4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp140-145)

C^3LRTA*, Color Code Coordinated LRTA* Algorithm

MUAZ NIAZI, UMAR MANZOOR, KIRAN IJAZ
Department of Computer Science, National University of Computer & Emerging Sciences-FAST,
FAST House, Rohtas Road,
G-9/4, Islamabad
PAKISTAN

Abstract: - In this paper we present an extension to the Learning Real-Time A* (LRTA*) algorithm by utilizing a color coded coordination scheme. The new algorithm (C^3LRTA*) has been applied to solve randomly generated mazes with multiple problem solvers. Our work suggests that by using the proposed modification, we get an improvement in LRTA*. Multiple agents coordinate their actions by using color code, a heritage from those agents, which have previous traversed the current state. We have evaluated this coordination scheme on a large number of test cases with random obstacles and varying obstacle ratio. Experimentation has shown that C^3LRTA* performs better then LRTA*. In addition, an increase in the number of agents and/or obstacle ratio, solution quality is improved as compared to LRTA*.

Key-Words: - LRTA*, A*, Online Search, Agents, Multi-agent, Coordination, Color Code, Real Time

1 Introduction

Searches have been divided into two classes by Weiss [8]; offline and online searches. Offline search algorithms execute the complete search before actually taking a step towards the target. However online search algorithms compute a plausible next move and then execute the move in constant time. Online algorithms such as LRTA* typically find solution faster than offline algorithms however solution quality may be compromised.

In case of multiagent searches, where several agents are looking for solution, another level of complexity is introduced. These agents can either solve the problem independently or else coordinate together to improve the efficiency in reaching a common goal in previously unseen environment with obstacles.

The work which is the basis of this paper, has demonstrated some exciting results. We have observed that as we increase the obstacle ratio our proposed algorithm solution quality gets better and better as compared to LRTA*. In proposed solution agents try to find alternate moves which leads to better search space exploration.

Concerning the structure of the paper, related work will be discussed in section 2. Section 3 contains our implementation. Section 4 draw a line between LRTA* and C^3LRTA* scheme. In section 5 C^3LRTA* in randomly generated mazes will be described. Section 6 critically analyze the performance of C^3LRTA* in the randomly generated mazes with original LRTA*. Section 7, 8 states our future work and conclusion respectively.

2 Related Work

Some of the Coordination mechanisms for Real-Time searches have already been proposed and we will discuss few of them. Two organizational strategies based on repulsion and attraction, to coordinate agents' move has been proposed in [2].

A new coordination strategy based on marking agents having visual depth; where agents mark the directions in which they move to inform other agents about their experiences [7].

Gordon and Matley introduce a sparse direction map using genetic algorithm to find the path in mazes. Maze is divided into sectors each of which contains a direction indicator in [15].

[4] "Developed a cooperative search algorithm that introduces competition among agents, just like the selection mechanism in genetic algorithm. If an agent in a good state with good heuristic value, it tends to have more offspring's in the next generation. Knight in [1] illustrates Multi Agent Real-Time A* algorithm in which multiple agents autonomously and concurrently executes RTA* where the look-ahead horizon is set to be 1.

A variation of LRTA* is a State Mark Gradient having exploration and exploitation phases; agents follow the ascending state marks gradient, which leads it from the initial state to the goal state explained in [14].

3 Test System and Algorithm Implementation

Our simulated environment dynamically generates mazes with varying obstacle ratio as well as other configuration settings such as starting locations, target locations etc. The goal of the agents is to cooperatively attempt to find the path to the goal state in various random scenarios.

Agents have been given a visual depth of one state. Color code is used for coordinating actions with each other. Agent can move in eight directions and each direction has an associated color value. Each agent takes action based on distance heuristic and color code. When Agent reaches the state with a specific color value, it chooses the alternate state which enhances the space exploration.

We have conducted 1200 trails on 100 randomly generated mazes with a varying obstacle ratio, from 5% to 55% with an increment of 10%. Statistical influence based on experimental results demonstrates that using a color coded coordination in the LRTA* online search algorithm, the solution quality, measured by solution length and algorithm runtime, is improved.

4 Discussion

In the learning Real-time A* search agents move randomly and no coordination among them exists. The theory of multiagent systems proposes that agents can better perform by coordinating with each other.

Our work is on using a color coded coordination scheme. As more agents are engaged in the search, agents update an estimated cost, which, in turn, improves the solution quality.

4.1 C^{3}LRTA*

A new color coded coordination scheme has been proposed and we have used this technique to solve varying size mazes with randomly positioned obstacles. Initial position of the agents and target will be randomly assigned. Agents can move in eight directions including diagonals in the search space as shown in figure (1) below.

Agents have to avoid the obstacles in the environment so agents can't pass through the blocked state. Each move in the search space has some associated color to represent agents' action from the current state.

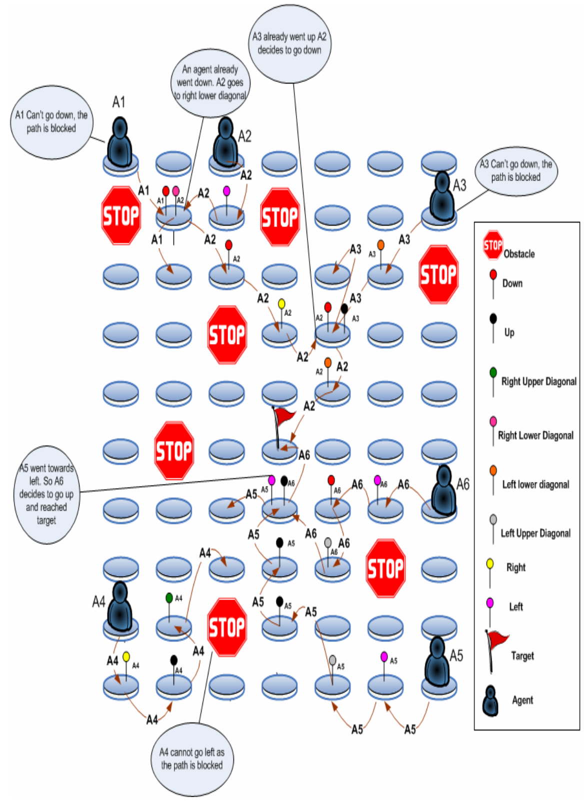


Figure (1) Agent's movement towards goal while avoiding obstacles at the same time

As shown in figure (1) Agent A1 moves downwards as the minimum heuristic state and put state color as red. When A2 reach the same state; assigns the state color for right lower diagonal as pink. So both A1 and A2 choose the different paths towards the goal. Depending upon the last agents' move the color of the state will changed accordingly. The minimum heuristic states are selected alternatively while considering the state color. So agents disperse themselves in the search area and explore different part of the search space.

Previously coordination is used for real-time A* (RTA*) search and no coordination was proposed for LRTA*. This paper makes the following contributions:

- An updated scheme of coordination
- The original LRTA* does not cater for diagonal moves.

Like the original LRTA*, we also cater for obstacles in the search space.

In addition, in this work, we have created a framework and test harness which can be used in analyzing online algorithms such as C^{3}LRTA* and LRTA* algorithms.

4.2 Efficiency of C^{3}LRTA*

Our proposed simulation contains randomly generated mazes with all edges having unit cost where environment contains obstacles. The agents and target initial states are randomly chosen. Where initial state heuristic values are zero and color values are white.

4.2.1 C^{3}LRTA*

The C^{3}LRTA* algorithm repeats the following steps until one of the problem solver reaches the goal state. It builds and updates two hash tables, one containing heuristic estimates of the cost and other containing the state color from each state in the problem space.

In the following description let i be the current position of the problem solver, and h(i) be the current heuristic estimate of the cost from i to the goal.

[C^{3}LRTA*]

Lookahead:

Calculate f(j) = k(i,j) + h(j) for each neighbor "j" of the current node "i" . where h(j) is the current lower bound of the actual cost "j" to goal state, k(i,j) is edge cost between node "i" and "j". where "j" contains no obstacle.

Check State Color If White then move to minimum heuristic f(j) else move to state pointed by color code scheme.

Update heuristic:

Update node "i" as follows:

h(i) = minj f(j)

where "j" is non-traversed node.

Update Color Code:

Update the color of node "i" as follows:

Color(i) = Color of Current_Move

Action Selection:

Move to the neighbor "j" that has non-traversed minimum f(j) value. Ties are broken randomly. If traversed then move to next available f(j) in clockwise direction. If no non-traversed move to "j" left then start again from best heuristic. Agents try to move in a coordinated fashion and heuristic values become exact values over repeated trials.

The original LRTA* algorithm takes O(N3) average moves but in this case we need to consider the Obstacle avoidance and color Coordination.

The obstacle and color values are stored in hash tables; it takes negligible time and space manipulation i.e O(1) making total of O(N3 + N + N). So for single agent in N state space the learning time would be same as O(N3).

4.3 Completeness Questioned

Under what conditions is LRTA* with color code coordination guaranteed to eventually find a goal state? The original LRTA* has a property that due to repeated problem solving trails the heuristic values converge to exact values. We have assumed that goal state and initial problem solver states are assigned randomly.

We also assume that initial heuristic values are admissible and do not overestimate the goal state. Ties are broken randomly.

Theorem 1: C^{3}LRTA* when given an admissible heuristic h and the non-traversed node color white. The color coordination will disperse agents in search space. Eventually C^{3}LRTA* will converge to a final solution of the cost O(N3).

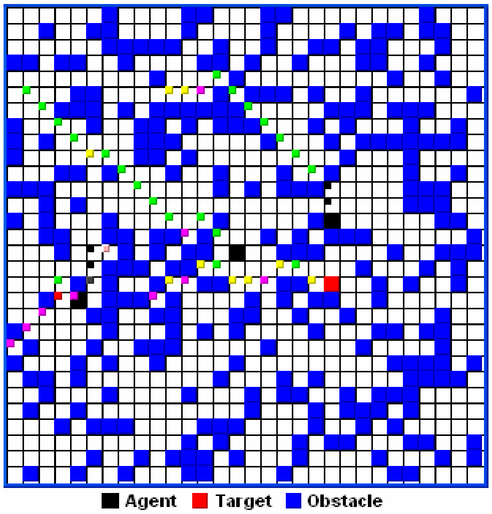
Proof: Assume the converse, that there exists a path to goal state but C^{3}LRTA* cannot reach it. There must be a finite cycle that it travels forever. For the traversed nodes heuristic and color value are stored in a hash table. Considering a single move of an agent, it reads the value of its neighbor heuristic and current color value, updates both color and heuristic values of current node. Any other agent when reach to this state try to choose the alternate path so maximum space will be explored. In an infinite cycle each visit to particular state will increase the heuristic value of the node. Similarly each time agent selects the next minimum and leaving the few choices for others. So at some point, the algorithm will leave the cycle and avoid other agents to be stuck in similar cycle using color code. So in a finite problem space, every node including goal will be visited at most once. When all agents cooperatively reached the goal node, algorithm will terminate successfully.

5 C^{3}LRTA* in random Obstacle Mazes

In our framework multiple agents are trying to find the single target through the mazes.

Agents and target positions is randomly defined. The grid space used is of different sizes with different ratio of obstacles. Agents can move in eight directions including the diagonals. Each agent is assigning the state color code according to their moves. The previous path traveled is avoided to be traverse again.

Initial heuristic values from the current state to the goal are calculated using the Euclidian distance. And these values are improved over the time while considering the coordination scheme.



Figure(2). C³LRTA* Framework

6 Performance Analysis

Performance of both LRTA* and C³LRTA* has been evaluated on different mazes with different obstacle ratio.

Both LRTA* and C³LRTA* are run on same configuration that is agent positions, target position and obstacle ratio is same. For each obstacle ratio we have generated ten different mazes with random obstacle positions. For each maze twenty trials are run then the average of these trials are taken. Average of these 10 mazes is generated against same obstacle position.

Following Table (1) shows the average of 10 mazes against obstacle ratio range from 5-55.

LRTA* VS C³LRTA* (Solution Length)			
Obstacle Ratio	LRTA*	C³LRTA*	Percentage Improvement
5	25.86	25.81	0.1937
15	31.68	31.55	0.4199
25	26.32	25.76	2.1739
35	24.42	24.15	1.1180
45	49.41	47.62	3.7589
55	156.74	141.56	10.7212

Table (1)

Results shows that as we increase obstacle ratio C³LRTA* outperform LRTA* because of the dispersion of agents using color coded coordination. Figure (3) shows the comparison of C³LRTA* Vs LRTA*. C³LRTA* has gradually improve with increasing obstacle ratio.

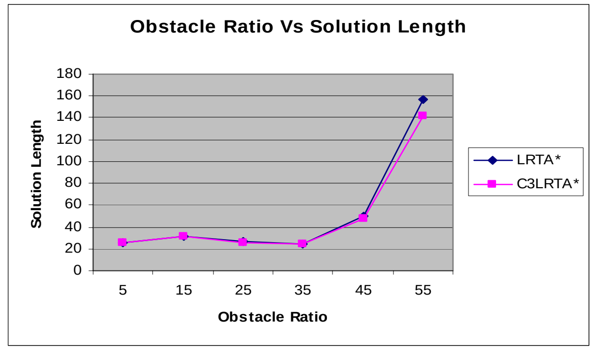


Figure (3) : Obstacle Ratio Vs Solution Length

Figure (4) depicts percentage improvement of C³LRTA* over LRTA* with respect to obstacle ratio. Performance of C³LRTA* increased as less number of available paths to agents. If the target is near to any agent then the expected performance will be less as shown in Figure(4) at obstacle position 35.

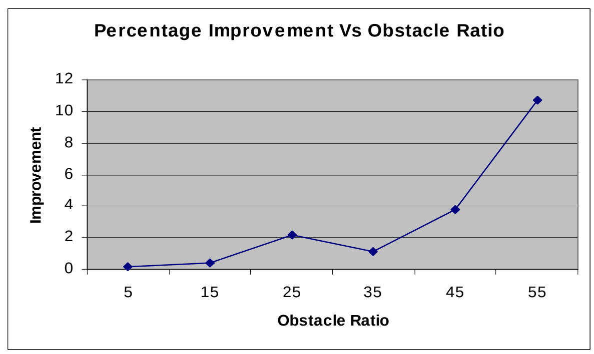


Figure (4) : Percentage Improvement Vs Obstacle Ratio

We have evaluated both LRTA* and C³LRTA* against the optimal moves for various obstacle ratio. LRTA* and C³LRTA* performance is same when obstacle ratio is less then 15. As we increase obstacle ratio LRTA* takes more moves then C³LRTA*.

LRTA* Solution VS Minimal Solution C³LRTA Solution VS Minimal Solution			
Obstacle Ratio	LRTA*	C³LRTA*	Minimal Solution
5	18.93	18.92	18.7
15	22.84	22.73	22
25	26.37	25.86	23.73
35	25.6	25.57	20.9
45	23.41	22.41	18.27
55	138.38	103.41	35.3

Table (2)

After 40% obstacle ratio there is a major difference between C^3LRTA* and LRTA* moves.

Table (2) shows the moves of both algorithms at various obstacle ratios.

We have generated 10 different mazes for same obstacle ratio. Afterwards obstacle ratio has been increased from 0 to 55 percent. Taking average of 25 trials for each such maze shows the LRTA* number of moves against the optimal moves.

Figure (5) explains that LRTA* is gradually taking the more moves than optimal as obstacle ration increased. But after 30% obstacle ratio LRTA* is taking considerable more moves than optimal moves.

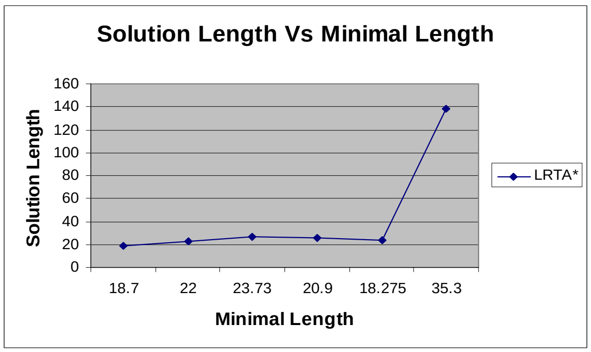


Figure (5) : LRTA* Solution Length Vs Minimal Length

C^3LRTA* is also taking the more moves than the optimal moves. If we compare the both algorithms than as the obstacle ratio increased C^3LRTA* takes considerable less moves than LRTA* due to coordination. At 55% obstacle ratio LRTA* is taking about 140 moves where C^3LRTA* takes about 100 moves to achieve the target.

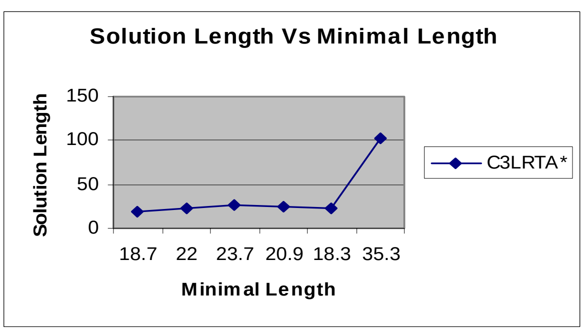


Figure (6) : C^3LRTA* Solution Length Vs Minimal Length

Table (3) shows the time taken by both LRTA* and C^3LRTA* on different mazes. At zero percent obstacle ratio both LRTA* and C^3LRTA* takes equal time but as we start to put obstacle in the search space LRTA* takes more time to reach the goal state.

Table (3) shows, as we increase obstacle ratio LRTA* takes more time to coverage as compared to C^3LRTA*.

LRTA* VS C^3LRTA (Time Taken)			
Obstacle Ratio	LRTA*	C^3LRTA*	Difference
5	5478	5442	36
15	6829	6647	182
25	5475	5355	119
35	4968	4895	73
45	5783	5554	229
55	31573	28536	3036

Table (3)

Figure (7) shows the comparison graph between LRTA* and C^3LRTA*.

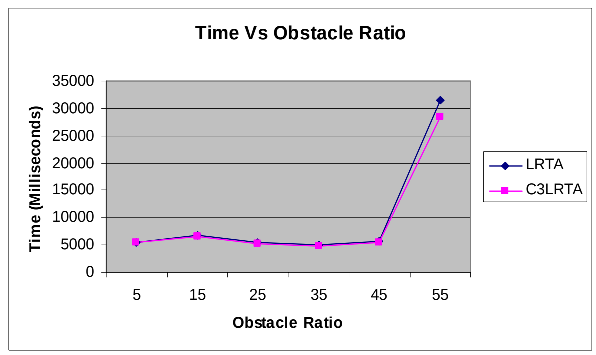


Figure (7) : LRTA* Vs C^3LRTA* (Time Taken)

Difference in time between LRTA* and C^3LRTA* is plotted. Time difference increases as we increase obstacle ratio.

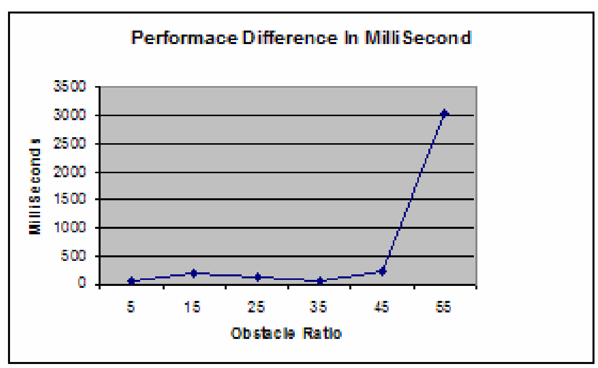


Figure (8) : Percentage Improvement Vs Obstacle Ratio

7 Future Work

This work can be extended in several dimensions. One possible extension is for unknown target. Agents do not have any information about target position in advance, agents coordinating to locate the target in search space. The LRTA* with coordination can be implemented in 3D space and can also be incorporated with GIS for Missile systems to hit the static target.

Several static and dynamic obstacle detection algorithms can be devised for multi-agent LRTA* with coordination.

8 Conclusion

We have here presented two innovations in this paper. Using C^3LRTA* enhances search space which leads to improvement in solution quality. In addition, LRTA* is a single agent search algorithm. We have implemented the LRTA* algorithm to work on multiple agents on a randomly generated maze with obstacles in search space.

This work can be regarded as an extension to Mark's strategy and leads to a better exploration of the search space in addition to improving the solution quality and solution length.

Experimental results have also shown that proposed coordination scheme performs better then LRTA*.

In addition, our proposed scheme becomes more effective as we increase the obstacle ratio in the search space.

9 Acknowledgement

We would like to thank Mr. Ahsan and Hina for there kind reviews. In addition we would like to thank Summiya to give her valuable time for developing Architectural Diagram.

References:

[1] Knight "Are Many Reactive Agents Better Than a Few Deliberative ones?" in Proceeding of the Thirteenth International joint Conference on Artificial Intelligence, 132-137. International Joint.

[2] Toru Ishida "Real-Time Search for Autonomous Agents and Multiagent Systems", Department of Social Informatics, Kyoto University 1998.

[2] Yasuhiko Kitamura, Ken-ichi Teranishi and Shoji Tatsumi "Organization Strategies for Multi-agent Real-Time Search", Faculty of Engineering, Osaka City University 1996.

[3] Richard E.Korf, "Real-Time Heuristic Search", Computer Science Department, University of California 1990.

[4] Makot Yokoo and Yasuhiko Kitamura, "Real Time A* Search with Selection: Introducing Competition in Cooperative Search" 1996.

[5] Stefan Edelkamp and Jurgen Eckerle, "New Strategies in Real Time Heuristic Search".

[6] Catgatay Undeger, Faruk Polat and Ziya lpekkan, "Real Time Edge Follow, A New Paradigm to Real-time Path Search".

[7] Armagan Cakir and Faruk Polat, "Coordination of Intelligent Agents in real-time Search" 2002.

[8] Gehard Weiss, "Search Algorithms for Agents Chap 4", Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence.

[9] "Informed and Uniformed Searches chapter 4,5,6" from AIMA 2003.

[10] Florin Leon, Mihai Horia Zaharia and Dan Galea,"A New Approach in Agent Path-Finding using State Mark Gradient" 2004.

[11] Sven Koenig, "A Comparison of Fast Search Methods for Real-Time Situated Agents" 2004

[12] Yasuhiko kitamura, Makoto Yokoo, tomohisa Miyaji and shoji Tatsumi,"Multi- State commitment Search".

[13] David Furcy and Steven Koenig, "Combining Two Fast-Learning Real-Time Search Algorithms Yields Even Faster Learning", Georgia Institute of Technology , Atlanta 2001.

[14] Florin Leon, Mihai Horia Zaharia, Dan Gâlea, "A new Approach in Agent Path-Finding using State Mark Gradient" Computer Science Journal of Moldova, Chişinău, 2004.

[15] V.Scott Gordon and Zach Matley, "Evolving Sparse Direction Maps for Maze Pathfinding" California State University and Digital Eclipse Software,Inc.