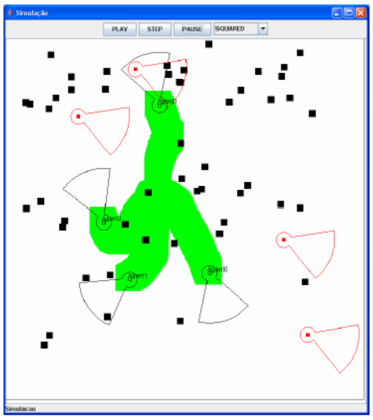
Multiagent exploration task in games through negotiation

Rodrigo Gabriel F. Soares [i] André M. C. Campos [ii]

[i] Federal University of Pernambuco
UFPE, CIn, PE-Brazil

[ii] Federal University of Rio Grande do Norte
UFRN, DIMAp, RN-Brazil



Screenshot of the multiagent exploration simulator. The thick darker area is the explored area, the cones represents the field of perception of the enemies and the agents of the exploration team.

Abstract

When a group of characters in a game aim to efficiently explore an environment, it is important that they coordinate their actions to cooperatively discover new areas. This paper tackles the exploration task as a multiagent problem in the context of computer games. Four simple strategies and an auction-based negotiation strategy were implemented and evaluated. Their performance was compared in different scenarios according to a set of metrics proposed in the paper. Then, it was possible to figure out an efficient strategy for random scenarios. A simulator has also been developed in order to perform the necessary tests.

Keywords: exploration, strategy game, negotiation, multiagent.

Authors' contact:
[i] rgfs@cin.ufpe.br
[ii] andre@dimap.ufrn.br

1. Introduction

Nowadays, computer games represent a fast-growing entertainment sector that requires the research for new techniques as well as the improvement of the existing ones. Among the several types of games, real-time strategy (RTS) games present a set of challenging problems for AI research. In recent year, several AI researchers have focused their work in those problems [Almeida et al. 2004; Madeira et al., 2004; Posen and Spronck 2004; Santana 2005]. Buro and Furtak enumerate the AI challenges for RTS games, citing problems like: adversarial real-time planning, decision making under uncertainty, opponent modeling and learning, spatial and temporal reasoning, resource management, collaboration and pathfinding [Buro and Furtak 2004]. Sempé also mentions a class of problem related to spatial coordination in games where he classifies the problems of foraging, formation displacement, covering and patrol [Sempé 2004].

The spatial coordination consists on spatially allocating agents on the environment in a synchronized way in order to optimize the resolution of a problem. Among the problems related to this kind of coordination, this paper tackles the exploration task. This problem is important in the game context since, in RTS games, it is necessary to discover resources or enemies around a region. This issue can be extended to other application domains like robotics, military simulations, search and rescue on sea, among others. In fact, several works have already been done in robotics dealing with the exploration task [Burgard et al. 2000; Zlot et al. 2002; Zlot and Stentz 2006].

The objective of this work is to approach the exploration task in the game context. The problem treated in this work is similar to the robot exploration. However, some characteristics in the game context make it a particular case. Agents in computer games usually deal with hostile entities and enemies units spread on the world. They need then to cooperatively explore the environment while avoiding undesired regions found out during the exploration. Taking into account the dynamic aspect of the hostile entities and enemy units, it is possible to address the exploration with dynamic undesired regions or static ones. This work addresses the static problem where the hostile regions are fixed and not previously known.

For this, it was initially implemented four simple reactive approaches. Those approaches were based on a planned coordination before execution and no planning communication during execution (therefore, indirect communication through the environment exists). After, a more robust negotiation-based mechanism (direct communication) of agent coordination was implemented and the results were compared to simple exploration mechanisms.

Before discussing about the implemented approaches, it is presented some related works in the next section. After, some metrics to evaluate the algorithms were introduced in order to compare those approaches. This section also describes some characteristics inherent to the problem dealt in this paper. The following section presents some multiagent coordination techniques to solve the exploration task. It is subdivided in simple reactive strategies and a strategy based on negotiation. Before concluding, the results of tests performed over those strategies are presented and commented.

2. Related work

Most of the terrain exploration approaches found in literature focuses on robot-based exploration. Balch and Arkin [1999] investigated the communication role in a set of common robots. Their conclusion was that communication is needless, if the robots store information about their movements throughout their paths along an area. Such conclusion leads us to try an approach without any communication among the agents: the first four approaches of this paper have this feature.

One of the exploration techniques is to slowly move the agents towards unexplored areas while trying to obtain detailed information about the terrain. The work presented by Iv [2002] showed that is possible to explore a region with minimum repeated coverage if the robots have a high degree of coordination. The robots move in a straight line parallel formation, if an obstacle is encountered, they break the formation in order to deviate it. The entities restore the formation in some point later. This kind of approach gets total terrain coverage, but it fails, if some robot does not succeed to complete its task, which is not appropriated for regions with hostile entities.

Rekleitis et al. [2001] presented another technique to this problem. A stationary robot visualizes the robots that sweep the terrain, inside its camera's field of vision. The obstacles are understood as blockages to its vision of the moving robots. There are always stationary robots that do not cover the area. If one robot fails, the others become useless.

The strategy proposed by Simmons et al. [2000] is based on frontiers search and uses an auction-based protocol. The robot evaluates a set of frontiers cells (known cells bordering unexplored areas) and determines the expected travel costs and information gain (estimated number of unknown map cells visible from that frontier). Then, the robots submit bids for each frontier cells. A central agent greedily assigns one task to each robot based on their bids. It is possible to get highly suboptimal results (as in many greedy algorithms) since the plans only consider the very near future. Such system is not totally distributed since it has central point of failure [Tanenbaum 1995]. If the central agent fails, the whole system also fails, and if some robot loses communication with the central agent, it ends up doing nothing. Such approach is not appropriate for games context due to a central control point which makes the exploring team vulnerable to enemies' attacks that may cause the entire team failure.

Yamauchi [1998] developed a completely distributed fault tolerant robot frontier-based exploration strategy. The robots share local sensor information in order to all robots produce similar frontier lists. Each robot moves to its closest frontier point, performs a sensor sweep, and broadcasts the resulting updates to the local map. Yamauchi's approach is completely distributed, asynchronous, and tolerant to the failure of a single robot. However, the amount of coordination is quite limited and thus cannot take full advantage of the number of robots available. For instance, more than one robot may decide (and is permitted) to go to the same frontier point. Since new frontiers generally originate from old ones, the robot that discovers a new frontier will often be the best suited to go to it (the closest). Another robot moving to the same original frontier will also be close to the newly discovered frontier. This can happen repeatedly; therefore, robots can end up following a leader indefinitely. A relatively large amount of information must be shared between robots. So, if there is a temporary communications drop, complete information will not be shared possibly resulting in a large amount of repeated coverage. Similar to the work by Simmons et al. [2000], plans are greedy and thus can be inefficient.

Zlot et al. [2002] also developed a completely distributed robot terrain exploration strategy. That work uses a market-based approach in order to achieve the robots' coordination. The robots continuously negotiate with one another, improving their current plans and sharing information about the covered and uncovered regions. Each robot has a set of goal points (tasks), each task is negotiated so that the most suitable robot to the task has that goal point in its list. Such strategy is appropriate to be extended to game context due to its fault tolerance and its success in experiments.

The strategy of Burgard et al. [2000] is probabilistic. It considers the travel costs and the utility of the goal points. When a goal point is assigned to a robot, the goal point's utility is decreased to the others robots. So, the system assigns different tasks to the robots. This approach succeeded in experimental results and may be adapted to games' context.

The presented approaches may be extended to the exploration task in strategy games. However the strategies having a central point of control might be inadequate when dealing with real-distributed games, since the loss of agents is frequent in strategy games. Most of the approaches use a discrete representation of the terrain (using cells). The work in this paper does not use grid cells, but a (pseudo) continuous area. The next section presents the characteristics of the problem dealt in this work.

3. The exploration task

The exploration task can be defined as the problem of discovering relevant information from an environment partial or completely unknown [Zlot et al. 2002]. However, the main issue in this problem is not only discovering the unknown environment, but also to minimize the time to discovery it [Burgard et al. 2000]. For an environment represented as a graph, this task is NP-Hard. For environments represented as a grid, it is always possible to implicitly handle the environment as a graph, just by thinking the grid cell as the graph nodes and the adjacent cells connected by the graph edges. Occupied cells (with obstacles) are not represented in the graph.

In the context of computer games, the exploration task is related to the problem of taking relevant information from a region before the opponent and so to plan the best strategy to win the game. Relevant information in this context is then the knowledge of areas where the agents can obtain resources (gains) and the areas where the hostile or opponent characters are situated, which can be static or dynamic. The problem addressed in this work considers: 1) unknown enemy's static positions, and 2) exploration priority to regions that are closer to the base.

3.1 Problem characterization

The problem tackled in this work has as objective function the maximization of the two predefined metrics presented in next subsection (3.2). Briefly, we attempt to cooperatively maximize the explored area in a given time t, and that the explored area be around the base.

For the problem characterization, it was also introduced some definitions and restrictions. Firstly, the agents described in this work are immersed in an environment that is:

- **Continuous:** agent movements are not tile-based, but defined as steering functions;
- **Partially observable:** the agents know only the explored area. In other words, the agents have a limited perception;
- **Deterministic:** there is no event other than the ones created by the agents. The environment does not randomly change during the simulation.
- **Sequential:** the next state of the agent is defined by the current state and the environment locally perceived.

At the beginning, the agent's base is randomly placed in the environment and all agents in the exploration team start from the base. The perception sensor in the agent is delimited by its field of view (FOV) and the area close by, representing the agent audio sensory system. The Figure illustrates the perceived area by an agent. It is important to point out that this field of perception is not related to the covering function presented in (Eq.2). It is mainly used for the fighting simplification, described in next paragraph.

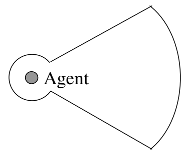


Figure 1: Schema representing the field of perception (FOV) of an agent.

Another simplification done was the fighting between enemies and agents of the exploration team. As the focus of this work is the exploration task (i.e. non-deterministic fighting is not treated here), it was considered two simple rules:

1. If an agent of the exploration team perceives an enemy unit before it (the enemy entered in the field of perception of the agent, but the agent did not entered in the enemy field of perception), the agent will avoid the area and will communicate to its colleagues that this area has enemy units. As consequence, all agents in the exploration team will avoid the area.
2. If an enemy unit perceives an agent of the exploration team before it, the agent is taken out from the simulation (i.e. it is considered that the enemy fought against the agent and won).

3.2 Exploration metrics

In order to evaluate the performance of the strategies for this problem, it is necessary to define some metrics taking into account the information gains and the costs associated to the task. It was defined two main criteria for comparing algorithms results:

1. **Explored area density**: this criterion focuses on how well explored is the area around the base of the agent team. Indeed, in a strategic game the base represents the region where the team must to take care in order to not be attacked. A bad exploration around the base might cache enemy units, causing, as consequence, undesired game results;
2. **Exploration quality**: This criterion tackles the issue of minimizing the distance traveled to explore a region. As we consider that an exploration agent will continuously keep exploring the environment while it is still alive, the exploration quality was defined as the ratio between the total explored area up to a time t and the sum of the distances traveled for all agents up to the time t. This metric evaluates the cost (distance) that the agent had to explore a region.

Another metric, the average covered area (the mean of the area that each agent has explored), could be defined in this work. However, the average and quality metrics are similar in a tile-based environment, since the agents will continuously travel with a fixed velocity. In a (pseudo) continuous environment, where the movements are based on steering behaviors, the distance traveled does not depend only on the number of agents, but also the environment itself. In other approaches, such function could be useful. So, we stick with the explored area density and exploration quality metrics.

Before formalizing the mentioned metrics, it is necessary to define some basic functions that will be used. Let us consider $\phi^a_x(t)$ and $\phi^a_y(t)$ to represent respectively the position of agent a over the x and y axis and $\delta(x,y)$ a function returning 1 whether the position (x,y) was not explored or 0 otherwise.

With such these functions, it is possible to present the first metric: the **Explored Area Density**, shown in (Eq.1). Considering *Ac* as the set of the active agents in the simulation (those which are still alive) and *dist*($\phi^a(t),B$) the distance from the agent *a* to the base (B) in time *t*, *r* is the distance of the farthest agent to the base in a given step *t*. In the density metric, *E* is the set of explored positions inside the circle of ratio *r* and center in the base (B), so *#E* means the numbers of elements in the set of explored area *E*. This metric show us how efficient is the spreading of the agents towards unexplored area: as previously mentioned, the area closer to the base has a higher exploration priority.

$$
\begin{aligned}
&Den(t) = \frac{\# E}{\pi r^2} \\
&\textit{Where}, \\
&r = D(t) = \max_{a \in Ac} \{dist(\phi^a(t), B)\} \\
&E = \{(x,y) \mid (x - B_x)^2 + (y - B_y)^2 \le r^2 \wedge \delta(x,y) = 0\}
\end{aligned}
\qquad \text{(Eq.1)}
$$

The second metric needs other basic functions. The first one is a covering function, which returns the unexplored area surrounding the agent. This area represents the region of perception of an agent. For simplicity, we have adopted a square area for the agent perception, resulting in the function on (Eq.2). In this function, *L* is the limit of the agent perception.

$$
v^a(t) = \sum_{i=\phi^a_x(t)-L}^{\phi_x(t)+L} \; \sum_{j=\phi^a_y(t)-L}^{\phi_y(t)+L} \delta(i,j) \qquad \text{(Eq.2)}
$$

With the agent perception function, it is possible to recursively define a function to return the explored area up to a given time *t* for an agent *a*, shown in (Eq.3). The function γ is the basis of the metrics proposed in this paper.

$$
\gamma^a(t) = \begin{cases} \gamma^a(t-1) + v^a(t) & \textit{if } t \ge 1 \\ v^a(t) & \textit{Otherwise} \end{cases} \qquad \text{(Eq.3)}
$$

Finally, the following function indicates the total explored area by the agents from the beginning of the simulation to the time t. The function is represented in (Eq.4), where *A* is the set of agents in the simulation.

$$
\Gamma(t) = \sum_{a \in A} \gamma^a(t) \qquad \text{(Eq.4)}
$$

Then, the **Exploration Quality** metric, can also be defined through the use of the total explored area. The quality is the ratio between the explored area and the sum of all distances traveled by the agents, which means that if a set of agents travels big distances to discover small regions, the quality is poor. The formulation is presented in (Eq.6). In the equation, $Dist^a$ (t) represents the distance traveled by an agent up to the time t and v is the velocity vector.

$$
\begin{aligned}
&Q(t) = \frac{\Gamma(t)}{\sum_{a \in A} Dist^a(t)} \\
&\textit{Where}, \\
&Dist^a(t) = \begin{cases} Dist^a(t-1) + |\vec{v}|, & \textit{if } t \ge 1 \\ |\vec{v}| & \textit{Otherwise} \end{cases}
\end{aligned}
\qquad \text{(Eq.5)}
$$

4. Exploration as a multiagent problem

As mentioned before, the exploration task is a sub-problem in the class of spatial coordination. Agents need then to coordinate their actions to avoid traveling in an area already explored by another agent. For this multiagent coordination task, it was proposed five strategies. Initially, four simple strategies were proposed and tested. They used a reactive-based approach to explore the environment. Aiming to improve the results, a deliberative-based approach was implemented. The agent deliberation is done by constructing its plan and combining it with the other agent plans through a negotiation mechanism.

The reactive and deliberative approaches are concerned only by the decision making layer of the agents, i.e. deciding where to explore. However, the displacement of the agents in all strategies uses a reactive approach based on the patterns of steering behavior proposed by Reynolds [Reynolds 1999]. The pattern used was *Obstacle Avoidance* for keeping away from obstacles and discovered enemies on the environment.

4.1 Simple approaches

The simplest approach to explore the environment would be to give to the agents a random function for their displacement. While it keeps the solution simple, there is no optimization for coordinating actions. Certainly, the performance of this approach evaluated by the proposed metrics (explored area density and exploration quality) would be very poor, and so, it was not considered in this study.

Some other simple reactive strategies that might optimize the explored area are related to group formation [Dawson 2002] and follows some concepts presented in [Iv 2002]. In this approach, the agents do not move randomly through the environment. Instead, they are inclined to follow a predefined way, keeping a formation with the other agents. Indeed, this pattern of displacement can be seen in human teams, when for instance, firemen need to search and rescue an individual lost on the sea. As in the real world, this does not mean that the agents know about the environment. They just adopt a strategy of displacement, but obstacles may force a specific agent to change its predefined way.

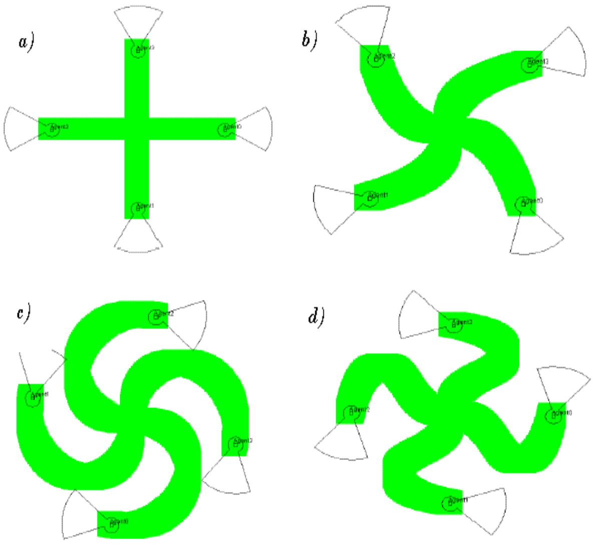


Figure 2: The visual representation of the four reactive strategies.

The strategies proposed in this work are denoted by functions that update the velocity vector of the agent (this vector is part of the agent steering behavior and also drives the agent direction). Each agent takes a different direction when leaves the base. For testing purposes, four formation-based strategies were considered. They are described below and illustrated in Figure 2.

a) **Straight line**: In this strategy the velocity vector of the agents is not altered. They follow a straight line while there is no obstacle on their path. In other words, only obstacles and discovered enemies might influence the agent path;
b) **Parabola**: Following this strategy, the agents are inclined to follow their path according to a parabola. They will then turn around the base continuously getting far from it step-by-step;
c) **Spiral**: In this strategy the agents will follow a spiral. It is similar to the parabola strategy at the beginning of the simulation. However it tends to be closer to base in a given time t compared to the parabola strategy;
d) **Sinusoid**: Differently of the parabola and the spiral strategies, following the sinusoid strategy the agent will not turn the base continuously. It will tend to follow a sine curve, small at the beginning, but growing when the agent gets far from the base.

The coordination mechanisms in these strategies are not dynamic. During the simulation, agents do not communicate to review their plans. This means that if the environment has several obstacles, the modification of the agent path will not be adjusted according to the areas already explored by other agents. An attempt to improve their results, the deliberative-based approach was proposed.

4.2 Resolution through negotiation

The exploration strategy proposed in this paper is based on some concepts proposed in [Zlot et al. 2002]. Differently from the original work, there is no centralized coordinator for the team. Instead, agents communicate between themselves without a predefined hierarchy, avoiding having a central point of failure [Tanenbaum 1995]. Another difference comes from the context of the application domain, i.e. computer games, like the ability to deal with undesired regions found out during the exploration (enemies).

According to the taxonomy proposed in [Wooldridge 2002], negotiation can be viewed as a class of the coordination through competition. Although the exploration agents are part of the same team, they compete between themselves for space that gives to them some benefits (profit). The more profitable an area is (unexplored area) the more attractive for the agents.

The main idea behind the strategy is that each agent defines a set of targets (positions) that it plans to explore in a sequence order, constituting the agent path. The waypoints of the agent path can be negotiated with another agent if the associated cost and benefits for the other agent present more profit. The route of the agents is continuously redefined through negotiation, as consequence, the team optimizes the explored area.

While the benefit in this approach is associated to information revealed during the exploration, the cost is related to the expense to get to the target. This includes the distance, the resources lost in the displacement (in some games, the agent might have limited resources) and the potential dangers in the path, like enemies. The benefit function R(x,y) of a target is then the quantity of unknown points around it. However, this function underestimates the real gain, since to get the agent close to the target it may discover new areas in the way. The cost function of a target is based on the distance to the target and the presence of enemies (already found by other agents) crossing the path to the target. If an enemy is on the path of an agent towards a new target, the target's cost associated to that agent is increased. However, if the same target is not risky to another agent (i.e., there are no enemies on its path towards its targets), the latter will have a higher priority than the mentioned agent to buy such target, because it will have a lower cost and so a higher profit. This way, the team attempts to avoid dangerous areas and unnecessary displacements by encapsulating the travel cost, revenue and risk of the areas in the profit function. For an agent, the real cost of a target $C_a(x,y)$ is the difference between the cost of the whole agent route with the target and the cost without it. Finally, with those functions, it is possible to define the profit function as shown in (Eq.6).

$$L^a(x,y) = \begin{cases} \dfrac{R(x,y)}{C^a(x,y)} & \text{if } C^a(x,y) > 0 \\ R(x,y) + \left|C^a(x,y)\right| & \text{Otherwise} \end{cases} \qquad \text{(Eq.6)}$$

Each agent tries then to maximize the quantity of information discovered while minimizes the cost in the travel. In this way, attempting to reach its own interests, each agent contributes for optimizing the exploration of the whole team, based on the negotiation of targets. However, up to now, nothing was mentioned about how the agents create their targets.

Zlot et al. [2002] proposed three approaches for the agents generating targets for the exploration task: 1) randomly; 2) a greedy approach; and 3) following a Quadtree. Among those strategies, the simplest one was chosen (randomly), since, according to Zlot et al., it presents a better performance than the greedy approach and equally efficient as the Quadtree.

In our problem, it is possible to get an unreachable target due to the obstacles in the environment. It is then necessary to provide a way to reevaluate the agent commitment to its target [Wooldridge 2002]. Since the environment is not previously known, usually it is not possible to know when a target in unreachable. Then, the agent will keep trying to get to the target while the distance traveled does not exceed a threshold. When this happens, the agent takes the target out of its waypoints and goes to the next target. Anyway, the traveled path might be precious for the agent since new areas might be discovered in the meanwhile.

At the beginning of the simulation, every agent is in the team base and they generate randomly a fixed number of targets in the environment. The targets are added in the agent path following a greedy strategy in order to define the sequence of targets that minimizes the path cost. With the first target defined, the agent will try to negotiate its target with other agents in an auction-based negotiation until all its targets are offered. When the agent offers a target to negotiation, it also informs the minimum required price for it. The highest bidder buys the target adding it to its route. If no bid is made higher than the minimum price, the target is kept with the first agent. The auction for this negotiation is the FPSB (*First Price Sealed Bid*) [Wolfstetter 1996], which means that each buyer make their bid in a single turn without knowing the value offered by the other ones.

The agent will try to negotiate all their targets. Once all targets are auctioned (probably it will keep some of them), the agent travels for the first target in its path. When an agent reaches a target or when all targets were sold, it generates a new set of targets, makes a new auction, orders the left targets and goes to the first one. The number of targets generated by an agent depends on the current number of target it already has in its path. An agent with few targets will generate more targets than an agent with a lot of targets in its path. Then, the agents have tours on the terrain that represents their auction-based coordination: each target is assigned to the agent that maximizes its utility by adding such target in the agent's tour.

5. Results

In order to put the described ideas on practice and evaluate their results, a simulator was developed. The simulator has as main aim to support the study, in an empiric way, the suitableness of different strategies of multiagent coordination in the exploration task.

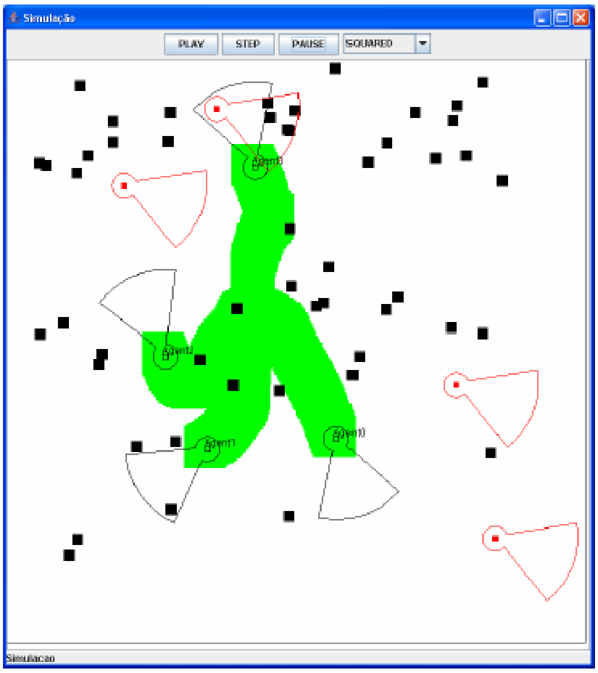


Figure 3: Screenshot of the simulator. The thick darker area is the explored area, the cones represents the field of perception of the enemies and the agents of the exploration team.

In order to evaluate the strategies, eight scenarios were defined. The first three scenarios has just one enemy on the environment, but the first one has no obstacles, and the second and third scenario, we augmented the number of obstacle for twenty and forty, respectively. In the next three scenarios (4th, 5th and 6th) we tested the strategies augmenting the number of enemies in the previous scenarios (1st, 2nd and 3rd) to four, evaluating the efficiency of the strategies in a presence of more enemies. Usually, RTS games move their units following a group formation. So, in the 7th scenario, we tested the strategies in a environment where the enemies are not randomly placed, but have a group formation (they make a square block, all of them point to the same direction). The 8th scenario, taking into account random environments, is described on 5.2.

5.1 Pre-defined scenarios

For each pre-defined scenario, several simulations limited to 500 iterations were executed in an environment defined by a region of 700x700 pixels and a perception area for each agent of 50x50. At the end of the simulation, the metrics of density and quality was evaluated and are shown in the Figures 4 and 5. The strategies in the figures are represented as *numberOfObstacle_numberOfEnemies* and the scenario where the enemies are in formation is represented by *40_4_form*.

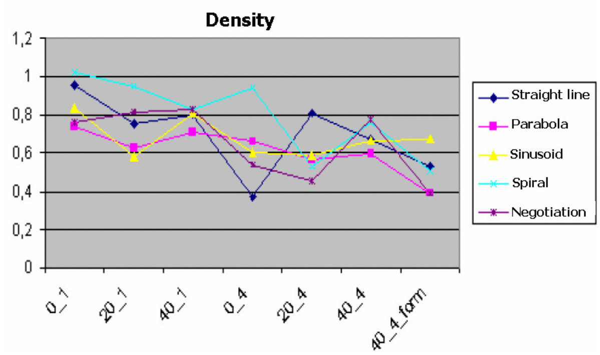


Figure 4: Density chart for the scenarios (represented in the X axes).

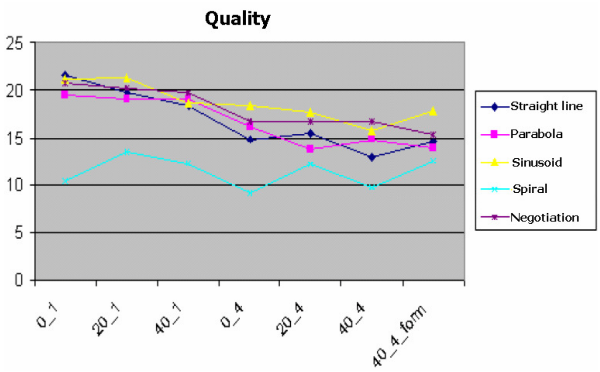


Figure 5: Exploration quality chart for the scenarios (represented in the X axes).

The metric of density (Figure 4) gives some feedback about how well explored is the area around the base, i.e. it evaluates the explored area defined by a circle which radius is the distance of the farthest agent. It is observed that the Spiral strategy is, in general, better than the other ones, which makes sense since the agents will turn around the base and this area becomes very well explored (this strategy seems then be useful for the problem of discovering a specific target close to the base). It is also important to observe in the chart that, even being more complex than the others, the negotiation strategy does not have a better performance. Indeed, the implemented negotiation-based strategy does not take into account the distance from the base.

The last metric, measuring the quality of the exploration (Figure 5), showed that the Sinusoid strategy had the best performance, the Spiral one had the least exploration quality and the others had similar results.

Analyzing those results, it was possible to understand the intrinsic mechanisms of the emergent behavior of the reactive approaches comparatively with the negotiation-based approach. Although its simplicity, the Sinusoid strategy seemed to be the better approach for the results. However, in order to generalize the problem for any environment, with variable enemies and obstacles (number and size), it was defined a random scenario, described here after.

5.2 Randomly generated scenarios

Several scenarios (Figure 3 illustrates one of them) were also tested in the simulator in order to evaluate the previously mentioned strategies. The randomly generated scenarios were based on the following specification: 1) the obstacles size was defined in a range of length and width in the interval [1,200]; 2) The position of obstacles and enemies were completely inside the environment; 3) The number of obstacles in the range of [0,40], and; 4) The number of enemies in the range of [0,8].

A sample of one thousand random scenarios for each strategy was generated. Among these scenarios, a set of tests was configured separating tests running with 300 and 500 iterations, and tests where the exploration team has 4 agents and 8 agents. Then, a set of 250 tests of each combination of these two variables was prepared. The number of exploring agents was fixed for better evaluate the performance of the strategies with an already known number of agents. The number of iterations was chosen as an attempt to estimate the performance of the algorithms in a short time and another with a longer time. The results are presented in Table 1.

Table 1. Average (Av.) and Standard Deviation (S.D.) of 1000 running tests for each strategy evaluating the metrics and the percentage of the total area explored at the end of the simulation

Strategy	**% Explored**		**Quality**		**Density**	
	Av.	**S.D.**	**Av.**	**S.D.**	**Av.**	**S.D**
Straight line	0.337	0.182	15.256	8.701	0.342	0.290
Parabola	0.417	0.178	18.774	8.770	0.292	0.252
Sinusoid	0.441	0.188	19.918	9.165	0.335	0.265
Spiral	0.333	0.144	15.353	7.981	0.477	0.356
Negotiation	0.497	0.174	22.621	8.826	0.333	0.292

In order to evaluate if the difference between the averages of distinct samples is statistically significant, the Hypothesis Test [Mitchell 1997] was performed. According to the test, the negotiation strategy obtained statistical results significantly better than the other strategies in the quality metric. However, for the density metric, the spiral strategy was the best one. Although there exist differences between the averages of the straight line, sinusoid and negotiation strategies in the density metric, such a difference is not significant from the statistical point of view. This means that these strategies can be considered as with equal performance in the density metric. Using the hypothesis test in the percentage of known area after all the iterations, it was also possible to validate that the negotiation was the strategy most suited to the several scenarios generated. Indeed, it tries to minimize the exploration in an already known area. This results was expected since, the other strategies do not deal with this objective. On the other hand, the spiral strategy presented a better performance in the density metric. In fact, while the spiral strategy was conceived to focus the exploration close to the base, the negotiation strategy does not care about the base. The agents are driven to explore any undiscovered area, regardless where it is.

The enemies make the scenario harder to the agent team. The presence of a large number of enemies causes loss of exploration quality, such effect is stronger in the reactive strategy. This is due to the lack of communication about the enemies positions. As shown, the negotiation-based approach is more robust to this type of scenario.

6. Conclusions

This paper presented an auction-based negotiation strategy for a group of agents to explore an unknown area. This strategy was compared to simpler reactive approaches according the two metrics: the explored area density and exploration quality. The results showed that using the proposed negotiation strategy would improve exploration, except when it is desired to enhance the explored area density. The spiral approach showed better results.

Depending on the game requirements, a simple strategy may be suitable. It might be a group formation approach, as presented in the paper, or a steering behavior like flocking [Reynolds 1999]. However, when the explorer team has a more significant role in the game, the negotiation strategy is recommended.

This work tackles an instance of the multiagent exploration task, conceived in order to simulate strategy games scenarios.

A multiagent simulator was developed in order to perform the experiments and to work as a framework for further simulations. As a continuation of this work, several other strategies may be implemented and compared, according to the metrics, to the algorithms proposed here.

References

ALMEIDA, A., RAMALHO, G. L., SANTANA, H. P., TEDESCO, P., MENEZES, T. R., CORRUBLE, V., CHEVALEYRE, Y., 2004. Recent Advances on Multi-Agent Patrolling. In *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence*, São Luis, MA, Brazil. Lecture notes on Artificial Intelligence 3171, Springer-Verlag., pp.474-483.

BALCH, T.; ARKIN, R. C. 1999. Behavior-based formation control for multi-robot teams. In: *IEEE Transactions on Robot and Automation*., pp.926-939.

BURGARD, W., MOORS, M., FOX, D., SIMMONS, R., THRUN S., 2000. Collaborative multi-robot exploration. *Proceedings of the IEEE International Conference on Robotics and Automation*. San Francisco, CA, April., pp.852-858.

BURO, M., FURTAK, T., 2004. RTS Games and Real-Time AI Research. *Proceedings of the Behavior Representation in Modeling and Simulation Conference* (BRIMS), Arlington VA., pp.27-34.

BURO, M., 2003. Real-time strategy games: A new AI research challenge. *Proceeding of the 18th International Joint Conference on Artificial Intelligence*. Acapulco, Mexico. pp.1534-1535.

DAWSON, C., 2002. Formation. In *AI Game Programming Wisdom*. Steve Rabin (Eds.) Charles River Media Ed., pp.272-281.

HUHNS, M., STEPHENS, L., 1999. Multiagent systems and society of agents. In *Multiagent systems: a modern approach to Distributed Artificial Inteligence*, Weiss, G. (Eds.), pp.79-120.

IV, D. L. 2002. Towards sensor based coverage with robot teams. In: *Proceedings of the IEEE International Conference on Robotics and Automation*., pp.961-967.

MADEIRA, C., CORRUBLE, V. RAMALHO, G., & RATICH B., 2004. Bootstrapping the learning process for the semiautomated design of challenging game AI. In D. Fu & J. Orkin (Eds.) *Challenges in Game AI: Papers of the AAAI'04 Workshop* (Technical Report WS-04-04). San José, CA: AAAI Press.,pp.72-76.

MITCHELL, T., 1997. Machine Learning. In *Machine Learning*. McGraw-Hill, chap.6.

PONSEN, M., & SPRONCK, P., 2004. Improving adaptive game AI with evolutionary learning. *Computer Games: Artificial Intelligence, Design and Education*. Reading, UK: University of Wolverhampton Press, pp. 389-396.

REKLEITIS, I. M.; DUDEK, G.; MILIOS, E. E. 2001. Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, v.31, n.4, pp.7-40.

REYNOLDS, C., 1999. Steering behaviors for autonomous characters. In *Proceedings of the Game Developers Conferenc*e. San Jose, CA, pp.763-782.

SANTANA, H., 2005. *Patrulha Multi-agente com aprendizagem por reforço.* Master thesis. CIn UFPE, Recife, Brazil.

SEMPÉ, F., 2004. *Auto-organisation d'une collectivité de robots: application à l'activité de patrouille en presence de pertubations*. Doctoral Thesis. Laboratoire d'Informatique de Paris IV, Paris, France.

SIMMONS, R. et al. 2000. Coordination for multi-robot exploration and mapping. In: *Proceedings of the National Conference on Artificial Intelligence*. AAAI., pp.852-858.

TANENBAUM, A., 1995. *Distributed Operating Systems*. Prentice-Hall .

WOLFSTETTER, E., 2002. Auctions: an introduction. In *Journal of Economics Surveys*. Vol.10, Num.4, pp.367-420 (1996).

WOOLDRIDGE, M., 2002. *An introduction of multiagent systems*. John Wiley & Sons.

YAMAUCHI, B. 1998. Frontier-based exploration using multiple robots. In: *AGENTS'98: Proceedings of the second international conference on Autonomous agents*. New York, NY, USA: ACM Press, p. 47–53.

ZLOT, R., STENTZ, A., DIAS, M. B., AND THAYER, S., 2002. Multi-Robot Exploration Controlled by a Market Economy. *Proceedings of the IEEE International Conference on Robotics and Automation*. Washington - DC, May., pp.3016-3023.

ZLOT, R., STENTZ, A., 2006. Market-based Multirobot Coordination for Complex Tasks, *International Journal of Robotics Research Special Issue on the 5th International Conference on Field and Service Robotics*, Vol. 25, No. 1., pp.73-101.