SIMPATROL
Towards the Establishment of Multi-agent Patrolling as a Benchmark for Multi-agent Systems

Daniel Moreira, Geber Ramalho and Patrícia Tedesco
Centro de Informática, Universidade Federal de Pernambuco, Prof. Moraes Rego Av., Recife, Brazil

Keywords: Multi-agent systems (MAS), Multi-agent patrolling (MAP), MAS benchmarks, MAS testbeds, *SimPatrol*.

Abstract: This paper discusses the establishment of multi-agent patrolling (MAP) as a benchmark for multi-agent systems (MAS). It argues that MAP can be a good benchmark for MAS, and points out what is lacking in order for this to happen. From the identified lacuna of not having a testbed, it presents *SimPatrol*, a simulator of MAS constructed strictly for the patrolling task. With such testbed, new results of performance are obtained for some of the previously proposed patrolling strategies.

1 INTRODUCTION

AI began to focus less on component technologies and more on complete, integrated systems (Hanks, Pollack and Cohen, 1993). In this scenario, multi-agent systems (MAS) offer a good way to design such integrated systems (Russell and Norvig, 2003). The concept of multiple agents perceiving and acting on an environment allows the integration of many distinct technologies, and also systemizes the use of such combined technologies to craft a solution that meets the needs of a specific application.

From the industrialization process, Computer Science inherited the necessity to evaluate and compare the performance of distinct logic systems (algorithms, architectures, etc.). Drogoul, Landau and Muñoz (2007) justified such necessity as a manner to determine what systems are better indicated to solve a given problem, usually called benchmark.

Succinctly, benchmarks are precisely described problems, solvable by distinct techniques and representative of a class of problems. Hanks, Pollack and Cohen (1993) discussed the importance of establishing benchmarks for MAS, as well as Stone (2002), and Drogoul, Landau and Muñoz (2007) criticized the currently most popular MAS benchmarks: the *RoboCup* and the TAC competitions.

Initially not aware of the benchmark issue, many researchers have been conducting studies about multi-agent patrolling (MAP) since 2002 (Machado, 2002; Almeida, 2003; Santana, 2004; Chevaleyre, 2005; Menezes, 2006). Informally, patrolling is the act of moving around a field in order to visit regularly its areas. The goal is to minimize the time lag between two visits to a same place.

MAP is useful for domains where distributed surveillance, inspection or control is required. For instance, patrolling agents can help administrators in the surveillance of failures or specific situations in an Intranet, or detect recently modified or new web pages to be indexed by search engines (Machado, 2002).

Despite the quantity of developed work, up until this point, MAP researchers did not have the means to accurately compare the proposed solutions. Dealing with such difficulties and being inspired by the criticism done about the *RoboCup* and TAC competitions (Stone, 2002; Drogoul, Landau and Muñoz, 2007), one additional issue born was just related to the potential of the MAP problem as a benchmark for MAS.

Thus, this work aims to discuss the establishment of the MAP problem as a benchmark for MAS. Also, as a very first step to fulfil the identified gaps, *SimPatrol* is introduced as a MAS software simulator that offers a testbed for the MAP problem.

Next section discusses benchmarks for MAS, and section 3 presents MAP as a MAS benchmark. Section 4 introduces *SimPatrol*, bringing some interesting new results obtained with the simulator.

Moreira D., Ramalho G. and Tedesco P. (2009).
SIMPATROL - Towards the Establishment of Multi-agent Patrolling as a Benchmark for Multi-agent Systems.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 570-575
DOI: 10.5220/0001807005700575
Copyright © SciTePress

2 BENCHMARKS FOR MULTI-AGENT SYSTEMS

Benchmarks are well-defined problems that simplify a more complex reality. The first goal of such simplification is to support fair comparisons of distinct solutions, as long as it facilitates the submission of such solutions to the same situations.

Being a science of the artificial, besides an analytic-theoretical vein, AI has a growing empirical facet where controlled experimentation is fundamental (Pereira, 2001). So, for such science, benchmarks play an important role that exceeds the issue of comparing competing systems. They are part of the apparatus of empirical AI.

As pointed by Hanks, Pollack and Cohen (1993), AI systems are intended to be deployed in large, extremely complex environments. But before the application to real problems, AI researchers submit their methods, algorithms and techniques to simplified, simulated versions of such environments (i.e. to benchmarks).

Thus, the experimental process consists in the researcher varying the features of the simulated environment, or even the benchmark task, in order to measure the resulting effects on system performance. With such information, the researcher shall be able to discriminate between uninteresting isolated phenomena, and relevant general results, adequately explaining why his (or her) system behaves the way it does.

2.1 Definition

For AI, a benchmark is a problem sufficiently generic in order to be solved by various different techniques, sufficiently specific to let such distinct techniques be compared, and sufficiently representative of a class of real applied problems (Drogoul, Landau and Muñoz, 2007).

Well-known AI benchmarks are the knapsack, the n-queens, and the traveling salesman problems (Drogoul, Landau and Muñoz, 2007). As one can see, such problems are sufficiently generic – since they can be solved by plenty of techniques – and also sufficiently specific, as shown by the number of studies comparing such techniques (Martello and Toth, 1990; Russell and Norvig, 2004; Cook, 2008). These problems are sufficiently representative, too, since they are NP-complete problems (Garey and Johson, 1979.)

However, as pointed by Hanks, Pollack and Cohen (1993), since AI began to focus less on component technologies and more on complete, integrated systems (specially MAS), such classic benchmarks revealed their limitations. They evaluate the performance of component technologies individually, ignoring the interactions of such components one with the others and with the environment.

So, what characterizes a benchmark for MAS? In the same manner as the other benchmarks for AI, benchmarks for MAS must be sufficiently generic and sufficiently specific. Their particularity is constituted just by the *sufficiently representative* issue. As explained by Stone (2002), as long as the complexity of MAS exceeds the NP-completeness theory, their representativeness is related to AI subfields like collaboration, coordination, reasoning, planning, learning, sensor fusion, etc.

2.2 Good Benchmarks

Besides defining what a benchmark is for MAS, an important aspect is defining what turns a benchmark into a good benchmark.

As defined by Drogoul, Landau and Muñoz (2007), a good benchmark is the one that makes the representation and the understanding of new methods easier, letting the researcher focus on the solution rather than on the representation of the problem.

From the point of view of Hanks, Pollack and Cohen (1993), a good benchmark must also count with a testbed. Succinctly, a testbed is a complete software environment that offers an interface to configure the parameters of a benchmark, as well as assures that distinct techniques are being tested and evaluated in equivalent situations.

2.3 *RoboCup* and TAC Competitions

As pointed out by Stone (2002), the *RoboCup* and the TAC competitions comprise the currently most popular benchmarks for MAS. The reason for such popularity is related to the problems that such competitions simplify through their benchmarks.

RoboCup challenges competitors to win soccer games (one of the most popular sports of the world), played by computer agents (Robocup, 2008). TAC, the Trading Agent Competition, challenges competitors to build trading agents that dispute for resources in an e-market (Tac, 2008).

Both competitions have yearly promoted international tournaments. Their benchmarks are indeed sufficiently generic, given the diversity and the quantity of proposed solutions by the many competitors. Such benchmarks are also sufficiently specific, once the proposed techniques are

confronted in tournaments. In other words, it is expected that the winner presented a better performance than the looser.

Having the representativeness issue in mind, TAC can be considered sufficiently representative, once the trading agents deal with many situations of which solutions contribute in the fields of reasoning, planning and learning. However, in the case of *RoboCup*, its representativeness is somehow questioned (Drogoul, Landau and Muñoz, 2007).

Considering, for example, a solution for gathering relevant data from the combination of the sensors of a player, it contributes for the sensor fusion field, being representative. Nevertheless, a solution for the problem of avoiding off sides during a goal attack may not be so representative.

Stone (2002) presented more criticism to the two competitions, pointing out problems like obsession with winning, barriers to enter and even monetary advantages of one competitor over the others.

Despite such drawbacks, both competitions count with stable and consolidated testbeds that improve the quality of their benchmarks. These testbeds are the *Soccerserver*, that supports the simulation benchmark of *RoboCup* (Robocup, 2008), and the *Classis Server*, that supports the *TAC Classic* benchmark of TAC (Tac, 2008).

Inspired by the discussion about benchmarks for MAS, one issue that occurs to MAP researchers is related to the possibility of establishing the MAP problem as one of these benchmarks.

3 MULTI-AGENT PATROLLING

The multi-agent patrolling problem is stated as a MAS of which environment is represented by a graph and of which agents act as tokens moving on it, visiting the nodes through the edges. So, the goal is to minimize the time lag between two visits for each node.

3.1 Can Multi-agent Patrolling be a Benchmark for Multi-agent Systems?

In order to evaluate the possibility of establishing the MAP problem as a benchmark for MAS, the following questions must be answered:

- Is MAP sufficiently generic?
- Is MAP sufficiently specific?
- Is MAP sufficiently representative?

The answer for the first question is shown to be positive by the number of existing techniques developed to solve the MAP problem. Since 2002, a series of works has been carried out. While some researchers tested empirically various techniques – like reactive behaviour (Machado, 2002), machine learning (Santana, 2004) and negotiation mechanisms (Menezes, 2006) – Chevaleyre (2005) presented a graph-based theoretical analysis of the topic.

Taking the second question into consideration, the answer is again positive. MAP is specific enough in order to let the distinct proposed techniques be compared; examples of such comparative studies are Almeida et al. (2004), and Santana (2004).

In all studies, the proposed strategies were compared based on the performance of the patroller agents when submitted to each one of the six maps described by Santana (2004) – actually, an effort to represent the topologies he found to be the most likely to face in real problems.

Finally, having in mind the third question, MAP can be considered sufficiently representative for two reasons. First, from the AI point of view, it deals with situations where solutions contribute to the fields of collaboration and coordination of agents (Machado, 2002), planning (Almeida, 2003), learning (Santana, 2004), communication of agents (Menezes, 2006), etc.

Second, MAP is proven representative enough given the diversity of possible applications. Machado (2002) noticed that it is useful for every domain where distributed surveillance, inspection or control is required.

Answered the three questions, a relevant issue is related to how good MAP is, as a benchmark for MAS. MAP turns the representation and the understanding of new patrolling methods easier in the following ways:

1. The territories to be patrolled are represented by generic graphs. So, when a virtual patroller agent visits the nodes of a graph, it can be the night watchman visiting the rooms of a museum, or maybe a bot inspecting the links of an intranet.
2. The collected metrics are all based on the time lag between two visits to a node (i.e. the idleness of the node). So, it does not matter how the agents decided the order and the exact time to visit the nodes; to impact on the performance of the system, all they have to do is to visit the nodes quickly.

Despite the benefits inherited from the definition of the MAP problem, some criticism can be done about its state-of-art.

3.2 Criticism to the State-of-art of Multi-agent Patrolling

One source of criticism to the previous works of MAP is related to the features that were not taken into consideration by researchers yet. Issues like graphs of which nodes have distinct visiting priorities (like it can occur in the patrolling of a bank, where the room of the main safe must be visited more frequently than the others), or of which nodes and edges can become unavailable in simulation time were not considered in the first proposed patrolling strategies.

In the same way, open societies of agents (i.e. societies of which agents can appear or disappear during simulation) and the possibility to associate energetic costs to the actions and perceptions of the patrollers were not explored.

Actually, one of the most important features not analyzed until this work is related to the counting of time. Taking into consideration the comparisons done among the various proposed patrolling strategies (Almeida et al., 2004), all of them were experimented in simulators that counted the time of simulation based on the agents' perceive-think-act cycle.

So, when researchers simulated their strategies in 1000 cycles (for example), they were actually letting their agents perceive, think and act 1000 times. Similarly, the metrics of idleness were calculated based on such numbers, what means that in the worst case, the maximum possible idleness for the graph being patrolled was 1000.

Due to the diversity of applied techniques and reasoning mechanisms, it was perceived that such method of time simulation represents a problem to the comparing studies. It does not take into consideration the real time spent by the agents when deciding their next actions. So, if a strategy really chooses the best actions but, as a consequence, spends too much time reasoning, it will not be penalized in any form.

Another source of criticism is the lack of a testbed to let researchers easily implement their own techniques and compare it to the previous ones. If deciding to use one of the simulators applied to the previous works, a developer should study carefully the internal mechanisms of such programs, due to the high coupling of modules.

Someone wanting to add new behaviours to the patroller agents, or maybe to implement a new graphical interface to show the simulation, should first know the internal models of graphs, agents, actions and perceptions, and also understand how the eventual controlling classes manipulate them. Add to that, they should also codify their routines in the C/C++ programming language.

Bearing in mind the issues discussed, the necessity for the creation of a better testbed was noticed, in order to support the promotion of the MAP problem as a benchmark for MAS.

In this light, *SimPatrol* was developed.

4 *SIMPATROL*: A TESTBED FOR MULTI-AGENT PATROLLING

SimPatrol is a simulator of multi-agent systems constructed for the patrolling task. Since it is open source, someone wanting to check its code can do it at any time in a SVN-manner by the following address:
http://*SimPatrol*.googlecode.com/svn/*SimPatrol*/.

Its initial version has the basic requiements of territory simulation, where a new graph to be patrolled can be initially loaded, since it is in a XML adequate form. Besides this, there is also the simulation of perceptions and actions provided to the patroller agents: fundamental mechanisms of movement through the nodes and edges of the graph, communication and vision of the neighbourhood and near agents are implemented.

Technically inspired by *Soccerserver* – the simulator of *RoboCup* (Robocup, 2008) – and by *Classic Server* - the simulator of TAC (Tac, 2008) – *SimPatrol* operates using a client-server architecture.

Succinctly, the graph and other configurations are loaded at the server, while patrollers connect to ports on the server in order to perceive and act on the generated environment (working as clients). Due to this feature, such agents can be coded in any language, as soon as they implement the defined communication protocols.

So, the internal models are updated based on the actions intended by the agents, while the simulator periodically provides them their appropriated perceptions. Additionally, some auxiliary ports are reserved to output the MAP metrics, and others are reserved for logging the main events of the simulation, in a way that such events can be obtained online, or properly stored to be played later.

As an effort to extend the MAP research, the simulator also permits the creation of dynamic territories, i.e. graphs of which edges and nodes are associated to time probability distributions that rule the appearing and disappearing of such elements. Moreover, it is possible to give distinct visiting priorities to each node, as well as to create open

societies of agents on the simulation, i.e. in some user defined cases, new patrollers can be added or die in simulation time. With these features, developers can test their strategies in very dynamic environments, adding a new step to the MAP research.

Additionally, having in mind the robotic aspect of MAP, *SimPatrol* provides mechanisms to set a stamina feature to the agents, which means that they can spend an amount of energy when perceiving or acting on the environment. In such cases, the territory can have some supply points where agents can recharge their stamina.

Finally, the simulator also provides the option to count the time of simulation considering the time spent by the agents to think and decide for their next action.

4.1 New Results Obtained with *SimPatrol*

In the work of Almeida et al. (2004), some of the previously proposed patrolling strategies were compared. The set-up of experiments was compounded by the six maps proposed by Santana (2004), all expressed in figure 1. For each map, randomly generated configurations for the initial positions of the patroller agents were equally applied to each strategy.

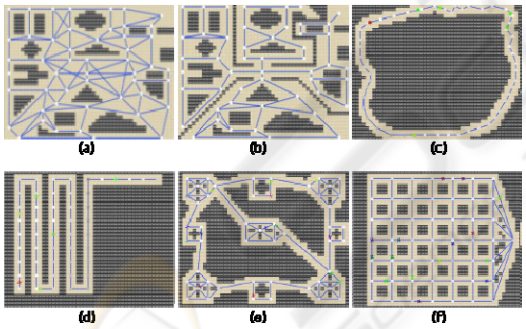

Figure 1: Maps proposed by Santana (2004). In (a), there is the so-called *A Map*. In (b), there is the *B Map*. In (c), there is the *Circular* one, while in (d) there is the *Corridor*. In (e) there is the *Islands* map, and in (f) comes the *Grid*.

Figure 2 shows the results of 4 patrolling strategies applied to each one of the six maps. In each experiment, 5 agents were used to patrol the graph territory.

The y-axis indicates the average idleness of the graphs (see equation 1), measured in the agents' perceive-think-act cycles.

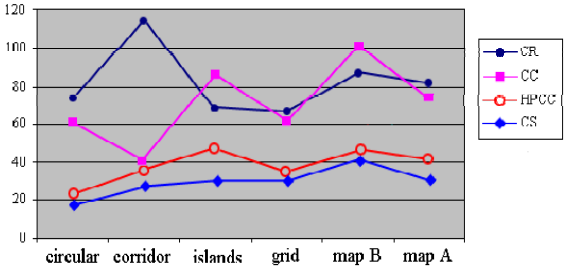

Figure 2: Results for the patrolling task executed by 5 agents (Almeida et al., 2004).

$$I(t) = \frac{1}{t}\int_0^t \frac{\sum_{k=0}^{ord} i_k(t)}{ord}\,dt \qquad (1)$$

In equation 1, $i_k(t)$ is the time lag between the last visit of node k and time t (i.e. its idleness), and *ord* is the order of the graph to be patrolled.

Admitting that the average idleness is inversely proportional to the performance of a strategy, the strategy that presented the best performance was the SC one (Single Cycle strategy). In such architecture, agents are let to patrol the graph moving on its TSP solution, all to the same direction and equally far from its predecessor (Chevaleyre, 2005).

From figure 2, the second best strategy was the HPCC one (Heuristic Pathfinder Cognitive Coordinated agents). In that solution there is a 6th agent that coordinates the others and decides their next goal nodes to visit. In such decision, the coordinator uses a heuristic that takes into consideration the idlenesses and the distances of the nodes (Almeida, 2003).

Similarly, the CC strategy (Cognitive Coordinated agents) also counts with a 6th coordinator, but this one decides the next goals of the patrollers based only on the idlenesses of the nodes (Machado, 2002).

Finally, the strategy that showed worst performance was the CR one (Conscientious Reactive agents). As its name suggests, its agents are reactive, deciding as the next node to visit the one that has the biggest idleness on the neighbourhood (Machado, 2002).

Interested in the impact of the time used by the agents to decide their next nodes, the same described experiments were reproduced in *SimPatrol*. Nevertheless, with the new method of time counting, the obtained average idleness was not based on the agents' perceive-think-act cycles anymore. It was measured in seconds, and reflected the deliberating spent times.

Figure 3 shows the new results obtained with *SimPatrol*. As it was expected, the CR agents, being

reactive, presented a better performance, due to the small time spent to decide their next actions.

On the other hand, the coordinated strategies (CC and HPCC agents) were penalized for the concurred and long service executed by the coordinators in order to determine the goals of all the others agents.

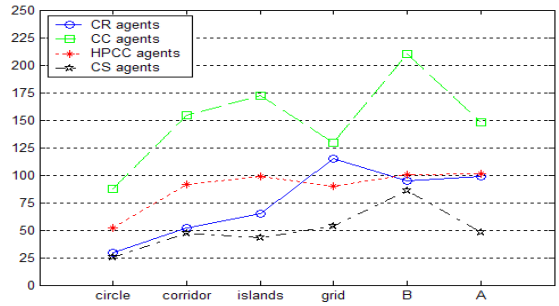

Figure 3: New results obtained with *SimPatrol* for the patrolling task executed by 5 agents.

Curiously, for the maps of which nodes had the smallest degrees (*Circular* and *Corridor* ones), the CR agents presented performance similar to the so-good SC ones. The reason for such behaviour was the small amount of nodes and idlenesses to check due to the small neighbourhoods.

5 CONCLUSIONS

Multi-agent patrolling (MAP) has recently gained attention from MAS researchers due to its representativeness and potential for practical applications. As a contribution to such fields, this paper argued that MAP can be a good benchmark for MAS. Additionally, it also identified what was lacking in order to assure this to happen: a testbed that could let researchers primarily focus on the solution, rather than on the representation of the problem, as well as define parameters that could let them easily apply the proposed techniques in many varied situations (with dynamic environments, open societies of agents, energetic-expensive actions and perceptions, for example).

So, *SimPatrol* was proposed as a software simulator of MAS constructed strictly for the patrolling task. With such testbed, new results of performance were shown for some of the previously proposed patrolling strategies. From these first results, it was perceived the necessity of re-evaluate the performance of all the previously proposed strategies, as a further work.

Due to the new method of time counting implemented, that considers the time spent by the agent patrollers to decide their next actions, different results of the comparative study of the patrolling strategies have been obtained.

REFERENCES

Almeida, A. 2003. *Patrulhamento Multiagente em Grafos com Pesos*. MSc dissertation. Universidade Federal de Pernambuco.

Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V., Chevaleyre, Y. 2004. Recent Advances on Multi-agent Patrolling. In: *XVII Brazilian Symposium on Artificial Intelligence*. Springer-Verlag. São Luís. p. 474-483.

Chevaleyre, Y. 2005. Le probleme multi-agents de la patrouille. In: *Analles du LAMSADE n. 4*. [S.n.]. Paris.

Cook, W. 2008. *The Traveling Salesman Problem*. [online]. Available from: http://www.tsp.gatech.edu/. [Accessed 12 April 2008].

Drogoul, A., Landau, S., Muñoz, A. 2007. RoboCup: un benchmark pour l'IAD? *Unpublished paper*.

Garey, M., Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Bell Telephone Labs. Murray Hill.

Hanks, S., Pollack, M., Cohen, P. 1993. *Benchmarks, Testbeds, Controlled Experimentation and the Design of Agent Architectures*. Technical Report – Department of Computer Science and Engineering, University of Washington.

Machado, A. 2002. *Patrulhamento Multiagente: uma Análise Empírica e Sistemática*. MSc dissertation. Universidade Federal de Pernambuco.

Martello, S., Toth, P. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons. West Sussex.

Menezes, T. 2006. *Negociação em Sistemas Multiagente para Patrulhamento*. MSc dissertation. Universidade Federal de Pernambuco.

Pereira, L. 2001. *Inteligência Artificial: Mito e Ciência*. [online]. Available from: http://www.geocities.com/revistaintelecto/iamc.html. [Accessed 30 September 2008].

Robocup. 2008. *Robocup*. [online]. Available from: http://www.robocup.org. [Accessed 21 May 2008].

Russell, S., Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall. New Jersey. 2nd ed.

Santana, H. 2004. *Patrulha multiagente com aprendizagem por reforço*. MSc Dissertation. Universidade Federal de Pernambuco.

Stone, P. 2002. Multi-agent Competitions and Research: Lessons from RoboCup and TAC. In: *RoboCup 2002. Fukuoka. Robot Soccer World Cup VI, Lecture Notes in AI*. Springer-Verlag. Berlin. p. 224-237.

TAC. 2008. *Trading Agent Competition*. [online]. Available from: http://www.sics.se/tac/page.php?id=1. [Accessed 23 May 2008].