University of California
Heuristic Theory Project
Memo HP-76-1

Adversary Arguments for the Analysis of
Heuristic Search in General Graphs*

by

Jay Munyer and Ira Pohl
University of California
Santa Cruz
July 30, 1976

INFORMATION SCIENCES
UNIVERSITY OF CALIFORNIA, SANTA CRUZ

This research is supported by National Science
Foundation Grant DCR-74-20460.

* To be presented at CSCSI/SCEIO, Vancouver, B.C.,
August, 1976.
Adversary Arguments

\[
\begin{align*}
\text{w} \leq \frac{1}{2}, \text{ the weight on the heuristic term} \\
\text{where} \\
\alpha = \lceil (k-1)/2 \rceil \cdot \left( \frac{w}{1-w} \right) \cdot \left( 2e+1 \right) + 2
\end{align*}
\]

The Cook [1971] results on NP-complete problems and the extension of these results through reduction arguments [Karp 1972] led to a realization that only heuristics can achieve reasonable solutions for otherwise intractable problems (e.g., [Garey et al. 1972]). These results and advances in understanding of a formal model for heuristic search have together reignited a quest for analytic results in the theory of heuristic search.

The basic algorithms for heuristic search as described in the A. I. literature are outlined in Pohl [1970a] or Nilsson [1971]. These models viewed deductive searches as path problems in state spaces. Originally, they were primarily inspired by the empirical work of the University of Edinburgh researchers ([Doran and Michie 1966], [Ross 1973]). Two groups analyzed the efficiency of these methods, the SRI robotics team [Hart, Nilsson and Raphael 1968] and the University of California, Heuristic Theory Project ([Pohl 1969], [Pohl 1973]).

The Pohl model, based on a generalization of the Graph Traverser and algorithm A*, used an algorithm HPA.\(^1\) The different algorithms yielded a variety of results. The GT work focused on the empirically observed efficiency of heuristic search as it related to the quality of the heuristic function. The A* work

\[^1\] See appendix for a formulation of this model.
Adversary Arguments

In dynamic weighting, the relative weighting of \( h(x) \) and \( g(x) \) are affected by their depth in the search tree. The deeper into the search the less weight is placed on the heuristic term.

\[
w(x) = 0.5 + B(x), \quad 0 \leq B(x) < \frac{1}{2}
\]

where \( 1/B(x) \propto \) search depth

In intuitive terms, this is done to avoid being excessively misled by an overly optimistic heuristic. However at the top levels of the search there is stronger reliance on the heuristic term in order to promote a depth first search. Dynamic weighting retains some of the advantages of \( A^* \) where a least costly solution is desired. These advantages were demonstrated by applying these techniques to the Traveling Salesman Problem using the Held-Karp scheme to obtain a heuristic estimator [Held, Karp 1971]. Similar ideas are applicable to game tree searches such as the Harris Bandwidth search [Harris 1973].

Pruning and Partial Development: [Ross 1975], [Michie and Ross 1970]

Ross has considered an algorithm GT4 which only partially develops nodes and is used in conjunction with search tree pruning. He has extended many of the analytic results of the HPA scheme to this more general method.

Theorem [Ross 1975]: Given that the heuristic function satisfies the monotone condition, then \( GT4^* \) over tree domains will in the worst-case look at no more nodes by using \( f = g + h \) in comparison to \( f = h \).
Adversary Arguments

Adversary Arguments

Algorithms performances in problem domains can be compared with respect to their effectiveness on benchmark problems. Adversary arguments are attempts to manufacture very difficult conditions that conform to the problem constraints. The best adversaries lead to the worst-case performance for algorithms in a given problem domain [Knuth 1973].

\[ \text{e.g. adversary for HPA in tree domains:} \]
\[ \text{given: } h^*(x) \text{ is the perfect estimator} \]
\[ h(x) \text{ is of bounded error } \epsilon \]

Let \( h(x) = h^*(x) + \epsilon \) along shortest path otherwise \( h(x) = h^*(x) - \epsilon \); it can be proved that this adversary leads to worst-case performance for HPA acting on tree domains [Pohl 1970b]. (A recent analysis of alpha-beta game tree search was published using this form of argument [Knuth, D. E. and R. W. Moore 1975].)

We now turn to a summary of results extending our analysis from tree domains to general undirected graph domains. An interesting aspect of these results is a comparison between HPA, the original algorithm of Pohl which did not allow nodes in \( S \) to be placed back into \( S \) (i.e., did not allow closed nodes to be reopened), and HPA+, which does allow a closed node to be reopened when a shorter path to it is found. It is necessary to be able to do this when a shortest path to a goal node is desired, but it had not been thought that this would be efficient when any solution path found is satisfactory. Our results however show that, for the worst-case at least, allowing closed nodes to be
Adversary Arguments

in the candidate set which does not exceed this value — no node
can be placed in S of larger f-value. q.e.d.

A similar argument works to give the result for \( w > \frac{1}{2} \).

**Definition** for any node \( n \) in a graph,

\[ h_p(n), \text{ the "perfect heuristic"}, \text{ is the length of the} \]
\[ \text{shortest path from } n \text{ to the goal node } t. \]

\[ g_p(n) \text{ is the length of the shortest path from the start} \]
\[ \text{node } s \text{ to } n \text{ which does not include the goal node} \]
\[ (i.e., \text{ the shortest path from } s \text{ to } n \text{ which would be} \]
\[ \text{found by HPA}). \]

\[ f_p(n) = w \cdot h_p(n) + (1-w) \cdot g_p(n). \]

**Theorem 2.** If HPA+ uses \( w \leq \frac{1}{2} \) and a heuristic function of
bounded error \( e \), then in a graph of shortest solution path \( k \), the
set \( S \) of expanded nodes will always be a subset of a set of nodes
\( T \) where for \( n \in T \), \( f_p(n) \leq 2 \cdot w \cdot e + (1-w) \cdot k. \)

**Pf:** By Lemma 1 we know that for any node \( n \) which is expanded,

\[ f(n) = w \cdot h(n)^+ (1-w) \cdot g(n) \leq w \cdot e + (1-w) \cdot k \]

Since \( h(n) \geq h_p(n) - e \) and \( g(n) \geq g_p(n) \), this becomes

\[ w \cdot h_p(n) - w \cdot e + (1-w) \cdot g_p(n) \leq w \cdot e + (1-w) \cdot k \]

or

\[ w \cdot h_p(n) + (1-w) \cdot g_p(n) = f_p(n) \leq 2 \cdot w \cdot e + (1-w) \cdot k. \]

**Definition:** The **branching rate** \( b \) of a graph \( G \) is the maximum
of \((\text{degree} - 1)\) of any node in \( G \).
Adversary Arguments

Figure 1
Adversary Arguments

Appendix: HPA model of heuristic search [Pohl 1970].

A problem space is a locally finite directed graph G.

\[ G: X = \{x_1, x_2, \ldots \}, \quad X \text{ is the set of nodes and can be infinite} \]

\[ E = \{(x_i, x_j) \mid x_i, x_j \in X, \ x_j \in \Gamma(x_i)\}, \quad E \text{ is the set of edges} \]

and can be infinite if \(|X|\) is infinite.

\[ \Gamma \text{ is the successor map} \]

\[ \Gamma: X \rightarrow 2^X \text{ where for all } x, \ |\Gamma(x)| \in N \]

Heuristic Path Algorithm (HPA)

\[ s = \text{ start node, } t = \text{ terminal node, } x = \text{ any node} \]

\[ g: X \rightarrow N, \text{ the number of edges from } s \text{ to } x \text{ enumerated by HPA—} \]

\[ \text{distance-to-date term} \]

\[ h: X \rightarrow R^+ \text{ (the nonnegative reals), an estimate of the number of} \]

\[ \text{edges on a shortest path from } x \text{ to } t—\text{heuristic function} \]

\[ f(x) = (1-w)g(x) + wh(x), \ 0 \leq w \leq 1 \text{—evaluation function} \]

\[ S = \text{ set of nodes already visited and expanded} \]

\[ \tilde{S} = \text{ set of nodes one edge removed from those in } S, \text{ but} \]

not in \(S\)—candidate set

1. Place \(s\) in \(S\) and calculate \(\Gamma(s)\), placing them in \(\tilde{S}\). If \(x \in \Gamma(s)\), then \(g(x) = 1\) and \(f(x) = (1-w) + wh(x)\).

2. Select \(n \in \tilde{S}\) such that \(f(n)\) is a minimum.

3. Place \(n\) in \(S\) and \(\Gamma(n)\) in \(\tilde{S}\), discarding any nodes already in \(S \cup \tilde{S}\). Calculate \(f\) for these new successors of \(n\). If \(x \in \Gamma(n)\), then \(g(x) = 1 + g(n)\) and \(f(x) = (1-w)g(x) + wh(x)\).

4. If \(n\) is the goal state, then halt, otherwise go to step 2.
Adversary Arguments

References


Adversary Arguments


Pohl, I. [1975], Practical and theoretical consideration in heuristic search algorithms, UCSC Heuristic Theory Project HP-75, NATO-ASI Conference on the Machine Representation of Knowledge, Santa Cruz.


Ross, R. [1975] Recent results on the efficiency of heuristic search, draft report, Machine Intelligence Research Unit, Edinburgh University.