A WEIGHTED TECHNIQUE IN HEURISTIC SEARCH

Zhang, Bo

Dept. of Computer Science
Tsinghua University
Beijing, China

Zhang, Ling

Dept. of Mathematics
Anqing Teachers' College
Anqing, Anhui, China

ABSTRACT

As shown in [1], we examine search as a statistic sampling process. Based on some statistical inference method the probability that a subtree in search tree contains the goal can be decided. Thus some weight is in tentionaly added to the evaluation function of those nodes which are unlikely in the solution path so that the search will concentrate on the most promising path. It results in a new weighted algorithm-WSA.

Tn a uniform m-ary tree, we show that a goal can be found by WSA in the polynomial time, although the computational complexity of A (or A*) may be 0(e) for searching the same space. Where N is the depth at which the goal is located.

INTRODUCTION

Weighted techniques in heuristic search have been investigated by several researchers (e.g., see [2]- (4)). Although thoes methods made the aearch more efficiency, the improvement is rather limitted because weights are usually added to all nodes undicriminally, for example, in [2] the? same weight X_0 is applied to each node.

The alternative weighted technique presented here is the following. According to decisions made by some statistic inference method during A (or A*) search a weight will Only be added to the evaluation of some nodes which are unlikely in the solution path. It results in a new weighted technique that will provide better results.

A NEW WEIGHTED METHOD

As shown in (1], under certain conditions we examine search as a statistic sampling process so that statistic inference method can be used during the search. Assume the Wald sequential probability radio test (SPRT) is used as a testing hypotheses. In some searching stage, if the hypothesis that some subtree T contains solution path is rejected, from (1) it's known that subtree T contains the goal with lower probability. Rather than pruning T (as in[1]) a fixed weight w is added to the evaluation function of nodes in T, i.e., f (n)=f(n)+w. If the hypothesis that the subtree T' contains the goal is accepted, the same weight is added to all nodes in the brother-subtrees of T^1 which roots are the brothers of the root of TV. If no decision can be made the search process is continued as in A search. Thus the search will concentrate on the subtrees which contain the goal with higher probability due to the weighting. This new algorithm is called the weighted SA search—WSA.

THE COMPLEXITY OF WSA

Assume the search space is a uniform m-ary tree, the SPRT is used as the testing hypotheses and the given significance level is (α, β), $\alpha + \beta = b$. The complexity of an algorithm is defined as the expected number of nodes expanded by the algorithm when a goal is found. We have proved the following theorems (the proof is presented in the Appendix).

Theorem 1: Assume $P(A) \sim O(e^{CN})$ and $C > 0$ C is a known constant, N is the depth at which the goal is located, P(A) is the complexity of algorithm A when it searches the same space. Using the weighted f u n $f_1(n)=f(n)+w_0$, e r e $w_0 = \frac{1}{?C} \ln \frac{1-b}{b}$ (the optimal weight) the complexity of algorithm WSA is

$$P(WSA) \sim O(N).$$

Theorem ?: If $P(A) \sim O(N^a)$, $a > 1$, using the weighted function $f_1(n) = \lambda_0 f(n)$ where $\lambda_0 = {}^{2a}\sqrt{\frac{1-b}{b}}$, then the complexity of WSA is

$$P(WSA) \sim O(N).$$

Obviously, the new weighted method can improve the computational complexity greatly.

Generally, $P(A)$ is either $\sim o(e^{CN})$ or $\sim (N^a)$, and it's unknown. When an arbitrary weight $w \neq w_0$ is used in WSA search, how about its complexity?

Theorem *3:* If $P(A) \sim O(e^{CN})$, $C > 0$ and C is unknown, using the weighted function $f_1(n)=f(n)+w$, $w \neq w_0$, and a monotonously decreased significance level $b_i = \frac{b}{i^2}$ is used for testing T_i-subtrees, then

$$P(WSA) \sim O(N.\ln N)$$

Theorem 4: If $P(A) \sim O(N^a)$, $a > 1$, using the same weighted function $f_1(n)=f(n)+w$, the complexity of WSA search remains in the polynomial time.

ALGORITHM WSA

Given an evaluation function (statistic) f(n) and a testing hypotheses method, the WSA search procedure as follows:

(1) Create a list called OPEN. Expand initial node 3 , generating its m successors, put them on OPEN.

Create a list CLOSED, It's initially empty. Create a list ROOTS. Put m successors of S on it, The node in ROOTS is called a.

(2) LOOP: If OPEN is empty, exit with failure (it's impossible that OPEN is empty when there exists a goal in the search tree).

(3) Select the first node on OPEN, remove it from OPEN, and put it on CLOSED. Call this node n.

(4) If n is a goal, exit successfully with the solution obtained by tracing a path along the pointers which are established in step 5.

(5) Expand node n, generating its m successors, Put them on OPEN.Establish a pointer to n from these successors.

(6) If some subtree T(a) is accepted according to some testing hypothese, add a weight w to the statistics of all nodes in the brother-subtree of T(a), remove the root nodes of T(a) and its brother-subtrees from ROOTS and put their successors on ROOTS.

(7) Reorder the list OPEN according to statistics. Go LOOP.

CONCLUSIONS

A new weighted technique is incorporated in A (or A*) search. While the algorithm A searches a space using evaluation function f(n), some weight w is added to f(n) (f (n)=f(n)+w) of the nodes which are unlikely in "the solution path according to decisions made by some statistic inference method. Thus the paths that contain the goal with higher probability will be expanded more due to the weighting.

In a uniform m-ary tree, we show that a goal can be found by WSA in the polynomial time, although the complexity of A may be 0(e) for searching the same space.

Both algorithm A (3] and *SA* are special cases of this more general algorithm WSA. Note that when w =0 algorithm WSA is idential to A search. While w = +oo algorithm WSA degenerates into SA search.

APPENDIX

The proof of Theorem 1:

For simplicity, we assume the search space is a uniform 2-ary tree, m-2, in the following discussion. There is no loss of generality in assuming that $P(A)=e^{CN}$

If a statistic decision is made in some search stage,a weight w is added to evaluation of nodes of the rejected subtrees. A subtree is called a completely weighted if all its subtrees have been decided to be rejected or accepted. The subtree shown in Fig.1 is completely weighted (Where the rejected subtrees are marked with sign "X").

Obviously, a completely weighted subtree has more expanded nodes than the incompletely weighted one. Thus if an upper estimate of the mean complexity of the completely weighted subtree is computed, it certainly is an upper estimate of the mean complexity in general cases.

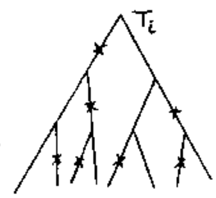

Fig. 1.

We now discuss this upper estimate.

Let P_d be a set of nodes at depth d. Given $n \in P_d$. From initial node S_o to n^d there exists a unique path consisting of d arcs. Among those arcs if there are $i (o \leqslant i \leqslant d)$ arcs marked by "X", node n is referred to as an i-type node or i-node.

So P_d can be divided into the following subsets:

0-node: there is only one, 1-node: $C_d^1=d, \ldots,$ i-node: $C_d^i, \ldots,$ d-node: $C_d^d=1$.

In considering the complexity for finding a goal, we first ignore the cost of the statistic inference. Assume that the goal belongs in 0-node so that its evaluation is $f_g(n)=N$. From algorithm A, it's known that every node which $f(n)< N$ must be expanded in the searching process.

If node n is an i-node, its evaluation function is $f_1(n)=f(n)+iw$.All nodes which evaluation satisfy the following inequation will be expanded.

$$f_1(n)=f(n)+iw \quad N, \quad \text{i.e.,} \quad f(n)< N-iw.$$

Using evaluation function f(n) the complexity of A search is known to be $P(A)=e^{CN}$, thus the complexity corresponding to the evaluation function $f_1(n)=f(n)+iw$ is $e^{C(N-iw)}$. The mean complexity of each i-node (the possibility that an i-node may be expanded) is $\frac{e^{C(N-iw)}}{2^{N+1}}=e^{-Ciw}\frac{e^{CN}}{2^{N+1}}$

The mean complexity for finding a goal at depth N is at least N. Thus the mean complexity of each i-node is

$$\max\left(\frac{e^{C(N-iw)}}{2^{N+1}}, \frac{N}{2^{N+1}}\right) \leqslant \frac{1}{2^{N+1}}(e^{C(N-iw)}+N).$$

When the goal is an 0-node, the upper estimate of the mean complexity for computing all d-th depth nodes is the following:

$$\frac{1}{2^{N+1}}\sum_{0}^{d} C_d^i (e^{C(N-iw)}+N)=\frac{e^{CN}}{2^{N+1}}(1+e^{-Cw})^d+\frac{N}{2^{N-d+1}}.$$

On the other hand, if $\alpha+\beta=b$ is a constant, from [1] for making the statistic inference of a node, the mean computational cost of SPRT is a constant Q. When the goal is an 0-node, accounting for this cost, the mean complexity is

$$P_o(WSA) \leqslant Q\left(\frac{e^{CN}}{2^{N+1}}(1+e^{-Cw})^d+\frac{N}{2^{N-d+1}}\right).$$

Similarly, if the goal belongs in i-node, the mean complexity for computing all d-th nodes is

$$P_i(WSA) \leqslant Q\left(\frac{e^{CN}+e^{iwC}}{2^{N+1}}(1+e^{-Cw})^d+\frac{N}{2^{N-d+1}}\right).$$

From algorithm SA [1], the goal falls into an i-node with probability $(1-b)^{N-i}b^i$, if the given level is (α, β), $\alpha+\beta=b$. At depth N there are C_N^i i-nodes, so the probability that the goal belongs in i-node is

$$C_N^i(1-b)^{N-i}b^i, \qquad i=0,1,\ldots,N-1.$$

Accounting for all possible cases of the goal node, the mean complexity for computing all d-th depth nodes is

$$\sum_{i=0}^{N-1} C_N^i(1-b)^{N-i}b^i P_i(WSA)$$
$$\leqslant \frac{1}{2^{N+1}}\left[e^{CN}\frac{(1+e^{-Cw})^d(1-b+be^{Cw})^N}{2}\right]+\frac{NQ}{2^{N-d+1}}$$

Let $F(w)=(1-e^{-Cw}).(1-b+be^{Cw})$ (1)

$F(w)$ attains its minimum for a value of w given by

$$w_0=\frac{1}{2C}\ln\frac{1-b}{b}$$
$$F(w_0)=1+2\sqrt{b(1-b)}\ .$$

Under the optimal weight, the upper bound of the mean complexity of algorithm WSA is

$$P(WSA)\leqslant Q\left[\frac{e^{CN}}{2^{N+1}}\sum_{0}^{N}(1+2\sqrt{b(1-b)}\ )^i+\sum_{0}^{N}\frac{\cdots}{\cdots}\right]$$

$$\sim O(1)\ \frac{e^{CN}}{2\sqrt{b(1-b)}}\left(\frac{1+2\sqrt{b(1-b)}}{2}\right)^{N+1}. \qquad (2)$$

Assume $0<C<\ln 2$, we shall show that there exists an b_0 such that $C<\ln\left(\frac{2}{1+2\sqrt{b_0(1-b_0)}}\right)$.

Let $C=\ln f<\ln 2$, i.e., $f<2$.

From $C<\ln\left(\frac{2}{1+2\sqrt{b_0(1-b_0)}}\right)$ (3)

obtain

$$\frac{f}{2}<\frac{1}{1+2\sqrt{b_0(1-b_0)}} \text{ or } 2\sqrt{b_0(1-b_0)}<\frac{2}{f}-1>0(\because f<2)$$

Let $\frac{1}{2}(\frac{2}{f}-1)=h$, $b_0(1-b_0)=h^2$.

If $4h^2>1$, given any $0<b_0<1$, Form (3) holds.

If $4h^2-1\leqslant 0$, as long as $0<b_0<\frac{1-\sqrt{1-4h^2}}{2}$, Form (3) holds

Substitute (3) into (2), have

$$P(WSA)\leqslant Q\left\{\frac{1+2\sqrt{b_0(1-b_0)}}{4\sqrt{b_0(1-b_0)}}e^{\left(C+\ln\frac{1+2\sqrt{b_0(1-b_0)}}{2}\right)N}+O(N)\right\}\sim O(N) \qquad (4)$$

Similarly, Theorem 2 can be proved.

The proof of Theorem 3:

We discuss $P(A)\sim O(e^{CN})$ and C is unknown. So the optimal weight w_0 is also unknown.

Assume $w=w_0+\Delta w$,

$$e^{-Cw}=e^{-Cw_0}.e^{-C\Delta w}$$

Let $u=e^{-C\Delta w}$

Thus $(1-e^{-Cw})(1-b+be^{Cw})=(1+u\sqrt{\frac{b}{1-b}})(1-b+\frac{1}{u}\sqrt{b(1-b)})$
$$=1+(u+\frac{1}{u})\sqrt{b(1-b)}$$

$$P(WSA)\leqslant Q\left\{\frac{1+(u+\frac{1}{u})\sqrt{b(1-b)}}{2(u+\frac{1}{u})\sqrt{b(1-b)}}e^{\left(C+\ln\frac{1+(u+\frac{1}{u})\sqrt{b(1-b)}}{2}\right)N}+N\right\} \qquad (5)$$

In order to obtain $P(WSA)\sim O(N)$, value b must be so small that

$$C\leqslant\ln\frac{2}{1+(u+\frac{1}{u})\sqrt{b(1-b)}} \qquad (6)$$

But C is unknown, it is unable to select a fixed b so that Form (6) is satisfied. If we use a monotonously deceased value b_i for testing T_i-subtrees as we did in [1][6]. Due to the gradual decrement of $b_i=\alpha_i+\beta_i$, in some search stage, Form (6) must hold.

For example, if $b_i=\frac{b}{i^2}$, b is a constant, for making the statistic inference(SPRT) once, the mean complexity Q is $O(\ln N)$ at most. From (5) obtain $P(WSA)\sim O(N.\ln N)$.

The proof of Theorem 4 is omitted.

REFERENCES

[1] Zhang, Bo and Zhang, Ling, "A New Heuristic Search Technique—Algorithm SA". IEEE Trans. on PAMI, VOL. PAMI-7, NO. 1 (1985) 103-107.
[2] Pearl, J., "Some recent Results in Heuristic Search Theory". IEEE Trans, on PAMI, VOL. PAMI-6, NO. 1, (1984) 1-12.
[3]Nilsson, N.J,, Principles of Artificial Intelligence, Palo Alto CA: Tioga Publishing Company, 1980.
[4] R. Field, K. Mohyeldin-Said & I. Pohl, " An Investigation of Dynamic Weighting in Heuristic Search". In Proc. ECAI-84, Pisa, Italy, Sept. 19 1984, pp.277-278.
(5)0. Zacks, The Theory of Statistic Inference, John Wiley & Sons, Inc., New York, 1971.
[6]Zhang, Ling & Zhang, Bo, "The Successive SA* Search and Its Computational Complexity". In Proc. ECAI-84, Pisa, Italy, Sept. 1984, pp.249-258.