Periphery Sweep Algorithm: Conquering A* Algorithm at Graph Traversal Solutions

Sen, Richik (School: Delhi Public School - Vasant Kunj)

Pathfinding is the process of using a computer application to find a route between two points which best meets certain criteria (shortest, fastest, etc.) in a large network. It is a crucial component of many intelligent agent applications, ranging from commercial computer games to robotics. Through this project, the Periphery Sweep algorithm is introduced, an algorithm that aims to provide faster graph searching than A* algorithm, the current industry standard for pathfinding applications. A* search is an informed search algorithm that uses a heuristic to speed up Dijkstra's algorithm for single-source shortest path. However, as the search space grows, A* slows down because of continuous insertion of nodes into its sorted Open List (the cost of insertion is logarithmic in the length of the list). IDA* is a space-efficient variant of A* (i.e. devoid of a sorted Open List), but it suffers from repeated states (the overhead of iterative deepening), cycles in the search space (the tradeoff for using no storage), reconstruction of search frontier in every iteration, and left-to-right traversal of the search tree. Inspired by the problem of eliminating the above inefficiencies in IDA*, Periphery Sweep algorithm adopts a modified iterative-deepening approach. On one hand, Periphery Sweep algorithm visits frontier nodes in the same order as IDA*. On the other hand, it expands them in the exact same order as A*. Experimental results indicate that Periphery Sweep runs roughly 40-60% faster than A* in our domain of graph-based pathfinding. This performance improvement increases with a growing search space.

Awards Won:
Association for Computing Machinery: Fourth Award of $500