

DELTA-MIN: A Search-Control Method for Information-Gathering Problems

Jaime G. Carbonell
Computer Science Department
Carnegie-Mellon University

Abstract

The Δ -MIN method consists of a best-first backtracking algorithm applicable to a large class of *information-gathering problems*, such as most natural language analyzers, many speech understanding systems, and some forms of planning and automated knowledge acquisition. This paper focuses on the general Δ -MIN search-control method and characterizes the problem spaces to which it may apply. Essentially, Δ -MIN provides a best-first search mechanism over the space of alternate interpretations of an input sequence, where the interpreter is assumed to be organized as a set of cooperating expert modules.¹

1. Introduction

A present trend in AI is to design large systems as cooperating collections of experts, whose separate contributions must be integrated in the performance of a task. Examples of such systems include HEARSAY-II [4], POLITICS [1], PSI [5], SAM, [3]. The division of task responsibility and grain size of the experts differs markedly. For instance, the latter two systems contain few large-scale experts, which are invoked in a largely predetermined order, while the former two systems contain a larger number of smaller modules whose order of invocation is an integral part of problem solving task itself.

In this paper I discuss a new search method, called Δ -MIN, that incorporates some of the desirable features from best-first search and some properties of gradient search. (Gradient search is locally-optimized hill-climbing.) The primary objective is to make *global control decisions based on local knowledge provided by each expert module*. No module is required to know either the internal structure of another module, or the overall controlling search mechanism. In this way, I depart somewhat from the freer blackboard control structure of HEARSAY-II, where search was controlled by the experts themselves. The module that "shouted loudest" was given control, hence each module had to know when and how loud to shout with respect to other expert modules. In addition, there was a "focus knowledge source" [6] that helped guide forward search. This method acquires its flexibility by placing a substantial amount of global control responsibility on local experts. Moreover, it entails no externally-transparent search discipline. Finally, the primary emphasis is on forward search, not reconsidering wrong decisions in favor of choosing an alternate interpretation. In light of these considerations, I attempted to factor domain knowledge (what the experts know) from search discipline (when to pursue alternate

paths suggested by different experts), so that each problem may be investigated in its own right. Here, I focus on the search control aspect, and consider the internal structure of each domain expert as a virtual "black box".

To simplify matters, I confine my discussion to tasks whose terminating condition is defined by processing an input sequence to completion without error. This class of problems is exemplified by natural language analysis, where an input sentence is processed left-to-right and the goal state is the formation of a consistent semantic representation of the input. Clearly, this is a satisficing rather than optimizing task [8], in the sense that only the first of potentially many solutions is sought. Since I want the language analysis to give the same parse of the sentence as a human would, the process must be biased to favor reaching the appropriate solution first. This biasing process, based on local decisions made by expert modules is the primary input to the Δ -MIN search method described below. It must be noted, however, that the left-to-right processing assumption is more restrictive than the HEARSAY paradigm, where "islands of interpretation" could grow anywhere in the input sequence and, when possible, were later merged into larger islands until the entire input sequence was covered [6, 4].

2. An Information-Gathering Search Space

Consider a search space for the task of processing a finite sequence of input symbols (such as an English sentence) and producing an integrated representation incorporating all the information extracted from the input (such as a semantic representation of the meaning encoded in the sentence). The problem solver consists of a set of experts that may be applied at many different processing stages, without fixed constraints on their order of application. For instance, in the language analysis domain, one can conceive of a verb-case expert, a morphological-transformation expert, an extra-sentential referent-identifier expert (or several such experts based on different knowledge sources), a dialog-context expert, an immediate-semantics expert, a syntactic-transformation expert, etc... A robust language analyzer must be capable of invoking any subset of these and other experts according to dynamically determined needs in analyzing the sentence at hand.

Now, let us back off the natural language domain and consider the general class of problem spaces to which an information-gathering,² cooperating-expert approach appears useful. First, we draw a mapping between the general problem solving terminology and the expert module approach. The search space outlined below is a considerably constrained version of a general search space. This property is exploited in the Δ -MIN search method described in the following section.

¹This research was sponsored in part by the Office of Naval Research (ONR) under grant number N00014-79-C-0661.

- The operators in the search space are the individual expert modules. Each module may search its own space internally, but I am concerned only with the macro-structure search space. Each expert has conditions of strict applicability and preference of applicability. The latter are used for conflict resolution decisions when more than one expert is applicable.
- A state in the space consists of the total knowledge gathered by invoking the set of experts that caused the state transitions from the initial state to the present. This definition has two significant implications: An expert that adds no new knowledge when invoked does not generate a new state; therefore, it can be ignored by the search control. There is a monotonicity property in that each step away from the initial states adds information to the analysis, and therefore is guaranteed to "climb" to a potential final state. (Left-to-right, single-pass natural language analysis can exhibit such monotonic behavior.)
- A final state is defined by having reached the end of the input sequence without violating a path constraint, and no expert can add more information (i.e., no transition in the state space is possible).
- A path constraint is violated if either a new segment of the input cannot be incorporated or an expert asserts information that contradicts that which is already part of the current state. When this situation arises, directed backtracking becomes necessary.
- The initial state is a (possibly empty) set of constraints that must be satisfied by any interpretation of the input sequence. For instance, a dialog or story context constrains the interpretation of an utterance in many natural language tasks.
- Each expert can draw more than one conclusion when applied. *Choosing the appropriate conclusion and minimizing backtracking on alternatives is where the real search problem lies.* Choosing the next expert to apply is not a real problem, as the final interpretation is often independent of the order of application of experts. That is, since information-gathering is in principle additive, different application sequences of the same experts should converge to the same final state. The experts preselect themselves as to applicability. Selecting the expert who thinks it can add the most information (as in HEARSAY-II) only tends to shorten the path to the final state. The real search lies in considering alternate interpretations of the input, which can only be resolved by establishing consistency with later information gathered by other experts. Finally, given the possibility of focused backtracking from a dead end in the forward search, less effort needs to be directed at finding the "one and only correct expert" to apply.

²"Information gathering" is a term coined by Raj Reddy to refer to search problems where progress towards a goal state is characterized by accruing and integrating information from outside sources.

3. The DELTA-MIN Search Method

Δ -MIN is a heuristic search method specifically tailored to the class of search spaces described above. It combines some of the more desirable features of gradient search and best-first search, with the modularized information sources of a cooperating-expert paradigm. Figure 3-0 is the search-control strategy algorithm in an MLISP-style form. Subsequently I discuss how Δ -MIN works, exemplifying the discussion with a sample search-tree diagram.

Every expert is responsible for assigning a likelihood-of-correctness value to each alternative in the interpretation it outputs. These values are only used to determine how much better the best alternative is than the next best alternatives, an item of information crucial to the backtrack control mechanism. There is no global evaluation function (the outputs of different experts are not directly comparable - it makes no sense to ask questions like: "Is this anaphoric referent specification better than that syntactic segmentation?") Nor is there any mechanism to compute differences between the present state and the goal state. (Recall our definition of goal state -- only upon completion of the input processing can the goal state be established.)

```

PROCEDURE  $\Delta$ -MIN(initial-state, experts)
  altlist := NULL
  globaldelta := 0
  state := initial-state
  input := READ(first input)
NEXTOP:
  IF NULL(input)
    THEN RETURN(state)
    ELSE operator := SELECTBEST(APPLICABLE(experts))
  IF NULL(operator)
    THEN input := READ(next input)
    ALSO GO NEXTOP
    ELSE alts := APPLY(operator, state)
  IF NULL(alts)
    THEN MARK(operator, 'NOT-APPLICABLE, 'TEMP)
    ALSO GO NEXTOP
  bestalt := SELECTMAX(alts)
  IF ||alts|| > 1
    THEN alts := FOR-EACH alt IN REMOVE(bestalt, alts)
      COLLECT <'ALT: alt,
              'STATE: state,
              'DELTA: globaldelta + VALUE(bestalt)
                - VALUE(alt)>
  altlist := APPEND(alts, altlist)
NEWSTATE:
  state := MERGE-INFORMATION(state, bestalt)
  IF NOT(state = 'ERROR')
    GO NEXTOP ;if no error, continue gradient search,
              ;else delta-min backup below
  WHILE state has no viable alternatives
    DO BEGIN
      MARK(state, 'DEAD-END, 'PERM) ;delete dead ends
      state := PARENT(state) ;from search tree
    END
  backup-point := SELECTDELTA-MIN(altlist)
  state := GET(backup-point, 'STATE:)
  globaldelta := GET(backup-point, 'DELTA:)
  bestalt := GET(backup-point, 'ALT:)
  altlist := REMOVE(backup-point, altlist)
  GO NEWSTATE
END  $\Delta$ -MIN

```

Figure 3-1: The Δ -MIN Search-Control Algorithm

Let us see how Δ -MIN can be applied to an abstract example, following the diagram in figure 3-2. The roman numerals in the arcs reflect the order in which they are traversed. At the initial state, expert-4 applies and generates three alternate interpretations of the input. One alternative is ranked as most likely. A " Δ " value is computed for the remaining alternatives, encoding the difference in confidence that expert-4 had between them and the most likely alternative. The more sure expert-4 is of its best choice relative to the other alternatives, the larger the Δ

values. The best interpretation generated by expert-4 is integrated with the initial state constraints and found to be consistent. At this point, a new state has been generated and expert-2 applies to this state generating no new information. More input is read and expert-1 applies, generating only one alternative, which is found to be consistent with the information in the present state. In a similar fashion, the rest of the tree in figure 3-2 is generated.

Up to now, we have witnessed an instance of gradient search, where a different evaluation function is applied at each node (The local evaluation function is, in effect, the expert who generated the likelihood values.) If no error occurs, (i.e., if the interpretation of the input remains consistent) no backup is needed. The likelihood rankings clearly minimize the chance of error as compared to straightforward depth first search. Now, let us consider the possibility of an inconsistency in the interpretation, as we continue to examine figure 3-2.

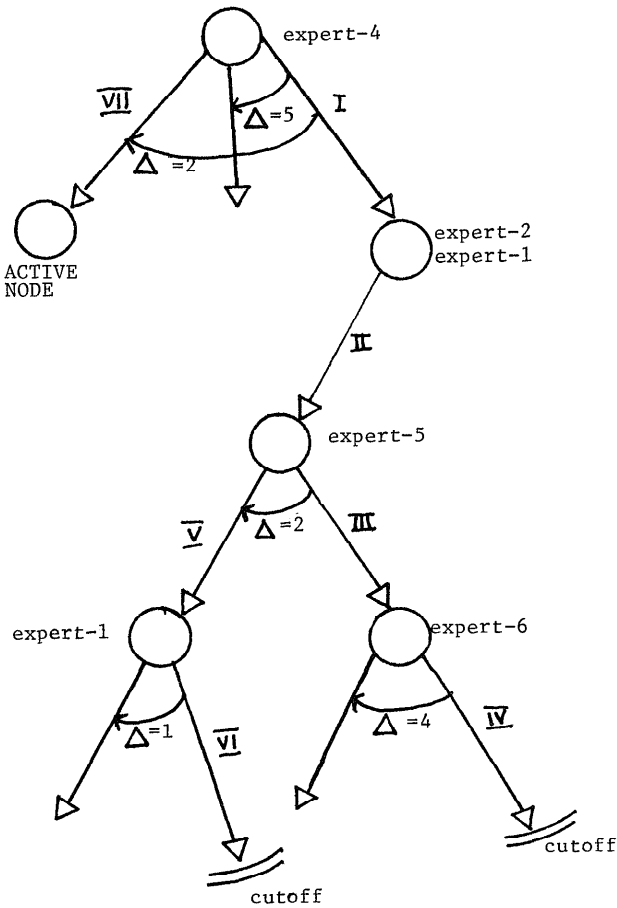


Figure 3-2: Δ -MIN Search Tree With Directed Backup

The most likely interpretation generated by expert-6 was found to be inconsistent with the information in the present state. Backup is therefore necessary, given the depth-first nature of the search. But, where do we back up to? Normally, one might consider a depth-first unwinding of the search tree; but, is this the most reasonable strategy? Expert-5 was much less certain in its choice of best alternative than expert-6 ($\Delta = 2$ vs $\Delta = 4$). It seems

more reasonable to doubt expert-5's decision. Therefore, one wants to back up to the point where the probability of having chosen a wrong branch is highest, namely to the choice point with the minimal Δ (hence the name Δ -MIN).

Continuing with figure 3-2, we restore the state at expert-5 and incorporate the $\Delta = 2$ interpretation. It is found to be consistent, and we apply expert-1 to the new state. The best interpretation of expert-1 leads to error, and backup is again required. Where to now? The minimal Δ is at expert-1, but this would mean choosing a non-optimal branch of a non-optimal branch. Lack of confidence in the choice from expert-5 should be propagated to the present invocation of expert-1. Hence, we add the two Δ s in the path from the initial state and get the value: $\Delta = 3$, which is greater than the minimal Δ at expert-4 ($\Delta = 2$). Therefore, we back up to expert-4. This process continues until a consistent interpretation of the entire input is found (i.e., a goal state is reached), or the search exhausts all viable alternate interpretations.

Essentially, Δ -MIN is a method for finding one globally consistent interpretation of an input sequence processed in a predetermined order. In natural language analysis, the problem is to find a semantically, syntactically, and contextually consistent parse of a sentence. In speech understanding the constraint of formulating legal phonemes and words is added, but the nature of the problem and the applicability of the Δ -MIN approach remains the same. For instance, Δ -MIN is an alternate control structure to HARPY's beam search [7], which also processes a sequence of symbols left to right, seeking a globally consistent interpretation.

4. Concluding Remarks

To summarize, the Δ -MIN method exhibits the following properties:

- Δ -MIN is equivalent to gradient search while no error occurs. Path length (from the initial state) is not a factor in the decision function.
- The backtracking mechanism is directed to undo the choice most likely to have caused an interpretation error. This method compares all active nodes in the tree, as in best-first search, but only when an error occurs (unlike best-first search).
- Perseverance in one search path is rewarded, as long as the interpretation remains consistent, while compounding less-than-optimal alternate choices is penalized. This behavior falls out of the way in which Δ values are accrued.
- No global evaluation function forces direct comparisons among information gathered by different knowledge sources. Such an evaluation function would necessarily need to encode much of the information contained in the separate experts, thus defeating the purpose of a modular cooperating-expert approach. The Δ comparisons contrast only the differences between locally-optimal and locally-suboptimal decisions. These differences are computed by local experts, but the comparisons themselves are only between relative ratings on the desirability of alternate decisions.

Additional discussion of implementation, analysis, and details of the Δ -MIN search method may be found in [2], where an effective application of Δ -MIN is discussed for constraining search in a natural language processing task.

References

1. Carbonell, J. G., "POLITICS: An Experiment in Subjective Understanding and Integrated Reasoning," in *Inside Computer Understanding: Five Programs Plus Miniatures*, R. C. Schank and C. K. Riesbeck, eds., New Jersey: Erlbaum, 1980.
2. Carbonell, J. G., "Search in a Non-Homogeneous Problem Space - The Δ -MIN Algorithm," Tech. report, Dept. of Computer Science, Carnegie-Mellon University, 1980.
3. Cullingford, R., *Script Application: Computer Understanding of Newspaper Stories*, PhD dissertation, Yale University, Sept. 1977.
4. Erman L. D. and Lesser, V. R., "HEARSAY-II: Tutorial Introduction & Retrospective View," Tech. report, Dept. of Computer Science, Carnegie-Mellon University, May 1978.
5. Green, C. C., "The Design of the PSI Program Synthesis System," *Proceedings of the Second International Conference on Software Engineering*, October 1976, pp. 4-18.
6. Hayes-Roth, F. and Lesser, V. R., "Focus of Attention in the Hearsay-II Speech Understanding System," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 1977, pp. 27-35.
7. Lowerre, B., "The HARPY Speech Recognition System," Tech. report Computer Science Department, Carnegie-Mellon University, April 1976.
8. Newell, A. and Simon, H. A., *Human Problem Solving*, New Jersey: Prentice-Hall, 1972.