Artificial Intelligence 230 (2016) 51–73

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Predicting optimal solution costs with bidirectional stratified sampling in regular search spaces

Levi H.S. Lelis [a,*], Roni Stern [b], Shahab Jabbari Arfaee [c], Sandra Zilles [d], Ariel Felner [b], Robert C. Holte [c]

[a] *Departamento de Informática, Universidade Federal de Viçosa, Viçosa, MG, Brazil*
[b] *Information Systems Engineering, Ben Gurion University, Beer-Sheva, Israel*
[c] *Computing Science Department, University of Alberta, Edmonton, AB, Canada*
[d] *Department of Computer Science, University of Regina, Regina, SK, Canada*

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form 7 September 2015
Accepted 29 September 2015
Available online 3 October 2015

Keywords:
Heuristic search
Solution cost prediction
Stratified sampling
Type systems
Learning heuristic functions

ABSTRACT

Optimal planning and heuristic search systems solve state-space search problems by finding a least-cost path from start to goal. As a byproduct of having an optimal path they also determine the optimal solution cost. In this paper we focus on the problem of determining the optimal solution cost for a state-space search problem directly, i.e., without actually finding a solution path of that cost. We present an algorithm, BiSS, which is a hybrid of bidirectional search and stratified sampling that produces accurate estimates of the optimal solution cost. BiSS is guaranteed to return the optimal solution cost in the limit as the sample size goes to infinity. We show empirically that BiSS produces accurate predictions in several domains. In addition, we show that BiSS scales to state spaces much larger than can be solved optimally. In particular, we estimate the average solution cost for the 6×6, 7×7, and 8×8 Sliding-Tile puzzle and provide indirect evidence that these estimates are accurate. As a practical application of BiSS, we show how to use its predictions to reduce the time required by another system to learn strong heuristic functions from days to minutes in the domains tested.

1. Introduction

Many real-world problems can be cast as state-space search problems. For instance, state-space search algorithms have been used in a number of applications: robotics [37], domain-independent planning [2], chemical compounds discovery [14], bin packing [23], sequence alignment [22], automating layouts of sewers [4], and network routing [36], among others.

Heuristic search algorithms such as A* [13] and Iterative-Deepening-A* (IDA*) [24] are guided by the cost function $f(s) = g(s) + h(s)$ while finding solutions for state-space problems. Here, $g(s)$ is the cost of reaching state s from the root of the underlying search tree representing the state-space problem and $h(s)$ is a heuristic function providing an estimate of the cost-to-go of a solution going through state s in the underlying search tree.

Such algorithms are designed to find a least-cost path from a start state to a goal state in state-space search problems. The solution cost of such a path is found as a byproduct. In this paper we are interested in applications for which one only needs to know the optimal solution cost or an accurate estimate of the optimal solution cost – the solution path is not

* Corresponding author.
E-mail address: levilelis@gmail.com (L.H.S. Lelis).

http://dx.doi.org/10.1016/j.artint.2015.09.012
0004-3702/

needed. For example, consider the owner of a construction company that is required to quickly assess the monetary cost of a project for bidding purposes. In this case, only the cost of executing the project is needed. The actual construction plan could be formulated later, if the bid is won. Thus, an important question to be answered is the following. Can one accurately and quickly predict the optimal solution cost of a problem without finding an optimal sequence of actions from the start to a goal?

1.1. Heuristic functions

The heuristic function $h(\cdot)$ used by heuristic search algorithms is in fact an estimate of the optimal solution cost. This estimate is called *admissible* if it never overestimates the cost of the lowest-cost path from state s to the goal. Heuristic search algorithms, such as A* and IDA* guided by the f function are guaranteed to find an optimal solution when h is admissible [13,24]. A considerable amount of effort has been devoted to creating admissible heuristics [8,16,48,51] and inadmissible heuristics [11,20,44,49]. Admissible heuristics frequently provide inaccurate predictions of the optimal solution cost as they are biased to never overestimate the actual cost [12]. In some cases, even inadmissible heuristics are biased towards admissibility [11,44].

Regardless of admissibility, heuristics share a property: the heuristic evaluation must be fast enough to be computed for every node generated during search (in some settings it is more efficient to perform lazy heuristic computation during node expansion [42,50]), while a solution cost predictor is run only on the start state. In fact, often, heuristic functions sacrifice accuracy for speed. By contrast, solution cost predictors aim at accurately predicting the optimal solution cost of a problem instance. While algorithms for predicting the optimal solution cost can be viewed as a heuristic, they differ from a heuristic conceptually in that: 1) they are not required to be fast enough to guide search algorithms; 2) they do not favor admissibility; 3) they aim at making accurate predictions and thus our measure of effectiveness is prediction accuracy, in contrast to the solution quality and search time used to measure the effectiveness of heuristic functions.

1.2. Contributions

In this paper we present an algorithm for quickly and accurately predicting the optimal solution cost of state-space search problems. Our solution cost predictor, Bidirectional Stratified Sampling (`BiSS`), overcomes the two drawbacks of the Solution Cost Predictor (`SCP`) [33], another algorithm we introduced for predicting the optimal solution cost. Namely, in contrast to `SCP`, `BiSS` scales to very large state spaces and it has the guarantee of eventually converging to the correct answer. We describe `SCP` in Section 2.4 below.

A preliminary version of this paper appeared in the Proceedings of the International Conference on Automated Planning and Scheduling (2012) [28] and as a short paper in the Proceedings of the Symposium on Combinatorial Search [30]. The current paper substantially extends its preliminary versions. In addition to a comprehensive explanation of the algorithm, we include new experimental results. To be specific, in this paper we make the following contributions.

- We introduce `BiSS`, a prediction algorithm that has two advantages over `SCP`: (1) it entirely avoids the time-consuming preprocessing required by `SCP`; and (2) unlike `SCP`, `BiSS` is guaranteed to return the optimal solution cost in the limit as its sample size goes to infinity.
- We show empirically that `BiSS` scales to state spaces much larger than can be solved optimally. In particular, we predict the average solution cost for the Sliding-Tile puzzles up to the 8×8 configuration, which has more than 10^{88} reachable states, and provide indirect evidence that `BiSS`'s predictions for these huge state spaces are quite accurate.
- As an application of `BiSS` we show how to quickly learn strong heuristics from predictions. We show that it is possible to reduce the time required for learning strong heuristic functions from days to minutes by using `BiSS`'s predictions to label the training set.

Although `BiSS` overcomes two of the main limitations of `SCP`, it has two disadvantages `SCP` does not have. First, `BiSS` only produces predictions for domains with single goal states. Second, `BiSS` is applicable only to domains for which it is possible to "reason" backwards from the goal state. We discuss `BiSS`'s weaknesses in Section 6.

This paper is organized as follows. In the next section we review Chen's Stratified Sampling (`SS`) [7], an algorithm for efficiently estimating the size of search trees. In Section 3 we describe `BiSS`, which is a bidirectional variation of `SS` for predicting the optimal solution cost, and show that `BiSS` is guaranteed to produce perfect predictions as the sample size tends to infinity. In Section 4 we show empirically that `BiSS` produces accurate predictions of the optimal solution cost, scaling to state spaces much larger than can be solved optimally. In Section 5 we show how to use `BiSS` to generate a training set to learn strong heuristic functions. In Section 6 we list `BiSS`'s limitations and finally, in Section 7, we conclude the paper.

1.3. Problem formulation

Given a directed and implicitly defined search tree representing a state-space search problem rooted at start state s^* [38], called the underlying search tree (*UST*), we are interested in estimating the optimal solution cost of a path from s^* to the