A Study on Seamless Information Sharing between Robots in Identifying the Optimal Path: An Agent based Approach

M.Bala Subramanian¹, Dr.K.Sudhagar², G.Rajarajeswari³
¹Bharath University, Department of Computer Science and Engineering, Chennai, India
Email: balamadevan@gmail.com
²Agni College of Technology, Department of Mechanical Engineering, Chennai, India
Email: mechod@act.edu.in
³Rrase College of Engineering, Department of Information Technology, Chennai, India
Email: sidhi_raji@yahoo.com

Abstract—Developing a concrete agent based architecture model through a heterogeneous agents and designing the best possible lane for a autonomous mobile robots are one of the key area, which gains much attention to the researchers in this field. This paper aims the study on behavioural aspects, navigational and optimal path analysis of an autonomous mobile robot, based on agent based system. The key objective of this paper is to find a sub-optimal lane for an autonomous mobile robot using master-subordinate architecture in a known environment based on agents. The sub-optimal lane is determined by a heuristics approach, called A Star (A*) algorithm. The communication control between a robot and its server will be accomplished based on the defined open agent architecture, which will be treated as a master-subordinate architecture. The robots used to communicate with its server to update the optimized path details, required to achieve its target position, in an optimized way. The system evaluation was done using a graphical user interface (GUI) based test-bed environment for robots, called MobileSim and the efficiency of the system was measured, via the simulation results.

Index Terms—Mobile Agents, Open Agent architecture, Path planning, Handoff negotiation, A* algorithm

I. INTRODUCTION

The mobile robots are designed for wide range of mission on critical application, in which most of the applications are depend on its navigation from the initial position to target position, in a known or unknown environment [1]. In an autonomous mobile robot (AMR) research community many approaches have been proposed and under proposal for optimal path planning problem. An AMR should reach to its final target position precisely, efficiently and reliably as possible, based on the given partial environmental knowledge and on the specified target positions. The decision making and execution process that the system utilizes to achieve its target point is typically meant by navigation competence, in which it directly linked by its representation called cognition. In an AMR research area, two challenging factors that depend on each other
are Planning and Reactive. When a Mobile robot (MR) is in execution, there are high chances for path obstacles before it reaches to its destination and without reacting to these unforeseen events, it will not reach to its goal point. So, in order to reach the target position, the MR must include its current location details and its environmental details, that it gains during its execution, in which it instantly produced new plan along with the new information details to, met its target goal point precisely.

A novel approach by introducing a new data structure called multi agent navigation graph, which is constructed from voronoi diagrams was discussed [1]. Since, the MR work in a changing (Dynamic) and non-changing environment (Static), a Heuristic approach based path planning algorithm is required to construct the navigational details, upon its real time execution. For this specific reason, a heuristic based path planning approach – A star algorithm, receive considerable attention in this literature survey.

The agent based approach also collects, much attention in this literature study. Ugur et-al included a study on agent based approach in a static environment [2]. Xi Li and Byung-Jae Choi proposed a new fuzzy control system and an optimal obstacle avoidance algorithm for path planning in unknown environment for MR [13]. Kawamura et-al presented a user centric human robot interface for supervisory control of a group of MR. The basic bi-directional human robot-interaction and event-triggered adaptation between the system’s agents was demonstrated [3]. Yen proposed a Belief-Desire-Intention (BDI) based multi agent architecture for reconfigurable embedded platform for a real time tracking system [18]. Omranpour and Shiry, proposed a new evolutionary algorithm to solve simultaneous localization and mapping issue [15].

In this paper, an agent based investigation have been made, to understand the optimal navigation process and study on path construction details along with the control of an MR for better knowledge gaining. Zavlanos and Pappas proposed a method to solve multi-agent assignment problems, in the nature of MR [4]. Rushan et-al applied agent approach to a heterogeneous MR team to find a solution for localization problem [8]. Wavelet based path planning problem using agent based approach was proposed [6]. In exploring to an unknown mission critical environment, multi robot pays much attention on its advantage of its concurrency, in which it reduces the time needed for its mission. Hence, co-ordination among the MR is needed among them [7]. The MR receives its current localization details, in which later it distributes to all other MR to perform their tasks [8].

The structure of the paper is organized as follows: In Section 2, the architectural frame work study on constructing the multi agent system is shortly presented. Also, a study on heuristic based approach called A* algorithm for sub-optimal navigational system and the application details are explained in Section 3. In Section 4, presents simulation study and the obtained results. Section 5 presents concluding remarks and guidelines for further work. It is assumed that the reader has basic knowledge on mobile agents.

II. ARCHITECTURAL FRAMEWORK STUDY

Agent is a concept used by wide range of people in wide range of application. In this agent based research community, this term has wide range of variants as, Autonomous agents, Behavioural skills agents, Compound agents, Commander agents, Distributed agents, Environment agents, Hardware/resource Agents, learning agents, Multi-type agents, Navigation/Path Planning agents, Sequencer agents, Simulation agents, Self agents, etc.[9]. The concept of agents comes from developing a thinking machine with the capability of solving a problem on its own [10]. An agent can be a piece of software that is capable of accomplishing tasks on behalf of its user. Agents are capable of perceiving their environment, but only a limited extent [11]. By exchanging information with other agents, they can acquire more information about the environment. Actions are taken by the agent to satisfy its objectives based on some satisfaction/survival functions that it tries to optimize using its skills. The agents are grouped together in the format of multi agents, in which it carries out their task independently to meet its goal.

A. Multi Agent System

In a multi agent system (MAS), agents carry out separate but interdependent tasks to meet their final goal. Every agent needs to send and receive messages and to make decisions based on near real-time site situations [11]. A meta-model for the analysis and design of organizations in MAS was proposed earlier [10]. One of the controversy or major issue in MAS is Path-planning. There are multiple ways in defining the path plan. As shown in Fig 1, the path planning for MAS will be in form of centralized or distributed or hybrid manner [11].
The centralized method requires, single complex agent system to generate a global path for its navigation where distributed method requires, plans to be generated only for the individual agents and uses coordination rules and techniques to combine the individually generated plans.

B. *Open Agent architecture*

A comprehensive study on Autonomous agents, Self Agents within MR and Path/Navigation Planning agent on an Open Agent Architecture (OAA) framework is presented in this work. The OAA provides a framework for the construction of distributed software systems, which facilitates the use of cooperative task completion by flexible, dynamic configurations of autonomous agents [10]. Fig 2 shows the OAA details. In the OAA the communication between the agents and the cooperation among them are done via Inter-Agent communication language called, ICL. ICL is used by an agent to task itself.

OAA agents employ ICL to perform queries, execute actions, exchange information, set triggers, and manipulate data in the agent community. Set of rules for the participating agents in OAA were pre-defined with its capabilities were articulated in ICL. The information is shared among the mobile agents, thru the ICL messages. The ICL messages contain the sender, the receiver and the message information in the message body. All of the messages are flow through the facilitator agent. The Facilitator agent is a pre-constructed agent in the OAA environment. A request for one of an agent’s services normally arrives in the form of an event from the agent’s facilitator [14]. The facilitator is used as a coordinator of the cooperative task completion.OAA triggers are used as mechanism for performing some specific set of actions, when it satisfies some condition. Each agent will make use of these triggers, characterized in the form of Communication, Data, task and Time based.

C. *Robot Agents: Co-ordinator Agents: Path Planning Agents*

In Fig 2, the robot agents used to communicate with the OAA thru the defined communication channels with its current location/co-ordinate points and requesting for the temporal target co-ordinates from path planning agents. It has the knowledge base which also includes the Kinematics and Dynamic constraints of a MR vehicle. The action protocol details for the robots are desired based on the heuristic expert judgment. The Path planning agents will create a new optimal map plan and updates the robot agent with the details. The coordinator agent is used as a facilitator, in which its knowledge base includes information about the task and action to be carried out. This agent will manage the priorities of the robots that are in the same area.
consolidates the micro and macro task that are assigned to a robot and it continuously monitor those tasks based on the priorities. Here, the overall functionality covers with sensing, communication among the robots and the decision making based on the priorities.

D. Path Planning Flow Diagram

The path planning agents, uses a Heuristic based A* algorithm for optimal path generations. Fig 3 shows a scenario of re-planning when two MR have potential obstacles/collision on their paths.

![Path Planning Flow Diagram](image)

The Robot agents start executing their current navigational plans and continuously sense the probable impact/obstacles for the next movements. Once a probable collision/obstacle is sensed, the robot agents send signals to stop the robot execution and retrieve the current robot’s configurations. The robot agents communicate with the coordinator agent to collect the information about the obstacle and to send information about their current paths and tasks to the coordinator agent.

The coordinator agent decides whether re-planning is needed or not. If there is no need for re-planning, the two robots agents negotiate with one another and with the coordinator agent to avoid collision by adjusting their priorities. In worst case scenario, when two robots are moving together for same goals, one robots will be tend to stops it operation and waits until when the other robot leaves the conflict area.
E. Communication and Hand-off negotiation within the agents:

Communication among the robots is limited to the agents within a part of its environment/area, where it performs a specific action. So, in order to resolve this communication blockage, we need to split the environment into different range zone’s (Zone/Area (Z-1, 2...N) based on the total area if it is a static environment, where the total area is known. Based on boundary of the monitored area, the controller of the agent system will activate and de-activates it robotic movement. Here, each MR is considered as Robotic mobile agents (RA1, RA2...RAN) similar like Nodes. These RA’s are considered as self-organized and it should have a communication system along some user pre-defined protocols.

Fig. 4 Mobility management and Hand-off negotiation

Fig 4 shows the mobility management for MAS, which resemble similar to a node in an ADHOC network, communicates via AODV routing protocol. Based on the current location of each RA1...N, using facilitator the path planning agent also decides to which Robotic agent (RA1,2...N) the information of that object should be sent. The coordinator agent communicates with all agents that are under its control and receive messages and partial plans from these agents. In addition, it decides the priorities for movement if any conflict occurs, and it sends commands to the agents to avoid any obstacles/collisions. Agent can use intelligent techniques for generating different protocols for performing specific actions based on the situation. Principled negotiation between the intelligent agents allows all the agents in the system being benefitted from multiple independent declarative analysis of the same situations [5]. In a multi agent environment, Robotic agents often need to interact with other agents in order to achieve their objectives or improve their performance. Heuristic’s based negotiation models were included [16].

Negotiation is a form of interaction in which a group of agents, with conflicting interest and a desire to cooperative try to come to a mutually acceptable agreement on the division of scarce resources. Negotiation Protocols are the set of rules that govern the interaction. It covers the permissible types of participants, the negotiation states, the events that cause negotiation states to change and the valid actions of the participants in particular states [16]. The negotiation between the agents occurs when, 1) when probable collision/obstacles detected between the two robots, in which the two robot agents negotiate with one another and the collision between the robots will be avoided by adjusting the robot’s priority. 2) The robot agents sometime rejects the decision made by the coordinator agent based on the situation, in which it also suggest a better way by adjusting its coordination strategies [11].

F. Coordinator Agent Functions: Graph representation

The mobility management functions included in the robot agent, co-ordination agent is supported by the functional entities at the time of handoff between the two Robot agents [12]. This function enables the co-ordination agents to; create a geometrical map, based on the mode of transport as shown in Fig 5. Creates a graphical structural representation based on its current co-ordinates between Initial and target location as
shown in Fig 6, to create the path representation using the generated acyclic graph, based on the obstacles between the path and the network traffic as shown in Fig 7.

The mobility and the handoff techniques are further discussed [12]. A good negotiation provides a powerful and flexible foundation for MAS coordination. Internally as the handoff process is considered it can be further carried out using the following main steps: system discovery, handoff decision, and handoff execution. During the system discovery phase, mobile terminals equipped with multiple interfaces have to determine which networks can be used and the services available in each network. During the handoff decision phase, the mobile device determines which network it should connect to. During the handoff execution phase, connections need to be rerouted from the existing network to the new network in a seamless manner. There are three strategies for handoff decision mechanisms: mobile-controlled handoff, network controlled handoff, and mobile-assisted handoff [17].

III. PATH PLANNING ALGORITHM

A robot navigation system includes a planner module which determines appropriate path based on its environment map and the obstacle avoidance algorithm determines a suitable direction of motion based on recent sensor data. Path planning considers a model or a map of the environment to determine the geometric path points for the MR to track from its initial starting position to the target or goal position. Many algorithms are proposed by many researchers for static and dynamic environment. Oualid et-al proposed a method using DVFF approach based on D* algorithm [19]. D* algorithm is the earliest proposed by the robot centre of the Carnegie Mellon Stentz, and firstly applied in the Mars scout of "Sojourner" Mars rover which carried by "Mars Pathfinder" detector in United States on December, 1996. The process-state function and the modify cost function in D* algorithm is used to calculate the optimal path cost to target and the cost function for arc cost.
There are reasonable number of algorithm, for search graphs for the shortest path between the nodes like Dijkstra’s algorithm, Variations of Moore algorithm, and A* algorithm. A Star algorithm (A*) is a best first graph search algorithm that finds a least cost path from a given initial node to one goal node. A* is generally considered to be the best path finding algorithm. Also in comparison the Dijkstra’s algorithm is essentially the same as A*, except that there is no heuristic (H is considered to be ‘0’ always). Since it has no heuristic, it searches/explores by expanding out equally in every direction. So it usually ends up exploring a much larger area before the target is found. This generally makes it slower than A* and less optimized compared to A* which is much preferred in identifying the optimal path planning.

A* search is one of the widely used informed search strategies. It uses heuristic functions, h(x) which provides approximation for the cost of the best route that goes through the nodes. It will always find a path if a path exists and report failure if a path does not exist and the path that A* returns will be optimal in terms of the heuristic function. This heuristic function is used to accelerate the search process. It is used to find a path from the starting node to the goal node in a graph [20].

It is assumed that, the environment is divided into 1m by 1m square grids. The center point of each square is considered as a node of the graph. Then A* uses this graph to construct the path. The cost for each grid is calculated as the sum of two costs:

\[ f(n) = h(n) + g(n) \]  

Here, the function g(n) is the cost to reach from the starting node to the node n, h(n) is the cost to reach from the node n to the goal node. (This function is often chosen as the straight line distance between a robot's current position and the goal position disregarding all obstacles in the environment). Since, g(n) gives the path cost from the start node to node n and h(n) is the heuristic distance which is the estimated cost of the closest path from n to goal, f(n) becomes the estimated cost of the shortest solution through node n.

A* is an optimal search strategy if h(n) is an admissible heuristic that is provided that h(n) never overestimates the exact cost to reach the goal. The input for A* search algorithm is the graph and the output is a back-pointer path which is a sequence of nodes starting from the goal and back to the start. In the case of O is the open set which is a priority queue and C is the closed set containing all processed nodes, the A* search algorithm can be expressed as below [3].

**While O is not empty**

1. **Find n<sub>best</sub>** from O such that f(n<sub>best</sub>) ≤ f(n) for all n in O
2. **Remove n<sub>best</sub>** from O
3. **Add n<sub>best</sub>** to C
4. **if n<sub>best</sub> = n<sub>goal</sub>**
   - **EXIT**
5. **Expand n<sub>best</sub>** for all x, neighbour of n<sub>best</sub> and not in C
   - **If x is not in O**
     - **Add x to O**
   - **Else if g(n<sub>best</sub>) + dist(n<sub>best</sub>,x) < g(x)**
     - **Update x’s back-pointer to point the n<sub>best</sub>**

Implementation point of view, we need to have the Initial and target position; Dimensional details (includes the work space cell details); Map file of the workspace; output the optimal path as a list of waypoints. The region of a grid cell uses eight-point connectivity relation and the heuristic distance (h(n)), which is calculated by using Euclidian distance which is always smaller than or equal to the actual distance.

**IV. SIMULATION RESULTS**

A numerous number of MR simulation systems were used in the past, all with the goal of creating an artificial robot environment, as real as possible, considered to be the test bed for implementing the MR algorithms. The realistic error model is for sensors and actuator used in the design model allows the code
development, testing and debugging the robustness of the robot programs that can handle with the real environments.

The majority of the robot simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems they all include the kinematic or dynamic models of robot manipulators. Which type of models will be used depends on the objective of the simulation system. For example, trajectory planning algorithms rely on kinematic models [21].

The effectiveness of the above defined approach is tested and evaluated in MobileSim simulation environment, which shows promising results in the existence and nonexistence of anonymous obstacles. MobileSim is a test-bed for Pioneer robots and the programs are written with Aria framework. The programs that work correctly in simulator also work in the real life. Working environment is drawn by Active Media Mapper program and imported to MobileSim program [4] Using this information and the map of the environment, the A* algorithm constructs a path between the starting point and the goal point. And the constructed path is passed through communication channel to the robot. The trace of the path followed by the robot is shown in below Fig. As a part of simulation activity, the input for this simulation is defined as grid position co-ordinates. The input details are passed to the robot agents. The initial position is of grid, (X, Y) is (1, 1). And the target point is (19, 9). The robot agent will identified the optimal path using the A* algorithm. The trace details of different simulation scenarios and environment are also captured. Fig 8, 9 shows the behavior movement aspects of a pioneer robot, defined with its initial position and the movement from initial to the target positions are captured. The path trace details are captured and shown in Fig 10. The MR movement details are captured and shown in below Fig 11, 12 for different case scenario.

![Fig 8. Behavioural map details of Pioneer robots](image1)
![Fig 9. MR movements from Initial to Target position](image2)

![Fig 10. Ray Trace of Pioneer MR with movement details](image3)
V. CONCLUSION

A comprehensive study was made on agent based real-time optimal path planning algorithm for AMR. The navigation path is constructed by A* heuristic algorithm, which is considered to be one of the best paths finding algorithm, based on its environment exploration. The discussed approach is simulated on a GUI based MobileSim simulation tool. The results are prominent and accurate. The robot agent navigates from its source point to the destination as per the defined input values. This simulation study is based on the static environment and further it can be extended for the dynamically changing environment and also it can be applied in real time scenario as future extended work.

REFERENCES

[17] Wenchao Ma, Yuguang Fang and Phone Lin, “Mobility Management Strategy Based on User Mobility Patterns in Wireless Networks” – IEEE transactions on vehicular technology, VOL. 56, NO. 1, JANUARY 2007