Towards Real-Time Search with Inadmissible Heuristics

Masashi Shimbo[1] and **Toru Ishida**[2]

Abstract. Real-time search has two aspects, one as an efficient search method (in a single problem solving trial), and the other as an overall problem solving architecture with learning ability (through repeated trials). In both respects, the use of inadmissible (and hence inconsistent) heuristic functions bring some merits such as improved performance, but there is no theory yet that well explains when and why these algorithms benefit from them. In this paper, as a step towards fully understand and take advantage of such nonstandard heuristic functions, we discuss the properties of LRTA* and the Moving-Target Search (MTS) algorithms under heuristic functions violating admissibility or consistency. In particular, we show (1) the completeness of MTS with inconsistent or even inadmissible heuristics, and present (2) a new proof technique for the convergence of LRTA* which is applicable regardless of consistency.

1 INTRODUCTION

Unlike classic off-line heuristic search whose sole concern is the construction of a complete plan, real-time search [9] tries to solve the problem by situating itself in the environment, and interleaving shallow look-ahead search and action execution in an on-line manner. This indicates that the real-time search has an aspect of overall problem solving architecture besides being an (efficient) search algorithm. As each local planning is based on the partial information available at each stage of the problem solving, it is not guaranteed that the algorithms follow an optimal path, but they never fail to arrive at a goal (*completeness*). In addition, some algorithms including the Learning Real-Time A* (LRTA*) [9] have another attractive property to compensate for their sub-optimality, called *convergence*; these algorithms may not find an optimal solution in a single problem solving trial, but will eventually identify an optimal path through repeated trials.

This property of convergence makes them an attractive candidate for the model of intelligent autonomous agents [3], as it demonstrates their ability to learn from experience and to adapt themselves to unknown environments. However, [6] pointed out that the behavior of LRTA* during the learning process is far from being rational, as it is too "optimistic in the face of uncertainty [10, 8]" and hence tends too much towards exploration. As a remedy, they showed that it is possible to make the agents more cautious by addition of small weights to heuristic function. It was also demonstrated that it may lead to performance improvement.

Adding such weights results in an *inadmissible* (and hence *inconsistent*) heuristic function. There are several other factors that justify the use of such heuristic functions as well. (1) It is sometimes difficult to construct a heuristic function that is admissible as well as effective. (2) Being sub-optimal search method by nature, (single-trial) real-time search has little reason to adhere to admissibility, a requisite for off-line search methods devised for optimal solutions. (3) The real-world problems addressed by AI are often inherently intractable, so one should give up the optimality anyway.

As such nonstandard heuristic functions invalidate several fundamental assumptions, that were essential in proving the properties of real-time search in the standard case, e.g., monotonic increase of the heuristic estimates, the question arises whether they preserve the properties. This paper tries to clarify the effect incurred by the heuristic functions violating admissibiilty or consistency, upon the two popular real-time search algorithms LRTA* and the Moving-Target Search (MTS) [4, 5]. The rest of the paper consists of two parts: (1) the completeness proof of the MTS algorithm when heuristic function violates admissible or monotonically increasing property and (2) the convergence proof of LRTA* under admissible but inconsistent heuristics. Table 1 shows the position of this paper as well as the previous work on the formal property of LRTA* and MTS algorithms.

2 PRELIMINARIES

We begin with the description of the Moving-Target Search (MTS) problem, since it also subsumes the one addressed by LRTA* as a special case.

2.1 The MTS problem

The MTS problem involves two agents, called the *problem solver* and the *target*, both traveling in a state space of which the numbers of states and actions are finite, every state is reachable from each other, and all actions are invertible (i.e., has a counter-action) and cost uniformly one. In addition, the state space does not contain actions that do not change the location of the agents. The task of the problem solver, for whom we are responsible, is to catch (i.e., to occupy the same state as) the target who is totally out of control of us.

The problem solver has no clue on how the target behaves. It can only observe the position of the target as well as its own, which actions are available at their present states, and the outcome of these actions. On the other hand, the target may or may not have the complete knowledge of the problem solver. Since it is hopeless to always catch such a target that moves much faster than the problem solver, it is assumed that the problem solver has a speed advantage over the target, however small it might be.

As usual with the real-time search model, the problem solver is required to be *reactive*, in the sense that there is a limit on the time and resources allowed before deciding which action to take in each stage; In this paper, this constraint will be modeled by restricting the problem solver to the look-ahead of depth one.

[1] Department of Computer and Information Sciences, Ibaraki University, Hitachi, 316-8511, Japan, email: shimbo@cis.ibaraki.ac.jp.
[2] Department of Social Informatics, Kyoto University, Kyoto, 606-8501, Japan, email: ishida@i.kyoto-u.ac.jp.

Table 1. Known results on the effect of heuristic functions upon the real-time search methods

	Heuristics		Algorithms	
	Admissible	Consistent/Nondecreasing	LRTA*	MTS
Completeness	Yes	Yes	[5]	[5], this paper
	Yes	No	Straightforward application of [5]	This paper
	No	Yes	Straightforward application of [5]	This paper
	No	No	Straightforward application of [5]	This paper
Convergence	Yes	Yes	[9, 1], this paper	Straightforward
	Yes	No	[1], this paper	–
	No	Yes	[6]*	–
	No	No	–	–

* Not in the normal sense of convergence, but a (non-trivial) upper bound of the solution after infinite trials is derived.

2.2 State space and heuristics

Now we formulate the MTS problem formally. Let $\mathcal{N}$ and $\mathcal{R}$ be the sets of natural and real numbers, respectively. The state space is formulated by a graph (X, A), which is finite, undirected, simple, and connected. Here, X is the finite set of nodes, or *states*, and reflexive relation $A \subset X^2$ is the edges between states. Since the graph is simple, we identify the set of *actions* with the set A of edges; the state pair $(x, y) \in A$ means there is an action available at state x that makes the agent move to y (we hereby call such action as action (x, y)). It follows that since there is no action that leaves the agent at the same state, there is no loop (i.e., cycle of length one) contained in the graph; i.e., $(x, x) \notin A$ for any $x \in X$.

In such a state space, it is well known that the optimal cost function $h^\star : X^2 \mapsto \mathcal{R}$ is definable, where $h^\star(x, y)$ denotes the shortest distance between states $x, y \in X$. The problem solver is assumed to have at hand an inexpensive mechanism to compute an estimate $h(x, y)$ of $h^\star(x, y)$ for each state pair (x, y). The estimated values are called *heuristic estimates*, or simply *heuristics*. If no information is available as to the optimal costs, we can use the trivial heuristic function that gives $h(x, y) = 0$ for all $(x, y) \in X^2$. When $x, y \in X$ and $w = (x, y) \in X^2$ is a pair of states, we use $h(w)$ and $h^\star(w)$ as the shorthands for $h(x, y)$ and $h^\star(x, y)$, respectively.

Admissibility is a property of a heuristic function that it never overestimates the optimal cost. Ishida and Korf's MTS algorithm is known to catch the target when the initial heuristic function enjoys this property. In the rest of this section and the next, however, we do not require the heuristic function be admissible. To measure the quality of such (possibly overestimated) heuristics, the notion of admissibility is extended to allow a fixed amount of error.

Definition 1 The heuristic estimate $h(x, y)$ of state pair $(x, y) \in X^2$ is said to be e-admissible iff it does not overestimate the optimal cost $h^\star(x, y)$ by an amount of e; i.e., $h(x, y) \leq h^\star(x, y) + e$. Heuristic function that assigns e-admissible value to every state is called e-admissible heuristic function. In particular, 0-admissible heuristic function is simply called *admissible*.

A *consistent* heuristic function is the one that satisfies a form of triangle inequality.

Definition 2 A heuristic function is *consistent* (or *monotonic*) iff for any pair of adjacent states $(x, y) \in A$ and any state $z \in X$, $h(x, z) \leq h(y, z) + 1$ and $h(z, x) \leq h(z, y) + 1$ hold.

It is well-known that consistent heuristics are (0-)admissible. They also imply monotonic increase of the estimates in LRTA*.

2.3 Modeling the speed of agents

The speed advantage of the problem solver is modeled as follows. we assume a problem solver's move and a target's move are performed *alternately* in principle, except that the target occasionally skips (or fails to make) its move. The problem solver, on the other hand, never fails to make a move.

Formally, we introduce the set of fictitious discrete time instants identified with $\mathcal{N}$, and partition it into two disjoint sets $\mathcal{N}_T$ of odd numbers and $\mathcal{N}_P$ of even numbers. $\mathcal{N}_T$ corresponds to the set of times at which the target is allowed to move, and $\mathcal{N}_P$ the times when the problem solver is allowed to move. As a result, we only take into account the situation where the target moves first. This is not a restriction, since the case where the problem solver moves first can be identified with the case where the target skips its first move.

To formulate the target's skipping its move, we further introduce the set $\mathrm{Skip} \subset \mathcal{N}_T$ of times at which the target could have moved but actually would not (or would fail to do so). In this case, both the problem solver and the target remain at their present states, and no extra deliberation is allowed for the problem solver.

Let $\mathrm{Skip}(t)$ be the set of times before and including time t, at which the target is skipping its move. Formally, $\mathrm{Skip}(t) = \{s \in \mathrm{Skip} \mid s \leq t\}$. This implies $\mathrm{Skip}(2n) = \mathrm{Skip}(2n - 1)$ for all $n \in \mathcal{N}$ and if we let $\mathcal{I}_{\mathrm{Skip}}$ be the indicator function of Skip, i.e.,

$$\mathcal{I}_{\mathrm{Skip}}(t) = \begin{cases} 1, & \text{if } t \in \mathrm{Skip}, \\ 0, & \text{otherwise;} \end{cases} \tag{1}$$

then, observe that

$$|\mathrm{Skip}(t)| = \sum_{s=1}^{t} \mathcal{I}_{\mathrm{Skip}}(s). \tag{2}$$

To guarantee that the problem solver retains its speed advantage, we make the assumption that the target skips its move forever. It follows that Skip is an infinite set, and

$$\lim_{t \to \infty} |\mathrm{Skip}(t)| = \infty. \tag{3}$$

3 COMPLETENESS OF MTS WITHOUT NONDECREASING PROPERTY

3.1 The MTS algorithm

Ishida and Korf proposed the MTS algorithm as a solution to the above mentioned MTS problem. Figure 1 depicts the algorithm, but

with modified value-update formulas that do not force the heuristic estimates to be nondecreasing. This is because we are interested in the heuristic function violating admissibility and/or nondecreasing properties (See Remark 1 for the original value-update formulas). The variables x and y are used to maintain the locations of the problem solver and the target, respectively, and t, the number of iterations of Step 4, corresponds to the fictitious time introduced previously.

1. For each $u, v \in X$, initialize $h(u, v)$ with an initial heuristic function h_0; i.e., $h(u, v) \leftarrow h_0(u, v)$.
2. Observe the initial location $x = x_0$ of the problem solver.
3. Observe the initial location $y = y_0$ of the target.
4. Repeat for $t = 1, 2, \ldots$, until $x = y$,

   (a) If it is problem solver's turn to move, i.e., $t \in \mathcal{N}_p$, then update $h(x, y)$ by

   $$h(x, y) \leftarrow \min_{(x,z)\in A} h(z, y) + 1, \tag{4}$$

   and move to an adjacent state x' s.t.

   $$x' \in \operatorname*{argmin}_{(x,z)\in A} h(z, y), \tag{5}$$

   i.e., set $x \leftarrow x'$.

   (b) Otherwise, if the target has successfully moved from y to another state y', i.e., $t \in \mathcal{N}_P - \mathrm{Skip}$, update $h(x, y)$ by

   $$h(x, y) \leftarrow h(x, y') - 1. \tag{6}$$

   Set $y \leftarrow y'$.

Figure 1. The MTS algorithm

Remark 1 The algorithm of Figure 1 contains the value-update formulas different from the original version by Ishida and Korf. Their original formulas are

$$h(x, y) \leftarrow \max\left\{h(x, y), \min_{(x,z)\in A} h(z, y) + 1\right\}$$

when the problem solver moves, and

$$h(x, y) \leftarrow \max\left\{h(x, y), h(x, y') - 1\right\}$$

when the target moves, instead of (4) and (6), respectively. These original formulas force the heuristic estimates to be nondecreasing. The following argument is equally valid with minor modification when the original formulas are used.

Throughout the subsequent analysis, by *time* t, we mean the moment of time at the end of t-th iteration (or the beginning of $(t+1)$-st iteration) of Step 4. Furthermore, we use x_t and y_t respectively to denote the locations of the problem solver and the target at time t, and $h_t(x, y)$ to denote the heuristic estimate $h(x, y)$ at the same instant[3]. Combining the above notations with (4) and (5), we obtain

$$h_t(x_{t-1}, y_{t-1}) = h_t(x_t, y_t) + 1 = h_{t-1}(x_t, y_t) + 1 \tag{7}$$

when the problem solver moves at t-th iteration. Similarly, when the target successfully moves at t-th iteration, we have from (6),

$$h_t(x_{t-1}, y_{t-1}) = h_t(x_t, y_t) - 1 = h_{t-1}(x_t, y_t) - 1. \tag{8}$$

In either case, $h_t(x, y) = h_{t-1}(x, y)$ holds for any $(x, y) \neq (x_{t-1}, y_{t-1}) \in X^2$.

3.2 Completeness proof

A heuristic search algorithm is said to be *complete* if the problem solver never fails to arrive at a goal state. In the MTS setting, since the target changes its location during the course of the search, an algorithm is complete if the problem solver eventually catch the target.

We will establish the completeness of MTS through a series of lemmas. First, we show that updates of heuristic estimates preserve e-admissibility. Lemma 1 is the main achievement of this section, which generalizes the result of the original MTS paper to the case of inadmissible heuristics.

Lemma 1 *For any constant $e \geq 0$, if the initial heuristic estimate is e-admissible for every pair of states, then updates of MTS preserve the same property.*

Proof The proof is by induction on time $t \in \mathcal{N} \cup \{0\}$. The base case when $t = 0$ is obvious. Assume that at the beginning of t-th iteration ($t \geq 1$), we have $h_{t-1}(x, y) \leq h^\star(x, y) + e$ for each pair (x, y) of states. We will show that even after the t-th iteration, $h_t(x, y) \leq h^\star(x, y) + e$ holds for every (x, y).

Suppose t-th iteration belongs to problem solver's turn, i.e., $t \in \mathcal{N}_P$, and the problem solver has moved from state x_{t-1} to state x_t, while the target is at state $y = y_{t-1} = y_t$. Using (7), we have $h_t(x_{t-1}, y) = h_t(x_t, y) + 1 = \min_{(x_{t-1},z)\in A} h_t(z, y) + 1 \leq \min_{(x,z)\in A} [h^\star(z, y) + e] + 1 = h^\star(x_{t-1}, y) + e$.

Suppose on the other hand that t-th iteration belongs target's turn. It is trivial when $t \in \mathrm{Skip}$, since no update takes place. So further suppose $t \notin \mathrm{Skip}$ and the target has moved from state y_{t-1} to state y_t, while the problem solver remains at state $x = x_{t-1} = x_t$. Since y_{t-1} and y_t are only one unit apart, $h^\star(x, y_t) \leq h^\star(x, y_{t-1}) + 1$. It follows from (8) that $h_t(x, y_{t-1}) = h_{t-1}(x, y_t) - 1 \leq h^\star(x, y_t) + e - 1 \leq h^\star(x, y_{t-1}) + e$. □

Now let us define the quantity[4] $H_t = \sum_{w\in X^2} h_t(w) - h_t(w_t)$. Since there is an upper bound of the heuristic estimates, we have an upper bound of the quantity H_t as well.

Corollary 1 *If the initial heuristic function h_0 is bounded (but* not *necessarily admissible), $\{H_t \mid t \in \mathcal{N} \cup \{0\}\}$ is upper bounded.*

The next lemma states that the amount of speed advantage of the problem solver accumulates upon the quantity H_t.

Lemma 2 *The following relation holds for all $t \in \mathcal{N} \cup \{0\}$.*

$$|\mathrm{Skip}(t)| = \begin{cases} H_t - H_0 + 1, & \text{if } t \in \mathcal{N}_T; \\ H_t - H_0, & \text{otherwise.} \end{cases} \tag{9}$$

[3] This notation is compatible with the one we have already introduced, viz. the initial states x_0 and y_0 and the initial heuristic function h_0.

[4] This definition is reminiscent of the *heuristic disparity* of the original MTS paper [5], and H_t essentially plays the same role in our restated proof.

Proof The proof is again by induction on t. It is trivial when $t = 0$. If $t \in \mathcal{N}_T$ (target's move), and if $t \in \mathrm{Skip}$, we have $H_t = H_{t-1}$ since update does not take place. Otherwise, if $t \in \mathcal{N}_T$ (target's move), but $t \notin \mathrm{Skip}$, then $\sum_{w \in X^2} h_t(w) = \sum_{w \in X^2} h_{t-1}(w) - h_{t-1}(w_{t-1}) + h_t(w_{t-1}) = \sum_{w \in X^2} h_{t-1}(w) - h_{t-1}(w_{t-1}) + h_t(w_t) - 1$, or, equivalently, $H_t = H_{t-1} - 1$, where $w_{t-1} = (x_{t-1}, y_{t-1})$ and $w_t = (x_t, y_t)$. Using the indicator function $\mathcal{I}_{\mathrm{Skip}}$ of (1), we can combine the above equations for $t \in \mathcal{N}_T$ into a single formula

$$H_t = H_{t-1} - 1 + \mathcal{I}_{\mathrm{Skip}}(t). \tag{10}$$

On the other hand, if $t \in \mathcal{N}_P$ (problem solver's move), by similar argument, we have

$$H_t = H_{t-1} + 1. \tag{11}$$

Summing equations (10) and (11) over $t = 1, 2, \ldots$, we have

$$H_t = \begin{cases} H_0 - 1 + \sum_{s=1}^{t} \mathcal{I}_{\mathrm{Skip}}(s), & \text{if } t \in \mathcal{N}_T; \\ H_0 + \sum_{s=1}^{t} \mathcal{I}_{\mathrm{Skip}}(s), & \text{otherwise.} \end{cases}$$

The statement of the lemma follows from (2). □

Finally, the following theorem extends Ishida and Korf's completeness theorem to the case of inadmissible heuristics.

Theorem 1 *MTS is complete, if the initial heuristic estimate is bounded (but not necessarily admissible) for every pair of states.*

Proof Assume conversely that the problem solver cannot catch the target forever. Then, by equations (3) and (9), we have $\lim_{t \to \infty} H_t = \infty$, but this contradicts Corollary 1, or the boundedness of H_t. □

4 CONVERGENCE OF LRTA* WITH INCONSISTENT HEURISTICS

In this section, we show that the convergence of LRTA* can be developed by extending the lemmas used for proving the completeness. Although several other techniques for proving the convergence are known, our technique leads to much more succinct proof. Unlike the previous section, we assume the initial heuristic estimates are given by some (0-)admissible (but not necessarily consistent) heuristic function h_0.

4.1 The LRTA* algorithm

The LRTA* algorithm can be considered as a special case of MTS algorithm in which the target skips all its moves, i.e., $\mathrm{Skip} = \mathcal{N}_T$. In other words, it is obtained by fixing the target's location $y = y_0$ in Figure 1 throughout the problem solving and thereby ignoring Step 4b. Hence, in the following, the domain of function h corrupts to X and we use the shorthand notation $h(x)$ to denote $h(x, y)$, as well as $\tau = t/2$ as the iteration counter. The LRTA* algorithm thus obtained through the reduction from MTS is depicted in Figure 2.

4.2 Lemmas utilized

Our convergence proof is built upon the lemmas used for the proof of completeness. This is the novelty of our proof, since all the previously known proof for convergence relied on the techniques totally different from that of completeness. We start with an important lemma by Korf. It can also be regarded as a special case of Lemma 1 when $e = 0$ and the target never makes a move.

1. For each $x \in X$, initialize $h(x)$ with an admissible initial heuristic function h_0; i.e., $h(x) \leftarrow h_0(x)$.
2. Set the problem solver's location x to be x_0; $x \leftarrow x_0$.
3. Repeat the following steps for $\tau = 1, 2, \ldots$, until $x = y$,
   (a) Update the heuristic estimate $h(x)$ of the problem solver's current state x as follows.
   $$h(x) \leftarrow \min_{(x,z) \in A} h(z) + 1. \tag{12}$$
   (b) Move to a successor z of the current state x such that
   $$z \in \operatorname*{argmin}_{(x,z) \in A} h(z) + 1.$$
   If there are more than one such state, choose among them arbitrarily (tie break).

Figure 2. The LRTA* algorithm

Lemma 3 ([9]) *Updates by formula (12) preserve the admissibility of heuristic function. That is, if the initial heuristic function h_0 satisfies $h_0(x) \leq h^*(x)$ for all $x \in X$, then for every $\tau = 1, 2, \ldots$, and for each $x \in X$, $h_t(x) \leq h^*(x)$ still holds.*

Since the target (goal) is stationary in this case, every move by the target is skipped; hence we have $|\mathrm{Skip}(2\tau)| = \tau$ and $|\mathrm{Skip}(2\tau + 1)| = \tau + 1$ and $H_{2\tau} = H_{2\tau+1}$ for all $\tau \geq 0$. Then, (9) corrupts to a single equality

$$\tau = K_\tau - K_0. \tag{13}$$

where $K_\tau = H_{2\tau}$. This is the lemma also used by [7], but only for the completeness of LRTA*.

Lemma 4 ([4, 5, 7]) *For each time instant $\tau = 0, 1, 2, \ldots$, relation (13) holds.*

4.3 Convergence proof

The results in the previous sections show that LRTA*, as well as MTS, arrives at a goal after making a finite number of moves. A *problem solving trial*, or simply a *trial*, is the event beginning with the problem solver's departing from the initial state and ending at its arrival at the goal. This section considers the *repeated* trials of LRTA* algorithm on the same problem; once the problem solver reaches a goal, it is reset to the initial state, and the search continues. The heuristic estimates altered in the last trial are inherited as the initial heuristics of the present trial. From Lemma 3, we know that the admissibility of heuristics is always maintained, and hence the prerequisite for the completeness of LRTA* is also maintained during such repeated trials. It follows that such problem solving trials can be repeated forever.

For each trial $i = 1, 2, \ldots$, let $c(i)$ be the number of moves the problem solver has made in the i-th trial. We denote by x_τ^i the state at which the problem solver has arrived by the τ-th move in the i-th trial (for $0 \leq \tau \leq c(i)$), and for any $x \in X$ denote by $h_\tau^i(x)$ the

heuristic estimate $h(x)$ at the same moment. The fact that the heuristic estimates on arriving at a goal is reused as the initial heuristics for the subsequent trial is stated as follows: for any trial $i = 1, 2, \ldots$, and any state $x \in X$,

$$h^i_{c(i)}(x) = h^{i+1}_0(x). \tag{14}$$

Lemma 5 (Corollary of Lemma 4) *The following equation holds for each trial $i = 1, 2, \ldots$:*

$$c(i) = \sum_{x \in X} h^i_{c(i)}(x) - \sum_{x \in X} h^i_0(x) + h^i_0(x^i_0). \tag{15}$$

Lemma 6 *For each trial $n = 1, 2, \ldots$, equation (16) holds.*

$$\sum_{i=1}^{n} c(i) = \sum_{x \in X} h^n_{c(n)}(x) - \sum_{x \in X} h^1_0(x) + \sum_{i=1}^{n} h^i_0(x^i_0). \tag{16}$$

Proof By summing (15) over $i = 1, 2, \ldots, n$ and using (14). □

Theorem 2 *There is a trial such that all the paths traversed by the problem solver in the subsequent trials are optimal.*

Proof From (16), we have

$$\begin{aligned}
&\sum_{i=1}^{n} \left[c(i) - h^\star(x^i_0)\right] \\
&= \sum_{x \in X} h^n_{c(n)}(x) - \sum_{x \in X} h^1_0(x) + \sum_{i=1}^{n} h^i_0(x^i_0) - \sum_{i=1}^{n} h^\star(x^i_0) \\
&\leq \sum_{x \in X} h^\star(x) - \sum_{x \in X} h^1_0(x), \qquad (17)
\end{aligned}$$

where Lemma 3 is used to derive the inequality. In other words, the series $\sum_i \left[c(i) - h^\star(x^i_0)\right]$ of positive terms has an upper bound, and hence converges. Thus we have $\lim_{n\to\infty} [c(n) - h^\star(x^n_0)] = 0$. □

Once we know there exists trial m such that all the paths traversed in the subsequent trials are optimal, it is easy to establish the convergence of the heuristic estimates to optimal values on every state along the paths traversed after trial m, by the similar argument as the original convergence proof by Korf. Note that our proof is valid regardless of the consistency of the heuristic function, unlike the original one which implicitly assumed consistency.

5 RELATED WORK

Moving-Target Search Our motivation of studying the MTS algorithm with inadmissible heuristics arose out of the difference between the value-update rules used by LRTA* and MTS. In addition, it was believed that MTS was incomplete when heuristics overestimate. This contrasts with LRTA*, which was already known to be complete even with inadmissible heuristics.

LRTA* If we assume an initially consistent heuristic function, then it will guarantee the monotonic increase of the estimates as well as the preservation of the consistency among them. As the original convergence proof of LRTA* [9] takes advantage of this property, it is not generally valid when it is inconsistent.

Barto *et al.* [1] proposed trial-based RTDP, which is a generalization of LRTA* to the stochastic state space. It is also applicable to deterministic state space, in which case the algorithm reduces to LRTA*. Their proof of convergence is via reduction to asynchronous dynamic programming [2], which is equally applicable to inconsistent heuristics.

Compared with Barto et al.'s proof technique, our proof lacks the generality to cover the case of stochastic state space. However, it is much more succinct, as it fully exploits the nature of LRTA*. In addition, it does not require the knowledge of asynchronous DP, a theory that is powerful but relatively complex.

6 CONCLUSION

The contribution of the paper is as follows: (1) The completeness of the MTS algorithm even with inadmissible or decreasing heuristics. This extends the class of heuristic functions that are usable with MTS and thereby the application domain of it, as it is often difficult to construct admissible heuristic functions in many domains. (2) A new proof of convergence of LRTA* algorithm under inconsistent heuristics. The resulting proof is extremely succinct, and, unlike the previous proof techniques, is built on the same lemmas as the completeness proof. This sharing of lemmas also clarifies that these two properties are actually closely related to one another.

Yet, as the blank fields in Table 1 show, there remain several open problems concerning the properties of heuristic function and their impact on real-time search algorithms. The most challenging should be to prove the convergence of LRTA* with inadmissible heuristic function without the guarantee of nondecreasing property.

Although this paper only presented the technical results due to space limitation, we also plan to present the experimental results elsewhere.

REFERENCES

[1] Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh, 'Learning to act using real-time dynamic programming', *Artificial Intelligence*, **72**(1–2), 81–138, (1995).

[2] Dimitri P. Bertsekas and John N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[3] Toru Ishida, *Real-Time Search for Learning Autonomous Agents*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1997.

[4] Toru Ishida and Richard E. Korf, 'Moving-target search', in *Proceedings of the Twelfth International Conference on Artificial Intelligence (IJCAI-91)*, pp. 204–210, Sydney, Australia, (1991). Morgan Kaufmann.

[5] Toru Ishida and Richard E. Korf, 'Moving-target search: a real-time search for changing goals', *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, **17**(6), 609–619, (June 1995).

[6] Toru Ishida and Masashi Shimbo, 'Improving the learning efficiencies of real-time search', in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, volume 1, pp. 305–310, Portland, OR, USA, (1996). AAAI Press.

[7] Sven Koenig, 'The complexity of real-time search', Technical Report CMU-CS-92-145, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, (April 1992).

[8] Sven Koenig, 'Exploring unknown environments with real-time search or reinforcement learning', in *Neural Information Processing Systems 10 (NIPS* 98)*, (1998).

[9] Richard E. Korf, 'Real-time heuristic search', *Artificial Intelligence*, **42**(2–3), 189–211, (1990).

[10] Andrew W. Moore and Christopher G. Atkeson, 'Prioritized sweeping: reinforcement learning with less data and less time', *Machine Learning*, **13**, 103–130, (1993).