Paper Summary: Time-Bounded Adaptive A*

Carlos Hernández*
Depto. de Ingeniería Informática
Universidad Católica de la Ssma. Concepción
Caupolican 491, Concepción, Chile
chernan@ucsc.cl

Jorge Baier
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Santiago, Chile
jabaier@ing.puc.cl

Tansel Uras and Sven Koenig
Computer Science Department
University of Southern California
Los Angeles, CA 90089, USA
{turas,skoenig}@usc.edu

Abstract

This paper summarizes our AAMAS 2012 paper on "Time-Bounded Adaptive A*," which introduces the game time model to evaluate search algorithms in real-time settings, such as video games. It then extends the existing real-time search algorithm TBA* to path planning with the freespace assumption in initially partially or completely unknown terrain, resulting in Time-Bounded Adaptive A* (TBAA*). TBAA* needs fewer time intervals in the game time model than several state-of-the-art complete and real-time search algorithms and about the same number of time intervals as the best compared complete search algorithm, even though it has the advantage over complete search algorithms that the agent starts to move right away.

Game Time Model

Video games often partition time into game cycles, each of which is only a couple of milliseconds long. Each game character executes one movement at the end of each game cycle, which gives the players the illusion of fluid movement. Our game time model is motivated by such video games. Time is partitioned into uniform time intervals, an agent can execute one movement during each time interval, and search and movements are done in parallel. The objective is to move the agent from its start location to its goal location in as few time intervals as possible. The game time model addresses the fact that the standard way of evaluating search algorithms, namely using their CPU times or path costs, is problematic in real-time situations. For example, complete search algorithms first find a complete path from the start location of the agent to its goal location and then move the agent along it. The complete search algorithm (forward) A*, for example, needs the smallest CPU time of any search algorithm to find cost-minimal paths (up to tie breaking). Yet, A* typically needs several time intervals to find a cost-minimal path from the start location of the agent to its goal location, resulting in a long delay before the agent starts to move (which makes the agent unresponsive) and a long time until it reaches its goal location (which makes the agent inefficient) since there is no parallelism of search and movement – the agent does not move until the path is found and does not search afterwards. Real-time search algorithms execute A* searches and movements in parallel and might be able to move the agent to its goal location (along a suboptimal path) in fewer time intervals than A*, even though both their CPU time and resulting path costs could be larger than those of A*.

RTBA* and TBAA*

Time-Bounded A* (TBA*) (Björnsson, Bulitko, and Sturtevant 2009) is an existing real-time search algorithm for undirected terrain that performs an A* search from the start location of the agent to its goal location. At the end of each time interval, the agent executes a movement towards a location in the OPEN list with the smallest f-value, by repeatedly either following the path from its start location to the location in the OPEN list with the smallest f-value (if the current location is on this path) or moving to the parent of its current location in the search tree. However, TBA* cannot be used when the terrain is not known initially. We thus extend it to on-line path planning with the freespace assumption in initially partially or completely unknown (but static) terrain in two steps, namely via RTBA* to TBAA*. Both new real-time search algorithms use on-line path planning with the freespace assumption by taking all obstacles into account that the agent has observed so far but assuming that unknown terrain is free of obstacles (Koenig, Tovey, and Smirnov 2003).

RTBA* In the first step, we extend TBA* to Restarting Time-Bounded A* (RTBA*). RTBA* is almost identical to TBA*; the difference is that whenever the agent observes obstacles on its current path to a location in the OPEN list with the smallest f-value, RTBA* starts a new TBA* search from the current location of the agent to its goal location.

TBAA* In the second step, we extend RTBA* to Time-Bounded Adaptive A* (TBAA*). Each time RTBA* starts a new TBA* search, all information from the previous TBA* search is lost. However, real-time search algorithms often update the h-values to make them more informed. TBAA*

*The material is based upon research supported by NSF (while Sven Koenig was serving at NSF). It is also based upon research supported by ARL/ARO under contract/grant number W911NF-08-1-0468 and ONR in form of a MURI under contract/grant number N00014-09-1-1031. The research was performed while Carlos Hernández visited the University of Southern California. Jorge Baier was partly funded by Fondecyt grant number 11110321.

is almost identical to RTBA*; the difference is that it updates h-values of locations in the way (Lazy) Adaptive A* (Koenig and Likhachev 2006a) and Real-Time Adaptive A* (RTAA*) (Koenig and Likhachev 2006b) do to focus future TBA* searches better. TBAA* performs this h-value update only once the h-value of a location is needed by the current TBA* search for the first time, to avoid computing those h-values that are not needed later.

Analysis

We prove in our AAMAS paper that RTBA* and TBAA* correctly either move the agent from its start location to its goal location or detect that this is impossible. Many other real-time search algorithms cannot detect when there is no solution efficiently. Furthermore, RTBA* and TBAA* can eventually move the agent on a cost-minimal path from its start location to its goal location if they reset the agent into its start location whenever it reaches its goal location.

Experimental Evaluation

We evaluate all search algorithms experimentally in known and initially partially or completely unknown eight-neighbor grids with blocked and unblocked cells. The user-provided h-values are the octile distances. The agent knows the dimensions of the grid and its start and goal cells. It can always move from its current unblocked cell to one of the eight unblocked neighboring cells with cost one for horizontal or vertical movements and cost $\sqrt{2}$ for diagonal movements. We run all search algorithms with time intervals whose lengths range from 0.3 to 1.5 milliseconds and record the average number of time intervals and number of movements until the agent reaches the goal cell for the first time. We use six game maps (from Nathan Sturtevant's www.movingai.com) and generate 300 search problems with randomly chosen start and goal cells for each game map, for a total of 1,800 search problems. In known grids, the agent knows the blockage status of all cells initially. In initially partially or completely unknown grids, it does not know the blockage status of some or all, respectively, cells initially but always observes the blockage status of its eight neighboring cells. It can use path planning with the freespace assumption by assuming that all cells are unblocked, that is, edges connect every cell to its neighboring cells. If, after executing a movement, it observes that a neighboring cell is blocked, it increases the costs of all incoming and outgoing edges of that cell to infinity, which is equivalent to removing the edges.

Known Terrain We use the game time model to compare the real-time search algorithm TBA* in our AAMAS paper against the complete search algorithm (forward) A* and the real-time search algorithms RTAA* and daRTAA* (Hernández and Baier 2012). Our experimental results are as follows: All real-time search algorithms move the agent in known terrain from its start cell to its goal cell in about the same or more time intervals than A*. However, TBA* moves the agent to its goal cell in fewer time intervals than the two real-time search algorithms RTAA* and daRTAA* and in about the same number of time intervals as A*. TBA* has the advantage over A* that the agent moves right away.

Initially Partially or Completely Unknown Terrain We use the game time model to compare RTBA* and TBAA* in our AAMAS paper against the complete search algorithms (forward) Repeated A*, Adaptive A*, and D* Lite and the real-time search algorithms RTAA* and daRTAA*. Repeated A* is almost identical to (forward) A*; the difference is that Repeated A* starts a new A* search from the current cell of the agent to its goal cell whenever the agent observes blocked cells on its current path to its goal cell. Incremental search algorithms, such as Adaptive A* (Koenig and Likhachev 2006a) and D* Lite (Koenig and Likhachev 2002), behave in the same way but speed up the A* searches by using their experience with prior A* searches to speed up future ones. Our experimental results are as follows: All real-time search algorithms move the agent in initially partially or completely unknown terrain from its start cell to its goal cell in fewer time intervals than Repeated A*. The game time model is thus able to explain the importance of real-time search in this case. TBAA* moves the agent in initially partially or completely unknown terrain from its start cell to its goal cell in fewer time intervals than the two complete search algorithms Repeated A* and Adaptive A* and the two real-time search algorithms RTAA* and daRTAA* and in about the same number of time intervals as the best compared complete search algorithm D* Lite. TBAA* seems to have a slight advantage over D* Lite in initially partially unknown terrain and vice versa in initially completely unknown terrain (although this difference might not be statistically significant). In both cases, TBA* has the advantage over D* Lite that the agent moves right away.

References

Björnsson, Y.; Bulitko, V.; and Sturtevant, N. 2009. TBA*: Time-bounded A*. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 431–436.

Hernández, C., and Baier, J. 2012. Avoiding and escaping depressions in real-time heuristic search. *Journal of Artificial Intelligence Research* 43:523–570.

Koenig, S., and Likhachev, M. 2002. D* Lite. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, 476–483.

Koenig, S., and Likhachev, M. 2006a. A new principle for incremental heuristic search: Theoretical results. In *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS)*, 402–405.

Koenig, S., and Likhachev, M. 2006b. Real-Time Adaptive A*. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 281–288.

Koenig, S.; Tovey, C.; and Smirnov, Y. 2003. Performance bounds for planning in unknown terrain. *Artificial Intelligence* 147(1-2):253–279.