Including g tends to add a breadth-first component to the search and thus ensures that no part of the implicit graph will go permanently unsearched.

The relative weights of g and h in the evaluation function can be controlled by using $f = g + wh$, where w is a positive number. Very large values of w overemphasize the heuristic component, while very small values of w give the search a predominantly breadth-first character. Experimental evidence suggests that search efficiency is often enhanced by allowing the value of w to vary inversely with the depth of a node in the search tree. At shallow depths, the search relies mainly on the heuristic component, while at greater depths, the search becomes increasingly breadth-first, to ensure that some path to a goal will eventually be found.

To summarize, there are three important factors influencing the heuristic power of Algorithm **A**:

(a) the cost of the path,

(b) the number of nodes expanded in finding the path, and

(c) the computational effort required to compute h.

The selection of a suitable heuristic function permits one to balance these factors to maximize heuristic power.

2.5. RELATED ALGORITHMS

2.5.1. BIDIRECTIONAL SEARCH

Some problems can be solved using production systems whose rules can be used in either a forward or a backward direction. An interesting possibility is to search in both directions simultaneously. The graph-searching process that models such a bidirectional production system can be viewed as one in which search proceeds outward simultaneously from both the start node and from a set of goal nodes. The process terminates when (and if) the two search frontiers meet in some appropriate fashion.