Universidade Federal de Viçosa

Campus Viçosa

Centro de Ciencias Exactas e Tecnologicas (CCE)

Mestrado em Ciencia da Computacão

On Selecting of Heuristics Functions for Domain–Independent Planning

AUTHOR: Marvin Abisrror Zarate

ADVISOR: PhD. Levi Henrique Santana de Lelis
CO–ADVISOR: PhD. Santiago Franco

Viçosa (MG), 01 of December of 2015

This thesis is dedicated to my Mother.

Acknowledgements

I would like to express my sincere gratitude to my advisor PhD. Levi Henrique Santana de Lelis, for the continuous support and guidance during the thesis process. His valuable advice, patience and encouragement have been of great importance for this work.

Besides my advisor, I would like to thank to my co−advisor: PhD. Santiago Franco for his insightful feedback, interest and tough questions.

To the professors of the DTI, particularly the Master Degree program with all its members, played an invaluable role in my graduate education.

Last, but not least, I would like to thank my Mother.

Abstract

In this dissertation we present a greedy method based on the theory of supermodular optimization for selecting a subset of heuristics functions from a large set of heuristics with the objective of reducing the running time of the search algorithms.

Holte et al. [2006] showed that search can be faster if several smaller pattern databases are used instead of one large pattern database. We introduce a greedy method for selecting a subset of the most promising heuristicss from a large set of heuristics functions to guide the A* search algorithm. If the heuristics are consistent, our method selects a subset which is guaranteed to be near optimal with respect to the resulting A* search tree size. In addition to being consistent, if all heuristics have the same evaluation time, our subset is guaranteed to be near optimal with respect to the resulting A* running time. We implemented our method in Fast Downward and showed empirically that it produces heuristics which outperform the state of the art heuristics in the International Planning Competition benchmarks.

Table of Contents

Dedication I

Acknowledgements II

Abstract III

List of Tables VI

List of Figures VII

1. Introduction **1**

1.1. Background 1

1.2. Problem Statement and Motivation 1

1.3. Aim and Objectives 3

1.3.1. Aim 3

1.3.2. Objectives 3

1.4. Scope, Limitations, and Delimitations 3

1.5. Justification 3

1.6. Hypothesis 3

1.7. Contribution of the Thesis 4

1.8. Organization of the Thesis 4

2. Literature Review **5**

2.1. — 5

2.1.1. — 5

3. Greedy Random Heuristic Selection (GRHS) **6**

3.1. – 6

3.1.1. Overview 6

4. Stratified Sampling (SS) **7**

5. Experiments **8**

6. Conclusion **9**

6.1. Summary of Contributions 9

References **10**

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Background

Exists many problems of Artificial Intelligent (`AI`), such as: Finding the shortest path from one point to another in a game map, solve the games of PACMAN, 8−tile−puzzle, Rubick's cube, etc. The level of difficulty to solve the problems mentioned are linked with the size of the search space generated.

State−space search algorithms have been used to solve the problems mentioned above. And in this dissertation we study the approach to solve problems in order to reduce the size of the search tree generated and the running time of the search algorithm.

1.2 Problem Statement and Motivation

Every problem of Artificial Intelligent can be cast as a state space problem. The state space is a set of states where each state represent a possible solution to the problem and each state is linked with other states if exists a function that goes from one state to another. In the search space there are many solutions that represent the same state, each of this solutions are called node. So, many nodes can be represented as one state. To find the solution of the problem is required the use of search algorithms such as: Depth First Search

(DFS), which looks the solution of the problem traversing the search space exploring the nodes in each branch before backtracking up to find the solution. Another search algorithm is Breadth First Search (BFS), which looks for the solution exploring the neighbor nodes first, before moving to the next level of neighbors. The mentioned algorithms have the characteristic that when they do the search, they generate a larger search space, basically for two main reasons: a) Consider the total number of states to be analyzed in order to determinate if the solution is found. b) There is no guide to get to the solution. The search space that these algorithms generate are called Brute force search tree (BFST).

There are other types of algorithms called heuristic informed search, which are algorithms that requires the use of heuristics. The heuristic is the estimation of the distance for one node in the search tree to get to the near solution. The heuristic informed search generates a smaller search tree in comparison to the BFST, because the heuristic guides the search exploring the nodes that are in the solution path and prunes the nodes which are not. Also, the use of heuristics reduce the running time of the search algorithm.

There are different approaches to create heuristics, such as: Pattern Databases (PDBs), Neural Network, and Genetic Algorithm. These systems that create heuristics receive the name of Heuristics Generators. And one of the approaches that have showed most successfull results in heuristic generation is the PDBs, which is memory-based heuristic functions obtained by abstracting away certain problem variables, so that the remaining problem ("pattern") is small enough to be solved optimally for every state by blind exhaustive search. The results stored in a table, represent a PDB for the original problem. The abstraction of the search space gives an admissible heuristic function, mapping states to lower bounds.

Exists many ways to take advantage of all the heuristics that can be created, for example: [Holte et al., 2006] showed that search can be faster if several smaller pattern databases are used instead of one large pattern database. In addition [Domshlak et al., 2010] and [Tolpin et al., 2013] results showed that evaluating the heuristic lazily, only when they are essensial to a decision to be made in the search process is worthy in comparison to take the maximum of the set of heuristics. Then, using all the heuristics do not guarantees to solve the major number of problems in a limit time.

1.3 Aim and Objectives

1.3.1 Aim

The objective of this dissertation is to develop meta-reasoning approaches for selecting heuristics functions from a large set of heuristics with the goal of reducing the running time of the search algorithm employing these functions.

1.3.2 Objectives

- Develop an approach for selecting a subset of heuristic functions with the goal of reducing the running time of the search algorithms employing these functions.
- Develop approaches to obtain the cardinality of the subsets of heuristics found.
- Develop a method to find a subset of heuristics from a large pool of heuristics that optimize the number of nodes expanded in the process of search.
- Use Stratified Sampling (SS) algorithm for predicting the search tree size of Iterative-Deepening A* (IDA*). We use SS as our utility function.

1.4 Scope, Limitations, and Delimitations

TODO

1.5 Justification

TODO

1.6 Hypothesis

This thesis will intend to prove the hypotheses listed below:

- **H1:** The verification that our objective function of selection is related with two properties: Monotonicity and Submodularity .
- **H2:** Reducing the size of the search tree generated helps to solve more problems.

1.7 Contribution of the Thesis

The main contributions of this Thesis are:

- Provide a prediction method to estimate the size of the search tree generated.
- Provide a meta−reasoning approach based on the size of the search tree generated.
- Provide a meta−reasoning approach based on the evaluation cost of each heuristic.

1.8 Organization of the Thesis

The Thesis is organized as follows:

1. In Chapter 1, the introduction to the thesis is provided which also includes our motivation and defines its scope.
2. In Chapter 2, we review the State of the Art.
3. In Chapter 3, we introduce our meta−reasoning approach.
4. In Chapter 4, we introduce.
5. In Chapter 5, we .
6. We conclude in Chapter 6 by discussing further improvements and future work.

In the next chapter, the domain 8−tile−puzzle is used to understand the concepts that will be helpful for the other chapters.

CHAPTER 2

Literature Review

2.1 —

2.1.1 —

In the next chapter, ...

CHAPTER 3

Greedy Random Heuristic Selection (GRHS)

The purpose of this section is to introduce the meta−reasoning proposed for specifying ...

3.1 –

3.1.1 Overview

In the next chapter, ...

CHAPTER 4

Stratified Sampling (SS)

In this chapter, we present the Stratified Sampling.

CHAPTER 5

Experiments

In this chapter, we verify the the use-cases studied in the previous chapter using our meta.−reasoning approach introduced in Chapter 3. The main purpose of this chapter is to prove the hypotheses introduced in chapter 1:

CHAPTER 6

Conclusion

6.1 Summary of Contributions

In this thesis,

References

Robert C Holte, Ariel Felner, Jack Newton, Ram Meshulam, and David Furcy. Maximizing over multiple pattern databases speeds up heuristic search. *Artificial Intelligence*, 170 (16):1123–1136, 2006.

Carmel Domshlak, Erez Karpas, and Shaul Markovitch. To max or not to max: Online learning for speeding up optimal planning. In *AAAI*, 2010.

David Tolpin, Tal Beja, Solomon Eyal Shimony, Ariel Felner, and Erez Karpas. Towards rational deployment of multiple heuristics in a*. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 674–680. AAAI Press, 2013.