KYBERNETIKA — VOLUME 7 (1971), NUMBER 5

On Parameters of an Evaluation Function for Heuristic Search as a Path Problem*

ĽUDOVÍT MOLNÁR

The analysis about parameters of an evaluation function for heuristic search as a path problem is done. It is shown that parameters should be not constants but functions which change their values during search of a problem. The theoretical results were reached in the worst case analysis and verified by experiments in which the knowledge of the problem space was used.

INTRODUCTION

Many problems of Artificial Intelligence can be generally represented by a set of discrete states of a problem and a set of operators which can be applied to them. The task is to find a sequence of operators which produce a correct or asked sequence of states as a solution to a problem.

To describe such problem we will use a graph which contains a set of nodes corresponding to discrete states of a problem and a set of edges between nodes corresponding to operators. We distinquish one node as the initial and try to find a path to another designated the goal node. Such path is called a solution to our problem.

When we solve this kind of problem we can be seeking:

a) the shortest path,
b) the minimum number of nodes produced.

Our work will be oriented to the second aspect.

For finding a path Doran and Michie [1] have developed a general problem solving algorithm (Graph Traverser) which uses an evaluation function to direct search for the goal node. An evaluation function is purely heuristic and estimates a distance to the goal node.

* The paper was written while the author was the Visiting Research Worker at the Department of Machine Inteligence and Perception of the University of Edinburgh as the student of the British Council.

Hart, Nilsson and Raphael [4] have proposed a compound evaluation function. The first part is a current distance from the start node to any node x and the second one is an estimated distance from a node x to the goal node.

Pohl [6] has done experiments with a weighting of parameters for each part of an evaluation function and discovered that improvement in a solution can be reached for some values of parameters.

Similar experiments have been done also by Michie and Ross [5] who have done the optimalization of parameters of the heuristic part of an evaluation function. However, till now no analysis about a weighting of parameters of an evaluation function has been done and "the best" values for them were found only by experiments. The purpose of our paper is to show some relations between parameters and the heuristic part of an evaluation function. At first we will define the algorithm for heuristic search which is similar to the algorithms used in [1], [4] and [6].

THE PROBLEM SPACE AND ALGORITHM FOR HEURISTIC SEARCH

We will consider a graph

$$G = \{X, \Omega\}$$

where X is a set of nodes corresponding to discrete states of a problem, Ω is a predicate over $X \times X$.

A path over G is defined as a sequence $x_0, x_1, \ldots, x_k$ from X for which $\Omega(x_i, x_{i+1})$ is true i.e. there is the edge between x_i, x_{i+1} for $i = 0, 1, \ldots, k - 1$. We can define $x_0 = s$ as the start node and $x_k = t$ as the goal or terminal node. For the set of all immediate successors we define an operator Γ such that

$$\Gamma(x_i) = X_i = \{x / \Omega(x_i, x)\} .$$

We define also an operator set $\Gamma' = \{\Gamma'_1, \Gamma'_2, \ldots, \Gamma'_m\}$ on which is imposed an ordering such that an index gives a place in Γ', and where m is a number of operators. Γ'_i is (possible partial) function $\Gamma'_i : X \to X$. (Γ'_i is a function with domain X and codomain X or Γ'_i is a function on X to X.)

The relation between Γ' and Γ is:

$$\Gamma(x_i) = X_i = \bigcup_{j=1}^{n} \Gamma'_j(x_i)$$

where $\bigcup_{j=1}^{n} \Gamma'_j$ is union of all applicable operators.

If an operator is applied to any node x_i a new node, say x_{i+1} is produced. We say that x_{i+1} is the successor of x_i or x_i is the parent (predecessor) of x_{i+1}. If all applicable operators were applied to a some node x_j we say that a node x_j is fully developed. Otherwise a node x_j is partially developed.

A search for a solution is directed by an evaluation function $f : X \to R$ (a set of reals).

We will use an evaluation function in the form:

$$f(n) = \omega\, g(n) + \omega'\, h(n)$$

where ω, ω' are parameters, $g(n)$ is a number of edges from the start node to a node n, $h(n)$ is an estimate of a number of edges from a node n to the goal node.

Let s be the start node, t the goal node, S a set of fully developed nodes, S' a set of partially developed nodes. Γ_i' the first applicable operator for a corresponding node which wasn't applied on it yet.

The algorithm which will be used can be now written:

1. Place s in S' and assign $f(s)$ to it.
2. Select $n_{f_{\min}}$ from S' such that $f(n)$ is minimum.
3. If $S \cup S'$ is greater than limit then stop.
4. Apply Γ_i' to $n_{f_{\min}}$ and put it in S', assign $f(\Gamma_i'(n_{f_{\min}}))$ to $\Gamma_i'(n_{f_{\min}})$.
5. If $\Gamma_i'(n_{f_{\min}})$ is the goal node then stop.
6. If $n_{f_{\min}}$ is fully developed then take it from S' and put it in S.
7. Go to 2.

Note. In addition a value of an evaluation function it is attached to each node a pointer to its parent (predecessor) which allows us to trace a path from the start node to any other node.

THE RELATION BETWEEN PARAMETERS AND HEURISTIC PART OF AN EVALUATION FUNCTION

For our purpose we will use the easy analysable space the regular binary tree i.e. the tree, every node of which has two successors (Fig. 1).

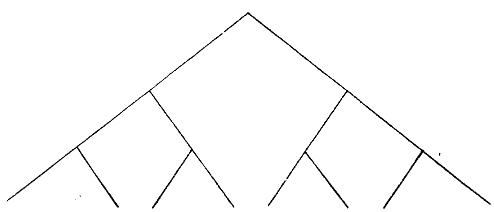

Fig. 1.

We will carry out a worst case analysis in the spirit of the error analysis in numerical problems. Let $h'(n)$ be a perfect estimator, ε a bound on the error $0, 1, 2, 3, \ldots,$

$h(n)$ a given heuristic function,

$$h'(n) - \varepsilon \leqq h(n) \leqq h'(n) + \varepsilon .$$

To make $h(n)$ as bad as possible we add ε to each node on the shortest solution path and subtract ε from each node off the shortest solution path. Suppose that the shortest solution path is k steps long.

If t is the goal node we can write:

$$f(t) = \omega\, g(t) + \omega'\, h(t) = \omega\, g(t) + \omega'\, h'(t) + \omega'\varepsilon = \omega k + \omega'\varepsilon .$$

For any other node n off the shortest solution path we can write:

$$f(n) = \omega\, g(n) + \omega'\, h(n) = \omega\, g(n) + \omega'\, h'(n) - \omega'\varepsilon .$$

Since we must visit all nodes on and off the shortest solution path whose value is less (in the case of ties i.e. nodes with an equal value of an evaluation function we choose "the worst" node) than a value of the goal node, therefore

$$f(n) > f(t) ,$$

$$\omega\, g(n) + \omega'\, h'(n) - \omega'\varepsilon > \omega k + \omega'\varepsilon ,$$

$$g(n) + \frac{\omega'}{\omega}\, h'(n) > k + 2\, \frac{\omega'}{\omega}\, \varepsilon .$$

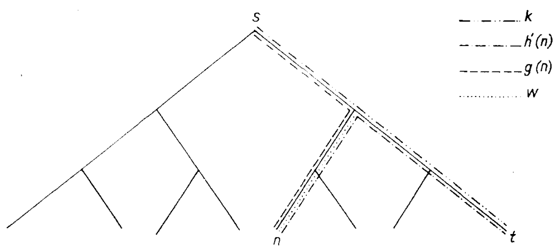

Fig. 2.

Let w be a number of steps off the shortest solution path. Then (see Fig. 2)

$$g(n) + h'(n) = k + 2w$$

and

$$g(n) + h'(n) - h'(n) + \frac{\omega'}{\omega}\, h'(n) > k + 2\, \frac{\omega'}{\omega}\, \varepsilon ,$$

$$2w + k + h'(n)\left(\frac{\omega'}{\omega} - 1\right) > k + 2\frac{\omega'}{\omega}\varepsilon\,,$$

$$w > \frac{\omega'}{\omega}\varepsilon - \frac{h'(n)}{2}\left(\frac{\omega'}{\omega} - 1\right).$$

Since we want to have w as small as possible we put

$$w = 1 + \frac{\omega'}{\omega}\varepsilon - \frac{h'(n)}{2}\left(\frac{\omega'}{\omega} - 1\right),$$

(1)
$$w = 1 + \frac{h(n)}{2} + \frac{\varepsilon}{2} - \frac{\omega'}{\omega}\left(\frac{h(n)}{2} - \frac{\varepsilon}{2}\right).$$

From (1) we can see that for given $h(n)$ and ε we can find such ω, ω' which give us the least value of w i.e. such an evaluation function which for a given heuristic function and a bound on the error visits the least number of nodes off the shortest solution path. If we put e.g. ω equal constant then we can find the best ω' and the contrary.

From (1) we can see also another interesting property of parameters ω, ω'. Since $h(n)$ changes its value with a change of n, it means that ω, ω' should be not constants but functions which change their values with a number of steps of a node n from the start node.

To have w as small as possible in a first part of search where the expression

$$\frac{h(n)}{2} - \frac{\varepsilon}{2}$$

has a value greater than 0, we must have

$$\omega' > \omega$$

in a relation which gives us a minimum value of w. Since w is a number of steps off the shortest solution path it must be $w \geqq 0$. It means that in a first part of search we should put a greater weight on the heuristic part of an evaluation function $h(n)$ and in a last part of search on $g(n)$.

These results, in spite of the fact that they were reached in a worst case analysis, gives us indications how to weight parameters also in a real problem space.

In our analysis we have used a constant bound on the error in each step of search. A similar analysis could be done with a bound on the error as a function. This would be more reasonable and closer to a real problem space. The results reached in this way indicate the same conclusion as the results reached above. In the same way we get

(2)
$$w = 1 + \frac{h(n)}{2} + \frac{\varepsilon(n)}{2} + \frac{\omega'}{2\omega}(\varepsilon(t) - h(n))$$

where $\varepsilon(n)$, $\varepsilon(t)$ are bounds on the error for nodes n and t respectively.

For our experiments we have chosen as the problem space the Fifteen puzzle (Fig. 3), the problem space complex enough (there are 15!/2 possible solvable positions) and which is well known from [1], [5], [6]. As the heuristic part of an evaluation function we have used

$$h(n) = \sum_{i=1}^{15} p_i^2$$

where p_i is a number of moves (steps) of an i-th piece from the home position. For

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	--

Fig. 3.

deriving of parameters we have used so called "reversals" i.e. positions of two pieces which are in a reverse order (in the column or line) and are in their "home" column or line. For reversals we can write:

$$R(n) = \sum_{j=1}^{15} p_j$$

where p_j is equal 1 if a piece j is in the reverse order with another piece, otherwise p_j is equal 0.

Parameters were derived from (2), but $\varepsilon(t)$ was put equal $\varepsilon(n)$ and ω was put equal 1.

Since we want to produce a minimum number of nodes we want to have w as small as possible i.e. $w = 0$. Then from (2) we can write

$$\frac{\omega'}{2\omega} = \frac{1 + \frac{h(n)}{2} + \frac{\varepsilon(n)}{2}}{h(n) - \varepsilon(n)},$$

$$\omega' = \frac{h(n) + \varepsilon(n) + 2}{(hn) - \varepsilon(n)}.$$

The evaluation function which has been used is

$$f(n) = g(n) + \frac{h(n) + \varepsilon(n) + 2}{h(n) - \varepsilon(n)} h(n)$$

where $g(n)$ is a number of edges from the start position to a position n, $h(n)$ is an estimate number of edges from a position n to the goal position, $\varepsilon(n)$ is equal $R(n)$ and $R(n) = \sum_{j=1}^{15} p_j$ defined as above.

This evaluation function was tested on 50 randomly generated Fifteen puzzles. The size of a partial tree was 200 nodes, resignation occered when 500 nodes were encountered.

The results were compared with the results reached with the evaluation function without weighting parameters (parameters equal 1) i.e.

$$f(n) = g(n) + h(n) .$$

The both results are in the table 1.

Table 1.

	Sample size	% of puzzles solved	The average number of nodes encountered/puzzle
without weighting parameters	50	4	493
with weighting parameters	50	48	402

CONCLUSION

The analysis about parameters of an evaluation function for heuristic search as a path problem was done. The results showed that parameters should be not constants as they were used till now, but functions which change their values during search of a problem. The theoretical results were verified by experiments in which the knowledge of the problem space was used for deriving of weighting parameters. The improvement which was reached in the solution showed us an applicability of the theoretical results to the real problem space.

Acknowledgements. The author wishes to thank to Professor Donald Michie — Professor of the Department of Machine Intelligence and Perception University of Edinburgh, under whose supervision the work was done, Dr. R. Kowalski and Mr. R. Ross of the same Department for many helpfull discussions and criticisms.

(Received October 14, 1970.)

REFERENCES

[1] Doran, J. E., Michie, D.: Experiments with the Graph Traverser program. Proc. R. Soc. *294* (1966), 1437, 235—259.
[2] Doran, J. E.: An approach to automatic problem solving. Machine Intelligence 1 (eds. N. L. Collins, D. Michie). Oliver and Boyd, Edinburgh 1967.
[3] Doran, J. E.: New developements of the Graph Traverser. Machine Intelligence 2 (eds. E. Dale, D. Michie). Oliver and Boyd, Edinburgh 1968.
[4] Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. I.E.E.E. Trans. on Sys. Sci. and Cybernetics *4* (1968), 2, 100—107.
[5] Michie, D., Ross, R.: Experiments with the Adaptive Graph Traverser. Machine Intelligence 5 (eds. B. Meltzer, D. Michie). Edinburgh University Press, Edinburgh 1970.
[6] Pohl, I.: First results on the effect of error in heuristic search. Machine Intelligence 5 (eds. B. Meltzer, D. Michie). Edinburgh University Press, Edinburgh 1970.

VÝŤAH

O parametroch vyhodnocovacej funkcie pre heuristické hľadanie ako problém cesty

ĽUDOVÍT MOLNÁR

V článku sa zaoberáme otázkou parametrov vyhodnocovacej funkcie pre heuristické hľadanie ako problém cesty. Ich analýzou pre najhorší prípad sme prišli k uzáveru, že by to nemali byť konštanty, ale funkcie, ktoré menia svoju hodnotu v priebehu hľadania cesty, pričom zmena by mala prebiehať tak, aby sa na začiatku hľadania kládla väčšia váha na heuristickú časť vyhodnocovacej funkcie, než na časť neheuristickú a táto by sa postupne zmenšovala.

Teoretické uzávery boli overované experimentálne a výsledky potvrdili ich správnosť.

Ľudovít Molnár, prom. fyz., Katedra matematických strojov EF SVŠT (Department of Computer Science - Slovak Technical University), Vazovova 1/b, Bratislava.