Using the k-Nearest Problems for Adaptive Multicriteria Planning

Grigorios Tsoumakas, Dimitris Vrakas, Nick Bassiliades, and Ioannis Vlahavas

Department of Informatics, Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
{greg,dvrakas,nbassili,vlahavas}@csd.auth.gr
http://lpis.csd.auth.gr/

Abstract. This paper concerns the design and development of an adaptive planner that is able to adjust its parameters to the characteristics of a given problem and to the priorities set by the user concerning plan length and planning time. This is accomplished through the implementation of the k nearest neighbor machine learning algorithm on top of a highly adjustable planner, called HAP. Learning data are produced by running HAP offline on several problems from multiple domains using all value combinations of its parameters. When the adaptive planner HAP_{NN} is faced with a new problem, it locates the k nearest problems, using a set of measurable problem characteristics, retrieves the performance data for all parameter configurations on these problems and performs a multicriteria combination, with user-specified weights for plan length and planning time. Based on this combination, the configuration with the best performance is then used in order to solve the new problem. Comparative experiments with the statistically best static configurations of the planner show that HAP_{NN} manages to adapt successfully to unseen problems, leading to an increased planning performance.

1 Introduction

In domain independent heuristic planning there is a number of systems that their performance varies between best and worse on a number of toy and real-world planning domains. No planner has been proved yet to be the best for all kinds of problems and domains. Similar instability in their efficiency is also noted when different variations of the same planner are tested on the same problem, when the value of one or more parameters of the planner is changed. Although most planners claim that the default values for their options guarantee a stable and averagely good performance, in most cases fine tuning the parameters by hand improves the performance of the system for the given problem.

Few attempts have been made to explain which are the specific dynamics of a planning problem that favor a specific planning system and even more, which is the best setup for a planning system given the characteristics of the planning problem. This kind of knowledge would clearly assist the planning community in producing flexible systems that could automatically adapt themselves to each problem, achieving best performance.

G.A. Vouros and T. Panayiotopoulos (Eds.): SETN 2004, LNAI 3025, pp. 132–141, 2004.

Some promising past approaches towards this goal, followed the methodology of utilizing Machine Learning in order to infer rules for the automatic configuration of planning systems [1],[2]. However, these approaches exhibited two important problems. The first one is that they used a fixed policy for what can be considered as a good solution to a planning problem and didn't allow users to specify their own priorities concerning the speed of the planner and the quality of the plans, which are frequently contradictious. The second one is that learning is very computationally expensive and thus extending the knowledge base of the planner is a non-trivial task.

This paper presents a different approach to adaptive planning that is based on instance-based learning in order to deal with the two aforementioned problems. Specifically, the k nearest neighbor machine learning algorithm is implemented on top of the HAP highly adjustable planner. Learning data are produced by running HAP offline on 30 problems from each one of 15 domains (i.e. 450 problems) using 864 combinations of values for its 7 parameters. When the adaptive planner HAP_{NN} is faced with a new problem, it retrieves the steps and time performance data for all parameter configurations of the k nearest problems and performs a multi-criteria combination, with user-specified weights. The best configuration is then used for running the planner on the new problem. Most importantly, the planner can store new problems and train incrementally from them, making the system highly extensible.

The performance of HAP_{NN} was thoroughly evaluated through experiments that aimed at showing the behavior of the adaptive system in new problems. The results showed that the system managed to adapt quite well and the use of different weights for steps and time had the expected effect on the resulting plan length and planning time of the adaptive planner.

The rest of the paper is organized as follows: Section 2 overviews related work combining Machine Learning and Planning. The planning system used for the purposes of our research and the problem analysis done for deciding the problem attributes are presented in Section 3 and 4 respectively. Section 5 describes in detail the methodology we followed for designing the adaptive planner. The experimental results are presented and discussed in the Section 6 and finally, Section 7 concludes the paper and poses future research directions.

2 Related Work

Machine learning has been exploited extensively in the past to support Planning systems in many ways. There are three main categories of approaches based on the phase of planning that learning is applied to and the consequent type of knowledge that is acquired.

Domain knowledge is utilized by planners in pre-processing phases in order to either modify the description of the problem in a way that will make it easier for solving it or make the appropriate adjustments to the planner to best attack the problem [1].