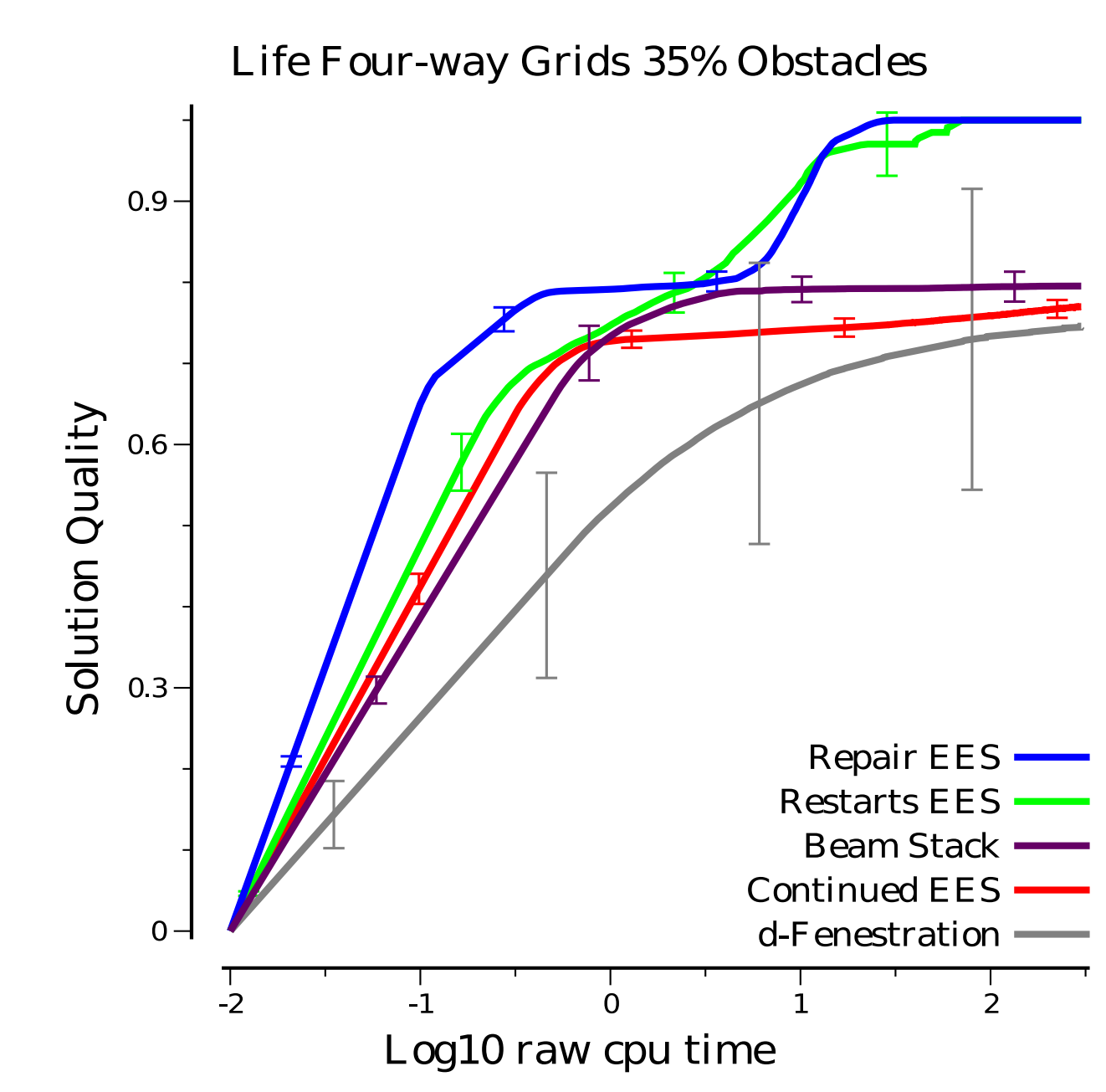
Main Contribution: Framework Is Key

There are enormous differences between the performance of bounded suboptimal search algorithms.



There is little difference when placed within anytime search frameworks.

Of all the tested algorithms, repairing and restarting anytime search consistently perform well. Other approaches have domains where they perform quite poorly.



Framework has a larger impact on performance than underlying algorithm.

Repairing search is frequently better than restarting search, and these techniques consistently outperform other approaches.