Planning by Guided Hill-Climbing

Seyed Ali Akramifar and Gholamreza Ghassem-Sani

Computer Engineering Department, Sharif University of Technology, P.O. Box 11365-9517, Tehran, Iran
akrami@ce.sharif.edu, sani@sharif.edu

Abstract. This paper describes a novel approach will be called guided hill climbing to improve the efficiency of hill climbing in the planning domains. Unlike simple hill climbing, which evaluates the successor states without any particular order, guided hill climbing evaluates states according to an order recommended by an auxiliary guiding heuristic function. Guiding heuristic function is a self-adaptive and cost effective function based on the main heuristic function of hill climbing. To improve the performance of the method in various domains, we defined several heuristic functions and created a mechanism to choose appropriate functions for each particular domain. We applied the guiding method to the enforced hill climbing, which has been used by the Fast Forward planning system (FF). The results show a significant improvement in the efficiency of FF in a number of domains.

1 Introduction

Hill climbing is a local search technique, which has been widely used in artificial intelligence fields such as AI planning [1], machine learning [2], and optimization [3, 4]. It attempts to minimize (or maximize) a function \( h(s) \), where \( s \) are discrete states. These states are typically represented by vertices (or nodes) in a graph, where edges (or actions) in the graph indicate nearness or similarity of vertices. Hill climbing traces the graph vertex by vertex, locally decreasing (or increasing) the value of \( h \), until a local minimum (or maximum) \( s_m \) is reached.

Two major approaches of hill climbing are the so-called simple hill climbing and steepest ascent hill climbing. In simple hill climbing, the first closer node to the solution is chosen, whereas in the steepest ascent hill climbing; all successors are compared and the closest node to the solution is chosen. Simple hill climbing is more efficient than the steepest ascent hill climbing specially in domains with a high branching factor using a costly evaluation function. Therefore, it is preferred to use simple hill climbing in the planning domains.

Many research efforts have paid attention to improve efficiency of hill climbing. Examples include dynamic hill climbing, which uses a genetic algorithm technique to hill climbing [5], stochastic hill climbing, which uses a random order to evaluate successor states [6, 7], and hill climbing augmented with learning techniques [8].

The usual approach to overcome the efficiency problem is looking for a more efficient evaluation function [9]. To reach to the goal state faster, the evaluation
function needs to be more cost effective and more accurate. However, since we need to design a new evaluation function for each new domain, this is not a generally useful solution. Using more efficient domain independent heuristic functions is another important solution to reduce the efficiency problem [4]. Although, the domain independent option is more attractive than the previous one, it has its own limitations. First, because of difference in the nature of domains, usually the performance of a domain independent heuristic function varies in different domains. In other words, a domain independent heuristic function cannot significantly improve the efficiency of hill climbing in some domains. In addition, there is a tradeoff between the efficiency and the accuracy of a heuristic function. Usually, informed heuristic functions need more computational resorts and consequently are less efficient. On the other hand, uninformed but efficient heuristic functions evaluate states inappropriately. Therefore, this tradeoff reduces the total efficiency of the search.

We introduced a new approach called guided hill climbing (GHC), to improve the efficiency of hill climbing. In GHC, an auxiliary cost effective heuristic function $g$ is used to order successor states. As simple hill climbing, GHC uses a primary heuristic function $h$ to evaluate successor states according to the order proposed by $g$.

We applied GHC to the enforced hill climbing (EHC), which is the main search strategy of the Fast Forward (FF) planning system [10]. EHC is a combination of a systematic search method and simple hill climbing. We tested the efficiency and quality of new planner in a number of domains. Also, to improve the performance of GHC in various domains, we defined several heuristic functions and implemented a mechanism to automatically learn appropriate function $g$ in each domain.

The remainder of this paper is organized as follows. Section 2 details the new approach to hill climbing. Section 3 explains our new planner, which employs GHC. We implemented this planner through applying GHC to FF. Section 4 presents results of several experiments demonstrating the advantages of GHC. This section also compares results of our planner with that of FF. Last section concludes.

2 Guided Hill Climbing

GHC is a different approach to overcome the efficiency problem of the simple hill climbing. Its search strategy is very similar to that of simple hill climbing in some ways. First, GHC employs an evaluation function to evaluate states. Furthermore, it searches to find a better successor state. Finally, GHC has basic problems of simple hill climbing such as facing a local maximum. However, it is different from simple hill climbing in that it uses an extra heuristic function called guiding heuristic function. By the means of this function, GHC orders successor states before evaluation by the main heuristic function of hill climbing. GHC algorithm is as follows:

Guided Hill-Climbing (state $s_{in}$): state $s_{out}$
Let Open = $\{s, ..., s_i\}$ be successor states of $s_{in}$
While (Open is not empty) do
    Let $s_i = g(\{s, ..., s_i\})$
    if ($h(s_i) < h(s_c)$) return $s_i$
    else remove $s_i$ from Open
    add successor states of $s_i$ to Open
return NULL