Hill Climbers and Mutational Heuristics in Hyperheuristics

Ender Özcan, Burak Bilgin, and Emin Erkan Korkmaz

Yeditepe University,
Department of Computer Engineering,
34755 Kadıköy/İstanbul, Turkey
{eozcan, bbilgin, ekorkmaz}@cse.yeditepe.edu.tr

Abstract. Hyperheuristics are single candidate solution based and simple to maintain mechanisms used in optimization. At each iteration, as a higher level of abstraction, a hyperheuristic chooses and applies one of the heuristics to a candidate solution. In this study, the performance contribution of hill climbing operators along with the mutational heuristics are analyzed in depth in four different hyperheuristic frameworks. Four different hill climbing operators and three mutational operators are used during the experiments. Various subsets of the heuristics are evaluated on fourteen well-known benchmark functions.

1 Introduction

The term hyperheuristics refers to a recent approach in search methodologies [2, 4, 5, 7, 17, 23]. The hyperheuristic concept involves a higher level of abstraction than metaheuristic methods. This term describes an iterative search approach which controls a set of heuristics. The method keeps track of the non problem-specific data such as the fitness change, the execution time and applies a heuristic at each iteration. Studies involving a number of heuristic selection and acceptance mechanism combinations are reported in the literature [2, 3, 4, 7, 17]. A comprehensive study on the performance of different heuristic selection and move acceptance strategies is reported in [3].

In this paper, the synergy of various heuristics and their contribution to the performance is evaluated on a set of benchmark functions. Furthermore, four different hyperheuristic frameworks that utilize a set of hill climbers as heuristics in addition to a set of mutational heuristics, are defined and assessed as well. The new frameworks are derived from the commonly used framework. The intention of this study is to answer the following questions: What type of heuristics is useful to be used in hyperheuristics? Do the hill climbers improve the performance if used within hyperheuristics? Can we use only hill climbers as heuristics? At which stage(s) and how can hill climbers be used to improve the performance? Is it possible to identify the problem domains where a specific framework might perform better as compared to the others?

2 Preliminaries

In general, exhaustive methods are impractical for solving real world problems, whereas meta-heuristics provide better means by intelligently seeking optimal solutions within

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 202 – 211, 2006.

a search space. For many practical problems meta-heuristics provide state-of-the-art solutions. Their success is due to the problem-specific implementations, which utilize knowledge about the problem domain and properties. The deployment of meta-heuristics requires expert level knowledge and experience on the problem tackled. Furthermore, fine tuning might be required [4, 23]. Hyperheuristics are general search methods that can be applied to any optimization problem easily [7]. Hyperheuristics describe a set of strategies that are used to choose a heuristic from a set of low level heuristics as illustrated in Fig. 1. There are very simple strategies that can be coded easily. Yet, a meta-heuristic can be used as a heuristic underneath a hyperheuristic as well as a hyperheuristic itself within this framework.

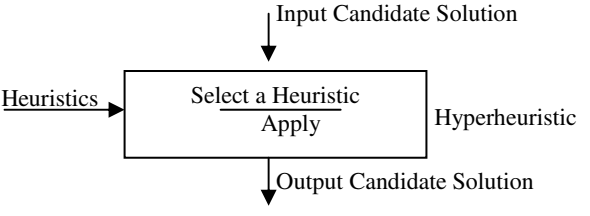

Fig. 1. Traditional hyperheuristic framework

Hyperheuristics operate on the search space of heuristics instead of candidate solutions. Non problem-specific data like heuristic execution time and changes in the fitness function can be used by hyperheuristics to select and apply a heuristic [2]. Although the methods of this type are reported in the literature before, the term hyperheuristic is first proposed by Cowling et al. [7] to name this approach. The early studies date back to Fisher and Thompson. They used a hyperheuristic based on probabilistic weighting of heuristics to solve the job-shop scheduling problem [12]. Kitano [19] used a genetic algorithm as a hyperheuristic for designing neural network topology. The hyperheuristic approach is utilized by Gratch et al. [15] to schedule earth-orbiting satellites and ground stations communications. Fang et al. [11] utilized this approach using the genetic algorithm to tackle the open-shop problem. Hart and Ross [17] tackled the dynamic job-shop problem with a similar approach. Hyperheuristics are applied to university exam timetabling problems by Terashima-Marin et al. [25].

A single iteration of a hyperheuristic method can be decomposed in two stages, heuristic selection and movement acceptance. In the previous studies, hyperheuristics might be named without discriminating between heuristic selection and acceptance criterion. Examples of heuristic selection methods are Simple, Greedy, Choice Function [7], Tabu-Search [5], and Case Based Heuristic Selection Methods [6]. Simple Hyperheuristics utilize randomized processes to select heuristics. Greedy Hyperheuristic chooses the best performing heuristic at each iteration. Choice Function Hyperheuristic keeps track of previous performance of each heuristic and makes a choice between them by evaluating their performance via a choice function. Two types of deterministic acceptance criteria are used in [5, 7]: All Moves Accepted (AM) and Only Improving Moves Accepted (OI). Non-deterministic acceptance criteria can be