IEEE --- 2005 International Conference on Emerging Technologies
September 17-18, Islamabad

Semantic Based Classification of Search Enhancements

Syed Jamal Hussain
Department of Computer Science
Mohammad Ali Jinnah University
Karachi, Pakistan
sjhussain@jinnah.edu

Faraz Ahmed Zaidi
Department of Computer Science
Mohammad Ali Jinnah University
Karachi, Pakistan
farazazaidi@yahoo.com

Abstract

AI research has developed numerous methods to solve state space problems. During the recent times, one such strategy, Search Enhancements has performed a pivotal role in solving complex real world problems. Many different properties and taxonomies for these search enhancements appear in the literature. This work presents a new parameter for the classification of search enhancements with the intent to add a new dimension to the process of creating new enhancements as well as to develop a better understanding. This classification is based on the semantics of the state space graph (or tree) generated and the problem domain. It is shown that semantics of a problem domain has been a vital aspect of the search enhancements. One semantic based search enhancement, the False-Move is described in this paper. This search enhancement in conjunction with the A* algorithm is used to solve the 8-puzzle problem and the results are presented.

Keywords

Search algorithms, Search enhancements, Semantic search enhancements and False-Move.

1 Introduction

For problems with complex state space like the two-player games, Go [3] and Shogi [4], or puzzles like Sokoban [5], approaches based on heuristic search alone are not enough to find a solution. These approaches need to be complimented by search enhancements in order to reduce the search space and improve the efficiency of the search process. Most introductory texts on artificial intelligence start off explaining the basic search algorithms and possibly their fundamental differences and that is where most AI books stop their technical discussion. In contrast, in real world AI applications, it is the next step, the search enhancements (or algorithm enhancements, both names are used interchangeably in the literature); that is the topic of interest, not so much the basic algorithm [4]. This is because the selection of an algorithm is based on well-defined rules as compared to the selection of search enhancements. The choice of search enhancements can have a dramatic effect on the efficiency of the search; often several orders of magnitude per search enhancement [4]. Therefore it is important to have a better understanding of the search enhancements to assist in the process of selecting search enhancements for a specific problem. This is an active area of research [4] and noticeable progress is yet to be made in this dimension. The understanding developed through different parameters can also play a fundamental role in the creation of new and efficient search enhancements, which result in better and improved search enhancements.

This paper is organized as follows. Section 2 presents a brief account of the taxonomies of search enhancements. Section 3 introduces the idea of semantics in search enhancements. Section 4 presents a new semantic search enhancement, the False-Move, and its application to the A* search algorithm. Section 5 shows experimental results in comparison to the breadth-first search, the A* search and the A*-False-Move algorithm. Final conclusions are drawn in sections 6, along with the potential possibilities for future work.

2 Search Enhancements

The success of a search process depends on the ability of the algorithm to visit most of the relevant parts of search space. Search enhancements play an important role in reducing the search space and focusing the search process. A search enhancement essentially performs one of the following tasks:

- ♣ Iterative Modification of lower Bounds
- ♣ Removal of Nodes
- ♣ Changing Order of Nodes for Expansion
- ♣ Collapsing the several nodes into one.
- ♣ Performing some Preprocessing

Out of the several classification parameters of search enhancements, one is on the basis of the generality of the knowledge used to create the search enhancement [1].

The classes are domain independent, domain dependent, instance dependent and sub-tree dependent (Another class belonging to the same taxonomy is partially domain dependent which is the hybrid of domain independent and domain dependent classes).

Another way to classify the enhancements is on the basis of the how the knowledge is obtained to create the search enhancement [1] i.e. static (such as advice from a human expert) or dynamic (gleaned from a search). (A hybrid of the two is known as semi-static search enhancement).

A search enhancement can be admissible (preserve optimality in a solution) or non-admissible [1]. Non-admissible knowledge can either preserve completeness of the algorithm or render it incomplete. Admissible knowledge is necessarily complete [6].

All the above mentioned classification help to better understand the search enhancements. This understanding can help in the development of new search enhancements.

3 Semantic Search Enhancements

One important aspect of the heuristic search algorithms is the heuristic function used in the algorithms. The semantics of the problem domain play an important part in developing better heuristic functions to solve a problem. More information helps to develop more accurate heuristics which in turn, results in an efficient algorithm [6]. The same idea can be carried over to the development of search enhancements.

Semantics plays an important role in the development of better search enhancements. The idea suggests the explicit application of domain dependent and implicit knowledge for the creation of enhancements. In essence, human players apply intelligence in a similar way, i.e. utilising the game semantics, game history along with the heuristic knowledge to cut down the possible set of moves to a reasonable bound. This makes the search for a reasonably good move easier and manageable. For the semantic based and non-semantic based search enhancements, an example each is described below.

It may be possible to reach the same state via two different paths, Transposition table search enhancement uses a table to keep track of the states already visited (usually implemented as a hash table). This reduces the size of the search space by several orders of magnitude, avoiding the repeated states. This search enhancement is an example of non-semantic search enhancement. There isn’t any requirement for the association of semantics for this search enhancement. As the only information stored is the state of the game and repeated states are not expanded further.

There are situations in a game, where if a certain move is made, a player ends up in making a series of moves to get out of that situation. For example, a tunnel is identified as a board situation in sokoban [1] where once a stone is pushed into, there is no way the player can make any other move due to walls on either side of the man pushing the stone. The basic idea behind the Tunnel Macros search enhancement is that, rather to perform this thorough search, pre-processing can identify these board situations and the sequence of moves can be collapsed into a single move. By collapsing several moves into one, the height of the search tree is reduced. This search enhancement is made possible only because of the association of the semantics of the game and the search space generated by the problem domain. Thus this search enhancement is an example of semantic search enhancement.

4 False-Move Search Enhancement

There are situations in games when there is a desire to play a specific move that could lead to a solution quickly, but we are constrained by the legality of the move. An example would be helpful to better comprehend the idea, consider the state of an 8-Puzzle problem in Figure 1.

1	2	3
6	7	4
	8	5

Figure 1: Initial state of an 8-Puzzle problem

The ideal move would be to move the empty tile across diagonal to the centre location; but this move is not allowed, since diagonal moves or cross-tile moves are not possible in a traditional 8-Puzzle problem. This move can be made possible with a little imagination, though. If Empty is to be moved in place of tile 7, this could be made possible by swapping the positions of tile 6 and 8. In effect, this would be equal to the sequence of legal moves: move the empty tile up, right, down, left, up, right. Diagrammatic representation of the intermediate states that are generated is provided in Figure 2.

1	2	3
6	7	4
	8	5

(a)

1	2	3
	7	4
6	8	5

(b)

1	2	3
7		4
6	8	5

(c)

1	2	3
7	8	4
6		5

(d)

1	2	3
7	8	4
	6	5

(e)

1	2	3
	8	4
7	6	5

(f)

1	2	3
8		4
7	6	5

(g)

Figure 2: Intermediate states of the False-Move allowing diagonal move

This shows that a False-Move to allow diagonal tile move is, in effect, collapsing the tree height of six levels. This helps in fast convergence towards the goal, as it does not apply all the operators, and generates all other nodes to achieve this virtually single move. Since the move such generated is not allowed directly in the game, hence the name False-Move fits and the algorithm uses this new set of moves containing both the existing legal moves as well as the False-Moves. This, in turn, leads to the definition of the term False-Operator to refer to the operator used to generate this False-Move.

5 Experimental Results

Table 1 summarises the experimental results of solving several randomly generated 8-Puzzle problems using breadth-first search, the A* search and the A*-False-Move search algorithms (described as an example in the previous section). The Manhattan distance heuristic is used to solve the problem for both the A* search and the A*- False-Move search. The initial state is defined as a string of numbers stating the tile position from left to right and top to bottom, where 'E' (Empty) represents the empty tile. The goal state is represented by 1238E4765. The numbers in each column against the algorithm name represents the 'Number of Nodes' expanded to reach the goal state from the given initial state.

Table 1: Experimental results of solving randomly generated 8-Puzzle problems using various algorithms

Case	Initial State	Breadth-First Search	A* Search	A*-False-Move Search
1	13E826754	81	12	27
2	12345687E	213	16	26
3	153E82764	7036	332	356
4	6E3184275	2291	34	26
5	713E28654	328	314	44
6	4231E8765	861	27	473

The results above show that the False-Move enhancement does not always improve the solution of the A* algorithm, and subject to a particular problem, it may worsen the solution to a large extent (for example, case 6). But in most of the cases, the behavior is similar to the A* algorithm (for example, case 1, 2, 3 and 4). And in some cases, this improves the solution to a large extent (for example, case 5).

This study has been conducted on only the earlier described instance of the False-Move, the Cross-Tile move. Other False-Moves can be generated using the same principle, not only for the 8-Puzzle problem but also for more complex problems like the 15-Puzzle problem and Sokoban.

6 Conclusions and Future Work

This research paper presents a new parameter to classify search enhancements on the basis of semantics of the state space and the problem domain. The classification is useful in developing a better understanding of existing search enhancements as well as adds a new dimension for the development of new search enhancements. The semantic based search enhancement presented in this study is the cross tile false move, many similar false moves can be generated for a variety of problem domains and prove to be a useful technique to create other semantic based search enhancements.

References

[1] Andreas Junghanns and Jonathan Schaeffer. Domain Dependent Single Agent Search Enhancements. IJCAI-99, pages 570-575, 1999.

[2] Jonathan Schaeffer. Search Ideas in Chinook: Games in AI Research (eds. H. Jaap van den Herik and H. Iida). Universiteit of Maastricht, The Netherlands. 2000.

[3] Erik C. D. van der Werf, H Jaap van den Herik and Jos W. H. M. Uiterwijk. Solving Go on Small Boards. ICGA Journal, 26(2):92-107, 2003. Available: http://www.cs.unimaas.nl/~vanderwerf/pubdown/solving_go_on_small_boards.ps.gz.

[4] Jonathan Schaeffer and Ask Plaat. Unifying Single Agent and Two-Player Search. Journal of Information Sciences, 134(3-4):151-175, 2001.

[5] Joseph Culberson. Sokoban is PSPACE-Complete. Technical Report TR97-02, Department of Computing Science, University of Alberta, 1997.

[6] Stuart J. Russell and Peter Norvig. Artificial Intelligence, A Modern Approach. Prentice Hall. 2002.