

Distance Learning in Agent-Centered Heuristic Search

Nathan R. Sturtevant

Computer Science Department
University of Denver
Denver, Colorado, USA
sturtevant@cs.du.edu

Abstract

Real-time agent-centric algorithms have been used for learning and solving problems since the introduction of the LRTA* algorithm in 1990. In this time period, numerous variants have been produced, however, they have generally followed the same approach in varying parameters to learn a heuristic which estimates the remaining cost to arrive at a goal state. This short paper discusses the history and implications of learning g -costs, both alone and in conjunction with learning h -costs as an introduction to the new f -LRTA* algorithm which learns both.

Introduction and Background

Agent-centered real-time heuristic search algorithms (Koenig 2001) model an agent trying to find a route through an initially unknown environment. With limited sensing the agent can only plan and reason in its local vicinity. With limited computational power, the agent is forced to record the result of any learning in the environment. In order to guarantee that an agent will not get permanently stuck in a local minima, learning rules are used to update knowledge about the environment. By far the most common type of learning is heuristic (h -cost) learning, or learning an estimate of the cost to reach the goal state. In a repeated learning paradigm or with multiple agents, previous learning can be re-used for better performance or until an optimal solution is found.

LRTA* (Korf 1990), is often considered to be the first agent-centered search algorithm. However, the current formulation of the problem is somewhat removed from the intent of this first work, which was to find suboptimal solutions to large problems which could not be solved directly. A wide variety of other researchers have worked on this and similar problems over many years (eg (Russell and Wefald 1991; Shimbo and Ishida 2003; Koenig and Sun 2009)), although it is beyond the scope of this paper to discuss their approaches in detail.

A reliance on similar heuristic learning approaches and a lack of standardized test suites has resulted in many algorithms which offer only incremental improvements over existing approaches. While some work has gone into standard-

izing the various techniques used, such as in LRTS (Bulitko and Lee 2006), more could be done in this regard.

A notable feature of the learning by LRTA* and similar algorithms is a ‘scrubbing’ behavior where agents repeatedly cover the same terrain while slowly performing heuristic learning. This behavior can be partially explained as a result of the learning rule being used. In a particular type of local minima on grid-maps, it has been shown (Sturtevant *et al.* 2010) that a local minima of N states requires $O(N^{1.5})$ learning steps to escape. While constant factors can be tuned to reduce this behavior, fundamentally the size of a local minima can scale, while it is assumed that the reasoning capabilities of an agent are fixed, regardless of the size of the environment. The analysis of more domains and local minima is an open research question. An algorithm would ideally require at most $O(N)$ learning to escape a local minima of N states, which would mean asymptotic performance similar to A* (Hart *et al.* 1968). A* is not agent-centric or real-time, but with a consistent heuristic, it expands each state only once.

As an aside note that, while the need for planning speed is correctly attributed to video games, heuristic learning and the need to act in constant time independent of the problem size is not. Memory allocation, and thus extensive online learning, is expensive and avoided when possible. In most games the maximum map size is also determined *a priori* and, if an editor is not provided with the game, planning only has to work on the maps that ship with the game. As a general rule, work intended for use in games should be compared against approaches already used in games, such as (Sturtevant 2007), in important metrics such as memory usage, planning speed, and implementation complexity.

Alternate Learning: g -costs

An approach which, in our view, has not received adequate attention is the learning of distances from the start state, or g -costs. One might ask why an agent would care where it came from, because all that matters is reaching the goal. The question is answered in more detail in (Sturtevant and Bulitko 2011), but from a high-level perspective g -cost learning provides more potential for an agent to escape a local minima of N states in $O(N)$ time.

The learning of g -costs is not new. In agent-centered search this learning was suggested in the FALCOLNS algo-

Algorithm	First Trial			All Trials			Trials
	Dist.	States Exp.	Time	Dist.	States Exp.	Time	
RIBS	2,861,553	1,219,023	4.46	2,861,553	1,219,023	4.46	1
LSS-LRTA*(1)	633,377	2,761,163	12.58	23,660,293	98,619,076	266.81	12,376
LSS-LRTA*(10)	113,604	952,402	2.18	3,844,178	21,970,613	53.30	1951
LSS-LRTA*(100)	18,976	305,518	0.63	559,853	5,920,709	14.74	295
f -LRTA _t *(1,10.0)	127,418	389,367	2.44	1,848,413	5,321,043	32.14	890
f -LRTA _t *(10,10.0)	52,097	188,144	0.89	663,378	1,808,493	9.18	264
f -LRTA _t *(100,10.0)	10,890	82,414	0.35	136,184	624,262	2.96	69

Table 1: Average results on Dragon Age: Origins maps with octile movement.

rithm (Furcy and Koenig 2000), which uses g -costs for tie-breaking. The RIBS algorithm (Sturtevant *et al.* 2010) suggested just learning g -costs, and showed that g -costs could be used for identifying local minima within a map. Work on merging RIBS and LRTA* resulted in f -LRTA* which learns both g - and h -costs, and uses g -costs for pruning local minima from the state space. f -LRTA* will converge to the optimal solution

Along any optimal path to a goal, g -costs must increase. Thus, if a state does not have successors with higher g -cost, that state cannot be on an optimal path to the goal. Note that in the simplest implementation, any pruning based on this insight would require that the g -cost of any given state be correct, but f -LRTA* prunes even when only estimates of the g -cost of a state are available. This pruning can be effective because it eliminates states that could otherwise be attractive: those with low heuristic values suggesting that they are close to the goal.

The same property holds in reverse for heuristics. Along an optimal path, heuristics will be decreasing, so if a state does not have predecessors with higher heuristic cost it cannot be on an optimal path between the start and the goal. But, this is a less useful observation for several reasons. Most importantly, the rule would prune states that are farther from the goal than the agent, which an agent would already naturally avoid.

We provide evidence of the effectiveness of combined g -cost and h -cost learning of f -LRTA* in Table 1. These results are taken from (Sturtevant and Bultiko 2011) and are the result of experiments on grid-based maps from the game Dragon Age: Origins. The first set of columns report average metrics required for reaching the goal state on the first exploration of the environment, followed by the metrics required for convergence to an optimal solution when repeatedly solving the same problem. LSS-LRTA* (Koenig and Sun 2009) is used as a sample algorithm for the purpose of comparison. f -LRTA* has much better performance on all metrics.

Even if only the first trial is used in some contexts, we assert that both first trial and convergence results are interesting and should be compared when possible, because they give insight into how an algorithm alters the nature of a problem.

Summary and Conclusions

The purpose of this paper is to introduce and motivate the use of g -cost learning in real-time agent-centered search.

Results have been provided showing the effectiveness of f -LRTA*, which learns both g - and h -costs.

While these results are strong, there are many open questions for future research. f -LRTA* has a number of parameters which could be dynamically modified to adapt for the environment and potentially improve performance. Even more importantly, we have devised new pruning schemes which have the potential to further reduce the planning costs of f -LRTA*.

In order to make it easier for others to compare against f -LRTA*, the implementation is available as open source¹. Additionally, we are in the process of building benchmark sets for grid-based worlds² with the hope that the standardized sets will provide improved performance comparisons for algorithms in the future.

References

- Vadim Bulitko and Greg Lee. Learning in real time search: A unifying framework. *JAIR*, 25:119–157, 2006.
- David Furcy and Sven Koenig. Speeding up the convergence of real-time search. In *AAAI*, pages 891–897, 2000.
- P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- Sven Koenig and Xiaoxun Sun. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18(3):313–341, 2009.
- Sven Koenig. Agent-centered search. *AI Mag.*, 22(4):109–132, 2001.
- Richard Korf. Real-time heuristic search. *AIJ*, 42(2-3):189–211, 1990.
- S. Russell and E. Wefald. *Do the right thing: Studies in limited rationality*. MIT Press, 1991.
- Masaki Shimbo and Toru Ishida. Controlling the learning process of real-time heuristic search. *AIJ*, 146(1):1–41, 2003.
- N. Sturtevant and V. Bultiko. Learning where you are going and from whence you came: h - and g -cost learning in real-time heuristic search. In *International Joint Conference on Artificial Intelligence (IJCAI-11)*, 2011.
- Nathan R. Sturtevant, Vadim Bulitko, and Yngvi Börnsson. On learning in agent-centered search. In *AAMAS*, pages 333 – 340, 2010.
- Nathan R. Sturtevant. Memory-efficient abstractions for pathfinding. In *AIIDE*, pages 31–36, 2007.

¹<http://code.google.com/p/hog2/>

²<http://movingai.com/benchmarks/>