HIGH-PERFORMANCE A* SEARCH [NG RAPIDLY GROWING HEURISTICS

Stephen V. Chenoweth
Research & Development
NCR Corporation-WHQ5E
Dayton, OH 45479

Henry W. Davis
Wright State University
Dept. of Computer Science
Dayton, OH 45435

Abstract

In high-performance A* searching to solve satisficing problems, there is a critical need to design heuristics which cause low time-complexity. In order for humans or machines to do this effectively, there must be an understanding of the domain-independent properties that such heuristics have. We snow that, contrary to common belief, accuracy is not critical; the key issue is whether or not heuristic values are concentrated closely near a rapidly growing "central function." As an application, we show that, by "multiplying" heuristics, it is possible to reduce exponential average time-complexity to polynomial. This is contrary to conclusions drawn from previous studies. Experimental and theoretical examples are given.

1 Introduction

Two groups of studies, IBagchi and Sen, 1988; Pearl, 1984, Chapter 7], have appeared in which heuristics are modeled as random variables (RW); the model is used to determine what properties heuristics must have if A* is to have average polynomial, versus exponential, asymptotic time-complexity. The state space graph is assumed to be a uniform, b-ary tree, with start at the root, bi-directional arcs each of unit cost, and a single goal N units from start.[1]

Let $h^*(n)$, $h(n)$, respectively, return true and estimated distance to goal from node n. The studies assume that the errors, $h^*(n) - h(n)$, suitably normalized, are independent and identically distributed RV's. We call this the "IID model" of heuristics. In the IID model, attaining average polynomial A* complexity is essentially equivalent to requiring that the values of $h(n)$ be clustered near $h^*(n)$; the allowed deviation is a logarithmic function of $h^*(n)$ itself.[2] See Figure 1.

As a result of studies using the IID model, the impression has been given that high-performance A* search requires accurate heuristics. In this paper we describe a model which is more realistic than the IID model because it places no constraints on errors, or on h; we call it the "NC model." Conclusions from the NC model are not in agreement with those of the IID model: They predict polynomial A* complexity whenever the values of $h(n)$ are logarithmically clustered near $h^*(n)$ + n$(h^*(n))$, where n is an arbitrary, non-negative, non-decreasing function. See Figure 2. Heuristics whose values grow slower than distance to goal cause exponential complexity. See Figure 3. If we think of clustering as a kind of "precision" and clustering near h* as "accuracy," then the difference is this: The IID model favors heuristics with logarithmic accuracy and the NC model favors those which are rapidly growing and have logarithmic precision.

Sections 2, 3 state our results more precisely. Proofs are in an appendix. In Section 4 we apply the NC model to the phenomenon of "multiplying": That is, replacing a heuristic h with zh for some $z > 0$. Studies with the IID model predict that multiplying is of no value in eliminating exponential complexity. The NC model, however, shows that multiplying can often change exponential A* complexity to polynomial. Experimental and theoretical examples are given.

1.1 Motivation for the Mathematical Modeling of Heuristics

John Gaschnig's 1979 dissertation [Gaschnig, 1979] gave the first thorough experimental results on the correlation between the mathematical properties of heuristics and A* performance. His opening sentence states that the work is "based on the premise that in the future more of the subject matter of artificial intelligence (AI) research will be understood mathematically than at present." We agree with Gaschnig's view that establishing such foundations is a challenging, long-term AI objective.

Gaschnig proposed a mathematical model to explain his data, but recognized that in large part it failed to do so [Gaschnig, 1979, Section 3.5.1]. Our motivation is based on the view that heuristics are central to AI, and that we cannot claim to understand them until we have mathematical models which explain experimental results obtained by using them.

Two other reasons for studying the mathematical properties of heuristics and how they affect search are as follows:

(1) Intractability results [Garey and Johnson, 1979] indicate that many problems require a prohibitive amount of time to solve in the worst case. Compromises must be made. One type of compromise is to accept *average* polynomial time complexity. Heuristic search provides a mechanism for this, but we need a theory which tells us precisely when a heuristic provides such payoff.

(2) As our interest in increasingly hard problems grows, our need for good heuristics also grows. Several studies have been made about how humans and machines

[1] In [Bagchi and Sen, 1988] the primary interest is in multiple goals. However, we discuss their results only for the single-goal problem, as that is our interest here.

[2] Confer Theorems 3.1, 3.2 in [Bagchi and Sen, 1988] or Theorem 1 in [Pearl, 1984, Chapter 7].

may design such heuristics [Rendell, 1983; Pearl, 1984, Chapter 4; Christensen and Korf, 1986; Politowski, 1986; Mostow and Prieditis, 1989; Bramanti-Gregor and Davis, 1991]. We believe that such research requires a predictive mathematical model of heuristics. That is, there are domain-independent mathematical properties which a "good" heuristic has, and these must be understood if humans or machines are going to design them.

The A* algorithm is a sensible place to begin because (a) it is a very widely used heuristic search algorithm, and (b) although formally simple, it is not yet well understood how its performance relates to the mathematical properties of the heuristic it uses. Predictive models for A* should enable us to build better mathematical models for game-tree searching, IDA*, AO*, and more sophisticated production control strategies. One ultimate goal is that the search component in AI systems be based more on scientific theory and less on *ad hoc* discoveries of the designer.

1.2 Model Assumptions: Significance and Tradeoffs

A heuristic h is a function which returns a number when evaluated on a node n. What is the advantage in viewing $h(n)$ as a RV? It is that we may view our complexity conclusions as reflecting the aggregate properties of A* performance when A* is run over many problem instances. A possible disadvantage is that constraints made on the $h(n)$ in order to make the mathematics tractable may not reflect the aggregate behavior of real heuristics. The resulting conclusions about A* performance may be of limited use. This is a problem with the IID model. An appeal of the NC model is that no constraints are placed on the $h(n)$.

The assumption that the state space graph is a b-ary tree is useful in providing potential for exponential explosion. The assumption of a single goal is made here to simplify the mathematics. [Bagchi and Sen, 1988] have studied the multi-goal case in the IID model. This, plus the introduction of cycles, is interesting because it allows the possibility of compromising on solution quality when working on worst case intractable problems.[1]

2 Notation and Basic Relations

The graph, G, is an infinite uniform b-ary tree with bi-directional edges of unit cost. See Figure 4. The start node, n_1, is at the root and a unique solution path $n_1, \ldots, n_{N+1}$ leads to the goal, n_{N+1} which is N units away. The

[1] "Solution quality" refers to the extent to which the solution found is not optimal. It is not an issue when there is a single goal and the state space graph is a tree (as is assumed in this paper).

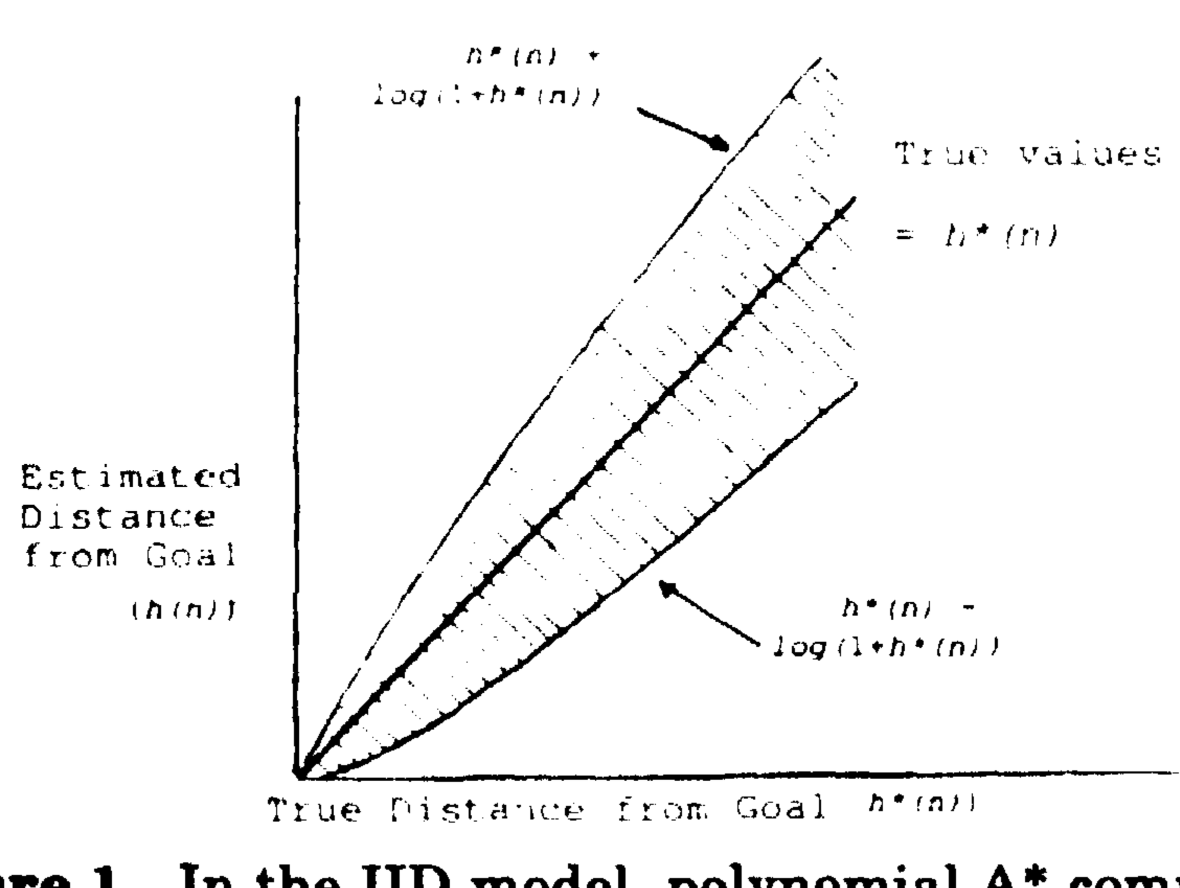

Figure 1. In the IID model, polynomial A* complexity is essentially equivalent to requiring that the heuristic of A* have its values "clustered" as in the shaded region.

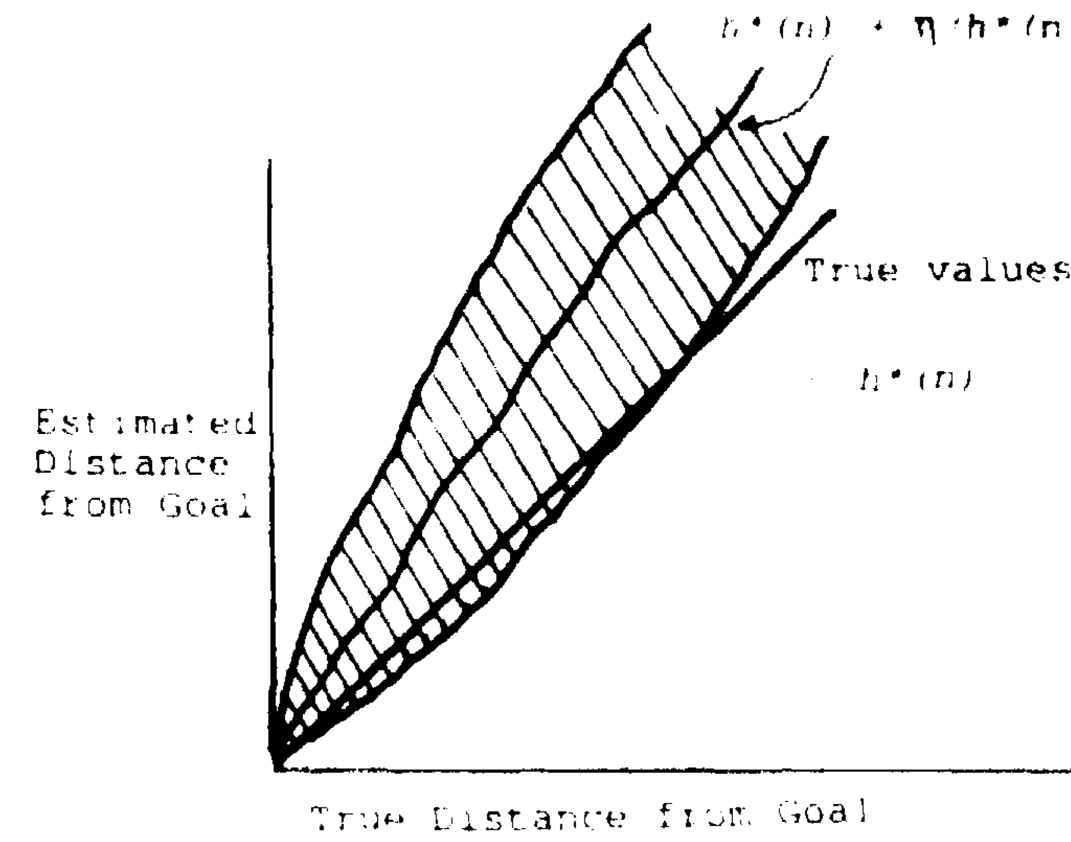

Figure 2. Assume the logarithmic cluster of Figure 1 is now centered on $h^*(n)+\eta(h^*(n))$ instead of on $h^*(n)$. The NC model predicts polynomial complexity; the IID model predicts exponential complexity (unless η is itself a logarithmic function).

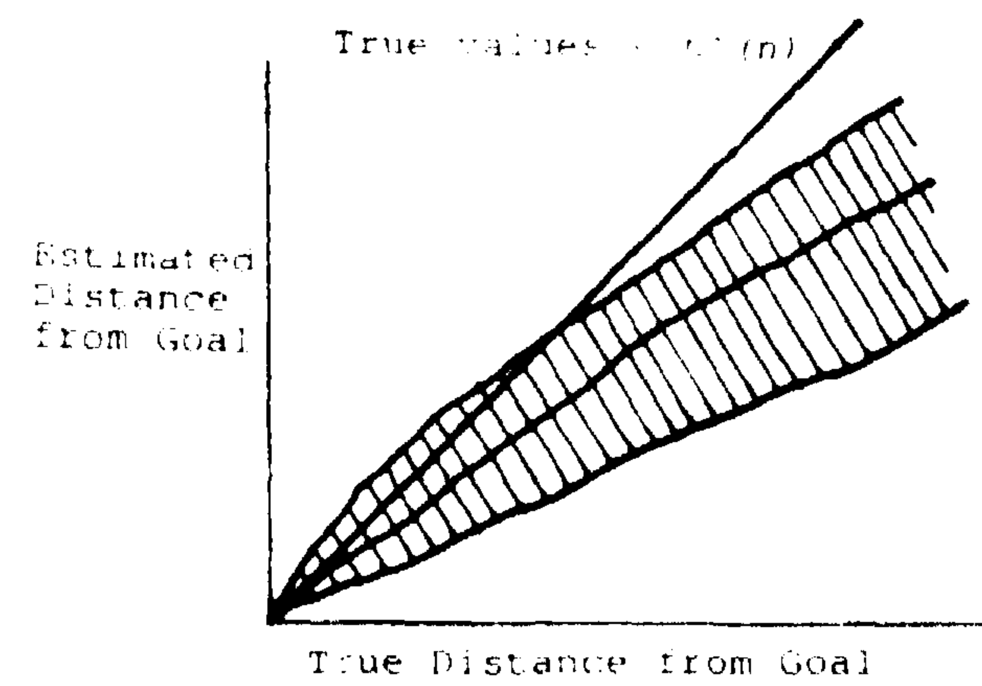

Figure 3. Heuristics whose values increase slower than those of h^* cause exponential A* complexity (Theorem 2).

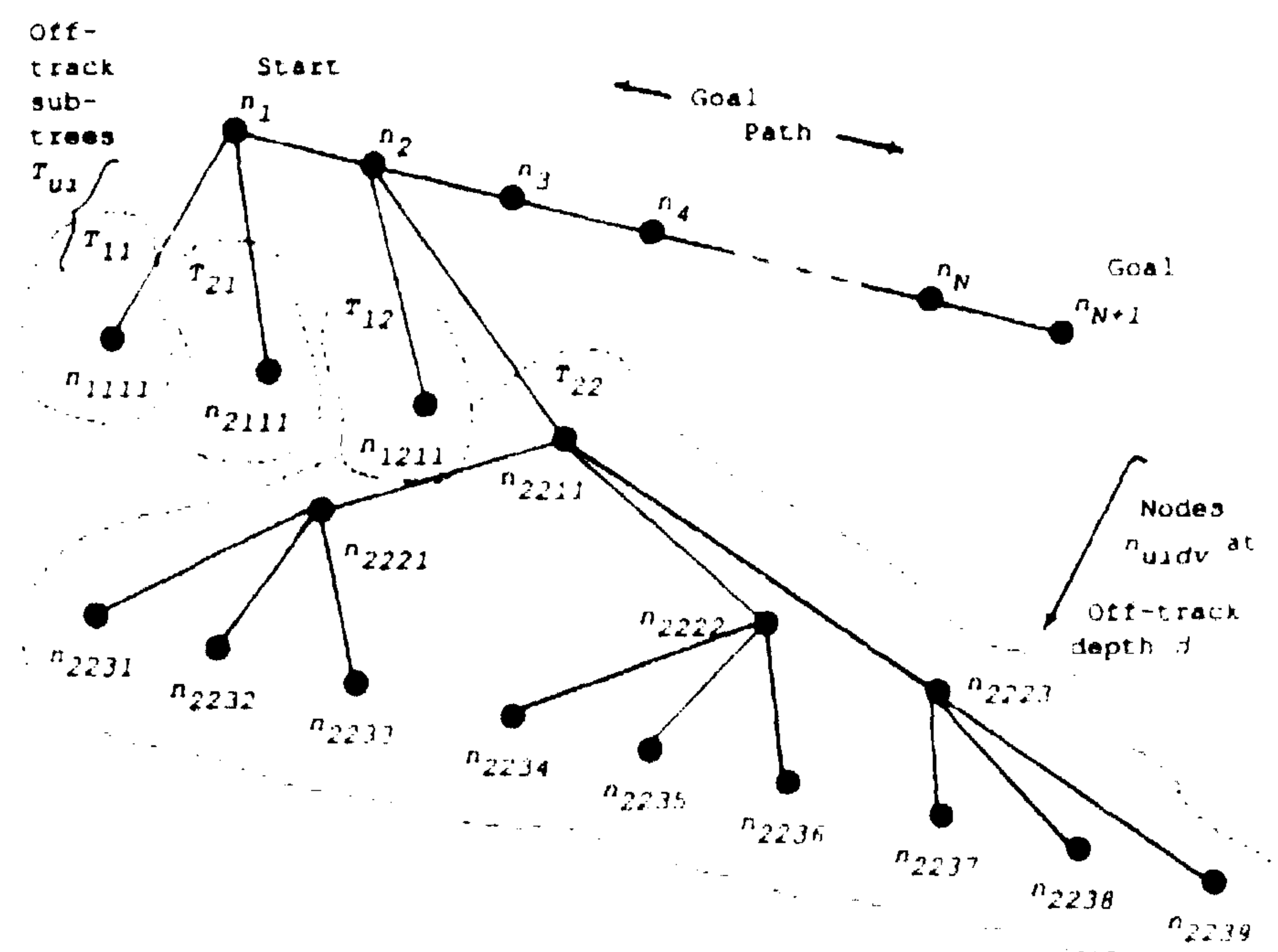

Figure 4. State space graph G with $b = 3$.

nodes $n_1, \ldots, n_N$ (excluding n_{N+1}) are called <u>on-track</u>. <u>Off-track nodes</u> are all other nodes, excluding n_{N+1} and its descendants. The b-1 off-track children of n_i form the roots of the <u>off-track subtree</u> T_{ui}, $1 \le u \le b\text{-}1$, $1 \le i \le N$.

Off-track nodes are denoted n_{uidv}, where T_{ui} denotes the off-track subtree containing n_{uidv}, $d \ge 1$ indicates its <u>depth</u>, and $v \in \{1,\ldots,b^{d-1}\}$ references a particular node on level d within T_{ui}.

We use the following symbolism:

P[...]	Probability of [...].
E[...]	Expected value of [...].
$k(n,m)$	Cost or distance of shortest path from node n to node m in G.
$h^*(n)$	$k(n,n_{N+1})$
$g(n)$	$k(n_1,n)$
$H(n,m)$	Heuristic estimate of $k(n,m)$, a RV. We assume $H(n,n)$=0 for all $n \in G$.
$h(n)$	$H(n,n_{N+1})$, a RV
f	$g + h$
$Z(N)$	Number of nodes expanded by A* when $h^*(n_1) = N$
$Z_{ui}(N)$	Number of nodes on T_{ui} expanded by A* when $h^*(n_1) = N$
exp(n)	Node $n \in G$ is expanded by A*
open(n)	Node n has been generated by A*
q_{uidv}	P[exp(n_{uidv}) | open(n_{uidv})], if P[open(n_{uidv})] > 0; otherwise it is zero.
L^N_i	max $\{f(n_j): i+1 \le j \le N+1\}$.
$\psi(i)$	max $\{j : f(n_j) = L^N_i, i+1 \le j \le N+1\}$, a RV.
n_c	If $n = n_i$ or n_{uidv}, then $n_c = n_{\psi(i)}$. Here $1 \le i \le N$.

The following formulas are used in the sequel:

$$\textbf{(2.1)} \quad \begin{cases} g(n_i) = i\text{-}1, & h^*(n_i) = N+1-i, \\ g(n_{uidv}) = i+d-1, & h^*(n_{uidv}) = d+N+1-i, \\ k(n_{uidv}, n_j) = d+j-i, & \text{where } 1 \le i < j \le N+1, \text{ and } d \ge 1. \end{cases}$$

3 Effects of Heuristic Growth and Clustering Patterns on A* Complexity

Let $Q(x)$ = tjlogCL+tzxts), where L > 0, i = 1,2,3. Let n be a non-negative, non-decreasing function. For x > 0, define

(3.1) $U(x)$=x + r\(x) + o(x), $L(x)$=x + n(x)-Q(x).

We say h is <u>rapidly growing with logarithmic cluster</u> if there exists B e [0,1) such that

(3.2) For all off-track $n = n_{uidv}$ we have

$$P[(h(n) \ge L(h^*(n))) \wedge (h(n_c) \le U(h^*(n_c)))] \ge 1 - (\beta/b)^d.$$

Notice that the "central function," x + n(x), grows at least as fast as the diagonal, but is otherwise arbitrary. The intuitive meaning of the definition is that the $h(n)$ are largely clustered within a logarithmic function of the central function $h^*(n)$ + n$(h^*(n))$. In more detail, off-track nodes have h-values mostly concentrated above a logarithmic lower bound, and on-track nodes mostly below a logarithmic upper bound, with respect to this central function. For example, a special case occurs when each $h(n)$ satisfies \h*(n) + n(h*(n)) - h(n) I ≤ log(l+h*(n)), but is otherwise arbitrarily distributed. See Figure 2. Another example is shown in Figure 6 (which is discussed later); in the figure, h_2 denotes the Manhattan distance for the 8-Puzzle.

According to the theorem below, such clustering causes A* polynomial complexity on average, and also in the worst case if (3 = 0 (which happens in Figures 2, 6).

<u>Theorem</u> 1 (Logarithmic Cluster Complexity Theorem). If h is rapidly growing with logarithmic cluster then $E[Z(N)]$ is polynomial in N; also, $Z(N)$ is polynomial if B m 0 in (3.2).

As an example, we ran A* on the graph of Figure 4 with N up to 20 and 6 = 2. The heuristic h had values as in Figure 2 (logarithmic cluster) with n(x) = x^2. Within these constraints h was designed to be as misleading as possible, returning the maximum allowed number on-track and the minimum allowed number off-track. A* expanded N nodes; i.e., it had linear complexity. The high-performance of A* using this heuristic is surprising because the traditional view, based on the IID model, is that good heuristics need to be accurate.

In contrast to Theorem 1, the theorem and corollary below show that slowly growing heuristics cause exponential complexity. The corollary says to expect exponential complexity if the maximum of /ifnj-values eventually grows slower than does $h^*(n)$. This is shown in Figure 3. The theorem has a more delicate interpretation: If $h(n)$ for open off-track nodes is likely to be growing slower than $h^*(n)$ eventually, then we may expect exponential complexity.

<u>Theorem</u> <u>2</u> (Slowly Growing Heuristic Complexity Theorem). Suppose there exists a differentiable function w such that for some $\varepsilon \in (0,1)$,

$$\lim_{x \to \infty} w'(x) < 1 - \varepsilon ,$$

and for some $\alpha \in (1,b]$, some N_1 a positive integer, we have that node n off-track and $h^*(n) > N_1$ imply

$$P[h(n) \le w(h^*(n)) \mid open(n)] \ge \alpha/b \quad \text{whenever } P[open(n)] > 0.$$

Then E[$Z(N)$] is exponential in N.

<u>Corollary</u>. Let w be as in Theorem 2. If there exists integer N_1>0 such that P[$h(n) \le w(h^*(n))$] = 1 for all n satisfying $h^*(n) > N_1$. then E[$Z(N)$] is exponential.

4 Multiplying Heuristics to Reduce A* Complexity

If h is a heuristic function one might hope to lower A* complexity by using, instead of h, the <u>multiplied heuristic</u>, zh, where z > 0 is a real number. We call z a <u>multiplier</u>. [Gaschnig, 1979] first demonstrated via compelling statistical evidence that the use of multipliers with heuristics can substantially reduce A* complexity.[1] Two other attempts have been made to explain with a mathematical model the effects of multiplying. Both use the IID model. [Pearl, 1984, see especially pp. 206, 209] concludes that multiplying cannot turn exponential growth rate into sub-exponential. IBagchi and Sen, 1988, page 158] conclude that search with multiplied heuristics, "in its average performance, appears to enjoy no clear advantage over ordinary heuristic search."[2]

The NC model provides a simple rationale for expecting multiplications to reduce exponential search complexity to sub-exponential in many cases; it also suggests a good multiplier: Suppose the maximum values of a heuristic h are growing slower than h^* but that overall the heuristic values show a logarithmic cluster around some "central function." Choose z > 0 such that the central function, multiplied by z, grows faster than the diagonal, if such is possible. According to Theorems 1 and 2, above, replacing h by zh reduces A* complexity from exponential to polynomial. We give below experimental and theoretical examples of this.

4.1 Example 1 (Experimental)

The 8-Puzzle [Pearl, 1984, Section 1.1.2] is a sliding block problem whose state space graph is only a first approximation to the tree state space graph of our model. While relatively large (181,440 nodes, maximum h*-value 30), it is finite, does not have a uniform branching factor,

1 Gaschnig proposed a mathematical model to explain his data, but recognized that it had significant shortcomings [Gaschnig, 1979, Section 3.5.1]. In particular, for large multipliers the model predicted increasing complexity, the reverse of his experimental results [Gaschnig, 1979, p. 112].

2The problem with using the IID model to study multiplied heuristics is that multiplying an IID-model heuristic may throw it out of the IID model. An example of this is given in [Chenoweth, 1990, Chapter 5]. This reference also contains a more detailed analysis than is given here of the effects of multiplying heuristics.

and contains cycles. However, most cycles are quite long, making the state space graph rather "tree-like"; search trees built by A* on 8-Puzzle problems have average branching factors close to 1.4. We would hope that, to a first approximation, our model is applicable to this domain.

We use as a heuristic for A* the Manhattan distance, *h2.* This is the sum over all tiles of the horizontal and vertical distance of the tile from its goal position. Figure 5 shows the observed values of h_2 from a sample of 1998 randomly generated problems. From the *maxh* plot and Theorem 2 we expect exponential complexity when A* uses $g + h_2$ as its evaluation function. Figure 7, taken from a sample of 605 randomly generated problems, shows that this indeed occurs (z = 1 curve: Notice that the ordinate has a logarithmic scale).

Consider the *avgh* curve of Figure 5 as a "central function" around which h_2-values cluster. We multiply h_2 by 4 so that *4avgh* increases at least as fast as the diagonal, a requirement of Theorem 1. Figure 6 shows the result, and also that the values of $4h_2$ are concentrated within a logarithmic envelope of 4-*avgh.* According to Theorem 1 we may expect polynomial complexity when A* uses $g + 4h_2$. The pattern of data observed in Figure 7 corroborates this expectation. It turns out that increasing the multiplier above 4 causes relatively little improvement in A* performance.

4.2 Example 2 (Theoretical)

There are many theoretical examples of heuristics for which multiplication transforms A* complexity from exponential to polynomial. For instance, let $R(x) = (x + \log(1+x))/2$, $S(x) = x/2$, and let each $h(n)$ be distributed arbitrarily between $S(h^*(n))$ and $R(h^*(n))$, as is shown in Figure 8. By Theorem 2, A* search is exponential when using *h*. By Theorem 1, it is polynomial when using *zh,* for any $z > 2$.

5 Conclusions

In high-performance A* searching to solve satisficing problems, there is a critical need to design heuristics which cause low time-complexity. In order for humans or machines to do this effectively, there must be an understanding of the domain-independent properties that such heuristics have. We have shown that, contrary to previous belief, accuracy is not critical; the key issue is whether or not heuristic values are concentrated closely near a rapidly growing "central function."

An open question is how much concentration, or clustering, is needed to assure polynomial time-complexity. We have shown that logarithmic clustering is adequate. Results reported in [Chenoweth, 1990; Chenoweth and Davis, 1990] show that the clustering can be polynomial when heuristic values grow fast enough. A necessary and

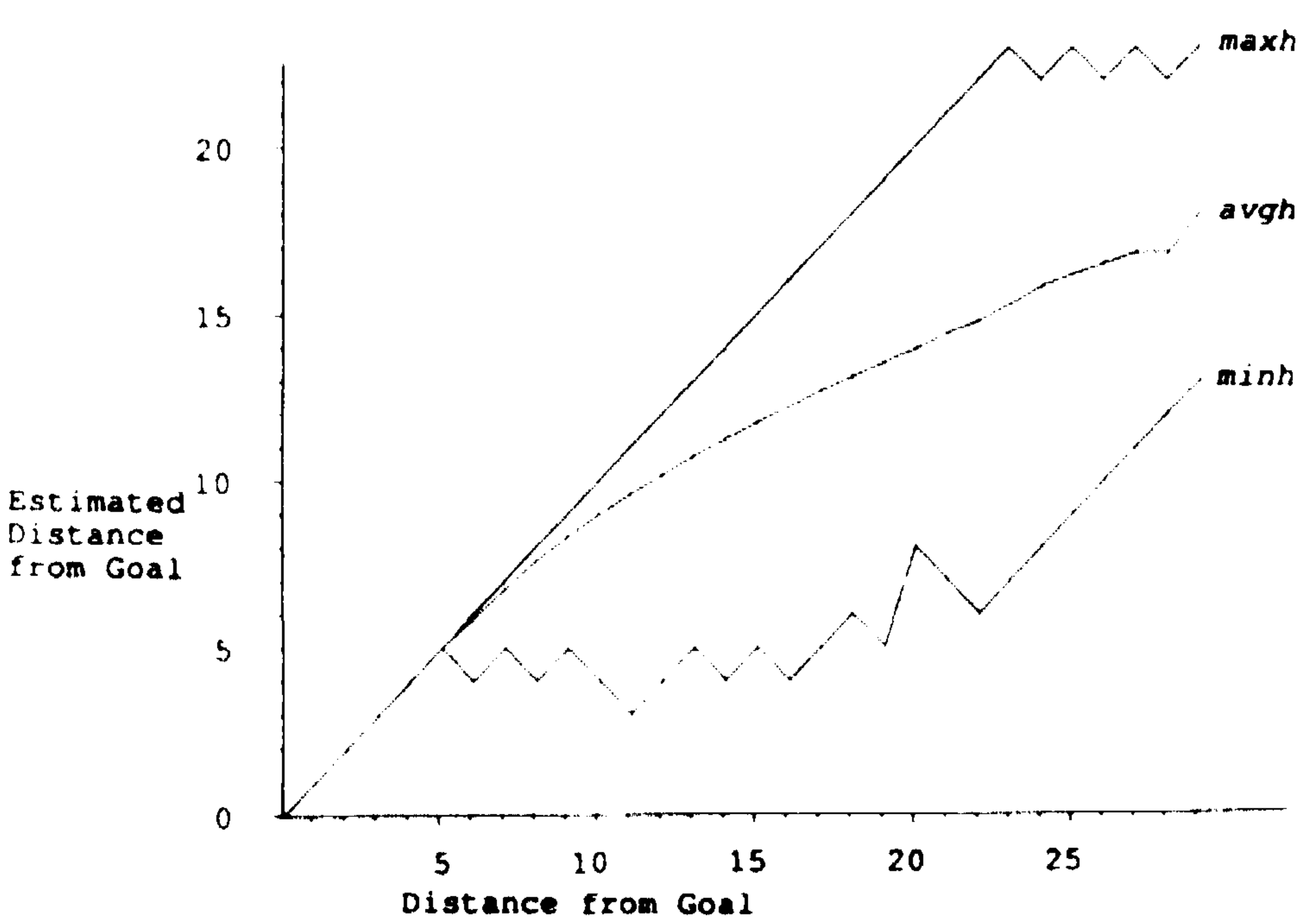

Figure 5. Observed values of h_2 [Chen, 1989]. *avgh, maxh, minh,* respectively, denote average, maximum, and minimum values returned by $h_2(n)$ for various values of $h^*(n)$.

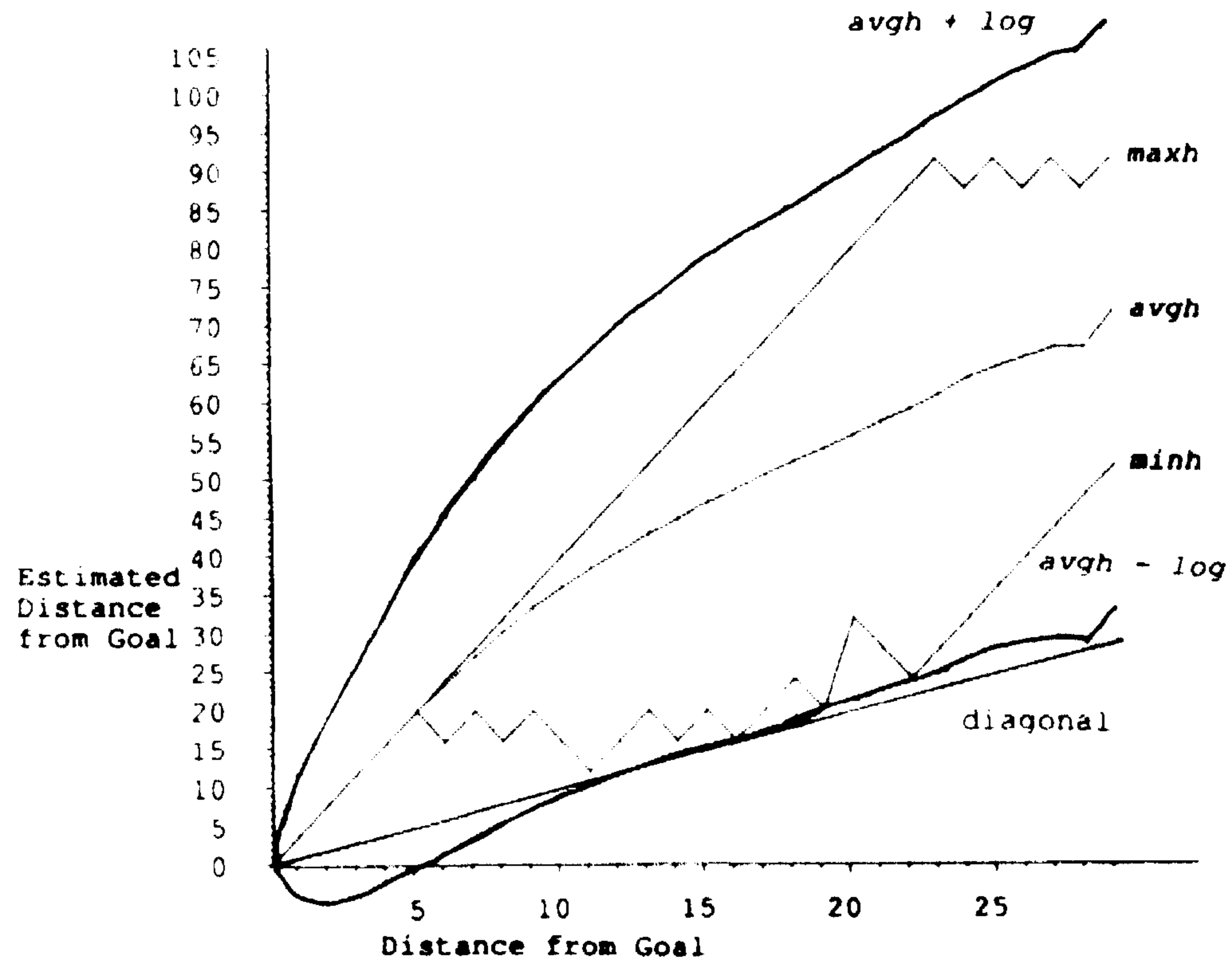

Figure 6. Observed values of $4h_2(n)$ lie between the functions $avgh(x) \pm 11.35\log(1+x)$, where $x = h^*(n)$. This is denoted *avgh*±log in the Figure.

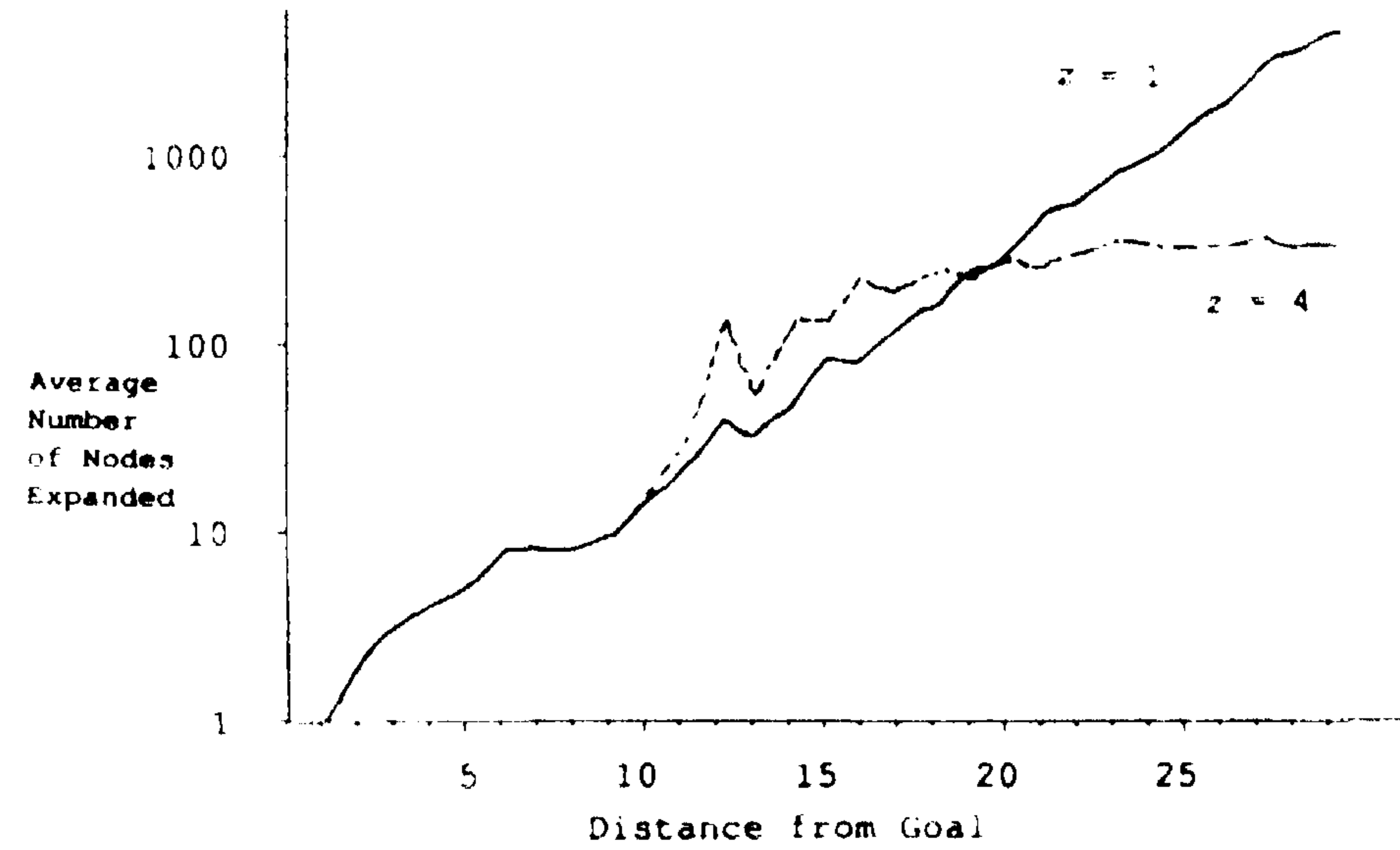

Figure 7. Number of nodes expanded by A* versus distance to goal when using heuristic zh_2, $z = 1, 4$. Data from [Chen, 1989].

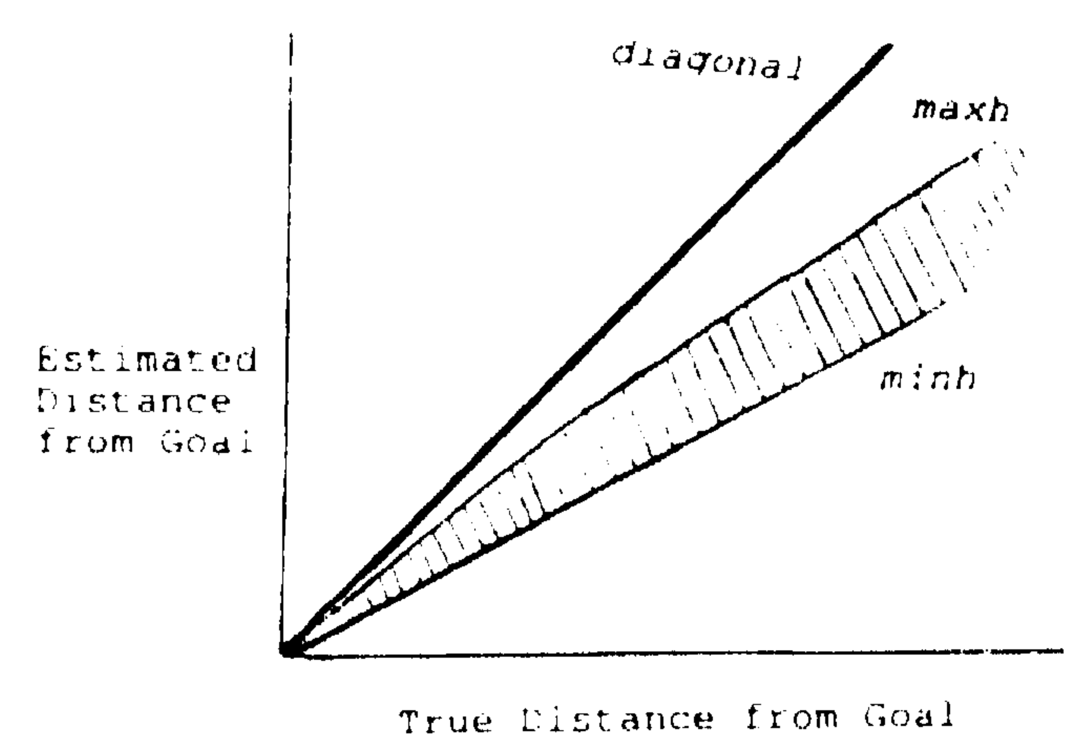

Figure 8. Values assumed by *h* in Example 2: *maxh, minh* are determined by the functions *R*, *S* described there. The state space graph is as in Figure 4.

sufficient condition for polynomial time-complexity is needed.

As an application, we showed that, by "multiplying" heuristics, it is possible to reduce exponential average time-complexity to polynomial. This is contrary to the conclusion drawn from the traditional (IID) model. Experimental and theoretical examples demonstrating this result are given.

An application-related research question is as follows: Admissibility and consistency of a heuristic can often be deduced by relating the heuristic to constraints of the problem domain. See fPearl, 1984, Chapter 4]. Can similar considerations provide insight about heuristic clustering patterns and growth rates?

Appendix: Proofs for Theorems 1, 2

We make use of a function D^N defined on nodes which are off-track or of the form n_i, $1 \le i \le N$: We define $D^N(n) = k(n,n_c)+h(n_c)-h(n)$.

Lemma 3. Let $i \in \{1, \ldots, N\}$, $d \ge 1$. Then[1]

$$f(n_{uidv}) \le L^N{}_i \quad \text{iff} \quad D^N(n_{uidv}) \ge 2d.$$

Lemma 3 is important because a node n_{uidv} cannot be expanded if $f(n_{uidv}) > L^N{}_i$. This fact is used in the proof of Theorem 1.

Proof. By definition $\psi(i)$ is an index for which $f(n_{\psi(i)}) = L^N{}_i$. Thus, $f(n_{uidv}) \le L^N{}_i$ is equivalent to

$$g(n_{uidv}) + h(n_{uidv}) \le g(n_{\psi(i)}) + h(n_{\psi(i)}) \text{ , or}$$
$$(i + d - 1) + h(n_{uidv}) \le [\psi(i) - 1] + h(n_{\psi(i)}) \text{ , or}$$
$$2d \le [d + \psi(i) - i] + h(n_{\psi(i)}) - h(n_{uidv}),$$

which is $\quad 2d \le D^N(n_{uidv})$. **Q.E.D.**

The next lemmas are of a technical nature.

Lemma 4. Assume $\gamma > 0$. Then

(A.1) $\quad x/log(N+x) \ge \gamma$

for all $x \ge log_a(2N)$ provided $N \ge N_0$ where $e = a^\gamma$, and $N_0 \ge 1$ is chosen so large that

(A.2) $\quad y \ge N_0$ **implies** $log_a(2y) \le y$.

Proof. If $x \ge log_a(2N)$ and $N \ge N_0$, then

(1) $\quad \gamma \, log(N+x) = (log_a e) \, log(N+x)$, since $e = a^\gamma$,
$\qquad = log_a(N+x)$.

Take any $N \ge N_0$. By (A.2), $log_a(2N) \le N$. We consider two cases:

Case (1): $log_a(2N) \le x \le N$. Then by (1),
$\gamma \, log(N+x) \le log_a(2N)$, as $x \le N$
$\qquad \le x$, by assumption on x,
which is (A.1).

Case (2): $x > N$. Since $x > N \ge N_0$, (A.2) implies

(2) $\quad log_a(2x) \le x$. Thus, by (1),
$\gamma \, log(N+x) < log_a(2x)$, since $x > N$
$\qquad \le x$, by (2) ,
again yielding (A.1). **Q.E.D.**

Lemma 5. Let $\theta(x) = t_1 log(1 + t_2 x^{t_3})$, where $t_i > 0$, $i = 1,2,3$. Let $\lambda > 0$, and suppose M is a positive integer. Then

(A.3) $\quad 2d/\theta(d+M) \ge \lambda$ **whenever**

(A.4) $\quad \begin{cases} d \ge log_a(2M) \text{ , and} \\ a = exp(1/\lambda c t_1 t_3) \text{ ; where} \end{cases}$

(A.5) $\quad \begin{cases} c = \log 3 / \log 2 \text{ ; and where } M \text{ satisfies} \\ M \ge max\{ (2/t_2)^{1/t_3}, (1/2) a^{\lambda c t_1 \log t_2}, N_1 \}, \end{cases}$

(A.6) **where** $y \ge N_1$ **implies** $log_a(2y) \le y$.

Proof. First note that our choice of c in (A.5) gives $c \, log(v) \ge log(1+v)$ for all $v \ge 2$, which is easily checked. (A.3) is true provided

(1) $\quad d/[c \, log(t_2[d+M]^{t_3})] \ge \lambda t_1 /2$

because of (A.5) and the first part of (A.6). But (1) is true provided $d/\{log \, t_2 + t_3 log[d+M]\} \ge \lambda ct /2$, or

(2) $\quad (log \, t_2)/d + (t_3 log(d+M))/d \le 2/[\lambda c t_1]$.

By Lemma 4 (with $x=d$, $\gamma=\lambda c t_1 t_3$, and $N_0=N_1$),

$$[t_3 log(d+M)]/d \le 1/[\lambda c t_1]$$

provided (A.4) holds and $M \ge N_1$. For such d the left side of (2) is $\le [log \, t_2]/[log_a(2M)] + 1/[\lambda c t_1] \le 2/[\lambda c t_1]$ provided $[log \, t_2]/[log_a(2M)] \le 1/[\lambda c t_1]$; that is,

(3) $\quad log_a(2M) \ge \lambda c t_1 log(t_2)$,

which is included in (A.6). This proves (A.3). **Q.E.D.**

Lemma 6. Let θ be as in Lemma 5. Set $\lambda = 3$ and $M = N$ $(= h^*(n_1))$ in (A.4), (A.5), and (A.6). Assume there exist $\beta \in [0,1)$ such that for all off- track nodes n_{uidv} we have $P[exp(n_{uidv})] \le (\beta/b)^d$ whenever (A.4), (A.5), and (A.6) hold. Then $E[Z(N)]$ is polynomial in N. If $\beta=0$, then $Z(N)$ is polynomial.

Proof. Set $M = N$ in the statement of Lemma 5. Let $L(N) = log_a(2N)$, where a is as in (A.4). Consider any N greater than the right side of (A.6). Then, for any off-track sub-tree T_{ui}, we have

$$\text{(1)} \quad E[Z_{ui}(N)] = \sum_{d=1}^{\infty} \sum_{v=1}^{b^{d-1}} P[exp(n_{uidv})]$$

$$\le \sum_{d<L(N)} \sum_{v=1}^{b^{d-1}} 1 + \sum_{d \ge L(N)} \sum_{v=1}^{b^{d-1}} P[exp(n_{uidv})]$$

$$\le \sum_{d=1}^{L(N)} b^{d-1} + \sum_{d \ge L(N)} b^{d-1} (\beta/b)^d \quad \text{by hypothesis,}$$

$$\le \frac{b^{L(N)} - 1}{b - 1} + \frac{\beta^{L(N)}}{b(1 - \beta)} \le \frac{2}{1 - \beta} b^{L(N)}.$$

Therefore,

$$E[Z(N)] = N + \sum_{u=1}^{b-1} \sum_{i=1}^{N} E[Z_{ui}(N)]$$

$$\le N + [2Nb \, b^{L(N)}] / [1-\beta],$$

which is polynomial. If $\beta=0$, then, with probability 1, the left side of (1) may be replaced by $Z_{ui}(N)$ and the summation on the right side involves only $d < L(N)$. Thus, each $Z_{ui}(N)$, and hence $Z(N)$, is polynomial. **Q.E.D.**

Proof of Theorem 1. We will prove the theorem by applying Lemma 6. To this end, let β be as in (3.2), and let $n=n_{uidv}$ be an arbitrary off-track node. Let (A.4), (A.5), (A.6) hold with M replaced by N, and $\lambda = 3$. We must show that $P[exp(n)] \le (\beta/b)^d$.

Applying (3.2) gives, with probability $\ge 1-(\beta/b)^d$, that

$$D^N(n) = k(n,n_c)+h(n_c)-h(n) \le k(n,n_c)+U(h^*(n_c))-L(h^*(n))$$
$$= \eta(h^*(n_c))-\eta(h^*(n))+\theta(h^*(n_c))+\theta(h^*(n)) \le 2\theta(h^*(n)).$$

Thus $P[D^N(n) \ge 3\theta(h^*(n))] \le (\beta/b)^d$. As $h^*(n) \le d+N$ and θ is an increasing function, we get $P[D^N(n) \ge 3\theta(d+N)] \le (\beta/b)^d$. Now apply (A.3) of Lemma 5 to this to get that $P[D^N(n) \ge 2d] \le (\beta/b)^d$. By Lemma 3, n cannot be expanded if $D^N(n) < 2d$. Thus $P[exp(n)] \le P[D^N(n) \ge 2d] \le (\beta/b)^d$, as was to be shown. **Q.E.D.**

The next lemma is an extension to our environment of the formula for $E[Z(N)]$ that occurs in [Pearl, 1984, Chapter 6]. For q_{uidv} definition, see Section 2.

Lemma 7. Let w_{uid} be the number of nodes A* expands at level d of the off-track subtree T_{ui}. Then

[1] When one relation involving RV's is derived from another using simple algebraic transformations, we omit stating the "with probability 1" constraint.

$$\text{(A.7)}\quad E[w_{uid}] = \sum_{v=1}^{b^{d-1}} \prod_{m=1}^{d} q_{u,i,m,a(d,m,v)}$$

$$\text{(A.8)}\quad P[\exp(n_{uidv})] = \prod_{m=1}^{d} q_{u,i,m,a(d,m,v)}$$

where $a(d,d,v) = v$ and, in general, $n_{u,i,m,a(d,m,v)}$ is the level m ancestor in T_{ui} of n_{uidv}, $m = 1,2, ..., d$.

Proof. As A* finds a solution, n_i is expanded, so open(n_{ui11}) is certain. Consequently $P[\exp(n_{ui11})] = q_{ui11}$. As T_{ui} contains only one node at level 1, q_{ui11} is also the expected number of expanded nodes at level 1. This proves the lemma when $d = 1$.

Assume the lemma is true for the d - 1 case. We must show it is true for the d case. Now, since open(n_{uidv}) if and only if $\exp(n_{u,i,d-1,a(d,d-1,v)})$,

$$P[\exp(n_{uidv})] = P[\exp(n_{uidv}) \wedge \exp(n_{u,i,d-1,a(d,d-1,v)})]$$
$$= 0 \text{ if } P[\exp(n_{u,i,d-1,a(d,d-1,v)})] = 0;$$

otherwise,

$$P[\exp(n_{uidv})] = P[\exp(n_{uidv}) \mid \exp(n_{u,i,d-1,a(d,d-1,v)})] \cdot P[\exp(n_{u,i,d-1,a(d,d-1,v)})] ,$$
$$= P[\exp(n_{uidv}) \mid \text{open}(n_{uidv})] \cdot P[\exp(n_{u,i,d-1,a(d,d-1,v)})] .$$

Then, in either case,

$$P[\exp(n_{uidv})] = q_{uidv} \prod_{m=1}^{d-1} q_{u,i,m,a(d,m,v)}$$

by the induction hypothesis,

$$= \prod_{m=1}^{d} q_{u,i,m,a(d,m,v)} .$$

This establishes (A.8) for the inductive step, whence it is generally true. Now (A.7) follows for the inductive step by applying (A.8) : $E[w_{uid}]$ is simply the sum over all level d nodes of the probability that each is expanded. **Q.E.D.**

Proof of Theorem 2 and the Corollary. Let $q = 1 - \varepsilon$. We first note that

(1) $w(x) \le q(x-t) + t$

for some $t \ge 1$ and all $x \ge t$. This is because for x sufficiently large, say, $x \ge t \ge 1$, $w(x)$ both lies below the diagonal and has slope $< q$. Thus, for all $x \ge t$, $w(x)$ lies below the line through (t, t) whose slope is q, that is, (1) holds. Let $\varphi(x) = (1-q)^2x/(1+q)$ and let $N_0 = max\{N_1+t,[(1+q)t-1]/q\}$. We first show that, if $n = n_{utdv}$, $N > N_0$ and $d \le \varphi(N-t)$, u and v are arbitrary, then

(2) $P[\exp(n) \mid \text{open}(n_{utdv})] \ge \alpha/b$ whenever $P[\text{open}(n)] > 0$.

For all such N and d, $N > N_0 \ge [(1+q)t - 1]/q$, so $qN + N > (1+q)t - 1 + N$; that is,

$$N+1-2t > qt - t + N - qN = (1-q)(N-t)$$
$$= [(1+q)/(1-q)]\varphi(N-t) \ge [(1+q)/(1-q)]d.$$

Multiplying both sides by $1-q$ and separating N gives

(3) $N > q(N+1-2t+d)-1+2t+d = \{q[(N+d+1-t)-t]+t\}+\{d+t-1\}$
$\ge \{w(N+d+1-t)\} + \{d+t-1\}$ by (1),
$= w(h^*(n_{utdv})) + g(n_{utdv})$.

Notice that $h^*(n_{utdv}) = N+d+1-t \ge (N_1+t)+d+1-t > N_1$. Thus, if $P[\text{open}(n_{utdv})] > 0$ then

$\alpha/b \le P[h(n) \le w(h^*(n)) \mid \text{open}(n)]$ by our hypothesis
$= P[f(n) \le w(h^*(n)) + g(n) \mid \text{open}(n)]$
$\le P[f(n) < N \mid \text{open}(n)]$, by (3)
$\le P[f(n) < f(n_o) \mid \text{open}(n)]$, since $f(n_{N+1})=N$
$\le P[\exp(n) \mid \text{open}(n)]$, which establishes (2).

To complete the proof of the theorem apply Lemma 7:

$E[Z(N)] \ge E[Z_{ut}(N)]$, where $u \in \{1,...,b-1\}$,

$$\ge \sum_{d=1}^{\varphi(N-t)} \sum_{v=1}^{b^{d-1}} \prod_{m=1}^{d} q_{u,t,m,a(d,m,v)} , \text{ by (A.7)}$$

$$\ge \sum_{d=1}^{\varphi(N-t)} b^{d-1}\left(\frac{\alpha}{b}\right)^d , \text{ by (2)},$$

$$\ge [\alpha^{\varphi(N-t)}] / \alpha b ,$$

which is exponential in N. This proves Theorem 2.

The corollary follows from the theorem by noting that $P[h(n) \le w(h^*(n))] = 1$ implies that $P[(h(n) \le w(h^*(n))) \wedge \text{open}(n)] = P[\text{open}(n)]$ and so $P[h(n) \le w(h^*(n)) \mid \text{open}(n)] = 1$. **Q.E.D.**

References

[Bagchi and Sen, 1988] Bagchi, A., and Sen, A. Average-case analysis of heuristic search in tree-Like networks. In *Search in Artificial Intelligence,* ed. by L. Kanal and V. Kumar, New York: Springer-Verlag, pages 131-165,1988.

[Bramanti-Gregor and Davis, 1991] Bramanti-Gregor, A., and Davis, H.W. Learning admissible heuristics while solving problems. *Proceedings of the 12th International Joint Conference on Artificial Intelligence,* 1991.

[Chen, 1989] Chen, X. Generating and transforming heuristic functions to get desired properties. Master's Thesis, Wright State University, Department of Computer Science and Engineering, 1989.

[Chenoweth, 19901 Chenoweth, S.V. *An Analysis of the Asymptotic Performance of the A* Algorithm.* Ph.D. Dissertation, Wright State University, Department of Computer Science and Engineering, 1990. See also: New approaches for understanding the asymptotic complexity of A* tree searching, by S.V. Chenoweth and H.W. Davis and The mathematical modeling of heuristics, by H.W. Davis and S.V. Chenoweth, to appear in *Annals of Mathematics and Artificial Intelligence,* Volume F.

[Chenoweth and Davis, 1990] Chenwoeth, S.V., and Davis, H.W. Heuristics with polynomial growth patterns and the complexity of A*. In *Methodologies for Intelligent Systems, Selected Papers,* ed. by M.L. Emrich, *et al,* International Center for Application Technology, Knoxville, TN, pages 49-62,1990.

[Christensen and Korf, 1986] Christensen, J., and Korf, R.E. A united theory of heuristic evaluation functions and its application to learning. *Proceedings of the AAAI,* pages 148-152,1986.

[Garey and Johnson, 1979] Garey, M.R., and Johnson, D.S. *Computers and Intractability, a Guide to the Theory of NP-Completeness.* New York: W.H. Freeman and Company, 1979.

[Gaschnig, 1979] Gaschnig, J. *Performance Measurement and Analysis of Certain Search Algorithms.* Ph.D. Dissertation, Carnegie-Mellon University, 1979.

[Mostow and Prieditis, 1989] Mostow, J., and Prieditis, A.E. Discovering admissible heuristics by abstracting and optimizing: A transformational approach. *Proceedings of the 11th International Joint Conference on Artificial Intelligence,* pp. 701-707,1989.

[Pearl, 1984] Pearl, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Reading, Massachusetts: Addison-Wesley Publishing Company, 1984.

[Politowski, 1986] Politowski, G. *On the Construction of Heuristic Functions.* Ph.D. Thesis, U. C. Santa Cruz, 1986.

[Rendell, 1983] Rendell, L. A new basis for state-space learning systems and a successful implementation. *Artificial Intelligence* 20, pp. 369-392,1983.