Journal of Heuristics manuscript No.
(will be inserted by the editor)

Hardness Measures for Gridworld Benchmarks and Performance Analysis of Real-Time Heuristic Search Algorithms

Masataka MIZUSAWA, Masahito KURIHARA

Graduate School of Information Science and Technology, Hokkaido University, JAPAN
e-mail: `mizu@main.ist.hokudai.ac.jp`
e-mail: `kurihara@ist.hokudai.ac.jp`

Received: date / Revised version: date

Abstract Gridworlds are one of the most popular settings used in benchmark problems for real-time heuristic search algorithms. However, no comprehensive studies have existed so far on how the difference in the density of randomly positioned obstacles affects the hardness of the problems. This paper presents two measures for characterizing the hardness of gridworld problems parameterized by obstacle ratio, and relates them to the performance of the algorithms. We empirically show that the peak locations of those measures and actual performance degradation of the basic algorithms (RTA* and LRTA*) almost coincide with each other for a wide variety of problem settings. Thus the measures uncover some interesting aspects of the gridworlds.

Key words real-time search – gridworlds – benchmark – phase transition

Introduction

Real-time (or *on-line*) *heuristic search* is an attractive framework for real-world-oriented agents. Traditional *off-line search* makes a complete plan before action execution, often resulting in exponential time complexity which limits its applicability to real-world problems. On the other hand, real-time search interleaves partial look-ahead and action execution, often resulting in a practical model to cope with real-world problems.

Two-dimensional grids with randomly positioned obstacles (*gridworlds*) are one of the most popular settings used in benchmark problems for real-time search algorithms. Their usefulness lies in their simplicity of problem description and easy visualization of search processes. However, no comprehensive studies have existed so far on how the difference in the density of randomly positioned obstacles affects the structure of the state spaces and the performance of the algorithms. In particular, recent studies of the so-called phase transition phenomena which could cause dramatic change in their performance in a relatively small parameter range suggest that we should evaluate the performance in a parametric way with the parameter range wide enough to cover potential transition areas.

In this paper, we present two measures for characterizing the hardness of the gridworld problems parameterized by the obstacle ratio, and relate them to the performance of real-time search algorithms. One is a measure based on the entropy calculated from the probability of existence of solutions. The other is a measure based on the total errors of initial heuristic cost estimation against the actual cost. We show that the gridworlds are the most complicated in both measures when the obstacle ratio is around 41%. We then solve the parameterized gridworlds with the well-known basic real-time search algorithms RTA* and LRTA* to relate their performance to the proposed measures. Evaluating the number of steps required for obtaining solutions with the two algorithms, we show that they both have a peak when the obstacle ratio is around 41%. This coincidence supports the relevance of the proposed measures. We also show that this kind of coincidence can be observed for a wide variety of heuristic functions and gridworld types. Using wide-range settings of the obstacle ratio, this paper provides guidelines to set up problems appropriately as the benchmarks, and reveals some interesting aspects of the gridworlds and the algorithms.

The rest of this paper is organized as follows. Section 1 reviews typical real-time search algorithms and their properties, and then gives a description of the gridworld problems. In Section 2, we introduce two kinds of measures for evaluating the hardness of the gridworlds parameterized by the obstacle ratio, and relate them to the performance of the algorithms. Section 3 demonstrates that those measures are applicable to the problems with wide variety of settings. In Section 4, we compare our work with related works, and discuss the novelty of our studies. Finally, we conclude the paper in Section 5.

1 Real-Time Search Algorithms and Gridworld Problems

1.1 State Space Search Problem and Real-Time Search

A *state space search problem* is represented by a tuple $< N, s, G, O >$, where N denotes a set of *states* (of the problem solver or agent), including a *start state* $s \in N$ and a set of *goal states* $G \subseteq N$, and $O \subseteq N \times N$ denotes a set of *operators* which represent transition of the states. The pair $< N, O >$ defines a directed graph called a *state space graph*. A *solution* of the problem is a path from the start state to a goal state. The cost of the operator $o = (v, v') \in O$ for changing the state from v to v' is denoted by $c(v, v')$ (> 0), and the sum of the cost of those operators on a solution path defines the cost of the solution. A solution with the smallest cost is an *optimal* solution. For improving the efficiency, the heuristic search algorithms exploit the so-called *heuristic function* $h(v)$ for estimating $h^*(v)$, the cost of optimal solutions starting from state v. The h-values are *admissible* if and only if $0 \leq h(v) \leq h^*(v)$ for all states v.

The search algorithms ever proposed can be classified into two types, off-line search and real-time search. The off-line search algorithms, including classical ones such as breadth-first, depth-first, and A*, only concern making a complete plan to a goal before it is actually executed by the agent. On the other hand, the real-time search algorithms concern interleaved decision making and action execution for navigating the reactive agent to a goal. They repeat a cycle in which, based on a local search, they decide a single action in a constant time and execute it immediately.

The real-time search algorithms have the following advantages.

- They can take care of the environments that change dynamically during the problem solving process. In such environments, the complete plans made by the off-line search algorithms before action execution often become inappropriate in the course of the process.
- They need no whole map of the state space. Using only local map information around the agents, the real-time search algorithms can work under the situation where the area observed by the agents spreads out as they move.
- They are more suitable for problem solving in the real world where the computational time and memory capacity are limited, and provide more human-like framework of problem solving than that of the off-line search algorithms.

1.2 RTA and LRTA* Algorithms*

Korf is the first to present the basic framework of the real-time heuristic search (Korf, 1990). He proposed Real-Time A* (RTA*) and Learning Real-Time A* (LRTA*) algorithms.

RTA* repeats the following steps until a goal is reached.

1. Compute $f(v') = c(v, v') + h(v')$ for each neighbor v' of the current state v.
2. Update the h-value of the state v by $h(v) = secondmin_{v'} f(v')$, where *secondmin* denotes the function that returns the second smallest value.
3. Move to a neighbor v' with the smallest $f(v')$ value. Ties are broken randomly.

In step 2, updating $h(v)$ to the second smallest value of $f(v')$ prevents the agent from unnecessarily visiting the same state.

RTA* has the properties of *completeness* and *correctness* (Korf, 1990).

LRTA* is the same as RTA* except that the step 2 is replaced by the following.

2. Update the h-value of the state v by $h(v) = min_{v'} f(v')$.

Whereas RTA* updates $h(v)$ to the second smallest value of $f(v')$, LRTA* updates it to the smallest one. In this way, LRTA* never overestimates h-values. As a result, those h-values will gradually approach the accurate estimation if their initial values are admissible.

LRTA* has the property of *convergence*, in addition to *completeness* (Korf, 1990). The proofs of these properties are precisely described in (Ishida and Korf, 1995; Shimbo and Ishida, 2000).

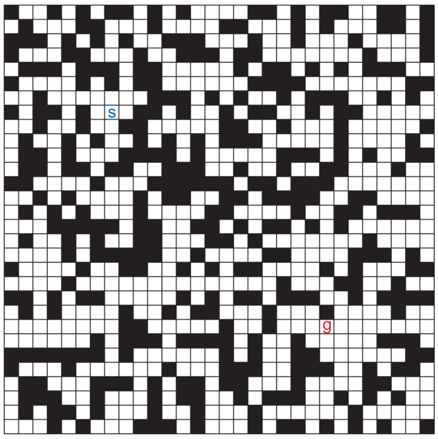

Fig. 1 An example of gridworlds (30 × 30, 40% obstacles).

1.3 Gridworld Problems

Gridworlds are search problems for finding a path from a start state s to a goal state g in a 2-dimensional grid environment with randomly positioned obstacles. Fig. 1 shows an example of gridworlds of size 30×30.

The problem setting in this paper is as follows: The state space consists of the $m \times n$ cells (positions) in the grid. The agents are allowed to move to a vertically- or horizontally-adjacent cell in a single step, unless the cell is not occupied by an obstacle. Such a move defines a state transition. Moreover, we consider the grid as a torus. This means that when the agent moves out of a bound, it just comes inside from another bound at an appropriate position (as formally described later). The start and the goal are placed at one of the most separate pair of positions of the grid. The cost of each move is a constant, say, 1.

Formally, a gridworld is defined by a tuple $< m, n, obs >$, where m and n are integers that define the set $C_{m,n}$ of cell positions $C_{m,n} = \{(x, y) \mid 0 \leq x < m, 0 \leq y < n\}$, and obs is a Boolean function such that $obs(x, y)$ is true if and only if the cell at (x, y) is occupied by an obstacle. In this paper, we consider a class of gridworlds generated by a random mechanism to define obs in terms of the obstacle ratio $r = R/mn$ as a parameter, where R is the number of cells occupied by obstacles. Note that we can think of mainly two ways to place obstacles: place an obstacle with probability r at each cell, or, place exactly rmn obstacles at random. We want to make many problem instances which have solutions for large r, but such problem instances can be hardly generated by the former one. Therefore, we adopt the latter way in this paper.

Given a gridworld $< m, n, obs >$, the associated state space search problem is uniquely defined by the tuple $< V, s, g, T >$, where $V = C_{m,n} - \{(x, y) | obs(x, y)\}$ is the set of states, s is the start state, $s = (\lfloor m/4 \rfloor, \lfloor n/4 \rfloor)$, g is the goal state, $g = (\lfloor 3m/4 \rfloor, \lfloor 3n/4 \rfloor)$, and $T \subseteq V \times V$ is the set of operators defined by

$((x, y), (x', y')) \in T$ if and only if either (1) $x' = x$ and $y' = y \pm 1$ (mod n), or (2) $x' = x \pm 1$ (mod m) and $y' = y$.

Such a state space search problem $< V, s, g, T >$ is associated with a directed graph $D = (V, L)$, defined by the set of nodes V and the set of edges L which connect each node (x, y) with each of its successors (x', y') such that $((x, y), (x', y')) \in T$. The solutions of the problem are the paths of the graph starting from s and ending at g. Note that D is not necessarily a connected graph, because there can be a node which is not reachable from the start state s. In particular, the goal state g is not necessarily reachable from s. Since the problems are generated by a random mechanism, we can talk about the probability that a problem has solutions. This is a topic discussed in the next section.

Note that the gridworlds are relatively easy problems for off-line search algorithms, because the size of the state space is $O(mn)$ for $m \times n$ grids and the algorithms are allowed to spend enough time to find a solution using whole knowledge about the state space. On the other hand, real-time search algorithms cannot get additional knowledge about the state space without actual actions of the agents, and each action should be decided within the constant time (in *real time*). Therefore, the gridworlds have been thought to be suitable benchmark problems for real-time search algorithms and widely used to demonstrate the performance of the algorithms.

2 Hardness Measures for Gridworld Problems

Given the size m and n of the problem, there exist exponentially many gridworlds, each corresponding to an allocation pattern of the obstacles, and their hardness statistically depends on the number of obstacles. In general, the gridworlds will get more complicated as the number of obstacles increases to some extent, but placing too many obstacles would make them simpler. For a deep understanding of the properties of the problems, it is important to see how the obstacle ratio r affects the hardness to solve them.

In this section, we present two measures for the hardness of the gridworld problems, based on the notions of the *probability of existence of solutions* and the *initial heuristic error*. The measures will uncover some interesting aspects of the gridworlds from each viewpoint.

2.1 Probability of Existence of Solutions and Its Entropy

In this subsection, we introduce the hardness measure based on the probability p that a problem has a solution, depending upon the obstacle ratio r. Fig. 2 shows the plot of p for various obstacle ratios and gridworld sizes. Each data point has been computed from 10,000 randomly generated problem instances.

To interpret the results, we identify three ranges, $0 \leq r < r_1$, $r_1 \leq r < r_2$, and $r_2 \leq r \leq 1$, where r_1 and r_2 are some values around 0.35 and 0.43, respectively.

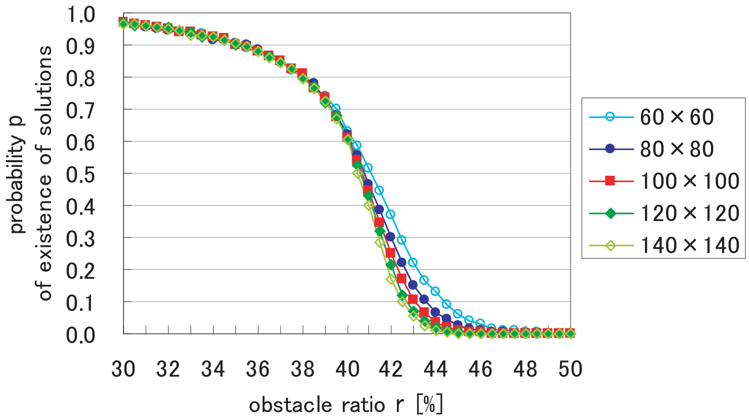

Fig. 2 Probability p of existence of solutions.

In the first range, where r is relatively small, p is decreasing from 1 but its change is modest. As we can easily imagine, in this case, those small number of obstacles rarely prevent a path from reaching the goal to form a solution.

In the third range, where r is relatively large, p gradually approaches 0. In this case, a lot of obstacles tend to be placed adjacent to each other, making long *walls* to prevent a path from going straight to the goal.

The second range, where r is between 0.35 and 0.43, is the most interesting, because p is decreasing from 0.9 to near 0.1 very rapidly. In particular, the slope of the decreasing curve is the steepest around $r = 0.41$. This kind of phenomena are sometimes called the *phase transition*, suggesting that a big change may be taking place in the structure of the state space in this relatively small parameter range.

Intuitively, the gridworlds are the most complicated and the hardest to solve when p is around 0.5, because in that region it is most difficult to predict whether solutions exist or not. Here we introduce the first hardness measure, the *entropy* H based on p, incorporating such an intuition.

$$H = -p \log_2 p - (1-p) \log_2 (1-p)$$

H is the average information content to determine whether solutions exist or not, and we interpret it as a measure of disorder of problem instances.

Fig. 3 shows the entropy H calculated from Fig. 2. H takes the maximum around $r = 41\%$, regardless of the gridworld size. According to our interpretation, the gridworlds are the most complicated in that region.

2.2 Total Initial Heuristic Errors

In this subsection, we introduce the second hardness measure, based on the initial heuristic error.

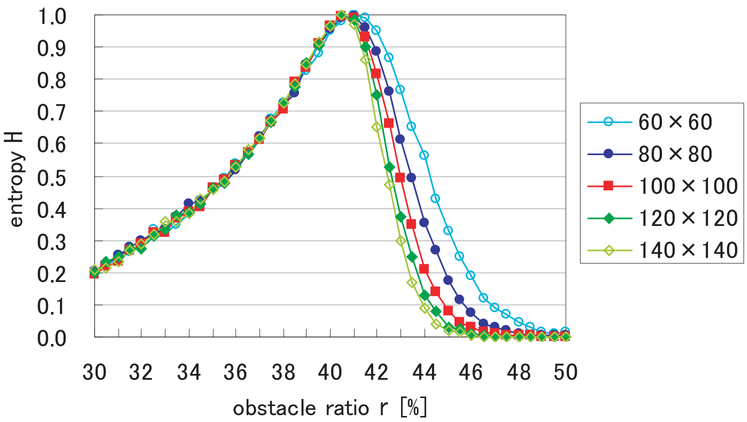

Fig. 3 Entropy H calculated from the probability p of existence of solutions.

The performance of real-time heuristic search algorithms are greatly influenced by the accuracy of h-values, because the selection of a move in those algorithms depends solely on h-values. For example, if the h-values are accurate, the agents will move toward a goal along an optimal path. Otherwise, they may choose a wrong move along a path which will never lead to the goals without backtracking. Therefore, there is a good reason to believe that the difficulty of the gridworlds for real-time search algorithms is defined appropriately in terms of the *initial heuristic error*, the error between the actual cost h^* and the estimated cost h at the beginning of the problem solving.

Here we introduce the second hardness measure, *total initial heuristic errors* E, defined by the sum of the initial heuristic errors for all states

$$E = \sum_{v \in V'} \left| h^*(v) - h^0(v) \right|$$

where $h^0(v)$ denotes the initial h-value for state v, and V' the set of all the states on some path connecting the start and the goal. Note that the states unreachable from the start and the goal are ruled out. As a result, the size of V' is variable, even if the number of obstacles is fixed. Therefore, E depends on the size of V' as well as the accuracy of initial h-values. When computing E and the performance of the algorithms, we employ only the problem instances which have solutions, because they are computable only for such problems.

Fig. 4 shows how E is related to the obstacle ratio r, when we adopt the Manhattan distance $Manhattan(v, g) = \min(|x-x'|, m-|x-x'|)+\min(|y-y'|, n-|y-y'|)$ for a heuristic function $h^0(v)$, where $v = (x, y)$ and the goal $g = (x', y')$. Note that this definition of Manhattan distance is adapted from the ordinary one for our torus gridworlds. For each point (r, E), E is obtained as the mean of 10,000 randomly generated problem instances of obstacle ratio r. The relative standard errors smaller than 1% have been ensured by the statistical analysis.

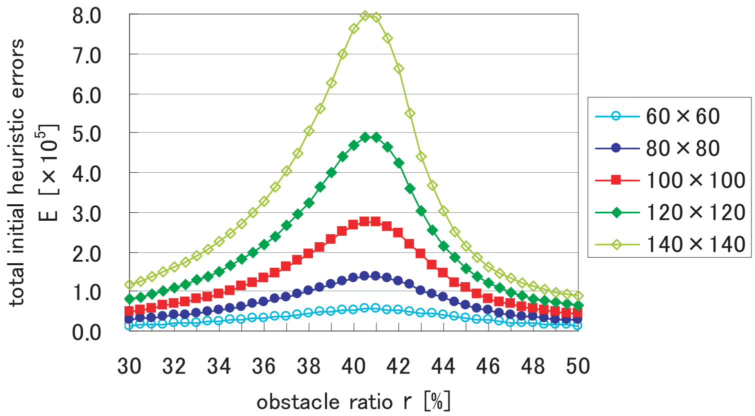

Fig. 4 Total initial heuristic errors E using Manhattan distance as initial h-values $h^0(v)$.

We can see that when r is relatively small, E is monotonically increasing as r increases, and when r exceeds some point, it starts to be monotonically decreasing. Note that our hardness measure E is maximal when r is around 41%, which is common to all of the five curves. Moreover, it is interesting to notice that our first hardness measure H, the entropy we have introduced in the previous subsection, was also maximal around $r = 41\%$. Thus, intuitively, both measures strengthen the relevance of each other. This is a good reason for us to say that the gridworlds are the most *complicated* around $r = 41\%$. In fact, in the next subsection, we will see that the amount of computation required for the real-time search algorithms also takes the maximum around $r = 41\%$.

2.3 Performance Evaluation of Real-Time Search Algorithms

In this subsection, we investigate the performance of real-time search algorithms. More precisely, we evaluate the actual cost C (equal to the number of steps in this case) required for obtaining solutions with RTA* and LRTA* on the gridworlds parameterized by the obstacle ratio r.

The experimental results are depicted in Fig. 5. Each point is obtained as the mean of 10,000 randomly generated problem instances of size 100×100, adopting Manhattan distance for a heuristic function. The relative standard errors of the data are within 2%.

The results show that in both algorithms the actual cost C takes its maximum when the obstacle ratio r is around 41%. Recall that both the entropy H and the total initial heuristic errors E have a peak when r is around 41% as we have seen in Section 2.1 and 2.2. Therefore, these parameter regions are approximately identical. This supports the validity of two kinds of hardness measures H and E.

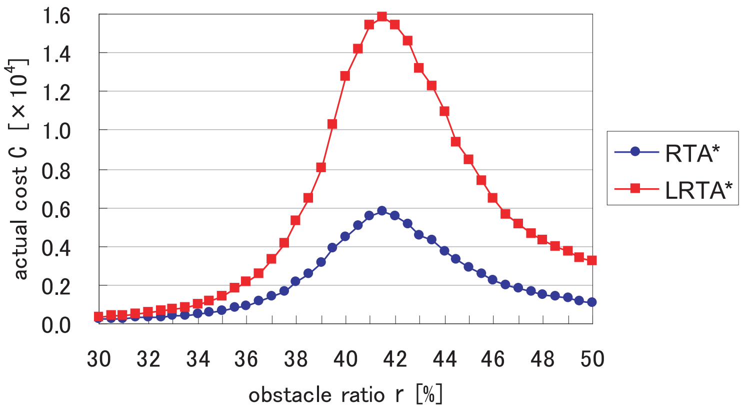

Fig. 5 The performance of RTA* and LRTA*.

Table 1 Peak locations for a variety of heuristic functions.

heuristic function		peak location H	E	C(RTA*)	C(LRTA*)
Zero	0	41%	40%	40%	41%
Minimum	$\min(x_d, y_d)$		40%	41%	42%
Maximum	$\max(x_d, y_d)$		40%	41%	41%
Multiple	$\sqrt{x_d y_d}$		40%	42%	41%
Euclidean	$\sqrt{x_d{}^2 + y_d{}^2}$		41%	41%	41%
Manhattan	$x_d + y_d$		41%	42%	41%

3 Coincidence of Peak Locations

In this section, we demonstrate that the hardness measures H and E are applicable to the problems with a variety of settings. More precisely, we will see the coincidence of the peak locations for a variety of heuristic functions, for some variants of gridworld problems, and even for search problems on random graphs.

3.1 Various Kinds of Heuristic Functions

We have seen that the peak locations of the entropy H, the total initial heuristic errors E, and the actual cost C almost coincide with each other when using Manhattan distance for a heuristic function. In this subsection, we will see that this coincidence can be observed commonly for other heuristic functions as well.

Table 1 summarizes the peak locations of H, E, and C for a variety of heuristic functions. Each heuristic function estimates the cost from a state $v = (x, y)$ to the goal state $g = (x', y')$ using the formula in the table, where $x_d = \min(|x - x'|, m - |x - x'|)$ and $y_d = \min(|y - y'|, n - |y - y'|)$.

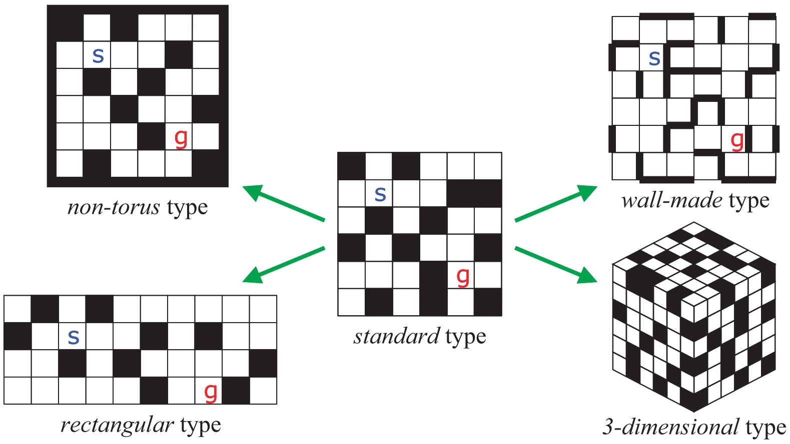

Fig. 6 Types of gridworlds.

Table 2 Peak locations for a variety types of gridworlds.

type	size	peak location H	E	C(RTA*)	C(LRTA*)
standard	100×100	41%	41%	42%	41%
non-torus	100×100	39%	39%	40%	40%
rectangular	200×50	40%	40%	40%	40%
	500×20	34%	34%	34%	35%
	1000×10	26%	26%	25%	25%
wall-made	100×100	49%	50%	51%	51%
3-dimensional	$20 \times 20 \times 20$	66%	67%	68%	68%

Note that the peak locations of H, E, and C almost coincide at around 41% for all heuristic functions we have considered. In particular, it is interesting to see that even the *Zero* case (or the *uninformed* case) where there is no prior knowledge given to the agents has the same peak location.

All these heuristic functions have two properties in common: they are *admissible* and they are *weakly decreasing* for the empty grid. A heuristic function is admissible if it never overestimates the actual cost, as defined in Section 1.1. A heuristic function is weakly decreasing for a particular state space if the estimated cost for the space is monotonically non-increasing along all the optimal paths to the goal. We conjecture that the peak locations of H, E, and C almost coincide at around 41% for *all* heuristic functions that are admissible, weakly decreasing for the empty grid, and defined only in terms of x_d and y_d; but this conjecture is yet to be verified.

3.2 Various Types of Gridworlds

We have seen that the peak locations almost coincide with each other for the 2-dimensional torus- and square-type gridworlds (we call this type of gridworlds *standard*). In this subsection, we will see that this kind of coincidence is commonly observed for other types of gridworlds as well. The gridworlds we consider are natural extension of the *standard* type, as depicted in Fig. 6. We adopt the Manhattan distance for a heuristic function (for the *3-dimentional* type, the precise definition of Manhattan distance is 3-dimentional as well).

Table 2 summarizes the peak locations for a variety types of gridworlds. The peak locations almost coincide with each other, because their difference is at most 2 (points). The actual peak locations depend on the gridworld types and sizes (ratios of m:n). From these results, we can use H or E to predict the peak location of the actual performance degradation of RTA* and LRTA* algorithms for various types of randomly generated gridworlds.

3.3 Random Graphs

We have seen that the peak locations almost coincide with each other for a variety types of gridworlds. In this subsection, we will see that this kind of coincidence is observed even for general random graphs.

Extending the *standard* type of gridworlds, we generate random graphs in 2-dimensional torus Euclidean planes (Fig. 7). More precisely, each graph consists of n nodes: the start s and the goal g placed at one of the most separate pair of positions; $n-2$ nodes placed randomly according to the uniform distribution. Pairs of nodes are connected by edges according to either of the following two models: one is the *fixed radius model* $G_F(n, R)$ which connects any pairs of nodes within a distance R; the other is the *Bernoulli model* $G_B(n, P)$ which connects each pair of nodes with a probability P. These R and P are parameters on which we are to focus.

Table 3 summarizes the peak locations for two kinds of random graphs, when we adopt the Euclidean distance for a heuristic function. We can see that the peak locations almost coincide with each other for any cases of the model and size. This suggests that H and E are suitable also for the general mazes, not only the gridworlds.

4 Related Works

Real-time heuristic search is the framework of search technique proposed by Korf (Korf, 1990). The benchmark problems used in his work were N-puzzles ($N = 8, 15, 24$). As far as we know, Ishida is the first to adopt the gridworlds as the benchmark problems for real-time search. In his joint work with Korf (Ishida and Korf, 1991), Ishida introduced new search problems called Moving-Target Search (MTS), in which a goal state (target) may move continuously during problem solving processes. Therefore, N-puzzles were inappropriate benchmark problems for

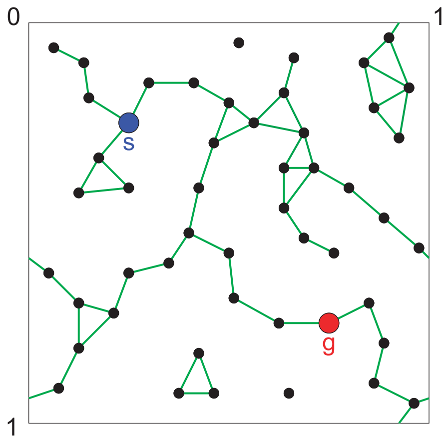

Fig. 7 An example of random graph $G_F(50, 0.13)$.

Table 3 Peak locations for random graphs.

graph model	n	peak location for R / P			
		H	E	C(RTA*)	C(LRTA*)
$G_F(n,R)$	100	0.111	0.113	0.111	0.112
	500	0.052	0.053	0.052	0.052
	1000	0.037	0.038	0.037	0.037
$G_B(n,P)$	100	0.018	0.018	0.019	0.019
	500	0.0035	0.0032	0.0035	0.0032
	1000	0.0017	0.0015	0.0018	0.0016

MTS, because in those puzzles the goal states are stationary. On the other hand, the gridworlds are appropriate problems for MTS, because it is very natural to think of applications where target agents are moving around in the gridworlds. He showed experimental results (for the *standard* type) for obstacle ratios changing from 0% to 35%, observing that the obstacles tend to disconnect the state space when the ratio reaches 40%. Probably, this is why Ishida and other researchers following his work have rarely considered the obstacle ratios greater than 40% in the literature. In this paper, working on the full-range of obstacle ratios, we have shed light on a new aspect of the gridworlds and real-time search algorithms from the unique viewpoint.

Various problems such as constraint satisfaction problems (CSPs) have an aspect that a property of problems or performance of algorithms could dramatically change in a relatively small parameter range. This phenomenon is often called *phase transition* (Cheeseman, Kanefsky, and Taylor, 1991; Hogg, Huberman, and Williams, 1996). This word has been used in the field of thermodynamics for commonly describing an abrupt sudden change (such as the change from water to ice) in physical properties with a small change in a parameter such as the temperature. Furthermore, some attempts to relate the hardness of problems to the probability of existence of solutions have been made in some CSPs such as Random-SAT and Graph-Coloring Problem. Their empirical results show that harder problem instances are often generated at around *crossover point* where the probability is 0.5 (Mitchell, Selman, and Levesque, 1992; Crawford and Auton, 1996; Hogg 1998). In this paper, we demonstrated that the similar phenomena are observed in the gridworlds as well, although the gridworlds are not CSPs.

As for random graphs, aiming at applications to wireless networks, one of the main research topics in recent years has been the analysis of their connectivity (Santi, Blough, and Vainstein, 2001; Krishnamachari et al., 2002). In this paper, for the fixed radius model and the Bernoulli model, we demonstrated that the peak locations of H, E, and C almost coincide with each other at the phase transition area where the probability of existence of solutions (a sort of connectivity) rapidly decreases. This kind of coincidence is the same as the one observed in the gridworld problems.

5 Conclusion

In this paper, we have studied the hardness of randomly generated gridworlds with the whole parameter range. We have empirically showed that the peak locations of the entropy H, the total initial heuristic errors E, and the actual cost C almost coincide with each other in all variants of gridworlds we have considered. This suggests that H and E can be used as useful hardness measures for a wide variety of gridworld-like search problems.

The gridworlds have an advantage in that it is easy to control their hardness through their sizes and obstacle ratios. However, we should not discuss the performance of the algorithms only with easy problem instances. It seems that some previous works have set obstacle ratios only to the values taken from an easy problem region. We believe that our study has made a contribution to understanding the properties of the gridworlds, and provides good tips for using them as the benchmark problems.

Future research directions include: theoretical analysis of the gridworlds and real-time search algorithms for explaining the empirical results of this paper more clearly; design of useful (possibly inadmissible) heuristic functions based on the knowledge on the obstacle ratio; application of our approach to wider variety of graphs such as *small-world networks* and *scale-free networks*. Although we have focused only on well-known basic algorithms RTA* and LRTA* as the subject of discussion with the intention of clarifying the nature of phenomena observed in the gridworlds, more sophisticated algorithms such as Multiple-Agents RTA* (Knight, 1993), Moving-Target Search (Ishida and Korf, 1991, 1995), Real-Time Bidirectional Search (Ishida, 1996), FALCONS (Furcy and Koenig, 2000), $\varepsilon\delta$-LRTA* (Shimbo and Ishida, 2003), LRTA*(k) (Hernández and Meseguer, 2005), and their variants are also interesting, and our approach may contribute to their further understanding.

References

1. Cheeseman,P., Kanefsky,B., and Taylor,W.M. (1991). "Where the *Really* Hard Problems Are", Proceedings of the 12th International Joint Conference on Artificial Intelligence, 331-337.
2. Crawford,J.M. and Auton,L.D. (1996). "Experimental Results on the Crossover Point in Random 3-SAT", Artificial Intelligence 81(2/3), 31-57.
3. Furcy,D. and Koenig,S. (2000). "Speeding up the Convergence of Real-Time Search", Proceedings of the 17th National Conference on Artificial Intelligence, 891-897.
4. Hernández,C. and Meseguer,P. (2005). "LRTA*(k)", Proceedings of the 19th International Joint Conference on Artificial Intelligence, 1238-1243.
5. Hogg,T., Huberman,B.A., and Williams,C. (1996). "Phase Transitions and the Search Problem", Artificial Intelligence 81(1), 1-15.
6. Hogg,T. (1998). "Exploiting Problem Structure as a Search Heuristic", International Journal of Modern Physics C 9, 13-29.
7. Ishida,T. and Korf,R.E. (1991). "Moving-Target Search", Proceedings of the 12th International Joint Conference on Artificial Intelligence, 204-210.
8. Ishida,T. and Korf,R.E. (1995). "Moving-Target Search: A Real-Time Search for Changing Goals", IEEE Transactions on Pattern Analysis and Machine Intelligence 17(6), 609-619.
9. Ishida,T. (1996). "Real-Time Bidirectional Search: Coordinated Problem Solving in Uncertain Situations", IEEE Transactions on Pattern Analysis and Machine Intelligence 18(6), 617-628.
10. Knight,K. (1993). "Are many reactive agents better than a few deliberative ones?", Proceedings of the 13th International Joint Conference on Artificial Intelligence, 432-437.
11. Korf,R.E. (1990). "Real-Time Heuristic Search", Artificial Intelligence 42(2/3), 189-211.
12. Krishnamachari,B., Wicker,S.B., Bejar,R., and Pearlman,M. (2002). "Critical Density Thresholds in Distributed Wireless Networks", Book chapter in Communications, Information and Network Security, Kluwer Publishers.
13. Mitchell,D., Selman,B., and Levesque,H. (1992). "Hard and Easy Distributions of SAT Problems", Proceedings of the 10th National Conference on Artificial Intelligence, 459-465.
14. Santi,P., Blough,D.M., and Vainstein,F. (2001). "A Probabilistic Analysis for the Range Assignment Problem in Ad Hoc Networks", Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing, 212-220.
15. Shimbo,M. and Ishida,T. (2000). "Towards Real-Time Search with Inadmissible Heuristics", Proceedings of the 14th European Conference on Artificial Intelligence, 609-613.
16. Shimbo,M. and Ishida,T. (2003). "Controlling the Learning Process of Real-Time Heuristic Search", Artificial Intelligence 146(1), 1-41.