Exceeding Human Intelligence in a Non-playable Character

Joe Template and **Jane Example**
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8, Canada
`{jtemplate|jexample}@ualberta.ca`

Abstract

Introduce and motivate the problem. State the main idea of your solution approach and highlights of its evaluation.

1 Introduction

Informally introduce and motivate your problem here.

2 Problem Formulation

Define the problem formally, including the performance criteria motivated by the introduction. The reader needs to know how to assess your approach's success or failure.

3 Related Work

Review related research and discuss its shortfalls with respect to the problem in the previous section.

4 Proposed Approach

Describe your ideas for the proposed approach informally at first. Show how the proposed approach addresses the shortfalls of the related research that you discussed above. Then describe the approach formally and precisely enough to be reproduced by an independent researcher.

If you decide to use pseudocode (e.g., Algorithm 1)[1], make sure you complement it with a plain English description. Additionally hand trace your pseudocode line by line (make sure you refer to the line numbers in your pseudo-code) on a simple example.

End this section formulating your hypothesis. For instance, "my algorithm converges faster than X but uses more memory than Y under conditions A,B,C". Link your hypothesis to the motivation of the problem you gave earlier in the paper. For instance, if your hypothesis is about fast real-time operation then you should link it to the requirements of the problem at hand described earlier in the paper.

[1]Reproduced with permission from work by Bulitko & Brown (2012).

5 Theoretical Analysis

Discuss expected properties of your algorithm. For instance, what is the run-time complexity in time and space? Under what assumptions is your algorithm guaranteed to find a solution? If it is a learning algorithm, under which conditions will it converge?

6 Empirical Evaluation

Support your hypothesis formulated above. First describe your empirical testbed in enough detail for an independent researcher to reproduce. Then describe all parameters of your algorithm as applied to the testbed. Finally present the results.

Do not be shy to list negative results as well — they still improve our understanding of the field. Make sure your describe your experimental set up in enough detail for other researchers to replicate your work.

A solid empirical study not only offers some findings but also sheds light on how confident we should be about them. This can be accomplished through statistical tests (e.g., the T-test) and/or confidence intervals (e.g., Figure 1).

Complement your figures with compact and readable tables (e.g., Table 1).

7 Discussion

Discuss your results with respect to the hypotheses you formulated and the demands of the problem at hand. Do not be shy to admit that you do not understand some of the results you obtained. Leave them as open questions for further research.

Algorithm 1: Real-time Q-learning with lookahead (γ, r, p, T)

```
initialize $Q_1$ to uniformly random values in $[0,1]$
for $t = 1, 2, \ldots, T$ do
    determine the amount of deliberation: $n_t \leftarrow \Delta(s_t)$
    set probe length: $m_t \leftarrow \lfloor\sqrt{n_t}\rfloor$
    number of probes: $k_t \leftarrow \lfloor\sqrt{n_t}\rfloor$
    initialize enhanced state-action return estimates: $Q'_t \leftarrow Q_t$
    for $a \in A$ do
        for $i = 1, \ldots, k_t$ do
            $s'_1 \leftarrow s_t$
            $\gamma_1 \leftarrow 1$
            for $j = 1, \ldots, m_t$ do
                select probe action: $a'_j \leftarrow \begin{cases} a & \text{if } j = 1 \\ \arg\max_{a' \in A} Q'_t(s'_j, a') & \text{otherwise.} \end{cases}$
                imagine the next state: $s'_{j+1} \leftarrow \Lambda(s'_j, a'_j)$
                imagine the reward to be collected: $r'_j \leftarrow r(s'_j, a'_j)$
                adjust the discount factor $\gamma_{j+1} \leftarrow \gamma_j \gamma$
            imagine the residual reward $r'_{m_t+1} \leftarrow \max_{a \in A} Q'_t(s'_{m_t+1}, a)$
            for $j = 1, \ldots, m_t$ do
                update $Q'_t(s'_j, a'_j) \leftarrow (1-\alpha_t) Q'_t(s'_j, a'_j) + \alpha_t \sum_{x=j}^{m_t+1} \gamma_x r'_x$
                adjust discount factor: $\gamma_x \leftarrow \gamma_x / \gamma, x \in \{j+1, \ldots, m_t+1\}$
    execute the action: $a_t \leftarrow \arg\max^{\epsilon_t}_{a \in A} Q'_t(s_t, a)$
    observe the resulting state $s_{t+1}$
    update the return estimate using the collected reward and the new state:
      $Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha_t) Q_t(s_t, a_t) + \alpha_t [r(s_t, a_t) + \gamma \max_{a \in A} Q_t(s_{t+1}, a)]$
```

Table 1: Typical results averaged over 50 convergence runs on 10 maps. The average shortest path length is 59.6. Reproduced with permission from work by Bulitko, Sturtevant, & Kazakevich (2005).

Algorithm	1st move time	Conv. travel	Subopt.
A*	5.01 ms	186	0.0%
LRTA*	0.02 ms	25,868	0.0%
LRTS	0.93 ms	555	2.07%
PR-LRTS	**0.95 ms**	**345**	**2.19%**

8 Future Work

Describe possible extensions (e.g., if you were to take on this project as your thesis work).

9 Conclusions

Refresh the reader on the problem at hand, motivation, shortfalls of the previous work, ideas of your new algorithm and the results of the empirical study. Briefly re-state the contributions your paper makes here.

Acknowledgments

Acknowledge any help you already received.

References

Bulitko, V., and Brown, M. 2012. Flow maximization as a guide to optimizing performance: A computational model. *Advances in Cognitive Systems* 2:239–256.

Bulitko, V.; Sturtevant, N.; and Kazakevich, M. 2005. Speeding up learning in real-time search via automatic state abstraction. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1349 – 1354.

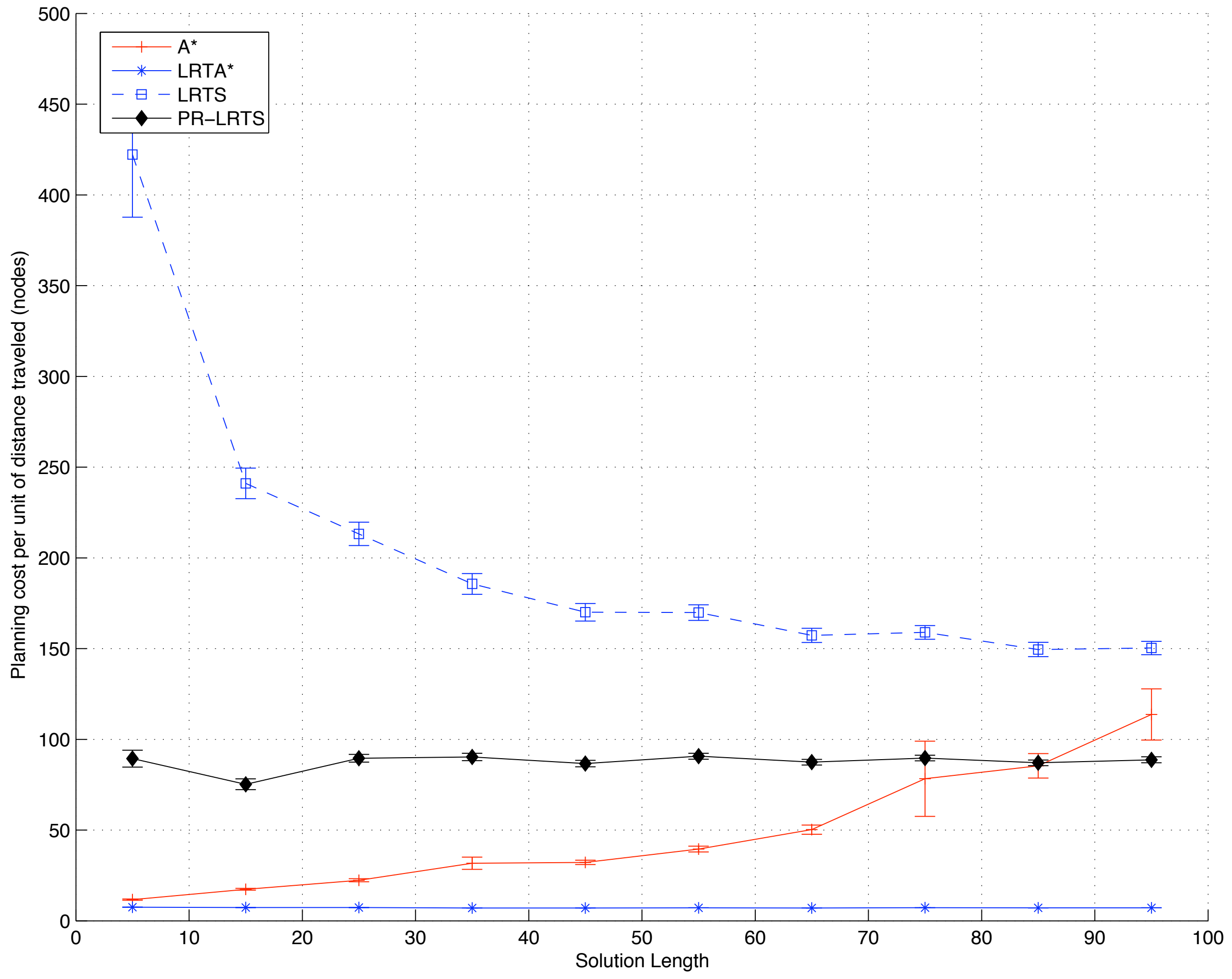

Figure 1: Planning cost of various algorithms during convergence on initially unknown maps. Problems are bucketed according to their difficulty. Each of the ten buckets contains 100 random path-planning problems.