To appear in the *Proceedings of ECP-01*, Toledo (Spain), Sept. 2001

Combining Two Fast-Learning Real-Time Search Algorithms Yields Even Faster Learning*

David Furcy and Sven Koenig

Georgia Institute of Technology
College of Computing
Atlanta, GA 30332-0280
{dfurcy, skoenig}@cc.gatech.edu

Abstract. Real-time search methods, such as LRTA*, have been used to solve a wide variety of planning problems because they can make decisions fast and still converge to a minimum-cost plan if they solve the same planning task repeatedly. In this paper, we perform an empirical evaluation of two existing variants of LRTA* that were developed to speed up its convergence, namely HLRTA* and FALCONS. Our experimental results demonstrate that these two real-time search methods have complementary strengths and can be combined. We call the new real-time search method eFALCONS and show that it converges with fewer actions to a minimum-cost plan than LRTA*, HLRTA*, and FALCONS.

1 Introduction

Real-time (heuristic) search methods have been used to solve a wide variety of planning problems. Learning Real-Time A* (LRTA*) [7] is probably the best-known real-time search method. Unlike traditional search methods it can not only act in real-time but also amortize learning over several planning episodes if it solves the same planning task repeatedly. This allows it to find a suboptimal plan fast and then improve the plan until it follows a minimum-cost plan. Researchers have recently attempted to speed up its convergence while maintaining its advantages over traditional search methods, that is, without increasing its lookahead. Ishida and Shimbo, for example, developed ε-search to speed up the convergence of LRTA* by sacrificing the optimality of the resulting plan [5, 6]. In this paper, on the other hand, we study real-time search methods that speed up the convergence of LRTA* without sacrificing optimality, namely HLRTA* [8] (which is similar to SLRTA* [1]) and our own FALCONS [2]. We present the first thorough empirical evaluation of HLRTA* and show that it and FALCONS have complementary strengths that can be combined. We call the resulting real-time

* We thank Stefan Edelkamp for introducing us to HLRTA*, Richard Korf for making Thorpe's thesis about HLRTA* available to us, and James Irizarry for re-implementing HLRTA*. The Intelligent Decision-Making Group is partly supported by NSF awards to Sven Koenig under contracts IIS-9984827 and IIS-0098807 as well as an IBM faculty partnership award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.

Table 1. Comparison of HLRTA* and FALCONS with LRTA*

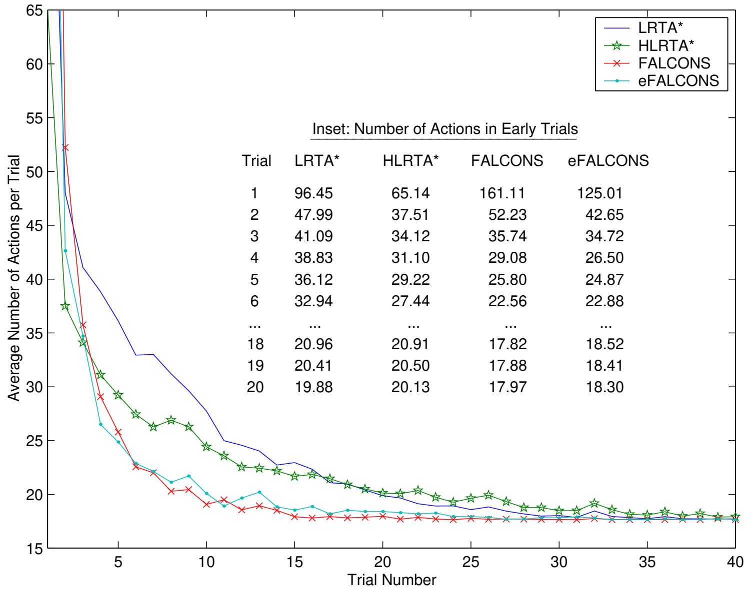
Performance Measure	Average Over	Speedup Over LRTA* HLRTA*	FALCONS
Number of Actions to Convergence	All 15 Cases	2.08%	18.84%
	7 Most Informed Cases	-0.69%	28.73%
	7 Least Informed Cases	4.50%	8.86%
Number of Trials to Convergence	All 15 Cases	-11.06%	42.33%
	7 Most Informed Cases	-16.68%	43.87%
	7 Least Informed Cases	-5.17%	40.52%
Number of Actions in First Trial	All 15 Cases	7.04%	-21.90%
	7 Most Informed Cases	11.65%	-49.81%
	7 Least Informed Cases	-0.99%	-0.14%

search method Even FAster Learning and CONverging Search (eFALCONS) and show that it converges with fewer actions to a minimum-cost plan than LRTA*, HLRTA*, and FALCONS, even though it looks at the same states when it selects successors on undirected graphs and is not more knowledge-intensive to implement.

2 Motivation for Combining HLRTA* and FALCONS

HLRTA* keeps the successor-selection rule of LRTA* but improves its value-update rule, while FALCONS keeps the value-update rule of LRTA* but improves its successor-selection rule. In the following, we compare these two real-time search methods with LRTA* averaged over 1000 runs in seven domains with two or three different heuristic functions each, for a total of fifteen distinct experimental cases that we have previously used in [3]. As required by the real-time search methods, all domains are finite, all of their states have finite goal distances, and all heuristic functions do not overestimate the true distances. We use three different performance measures. The main performance measure is the number of actions until convergence (that is, until the real-time search methods repeatedly execute a minimum-cost plan). The other performance measures are the number of trials to convergence and the number of actions in the first trial, where a trial consists of all actions until the goal is reached and the real-time search method is reset to the start. Table 1 summarizes our empirical results. The number of actions to convergence of FALCONS was smaller than that of HLRTA* and the number of actions to convergence of HLRTA* was smaller than that of LRTA*. In addition, we gained two other important insights:

– The number of trials to convergence of HLRTA* was larger than that of LRTA* but the number of actions in the first trial of HLRTA* was smaller than that of LRTA*. The opposite was true for FALCONS. Thus, the number of trials to convergence is a weakness of HLRTA* and the number of actions in the first trial is a weakness of FALCONS. Figure 1 illustrates this observation for one of the fifteen experimental cases, a four-connected gridworld with the Manhattan distance heuristic. The figure graphs the number of actions for each trial. The graph for HLRTA* started below that of LRTA*,



Trial	LRTA*	HLRTA*	FALCONS	eFALCONS
1	96.45	65.14	161.11	125.01
2	47.99	37.51	52.23	42.65
3	41.09	34.12	35.74	34.72
4	38.83	31.10	29.08	26.50
5	36.12	29.22	25.80	24.87
6	32.94	27.44	22.56	22.88
...	...	...	...	...
18	20.96	20.91	17.82	18.52
19	20.41	20.50	17.88	18.41
20	19.88	20.13	17.97	18.30

Fig. 1. Comparison of HLRTA* and FALCONS with LRTA* in a Four-Connected Gridworld with Manhattan Distance Heuristic

rose above it after the eighteenth trial (see inset) and then remained above it until the end of learning. The graph for FALCONS, on the other hand, started above that of LRTA*, dropped below it after the second trial and then remained below it until the end of learning.

– The improvement in number of actions until convergence of HLRTA* over LRTA* was smaller in the most informed cases than that of HLRTA* over LRTA* in the least informed cases. The opposite was true for FALCONS. Thus, the number of actions to convergence is a weakness of HLRTA* in the most informed cases and a weakness of FALCONS in the least informed cases.

Thus, there are two reasons for combining HLRTA* and FALCONS. First, the resulting real-time search method could reduce the number of actions to convergence by reducing the number of actions for early trials below that of FALCONS and the number of actions for later trials below that of HLRTA*. Second, the resulting real-time search method could be less sensitive to the level of informedness of the heuristic function and thus perform better across all experimental cases.

3 eFALCONS

LRTA* associates an h-value with every state to estimate the goal distance of the state (similar to the h-values of A*). LRTA* always first updates the h-value of the current state (value-update rule) and then uses the h-values of

Table 2. Comparison of eFALCONS with LRTA*, HLRTA* and FALCONS

Performance Measure	Average Over	Speedup of eFALCONS over LRTA*	HLRTA*	FALCONS
Number of Actions to Convergence	All 15 Cases	21.31%	19.34%	2.18%
	7 Most Informed Cases	28.73%	29.01%	0.00%
	7 Least Informed Cases	13.52%	9.53%	5.16%
Number of Trials to Convergence	All 15 Cases	36.79%	41.44%	-12.11%
	7 Most Informed Cases	38.13%	44.38%	-15.77%
	7 Least Informed Cases	34.29%	37.10%	-10.36%
Number of Actions in First Trial	All 15 Cases	-15.01%	-25.70%	4.72%
	7 Most Informed Cases	-36.86%	-56.21%	7.95%
	7 Least Informed Cases	0.38%	1.28%	0.53%

the successors to move to the successor believed to be on a minimum-cost path from the current state to the goal (action-selection rule). HLRTA* introduces h_s-values in addition to the h-values and uses them to modify the value-update rule of LRTA* so that the h-values converge faster to the goal distances. FALCONS, on the other hand, introduces g- and f-values in addition to the h-values (similar to the g- and f-values of A*) and uses them to modify the action-selection rule of LRTA* so that it moves to the successor believed to be on a shortest path from the start to the goal. eFALCONS, shown in Figure 2, then uses the value-update rule of HLRTA* for both the g- and h-values and the action-selection rule of FALCONS. eFALCONS and FALCONS access only the successors and predecessors of the current state, while LRTA* and HLRTA* access only the successors of the current state. Thus, all four real-time search methods access the same states on undirected graphs. A more detailed description of eFALCONS together with proofs of its properties is given in [4].

4 Empirical Study of eFALCONS

Table 2 compares eFALCONS with LRTA*, HLRTA*, and FALCONS. More detailed results are given in [4], including significance results obtained with the paired-samples Z test at the five-percent confidence level. The table demonstrates two advantages of eFALCONS over HLRTA* and FALCONS:

1. The number of trials to convergence of eFALCONS was 41.44 percent smaller than that of HLRTA*, and the number of actions in the first trial of eFALCONS was 4.72 percent smaller than that of FALCONS. We pointed out earlier that the number of trials to convergence was a weakness of HLRTA* and the number of actions in the first trial was a weakness of FALCONS. Thus, eFALCONS mitigates the weaknesses of both HLRTA* and FALCONS across performance measures. Indeed, Figure 1 shows that the number of actions of eFALCONS was smaller than that of FALCONS in the early trials and smaller than that of HLRTA* in the later trials. As a consequence, the number of actions to convergence of eFALCONS was 19.34 percent smaller than that of HLRTA* and 2.18 percent smaller than that of FALCONS. Thus, combining the value-update rule of HLRTA* and the action-selection

In the following, S denotes the finite state space; $s_{start} \in S$ denotes the start state; and $s_{goal} \in S$ denotes the goal state. $succ(s) \subseteq S$ denotes the set of successors of state s, and $pred(s) \subseteq S$ denotes the set of its predecessors. $c(s, s') > 0$ denotes the cost of moving from state s to successor $s' \in succ(s)$. We use the following conventions: $\arg\min_{s'' \in \emptyset}(\cdot) := \perp$, $\max_{s'' \in \emptyset}(\cdot) := -\infty$, and $\min_{s'' \in \emptyset}(\cdot) := \infty$. We use the following abbreviations, where $\perp$ means "undefined:"

$$\begin{aligned}
&\forall s \in S \text{ and } r \in succ(s): && h_s(r) := \begin{cases} h(r) & \text{if } dh(r) = \perp \text{ or } dh(r) \neq s \\ sh(r) & \text{otherwise,} \end{cases} \\
&\forall s \in S \text{ and } r \in pred(s): && g_s(r) := \begin{cases} g(r) & \text{if } dg(r) = \perp \text{ or } dg(r) \neq s \\ sg(r) & \text{otherwise, and} \end{cases} \\
&\forall r \in S: && f(r) := \max(g(r) + h(r), h(s_{start})).
\end{aligned}$$

The values are initialized as follows: $\forall r \in S$: $g(r) := h(s_{start}, r)$ and $h(r) := h(r, s_{goal})$, where $h(r, r')$ is a heuristic estimate of the distance from $r \in S$ to $r' \in S$. Furthermore, $\forall r \in S$: $dg(r) := \perp$, $dh(r) := \perp$, $sg(r) := \perp$, and $sh(r) := \perp$.

1. $s := s_{start}$.
2. $s' := \arg\min_{s'' \in succ(s)} f(s'')$.
   Break ties in favor of a successor s' with the smallest value of $c(s, s') + h_s(s')$.
   Break remaining ties arbitrarily (but systematically).

3 a. $p := \arg\min_{s'' \in pred(s)} (g_s(s'') + c(s'', s))$.
   $n := \arg\min_{s'' \in succ(s)} (c(s, s'') + h_s(s''))$.

b. Perform the following assignments in parallel:

$$\begin{aligned}
g(s) &:= \text{if } s = s_{start} \text{ then } g(s) \\
& \text{else } \max(g(s), \\
&\qquad\qquad g_s(p) + c(p, s), \\
&\qquad\qquad \max_{s'' \in succ(s)}(g(s'') - c(s, s''))). \\
sg(s) &:= \text{if } s = s_{start} \text{ then } g(s) \\
& \text{else } \max(g(s), \\
&\qquad\qquad \min_{s'' \in pred(s) \setminus \{p\}}(g_s(s'') + c(s'', s)), \\
&\qquad\qquad \max_{s'' \in succ(s)}(g(s'') - c(s, s''))). \\
dg(s) &:= p. \\
h(s) &:= \text{if } s = s_{goal} \text{ then } h(s) \\
& \text{else } \max(h(s), \\
&\qquad\qquad c(s, n) + h_s(n), \\
&\qquad\qquad \max_{s'' \in pred(s)}(h(s'') - c(s'', s))). \\
sh(s) &:= \text{if } s = s_{goal} \text{ then } h(s) \\
& \text{else } \max(h(s), \\
&\qquad\qquad \min_{s'' \in succ(s) \setminus \{n\}}(c(s, s'') + h_s(s'')), \\
&\qquad\qquad \max_{s'' \in pred(s)}(h(s'') - c(s'', s))). \\
dh(s) &:= n.
\end{aligned}$$

4. If $s = s_{goal}$, then stop successfully.
5. $s := s'$.
6. Go to 2.

Fig. 2. eFALCONS

rule of FALCONS indeed speeds up their convergence. The number of actions to convergence of eFALCONS, for example, is typically over twenty percent smaller than that of LRTA* and, in some cases, even over fifty percent smaller (not shown in the table).

2. The number of actions to convergence of eFALCONS was 29.01 percent smaller than that of HLRTA* in the most informed cases and 5.16 percent smaller than that of FALCONS in the least informed cases. We pointed out earlier that the number of actions to convergence was a weakness of HLRTA* in the most informed cases and a weakness of FALCONS in the least informed cases. Thus, eFALCONS mitigates the weaknesses of both HLRTA* and FALCONS across the levels of informedness of the heuristic function.

5 Conclusions

In this paper, we presented eFALCONS, a real-time search method that is similar to LRTA* but uses the value-update rule of HLRTA* and the action-selection rule of FALCONS. We showed experimentally that eFALCONS converges to a minimum-cost plan with fewer actions than LRTA*, HLRTA*, and FALCONS. For example, its number of actions to convergence is typically over twenty percent smaller than that of LRTA* and, in some cases, even over fifty percent smaller. It is future work to combine eFALCONS with ε-search to speed up its convergence even more by sacrificing the optimality of the resulting plan.

References

1. S. Edelkamp and J. Eckerle. New strategies in real-time heuristic search. In *Proceedings of the AAAI-97 Workshop on On-Line Search*, pages 30–35. AAAI Press, 1997.
2. D. Furcy and S. Koenig. Speeding up the convergence of real-time search. In *Proceedings of the National Conference on Artificial Intelligence*, pages 891–897, 2000.
3. D. Furcy and S. Koenig. Speeding up the convergence of real-time search: Empirical setup and proofs. Technical Report GIT-COGSCI-2000/01, College of Computing, Georgia Institute of Technology, Atlanta (Georgia), 2000.
4. D. Furcy and S. Koenig. eFALCONS: Speeding up the convergence of real-time search even more. Technical Report GIT-COGSCI-2001/04, College of Computing, Georgia Institute of Technology, Atlanta (Georgia), 2001.
5. T. Ishida. *Real-Time Search for Learning Autonomous Agents*. Kluwer Academic Publishers, 1997.
6. T. Ishida and M. Shimbo. Improving the learning efficiencies of real-time search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 305–310, 1996.
7. R. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, 1990.
8. P. Thorpe. A hybrid learning real-time search algorithm. Master's thesis, Computer Science Department, University of California at Los Angeles, Los Angeles (California), 1994.