On the Benefits of Randomly Adjusting Anytime Weighted A*

Abhinav Bhatia, Justin Svegliato and Shlomo Zilberstein

College of Information and Computer Sciences, University of Massachusetts Amherst

{abhinavbhati, jsvegliato, shlomo}@cs.umass.edu

Abstract

- Anytime Weighted A* (Hansen and Zhou, 2007; Hansen, Zilberstein, and Danilchenko, 1997) is an anytime heuristic search algorithm that uses a weight to scale the heuristic to manage the trade-off between solution quality and running time.
- We propose a randomized version of this algorithm, called *Randomized Weighted A**, that randomly adjusts its weight at runtime
- RWA* typically outperforms AWA* with static weights on a range of benchmark problems.

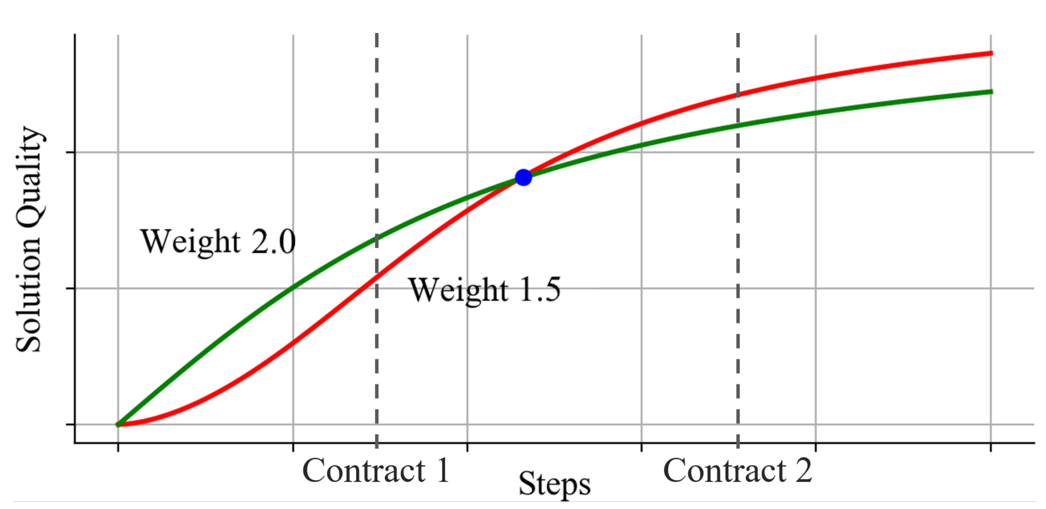

Figure: An example of two executions of AWA*. Weight=2.0 is better when the available running time is contract 1.

Introduction

- **Contract setting**: A fixed computation time is available to solve a problem.
- **Trade-off**: Higher weights lead to better solutions in short-term.
- **Best weight** depends on the characteristics of the domain, the details of the instance, and the available computation time (contract duration).
- Tune best static weight for a problem (Hansen and Zhou, 2007)
- Tune at runtime heuristically (Sun, Druzdzel, and Yuan, 2007; Thayer and Ruml, 2009, 2008)
- Adjust at runtime using deep-RL (Bhatia, Svegliato, and Zilberstein, 2021)
- **Adjust at runtime randomly?**
  Advantage: simplicity, no hyperparameters, no offline experimentation.

Randomized Weighted A*

- **For each node expansion, RWA* samples a weight uniformly** from a fixed set of weights e.g., $w \sim W = \{1, 1.5, 2, 3, 4, 5\}$.
- RWA* maintains an open list corresponding to each w. Same nodes, different priorities.
- Operate on the open lists in parallel for efficiency.

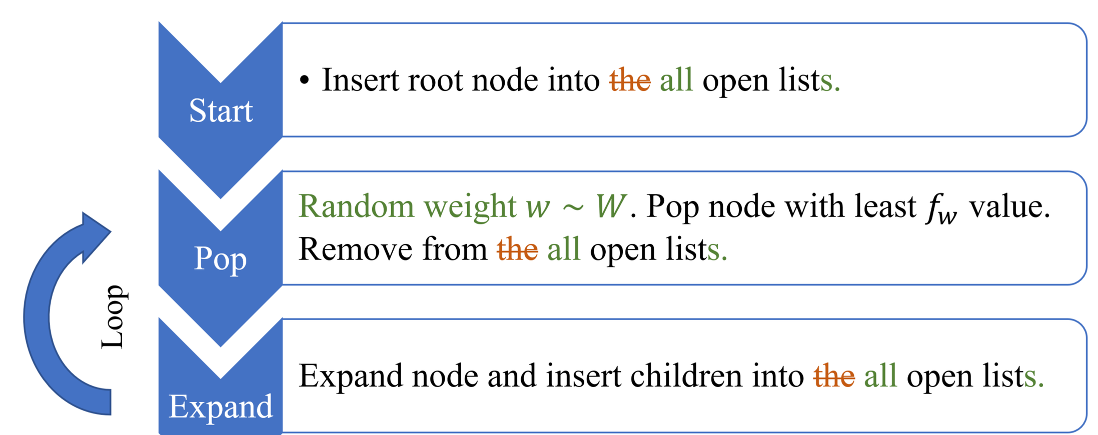

Figure: Differences between RWA* and AWA*

Experimental Setup

- **RWA* $\mathbf{w \sim \{1, 1.5, 2, 3, 4, 5\}}$** vs AWA* with static weights 1, 1.5, 2, 3, 4, 5 (commonly used).
- **Domains**: Sliding-Puzzle (SP), Inverse-Sliding-Puzzle (ISP), Travelling-Salesman-Problem (TSP), City-Navigation-Problem (CNP)
- **500 instances** per domain of varying difficulty.
- **Node-expansions budget** (contract) of 6000 for SP, ISP and 3000, 2400 for TSP, CNP.

Results

1. RWA* computes solutions with a **higher quality on average** than any static weight,
2. has the **highest probability of computing a solution that is at least as good as any other approach**,
3. has the **highest probability of computing at least one solution** compared to any static weight.

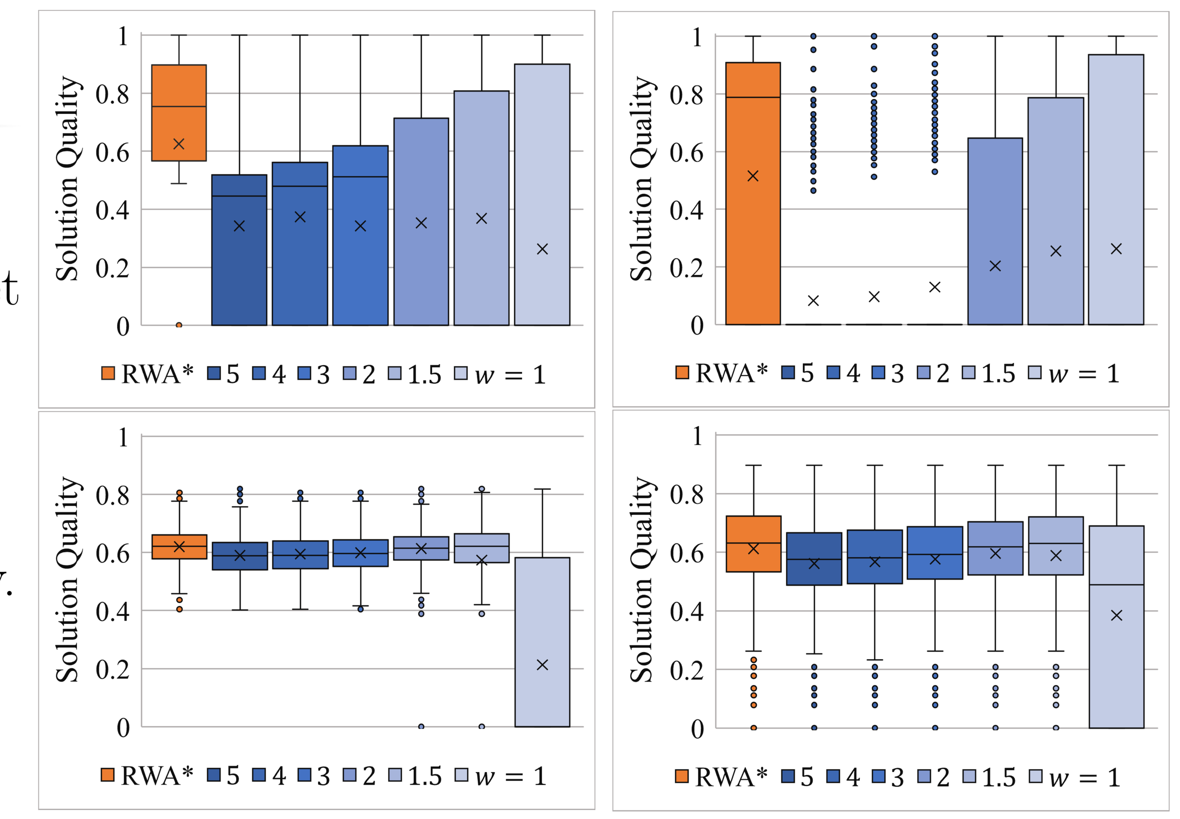

Figure: The solution quality box plots for the SP, ISP, TSP, and CNP benchmark problems (*top-left* to *bottom-right*). The *crosses* denote the mean and the *bullets* denote the outliers.

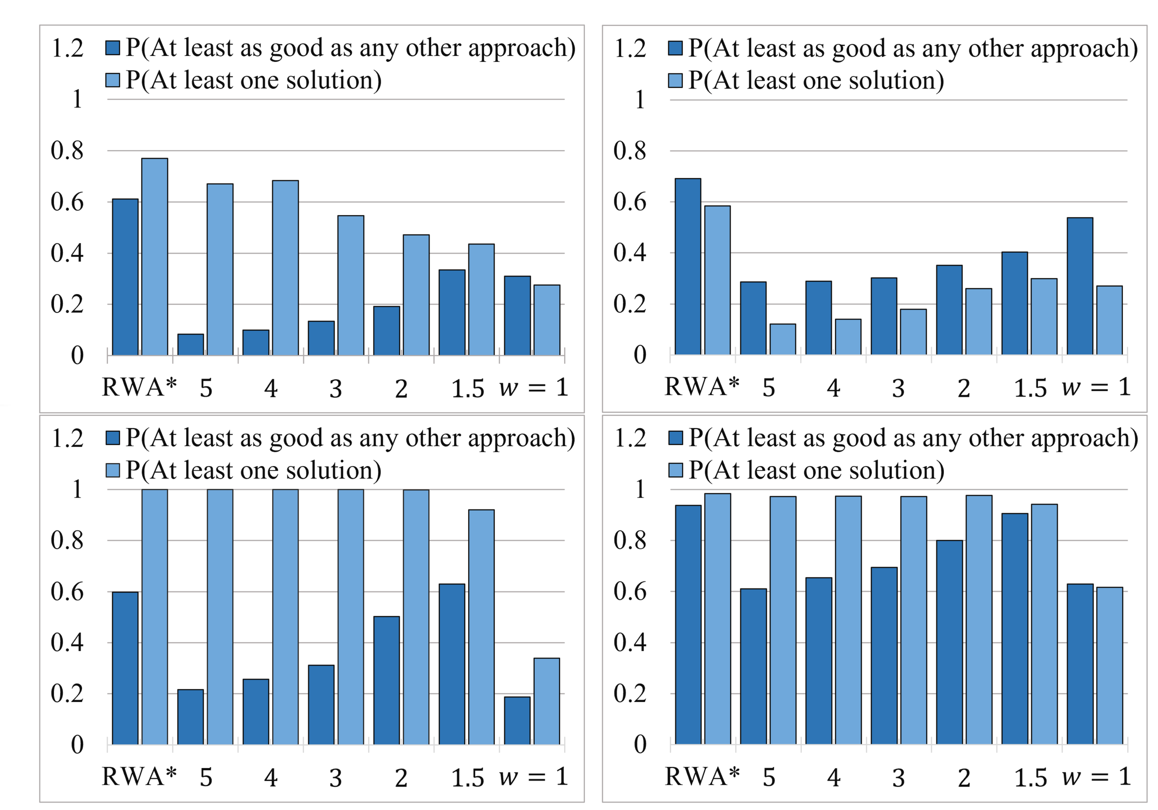

Figure: The performance bar graphs for the SP, ISP, TSP, and CNP benchmark problems (*top-left* to *bottom-right*).

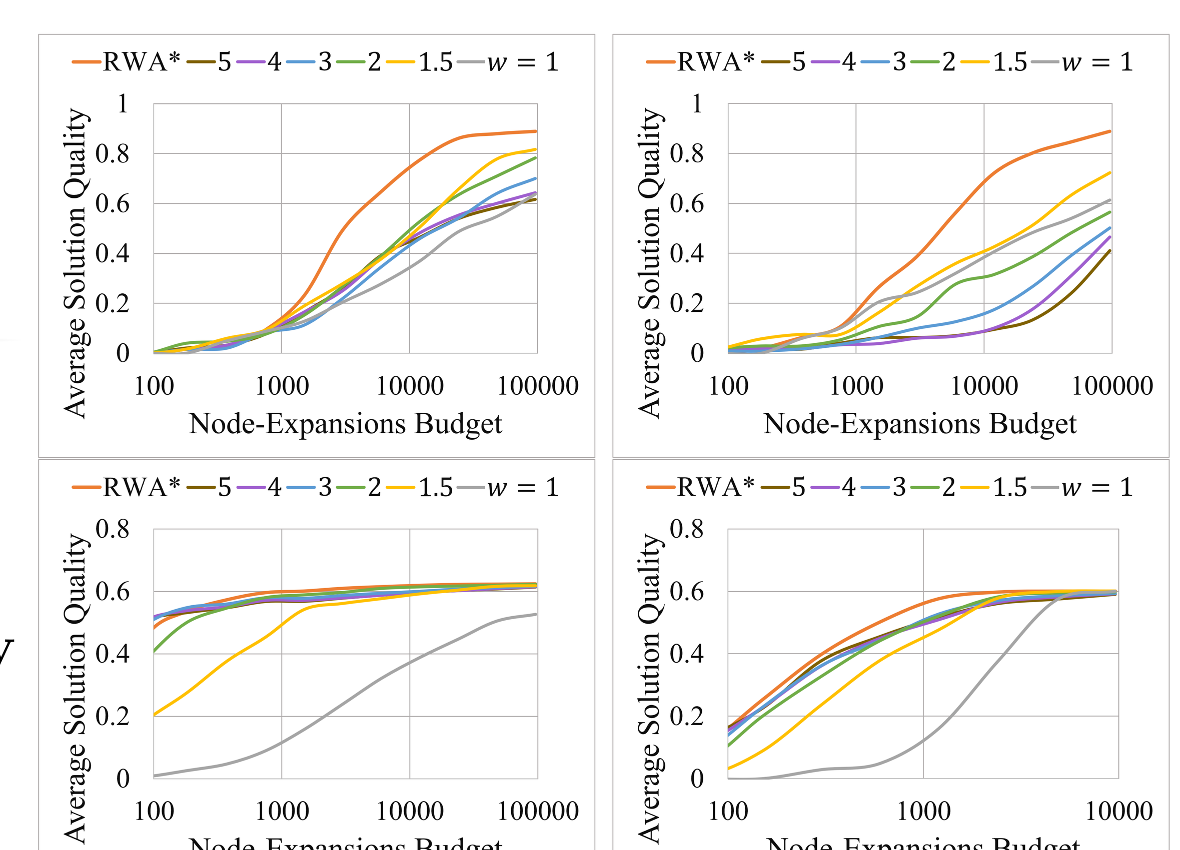

Figure: Each approach compared across a range of contract durations on SP, ISP, TSP, CNP (*top-left* to *bottom-right*).

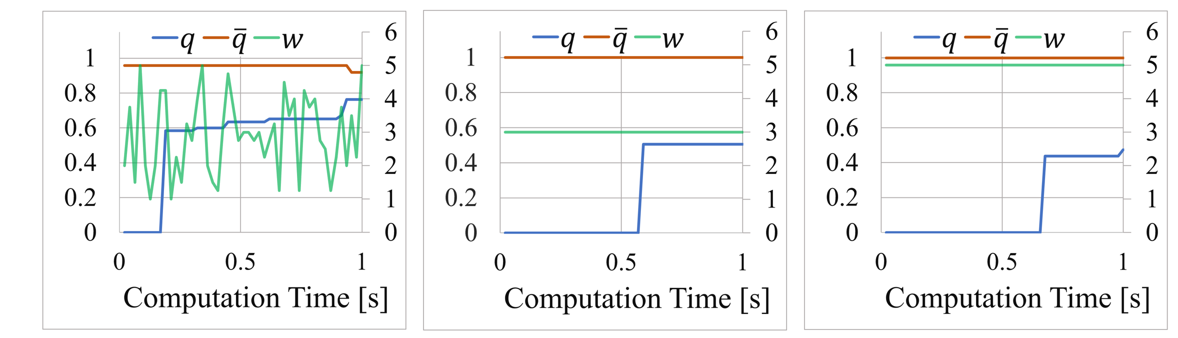

Figure: RWA* *(left)* and AWA* with a weight of $w = 3$ and $w = 5$ (*center* and *right*) on a specific instance of the SP benchmark problem. The weight curves are smoothed and plotted on the secondary vertical axis.

Conclusion

- RWA* (i) computes better solutions on average, (ii) exhibits a higher probability of computing any solution at all, and (iii) exhibits a higher probability of computing a solution at least as good as any static weight of AWA* in a contract setting across a range of contract durations on our benchmark domains.
- **RWA* is appealing because it is easy to implement and effective without any extensive offline experimentation or parameter tuning.**

References

Bhatia, A.; Svegliato, J.; and Zilberstein, S. 2021. Tuning the Hyperparameters of Anytime Planning: A Deep Reinforcement Learning Approach. In *ICAPS 2021 HSDIP*.

Hansen, E. A.; and Zhou, R. 2007. Anytime heuristic search. *Journal of Artificial Intelligence Research* 28: 267–297.

Hansen, E. A.; Zilberstein, S.; and Danilchenko, V. A. 1997. Anytime heuristic search: First results. Technical report.

Sun, X.; Druzdzel, M. J.; and Yuan, C. 2007. Dynamic weighting A* search-based MAP algorithm for Bayesian networks. In *20th IJCAI*, 2385–2390.

Thayer, J.; and Ruml, W. 2009. Using distance estimates in heuristic search. In *19th International Conference on Automated Planning and Scheduling*, 382–385.

Thayer, J. T.; and Ruml, W. 2008. Faster than weighted A*: An optimistic approach to bounded suboptimal search. In *18th International Conference on Automated Planning and Scheduling*, 355–362.