

A PARALLEL TREE SEARCH METHOD

Sei-ichi Nakagawa
Department of Information Science
Kyoto University
Kyoto, 606 Japan

Toshiyuki Sakai
Department of Information Science
Kyoto University
Kyoto, 606 Japan

We describe a new tree search method which searches best few paths with backtracking in parallel by the following algorithm (α - β - γ search). Expand the best α nodes at one time. If the number of new nodes which have been expanded by a selected node exceeds a pre-set threshold (β), only the best β nodes are kept and others are pruned. If the total number of newly generated node sequences and non-expanded node sequences exceeds a pre-set threshold (γ), only the best γ node sequences are kept, and others are removed. Repeat these consecutive processes until a complete sequence is generated. We evaluated this method by simulations and experiments, and obtained good performances both in the search efficiency and in the recognition rate.

1. INTRODUCTION

To recognize a perceptual input (ex. speech, image), all plausible candidates should be taken into consideration, because a system cannot know exactly what the input was, that is, the input is ambiguous. However, since the number of plausible candidate sequences becomes enormous in case of continuous speech recognition, image understanding and game, a system must select some best sequences and abandon the others. Therefore, the recognition of perceptual input contains the problems of tree search (pruning) and scoring mechanism.

In this paper, we describe a new tree search method employed in the LITHAN speech understanding system[1-4]. For convenience's sake, we would like to discuss the case of speech recognition as an example of recognition of perceptual input and the case of left to right parsing although various parsing strategies have been proposed[5,6].

2. SEARCHING STRATEGY OF ALTERNATIVE SEQUENCE

The aim of a speech recognition system is to determine, for an acoustic signal A and a task language L , a sequence of words W such that this sequence is consistent with the syntax, semantics and pragmatics of the task, and further that this sequence maximizes the a posteriori probability $P(W/A, L)$. We assume an equal probability for the appearance of each

sentence. Now $P(W/A, L) = P(W/L) \times P(A/W, L) / P(A/L)$. The term $P(A/L)$ in the denominator is independent of W , and therefore does not have to be known. We assumed that $P(W/L)$ was always constant for all correct sequences of words. Thus, $P(W/L)$ does not also have to be known. Therefore, we can regard that $P(W/A, L)$ is proportional to $P(A/W, L)$. However, this algorithm consumes much computing time, because the approach is equivalent to searching all possible paths or words.

Seeing this difficulty, we coped with the problem by dividing the whole process into two stages: the acoustic signal-to-phoneme sequence and the phoneme sequence-to-word sequence. This division corresponds to approximating $P(W/A, L)$ by $P(W/A, L) = \sum P(PH/A, L) \times P(W/PH, L) \propto P(W/PH_0, L)$, where PH denotes a sequence of phonemes and PH_0 a recognized sequence of phonemes (that is, we assume the probability of recognized phoneme $\gg$ that of other phoneme). Furthermore, the maximum of $P(W/PH_0, L)$ is approximately obtained from tree search methods.

If an utterance is composed of s words, and t alternative words are identified for each of these, the number of sequences to be considered is s^t . Therefore the problem of continuous speech recognition can be seen as that of searching through a space of s^t word sequences to find the most acceptable sequence (that is, the sequence with the highest score), that does not contradict all the given knowledge; syntax, semantics, pragmatics, etc.

Fig.1 illustrates an example of a parsing tree. In this figure, the alphabetical symbol and numerical value on a branch denote an identified word and its likelihood, respectively. An encircled numeral on a node indicates the average likelihood for a word sequence corresponding to the path from the root node to that node. A double encircled numeral on a terminal node indicates the score of a complete sentence. The LITHAN system rejects words having a score below a pre-determined value, say, 80. It also rejects sentences below another value, say 85. Thus, in Fig.1, there are five complete sentences: BHM, BHN, CJO, CJP, and CK. If the best-first tree search is applied, the sentence BHN will be selected after the node selection sequence: B→BH→C→CJ→BHN. On the other hand, if we use the depth-first search, BHM will be selected (A→AE→AF→B→BG→BH→BHM). In the case of depth-first search with ordering, BHN will be selected (B→BH→BHN). We would, however, like to select the sentence CJP. Now, the best-first search may guarantee the sequence with highest score, if the score of a word sequence has a true cost function for searching (namely the breadth-first search) [7]. However, such a method requires much computation time and many storage memories. Our searching scheme is not such a case. Although many techniques have been developed in artificial intelligence research [8,9], we will propose a new search technique and pruning method suitable for recognition of perceptual inputs.

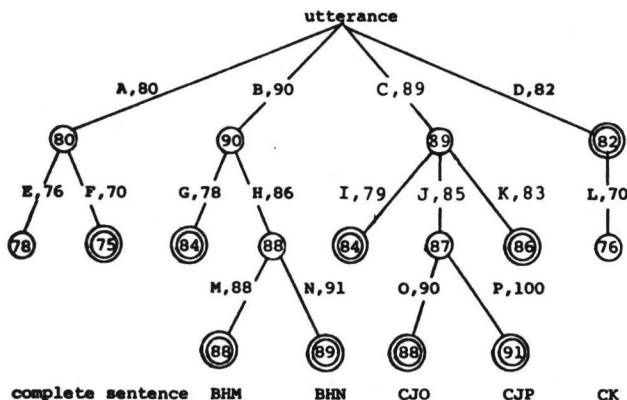


Fig. 1 An example of tree searching.

3. A PARALLEL SEARCH METHOD (α - β - γ METHOD)

We define the parameters which are used by this method.

- α : number of nodes which are expanded in parallel at the same time.
- β : number of nodes which are kept in all generated nodes at each branching point.

γ : total number of partial strings which are kept in a partial tree.

The score of a partial sentence is defined as the average of the score of words constituting this partial sentence.

Algorithm

- step 1: Enter the root node into the set of OPEN NODE.
 - step 2: Expand α nodes in the set, each of which is the last node of one of partial strings with the highest α average scores. If the number of nodes in the set is less than α , these all nodes are expanded. If the expanded node is a terminal node, go to step 5.
 - step 3: Enter the β nodes with the highest β scores out of the generated nodes at each branching point into the set of OPEN NODE.
 - step 4: If the number of nodes in the set of OPEN NODE is more than γ , select the γ nodes, each of which is the last node of one of partial strings with the highest γ average scores, and the other nodes (or strings) are pruned. Go to step 2.
 - step 5: The string from root node to this terminal node is regarded as an input string. If the number of strings which are regarded as an input string is more than δ , stop. Otherwise, go to step 2.
- We call the successive operations from the step 2 to the step 4 as one 'round'.

Although the method of Lowerre in the HARP system (called beam search [10]) does not have the function of backtracking, our method has it implicitly in the case of $\alpha < \gamma$. We call this method as α - β - γ tree search method (or best- α tree search) for convenience' sake.

For example, in Fig.1, let α , β , γ , and δ be 2, 3, 4 and 2, respectively. Here the following word sequences will be kept at each round (the two, $\alpha=2$, underlined sequences are to be expanded at the next round).

- Round 1: B, C, D
D may be a complete sentence in the case the score is less than 85, so it is rejected but may be reserved as a partial sentence.
- Round 2: BH, CK, CJ, D
G and I are rejected. CK is a complete sentence.
- Round 3: CK, BHM, BHN, CJO, CJP
These are all the complete sentences. In the end, CJP is selected as the recognized sentence.

Larger values of α , β and γ would bring a better recognition result, but such would also increase

the tree search time. Therefore, we have to decide the optimum thresholds of α , β and γ .

4. COMPUTER SIMULATION

Our tree search method was evaluated by computer simulations. The evaluation of a tree search method depends on the depth of tree (DEPTH), branching factor (BRANCH), scoring mechanism and so on.

For each [DEPTH, BRANCH], 100 trees and the score of each node were generated by a random generator of a uniform distribution [from 0 to 100]. The score of a partial sentence was defined as the average score over all words in the partial sentence. We regarded the trees of [DEPTH=5, BRANCH=5] as the standard tree because a computer had the limitation of an amount of stored memories. The depth of all terminal nodes and the number of branches are same for a tree in this simulation. We define that the input sequence is the sequence with the highest average score. Thus, our problem is to find the sequence with the highest average score. Table 1 shows the kinds of trees which were generated by a random generator. LSSS shows the logarithm of the search space size of a tree. This value can be calculated from multiplying logarithm of DABF (dynamic average branching factor) by ASL (average sentence length) [4,11]. The table 2 shows the set of parameters for tree search which were evaluated. We can divide these parameter sets into the two types. Although the type 1 ($\alpha=\gamma$) does not have the function of backtracking, the type 2 ($\alpha<\gamma$) has it implicitly. The parameter δ was always set to 1.

Table 1 Kind of trees

type	number of branch	depth of a tree	LSSS	number of nodes
1	4	5	10.0	1364
2	5	4	9.3	780
3	5	5	11.6	3905
4	5	6	13.9	19530
5	6	5	12.9	9330

Table 2 Set of parameters

type	method	α	β	γ
1	1	1	1	1
	2	2	2	2
	3	3	3	3
	4	4	4	4
	5	5	5	5
2	6	1	5	10
	7	2	5	10
	8	3	5	12
	9	5	5	15

The simulation results are shown in Table 3 and Fig.2. The column '1' in the 'rank of recognition result' corresponds to the number of trees which were recognized correctly. This number associates with 'recognition rate'. The column '2' indicates that the input was recognized as the second best candidate. As a logical conclusion, the larger LSSS becomes, the worse the recognition rate becomes. From Fig.2, we find when the tree is small, the type 2 is inferior to the type 1. On the other hand, when it is large, the type 2 is superior to the type 1. And also we can say that the best-first search which is simulated by [$\alpha=1, \beta=5, \gamma=10$] is inferior to anyone of parallel search ($\alpha \geq 2$).

Table 3 Simulation result (DEPTH=5, BRANCH=5)

α	β	γ	number of expanded nodes	rank of recognition result										
				1	2	3	4	5	6	7	8	9	10	11~
1	1	1	25	23	11	5	2	6	6	0	3	3	0	41
2	2	2	45	52	12	4	7	2	2	4	3	2	1	11
3	3	3	65	71	9	9	5	2	0	1	2	0	0	1
4	4	4	85	78	12	5	2	0	0	1	1	0	0	1
5	5	5	105	82	13	3	1	0	0	0	0	0	0	1
1	5	10	66	58	12	10	4	3	3	2	1	1	0	6
2	5	10	78	74	13	7	1	2	0	0	1	0	0	2
3	5	12	95	80	11	8	1	0	0	0	0	0	0	0
5	5	15	130	83	13	3	1	0	0	0	0	0	0	0

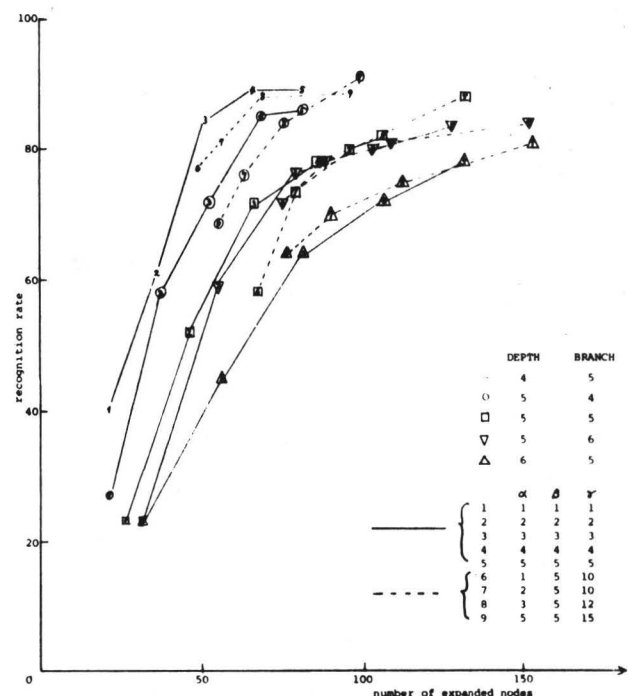


Fig.2 Relationship between recognition rate and number of expanded nodes.

This search method can control the search space size by taking the cost of search into consideration. That is, the scoring mechanism of a partial sentence is modified as follows. Let n be the number of words in a partial sentence and $S_1 \dots S_n$ be the score of each word, respectively. The new score of this sequence is defined by average $[S_1 \dots S_n] + W \times n$, where W is a weighting coefficient. If W is a negative value, the search space size becomes large and the recognition rate would be improved. In other words, this leads that a string of short length is treated significantly than that of long length. If W is a positive value, the opposite results will be obtained.

Table 4 and Fig.3 show the results of this tree search method. Methods from 1 [$\alpha=1, \beta=1, \gamma=1$] to 5 [$\alpha=5, \beta=5, \gamma=5$], the type 1, are independent of this modification, because these methods always expand the partial sentences of same length in parallel. The simulation results are in approximate agreement with expected results. However, when W was smaller than -10, the recognition results became wrong. We found even if W is selected to an optimum value for the best first search, it is also inferior to any parallel search (refer to Fig.4).

Next, we investigated the effectiveness of parameters: β and γ . The simulation results for $\alpha=1, 2$ are shown in Figs.4 and 5. In the case of these simulations [BRANCH=5], the best of β was 3-5. The larger the value of γ becomes, the better the recognition rate becomes and then it is saturated. On the other hand, the number of expanded nodes also becomes larger dramatically. Therefore, we must set the parameters α, β and γ according with the purpose of applications.

At least, we think that α should be larger than 2 in pattern understanding systems, although a best-first search has been often adopted in the first stage of speech understanding systems[12].

Table 4 Simulation result (DEPTH=5, BRANCH=5, WEIGHT=-4)

α	β	γ	number of expanded nodes	rank of recognition result										
				1	2	3	4	5	6	7	8	9	10	11~
1	1	1	25	23	11	5	2	6	6	0	3	3	0	41
2	2	2	45	52	12	4	7	2	2	4	3	2	1	11
3	3	3	65	71	9	9	5	2	0	1	2	0	0	1
4	4	4	85	78	12	5	2	0	0	1	1	0	0	1
5	5	5	105	82	13	3	1	0	0	0	0	0	0	1
1	5	10	86	70	14	7	2	2	1	1	0	0	0	3
2	5	10	93	76	16	5	2	1	0	0	0	0	0	0
3	5	12	113	83	12	4	1	0	0	0	0	0	0	0
5	5	15	145	85	12	3	0	0	0	0	0	0	0	0

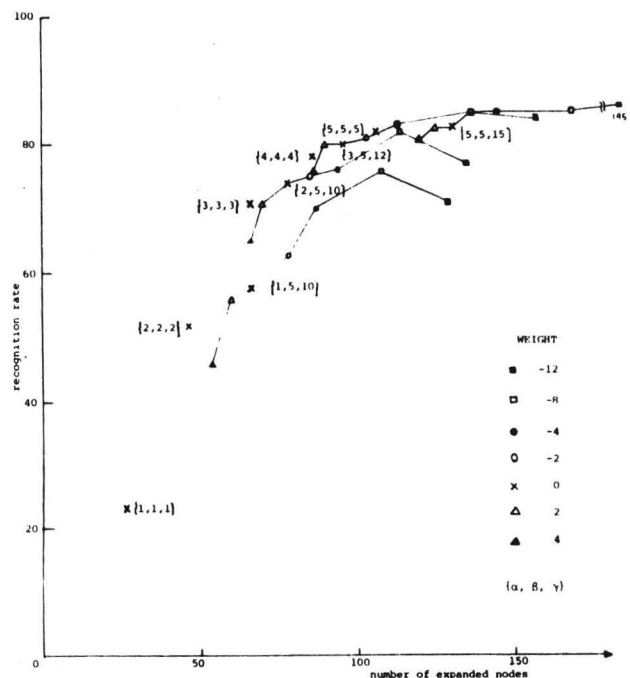


Fig.3 Relationship between recognition rate and number of expanded nodes. (DEPTH=5, BRANCH=5)

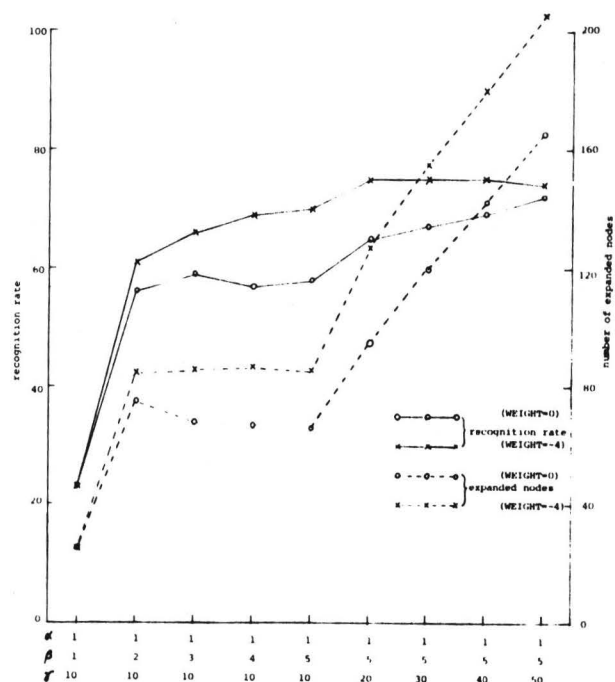


Fig.4 Relationship between recognition rate and number of expanded nodes. ($\alpha=1$, DEPTH=5, BRANCH=5)

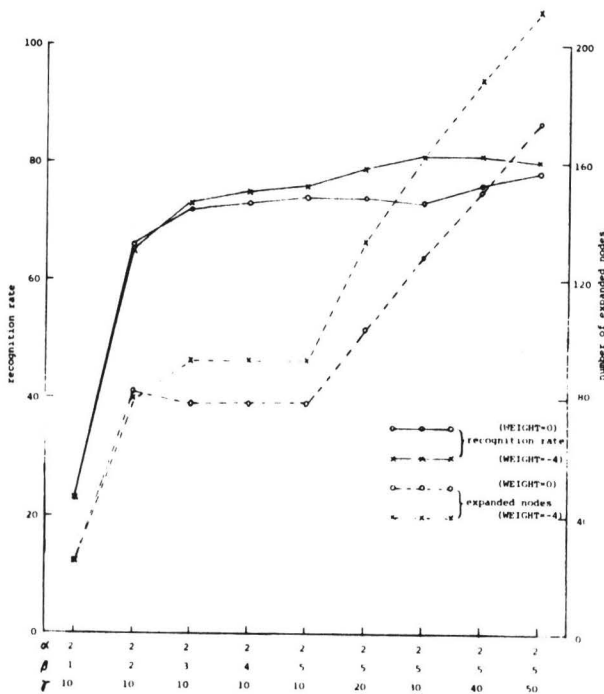


Fig.5 Relationship between recognition rate and number of expanded nodes.
($\alpha=2$, DEPTH=5, BRANCH=5)

5. SPEECH RECOGNITION OF 'ARITHMETIC EXPRESSION'

We applied our tree search method to the speech recognition of 'Arithmetic Expression' as an example in a real world field[1,2]. In this task, the vocabulary size is 23, where each word has an average of 2.1 syllables. For example, this task accepts the sentence such as '1.2*(345-6789.0)='.

The experiments of speech recognition of arithmetic expressions were tested on a total of 80 utterances containing 560 words, spoken by five male adults at a normal speed (these were uttered in Japanese). The reference patterns were common to all speakers. The DEPTH and BRANCH on the average were 7 and 10, respectively. The value of β is always pre-set at 5, because if a word really uttered belongs to one of predicted words probably it will be identified as one of the best five words (or candidates). The value of δ was set at 2.

The relationship between recognition rates and parameters of the tree search is shown in Table 5. In this table, the search time column indicates for each case the ratio of the number of verified words to the case where $\alpha=1$, $\beta=1$ and $\gamma=1$. The results are in approximate agreement with simulation on the relative merits for nine search methods. However, the care is necessary

to compare it with simulations, because the paths with highest scores in trees of these experiments are not always correct. From these results, we can also conclude that the best-first search (or parallel search) is superior to the best-first search.

Table 5 Recognition results of 'Arithmetic Expression'.

α	β	γ	search time	recognition rate		notes
				word	sentence	
1	1	1	1.0	80.1%	31.0%	no back tracking
2	2	2	2.0	88.4	47.0	
3	3	3	3.0	92.1	56.0	
4	4	4	4.0	93.1	58.0	
5	5	5	5.0	93.5	60.0	
6	5	6	6.0	93.8	61.0	
1	5	10	1.9	90.3	52.0	(nearly) best-first search
2	5	10	2.6	91.9	58.0	
3	5	12	3.3	93.2	60.0	
5	5	15	5.1	93.6	62.0	(rough) breadth-first search

6. CONCLUSION

We proposed a new tree search method which is suitable for the recognition of perceptual input. This is a parallel tree search method with backtracking implicitly. We evaluated this method by computer simulations and experiments of speech recognition. From these results, we found that the parallel search was superior to the best first search. By taking a weighting function into consideration, we would control the recognition rate and search space size. Further, we found that the larger the tree becomes, the more the performance of parallel search becomes effective.

ACKNOWLEDGEMENT

The authors would like to thank Mr. K. Goto for his co-operative help in computer simulations.

References

- [1] T. Sakai and S. Nakagawa: Proceedings of the 3rd IJCP, p.621 (1976).
- [2] T. Sakai and S. Nakagawa: IECEJ Trans. Vol.E-60, No.1, p.13 (1977).
- [3] S. Nakagawa and T. Sakai: Conference Record of 1978-ICASSP, p.726 (1978).
- [4] S. Nakagawa and T. Sakai: Joint Meeting of ASA and ASS, JASJ, Vol.64S, No.1 (1978).
- [5] W.A. Woods: Proceedings of the 5-th IJCAI, p.18 (1977).
- [6] F. Hayes-Roth and V.R. Lesser: Proceedings of the 5-th IJCAI, p.27 (1977).
- [7] Y. Niimi and Y. Kobayashi: Conference Record of 1978-ICASSP, p.425 (1978).
- [8] N.J. Nilsson: Problem-Solving Methods in Artificial Intelligence (1971).
- [9] P.H. Winston: Artificial Intelligence (1977).
- [10] B.T. Lowerre: Ph.D. Thesis, Carnegie Mellon University (1976).
- [11] R.G. Goodman: Ph.D. Thesis, Stanford University (1976).
- [12] W.H. Paxton: IEEE Trans. Vol.ASSP-23, No.5, p.426 (1975).