CS261A Project Proposal Revision
Joseph D. Kulisics
2005 April 17

An analysis of basic, brute-force search algorithms reveals several implicit assumptions—BFS and DFS both assume that searching will start from the initial state of the search space only one time and commit to a single search direction, breadth-first or depth-first, respectively—and by discarding the assumption that search starts only one time, DFID appears as a generalization of depth-limited DFS. By discarding the second assumption that a search strategy commits to a search direction, we can develop new search strategies.

A survey of memory-limited searches in the area of heuristic search shows several strategies like MREC and MA* already follow the strategy of mixing search directions. MREC and MA* perform a breadth-first expansion of the search space followed by an IDA* search of the resulting frontier differing only in that MA* uses a strategy to replace some frontier nodes while MREC maintains a static frontier. Essentially both MREC and MA* use breadth-first expansion of the search tree only one time continuing the search exclusively by depth-first techniques.

DFID works well in almost all cases, but there are pathological trees on which DFID performs strictly worse than other searches. For example, DFID performs worse than BFS and DFS on trees with a branching factor of one or trees where there is only one node at each level of the graph. By committing to depth-first search techniques after generating a frontier, searches like MREC become vulnerable to pathologically evolving search spaces, and searches like MA* perform prohibitively expensive analyses to make their frontiers dynamic. Vulnerabilities in the practical, depth-first dominated searches suggest a benefit from reversing search directions.

Analysis of MREC and MA* as well as the basic, brute-force searches suggests two additional implicit assumptions. MREC and MA*, as well as other searches such as IDA_CR, assume that mixed searching will occur in two, fixed phases, and all searches mentioned so far except IDA_CR assume that searching will terminate on discovery of a solution, which must be optimal. By discarding these two assumptions, another search might be developed that does not succumb to performance problems in pathological cases. Specifically, by alternating depth-first construction of frontiers with breadth-first expansion of the search space, we might construct a search that performs strictly better than DFID or depth-first dominated, two-phase searches in pathological cases.

A search that begins by depth-limited construction of a frontier followed by breadth-first expansion of the nodes at the frontier and iteratively extends the depth-limited frontier cannot guarantee that the first solution discovered is optimal, but the search strategy does not have to sacrifice optimality or completeness. Such a search can continue examining the search space past the discovery of the first solution to guarantee the discovery of an optimal solution. Though we can guarantee discovery of an optimal solution, we might regard the ability to discover a suboptimal solution as a strength of the algorithm; a user of a search algorithm might prefer the possibility of faster discovery of a suboptimal solution to discovery of an optimal solution in situations where solution quality is unimportant.

In detail, my proposed progressive broadening search will do breadth-first search of a problem space tree to a fixed depth-limit, n. If a solution is found in this phase, the solution will be optimal because BFS produces optimal solutions. If no solution is found, the data from the breadth-first search will be discarded, and the search tree will be depth-first expanded to depth n defining a frontier of nodes. At each node, a breadth-first expansion of the tree will occur to a local depth of n. If a solution is found, the search will continue from the candidate solution across the frontier nodes to identify any better solutions. The continued search across the frontier will be bounded by the location of the candidate solution. If no solution is found, the search tree will be depth-first expanded to a frontier at depth $2n$, and a breath-first search will be conducted from each node in the frontier to a local depth of n. The process will continue as in the last iteration and through successive iterations until a solution is found.

My progressive broadening strategy is completely different from the Ginsberg and Harvey iterative broadening strategy. In the iterative broadening strategy, a *breadth limit* is used to limit the width of the search tree at each level in a depth-limited search. The breadth-limit is increased incrementally until a solution is found, and if desired, the search is continued past the discovery of the first solution to guarantee that an optimal solution is found. Iterative broadening is not complete on infinite trees because it requires that a depth-limit be observed to increment broadening.

My claim is that an efficient implementation of the strategy of progressively broadening the search tree has acceptably good asymptotic space and time complexity, which is complicated and depends on the size of the depth-first steps and the depth of the progressive broadening, compared to the practical, memory-limited searches like MREC and strictly better performance on pathological search spaces with effective branching factors of one than all searches mentioned other than BFS and DFS, and progressive broadening has the same asyptotic space and time complexity on pathological cases with effective branching factors of one as BFS and DFS. I intend to study the behavior of progressive broadening analytically to verify its space and time complexity, and I intend to use generated instances of the sliding-tile puzzle, which according to the course reader is a standard problem for empirical comparison of search algorithms, to collect empirical data on the behavior of progressive broadening with heuristics and compare the performance to data collected on an efficient implementation of IDA*. We should regard as a success data that places the performance of progressive broadening on random puzzle instances within one order of magnitude of IDA* in discovery of an optimal solution. I intend to demonstrate empirically with constructed, pathological examples that progressive broadening performs better than IDA*. I fully expect to complete a brute-force implementation of progressive broadening and a reference implementation of IDA* for comparison. I would like to incorporate a heuristic into the progressive broadening search for comparison with IDA*, but if I cannot incorporate any heuristics due to time constraints, I intend to replace the heuristic of IDA* with a constant function causing IDA* to default to DFID, which will allow for fair comparison with progressive broadening without heuristic enhancement. If I cannot complete heuristic enhancement of progressive broadening, I shall use smaller puzzle instances to allow data to be collected in a reasonable amount of time.

From discussions with other students, I knew that this course involved a substantial project requiring original ideas for search, but I am solely responsible for the idea of progressive broadening search. I had thought in passing about novel search strategies before this quarter, but I have never seen the course reader or other course materials before this quarter and have not done any substantial work with this idea. This proposal represents the first work that I have done in relation to progressive broadening search, and the idea is not related to my own research or other coursework.

Joseph D. Kulisics