# Best First Search in And/Or Graphs

P.P. Chakrabarti
S. Ghose
S.C. DeSarkar

Department of Computer Science & Engg.
Indian Institute of Technology
Kharagpur 721302, India

## 1. Introduction

An analysis of the informed best first search strategy has been carried out for AND/OR graphs. The heuristic cost function F has been decomposed as F = G+H, which is a generalization of f = g+h for ordinary graphs [10, 2, 4]. The idea of minimum of pathmax of F has been used to show that most of the properties, which hold for path-finding algorithms as shown in [2,4], also hold for AND/OR graphs. Compared to those found in literature, more relaxed conditions for admissibility and consistency for AND/OR graphs have been established. The best-first search strategy has been analysed under admissible and monotone restrictions. Questions of optimality in terms of node expansions have been studied. The use of weighted cost functions is analysed. Non-additive cost measures have also been discussed.

 

## 2. Preliminary Definitions and Algorithm BFS

The sink nodes of any AND/OR graph are called tip nodes. If the tip nodes are primitive subproblems which can be solved directly, they are called terminal nodes. If the tip nodes are unsolvable they called non terminal nodes. The other nodes are of two types: OR and AND. Since the complete AND/OR graph, G (referred to as the implicit graph) has a large number of nodes, the search algorithm works on an explicit graph, G', which initially consists of the start node s. When a tip node of G' is expanded, its successors are added to G'.

A potential solution graph(psg) D' is a finite subgraph of G' as defined below: (i) s is in D', (ii) if n is an OR node in G' and n is in D' then exactly one of its successors in G' is in D', (iii) if n is an AND node in G' and n is in D' then all its immediate successors in G' are in D', (iv) every maximal (directed) path in D' ends in a tip node of G'.

A solution graph D is a finite subgraph of the implicit graph G which is defined below:

(i) s isin D, (ii) if n is an OR node in G and n is in D, then exactly one of its immediate successors in G is in D, (iii) if n is an AND node in G and n is in D, then all its immediate successors in G are in D, (iv) every maximal (directed) path in D ends in a terminal node.

A partial solution(ps) P of a solution graph D is a subgraph of D as defined below: (i) s is in P, (ii) if for any node n in P, any of its immediate successors in D is in P, then all these successors are in P.

A psg D' is said to be solved if it is a solution graph. A psg is said to be unsolvable if it contains a non-terminal node as a tip node.

The best-first search algorithm, BFS, is given below:

Algorithm BFS

1. Initially the explicit graph, G', consists only of the start node, s.
2. From G' constructed so far select the most promising psg D', which is not unsolvable. If all psg's are unsolvable, exit with failure.
3. If D' is solved, exit with success. Otherwise expand a node n in D' generating all its successors. Add them to G'.
4. Goto 2.

In pathfinding problems, algorithms like $A^*$[10], $BF^*$[4], A[2] fall in this category. For AND/OR graph $AO^*$[10] is of this type.

We now introduce the idea of cost. Initially we assume that the costs are additive and the graphs are additive AND/OR graphs like those of [6, 7]. Each arc from nodes m to n has a finite positive cost C(m,n). Each node n in G has a heuristic estimate h(n). The actual minimal cost is $h^*(n)$. For terminal nodes $h(n) = h^*(n) = 0$. For non terminal nodes $h(n) = h^*(n) = \infty$.

Let R be a psg or ps. Corresponding to every node n in R we define

$$G(R,n)=\begin{cases}0 \text{ if n is a tip node of R}\\ \sum_j \{G(R,n_j) + C(n,n_j)\} \text{ for non tip nodes with immediate successors } n_j.\end{cases}$$

$$H(R,n)=\begin{cases}h(n) \text{ if n is a tip node of R}\\ \sum_j H(R,n_j) \text{ for non tip nodes with immediate successors } n_j.\end{cases}$$

$$H^*(R,n)=\begin{cases}h^*(n) \text{ if n is a tip node of R}\\ \sum_j H^*(R,n_j) \text{ for non tip nodes with immediate successors } n_j.\end{cases}$$

$G(R) = G(R,s)$. $H(R) = H(R,s)$.
$H^*(R) = H^*(R,s)$. $F(R) = G(R) + H(R)$.

For any psg D', BFS selects the psg D' with minimum F(D') as the most promising psg. Observe that $h^*(s)$ is the minimal cost solution of G.

## 3. Conditions for Obtaining Minimal Cost Solutions

We analyse BFS in the light of the general theory of heuristic search[2, 4, 11]. Definitions: (Generalization of Bagchi and Mahanti [2])

(i) Let $D_1$, $D_2$, ... be solution graphs of G. We write $P \in D_i$ if P is a ps of $D_i$. For each i, let pathmax $M_i = \max_{P \in D_i} F(P)$.

(ii) Let $Q = \min_{i>=1} M_i$

(iii) Let $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$, for some $k>=1$, be minimal cost solution graphs of G.

Let $Q_{opt} = \min_{1<=j<=k} M_{i_j}$

Lemma 1: At any step before BFS terminates, G' contains at least one ps from every solution graph as a psg.

Proof: An induction on the steps of the algorithm.

Lemma 2: At any step before BFS terminates there is a psg D' in G' such that $F(D')<=Q$.

Proof: Consider the solution graph D which determines the value of Q. By Lemma 1, we have D' as a psg of G' which is also a ps of D. Clearly, $F(D') <= Q$.

Theorem 1: Algorithm BFS terminates with a solution graph D (if it exists) of cost $F(D) <= Q$.

Proof: Similar to Nilsson [10, pg. 78], assuming the cost of the arcs to be positive and finite. A contradiction can be reached using Lemma 2.

Corollary 1: If $Q = Q_{opt} = h^*(s)$ then BFS terminates with a solution graph D of cost $F(D) = h^*(s)$.

Corollary 2: If $H(P) <= H^*(P)$ for all ps then BFS terminates with minimal cost solution.

Proof: For the minimal cost solution graph we have

$$F(D_{min}) = \max_{P \in D_{min}} \left\{ G(P) + H(P) \right\} \text{ as } H<=H^*$$

So $F(D_{min}) = Q_{opt}$. Also $F(D_{min}) = h^*(s)$. Thus $Q = Q_{opt} = h^*(s)$. Result follows from Corollary 1.

It may be noted that this is a more relaxed admissibility criteria than $h<=h^*$, because $h<=h^* \Rightarrow H<=H^*$ but not the reverse.

Corollary 3: If $h<=h^*$ for all nodes in G, BFS terminates with minimal cost solution.

Theorem 2: If for any AND/OR graph G, there exists at least one minimal cost solution graph $D_{min}$ for which $F(D_{min}) = \max_{P \in D_{min}} F(P)$, then BFS terminates with minimum cost solution.

Proof: Let BFS terminate with solution graph D. Clearly, $F(D)<= Q<= \max_{P \in D_{min}} [F(P)] = F(D_{min}) = h^*(s)$. Also since $D_{min}$ is the minimal cost solution, $F(D)>=F(D_{min})$. Thus $F(D) = F(D_{min}) = h^*(s)$.

## 4. The Monotone (Consistent) Restriction

The monotone restriction given in [10] is one of the most popular types of heuristic estimates studied for pathfinding algorithms. We now concentrate on some of the properties of such restrictions in AND/OR graphs. We define the monotone restriction as:

Definition: If two ps's or psg's $R_1$ and $R_2$ are such that $R_1$ is a subgraph of $R_2$ then $F(R_1) <= F(R_2)$ ( or $H(R_1) <= cost(R_1,R_2) + H(R_2)$, where $cost(R_1, R_2) = G(R_2) - G(R_1)$ ).

It is obvious that Martelli and Montanari's [6] consistent restriction implies the above but not the reverse. In fact, similar to the admissibility criteria, it is defined over H and not h. Also, for pathfinding algorithms, this condition clearly becomes Nilsson's monotone restriction[10].

Theorem 3: The monotone restriction implies that the F values of sequences of psg's selected by BFS is nondecreasing.

Proof: Obvious.

Theorem 4: If the heuristic estimate follows monotone restriction, then BFS terminates with minimal cost solution.

Proof: For all solution graphs $D_i$ of G, we have by the monotone restriction $F(D_i) = \max_{P \in D_i} F(P)$. So for the minimal cost solution $D_{min}$, $F(D_{min}) = \max_{P \in D_{min}} F(P)$. Result follows from Theorem 2.

5. Questions of Optimality

The question of optimality of BFS in terms of node expansions raises a number of problems. For AND/OR graphs with two admissible estimating functions $h_1$ and $h_2$, such that $h^* >= h_1 > h_2$ for all nodes in G, we cannot conclude that $h_1$ expands no more nodes than $h_2$. An example can easily be found to support the above statement. But this seems to put the idea that the more informed heuristic is more efficient at fault. The problem arises out of two major reasons;

(i) for AND/OR graphs we have two evaluation functions, one to select the most promising psg D' in G' and then to select the node in D' to expand.

(ii) the expansion of a single node in G' may lead to the extension of more than one psg. So the promise of a psg may change even when that psg is not selected for expansion by BFS.

Now if we impose two restrictions to overcome these problems, certain results hold. The restrictions are:

R1 : Whenever two algorithms $BFS_1$ and $BFS_2$ select the same psg for extension, they select the same node for expansion(that is the second evaluation function in (ii) above is consistent).

R2 : Expansion of a node leads to the extension of only one psg in G'.

Lemma 3: If $Q = Q_{opt} = h^*(s)$ then the conditions for a psg D' in G' to be selected is: necessary condition : $F(D') <= h^*(s)$

sufficient condition: $F(D') < h^*(s)$

Proof: Clear from Lemma 2 and Corollary 1.

Lemma 4: R1 and R2 implies that whenever two algorithms $BFS_1$ and $BFS_2$ select the same node for expansion, they have selected the same psg.

Proof: By induction on the sequence of psg's generating the psg D' containing the node n.

Theorem 5: For two such algorithms $BFS_1$ and $BFS_2$ using heuristic estimates $h_1$ and $h_2$, respectively, if for all psg's D' we have $H^*(D') >= H_1(D') > H_2(D')$, where $H_i$ corresponds to $h_i$, then $BFS_2$ expands every node expanded by $BFS_1$, provided R1 and R2 hold.

Proof: Similar to Nilsson [10, pp. 81]. By induction on the such trees of psg's formed by the two algorithms and using Lemma 3 and Lemma 4.

Corollary 4: If $h^* >= h_1 > h_2$ for G then Theorem 5 holds with restrictions R1 and R2.

Corollary 6: For ordinary graphs Theorem 5 holds without restrictions.

## 6. Weighted Heuristic Search

As in pathfinding algorithms [10, pp. 88], for AND/OR graphs the relative weights of G and H in the cost function F can be controlled using $F(W,D') = (1-W)\,G(D') + W\,H(D')$, $0 \le W \le 1$. For ordinary graphs some recent results on weighted evaluation functions can be found in [1, 4]. Here we derive two important results for AND/OR graphs.

Theorem 6: If BFS uses an evaluation function $F(W,D') = (1-W)\,G(D') + W\,H(D')$ for all psg's and $H/H^*$ is bounded above by d, then $W < 1/(1+d)$ constitutes the range of admissibility of BFS.

Proof: Similar to Dechter and Pearl [4, pp. 513] utilizing Theorem 2 to force F to attain its maximal value for some minimal cost solution graph $D_{min}$.

Corollary 7: If $H \le H^*$, then BFS gives minimum cost solution for $W < 1/2$.

In case of dynamic weighting we generalize the result of Pohl as in Dechter and Pearl [4].

Theorem 7: Let $F(D') = G(D') + H(D')[1 + e\,\xi(D')]$ be a weighted cost function of BFS, where $\xi(D')$ is a pathdependent function of D' satisfying $\xi(D') \le 1$, then BFS terminates with cost at most $H^*(s)[1 + e]$ provided $H \le H^*$.

Proof: Let $D_{min}$ be a minimal cost solution graph.

$$Q \le \max_{P \in D_{min}} F(P) \le \max_{P \in D_{min}} \left\{ G(P) + H^*(P) + e H^*(P) \cdot \xi(P) \right\}$$

$$Q \le H^*(s) + e\,h^*(s) = h^*(s)[1 + e].$$

It may be noted that $H(D')/[G(D') + H(D')]$ is an interesting form of $\xi(D')$ which can be used.

## 7. Non-Additive Cost Measures

Finally we discuss the case where F need not be additive (that is $F = G + H$ type) in nature. Dechter and Pearl [4] have discussed such functions for pathfinding algorithms. We generalize the same idea for AND/OR graphs. We assume that for solution graph F is monotonic in nature, such that, $C(D_1) > C(D_2) \Rightarrow F(D_1) > F(D_2)$, where $C(D_i)$ is the actual cost of the solution graph $D_i$. Thus we define

$$F(D') = \begin{cases} \Psi[C(D')] & \text{where D' is a solution graph and } \Psi(.) \text{ is monotonic.} \\ \emptyset(D') & \text{where D' is not a solution graph and } \emptyset(.) \text{ is arbitrary.} \end{cases}$$

Theorem 8: BFS is $\Psi^{-1}(Q)$ admissible, that is, the cost of the solution graph found by BFS is at most $\Psi^{-1}(Q)$.

Proof: Similar to [4, pp. 51].

Theorem 9: If for any AND/OR graph searched by BFS, there exists at least one optimal cost solution graph $D_{min}$ for which $F(D_{min}) = \max_{P \in D_{min}} F(P)$ then BFS terminates with minimum cost solution.

Proof: Similar to Theorem 2 where $\Psi(C) = C$ and utilizing the monotonicity of $\Psi$ as in [4, pp. 512].

## 8. Conclusion

Finally to conclude, we would like to stress that both pathfinding and problem reduction searches are the same problem, the former being a special case of the latter. For example, in the popular case of additive cost, both are the same problem of optimizing F = G + H. Though the branch-and-bound framework has been a method to study these two problems together as shown in [9], the present approach helps us to establish that nearly all the results for pathfinding search strategies also hold for AND/OR graphs.

## References


1. A. Bagchi and P.K. Srimani: "Weighted Heuristic Search in Networks", Journal of Algorithms 6, (1985) pp. 550-576.
2. A. Bagchi and A. Mahanti: "Search algorithms under different kinds of heuristics - A comparative study", J. ACM 30, 1 (Jan. 1983) pp. 1-21.
3. A. Bagchi and A. Mahanti: "Admissible heuristic search in AND/OR graph", Theoretical Computer Science 24, 2 (July 1983) pp. 207-219.
4. R. Dechter and J. Pearl: "Generalized Best First Search Strategies and the Optimality of $A^*$", J. ACM 32, 3 (July 1985) pp. 505-536.
5. A. Mahanti and A. Bagchi: "AND/OR graph heuristic search methods", J. ACM 32, 1 (Jan. 1985) pp. 28-51.
6. A. Martelli and U. Montanari: "Additive AND/OR graphs", in Proc. of 3rd Int. Joint Conf. on AI(Stanford, Calif. Aug. 1973) pp. 1-11.
7. A. Martelli and U. Montanari: "Optimizing decision trees through heuristically guided search", Commun of ACM 12, 12 (Dec. 1978) pp. 1025-1039.
8. L. Mero: "A heuristic search algorithm with modifiable estimate", Artificial Intelligence 23, 1 (1984) pp. 13-27.
9. D. S. Nau, V. Kumar and L. N. Kanal: "General branch-and-bound and its relation to $A^*$ and $AO^*$", Artificial Intelligence (1984).
10. N. J. Nilsson: "PRINCIPLES OF ARTIFICIAL INTELLIGENCE", Tioga, Palo Alto, Calif. 1980.
11. J. Pearl: "HEURISTICS: INTELLIGENT SEARCH STRATEGIES FOR COMPUTER PROBLEM SOLVING", Addison Wesley, Reading, Mass 1984.