

Development of an Intelligent Routing Algorithm for Multi-Agent System in Grid Environment

Hardik Joshi¹, C. Lakshmi²

¹Department of Computer Science, SRM University, Chennai, India

²Department of Software Engineering, SRM University, Chennai, India

Abstract: *The paper illustrates the possible problematic scenarios in multi-agent system and proposes a solution. The aim is to develop and illustrate a solution that can be applicable to an environment that has multiple agents in grid domain and can deal with scenarios like deadlock and collision. The developed solution has been tested on a simulated environment having a grid structure and having Agents and Obstacles, which draws an analogy with a “store-floor” like scenario in a shopping mall where Obstacles can be compared with shops, counters and Agents as moving robots.*

Keywords: Multi-agent system, grid world domain, deadlock, collision, field allotment.

1. Introduction

A multi-agent system is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity [1]. An Agent is the basic and fundamental unit of an agent system, situated in an environment, which is capable of flexible autonomous action in order to achieve its goals [2]. Multi-Agent systems have become prominent and may become milestones for the future as they have multiple areas of application like designing of transport system [3], e-commerce [4], artificial intelligence, complex game development, network management etc. The area has been touted well in [5] and is described in a much precise manner in [6], although still a unanimous definition and description is not available.

Cooperation refers to distributed and communicated group of agents that share a common interest and work together to achieve a common goal in dynamic environment [7], [8]. In multi-agent systems, while working for shared common interest, sometimes situations come that have to be tackled precisely. On the other hand, non-cooperation may provide faster operations as it does not have to store or retrieve any information related to any other similar entity.

2. Related work

Some previous work that has been accomplished for multi-agent system in grid world is reported in this section. A Multi-Agent System for Solving Packet World Problem in Grid was proposed by [9] which use cooperative agents. The objective of this research is to clean the environment of packets in minimum number of steps and minimum time cost and less network consumption. The environment of packet world problem consists of three main components: packets, agents and basket. The global goal is to clean the environment from packets; main task of agent here is to pick up a packet, bring it to basket and drop it down. Each packet requires single agent to clean it.

The research carried out by [10], proposes a continual planning technique. The importance of communication for MAS was shown by [11]. In [11] there are Agents that are intended to move from an initial position to a target position. In the same, it has been shown that communication has a significant beneficial impact on Multi-Agent System. It has been shown in [12] that coordination and criteria for choosing coordination are essential and pave a way for cushy proceeding. In the same, an overview of coordination techniques is discussed.

3. Scenarios taken into consideration

If there are multiple entities working together in a situated environment, then conditions like deadlock and collision arise. Another condition, where overlapping of allotted resources may also have to be taken into consideration, may arise. These can be seen as

3.1. Deadlock situation

A situation or condition when no decision can take place and no progress is done at all. For grid domain if an agent tries to acquire a cell which is not available; or tries to go in a direction where there is no place available to move, then deadlock occurs. A cell being not available has the meaning that the cell is out of bounds i.e. out of the defined range. Such a situation can be seen in Figure 1.

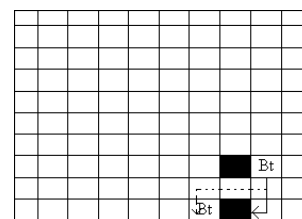


Figure 1: Deadlock

Here the agent Bt cannot move further if it chooses the indicated path (not dashed line) as an obstacle comes ahead and no other path exists.

3.2. Collision situation

A situation when two or more agents try to occupy the same position at the same time. If this conflict is not handled in time then it may lead to deadlock also since no progress can be made. In grid world this is shown in Figure 2

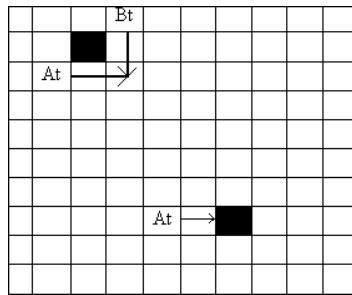


Figure 2: Collision

Here both the agents try to occupy the same cell at the same time causing collision. An agent trying to move into a cell occupied by an obstacle can also be called collision.

3.3. Field allotment

Suppose an agent has a capacity of covering a 3*3 unit grid space. If two agents are placed like A & B are in figure 3, then they are covering 15 units grid space resulting in a potential loss of 3 units, but if both Agents are placed as C and D are, then they cover 18 units. The field allotment is shown below in Figure 3.

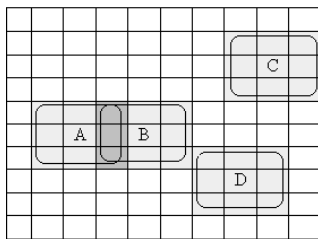


Figure 3: Field Allotment

The overlapping of allotted area results in a lack of efficient usage of resources.

Focus on the above three scenarios is important. For example, these scenarios arise in the case of Robots in real world. In the real world conditions may arise in which they may come in each-others way. A Robot having a capability of searching an area of 3X3 grid like a bomb detector agent or a moping agent(of the same capacity), has to deal with another robot in field allotment for its effective utilization. On the other hand, Collision is natural as there may be other Obstacles also on the field. Deadlock may arise on the real world field as when a Robot chooses a path which leads it into a dead end.

4. Grid World domain and pre-conditions

The grid world is made of rows and columns. Here a grid of size 10X10 cells is used. The cells that are dark in color are considered as obstacles. Here two types of agents are used:

1. Transport Agents - Indicated by AT and BT.

2. Cleaning Agents - Indicated by AC and BC.

The primary purpose of using the Transport Agents is to show deadlock and collision, while the primary purpose of using the Cleaning Agents is to show field allotment.

The cleaning agents have their own space of 3X3 grid cells each. They can be seen as robots that can mop an area of 3X3 grid cell space at a time. There are some other considerations such as:

- i. A cell can have only one entity, either an agent or an obstacle at a time.
- ii. A Transport Agent can be in the cells owned by a Cleaning Agent, but another Cleaning Agent cannot come inside the cells owned by a Cleaning Agent i.e. no overlapping of owned areas.
- iii. An agent can have only local information i.e. the information of the eight cells surrounding it.
- iv. For acting in grid world domain an agent can move in eight surrounding directions like NORTH, EAST, SOUTH, WEST, NORTHEAST, NORTHWEST, SOUTHEAST and SOUTHWEST.
- v. The direction consideration is taken as compass directions. On deciding to act to move in the SOUTH direction, the agent moves one cell down. Other directions work in a similar fashion. With these all an agent is instructed to take a step.
- vi. An Agent's steps are contiguous to show the occurrence of deadlock, collision and field allotment; and are not intended to reach a given target position.

5. An approach towards the grid world domain

The Agents and the Obstacles are put in the situated environment. For this, they are using an Actor module to interface with the Grid module. The notion of interface is influenced by [9], [13]. The abstract of the overall system is shown in Figure 4.

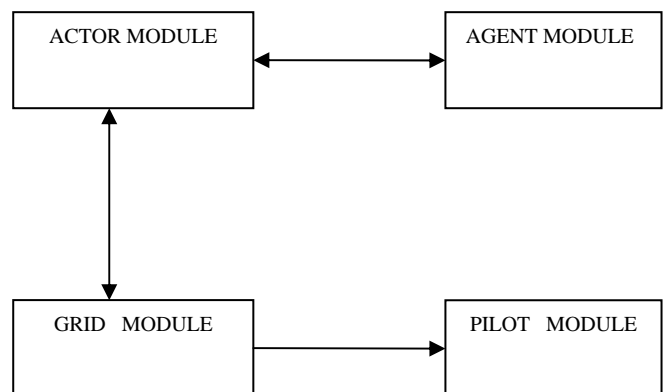


Figure 4: Abstract of Overall System Simulation

5.1. Agent Module

This module defines two types of Agents i.e. moving agents as AGENTS and stand-still agents as OBSTACLES. It defines the actions and the nature of the agents.

5.2. Actor Module

Both kinds of agents are cumulatively referred to as Actors. It provides an interface with the grid world. The agents of both kinds i.e. Agents (moving) and Obstacles (stand-still) are put in the situated grid world via this module.

5.3. GRID Module

Situated grid world system is created with this module. The grid world contains cells in grid architecture in which actors are to be placed.

5.4. Pilot Module

This module simulates the overall system and demonstrates the intended movements and actions of agents in grid world.

A flow chart of the non-cooperative algorithm used can be seen in Figure 5, if the Agent gets the cell valid i.e empty and bounded then it would take an action. If the next cell to the Agents current position is not valid, the Agent would make a decision for further action.

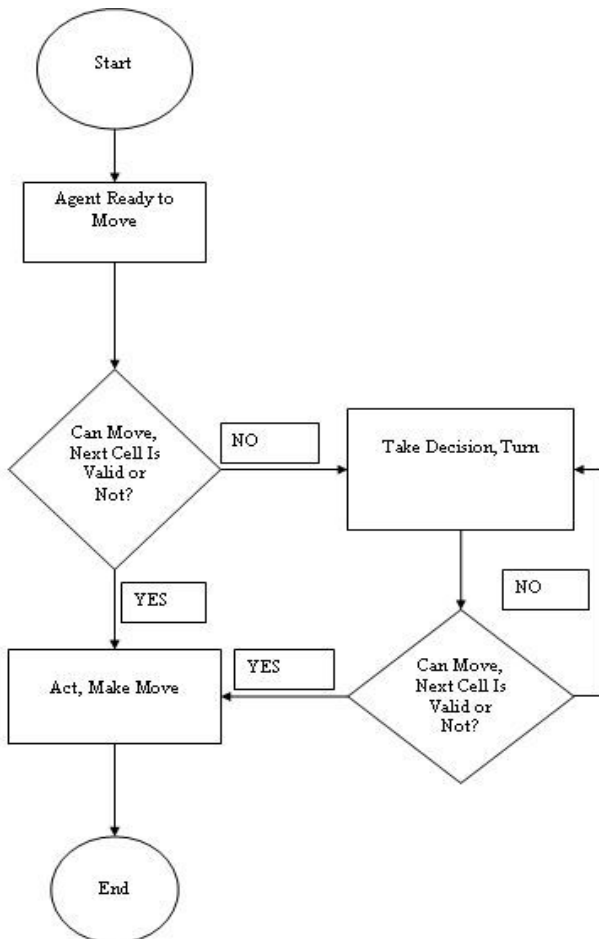


Figure 5: Flow of the non-cooperative algorithm.

6. Simulation work and output

The aim of the experiment is to demonstrate the possible problematic scenarios in grid world and to show a model to handle them. Here the approach is contiguous in steps that

show the Agents' movements to deal with the proposed scenarios; they do not have to reach a given location.

To do this, a situated grid world with actors is used i.e. Agents and Obstacles with their predefined characteristics. All the experiments are performed on a system with a 1.75 GHz Intel Premium processor with 1 GB of RAM and Window XP (SP-3) installed.

The grid cells are counted 1 to 10 from left to right for rows and 1 to 10 from top to down for columns, defining a grid of 10X10. The multi-agent system for the grid world is developed through JAVA programming language with some user defined and some standard methods provided by JAVA library itself. Greedy Approach is used in some cases like for decision making actions that are to be taken at collision or deadlock conditions.

The simulation environment is shown in Figure 5, where;

- i. AT, BT,AC,BC are moving Agents and,
- ii. Black cells are stand-still Obstacles.

Going with the jargon of Multi-Agent System, both Agents and Obstacles are also agents, but with different properties.

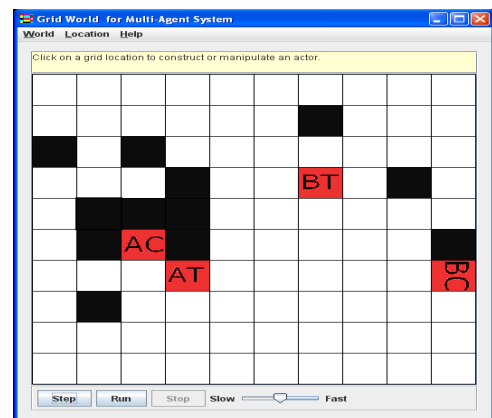


Figure 6: Simulation Environment

Here in Figure 6, it can be seen that agent AT and BC has collision and AC has a deadlock like condition. Here the aim is to provide some decision making ability to get rid of such situations.

Now to avoid or to escape from such situations the agents take an action accordingly. AC and AT in Figure 7 move themselves in North-East direction.

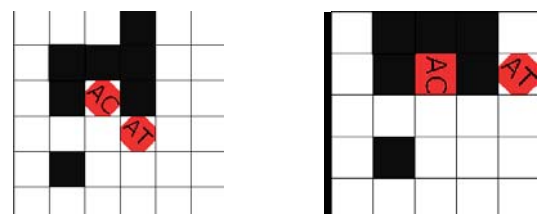


Figure 7: Action Taken By the Agents

Agent AT got a proper cell to move and it moved ahead while Agent AC is on the way to find the solution for getting rid of the situation. Figure 8 shows this.

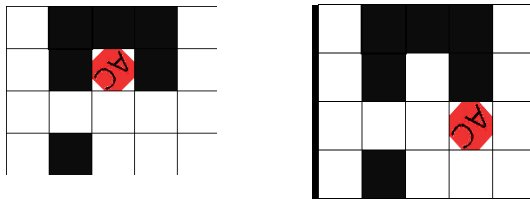


Figure 8: The Agent AC Got a Cell to Move On

The Agent got the appropriate cell to move on and finally got out of the stuck situation. Here the Agents are moving onto the next valid cell and not proceeding towards any particular target position.

7. Conclusion and Future Work

The scenarios like deadlock, collision and field allotment are shown here. The experiment shows that with contiguous decision making actions using pre-information about the state of objects in grid world, such scenarios can be handled in a cushy way just like in the stuck situation, the Agents moved in a decision based direction to get rid of the situation. Simulation work is done for deadlock and collision scenario. In future the same grid world for different sized and shaped obstacles with more agents would like to be studied. And the considered scenario of field allotment shown in Figure 3 above would be simulated. The solution approach can be applied with some enhancement on other areas such as computer networks to show the same scenarios i.e. deadlock, collision and field allotment (resource allotment) according to the requirement.

8. Acknowledgement

The simulation environment design is taken from the College Board [14]. It provided a case study manual that worked as an unspoken guide.

References

- [1] Durfee, E.H., Lesser, V.R. and Corkill, D.D. "Trends in Cooperative Distributed Problem Solving", IEEE Transactions on Knowledge and Data Engineering, pp. 63-83, (1989)
- [2] Nicholas R. Jennings, Katia Sycra, Michael Wooldbridge, "A Roadmap of Agent Research and Development", Autonomous Agents and Multi-Agent Systems, pp.275-306, Kluwer Academic Publisher, Boston, (1998).
- [3] De Schutter, B.; Hoogendoorn, S.P.; Schuurman, H.; Stramigioli, S, "A multi-agent case-based traffic control scenario evaluation system", 2003 IEEE , Page(s): 678 - 683 vol.1
- [4] Zhiyong Sui and Lixin Huang, "Study on an E-Commerce System based on Multi-Agent Technology," International Symposium on Electronic Commerce and Security, 2008, pp.714-717, 3-5 Aug. 2008
- [5] N. R. Jennings and M. Wooldridge (eds.) (1998) "Agent technology: foundations, applications and markets" Springer Verlag.

- [6] M. Wooldridge and N. R. Jennings (1995) "Intelligent agents: theory and practice" The Knowledge Engineering Review 10 (2) 115-152.
- [7] N. vlassis, A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence, 1st ed., Morgan & Claypool, 2007.
- [8] I. Timm, D. Pawlaszczyk, Large Scale Multiagent Simulation on the Grid, Proceedings of the Workshop on Agent-based grid Economics (AGE 2005) at the IEEE International Symposium on Cluster Computing and the Grid (CCGRID). Cardiff University, UK, 2005.
- [9] Abusnaina, A.A.A, Chan Huah Yong, "Cooperative Multi-Agent System for solving Packet World Problem in Grid," 2009 IEEE 9th Malaysia International Conference on Communications (MICC), pp.754-758, 15-17 Dec. 2009.
- [10] Edidiong, Behrouz "Coordination in Multi-Agent Systems: An overview" SEGN 609.22-Agent Based Software Engineering Tutorial report December, 2002
- [11] Chouhan, Niyogi "Multi-Agent planning in a grid world doamin", 2012 Second International Conference on Advanced Computing & Communication Technologies DOI 10.1109/ACCT.2012.75
- [12] M. Brenner and B. Nebel, "Continual planning and acting in dnamic multiagent environments," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 3, pp. 297-331, 2009. 802.11, 1997.
- [13] Paul, John,, Carles, Juan, "Applications and environments for multi-agent systems", *Auton Agent Multi-Agent Syst* (2007) 14:61-85 DOI 10.1007/s10458-006-9002-5 Published online: 25 August 2006 Springer Science+Business Media, LLC 2006
- [14] <http://www.collegeboard.com>

Author Profile



Hardik Joshi received the B.E degree in Computer Engineering from University of Rajasthan in 2009 and is currently pursuing M.TECH in Computer Science & Engineering from SRM University (2011-2013).



Dr. C. Lakshmi is professor and Head of Software Engineering Department in SRM University. She received her PhD in 2010 from SRM University. Her areas of Research Interests are Image processing, Pattern Recognition, Software engineering and fuzzy logic.