THE OPTIMALITY OF A* REVISITED*

Rina Dechter and Judea Pearl

Cognitive Systems Laboratory
Computer Science Department
University of California, Los Angeles, CA 90024

ABSTRACT

This paper examines the optimality of A^*, in the sense of expanding the least number of distinct nodes, over three classes of algorithms which return solutions of comparable costs to that found by A^*. We first show that A^* is optimal over those algorithms guaranteed to find a solution at least as good as A^*'s for every heuristic assignment h. Second, we consider a wider class of algorithms which, like A^*, are guaranteed to find an optimal solution (i.e., admissible) if all cost estimates are optimistic (i.e., $h \le h^*$). On this class we show that A^* is not optimal and that no optimal algorithm exists unless h is also consistent, in which case A^* is optimal. Finally we show that A^* is optimal over the subclass of *best-first* algorithms which are admissible whenever $h \le h^*$.

1. INTRODUCTION AND PRELIMINARIES

1.1 A* and Informed Best-First Strategies

Of all search strategies used in problem solving, one of the most popular methods of exploiting heuristic information to cut down search time is the *informed best-first* strategy. The general philosophy of this strategy is to use the heuristic information to assess the "merit" latent in every candidate search avenue, then continue the exploration along the direction of highest merit. Formal descriptions of this strategy are usually given in the context of path searching problems, a formulation which represents many combinatorial problems such as routing, scheduling, speech recognition, scene analysis, and others. Given a weighted directional graph G with a distinguished start node s and a set of goal nodes Γ, the *optimal path problem* is to find a lowest cost path from s to Γ where the cost of the path may, in general, be an arbitrary function of the weights assigned to the nodes and branches along that path.

By far, the most studied version of informed best-first strategies is the algorithm A^* (Hart, Nilsson and Raphael, 1968) which was developed for *additive cost measures*, i.e, where the cost of a path is defined as the sum of the costs of its arcs. To match this cost measure, A^* employs a special additive form of the evaluation function f made up from the sum $f(n)=g(n)+h(n)$, where $g(n)$ is the cost of the currently evaluated path from s to n and h is a heuristic estimate of the cost of the path remaining between n and some goal node. A^* constructs a tree T of selected paths of G using the elementary operation of *node expansion*, i.e., generating all successors of a given node. Starting with s, A^* selects for expansion that leaf node of T which has the lowest f value, and only maintains the lowest-g path to any given node. The search halts as soon as a node selected for expansion is found to satisfy the goal conditions. It is known that if $h(n)$ is a lower bound to the cost of any continuation path from n to Γ, then A^* is *admissible*, that is, it is guaranteed to find the optimal path.

1.2 Previous Works

The *optimality* of A^*, in the sense of expanding the *least number of distinct nodes*, has been a subject of some confusion. The well-known property of A^* which predicts that decreasing errors h^*-h can only improve its performance (Nilsson, 1980, result 6) has often been interpreted to reflect some supremacy of A^* over other search algorithms of equal information. Consequently, several authors have assumed that A^*'s optimality is an established fact (e.g., Nilsson, 1971; Martelli, 1977; Méro, 1981; Barr and Feigenbaum, 1982). In fact, all this property says is that some A^* algorithms are better than other A^* algorithms depending on the heuristics which guide them. It does not indicate whether the additive rule $f=g+h$ is better than other ways of combining g and h (e.g., $f=g+h^2/(g+h)$); neither does it assure us that expansion policies based only on g and h can do as well as more sophisticated best-first policies using the entire information gathered along each path (e.g., $f(n)=\max\{f(n') \mid n'$ is on the path to $n\}$). These two conjectures will be examined in this paper, and will be given a qualified confirmation.

Gelperin (1978) has correctly pointed out that in any discussion of the optimality of A^* one should compare it to a wider class of equally informed algorithms, not merely those guided by $f=g+h$, and that the comparison class should include, for example, algorithms which adjust their h in accordance with the information gathered during the search. His analysis, unfortunately, falls short of considering the entirety of this extended class, having to follow an over-restrictive definition of *equally-informed*. Gelperin's interpretation of the statement "an algorithm B is *never more informed* than A" not only restricts B from using information unavailable to A, but also forbids B from processing common information in a better way than A does.

*Supported in part by NSF Grant No. MSC 81-142209

For example, if B is a best-first algorithm guided by an evaluation function $f_B(\cdot)$, then to qualify for Gelperin's definition of being "never more informed than A," B is restricted from ever assigning to a node n an $f_B(n)$ value higher than A would, even if the information gathered along the path to n justifies such an assignment. We now remove such restrictions.

1.3 Summary of Results

In our analysis we use the natural definition of "equally informed," allowing the algorithms compared to have access to the same heuristic information while placing no restriction on the way they use it. We will consider a class of heuristic algorithms, searching for a lowest (additive) cost path in a graph G, in which an arbitrary heuristic function $h(n)$ is assigned to the nodes of G and is made available to each algorithm in the class upon generating node n. From this general class we will discuss three special subclasses: We first compare the complexity of A^* with those algorithms which are *at least as good as* A^* in the sense that they return solutions at least as cheap as A^*'s in *every* problem instance. We denote that class of algorithms by $\mathbf{A}_g$. We then restrict the domain of instances to that on which A^* is admissible, that is, $h \le h^*$ and consider the wider class of algorithms which are as good as A^* *only* on this restricted domain. Here we shall show that A^* is *not optimal* and that *no optimal algorithm exists* unless h is also restricted to be *consistent*. Finally, we will consider the subclass of *best-first* algorithms that are admissible when $h \le h^*$ and show that A^* is optimal over that class.

1.4 Notation and Definitions

G -	directed locally finite graph, $G=(V,E)$
G_e -	the subgraph of G exposed during the search
s -	start node
Γ -	a set of goal nodes, $\Gamma \subseteq V$
P^s -	a *solution path*, i.e., a path in G from s to some goal node $\gamma \in \Gamma$
$C(P)$ -	the cost of path P
$P_{n_i-n_j}$ -	A path in G between node n_i and n_j
$g^*(n)$ -	The cost of the cheapest path going from s to n
$g(n)$ -	The cost of the cheapest path found so far from s to n
$h^*(n)$ -	The cost of the cheapest path going from n to Γ
$h(n)$ -	An estimate of $h^*(n)$, assigned to each node in G
C^* -	The cost of the cheapest path from s to Γ
$c(n,n')$ -	The cost of the arc between n and n', $c(n,n') \ge \delta > 0$.
$k(n,n')$ -	The cost of the cheapest path between n and n'
(G,s,Γ,h) -	A quadruple defining a problem instance

Let the domain of instances on which A^* is admissible be denoted by $\mathbf{I}_{AD}$, i.e.:

$$\mathbf{I}_{AD} = \left\{ (G,s,\Gamma,h) \,\middle|\, h \le h^* \text{ on } G \right\}$$

Obviously, any algorithm in $\mathbf{A}_g$ is also admissible over $\mathbf{I}_{AD}$.

A path on G is said to be *strictly d-bounded relative to f* if every node n' along that path satisfies $f(n')<d$. It is known that if $h \le h^*$, then A^* expands any node reachable by a strictly C^*-bounded path, regardless of the tie-breaking rule used. The set of nodes with this property will be referred to as *surely expanded* by A^*. (Nodes outside this set may or may not be expanded depending on the tie-breaking rule used.) In general, for arbitrary constant d and an arbitrary evaluation function f over (G,s,Γ,h), we denote by $\mathbf{N}_f^d$ the *set of all nodes reachable by a strictly d-bounded path* in G. For example, $\mathbf{N}_{f=g+h}^{C^*}$ is the set of nodes surely expanded by A^* over $\mathbf{I}_{AD}$.

The notion of *optimality* that we examine in this paper is robust to the choice of tie-breaking rules and is given by the following definition:

Definition: An algorithm A is said to be *optimal* over a class $\mathbf{A}$ of algorithms relative to a set $\mathbf{I}$ of problem instances if in each instance of $\mathbf{I}$, every algorithm in $\mathbf{A}$ will expand all the nodes surely expanded by A in that problem instance.

2. RESULTS

2.1 Optimality Over Algorithms As Good As A*

Theorem 1: Any algorithm which is *at least as good* as A^* will expand, if provided the heuristic information $h \le h^*$, all nodes that are *surely expanded* by A^*, i.e., A^* is optimal over $\mathbf{A}_g$ relative to $\mathbf{I}_{AD}$. ∎

Proof: Let $I=(G,s,\Gamma,h)$ be some problem instance in $\mathbf{I}_{AD}$ and assume that n is surely expanded by A^*, i.e., $n \in \mathbf{N}_{g+h}^{C^*}$. Therefore, there exists a path P_{s-n} such that

$$f(n') = g(n')+h(n') < C^* \quad \forall n' \in P_{s-n}$$

Let $D = \max_{n' \in P_{s-n}} \{f(n')\}$ and let B be an algorithm in $\mathbf{A}_g$. Obviously both A^* and B will halt with cost C^*, while $D<C^*$.

Assume that B does not expand n. We now create a new graph G' (see figure 1) by adding to G a goal node t' with $h(t')=0$ and an edge from n to t' with non-negative cost $D-C(P_{s-n})$. Denote the extended path $P_{s-n-t'}$ by P^*, and let $I'=(G',s,\Gamma \cup \{t'\},h)$ be a new instance in the algorithms' domain. Although h may no longer be admissible on I', the construction of I' guarantees that $f(n') \le D$ if $n' \in P^*$, and thus, algorithm A^* searching G' will find a solution path with cost $C_t \le D$ (Dechter & Pearl, 1983). Algorithm B, however, will search I' in exactly the same way it searched I; the only way B can reveal any difference between I and I' is by expanding n. Since it did not, it will not find solution path P* but will halt with cost $C^*>D$, the same cost it found for I. This contradicts its property of being as good as A^*. ∎

2.2 Nonoptimality Over Algorithms Compatible with A*

Theorem 1 asserts the optimality of A^* over a somewhat restricted class of algorithms, those which *never* return a solution more expensive than A^*'s, even in instances where non-admissible h are

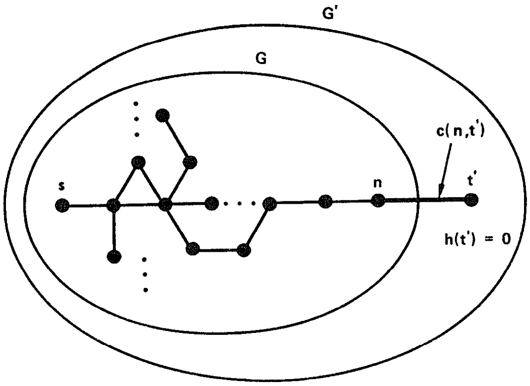

Figure 1

provided. If our problem space includes only admissible cases we should really be concerned with a wider class $\mathbf{A}_c$ of competitors to A^*, those which only return as good a solution as A^* in instances of $\mathbf{I}_{AD}$, regardless of how badly they may perform hypothetically under non-admissible h. We shall call algorithms in this class *compatible* with A^*.

Disappointedly, A^* cannot be proven to be optimal over the entire class of algorithms compatible with it, and, in fact, some such algorithms may grossly outperform A^* in specific problem instances. For example, consider an algorithm B guided by the following search policy: Conduct an exhaustive right-to-left depth-first search but refrain from expanding one distinguished node n, e.g., the leftmost son of s. By the time this search is completed, examine n to see if it has the potential of sprouting a solution path cheaper than all those discovered so far. If it has, expand it and continue the search exhaustively. Otherwise, return the cheapest solution at hand. B is clearly compatible with A^*; it cannot miss an optimal path because it would only avoid expanding n when it has sufficient information to justify this action, but otherwise will leave no stone unturned. Yet, in the graph of Figure 2a, B will avoid expanding many nodes which are surely expanded by A^*. A^* will expand node J_1 immediately after s ($f(J_1)=4$) and subsequently will also expand many nodes in the subtree rooted at J_1. B, on the other hand, will expand J_3, then select for expansion the goal node γ, continue to expand J_2 and at this point will *halt without expanding node* J_1. Relying on the admissibility of h, B can infer that the estimate $h(J_1)=0$ is overly optimistic and should be at least equal to $h(J_2)-1=19$, thus precluding J_1 from lying on a path cheaper than (s, J_3, γ).

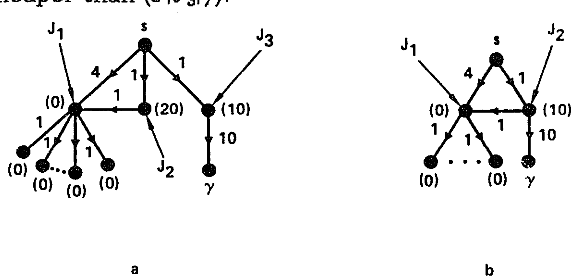

Figure 2

Granted that A^* is not optimal over its compatible class $\mathbf{A}_c$, the question arises if an optimal algorithm exists altogether. Clearly, if $\mathbf{A}_c$ possesses an optimal algorithm, that algorithm must be better than A^* in the sense of expanding, in some problem instances, fewer nodes than A^* while never expanding a node which is surely skipped by A^*. Note that algorithm B above could not be such an optimal algorithm because in return for skipping node J_1 in Figure 2a it had to pay the price of expanding J_2, yet J_2 will not be expanded by A^* regardless of the tie-breaking rule invoked. If we could show that this "node tradeoff" pattern must hold for every algorithm compatible with A^*, and on every instance of $\mathbf{I}_{AD}$, then we would have to conclude that no optimal algorithm exists. Figure 2b, however, represents an exception to the node-tradeoff rule; algorithm B does not expand a node (J_1) which must be expanded by A^* and yet, it never expands a node which A^* may skip.

We now show that cases such as that of Figure 2b may occur only in rare instances.

Theorem 2: If an algorithm B, compatible with A^*, does not expand a node which is surely expanded by A^* and if the graph in that problem instance contains at least one optimal solution path along which h is not fully informed ($h<h^*$), then in that very problem instance B must expand a node which may be avoided by A^*.

Proof: Assume the contrary, i.e., there is an instance $I=(G,s,\Gamma,h)\in\mathbf{I}_{AD}$ such that a node n which is surely expanded by A^* is avoided by B and, at the same time, B expands no node which is avoided by A^*, we shall show that this assumption implies the existence of another instance $I'\in\mathbf{I}_{AD}$ where B will not find an optimal solution. I' is constructed by taking the graph G_e exposed by a specific run of A^* (including nodes in OPEN) and appending to it another edge (n,t') to a new goal node t', with cost $c(n,t')=D'-k_e(s,n)$ where

$$D' = \max\left\{f(n') \,\middle|\, n' \in \mathbf{N}^{C^*}_{g+h}\right\},$$

and $k_e(n_1,n_2)$ is the cost of the cheapest path from n_1 to n_2 in G_e.

Since G contains an optimal path $P^*_{s-\gamma}$ along which $h(n')<h^*(n')$ (with the exception of γ and possibly s), we know that there is a tie-breaking rule that will guide A^* to find $P^*_{s-\gamma}$ and halt without ever expanding another node having $f(n)=C^*$. Using this run of A^* to define G_e, we see that every nonterminal node in G_e must satisfy the strict inequality $g(n)+h(n)<C^*$.

We shall first prove that I' is in $\mathbf{I}_{AD}$, i.e., that $h(n')\le h^*_{I'}(n')$ for every node n' in G_e. This inequality certainly holds for n' such that $g(n')+h(n')\ge C^*$ because all such nodes were left unexpanded by A^* and hence appear as terminal nodes in G_e for which $h^*_{I'}(n')=\infty$ (with the exception of γ, for which $h(\gamma)=h^*_{I'}(\gamma)=0$). It remains, therefore, to verify the inequality for nodes n' in $\mathbf{N}^{C^*}_{g+h}$ for which we have $g(n')+h(n')\le D'$. Assume the contrary, that for some such $n'\in A^{C^*}_{g+h}$ we have $h(n')>h^*_{I'}(n')$. This implies

$$h(n') > k_e(n',n) + c(n,t')$$
$$= k_e(n',n) + D' - k_e(s,n)$$
$$\geq k_e(n',n) + k_e(s,n') + h(n') - k_e(s,n)$$

or

$$k_e(s,n) > k_e(n',n) + k_e(s,n')$$

in violation of the triangle inequality for cheapest paths in G_e. Hence, I' is in $\mathbf{I}_{AD}$.

Assume now that algorithm B does not generate any node outside G_e. If B has avoided expanding n in I, it should also avoid expanding n in I'; all decisions must be the same in both cases since the sequence of nodes generated (including those in OPEN) is the same. On the other hand, the cheapest path in I' now goes from s to n to t', having the cost $D'<C^*$, and will be missed by B. This violates the admissibility of B on an instance in $\mathbf{I}_{AD}$ and proves that B could not possibly avoid the expansion of n without generating at least one node outside G_e. Hence, B must expand at least one node avoided by A^* in this specific run.

∎

Theorem 2 can be given two interpretations. On one hand it is discomforting to know that neither A^* nor any other algorithm is truly optimal over those guaranteed to find an optimal solution when given $h \leq h^*$, not even optimal in the restricted case of ensuring that the set of nodes *surely* expanded by that algorithm is absolutely the minimal required.* On the other hand, Theorem 2 endows A^* with some optimality property, albeit weaker than hoped; the only way to gain one node from A^* is to relinquish another. Not every algorithm enjoys such strength.

2.3 Optimality Under Consistent Heuristics

We shall now prove that conditions like those of Figure 2, which permit other algorithms to outmaneuver A^*, can only occur in instances where h is nonconsistent; in other words, if in addition to being admissible h is also consistent (or monotone) then A^* is optimal over the entire class of algorithms compatible with it.

Definition: A heuristic function h is said to be consistent if for any pair of nodes, n' and n, the triangle inequality holds: $h(n') \leq k(n',n)+h(n)$. Clearly, consistency implies admissibility but not vice versa.

Theorem 3: Any algorithm which is admissible over $\mathbf{I}_{AD}$ (i.e., compatible with A^*) will expand, if provided a consistent heuristic h, all nodes that are surely expanded by A^*.

Proof: We again construct a new graph G', as in Figure 1, but now we assign to the edge (n,t') a cost $c=h(n)+\delta$, where

$$\delta = \tfrac{1}{2}[C^* - D'] > 0$$

and

$$D' = \max\left\{f(n') \,\middle|\, n' \in \mathbf{N}^{C^*}_{g+h}\right\}$$

This construction creates a new solution path P* with cost at most $C^*-\delta$ and, simultaneously, (due to h's consistency) retains the admissibility of h on the new instance I'. For, if at some node n' we have

$$h(n') > h^*_{I'}(n') = \min\left[k(n',n) + c \; ; \; h^*_I(n)\right],$$

then we should also have (given $h(n') \leq h^*_I(n)$):

$$h(n') > k(n',n) + c = k(n',n) + h(n) + \delta$$

in violation of h's consistency.

In searching G', algorithm A^* will find the extended path P^* costing $C^*-\delta$, because:

$$f(t') = g(n)+c = f(n)+\delta \leq D'+\delta = C^*-\delta < C^*$$

and so, t' is reachable from s by a path strictly bounded by C^* which ensures its selection. Algorithm B, on the other hand, if it avoids expanding n, must behave the same as in problem instance I, halting with cost C^* which is higher than that found by A^*. This contradicts the supposition that B is both admissible and avoids the expansion of node n.

∎

2.4 Optimality Over Generalized Best-First Algorithms

The next result establishes A^*'s optimality over the set of *generalized best-first* (GBF) algorithms which are admissible if provided with $h \leq h^*$. Algorithms in this set operate identically to A^*; the lowest f path is selected for expansion, and the search halts as soon as the first goal node is selected for expansion. However, unlike A^*, these algorithms will be permitted to employ any evaluation function $f(P)$ where $f(P)$ is a function of the nodes, the edge-costs, and the heuristic function h evaluated on the nodes of P, i.e.

$$f(P) \triangleq f(s,n_1,n_2,\ldots,n) = f\left[\{n_i\},\{c(n_i,n_{i+1})\},\{h(n_i)\} \middle| n_i \in P\right].$$

Due to the path-dependent nature of f, a GBF algorithm would, in general, need to unfold the search graph, i.e., to maintain multiple paths to identical nodes. Under certain conditions, however, the algorithm can discard, as in A^*, all but the lowest f path to a given node, without compromising the quality of the final solution. This condition applies when f is *order preserving*, i.e., if path P_1 is judged to be more meritorious than P_2, both going from s to n, then no common extension (P_3) of P_1 and P_2 may later reverse this judgement. Formally:

$$f(P_1) \geq f(P_2) \Rightarrow f(P_1P_3) \geq f(P_2P_3)$$

Clearly, both $f=g+h$ and $f(P)=\max\left\{g(n')+h(n') \middle| n' \in P\right\}$ are order preserving, and so is every combination $f=F(g,h)$ which is monotonic in both arguments.

The following results are stated without proofs (for a detailed discussion of best-first algorithms see Dechter & Pearl (1983) or Pearl (1983)).

*This finding, by the way, is not normally acknowledged in the literature. Méro (1981), for example, assumes that A^* is optimal in this sense, i.e., that no admissible algorithm equally informed to A^* can ever avoid a node expanded by A^*. Similar interpretations are suggested by the theorems of Gelperin (1977).

Theorem 4: Let B be a best-first algorithm using an evaluation function f_B such that for every $(G, s, \Gamma, h) \in \mathbf{I}_{AD}$, f_B satisfies:

$$f(P_i^s) = f(s, n_1, n_2, \ldots, \gamma) = C(P_i^s) \quad \forall \gamma \in \Gamma.$$

If B is admissible for $\mathbf{I}_{AD}$, then $\mathbf{N}_{g+h}^{C^*} \subseteq \mathbf{N}_{f_B}^{C^*}$, and B expands every node in $\mathbf{N}_{g+h}^{C^*}$. Moreover, if f_B is also of the form $f = F(g, h)$ then F must satisfy $F(x, y) \leq x + y$.

∎

An interesting implication of Theorem 4 asserts that any admissible combination of g and h, $h \leq h^*$, will expand every node surely expanded by A^*. In other words, the additive combination $g+h$ is, in this sense, the optimal way of aggregating g and h for additive cost measures.

Theorem 4 also implies that $g(n)$ constitutes a sufficient summary of the information gathered along the path from s to n. Any additional information regarding the heuristics assigned to the ancestors of n, or the costs of the individual arcs along the path, is only superfluous, and cannot yield a further reduction in the number of nodes expanded with admissible heuristics. Such information, however, may help reduce the number of node evaluations performed by A^* (Martelli, 1977; Méro, 1981).

REFERENCES

Barr, A. and Feigenbaum, E.A. 1981. *Handbook of Artificial Intelligence*, Los Altos, Calif.: William Kaufman, Inc.

Dechter, R. and Pearl, J. 1983. "Generalized best-first strategies and the optimality of A^*," UCLA-ENG-8219, University of California, Los Angeles.

Gelperin, D. 1977. "On the optimality of A^*". *Artificial Intelligence*, vol. 8, No. 1, 69-76.

Hart, P.E., Nilsson, N.J. and Raphael, B. 1968. "A formal basis for the heuristic determination of minimum cost paths." *IEEE Trans. Systems Science and Cybernetics*, SSC-4, No. 2, 100-107.

Martelli, A. 1977. "On the complexity of admissible search algorithms." *Artificial Intelligence*, vol. 8, No. 1, 1-13.

Méro, L. 1981. "Some remarks on heuristic search algorithms." *Proc. of Int. Joint Conf. on AI*, Vancouver, B.C., Canada, August 24-28, 1981.

Nilsson, N.J. 1971. *Problem-Solving Methods in Artificial Intelligence*, New-York: McGraw-Hill.

Nilsson, N.J. 1980. *Principles of Artificial Intelligence*, Palo Alto, Calif.: Tioga Publishing Co.

Pearl, J. 1983. *HEURISTICS: Intelligent Search Strategies*, Reading, Mass.: Addison-Wesley, in press.