THE STATISTICAL INFERENCE METHOD IN HEURISTIC SEARCH TECHNIQUES

Zhang, Ling
Dept. of Mathematics
Anqing Teachers' College
Anqing, Anhua, China

Zhang, Bo
Dept. of Computer Engineering & Science
Tsinghua University
Beijing, China

ABSTRACT

In this paper we present a new heuristic searching algorithm by introducing statistical inference method on the basis of algorithm A*. It's called algorithm SA*. The following results have been proved.

(1) Algorithm SA* is superior to algorithm A*.

(2) The mean complexity of SA* is CN^2, but in some case A* exhibits exponential complexity (e^N).

(3) In a (N,d,F)-game tree, the mean complexity of SA* is CN^2, but the complexity of other known game-searching algorithm ($\alpha-\beta$, SSS* etc.) is at least d^N.

(4) The maximal storage-space required by SA* is C_1N.

This shows that under a given significance level SA* is superior to other known algorithm (e.g. A*, B*, $\alpha-\beta$, SSS* etc.).

1. INTRODUCTION

The heuristic search theory has been investigated by many researchers [1]-[9]. All results obtained can't completely avoid the exponential explosion of searching complexity. We improve it by applying statistic inference method (s. i. m.) to heuristic search. The results we obtain are that the mean complexity of SA* is CN^2 and the maximal storage-space is CN.

2. ALGORITHM SA* IN TREE G

2.1. Statistic a(n)

For simplicity, we assume the following search space: A uniform m-ary tree G has an initial node S_0(root) and a unique goal node S_N at depth N. Let $l=(s_0,s_1,\dots,s_N)$ be the shortest-path from S_0 to S_N. The subtrees having root S_i are called T_i-type subtrees. They are $T_i^1,T_i^2,\dots,T_i^m$, i=0,1,... Assume that T is an T_i-subtree. If n is a node of T, the generation(depth) of the node is n, and T doesn't contain l. We have

$$g^*(n)=n,\ h^*(n)=(N-i)+(n-i),$$

$$f^*(n)=n+N-i+n-i=N+2(n-i).$$

(We use the same symbols as thoes used in most books, e.g. [4]).

Let $a^*(n)=\frac{f^*(n)-N}{2n}$, $a(n)=\frac{f(n)-N}{2n}$.

h(n) is a heuristic estimate of h*(n), so a(n) is an estimate of a*(n). While $n\in l$, f*(n)=N, $a^*(n)=\frac{f^*(n)-N}{2n}=0$. While $n\notin l$, $a^*(n)=\frac{f^*(n)-N}{2n}=1-\frac{i}{n}$. Given i, We have $a^*(n)\underset{n\to\infty}{=}1$. In a word, for any node the statistic a(n) can be computed from h(n). (If N is unknown, we may replace a(n) with some other statistic. For example, let the number of all nodes being expanded be k(n), the number of all nodes being expanded in T_0^i-subtree be $k_i(n)$. We replace a(n) with $b(n)=\frac{k_i(n)}{k(n)}$ as the statistic of T_0^i-subtree, and so on.)

<u>Hypothesis I:</u> Assume {a(n)} is an independent and identically distribution random variable. The mean of a(n) in the solution path l is μ_0. The mean of a(n) off l is μ_1. $\mu_1>\mu_0$. Under this hypothesis, when h(n) of each node is computed using A*, an a(n) is obtained. This {a(n)} forms a random sample, using testing statistical hypotheses(t.s.h.)[10][11], we exercise the statistical inference method(s.i.m.) over it. Under a given significance level of the test, whether a subtree contain l is decided. If not, the subtree is pruned off. Otherwise, algorithm A* and t.s.h. will be continued until the goal node is found.

<u>The sampling of statistics in subtree T</u>

Let T be a subtree, a_1 be the statistic of the root in T. Assume T is expanded by A*, and in some stage the corresponding statistics $\{a_1,a_2,\dots,a_k\}$ have been obtained. We say "observing T is continued." It means expanding node p at which f(n) is minimal among all nodes not being expanded in T. (If there exist several such nodes, choose one which has maximal generation. If there still exist several nodes, choose any one at your option.) Thus we obtain m successors of p and corresponding a(n)'s. Let a_{k+1} be the minimal value among these a(n)'s, then a_{k+1} is referred to a new observed value during the observation of T. we say "exercising some t.s.h. over T." It means exercising some t.s.h. over the statistics $\{a_k\}$ corresponding to T.

2.2. Algorithm SA*

Given a testing hypotheses method S. Applying this method to A*, We obtain algorithm SA*:

Step 1: From initial state S_0, m T_0-type subtrees are expanded. A set U_1 is composed by these subtrees. Let t←1, go to Step 2.

Step 2: Exercise the statistical inference over U_t.

(1) If U_t is an empty set, stop.

(2) If there is only one T_i-type subtree T in

U_t, expand the (i+1)-th generation nodes in T and obtain m T_{i+1}-type subtrees. Merging the T_{i+1}-type subtrees into U_t, obtain U_{t+1}. Let $t \leftarrow t+1$, go to Step 2.

(3) If there is more than one subtree, expand node p at which f(n) is the minimum among all nodes not being expanded in all subtrees(If there still exist several nodes, choose any one.)

(3.1) If there exists a goal node among the successors of p, stop.

(3.2) Assume p is in subtree T', observing T' and exercising the t.s.h. S over it are continued. If the hypothesis is rejected, let $U_{t+1} \leftarrow U_t - T'$, $t \leftarrow t+1$, go to Step 2. Otherwise, let $U_{t+1} \leftarrow U_t - T' + T''$ (T" is a subtree formed by adding the successors of p to T'), $t \leftarrow t+1$, go to Step 2.

<u>Proposition 1:</u> SA* is superior to A*. Assume A* and SA* both are directed by the same h(n). Finding an optimal solution path by SA*, every node expanded by SA* is also expanded by A*.

Proof: SA* is an algorithm formed by only adding an additional pruning subtrees stage to A*, so the nodes expanded by SA* are not more than the nodes expanded by A*.

It must be pointed out that the results obtained by SA* have some error probabilities, because of the application of s.i.m. . We'll discuss later on.

<u>SPRT testing hypotheses method in algorithm SA* (denited by SPA*)</u>

SPRT (Sequential Probability Ratio Test) was decribed in many books (e.g. [10]). We use SPRT as testing hypothese here. Let {a(n)} be {x_n} , having an $N(\mu, \sigma^2)$ distribution. Given a significance level (α , β) and two simple hypotheses H_0: $\mu=\mu_0$. H_1: $\mu=\mu_1$, $\mu_1 \neq \mu_0$.

Let
$$z \triangleq \log \frac{f(x;\mu_1)}{f(x;\mu_0)} = \frac{\mu_1-\mu_0}{\sigma}x + \frac{1}{2}\frac{\mu_0^2-\mu_1^2}{\sigma^2}$$
$$S_n \triangleq \sum_1^n z_i = \frac{\mu_1-\mu_0}{\sigma}\sum_1^n x_i + \frac{n}{2}\frac{\mu_0^2-\mu_1^2}{\sigma^2}$$
$$A \triangleq \frac{1-\beta}{\alpha}, \quad B \triangleq \frac{\beta}{1-\alpha}, \quad a \triangleq \log A, \ b \triangleq \log B.$$
Where $f(x;\mu) = \frac{1}{\sqrt{2\pi}} exp\{-\frac{1}{2}(x-\mu)^2\}$.

The stopping rules of SPRT are as follows:

If $\sum_1^n x_i \geq \frac{\sigma a}{\mu_1-\mu_0} + n\frac{\mu_1+\mu_0}{2\sigma}$

Hypothesis H_0 is rejected.

If $\sum_1^n x_i \leq \frac{\sigma b}{\mu_1-\mu_0} + n\frac{\mu_1+\mu_0}{2\sigma}$

Hypothesis H_0 is accepted.

Otherwise, observing x_{n+1} is continued.

Because parameters σ, μ_1, μ_0 are unknown, we usually use $S_n = \frac{1}{n-1}\sum_1^n (x_i-\bar{x})^2$ to estimate σ , where $\bar{x}=\frac{1}{n}\sum_1^n x_i$. Let a^* be the minimum value of a(n)'s among all k-th generation nodes which a(n)'s have been computed in G. Let the mean of {a^*} be the estimate of μ_0, and the mean of all a(n)'s, which have been computed in G, be μ_1.

If in SA* as testing hypotheses S, SPRT is exercised over m T_i-type subtrees, using a level $(\alpha^{i+1}, \frac{\alpha^{i+1}}{m-1})$, i=0,1,..., we define this SA* as SPA* under level (α , $\frac{\alpha}{m-1}$), denoted by SPA* for short.

2.3. The Mean Complexity of SPA*

From the approximation of the mean of stopping variable (sample size) N in SPRT [10] , if N has an $N(\mu, \sigma^2)$ distribution, level=(α , β) $A=\frac{1-\beta}{\alpha}$, $B=\frac{\beta}{1-\alpha}$, $\beta=\frac{\alpha}{m-1}$, we have
$$E_{\mu_0}(N) \approx \frac{(\alpha \log\frac{1-\beta}{\alpha} + (1-\alpha)\log\frac{\beta}{1-\alpha})\sigma^2}{-\frac{1}{2}(\mu_1-\mu_0)^2} \sim \frac{2\sigma^2}{(\mu_1-\mu_0)^2}|\log\alpha|$$
$$E_{\mu_1}(N) \approx \frac{((1-\beta)\log\frac{1-\beta}{\alpha} + \beta\log\frac{\beta}{1-\alpha})\sigma^2}{\frac{1}{2}(\mu_1-\mu_0)^2} \sim \frac{2\sigma^2}{(\mu_1-\mu_0)^2}|\log\alpha|$$

<u>Lemma:</u> The mean complexity (asymptotic) for deciding m T_i-type subtrees in SPA* is

$\sim mb|\log\alpha|\cdot(i+1)$, where $b=\frac{2\sigma^2}{(\mu_1-\mu_0)^2}$

Proof: From SPRT we know that deciding m T_i-type subtrees under level $(\alpha^{i+1}, \frac{\alpha^{i+1}}{m-1})$, the mean complexity (asymptotic) is $\sim mb|\log\alpha^{i+1}| = mb|\log\alpha|\cdot(i+1)$.

<u>Theorem 1:</u> Let $\alpha=\min(\frac{\alpha_0}{1+\alpha_0}, \frac{\beta_0}{1+\beta_0})$. Using SPA*, under level (α , $\frac{\alpha}{m-1}$) the mean complexity of finding an optimal solution path in G is $\sim CN^2$, where $C=\frac{mb|\log\alpha|}{2}$. The error probabilities of Type I $P_1 \leq \alpha_0$. The error probabilities of Type II $P_2 \leq \beta_0$.

Proof: Deciding m T_i-type subtrees there are m-1 subtrees not containing l but one. Due to the probability $P_2=\frac{\alpha^{i+1}}{m-1}$, $(m-1)\frac{\alpha^{i+1}}{m-1}=\alpha^{i+1}$ subtrees not containing l are left. Because of $P_1=\alpha^{i+1}$, $(1-\alpha^{i+1})$ subtrees containing l are left. Totally $\alpha^{i+1}+(1-\alpha^{i+1})=1$ subtree is left. The mean complexity for deciding one T_i-type subtree is $\sim b|\log\alpha|(i+1)$ (lemma). Thus using SPA*, the mean complexity of finding an optimal path is
$$\sim \sum_0^{N-1} mb|\log\alpha|(i+1) = mb|\log\alpha|\frac{N(N+1)}{2}$$
$$\sim \frac{mb|\log\alpha|}{2}N^2 = CN^2.$$

The probability $P_1(P_2)$ is as follows:
Deciding m T_0-type subtrees $P_1=\alpha$, In general deciding m T_i-type subtrees $P_1 \leq \alpha^{i+1}$. Totally
$$P_1 \leq \sum_{i=1}^{N}\alpha^i = \alpha\sum_0^{N-1}\alpha^i = \alpha\frac{1-\alpha^N}{1-\alpha} \leq \frac{\alpha}{1-\alpha} \leq \alpha_0.$$
Analogously, $P_2 \leq \beta_0$.

<u>Corollary:</u> Using SPA*, under level (α , $\frac{\alpha}{m-1}$) the maximal storage-space $\leq mb|\log\alpha| N=C_1N$.

Proof: Deciding m T_i-type subtrees, all information about these subtrees is storaged at most, That is, $\sim mb|\log\alpha|N$.

Note: Due to the process of pruning subtrees, the storage-space required by SPA* is not more than A*.

2.4. Comparson to recent results

Pearl[2] defined an estimate h(n) of $\Phi(n)$-type error and proved that when $\Phi(n)=n$ the complexity of A* is $O(e^N)$. We'll prove that in the same case the complexity of SPA* is CN^2.

<u>Theorem 2:</u> Assume h(n) is an admissible estimate, having $\Phi(n)=n$ type error. $P(|h^*-h|<h^*)>0$. Then the mean complexity of SPA* is CN^2.

Proof: If $\mu_1>\mu_0$ is proved (the proof is omitted). according to Theorem 1, we obtain Theorem 2.

<u>Corollary:</u> Assume h(n) is an admissible estimate having $\Phi(n)$-type error, $\overline{\lim}\frac{\Phi(n)}{n}_{n\to\infty} \leq 1$ $P(|h^*-h| \leq \Phi(h^*))>0$, then the mean complexity

of SPA* directed by $h(n)$ is CN^2.

Note: In SPA*, σ, μ_1 and μ_0 are unknown. They are replaced with their estimators. This will cause some error. For elimilating this disadvantage, we may use t-test as testing hypotheses S in SA*. The searching complexity is a little more than SPA*. But we may prove that Theorem 1 also holds and the mean complexity is $\sim CN^2$.

Theorem 2 also holds for $\{x_n\}$ having some sorts of distribution except $N(\mu, \sigma^2)$.

3. ALGORITHM SA* IN GENERAL GRAPH

Assume $h(n)$ is an admissible estimate, then using $b(n)=\frac{k_i(n)}{k(n)}$ (see 2.1) as the statistic of T_0^i-subtree (subgraph), and so on, we may obtain algorithm SA* for a general graph.

4. ALGORITHM SA* IN GAME TREE

We'll apply SA* to game-searching. A standard 2n-level game tree of degree m is indicated by (n,m,F)-tree where $F(v)$ is a distribution function of terminal value (the symbols used are the same as in [3]). In [1],[5],[6],[7], it has been proved that any known algorithm which evaluates a (n,m,F)-tree must evaluate at least m^n terminal positions. We'll apply SA* to a (N,m,F)-tree, and conclude that the mean complexity of SA* in game-searching is C_1N^2.

<u>The Sampling of Statistics in Game-Tree</u>

In a game-tree, the value $f^*(n)$ of each node is obtained by searching backward from terminal values (for example, from the standpoint of Max). Assume that for each node an estimate $f(n)$ of $f^*(n)$ can be computed. Let statistic $a(n)$ be

$$a(n)=\begin{cases}\max\ (f(n_i),\ i=1,2,\ldots,m) & n_i \text{ is the successor of } n,\ n \text{ is an even node}\\ \min\ (f(n_i),\ i=1,2,\ldots,m) & n_i \text{ is the successor of } n,\ n \text{ is an odd node.}\end{cases}$$

Let T be an T_i-type subtree. The sampling of its statistics is as follows:

Let $a_0=f(S_0)$, S_0 is the root of T.

Expanding S_0, Assume $a_1=f(S_1)=\max(f(n_i))$, n_i, S_1 are the successors of S_0.

Expanding S_1, Assume $a_2=f(S_2)=\min(f(n_i))$, n_i, S_2 are the successors of S_1.

In general, we obtain $\{a_0, a_1, \ldots, a_k\}$. If k=2j, let

$a_{k+1}=f(S_{k+1})=\max_{1\le i\le m}(f(n_i))$, n_i, S_{k+1} are successors of S_k.

If k=2j+1, let

$a_{k+1}=f(S_{k+1})=\min_{1\le i\le m}(f(n_i))$, n_i, S_{k+1} are successors of S_k.

Assume the statistic $\{a(n)\}$ satisfies the Hypothesis I, Similar to tree search we apply SA* to game-searching, and the following theorem holds.

<u>Theorem 1':</u> In an (n,m,F)-game tree, the mean complexity of SPA* under the level $(\alpha, \frac{\alpha}{m-1})$ is $\sim C_1N^2$, $C_1=\frac{mb|\log\alpha|}{2}\times 4=2mb|\log\alpha|$.
The $P_1(P_2)$ is $\alpha_0(\beta_0)$. where $\alpha=\min(\frac{\alpha_0}{1+\alpha_0}, \frac{\beta_0}{1+\beta_0})$.

<u>Corollary:</u> The maximal storage-space of SPA* in game-tree searching is C_2N, $C_2=mb|\log\alpha|$.

Note: The storage-space required by SSS* is at least m^n [7].

(The proof of Theorem 1' is omitted).

REFERENCES

[1] Pearl, J.,"Heuristic Search Theory: Survey of Recent Results." Proceedings of 7-th IJCAI, 1981, pp.554-562.
[2] Pearl, J.,"Probabilitic Analysis of the complexity of A*." A.I., Vol. 15, No. 3, 1980, pp. 241-254.
[3] Pearl, J., "Asymptotic Properties of Minimax Tree and Game-searching Procedures." A.I., Vol.14, No. 2, 1980, pp.113-138.
[4] Nilsson, N. J.,Principles of artificial Intelligence, Palo Alto, CA: Tioga Publishing Company, 1980.
[5] Baudet, G. M.,"On the Branching Factor of the Alpha-Beta Pruning Algorithm." A.I., Vol. 10, 1978, pp.173-199.
[6] Knuth, D. E. and R. W. Moore, "An Analysis of Alpha-Beta Ptuning." A.I.,Vol. 6, 1975, pp. 293-326.
[7] Stockman, G. C.,"Is a Minimax Algorithm Beter Than Alpha-Beta?" A.I., Vol. 12, 1979, pp.179-196.
[8] Kanal, L. and Kumar V., "Branch & Bound Formulation for Sequential and Parallel Game Tree Searching: Preliminary Results." Proceedings of 7-th IJCAI, 1981, pp.569-571.
[9] Berliner, H., "The B* Tree Search Algorithm: A Best-first Proof Procedure." Reading in Artificial Intelligence, edited by B.L. Webber, 1981, pp.79-87.
[10] Zacks, S., The Theory of Statistic Inference, John Wiley & Sons, Inc., New York, 1971.
[11] Degoot, M. H., Probability and Statistics, Menlo Park, California, 1975.