CENTRO DE CIENCIAS EXATAS E TECNOLOGICAS - CCE

DEPARTAMENTO DE INFORMATICA

PROJECT:

ON SELECTING HEURISTIC FUNCTIONS FOR DOMAIN-INDEPENDENT PLANNING.

Marvin Abisrror

Supervised by:
Levi Henrique Santana de Lelis, Santiago Franco

Contents

0.1 Introduction

Solving independent domains is a very hard task when we do planning. Sometimes we will find instances that takes too much time to find the solution, and the measure of that time could be hours, days or even years. Users of search algorithms usually do not know *a priori* how long it will take to solve a problem instance – some instances could be solved quickly while others could take long time. Many researchs in the area of heuristic search explains why this could be happening, some of those reasons are: That the size of the tree is very large, the inconsistency of the heuristic used to do the search or the complexity of the instance that could contains cycles or even dead end areas which avoid to make any movement in the search space tree, etc.

Furthermore, domain independent planning is related to the general problems solving where we basically require a very inteligent domain-independent search algorithm or heuristic to do the search in order to solve the instance without knowing in detail how is the problem and the search space it produces.

Also, the need to solve the major number of problems using restricted resources of memory and time is vital in the area of Artificial Intelligence (AI), because each time are released problems with high grade of difficulty that can not be solved using the last state of the art in planning. In addition, the seek for solving domain independent planning is critical is the areas manufacturing, space systems, software engineers, robotics, education, and entertainment.

On the whole, finding an strategy which allow us to solve the major number of problems using domain indepenent planning is a very important task in AI.

Finally, the proposal of this project is to develop an approach of selecting heuristics functions to optimize the search of the solution applied by the domain independent planning. In order to reduze the amount of memory and time.

0.2 Problem and Importance

This project is concerned with cost-optimal state-space planning using the A* algorithm [Hart and Raphael, 1968]. We asume that a pool, ζ, of hundreds or even thousands of heuristics is available, and that the final heuristic used to guide A*, h_{max}, will be defined as the maximum over a subset ζ' of those heuristics ($h_{max(s,\zeta')} = \max_{h \varepsilon \zeta'} h(s)$). The choice of the subset ζ' can greatly affect the efficiency of A*. For a given size N and planning task $\bigtriangledown$, a subset containing N heuristics from ζ is optimal if no other subset containing N heuristics from ζ results in A* expanding fewer nodes when solving $\bigtriangledown$.

0.3 Hypotheses

We present a greedy method based on the theory of supermodular optimization for selecting a subset of heuristic functions, without relying on domain knowledge, to guide the A* search algorithm. If the heuristics are consistent, our method selects a subset which is guaranteed to be near optimal to the resulting A* search tree size. Furthermore to being consistent, if all heuristics have the same evaluation time, we expect the subset would be near-optimal with respect to the resulting A* search tree size.

0.4 Objectives

The first contribution of this research is to show that the problem of finding the optimal subset of ζ of size N for a given problem task is supermodular, and therefore a greedy algorithm, which we call Greedy Heuristic Selection (GHS), that selects heuristics from ζ one at a time is guaranteed to produce a subset ζ' such that the number of nodes expanded by A* is no more than approximately 1.36 times optimal. Moreover, if all heuristics in ζ have the same evaluation time, then the A* running time using ζ' is also guaranteed to be no more than approximately 1.36 times optimal. An optimization procedure which is similar to ours is presented by [Rayner et al., 2013], but their procedure maximizes the average heuristic value. By contrast, GHS minimizes the search tree size.

GHS requires a prediction of the number of nodes expanded by A* using any given subset. Although there are methods for accurately predicting the number of nodes expanded by Iterative Deepening-A* [Korf et al., 2001] (IDA*), Stratified Sampling [Lelis et al., 2013] (SS), these methods can't be easily adapted to A* because A*'s duplicate pruning makes it very difficult to predict how many nodes will occur at depth d of A*'s search tree (the tree of nodes expanded by A*). In this research we want to make another contribution showing that the Stratification Sampling(SS) generates very good predictions for the A*'s search tree.

The second contribution of this research is an empirical evaluation of GHS in optimal domain-independent planning problems. Namely, we implemented and evaluated GHS on the planning problems from the 2011 International Planning Competition (IPC).

The main aim of this research is to develop an approach to resolves problems quickly. The approach we are proposing can be applied for solving optimal domain problems in the Fast-Downward Planner [Helmert, 2006]. A few of these domains are: Elevators opt08 and opt11, Floortile opt11, Nomystery opt11, etc.

0.5 Literature

State-space search algorithms, such as A* [Hart and Raphael, 1968], are fundamental in many AI applications. A* uses the $f(s) = g(s) + h(s)$ cost function to guide its search. Here, $g(s)$ is the cost of the path from the start state to state s, and $h(s)$ is the estimated cost-to-go from s to a goal; $h(.)$ is known as the heuristic function. A heuristic h is *admissible* if $h(s) \leq dist(s, goal)$ for every state **s** and goal state *goal*, where *dist(n, m)* is the cost of at least-cost path from n to m. If $h(n)$ is admissible, *i,e.* always returns a lower bound estimate of the optimal cost, these algorithms are guaranteed to find an optimal path from the start state to a goal state if one exists.

In addition to being admissible the heuristic also could be *consistent*. A heuristic h is *consistent* if for every pair of states, m and n, $h(m) - h(n) \leq dist(m, n)$

0.6 Metodology

sdfsdfsdf

0.7 Timetable

ssfdsf

0.8 Problem Formulation

When solving $\triangledown$ using the consistent function $h_{max}(\zeta')$, for $\zeta' \subseteq \zeta$, A* expands in the worst case $J(\zeta', \triangledown)$ nodes, where:

$$J(\zeta', \triangledown) = |\{s \varepsilon V | f_{max}(s, \zeta') \leq C^{\star}\}| \tag{1}$$

$$J(\zeta', \triangledown) = |\{s \varepsilon V | h_{max}(s, \zeta') \leq C^{\star} - g(s)\}| \tag{2}$$

We write $J(\zeta')$ instead of $J(\zeta', \triangledown)$ whenever $\triangledown$ is clear from context.

We present a greedy algorithm for approximately solving the following optimization problem.

$$\min_{\zeta' \varepsilon 2^{|\zeta|}} J(\zeta', \triangledown) \tag{3}$$

$$subject\ to\ \ |\zeta'| = N \tag{4}$$

where N could be determined by a hard constraint such as the maximum number of PDBs one can store in memory.

Bibliography

N. J.; Hart, P. E.; Nilsson and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC*, 4(2):100–107, 1968.

Chris Rayner, Nathan Sturtevant, and Michael Bowling. Subset selection of search heuristics. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 637–643, 2013.

Richard E Korf, Michael Reid, and Stefan Edelkamp. Time complexity of iterative-deepening-a*. *Artificial Intelligence*, 129(1):199–218, 2001.

Levi HS Lelis, Sandra Zilles, and Robert C Holte. Predicting the size of ida* search tree. *Artificial Intelligence*, 196:53–76, 2013.

Malte Helmert. The fast downward planning system. *J. Artif. Intell. Res.(JAIR)*, 26:191–246, 2006.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.