Comparisons of Search Algorithms

Joowon Kim

February 7, 2018

1 Summary

Best-first search see whether the node is at the most efficient way and visit the node preferentially. Weighted A* is efficiently bounded search and have is $f'(n) = g(n) + w \times h(n)$. When solution is found, the cost is in w of the optimal cost. Greedy best-first search ignores g so that it would be $f(n) = h(n)$. [4] A*e finds high quality solution than some other algorithms. Best first search is good for finding solutions like 48-puzzles when the domain is small since it will fail when the domain is large because of memory and time issues.

Hill-climbing [3] algorithm requires less work to get solutions by expanding the best child node. Enforced Hill-climbing is used with breadth-first search to make random behaviors. But it can be costly and fails many times. This is efficient for vacuum robot problem [5] compare to LSS-LRTA*. LSS-LRTA* uses Dijkstra and A*. In most cases, LSS-LRTA* [2] is better compare to the enforced hill-climbing. Hill-climbing algorithm is not good for dead-ends.

Beam search can be divided into two. First, best-first beam search is similar with best-first but useless nodes are pruned from open list. It is comparably inefficient than breadth-first beam search. Breadth-first beam search also works similar with breadth-first but only certain number of nodes are expanded. [1] Beam search is good when the list is closed and when expanding same nodes. Beam search is not good algorithm that has the dead-ends as well as hill-climbing algorithm.

2 Comments

This paper explains each algorithm well by comparing different types of algorithms. It also shows the advantage and disadvantage of each search with some examples so that it was easy to understand what author was trying to say. The use of comparison tables for each examples and graphs were properly used so in case readers might not fully understand, that can be helpful to understand which algorithms are better options for certain examples and which algorithms are more efficient compare to others. One thing is that it would be better if author gave more general explanation about how to use algorithms with some pseudo codes. It was a bit messy to just understand various kinds of algorithms by literally reading the explanation so that if there were pseudo codes and explanation about them that readers can understand more easily by looking at the codes examples line by line.

Overall, I learned lots of kinds of algorithms by reading this paper. In the class, we talked about the A* algorithm but beam search was new to me and hill-climbing was also not really familiar but from this paper, I learned by looking at the tables and some examples. Since we learned about the A* algorithm, it was interesting for me to get more information about A* algorithm. I thought A* algorithm was just one kind but it has many sub part in A* algorithm and the usage, efficiency and examples were varied. Hill-climbing algorithm was used along with breadth-first search and it recalled me last semester algorithm class in this university that when I was implementing Johnson's algorithm, I had to use Dijkstra algorithm and expanded and it was similar with the hill-climbing since it also uses the breadth-first together.

References

[1] R Bisiani. *Encyclopedia of Artificial Intelligence*, volume 2, chapter Beam Search, pages 1467–1468. John Wiley and Sons, 2 edition.

[2] S. Koenig and Sun. X. Comparing real-time and incremental heuristic search for real-time situated agents. *Journal of Autonomous Agents and Multi-Agents Systems*, 18(3):313–341, 2008.

[3] J. Pearl and J.H Kim. Studies in semi-admissible heuristics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1982.

[4] Thomas A. FeoMauricio G. C. Resende. Greedy randomized adaptive search procedures. Technical report, Kluwer Academic Publishers, 1995.

[5] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.*