



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 2      Issue: VII      Month of publication: July 2014**

**DOI:**

**[www.ijraset.com](http://www.ijraset.com)**

**Call: ☎ 08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

## Deadline Groovy Search

Shivani Chaudhary

M.Tech, Department of Computer Science, DCRUST Murthal

**Abstract--** A new approach for the contract search problem, the called Deadline Aware Search (DAS). DAS reacts to the approaching deadline during search. A method of measuring heuristic error on-line and using those errors in a model to construct corrected heuristic estimates is proposed and used in DAS.

**Keywords--** Gridworld navigation, 15-puzzle, Dynamic robot navigation.

### I. INTRODUCTION

A new approach for the contract search problem, the called Deadline Aware Search (DAS). Unlike anytime search algorithms that do not alter their search strategy in reaction to the approaching deadline, DAS reacts to the approaching deadline during search. We begin by presenting a general overview of the algorithm and its behavior. We then discuss the estimation of two quantities needed by the algorithm: the maximum achievable search depth  $d_{max}$  and distance to the cheapest solution beneath a node  $d_{cheapest}(s)$ . Finally, we discuss DAS's technique for recovering from situations in which it estimates that no goal is reachable given the current search behavior.

### II. NEW APPROACH

DAS is a simple approach, derived directly from the objective of contract search. It expands, among all the states leading to solutions deemed reachable within the time remaining, the one with the minimum  $f(s)$  value. Pseudo-code of the algorithm is presented in Figure 1.

**Deadline Aware Search** (starting state, deadline)

1.  $open \leftarrow \{\text{starting state}\}$
2.  $pruned \leftarrow \{\}$
3.  $incumbent \leftarrow \text{NULL}$
4. while (time) < (deadline) and open is non-empty
5.      $d_{bound} \leftarrow \text{calculate\_d\_bound}()$
6.      $s \leftarrow \text{remove state from open with minimal } f(s)$
7.     if  $s$  is a goal and is better than incumbent
8.      $incumbent \leftarrow s$

9.     else if  $\hat{d}(s) < d_{bound}$
10.     for each child  $s'$  of state  $s$
11.     add  $s'$  to open
12.     else
13.     add  $s$  to pruned
14.     if open is empty
15.      $recover\_pruned\_states(open, pruned)$
16.      $recover\_pruned\_states(open, pruned)$
17. return incumbent
18.  $exp \leftarrow \text{estimated expansions remaining}$
19. while  $exp > 0$  and pruned is non-empty loop
20.      $s \leftarrow \text{remove state from pruned with minimal } f(s)$
21.     add  $s$  to open
22.      $exp = exp - \hat{d}(s)$
- 23.

Figure 1: Pseudo-Code Sketch of Deadline Aware Search

The open list is first initialized with the starting state and then the search proceeds to expand nodes from the open list until either the search time expires or the open list is empty (indicating that there is no solution deemed reachable). At each iteration of the algorithm, the state with minimal  $f(s)$  is selected for expansion and the current maximum reachable distance,  $d_{max}$ , is calculated. If the distance to this state's best goal  $d_{cheapest}(s)$  is less than  $d_{max}$ , it is expanded and its children are added to the open list. Otherwise, it is added to the pruned list and the search will select the next best node for expansion.

Reachability, as estimated by DAS, is a function of a state's distance from its best possible solution,  $d_{cheapest}(s)$ . When there is not enough time to explore all interesting paths in the search

# INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

space, it makes sense to favor those paths that are closer to solutions. One result of using an admissible heuristic  $h(s)$  is that often the best  $f$  value of the states under a particular state will be higher than the value of  $f(s)$ . Assuming this heuristic error is distributed across a search space, the states that are farther from solutions have the potential of experiencing this increase in  $f$  value more often before reaching their respective solutions than states that are closer. For this reason, when selecting states for expansion ties on  $f(s)$  are broken in favor of smaller  $h(s)$ .

### III. EMPIRICAL ANALYSIS

We performed an empirical analysis over several benchmark domains in order to evaluate the performance of Deadline Aware Search in comparison to Anytime Repairing A\*, Restarting Weighted A\*, and Contract Search. In each domain

we tested over a range of deadlines covering around four orders of magnitude. In an attempt to judge algorithms fairly in the case that no solution is found within the deadline, all algorithms are required to run an initial search algorithm we call "Speedier". Speedier search is a greedy search on the cheapest(s) in which duplicate states are detected and, if already expanded, are ignored. This search completes very quickly, returning what is typically a highly suboptimal solution that all algorithms use as an incumbent. Therefore any algorithm that fails to find an improved solution within the deadline will return this sub-optimal Speedier solution. This both simplifies our analysis by assigning a meaningful cost to the null solution and is a realistic implementation in the case of any setting in which returning no solution is absolutely unacceptable.

For ARA\* and RWA\* we evaluated the following range of initial weight settings: 1.2, 1.5, 3.0, 6.0, 10.0 and a weight decrement of 0.2. The optimal initial weight setting found for the 15-Puzzle, Weighted 15-Puzzle, Unit-Cost Grid-World, Life-Cost Grid-World, and Dynamic Robot Navigation were 3.0, 3.0, 3.0, 6.0, and 1.5, respectively. In each plot the results for the top two weight settings are illustrated, as there were settings which did not produce the best result overall but performed better for a specific range of deadlines.

The deadline is still measured in seconds and when it arrives the algorithm returns the best solution found thus far.

#### 1) 15-Puzzle

Experiments were performed on the 100 instances of the 15-Puzzle using the Rohini distance heuristic (Korf 100 tiles). We evaluated both a uniform cost model as well as a model in which the cost of moving each tile was the inverse of the numeric value of the tile (1-15). Results are shown in Figure 2. The X-axis of the plots represents the deadline in seconds and is displayed on a logarithmic scale. The Y-axis of the plot represents solution quality, being defined as cost of the best solution found by any algorithm for the particular instance over the achieved solution cost. Solution quality is used rather than raw solution cost to reduce because we have many domains in which individual instances may have very different optimal solution costs. It is a standard metric used in the satisficing track of the International Planning Competition. In both cost models of the 15-Puzzle domain Deadline Aware Search is a clear improvement over both ARA\* and Contract Search for the full range of deadlines.

Figure 2 shows that with the short deadline of only around 0.5 seconds, DAS was able to find, on average, the same quality of solutions that took ARA\* with an optimal weight setting would find with more than twice that much time. For the standard tiles domain, contract search is competitive with deadline aware search for large deadlines, where both algorithms solve the problem optimally, and for small deadlines, where both algorithms return the Speedier solution.

#### 2) Dynamic Robot Navigation

The objective in these problems is to find the fastest path from the starting location of the robot to some goal location and heading, taking motion dynamics such as momentum into account. The instances used in our experiments were 500 by 500 cells in size. We scatter 75 lines, up to 70 cells in length, with random orientations across the domain and present results averaged over 100 instances. Results are shown in Figure 2. Results in this domain show Deadline Aware Search as a clear improvement over ARA\*, RWA\*, and Contract Search for the full range of deadlines. Contract Search performed particularly weakly in this domain. We believe this is in part attributed to the fact that the domain has a fairly accurate, albeit inadmissible, cheapest(s) heuristic. Taking advantage of this heuristic allows Deadline Aware Search to more accurately decide the reachability of states and may have contributed to its success.



# INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Outside of tie breaking, it is not obvious how the other algorithms could make use of distance estimates.

## 3) Grid-World Navigation

Experiments were performed on two sets of four-way movement grid-world navigation problems; unit-cost and lifecost. In both domains the starting state is in the lower-left corner of a 2000x1200 map with the goal state in the lower-right corner. Obstacles are distributed randomly and uniformly with a probability of 0.35. The life-cost grid-world, first proposed by Ruml and Do (2007), varies the cost of movement in different layers within the grid creating a clear distinction between shortest path and cheapest path. The cost of traversing each square in the map is equal to the Y coordinate of that location, with (0,0) being the bottom left corner. This implies that the cheapest path through the map would be to traverse to the top of the map, across, then back down to the solution. Results are shown in Figure 2. In the case of life-cost grid-world Deadline Aware Search was competitive with the best of the anytime methods at the optimal parameter settings for shorter deadlines and provided improved results for larger deadlines.

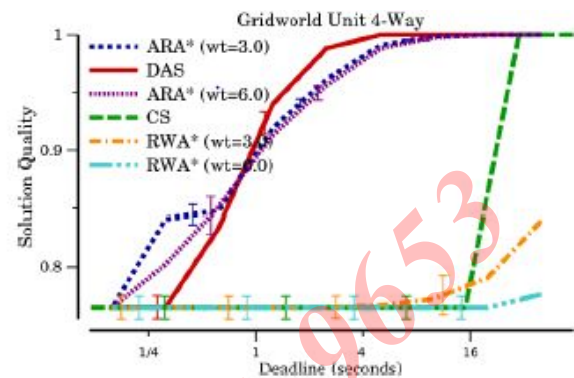


Figure2 Solution quality for a given solving time deadline

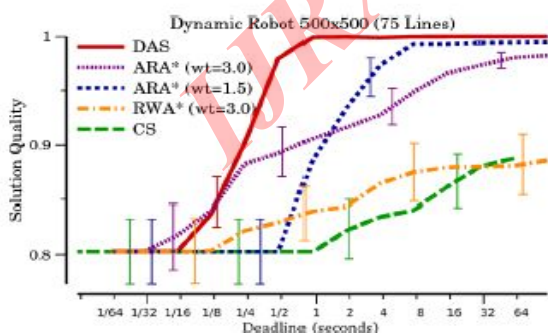
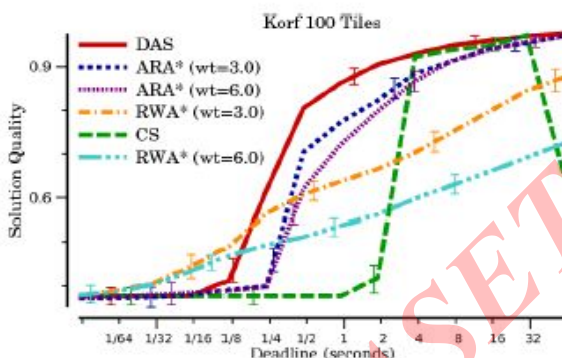
## IV. DAS BEHAVIOR

The purpose of  $d_{max}$  in DAS is to act as a decreasing upperbound on the distance between any state expanded and its respective best solution. This bound is intended to force the search to progress forwards to meet a particular deadline when it would normally have spent more time exploring different partial solutions sorting out the increasing  $f(s)$  values. Depending on the given deadline, the value should typically fall somewhere within the range of current  $b d(s)$  values such that some pruning will occur when

necessary and not all states will be pruned unnecessarily. In order to evaluate the behavior of  $d_{max}$  relative to the  $b d(s)$  of states expanded during a Deadline Aware Search, implemented a version which uses a limit on the number of state expansions as a deadline and records relevant information during the search. This way the overhead of recording could be factored out of the analysis.

## V. CONCLUSION

The choice to include a Speedier search first in our empirical analysis for all algorithms could lead to biasing results such that algorithms which do not return any solution before the deadline are rated closer to those which return sub-optimal solutions which are only marginally better than the Speedier solution. This is not a significant issue in our results, as most solutions found after the Speedier solution are substantially improved. We recorded the number of times each



# INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

algorithms succeeded in improving on the Speedier solution. One can see that in some domains such as the sliding tiles puzzle, the average solution quality for DAS was generally higher than the anytime approaches. One can also see that part of the success of DAS on Dynamic Robots and the Life-Cost Grid-world Navigation problems comes from the fact that it returns significantly more improved solutions for shorter deadlines than the other approaches. Overall, the experimental results indicate that Deadline Aware Search can lead to a significant improvement over ARA\* and RWA\* in some domains while remaining competitive in others.

## ACKNOWLEDGEMENT

I gratefully acknowledge support from the CSI professors for their responses against the doubts that congested the mind as a whole & especially to Rakesh Chawla for his responses to my inquiries about different anytime approaches.

## REFERENCES

- [1] Aine, S.; Chakrabarti, P.; and Kumar, R. 2010. Heuristic search under contract. *Computational Intelligence* 26(4):386–419.
- [2] Aine, S. 2011. Personal communication.
- [3] Dechter, R., and Pearl, J. 1988. The optimality of A\*. In Kanal, L., and Kumar, V., eds., *Search in Artificial Intelligence*. Springer-Verlag. 166–199.
- [4] Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems Science and Cybernetics* SSC-4(2):100–107.
- [5] Hiraishi, H.; Ohwada, H.; and Mizoguchi, F. 1998. Time constrained heuristic search for practical route finding. In *Pacific Rim International Conferences on Artificial Intelligence*.
- [6] Korf, R. E. 1990. Real-time heuristic search. *Artificial Intelligence* 42:189–211.
- [7] Likhachev, M.; Gordon, G.; and Thrun, S. 2003. ARA\*: Anytime A\* with provable bounds on sub-optimality. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS-03)*.
- [8] Pohl, I. 1973. The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computation issues in heuristic problem solving. In *Proceedings of IJCAI-73*, 12–17.
- [9] Richter, S.; Thayer, J. T.; and Ruml, W. 2009. The joy of forgetting: Faster anytime search via restarting. In *Symposium on Combinatorial Search*.
- [10] Ruml, W., and Do, M. B. 2007. Best-first utility-guided search. In *Proceedings of IJCAI-07*, 2378–2384.
- [11] Thayer, J. T., and Ruml, W. 2010. Anytime heuristic search: Frameworks and algorithms. In *SoCS 2010*.
- [12] Thayer, J. T.; Dionne, A.; and Ruml, W. 2011. Learning inadmissible heuristics during search. In *Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling (ICAPS-11)*.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)