An Analysis of Breadth-First Beam Search using Uniform Cost Trees

Sandip Aine and P. P. Chakrabarti
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB 721302, India

Abstract

In this paper, we present a probabilistic analysis of breadth-first beam search on trees. The work is divided into two parts. First, we introduce a *ranking* model to characterize the nodes in a search space and discuss the condition for provable optimality of beam search in terms of the ranking model. We also formulate a method to compute the probability of optimal termination of beam search for a specific beam width. Next, we analyze the node expansion properties of beam search using a uniform cost tree model and establish a relationship between the probability of optimal termination and the search space configuration. The analysis explains certain interesting characteristics of beam search including the possibility of *non-monotonic* progress with increasing search efforts.

1 Introduction

Beam search (Rich & Knight 1991; Zhang 1999) reduces the memory/time requirement of breadth-first (or best-first) search by focusing the search effort into the most promising parts of the search space. In general, beam search builds its search frontier in a breadth-first manner. At each level, it generates all successors of the states at the current level and sorts them in order of increasing heuristic values. However, it only stores a predetermined number (termed as the beam width) of states at each level, and the remaining nodes are pruned off permanently. By restricting the node expansions, the memory/time complexity of beam search becomes linear in the depth of search. However, as beam search enforces inadmissible and permanent pruning (beyond a limited choice), the search is not guaranteed to be complete or optimal[1]. In recent years, beam search and its variants have attained widespread application in various domains (Zhou & Hansen 2005; Habenicht & Mönch 2002; Tillmann & Ney 2003), successfully solving large scale problems under resource limitations.

Despite the wide and successful use, nothing much about the characteristics of beam search has been discussed in recent literature, apart from its obvious limitations of *incompleteness* and *non-optimality*. On the other hand, there are a number of questions regarding beam search, like,

1. *Under which conditions is beam search guaranteed to find the optimal solution?*,
2. *What is probability of a successful termination of beam search for a particular search space and beam width specification?*,
3. *Is the quality of results guaranteed to improve with an increase in beam width?*, etc,

which still remain unanswered and requires further investigation. In this work, we try to study some these questions from an analytical point of view.

For our analysis, we use the concept of 'rank' of a node in a particular level of a search tree (Aine, Chakrabarti, & Kumar 2009). We define a set of independent and conditional rank terminologies and derive the necessary and sufficient condition for optimal termination of beam search. We also present a generic probabilistic model for computing the probability of convergence[2] of beam search for a given beam width.

In the next phase, we analyze the node expansion characteristics of beam search in terms of the uniform cost search tree model (Pearl 1984; Chenoweth & Davis 1991) and present a methodology to compute the convergence probability directly from the search space configuration and the heuristic error distributions. We use the proposed model to investigate the performance of beam search across different search space configurations and heuristic error bounds. The results reveal that the expected quality of beam search can be non-monotonic with beam width[3]. We discuss the reasons behind such non-monotonic behavior and identify the condition under which beam search is guaranteed to have monotonic progress.

In a summary, this paper presents an analytical model to characterize the convergence of beam search along with a methodology to compute the convergence probability conditioned by the beam width cut-off. To the best of our knowledge, this is the first attempt to study the properties of beam

[1] It may be noted there are variants of beam search, such as beam stack search (Zhou & Hansen 2005), which overcome these drawbacks of incompleteness and non-optimality.

[2] We use the term convergence to denote the expansion of the optimal cost goal node. Convergence probability denotes the probability of expanding the optimal cost goal node.

[3] The possibility of such non-monotonic progress of beam search was experimentally observed in (Chu & Wah 1992a). Also, such pathological behaviors of heuristic search have been investigated in other works (Nau 1982; V.Bulitko *et al.* 2003)

search from such a viewpoint.

2 Rank of a Node

Pearl (Pearl 1984) presents an analysis of the expected node expansions of the A* algorithm depending on the heuristic accuracies. The A* algorithm expands nodes in the increasing order of their $f(n)$ values. This global competition policy performed by A* ensures that before termination it would expand all nodes having cost less than[4] the optimal solution cost. We observe that the nodes on the 'optimal-path' are usually very competitive among nodes at their locality and if we can restrict the node expansions up to the 'optimal-path' node[5] at a given level, we can attain optimality with fewer expansions (Aine, Chakrabarti, & Kumar 2009). Following this observation, we introduce the concept of 'rank' of a node ('rank' denotes the relative position of a node in terms of $f(n)$ value among a set of nodes at a particular level) and establish its relation with the convergence of a search algorithm. We assume that our search space is a finite depth tree (depth = N) with a unique optimal cost solution. We also assume that the heuristic used is *consistent* (Zhang 1999). We consider five sets of nodes for each level (l) of a search tree, defined as follows,

Definition 1. $V(l)$: *For a given level l, $V(l)$ denotes the set of nodes belonging to that level that are surely expanded by A*, thus,*

$$V(l) = \begin{cases} \{s\} & \text{if } (l = 0) \\ \{n \mid f(n) < f^*\} \cup \{n_{g,l}\} & \text{otherwise} \end{cases} \tag{1}$$

Where f^* is the cost of the optimal solution, s is the start node and $n_{g,l}$ is the 'optimal-path' node at level l.

Definition 2. $W(l)$: *For a given level l, $W(l)$ denotes the set of nodes (belonging to level l) that have f-value less than the 'optimal-path' node at that level (l) plus the 'optimal-path' node at that level, i.e.,*

$$W(l) = \begin{cases} \{s\} & \text{if } (l = 0) \\ \{n \mid f(n) < f(n_{g,l})\} & \\ \cup\{n_{g,l}\} & \text{otherwise} \end{cases} \tag{2}$$

Where $n_{g,l}$ is the 'optimal-path' node at level l.

Definition 3. $X(l,b)$: *For a given level l, $X(l,b)$ (for $b >= 1$) denotes the set of nodes expanded by the beam search algorithm (beam width = b) at that level. Thus,*

$$X(l,b) = \begin{cases} \{s\} & \text{if } (l = 0) \\ \{n \mid rank(n) <= b, \text{ and} & \\ Parent(n) \in X(l-1,b)\} & \text{otherwise} \end{cases} \tag{3}$$

Where $rank(n)$ among a set of nodes is defined as the position of n if the nodes in the chosen set are sorted (in an increasing order) according their $f(n)$ values.

[4]Node expansion characteristics of A* also depends on the possibility of ties and the tie-breaking choices. In this work, we ignore the tie-breaking overheads. However, the results can be extended to consider the possibility of ties.

[5]An 'optimal-path' node at a given level is the predecessor of the optimal cost goal node.

Definition 4. $Y(l,b)$: *For a given level l, $Y(l,b)$ denotes the set of nodes that have f-value less than the 'optimal-path' node at that level (l) and whose parents belong to $X(l-1,b)$, plus the 'optimal-path' node at that level, i.e.,*

$$Y(l,b) = \begin{cases} \{s\} & \text{if } (l = 0) \\ \{n \mid f(n) < f(n_{g,l}) \text{ and} & \\ Parent(n) \in X(l-1,b)\} & \\ \cup\{n_{g,l}\} & \text{otherwise} \end{cases} \tag{4}$$

Definition 5. $Z(l,b,v)$: *For a given level l, $Z(l,b,v)$ denotes the set of nodes at level l that have f-value less than the parameter val and whose parents belong to $X(l-1,b)$,*

$$Z(l,b,v) = \begin{cases} \{s\} \ \text{if } (l = 0) \\ \{n \mid f(n) < v \text{ and} \\ Parent(n) \in X(l-1,b)\} \\ \text{otherwise} \end{cases} \tag{5}$$

From the definitions, we observe that for each level A* expands at least $V(l)$ nodes, where as $W(l)$ denotes the set of nodes which always contains the 'optimal-path' node and all the nodes having estimated cost less than the 'optimal-path' node. Therefore, if we expand only $W(l)$ nodes at a given level, convergence is ensured. Next, we define a few rank related terminologies, denoting the cardinalities of the above mentioned sets.

Definition 6. $MRA^*(l)$ **:** *It is the maximum rank of a node at level l among nodes in the same level, which is surely expanded by A*, i.e.,*

$$MRA^*(l) = |V(l)| \tag{6}$$

Definition 7. $IOR(l)$ **:** *It is the rank of the 'optimal-path' node at a level l among all the nodes at level l , i.e.,*

$$IOR(l) = |W(l)| \tag{7}$$

Definition 8. $COR(l,b)$ **:** *It is the rank of the 'optimal-path' node at a level l among all the nodes in the open list which are at level l when for each level l' ($l' < l$), at most b nodes are expanded, i.e.,*

$$COR(l,b) = |Y(l,b)| \tag{8}$$

From the definitions, we observe that the $MRA^*(l)$ and $IOR(l)$ are inherent properties of the search space (and the heuristic values). On the other hand, the $COR(l,b)$ values depend on the choice of cut-off (b) values as well as on the search space characteristics. Next, we formally establish the relation between these rank values.

Theorem 1.

$$\forall l,\ 0 \le l \le N,\ \ IOR(l) \le MRA^*(l) \tag{9}$$

Theorem 2.

$$\forall l,\ 0 \le l \le N,\ COR(l,b) \le IOR(l) \\ \text{for any } b\ (b \ge 1) \tag{10}$$

Proofs of these two theorems can be obtained directly from the definitions. For a given level of the search tree, $IOR(l)$ is the rank of the 'optimal-path' node among all the nodes belonging to level l. A* will aways expand the 'optimal-path' node at a given level, additionally it may also expand some more nodes (which have estimated costs less than the f^* value), thus, $IOR(l) \leq MRA^*(l)$. On the other hand, $COR(l,b)$ restricts the number competing nodes through bounds on expansion at earlier levels. Therefore the $COR(l,b)$ value is upper bounded by the independent 'optimal-path' rank value ($IOR(l)$).

The above presented theorems establish the relationship between the different conditional and independent rank values. In the rest of this paper, we shall use these rank definitions to characterize the convergence properties of beam search.

3 Characterizing Beam Search

In this section, we formalize the properties of beam search in terms of the convergence probability. We start our discussion by defining the conditions for guaranteed convergence and subsequently move on to the probabilistic models to compute the convergence probability of beam search conditioned by the beam width choice.

Conditions for Guaranteed Convergence In Section 2 we introduced the concept of $COR(l,b)$ at a given level of a search tree, which denotes the rank of the 'optimal-path' node among nodes in the open list when at all earlier levels l' ($l' < l$), at most b nodes are expanded. Now, among the nodes in the open list only the best b nodes will be chosen for expansion (in beam search). This implies that the 'optimal-path' node (at level l) is expanded only when $b \geq COR(l,b)$. Therefore, the convergence of beam search depends on the $COR(l,b)$ values. In the following theorem we formally define the optimality criterion of beam search.

Theorem 3. *Beam Search (b) successfully converges if and only if*

$$\forall l, 0 \leq l \leq N,\ \ COR(l,b) \leq b \tag{11}$$

Corollary 1. *Beam Search (b) successfully converges if*

$$\forall l, 0 \leq l \leq N,\ \ IOR(l) \leq b \tag{12}$$

Expected Convergence Probability Theorem 3 discusses the condition under which beam search guarantees optimality and shows that if the beam width is equal or more than the conditional optimal node rank then search is guaranteed to terminate. However, the condition can not be directly applied to obtain a deterministic answer about the convergence of beam search as the actual rank values (both conditional and independent) are unknown. Moreover, determination of a strict rank bound is not likely, because if we get a constant bound on NP-hard problems (like TSP) then $P = NP$ is proved. Thus, instead of determining a constant rank value, we propose to obtain a probabilistic distribution of the $COR(l,b)$ values, and use that distribution to compute the convergence probability of beam search. For this, we define a term called *goal expansion probability*, $gep(l,b)$ of beam search (for a given level l and a beam width b).

Definition 9. $gep(l,b)$ **:** *$gep(l,b)$ denotes the probability of expanding an 'optimal-path' node at level l when it is already in the open list and the beam width chosen is b, i.e.,*

$$gep(l,b) = P(COR(l,b) \leq b) \tag{13}$$

$gep(l,b)$ represents the probability of obtaining the 'optimal-path' node within the best (in terms of $f(n)$) b-nodes at level l; when the 'optimal-path' node lies in the open list and for each predecessor level the node expansion is restricted by the beam width choice b. Thus, with $gep(l,b)$ we define a model, representing the chance of expanding the 'optimal-path' node at a level if a maximum number of b nodes are expanded at that level.

Using this definition, the (global) expected probability of convergence for beam search using a chosen beam width b can be defined as,

Definition 10. *Expected Convergence Probability $ECP(b)$ of a search space represents the probability of obtaining the optimal cost goal node if at each level of the search space at most b nodes are expanded. From the earlier definitions $ECP(b)$ can be expressed as,*

$$\begin{aligned} ECP(b) &= \Pi_{l=0}^{l=N} gep(l,b) \\ &= \Pi_{l=0}^{l=N} P(COR(l,b) \leq b) \end{aligned} \tag{14}$$

considering N to be the goal depth.

For any level, the 'optimal-path' node expansion probability is conditioned by the expansion probabilities of its predecessors, as a node can not be expanded if all of its predecessors are not expanded. Therefore, the convergence probability ($ECP(b)$) can be formulated as a joint probability of all $gep(l,b)$ values for all levels l ($0 \leq l \leq N$).

4 Beam Search on Uniform Search Tree Model

In this section, we compute the conditional rank values as a function of search space configuration and heuristic errors considering a uniform cost search tree model (a modified version of the search trees used in (Pearl 1984)). Our search space is modeled as a uniform m-ary tree T, with a unique start state S and a unique goal state G, situated at a distance N from S (Figure 1). The solution path is given as ($S- > n_{g,1}..- > n_{g,i}- > ..- > n_{g,N-1}, G$) where $n_{g,i}$ denotes the 'optimal-path' node at level i. The trees $T_1..T_i..T_N$ are sub-trees of T, one level removed from the 'optimal-path'. Each 'off-course' sub-tree T_i is rooted at a direct successor of $n_{g,i-1}$ which is off the solution path (there are $m-1$ such T_i for each i). An 'off-course' node[6] is labeled as $n_{i,j}$, where j denotes the level of that node and i denotes the root level of the 'off-course' sub-tree (T_i) to which the node belongs.

Using the above mentioned search tree model, first we compute the $COR(l,b)$ values for each level of the search tree. Later, we shall use these results to estimate the convergence probability of the beam search technique ($ECP(b)$). For this, we define the following terminologies.

[6]An 'off-course' node is a node which does not lie in the 'optimal-path' to goal.

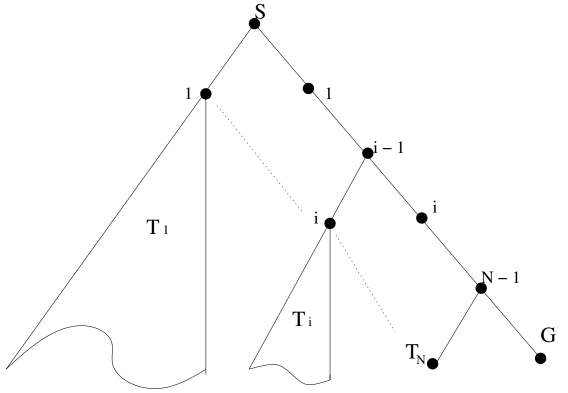

Figure 1: Uniform binary tree (model)

Definition 11. *competitive_goal_probability* $gcp(i,l)$ *:* $gcp(i,l)$ *denotes the probability of* $f(n_{i,l})$ *being less than the estimated cost (*f*-value) of the 'optimal-path' node at that level (*l*), i.e.,*

$$gcp(i,l) = P(f(n_{i,l}) < f(n_{g,l})) \quad (15)$$

Definition 12. *competitive_non-goal_probability* $ncp(i,j,l)$ *:* $ncp(i,j,l)$ *denotes the probability of* $f(n_{i,l})$ *being less than* $f(n_{j,l})$ *where* $n_{i,l}$ *and* $n_{j,l}$ *are two off-course nodes at the same level (*l*) but belonging to two different off-course subtrees (*T_i *and* T_j *respectively), i.e.,*

$$\begin{array}{l} ncp(i,j,l) = P(f(n_{i,l}) < f(n_{j,l})), \\ \textit{where } n_{i,l} \textit{ and } n_{j,l} \textit{ are two off-course nodes at level } l, \\ \textit{belonging to } T_i \textit{ and } T_j \textit{ respectively} \end{array} \quad (16)$$

The competitive probability values denote the chance of expansion for a node $n_{i,l}$, when directly competed against the 'optimal-path' node ($gcp(i,l)$) or some other 'off-course' node $n_{j,l}$ ($ncp(i,j,l)$) at level l. Next, we estimate these expansion probability values from the heuristic information. We assume that the relative estimation errors are independent random variables ($e_1 \leq \Gamma(n) \leq e_2$) with an arbitrary distribution function, i.e.,

$$\Gamma(n) = [h^*(n) - h(n)]/h^*(n) \quad (17)$$

Lemma 1. *For* m*-ary uniform search trees with the chosen heuristic error model (Equation 17),*

$$\begin{array}{l} gcp(i,l) = 1 - P(\Gamma(n) - \Gamma(g) \times \\ (N-l)/(N+l+2(1-i)) \\ \leq 2(1+l-i)/(N+l+2(1-i))) \end{array} \quad (18)$$

$$\begin{array}{l} ncp(i,j,l) = 1 - P(\Gamma(n_1) - \Gamma(n_2) \times \\ (N+l+2(1-j))/(N+l+2(1-i)) \\ \leq 2(j-i)/(N+l+2(1-i))) \end{array} \quad (19)$$

Where $\Gamma(n)$*,* $\Gamma(g)$*,* $\Gamma(n_1)$ *and* $\Gamma(n_2)$ *are identically distributed random variables within* (e_1, e_2) *bound.*

To further quantify the competitive probabilities and to understand its relation with heuristic accuracy, we choose an error distribution function. We assume that $\Gamma(n)$ is a random variable uniformly distributed over $(0, e)$.

Lemma 2. *With the uniform tree model and the uniform (*$[0,e]$*) heuristic error distribution, the* $gcp(i,l)$ *and* $ncp(i,l)$ *values are computed as,*

$$\begin{array}{l} gcp(i,l) = \begin{cases} (b-d)^2/(b-c)(b-a), \\ \textit{if } (d > c) \\ 1-(d-b)^2/(d-a)(b-a), \\ \textit{otherwise} \end{cases} \\ a = -e(N-l)/(N+l+2(1-i)), \\ b = e, \\ c = (a+b)/2, \textit{and} \\ d = 2(l+1-i)/(N+l+2(1-i)) \end{array} \quad (20)$$

$$\begin{array}{l} ncp(i,j,l) = \begin{cases} (b-d)^2/(b-c)(b-a), \\ \textit{if } (d > c) \\ 1-(d-b)^2/(d-a)(b-a), \\ \textit{otherwise} \end{cases} \\ a = -e(N+l+2(1-j))/(N+l+2(1-i)), \\ b = e, \\ c = (a+b)/2, \textit{and} \\ d = 2(j-i)/(N+l+2(1-i)) \end{array} \quad (21)$$

Next, we estimate the $COR(l,b)$ using these competitive probabilities. The basic idea is as follows, for a given level l, the relative 'optimal-path' node rank can be obtained by competing $n_{g,j}$ with all the nodes in the open list and finding out how many times the 'optimal-path' node gets beaten by an 'off-course' node. The probabilistic summation of of all such events will determine the distribution of $COR(l,b)$, which in turn can be used to obtain the $gep(l,b)$ and $ECP(b)$ values for a given beam width b.

A node at level l (whether 'on-course' or 'off-course') is present in the open list only when its parent at level $l-1$ has already been expanded. Thus, apart from the independent competitive (goal and non-goal) probabilities, we need to compute the expansion probabilities of its parent nodes. For the 'optimal-path' nodes, we have already defined the conditional rank probabilities. Therefore, the probability of $n_{g,l}$ to be in the open list can be calculated using the following equation.

$$\begin{array}{l} P(n_{g,l} \in OpenList) = \Pi_{j=0}^{j=l-1} gep(j,b) \\ = \Pi_{j=0}^{j=l-1} P(COR(j,b) \leq b)) \end{array} \quad (22)$$

The same rank properties have not yet been been established for the 'off-course' nodes. Next, we define the conditional rank probabilities of an 'off-course' node $n_{i,l}$ using the set $Z(l,b,v)$ (Equation 5, Section 2).

Definition 13. $CNOR(i,l,b)$ **:** *It is the rank of an 'off-course' node* $n_{i,l}$ *among the nodes in the open list that are also at level* l *when for all levels* $l' < l$*, at most* b *nodes in the open list are expanded, i.e.,*

$$CNOR(l,b) = |Z(l,b,f(n_{i,l})| \quad (23)$$

The $CNOR(i,l,b)$ values can be obtained by computing the cardinality of set $Z(l,b,v)$ using the f-value of the given node ($n_{i,l}$) as the cut-off parameter (val). Following this, we define the expansion probability $nep(i,l,b)$ of an 'off-course' node ($n_{i,l}$) depending on the chosen beam width b.

Definition 14. $nep(i,l,b)$ **:** *$nep(i,l,b)$ denotes the probability of expanding an 'off-course' node $n_{i,l}$ when it is already in the open list and the beam width chosen is b, i.e.,*

$$nep(i,l,b) = P(CNOR(i,l,b) \leq b)) \tag{24}$$

Using this definition the probability of an 'off-course' node $n_{i,l}$ to be in open list can be computed as,

$$P(n_{i,l} \in OpenList) = \begin{array}{l} \Pi_{j=0}^{j=i-1} gep(j,b) \times \\ \Pi_{j=i}^{j=l-1} nep(i,j,b) \end{array} \tag{25}$$

In the following theorems, we compute the conditional rank probabilities as a function of search space configuration and heuristic errors.

Theorem 4. *With the uniform cost m-ary tree model, the $COR(l,b)$ function can be approximated as,*

$$\begin{array}{l} COR(l,b) \approx \left\{ \begin{array}{l} 1 \;\; if \; l = 0 \\ ND(\mu,\sigma), \; otherwise \end{array} \right. \\ where \; \mu = 1 + (m-1)\sum_{i=1}^{i=l} \\ (\Pi_{j=i}^{j=l-1} nep(i,j,b)) m^{l-i} gcp(i,l) \\ and, \\ \sigma^2 = (m-1)\sum_{i=1}^{i=l} (\Pi_{j=i}^{j=l-1} nep(i,j,b)) \\ m^{l-i} gcp(i,l)(1 - gcp(i,l)) \end{array} \tag{26}$$

Where $ND(\mu,\sigma)$ denotes a Normal Distribution function having mean μ and standard deviation σ.

Theorem 5. *With the uniform cost m-ary tree model, the $gep(l,b)$ probability can be approximated as,*

$$gep(l,b) \approx \left\{ \begin{array}{ll} 1.0, & if \, b \geq m^l \\ 0.5[1 + \phi((b-\mu)/\sigma\sqrt{2})] & otherwise \end{array} \right. \tag{27}$$

where μ, and σ are same as used in Theorem 4, and $\phi(x)$ is the standard error function for Normal distribution.

Theorem 6. *With the uniform cost m-ary tree model, the $CNOR(l,b)$ function can be approximated as,*

$$\begin{array}{l} CNOR(i,l,b) \approx ND(\mu,\sigma) \\ where, \\ \mu = 1 + (m-1)\sum_{j=1}^{j=i-1} (\Pi_{k=j}^{k=l-1} nep(j,k,b)) \\ m^{l-j}(1 - ncp(i,j,l)) \\ +(m-1)\sum_{j=i+1}^{j=l} (\Pi_{k=i}^{k=j-1} gep(k,b)) \\ (\Pi_{k=j}^{k=l-1} nep(j,k,b)) m^{l-j}(1 - ncp(i,j,l)) \\ +(m-1)\sum_{j=i+1}^{j=l} \\ (\Pi_{k=j}^{k=l-1} nep(i,j,b)) m^{l-j}(1 - ncp(i,i,l)) \\ +(m-2)(\Pi_{k=i}^{k=l-1} nep(i,k,b)) m^{l-i}(1 - ncp(i,i,l)) \\ +(\Pi_{k=i}^{k=l-1} gep(k,b))(1 - gcp(i,l)) \\ and, \\ \sigma^2 = 1 + (m-1)\sum_{j=1}^{j=i-1} (\Pi_{k=j}^{k=l-1} nep(j,k,b)) \\ m^{l-j}(1 - ncp(i,j,l)) ncp(i,j,l) \\ +(m-1)\sum_{j=i+1}^{j=l} (\Pi_{k=i}^{k=j-1} gep(k,b)) \\ (\Pi_{k=j}^{k=l-1} nep(j,k,b)) \\ m^{l-j}(1 - ncp(i,j,l)) ncp(i,j,l) \\ +(m-1)\sum_{j=i+1}^{j=l} (\Pi_{k=j}^{k=l-1} nep(i,j,b)) m^{l-j} \\ (1 - ncp(i,i,l)) ncp(i,i,l) \\ +(m-2)(\Pi_{k=i}^{k=l-1} nep(i,k,b)) \\ m^{l-i}(1 - ncp(i,i,l)) ncp(i,i,l) \\ +(\Pi_{k=i}^{k=l-1} gep(k,b))(1 - gcp(i,l)) gcp(i,l) \end{array} \tag{28}$$

Theorem 7. *With the uniform cost m-ary tree model, the $nep(i,l,b)$ probability can be approximated as,*

$$nep(i,l,b) \approx \left\{ \begin{array}{l} 1.0, \;\; if \, b \geq m^l \\ 0.5[1 + \phi((b-\mu)/\sigma\sqrt{2})] \\ otherwise \end{array} \right. \tag{29}$$

where μ, and σ are same as used in Theorem 6, and $\phi(x)$ is the standard error function for Normal distribution.

The basic idea behind the above formulation is that rank of a node is determined by the number of nodes having cost less than its cost, i.e., rank of a node at a given level l can be expressed as the summation of independent Bernoulli trials with all possible competitive nodes. Therefore, the distribution function of the expected rank values can be calculated as a sum of Binomial Distributions guided by the heuristic errors. Approximating the Binomial distribution functions using the corresponding Normal distribution, we get the above presented approximations.

From the results, we observe that the convergence probability of beam search ($ECP(b)$) can be calculated directly using the search space configuration (depth, branching factor) and the heuristic error information. It should be noted that the $gep(l,b)$ and $nep(i,l,b)$ values depend on the gep and nep values at levels l' ($l' < l$), i.e., the formulation is recursive. For practical purposes, these values can be computed in a level by level manner (from $l = 0$ to $l = N$). Once we obtain the $gep(l,b)$ values for each level of the search space the $ECP(b)$ values can be computed by multiplying these gep values using Equation 14. It may be noted that the formulation presented can also be used to compute the minimum beam width required for a pre-specified convergence probability target; depending on the search space configuration and heuristic error estimates.

5 Performance of Beam Search on Uniform Cost Trees

In this section, we use the above mentioned formulation to evaluate the convergence probabilities of beam search with different search space configurations and heuristic error bounds. We parameterize the search space (uniform tree) using the following three parameters, search tree depth (N), branching factor (m) and the heuristic error margin (choice of e in the uniform distribution model presented).

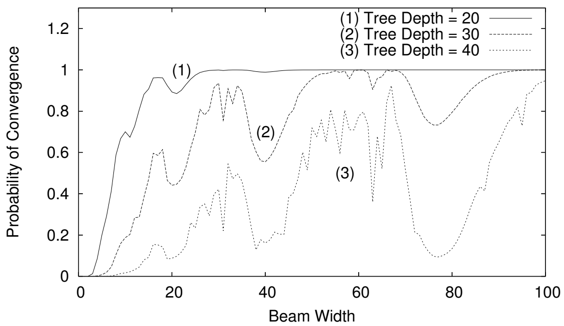

Figure 2: Convergence probability vs beam width for uniform binary trees of different depth (heuristic error margin $e = 1.0$)

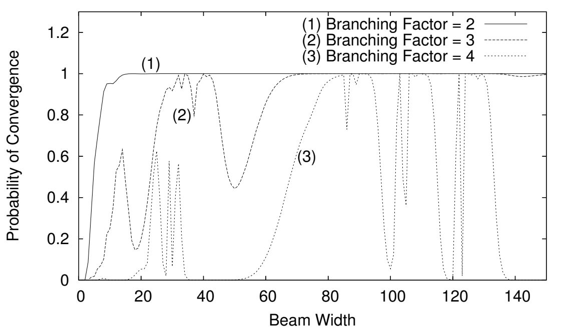

Figure 3: Convergence probability vs beam width for uniform trees (of depth 20) with different branching factors (heuristic error margin $e = 0.8$)

In Figure 2, we include the expected convergence vs. beam width choice results for three uniform binary tree with depth 20, 30 and 40, respectively, with $e = 1.0$. Figure 3 shows the curves for uniform search trees of depth 20 with different branching factors (m = 2, 3 and 4 respectively) with $e = 0.8$. In Figure 4, we present the convergence probability curves (with different beam choices) for a uniform binary tree of depth 20, with $e = 1.0$, 0.9, 0.8, and 0.5 respectively.

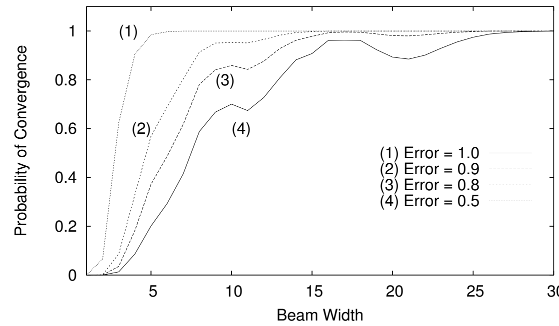

Figure 4: Convergence probability vs beam width for a uniform binary tree of depth 20, with different heuristic error margins

From the results we observe that the convergence probabilities have a monotonic dependency on the search space size and the heuristic error, i.e., the convergence is delayed with the increase of tree depth, branching factor and heuristic inaccuracy. However, similar consistency is not observed with the variation in beam width choices. The curves show that the expected convergence does not follow a monotonic pattern with the increase of beam width cut-off, and the probability may decrease with increasing beam width choices. This observation is in contrast to the general perception that with the increase in search effort the convergence probability should increase, and thus can be termed as an 'anomalous' behavior of beam search. The 'anomalous' variations reduce with the increase in beam width values and after a certain point the search progress becomes monotonic with the effort. This transition from non-monotonic to monotonic progress is attained earlier for smaller search spaces (Figures 2 and 3) as well as with increased heuristic accuracy (Figure 4).

Analysis of the Non-monotonic Progress In a generic situation, the non-monotonic behavior of beam search can be attributed to the fact that the increase in beam width impacts probability of expansion of the 'optimal-path' node in two seemingly contrasting directions. On one hand it improves the chances of an 'optimal-path' node at level l to be expanded by increasing the number of allowed expansions. On the other hand it may also increase the rank of the 'optimal-path' node by expanding more nodes at higher levels, i.e., with increase in beam width number of nodes in the open list increases, which in effect may increase the conditional rank of the 'optimal-path' node, thus reducing its expansion probability. Therefore, the progress of beam search is directly related to the conditional rank functions. In the following theorem, we use this observation to identify the condition for monotonic progress of beam search.

Theorem 8. *For a given search space configuration, beam search progress is monotonic if for each level of the search space, the COR function is monotonically non-increasing* *with beam width, i.e., beam search progress is monotonic if*

$$\forall l,\ COR(l, b_1) \geq COR(l, b_2),\ \textit{where}\ b_2 > b_1 \quad (30)$$

Corollary 2. *For any search space, there is a beam width b_{mon} such that the beam search progress is monotonic for all beam width choices $b > b_{mon}$.*

The above observations indicate that the 'optimal-path' node expansion is dependent on the conditional probabilities up to a certain point only. If the beam width is greater than or equal to the IOR value at a given level, the 'optimal-path' node at that level is always expanded. Therefore, with the increase of beam width values, the search non-monotonicity reduces as the expansion probabilities (for more number of levels) become independent rather than conditional. As the IOR values are comparatively less for accurate heuristic values, the monotonicity is attained earlier (Figure 4).

6 Discussions and Future Directions

Analytical Models In this paper, we analyzed the basic characteristics of beam search in the uniform cost search tree model and relative error based heuristic estimation. This model has been criticized (Korf & Reid 1998) as it is hard to estimate the actual error distribution for a heuristic and such inaccuracy may lead to incorrect predictions. Instead a static distribution of heuristic values has been recommended and it is shown that such a model can provide accurate estimations for problems like sliding tile puzzle, Rubik's cube etc. Moreover, some analysis uses random trees with a probabilistic branching factor rather than a constant branching (Zhang & Korf 1995). We used the uniform tree model mainly due to the following reasons, first, the static distribution model is good for cases having limited heuristic values (like sliding tile problems) but it is not a good choice for problems having wide range of heuristic values across problem instances (like TSP), and second, our study with the relative error based model with other search algorithms like AWA* (Aine, Chakrabarti, & Kumar 2007), Contract Search (Aine, Chakrabarti, & Kumar 2009) provided good correlation with the actual results on various real life problems (like TSP, 0/1 Knapsack etc). Also having the same model (as used for AWA* and Contract Search) provides a platform to analytically compare these algorithms (which can be a future direction). Finally, it may be noted that the basics of the formulation and ranking based methodology remains independent of the choice of models and can be easily extended to consider probabilistic branching and other distributions for heuristic estimation. Another interesting future challenge is to extend the ranking model to consider inconsistent (Zahavi *et al.* 2007) and inadmissible heuristics. It should be noted that the framework and the basic assumptions regarding the ranking remain equally applicable even if we consider inadmissible heuristics. However, it would be interesting to investigate the beam search progress under the possibilities of heuristics over-estimation.

Improving Beam Search The analysis presented indicates that the expected quality of results of beam search is critically and non-monotonically dependent on the beam width choices. It also shows that the optimal beam width choices can be selected using some properties of the search space. A future challenge is to use these information to improve the execution strategy of beam search depending on the desired quality-time trade-off. The analysis can also be used to provide a theoretical background to adaptive beam search approaches (Chu & Wah 1992a; 1992b; Aine, Chakrabarti, & Kumar 2009).

7 Conclusions

In this paper, we presented an analytical model to characterize the convergence properties of the breadth first beam search algorithm. We studied the node expansion behavior of beam search using a uniform search tree model and proposed a method to compute the expected convergence of beam search using the search space characteristics and the heuristic error distribution. The results presented explain the basic characteristics of the progress of beam search and highlight its non-monotonic relationship with the beam-width cut-off.

References

Aine, S.; Chakrabarti, P. P.; and Kumar, R. 2007. Awa* - a window constrained anytime heuristic search algorithm. In *IJCAI*, 2250–2255.

Aine, S.; Chakrabarti, P. P.; and Kumar, R. 2009. Contract search : An adaptive heuristic search strategy under node expansion constraints. In *Proceedings International Symposium on Combinatorial Search (SOCS-09)*.

Chenoweth, S. V., and Davis, H. W. 1991. High-performance a* search using rapidly growing heuristics. In *IJCAI*, 198–203.

Chu, L., and Wah, B. W. 1992a. Band search: An efficient alternative to guided depth-first search. In *Proceedings of the 4th International Conference on Tools with Artificial Intelligence*.

Chu, L., and Wah, B. W. 1992b. Solution of constrained optimization problems in limited time. In *Proceedings Workshop on Imprecise Computation, IEEE*.

Habenicht, I., and Mönch, L. 2002. Scheduling and dispatching: a finite-capacity beam-search-algorithm for production scheduling in semiconductor manufacturing. In *WSC '02: Proceedings of the 34th conference on Winter simulation*, 1406–1413. Winter Simulation Conference.

Korf, R. E., and Reid, M. 1998. Complexity analysis of admissible heuristic search. In *Proceedings of the 14th National Conference on Artificial Intelligence*, 305–310. Madison, Wisconsin, USA: AAAI Press.

Nau, D. S. 1982. An investigation of the causes of pathology in games. *Artif. Intell.* 19(3):257–278.

Pearl, J. 1984. *Heuristics: intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Rich, E., and Knight, K. 1991. *Artificial Intelligence*. McGraw-Hill Higher Education.

Tillmann, C., and Ney, H. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.* 29(1):97–133.

V.Bulitko; Li, L.; Greiner, R.; and Levner, I. 2003. Lookahead pathologies for single agent search. In *IJCAI*, 1531–1533.

Zahavi, U.; Felner, A.; Schaeffer, J.; and Stutervant, N. 2007. Inconsistent heuristics. In *Proceedings of 23rd National Conference on Artificial Intelligence*, 1211–1216.

Zhang, W., and Korf, R. E. 1995. Performance of linear-space search algorithms. *Artif. Intell.* 79(2):241–292.

Zhang, W. 1999. *State-space search: Algorithms, complexity, extensions, and applications*. New York, USA: Springer.

Zhou, R., and Hansen, E. A. 2005. Beam-stack search: Integrating backtracking with beam search. In *Proceedings of the ICAPS-05*, 90–98.