

Real-Time Adaptive A* with Depression Avoidance

Carlos Hernández

Departamento de Ingeniería Informática
Universidad Católica de la Santísima Concepción
Concepción, Chile

Jorge A. Baier

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Santiago, Chile

Abstract

Real-time search is a well known approach to solving search problems under tight time constraints. Recently, it has been shown that LSS-LRTA*, a well-known real-time search algorithm, can be improved when search is actively guided away of depressions. In this paper we investigate whether or not RTAA* can be improved in the same manner. We propose aRTAA* and daRTAA*, two algorithms based on RTAA* that avoid heuristic depressions. Both algorithms outperform RTAA* on standard path-finding tasks, obtaining better-quality solutions when the same time deadline is imposed on the duration of the planning episode. We prove, in addition, that both algorithms have good theoretical properties.

Introduction

Several real-world applications require agents acting in complex environments to repeatedly act quickly in a possibly unknown environment. That is the case of virtual agents that repeatedly perform pathfinding tasks in commercial games (e.g., *World of Warcraft*, *Baldur's Gate*, etc.).

Real-time heuristic search (e.g. Korf 1990; Koenig 2001) is a standard approach used to solve these tasks. It is inspired on A* search (Hart, Nilsson, and Raphael 1968), but a mechanism is provided to bound the time taken to produce a movement. Real-time heuristic search algorithms repeatedly perform planning episodes whose computational time is bounded. Like standard A*, real-time heuristic algorithms use a heuristic function to guide action selection. Unlike A* search, the heuristic function is updated by the algorithm during execution. Such a process is usually referred to as the *learning* of the heuristic. The learning process guarantees that algorithms like LRTA* (Korf 1990) always finds a solution if any exists in finite, undirected search spaces.

It has been observed that algorithms like LRTA* perform poorly when they enter *heuristic depressions* (e.g. Ishida 1992). Intuitively, a depression is a bounded connected component of the search space, D , in which the heuristic underestimates too much the cost to reach a solution in relation to the heuristic values of the states in the border of D .

Recently we proposed aLSS-LRTA*, a variant of LSS-LRTA* (Koenig and Sun 2009) that *actively* guides search

away of heuristic depressions; a principle called *depression avoidance* (Hernández and Baier 2011).

In this paper we report on whether or not depression avoidance can be incorporated into RTAA* (Koenig and Likhachev 2006), a real-time heuristic search algorithm that implements a faster but less informed learning mechanism than LSS-LRTA*. We propose two new real-time heuristic search algorithms. The first, aRTAA*, is a rather straightforward adaptation of aLSS-LRTA*'s implementation of depression avoidance into RTAA*. As aLSS-LRTA* does, it avoids moving to states that have been identified as belonging to a depression. We show aRTAA* outperforms RTAA* in path-finding tasks. The second algorithm, daRTAA*, is a finer implementation of depression avoidance that, like aRTAA*, will prefer to move to states that do not belong to a depression, but, unlike aRTAA*, when no such states are found, it prefers moving to states whose heuristic has changed the least. We show daRTAA* outperforms aRTAA*. Both algorithms have nice theoretical properties: they maintain the consistency of the heuristic, and they terminate finding a solution when such a solution exists.

Preliminaries

The objective of a real-time search algorithm is to make an agent travel from an initial state to a goal state performing, between moves, an amount of computation bounded by a constant. An example situation is pathfinding in previously unknown grid-like environments. There the agent has memory capable of storing its current belief about the structure of the search space, which it initially regards as obstacle-free (this is usually referred to as the *free-space assumption* (Koenig, Tovey, and Smirnov 2003)). The agent is capable of a limited form of sensing: only obstacles in the neighbor states can be detected. When obstacles are detected, the agent updates its map accordingly.

Most state-of-the-art real-time heuristic search algorithms, including RTAA*, can be described by the pseudocode in Algorithm 1. The algorithm iteratively executes a lookahead-update-act cycle until the goal is reached. Variables $s_{current}$, h , and c stand, respectively, for the current state, the heuristic function, and the cost associated to the arcs of the search space. The h function is given as a parameter and its initial value is stored in the variable h_0 .

RTAA* carries out an A* search that expands at most k

Algorithm 1: A real-time heuristic search algorithm

Input: A search space S , a set of goal states G , a heuristic function h , a cost function c .

```
1 for each  $s \in S$  do
2    $h_0(s) \leftarrow h(s)$ 
3  $s_{current} \leftarrow s_0$ 
4 while  $s_{current} \notin G$  do
5   LookAhead()
6   if  $Open = \emptyset$  then return no-solution
7    $s_{next} \leftarrow \text{Extract-Best-State}()$ 
8   Update()
9   move the agent from  $s_{current}$  to  $s_{next}$  through the path
   identified by LookAhead. Stop if an action cost along the path
   is updated.
10   $s_{current} \leftarrow$  current agent position
11  update action costs (if they have increased)
```

states in the lookahead phase (Line 5). The variable $Open$ (cf. Line 6) contains A*’s search frontier. Also, we assume that after executing an A* lookahead, the variable $Closed$ contains the states that were expanded by the algorithm. Finally, the next state to move to, s_{next} , is assigned in Line 7. In RTAA*, the `Extract-Best-State` procedure returns the state with lowest f -value in $Open$. Finally, the update procedure updates the h -value of the states in $Closed$ using the f -value of the s_{next} state (for more details, refer to the original paper).

RTAA* with Depression Avoidance

We implemented depression avoidance into RTAA* in two different ways, producing two new algorithms.

aRTAA* aRTAA* is a straightforward port of aLSS-LRTA*’s implementation of depression avoidance into RTAA*. RTAA* is modified as follows. First, its update procedure implements the same update rule of RTAA* but, like aLSS-LRTA*, it marks states that have been updated. Second, RTAA*’s `Extract-Best-State` procedure returns the state with lowest f -value that is not marked, if such a state exists; otherwise, it returns the state with lowest f -value, just as RTAA* would do. As a result aRTAA* is a version of RTAA* that avoids depressions using the same mechanism that aLSS-LRTA* utilizes.

daRTAA* daRTAA* is based on aRTAA*, but differs from it in the strategy used to select the next state to move to. daRTAA* will attempt to escape the depression by choosing the state with best f -value among the states whose heuristic has *changed the least*. More formally, let \mathcal{L} contain each state s in $Open$ such that there is no other s' in $Open$ such that $h(s') - h_0(s') < h(s) - h_0(s)$. daRTAA*’s `Extract-Best-State` procedure returns the state with least f -value from \mathcal{L} . On the other hand, daRTAA*’s update procedure is the same used by RTAA*.

Properties We have proven that if h is initially consistent, then it remains consistent throughout an execution of any of daRTAA* or aRTAA*. In addition, in finite undirected state spaces, both algorithms will find a solution if it exists. Finally, after running a finite number of trials, both daRTAA* and aRTAA* will converge to an optimal solution.

k	Solution Cost	# Planning Episodes	Time per Episode	Time	Percolations per episode
RTAA*					
1	553,152	510,579	0.0004	183.4	5.7
7	197,781	107,321	0.0016	174.7	39.9
25	56,989	16,606	0.0055	90.9	198.3
97	15,422	2,632	0.0214	56.4	992.4
aRTAA*					
1	432,806	399,603	0.0005	197.0	9.0
7	146,724	80,528	0.0021	170.0	61.3
25	45,336	13,660	0.0065	88.2	244.7
97	13,099	2,362	0.0231	54.6	1,071.1
daRTAA*					
1	50,906	47,836	0.0005	19.5	9.1
7	28,932	22,564	0.0023	51.3	90.8
25	14,116	8,393	0.0082	69.1	351.9
97	6,397	2,279	0.0263	59.8	1,265.0

Figure 1: The table presents average solution cost, number of planning episodes, time per planning episode in milliseconds, total search time in milliseconds, and number of heap percolations per planning episode for 6,000 path-planning tasks and 4 lookahead values (k). We performed our experiments on a Linux PC with a Pentium QuadCore 2.33 GHz CPU and 8 GB RAM.

Experimental Evaluation

We compared RTAA*, aRTAA* and daRTAA* at solving real-time navigation problems in unknown environments. We used 12 video games maps for the experiments. The first six are taken from the game *Dragon Age*; the remaining six are taken from the game *StarCraft*. We average our experimental results over 6,000 test cases (500 test cases for each game map). A summary of the results is shown in Figure 1. We observe that in terms of solution cost, for all lookahead values, aRTAA* consistently outperforms RTAA*; moreover, daRTAA* consistently outperforms aRTAA*. In addition, daRTAA*’s more refined mechanism for escaping depressions is better than that of aRTAA*. For small values for the lookahead parameter, daRTAA* obtains better solutions than aRTAA* used with a much larger lookahead. daRTAA* needs only a lookahead parameter of 25 to obtain solutions better than RTAA* with lookahead parameter of 97. With those values, daRTAA* requires about 2.6 times less time per planning episode than RTAA*.

References

- Hart, P. E.; Nilsson, N.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimal cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2).
- Hernández, C., and Baier, J. A. 2011. Real-time heuristic search with depression avoidance. In *IJCAI*.
- Ishida, T. 1992. Moving target search with intelligence. In *AAAI*, 525–532.
- Koenig, S., and Likhachev, M. 2006. Real-time adaptive A*. In *AAMAS*, 281–288.
- Koenig, S., and Sun, X. 2009. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems* 18(3):313–341.
- Koenig, S.; Tovey, C. A.; and Smirnov, Y. V. 2003. Performance bounds for planning in unknown terrain. *AIJ* 147(1-2):253–279.
- Koenig, S. 2001. Agent-centered search. *AI Mag.* 22(4):109–131.
- Korf, R. E. 1990. Real-time heuristic search. *AIJ* 42(2-3):189–211.