PERFORMANCE OF A* AND IDA* - A WORST CASE ANALYSIS

A. Mahanti  A. Pal  S. Ghosh and L.N. Kana1 and D.S. Nau*
University of Maryland  IIM  University of Maryland
College Park  D. H. Road  College Park
MD 20742  Calcutta 700027  MD 20742

This paper presents a detailed comparison between Algorithms A* and IDA*. A* [2] is a best first search algorithm which at each node n in the graph uses a node evaluation function \( f(n) = g(n) + h(n) \), where \( g(n) \) is the cost of the currently known best path from the start node \( s \) to \( n \) and \( h(n) \) is an estimate of \( h^*(n) \) (cost of the minimum cost path from \( n \) to a goal node). At each iteration, A* selects a node with minimum \( f \)-value for expansion. A* is known to be optimal in terms of number of node expansions, however the storage requirement of A* is very high - it is exponential in the depth of the solution found.

In a recent study Korf [1] presented a new algorithm namely IDA*. IDA* unfolds a graph into a tree. In every iteration, IDA* starts the search from the start node and makes a depth first search within the current threshold. Initially threshold is set to \( h(s) \). Threshold for the next iteration is calculated as the minimum \( f \)-value of the discarded nodes of the previous iteration. Since each iteration is a depth first search, IDA* requires only \( O(L) \) memory, where \( L \) is the maximum number of nodes on any minimum cost solution path, assuming that the branching factor of a node is constant. Thus, given sufficient execution time IDA* can solve relatively harder combinatorial optimisation problems.

It has been shown in [1] that IDA* is asymptotically optimal amongst all admissible best first search algorithms for tree searches. However, the optimality proof of IDA* assumes a very stringent condition that the number of new nodes grow exponentially from threshold to threshold. In this paper we present a necessary and sufficient condition for the \( O(N) \) time complexity of IDA*, where \( N \) is the total number of nodes having \( f \)-value \( \leq h^*(s) \). We show that IDA* can have \( O(N) \) time complexity under a more general condition than that was originally felt. Moreover, we illustrate through examples, that the different conditions [1] imposed in the analysis of IDA* are neither sufficient nor necessary. We also show that the worst case time complexity of IDA* can become \( O(N^2) \) for tree searches even in presence of all conditions stated in [1].

When the heuristic is monotone, A* can handle a graph like a tree and it never expands a node more than once [2]. But, for graphs IDA* can not prevent the reexpansion of a node through a costlier path. Thus, in practice, for graph search problems the performance of IDA* can become worse than A*. We show that for acyclic graphs the worst case time complexity of IDA* is \( O(2N) \). In the last iteration of IDA* itself there can be \( O(2N) \) node expansions. Total number of node expansions can increase in presence of cycles. These \( O(N^2) \) and \( O(2N) \) upper bounds for trees and acyclic graphs hold even under the assumptions such as monotone heuristic, constant relative error, heuristic branching factor \( > 1 \), and identical tie resolutions with A* [1]. Finally we show that there can not exist any limited-memory best first search algorithm which, in general, will have the linear time complexity as \( A^* \) in trees, and in graphs with monotone heuristic. We conjecture, IDA* is asymptotically optimal amongst all limited-memory best first search algorithms for trees.

References