Applying Coordination in Moving Target Search (MTS) Algorithm
UMAR MANZOOR, KIRAN IJAZ, ARSHAD ALI SHAHID
Department of Computer Science,
National University of Computer & Emerging Sciences-FAST,
H-11/4 A.K Barohi Road, Islamabad.
PAKISTAN

Abstract: - Multi Agent System, where multiple agents concurrently and autonomously search to find a target, is regarded as a promising paradigm for future distributed computing. Agents often do not know the terrain in advance but automatically observe it within a certain range around it. In real time heuristic search such as Moving Target search (MTS), we have to commit a move within a limited search horizon or time and these heuristic search algorithms learn and improve their performance over successive problem trails. Multi Agent Moving Target Search (MAMTS) is a multi-agent version of Moving Target Search (MTS) algorithm where multiple agents concurrently and autonomously try to achieve a common goal. In this paper, we propose C^3MTS algorithm which uses color code for coordination among multiple agents. Each agent observes the color code of the state and selects the next move on the basis of this color; multiple agents, therefore, will find and travel distinct paths towards the goal state. In contrast, multiple agents may travel the same path in the original MAMTS. We have evaluated this coordination scheme on a large number of problems of varying difficulty generated randomly. Experimentally we have shown that as we increase the difficulty level C^3MTS out performs MTS.

Key-Words: - MTS, A*, C^3MTS, Online Search, Agents, Multi-agent, Coordination, Color Code, Real Time

1 Introduction
Search, especially heuristic search is a fundamental problem solving method in artificial intelligence. Heuristic search takes advantage of a heuristic function for the shortest path to the goal state. The heuristic search is admissible or optimistic if the heuristic function is a lower bound [5]. The heuristic function should be inexpensive to compute and is used to focus the search in the direction of the goal. There are two different groups of heuristic search algorithms; offline and online searches [8]. Offline search algorithms such as A* execute the complete search before actually taking a step towards the goal state. These algorithms examine various paths towards the goal state before they follow the best one. However online search algorithms compute a plausible next move and then execute the move in constant time. Thus online search algorithms may decide to move to a node that is not on the optimal solution path [2]. Online algorithms such as MTS typically find solution faster than offline algorithms however solution quality may be compromised.

The efficiency of the online search algorithms in successive trails may improve since these algorithms can use the new heuristic information available at the nodes in the search tree. In some learning algorithms such as Learning Real Time A* (LRTA*) where the target goal is static, these values converge eventually to the optimal ones [3]. In Moving Target Search (MTS) where the target goal is moving these values converge eventually to the optimal ones if the movement of target is controlled, not random. The quality of the learning strategies reflects how fast the optimal values can be achieved and how many nodes have been expanded in each trail [5].

In case of multi-agent searches, where several agents are looking for solution, another level of complexity is introduced. These agents can either solve the problem independently or coordinate with each other to improve the efficiency in reaching a common goal in previously unseen environment with obstacles. By increasing the number of agents, we have more chances to find a better solution because more paths towards the goal state will be explored [4].

In MTS algorithm, each agent chooses its move independently and does not coordinate its move with others. There is high probability that more than one agent will travel the same path. In this paper, we have proposed Color Code Coordinated MTS (C^3MTS) by using color coded coordinated scheme proposed in [1] for coordination to show, how agents can coordinate their actions to find a better solution in the context of Multi-agent system.
By increasing the number of agents engaged in the search using C\(^3\)MTS algorithm, we have more chances to find a better solution because more distinct paths toward goal state will be explored, in contrast to MAMTS algorithm where multiple agents will travel the same path redundantly. Agents try to find alternate move in C\(^3\)MTS which leads to better search space exploration. Experimental results based on simulation have demonstrated some exciting results. We have observed that as we increase the difficulty level by increasing the obstacle ratio the solution quality of C\(^3\)MTS algorithm gets better and better as compared to MAMTS.

The paper is organized as follows. First, the brief overview of multi-agent search algorithms and coordination schemes proposed for different real time online search algorithms is discussed. Next the description of simulated system along with multi-agent MTS (MAMTS) and C\(^3\)MTS working will be discussed. After that C\(^3\)MTS algorithm is described in section 4. Next a critical analysis of the performance of C\(^3\)MTS in the randomly generated mazes with MAMTS is given in section 5. At the end we draw conclusion and outline some questions for further research.

## 2 Multi-agent Search Background

Searching is one of the universal problem-solving methods, especially in AI. Learning, planning, pattern recognition, robotics, and theorem proving are some of the areas in which search algorithms play a key role. A number of heuristic search techniques, such as the A* and iterative deepening A* (IDA*) [3] algorithms have proved their success in many application areas and these algorithms are regarded as offline search algorithm which computes an entire solution path before executing the first step in the path.

Real-time A* (RTA*) proposed by Korf [3] is a online search algorithm. Unlike traditional offline search algorithms which compute the entire solution path before executing the first step in the path, it performs sufficient computation to determine a plausible next move, executes that move, then performs further computation to determine the following move and so on, until the goal state is reached. RTA* does not find optimal solutions, but it can find semi-optimal solutions much quicker than the traditional offline search algorithms.

Learning Real time A* (LRTA*), a learning version of RTA* is the same as RTA* except for the estimation update step. This algorithm updates with the best estimation for not overestimating the actual cost. It guarantees that the heuristic values will eventually converge to the optimal values but it takes more time then RTA* because it needs to revisit the same state more times than RTA*[3].

Multi-agent Real Time A* (MARTA*) was proposed by [4] to improve solution quality by increasing the number of agents engaged in the search. Multiple agents execute RTA* autonomously and concurrently. By increasing the number of agents, a better solution can be achieved because more paths towards the goal state are explored and the computational complexity of MARTA* grows linearly as the number of agents increases. In MARTA*, each agents chooses its move independently and do not coordinate their actions with each other, so multiple agents may search the same path redundantly.

Naizi, et al, in [1] proposed Multi-agent Learning Real Time A* (MALRTA*) where multiple agents execute LRTA* autonomously and concurrently. Some of the Coordination mechanisms for Real-Time searches have already been proposed. 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 [11]. Gordon and Matley introduced 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 [7]. Two organizational strategies based on repulsion and attraction, to coordinate agents’ move has been proposed in [10].

Original moving target search was proposed by [12], where the problem space was represented as connected graph. Knight in [4] 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 [6].

An extension to the Learning Real-Time A* (LRTA*) algorithm by utilizing a color coded coordination scheme was proposed in [1]. The proposed Color Code Coordinated LRTA* (C\(^3\)LRTA*) algorithm uses color code for coordination among multiple agents. Each agent observes the color code of the state and selects the next move on the basis of this color, in contrast it simply moves randomly in the original MALRTA*.
3 Frame Work

In our framework multiple agents are trying to find a single target through randomly generated mazes of varying difficulty level. Agent(s) and target position is randomly defined. We have compared MTS and $C^3$MTS algorithm on 1,000 problems of varying difficulty randomly generated with varying obstacle ratio ranging from 5 to 50 percent with an increment of 10% percent as shown in Figure 1. Agents can move in eight directions including the diagonals. Diagonal movement is not allowed in the original MTS algorithm. Each agent is assigning the state color code, according to its move. The previous path traveled is not traverse again.

In figure (2), if agent moves to the right, the color of current state will be changed to yellow. If the agent moves to the right upper diagonal, the current state color will be changed to green and so on. 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.

In the MTS, agents move randomly and no coordination among them exists. The theory of multi-agent systems proposes that agents can better perform by coordinating with each other. Color coded coordination scheme has been proposed in [1] where agents use colors to coordinate their 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 used this technique to solve varying size mazes with randomly positioned obstacles. Initial position of the agents and target will be randomly assigned.

Agents have to avoid the obstacles in the environment (i.e. can’t pass through the blocked state). Each move in the search space has some associated color to represent agent action from the current state.

3.1 Simulation

As shown in figure (3) Agent A1 moves downwards as the minimum heuristic state and put state color as magenta. When A2 reaches 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.

Figure (1). Randomly generated mazes with varying difficulty level.
So agents disperse themselves in the search area and explore different parts of the search space. We assume that agent(s) and Target moves from one state to other alternatively.

we have created a generic framework and test harness which can be used to analyze online algorithms such as C⁢MTS and MTS. Our proposed simulation contains randomly generated mazes with all edges having unit cost where environment contains obstacles. The initial states of agents and target are randomly chosen. Where initial state heuristic values are zero and color values are white. Statistical influence based on experimental results demonstrates that using color coded coordination in the MTS search algorithm, the solution quality (measured by solution length and algorithm execution time) is improved.

4 Color Code Coordinated MTS (C⁢MTS)

Moving Target Search (MTS) algorithm is a generalization of LRTA* where the target can also move. Similar to MTS, C⁢MTS acquire heuristic information for each target location and maintains a matrix of heuristic values, representing the function \( h(x, y) \) for all pairs of states \( x \) and \( y \). There are two different events that occur in the algorithm, a move of the agent(s) and a move of the target, each of which may be accompanied by updating the heuristic value.

The MTS algorithm repeats the following steps until one of the agents 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.

**Steps in C⁢MTS**

**When the agent(s) moves**

1. Calculate \( h(x', y) \) for each neighbor \( x' \) of \( x \).
2. Update the value of \( h(x, y) \) as follow
\[
    h(x, y) \leftarrow \max \left\{ h(x, y), \min_t \{h(x', y) + 1\} \right\}
\]
3. Update the color of node “\( i \)” as follows:
   
   \[
   \text{Color}(i) = \text{Color of Current_Move}
   \]
   Check state color if white then moves to the neighbor \( x' \) with the minimum \( h(x', y) \) else move to state pointed by color code scheme.

**When the target moves**

1. Calculate \( h(x, y') \) for the target’s new position.
2. Update the value of \( h(x, y) \) as follow
3. Reflect the target’s new position as the new goal of the agent(s).
The original MTS algorithm takes $O(N^3)$ average moves but in case of $C^3$MTS we need to consider the color Coordination. The color code values are stored in hash tables; it takes negligible time and space where environment contains obstacles. The agent(s) manipulation takes $O(1)$ time, making total of $O(N^3 + N)$. So for single agent in $N$ state space the learning time would be same i.e. $O(N^3)$.

5 Performance Analysis
The performance and efficiency of the proposed algorithm as well as comparison with MTS was analyzed on different randomly generated mazes with different difficulty levels. Both MTS and $C^3$MTS were run on same configuration (i.e. agent positions, target position and obstacle ratio).

For each trial, agent’s position and target position as well as obstacle positions are same and for each run positions of agents, targets and obstacles in the grid space are defined randomly.
For each difficulty level (i.e. obstacle ratio) we have generated twenty different mazes with random obstacle positions. For each maze fifty trials are run then the average of these trials are taken. Average of these twenty mazes is generated against same obstacle position. Figure (4) graph is the average value of twenty different mazes against obstacle ratio ranging from 0-50 with an increment of 10 percent. Figure 4 shows that as we increase obstacle ratio $C^3$MTS outperform MTS because agents tries to find distinct paths towards the target using the coordination scheme.

Figure (4) shows the comparison of Coordinated MTS Vs Original MTS with respect to search time. Experimental results have shown that as we increase obstacle ratio in the search space $C^3$MTS gradually improves over original MTS with respect to time taken as agents try to find distinct paths towards the goal state.

Figure (5) shows the effect of increasing the number of agents engaged in the MTS algorithm. At obstacle ratio 0 percent increasing the number of agents has very less effect because agent path is not blocked by obstacles. But as we keep on increasing the obstacles, number of agents engaged in the search will have positive effect (i.e. improved solution length).

Figure (6) shows the same pattern for $C^3$MTS as figure (5) but as we increase obstacle ratio $C^3$MTS will perform better then MTS because more distinct paths towards the goal state will be explored. Figure (7) shows the performance improvement of $C^3$MTS with respect to obstacles.

\[
h(x, y) \leftarrow \max \left\{ h(x, y), h(x, y') + 1 \right\}
\]
6 Conclusion

In this paper, we have presented C³MTS algorithm which uses color code for coordination among multiple agents. Experimentally we have shown that use of C³MTS enhances search space exploration which leads to improvement in solution quality. MTS is a single agent search algorithm. We have implemented the MTS algorithm to work on multiple agents on a randomly generated maze with obstacles in search space. Experimental results have shown that proposed color code coordinated MTS performs better than MTS. Moreover, our proposed scheme becomes more effective as we increase the obstacle ratio in the search space.

References: