STRATEGIES FOR ROUTE PLANNING ON CATASTROPHE ENVIRONMENTS
Coordinating Agents on a Fire Fighting Scenario

Pedro Abreu
FEUP/DEI/LIACC, Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal

Pedro Mendes
EZ4U, Rua Guerra Junqueiro, 417, Porto, Portugal

Keywords: Multi-Agent System, Agent Collaboration, Simulated Environment.

Abstract: The concept of multi-agent systems (MAS) appeared when computer science researchers had the need to solve problems involving the simulation of real environments with several intervenients (agents). Solving these requires a coordination process between agents and in some cases negotiation. Such is the case of a catastrophe scenario with the need intervention to minimize the consequences, like for instance a fire. In this particular case the agents (firemen) must have a good coordination process to achieve as fast as they can their fire fighting position. The main goal of this project is to create an optimal strategy to calculate the best path to the fire fighting position. Tests were conducted on an existing simulator platform Pyrosim. Three factors have an important role: wind (intensity and direction), ground topology and vegetation variety. At the end the results were quite satisfactory, mainly in what concerns the agents main objective. The A* algorithm proved to be feasible for this particular problem, and the coordination process between agents was implemented successfully. In the future this project may have its agents ported to the BDI concept.

1 INTRODUCTION

The Multi Agent System (MAS) concept appeared in the early 80s although most of the relevant research trends only began being explored in the 90s (Wooldridge, 2002). At that time, several research publications (magazines, journals and books) started to attract researches all around the world and due to that growth, the European Union created AgentLink (Union, 2008) which is the premier Co-ordination Action for Agent Based Computing. Most of these systems find their application in domains where it is relevant to simulate a real environment with several intervenients with well known responses to stimulus and to solve complex constraint based problems with hundreds or thousands of variables, in which case, hopefully by making every agent fight for himself, a solution close to the optimal one will emerge in a short time. In this research project, the focus is on the systems described first. By using the Pyrosim platform(Thunderheadeng, 2008), which simulates the emergence of natural fires in a forest area, a MAS is proposed in order to coordinate a set of agents that personify typical firefighters against a fire with a single focus. The Pyrosim platform supports variable weather conditions, types of vegetation and other aspects that will be discussed in the next sections. In a catastrophe situation, like a big dimension fire, the main goal is to confine it, so to reduce the damage as much as possible in the forest area. The steps involved in tackling these problems are many. First, one must detect the exact position where the fire is consuming the green area, the second is to determine which is the optimal path to reach the most effective fire fighting position and finally how should the firefighters position themselves so to minimize the fire spreading process and actually confine it. This research project presents a strategy to effectively tackle these goals. The remainder of this paper is organized as follows. Section 2 describes the state of the art regarding MAS concepts and research topics. Section 3 shows some of the most relevant features of this particular approach in terms of applied algorithms and undertaken strategies. Section 4 exposes results obtained and finally in section 5 conclusions are presented and future work trends are discussed.

Abreu P. and Mendes P. (2009).
STRATEGIES FOR ROUTE PLANNING ON CATASTROPHE ENVIRONMENTS - Coordinating Agents on a Fire Fighting Scenario.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 277-280
DOI: 10.5220/0001852102770280

2 STATE OF THE ART

In the past few years, MAS proved to be a good approach to solve in cost-effective time a large set of several types of distinct problems. In short terms, a MAS is nothing more than a computer system where typically there are several computing entities (agents) with autonomous or semi-autonomous behaviors interacting simultaneously in a specific environment. The environment may directly related to real world, and in such case the agents (usually the program that a robot runs) base their behavior on sensor reading; or it may interact with a fully digital environment (simulators; industrial control engines etc.), (Lesser, 1999). The architecture of these is in most cases distributed as typically there is no need to centralize the processing of each individual agent and thus overload the computing capacity of the station at hand. Multi-Agent systems usually involve coordinating the agents' actions. This can be achieved following two approaches. In one of them each agent tries to optimize its well being which is something that suits well on competitive situations, as in most cases agents may have contradictory objectives. In the other approach, a global goal is the pursued by a team of agents and thus the coordination involved is quite different specially in what concerns the information to be shared (Jennings, 2000). Such shared information may be distributed among all the agents or there may be hierarchical topologies so to promote agents to leader agents. Coordinated agent positioning is one of most sonant research topics concerning MAS. The concepts involved point to strategies regarding decentralized algorithms to make a group of agents with a logical presence in a 2D or 3D space to position themselves efficiently in order to effectively achieve a common goal. One example of work conducted in this area is presented by Scheutz (Scheutz and Bauer, 2006), where he presents a simulator of a biological swarm where agents jointly achieve tasks using local rules rather than global centralized or distributed strategies. Each agent has a coverable area and the main goal is to maximize the global coverable area by the swarm by making the agents form a geometrical polygon autonomously. Nowadays, there are several real world systems that are simulated using MAS. Many of them are already associated with worldwide competitions while others are related to specific scenarios like particular ecosystems. Some of the MAS that best fit the first case are the ones involved in the Robocup competitions. The soccer competition was the first to be introduced in 1997 on a worldwide scale. In the last few years other competition with MAS emerged. The RoboCup Rescue was probably the most relevant one. Proposed by Kitano (Kitano et al., 1999), the idea behind this new competition is to simulate how do rescuing entities (Fire Brigades, Police Forces, Medical Teams) act after a large scale disaster like tsunamis, earthquakes, tornados, fires. The simulator creates an environment representing a virtual city with a set of rescuing objectives (save people and property) that must be addressed within a limited time frame. It supports six categories of agents with different missions. There is a type of agent for each rescuing force and also a leader agent for each of these forces. The common agents coordinate themselves by communicating with their respective leader. The leaders can also communicate among themselves to plan large scale actions.

3 APPROACH

3.1 The Pyrosim Platform

The massive destruction of our planet's forests is one of nowadays most critical issues for nearly every government on the globe. Following that perspective it is important to study this phenomenon in order to draw better and more precise fight plans. The Pyrosim platform simulates fire in a forest area with several possible focuses. Fire emerges as a result of a combination of multiple nature factors. The spread of the fire is also dependant on several factors that combine themselves on many different ways. The most relevant factors that are supported in Pyrosim simulator are: wind, ground topology and vegetation variety. The Wind has two sub variables that should be taken into account, namely its intensity and direction: Intensity. It is relevant to state that the fire propagation increases with the wind intensity; however, if it has extremely high values, the wind may transform itself in a fire extinction factor. Direction. In a situation analysis the direction of the wind is often the direction of the spread of the fire and thus it is also a relevant factor. In Pyrosim the terrain can be plain or heterogeneous with several valleys and hills. This topology not only because that influence the positions that the fireman (agents) must assume to fight the fire but also because it has an effect on the fire propagation speed. In the case of a *climbing fire* it tends to spread faster as the air overheats and moves up creating optimal conditions for this effect. On the opposite situation the propagation of the fire has a reverse behavior. In Pyrosim the terrain can be plain or heterogeneous with several valleys and hills. This topology not only because that influence the positions that the fireman (agents) must assume to fight the fire but also because it has an effect on the fire propagation speed. In the case of a *climbing fire* it tends to spread faster as the air overheats and moves up creating optimal conditions for this effect. In this specific simulator there are four types of vegetation: trees, sticks, ground plants and humus. Each randomly generated terrain by this platform has different ignition temperatures concerning this factor.

3.2 Multi-Agent System Architecture

In our approach there are two types of agents, namely leader and worker. The classes that control them are *FireBirde* and *FireFoz* respectively. They are both extensions of the *AgentSkeleton* class because each agent must have a physical representation on the map and so it can virtually die. If this happens before his death the leader passes his global information to the healthiest worker which is later promoted to leader. Every 20 ms the leader collects data on its workers. The agent's perceptions and their knowledge of the world are stored instances in the class *floatmap*. These are basically bidimensional matrixes representing an area surrounding the agent in what concerns a given factor of the world. In this research work the used matrixes were the ones related to presence of fire at short and medium distance. The matrixes that represent the presence of fire in the global map and the terrain medium height were also used. These last are less accurate than the previous ones because they are 32*32 and the map is 128*128 which means that each positions in the global map represents an average value of the four aggregates. These large scale matrixes intend to simulate an area pocket map.

3.3 Multi-Agent Communication System

When launched on the platform, the worker agents send their position (POS) to the leader. The leader agent acknowledges the worker as a team member, estimates a good position to fight the fire (the used method is described in the next subsection) and sends it the coordinates of such position (GOTO). After receiving the new coordinates, the worker agent consults his global firemap and calculates the best possible path to achieve these new coordinates. After getting to the assigned point, the agent notifies the leader (FORM) and the leader answers back with a request to enter the team formation. The worker agent also notify the leader of the impossibility to reach the desire destination (e.g the designated point is inhabitable).

3.4 Heuristical Route Algorithm

To store the paths that must be ran, the agents use graphs that are implemented the classes *Graph, Edge and Vertex*. The graph is based on the world map matrix, which is formed by a matrix of 32*32. By default it is assumed that the agents are able to go from a given point to every neighbor in both ways assuming 8 connectivity. This graph is generated in an extremely efficient way as from the top left corner, three threads are launched to quickly explore the map considering the south, southeast and east directions. After the generation process which assumes the same cost in both ways for each edge, a heuristic calculus is performed to reevaluate these variables. The first step is to recalculate all the distances using 3D coordinates and then for each edge the algorithm calculates the percentage difference of height between the two connecting points and adds a cost to one of the ways in case of a climb or a bonus in case of a descent (this is only applied if the considered slope is bigger than 20%). If the slope is larger than 70%, the route is considered to be impossible. The map has enough resolution so that this algorithm does not fail in the presence of valleys. The route calculus is based on the well known A* (Russel and Norving, 2003) star algorithm by providing it the fireman's current position and final position which is estimated according to what is described in the next paragraph. At this point it is relevant to state that each worker agent has a specific memory region to store the graph and process the A* algorithm, allowing a faster path recognition than what would be achievable by simply using a collections data structure provided by the Java Virtual Machine.To estimate a good position to fight the fire, the leader agent consults his world fire map and stores the coordinates where the temperature is higher (fire point). After that to estimate a good fire fighting position, the leader consults the current wind conditions and determines a position where the worker agents may center their formation. The formation will face the fire against the wind. It may slightly rotate if such rotation allows the agents to assume a higher ground position.

3.5 Fire Fighting

After the agent reaches the designated position it asks the leader where it should position itself to start fighting the fire. The leader that knows where every agent is and also knows the current wind direction, assigns each of them a base position on a line that is perpendicular to the wind direction and also that has its 0 axis point aligned with the front of the fire.

4 RESULTS

The obtained results were quite satisfactory, mainly in what concerns the agent's main objective, to extinguish the fire. Using the proposed strategy the agents are able to reach their designated fire fighting position really fast because the A* star algorithm seems to be correctly applied and the processing has been tuned up by using a memory region for the *Pathfinding* of each single agent. The tests were conducted with the default conditions, having the fire a single focus (point of ignition) and having the teams from five to fifteen members plus the leader agent.

5 CONCLUSIONS AND FUTURE WORK

The proposed strategy proved to be efficient in the terms assumed in the previous section. Even small five member teams are able to extinguish the fire, assuming that it ignites when all the agents are already in the map. The Pyrosim proved to be a mature platform mainly in what concerns fire generation and spreading and robustness in the communication protocol between the agents. The heuristics used in the A* algorithm proved to be close to optimal because the paths *walked* by the agents were apparently good choices. Some tests were conducted having worker agents use the A* and others the simulated annealing algorithm (Russel and Norving, 2003). In all the tests the A* agents got to the destination faster than the simulated annealing ones (considering the same departure point). From the visual analysis on Pyroviz, one could state that the second method disregards some interesting paths that seem to be bad on a short term perspective and that are chosen by the A* algorithm. As for future work trends, there are several possibilities. At this point the agents are only able to fight fires with a single focus. However it would be interesting to increase the number of focuses of the fire. The following figure (Figure 1) which comprises data about landscapes with of 1960 square kilometer of area, having the line segment between the agent's starting point to the fire focus about 1300 meters, confirms that the best suited method for this approach is A*.

Method \ Measures	Distance Traveled	Time Taken	Distance Traveled	Time Taken
A*	1396 m	4m20s	1820 m	6m13s
Simulated Annealing	1397 m	4m12s	2096 m	8m14s
Greedy Best First	1395 m	3m50s	1890 m	20m12s
Recursive	1392 m	4m30s	1932 m	18m07s

Figure 1: Results Obtained by different Methods.

Other methods were tried on landscapes with distinct levels of sinuosity. On the left side of the figure, (green background), the values represent tests conducted having Pyrosim configured for plain scenarios while on the right (brown background) mountain ones. On plain scenarios, A* might not be the best approach since there is no real gain with the processing overhead. In this case one might even say that the best solution is to use greedy algorithms. Counter measuring this last observation, A* is clearly the best method for mountain scenarios, which in fact correspond to the places where there is more need for a good planning in finding the correct path. In a real fire fighting situation, the firemen have emotions that affect their judgment and performance. To add these additional variables to our approach, an integration with the work presented by Sarmento in (Sarmento et al., 2004) would probably be a good contribution. The current version of Pyrosim also supports indoor spaces. Fighting fire in these spaces is quite different from fighting in outdoors. Indoors have different types of materials and a flammable conditions, spaces and accesses tend to small sized, smokes may be toxic and the propagation may differ a lot from room to room. To create a good fire extinction strategy that keeps every fireman alive is a challenge by itself.

REFERENCES

Jennings, N. (2000). On agent-based software engineering. *Artif. Intell.*, 117(2):277–296.

Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., and Shimada, S. (1999). Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research. 6:739–743.

Lesser, V. R. (1999). Cooperative multiagent systems: A personal view of the state of the art. *IEEE Trans. on Knowl. and Data Eng.*, 11(1):133–142.

Russel, S. and Norving, P. (2003). *Artificial Intelligence: A Modern Approach Second Edition.* Prentice Hall.

Sarmento, L., Moura, D., and Oliveira, E. (2004). Fighting fire with fear.Proceeding of the 2nd European Workshop on Multi-Agent Systems (EUMAS 2004)

Scheutz, M. and Bauer, P. (2006). A scalable, robust, ultra-low complexity agent swarm for area coverage and interception tasks. *Intelligent Control, 2006. IEEE International Symposium on*, pages 1258–1263.

Thunderheadeng (2008). Pyrosim oficial site. In *Avaiable online at http://www.thunderheadeng.com/pyrosim/.*

Union, E. (2008). Agent link oficial site. In *Avaiable online at http://www.agentlink.org/.*

Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems.* John Wiley and Sons,Ltd.