

Invited Speakers

Research Directions for Path Planning with A*

Sven Koenig (University of Southern California, USA)

In a couple of years, A* will celebrate its 50th birthday, yet research on it is still going strong. In this talk, I'll give an overview of various facets of research on path planning with A*. In particular, I will talk about approaches that my research group has contributed to, including versions of A* for agent-centered search (such as real-time search), incremental search, any-angle search and hierarchical search, focusing on recent developments and their connections to robotics and video games. For example, A* search with subgoal graphs speeds up path planning by utilizing a small amount of information compiled from map information in a pre-processing phase. It finds shortest paths by first finding high-level paths through subgoals and then shortest low-level paths between consecutive subgoals on the high-level path. The main insight here is that a large number of subgoals are only needed to connect certain start and goal locations to the other subgoals and can thus be ignored during path planning, resulting in a sparse graph without having to give up optimality.

Past, Present, Parallel: Heuristic Search in a Changing Computer World

Alexander Reinefeld (Zuse Institut and Humboldt Universität, Berlin, Germany)

Since its invention in 1968 by Hart, Nilsson and Raphael, many variants of the A* heuristic search algorithm have been devised. They are today found in many applications of practical relevance. Unfortunately, there exists a large class of very complex problems that can neither be solved with heuristic best-first search (because of limited memory space) nor with iterative depth-first search (due to limited computing time). How can we push the limitations a little bit further so that some of the previously intractable problems can be solved?

The answer to this question lies — in my view — in the efficient utilization of parallelism. Three things need to be done: (1) The space complexity must be reduced so that the solution space fits into the available distributed memories, (2) the scalability must be improved so that more processors can be employed to solve the problem in parallel, and (3) the efficacy of the heuristic must be improved so that a solution is found earlier. But above all, care must be taken to utilize all available parallel computational resources in a balanced way: CPUs, memories, and the network. With the example of the sliding-tile puzzle, we will show how previously intractable problem instances can be solved on large systems with thousands of CPUs and we will try to glimpse into the future of heuristic search on exascale computers with millions of processors.