Extending the RoboCup Rescue to Support Stigmergy: Experiments and Results

Gabriel Rigo da Cruz Jacobsen [1]
Carlos Alberto Barth [1]
Fernando dos Santos [1]

Abstract: Social insects have inspired computer scientists and engineers to develop models for coordination and cooperation in multiagent systems. One example of these models is the model of stigmergy. In this model agents use indirect communication (communication through the environment) in order to coordinate actions. The RoboCup Rescue simulator is used as a testbed to evaluate this model in a real world considering a highly constrained scenario of an earthquake. This paper investigates the feasibility of using stigmergy in the RoboCup Rescue by extending that environment in order to figure some improvements of performance that can be obtained. We compared the results of a multiagent system that uses stigmergy against two other approaches: a multiagent system that uses a greedy strategy and no communication, and a multiagent system where agents communicate via direct messages. Experimental results shown that the use of stigmergy leads to an improvement on agents' performance by 9.02% to 38.6% when compared to the system with no communication and can be statistically equivalent to the system which uses direct messages, depending on the scenario.

1 Introduction

An agent is a computational system situated in an environment, being capable of acting in an autonomous fashion to accomplish its own goals [17]. A multiagent system is composed by agents that can interact through coordination, cooperation or negotiation, in order to reach global goals [7]. A lot of approaches for coordination, cooperation or negotiation are available in the literature. Ideally, those approaches shall be evaluated on real world scenarios to check their effectiveness. The RoboCup Rescue (RCR) [8] was created for this purpose.

In the RCR simulator the agents were introduced in a catastrophic environment (earthquake), and should mitigate the situation in order to minimize the material and human damages. The simulator replicates a highly constrained environment regarding traffic and inter-agent communication. An example of these constraints is the radio channel that is a limited resource where a maximum number of bytes can be sent at once. These restrictions impose

[1] Universidade do Estado de Santa Catarina (UDESC)
gabrielrigoj@gmail.com; carlos.barth@udesc.br; fernando.santos@udesc.br

the necessity of an effective use of available resources. So, despite the ability of agents to communicate among themselves via radio channels, in some cases there would be no way to do interactions between agents. Either because the communication system is broken, or because the component responsible for communication is damaged.

Social insects had inspired computer scientists and engineers to develop approaches for coordination and cooperation. The book of Bonabeau et al. [1] presents a review of some computational models created from observations on social insects. One of those models, which is particularly interesting to this paper, is the model of stigmergy [2]. Through stigmergy, the insect colony reaches self-organization with no direct interactions among their individuals. All the interactions are done by indirect communication, through the environment, using pheromones.

We have not found any prior study regarding the use of stigmergy in the RCR environment. The available version of the RCR does not allow indirect communication between agents through stigmergy. So, this paper investigates three hypotheses: a) it is possible to use stigmergy in the RCR environment; b) the performance of agents is improved using stigmergy and; c) the stigmergy model can be as efficient as a direct message approach.

We extended the RCR to enable the use of stigmergy by the agents. So, we performed a set of experiments to verify: a) the feasibility of stigmergy in the simulator and; b) differences of performance comparing three approaches for the agents: application of stigmergy, direct communication and a greedy algorithm. The results showed that the use of stigmergy leads to an improvement on agents' performance by 9.02% to 38.6% if compared to agents without communication and can be equivalent to the direct communication system, depending on the scenario.

This paper is organized as follows. Section 2 presents the required background on swarm intelligence. Section 3 presents the RCR simulator. Section 4 presents related works. Section 5 describes the extension proposed to the RCR simulator to enable the use of stigmergy. Section 6 presents the empirical evaluation via a set of experiments. Finally, section 7 describes the conclusions and future works.

2 Swarm Intelligence

Social insects, like ants and termites, organize themselves in colonies. Despite the simplicity of each individual, the colony as a whole is able to deal with complex problems, like the cooperative transport of prey. According to [1], the collective activities of social insects are self-organized. The complex collective behavior may emerge from inter-actions among individuals that exhibit simple behavior, in a flexible and robust way. These abilities were the base to the development of computational models used to build swarm intelligent agents and systems.

Deneubourg et al. [2] developed a model of stigmergy, a phenomenon observed in some species of social insects. In such phenomenon, the colony reaches self-organization with no direct interactions among the individuals. All the interactions are done by indirect communication, through the environment, using a pheromone trail. The pheromone trail stimulates the individuals, which take certain actions in response to these stimulus.

Stigmergy is observed while ants are foraging. Initially there is no pheromone in the environment. As a result, the ants take random paths to search for food. If an ant finds a source of food, this individual moves back to the nest, laying a pheromone trail while walking. When the nestmates sense the pheromone, they are stimulated to follow it. As a result, they are led back to the source found earlier by some individual. After a while, a lot of individuals are engaged in the food transportation following the shortest pheromone path between the nest and the source of food. Dorigo et al. [3] used the model of stigmergy to solve the classical traveling salesman problem.

The model of [2] says that the decision of an ant to follow some path is probabilistic, and take into account the number of ants that already followed a certain path. In other words, given two paths A and B, after i ants followed some of these paths, there will be A_i pheromone units on path A and B_i on path B. The next ant $i+1$ chooses path A or B with probabilities $prob_A$ and $prob_B$, depending on A_i and B_i, as shown in equation 1.

$$prob_A = \frac{(k+A_i)^n}{(k+A_i)^n + (k+B_i)^n} \qquad\qquad prob_B = (1 - prob_A) \tag{1}$$

The higher is the value of A_i, the higher is the probability of choosing the path A. The parameter n is used to specify the degree of nonlinearity: when n is high, the path that contains the higher amount of pheromone will have the higher probability of being chosen. The parameter k quantifies the degree of attraction of an unmarked path: when k is high, a higher amount of pheromone is necessary to make the choice nonrandom.

Every ant that passes on a path already marked with pheromone marks it again to reinforce the stimulus on its nestmates. Thus the amount of pheromone present on some path depends directly on the number of ants have been passing on that path. Given that $\tau_A(t)$ represents the amount of pheromone on a path A at a time instant t, the amount of pheromone in the next time instant $t+1$ can be expressed by equation 2 [3].

$$\tau_A(t+1) = \rho * \tau_A(t) + \Delta\tau_A(t, t+1) \tag{2}$$

where $\Delta\tau_A(t, t+1)$ is the amount of pheromone dropped between the time t and $t+1$ by every ant i that passed on the path A, as show in equation 3.

$$\Delta\tau_A(t, t+1) = \sum_i \Delta\tau_A^i(t, t+1) \tag{3}$$

The value of ρ represents the coefficient of pheromone evaporation. By the time ants give up passing on some specific path (i.e. the food at the source is over), the pheromone evaporates as the time evolves. The lower is the amount of pheromone on some path, the lower is the stimulus to the ants to pass on it again. In that sense, the evaporation avoids the existence of paths that lead to nowhere, or suboptimal paths. According to [3], $0 \leq \rho < 1$, and its value can be experimentally set in a way that gives the best results to the system.

3 RoboCup Rescue Simulator

The goal of the RoboCup Rescue (RCR) Simulation League [8, 16] is to provide a testbed for simulated rescue teams acting in situations of urban disasters. Currently, the RCR simulator tries to reproduce conditions that arise after the occurrence of an earthquake in an urban area, such as the collapsing of buildings, road blockages, fire spreading as well as buried and/or injured civilians. The simulator incorporates some collaborative agents acting to mitigate the situation. Heterogeneity, limited information, limited communication channel, planning, and real time are issues that characterize this as a complex multiagent domain [8]. The RCR aims at being an environment where multiagent techniques that are related to these issues can be developed and benchmarked.

In the RCR, the main agents are fire brigades, police forces and ambulance teams. Agents have limited perception of their surroundings and can communicate by radio channels, but are limited on the number and size of messages they can exchange. Regarding information and perception, agents have knowledge about a map that describes the environment and the paths between all pairs of distinct locations. This allow the agents to realize possible paths from a current location to a given place (e.g. a refuge).

The architecture of the RCR simulator is presented in the Figure 1. It defines a main component, called kernel, and a set of 6 other components. Each of these component simulates an aspect of the scenario. As an example, the fire simulator simulates fire ignition and propagation over the buildings. The role of the kernel is to control the simulation, invoking every specific simulator as needed.

Each simulation follows a sequence of 300 timesteps. On each timestep, every agent must decide what action it will take, given its own perception. This perception is composed of objects representing the environment (e.g. a building object, a road object, etc). For each object, the agent is able to access its properties (e.g. whether a building is on fire, amount of blockages on the road, etc). When the decision is taken, the agent sends a command to the

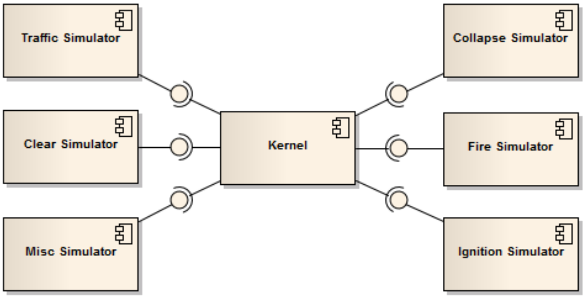


Figure 1. Architecture of the RCR, showing the components of the simulator.

kernel with the chosen action. As an example, consider that the chosen action is to move in the environment, then the agent will send a *move* command with the path it wants to move.

One important issue to be understood in this paper is the traffic simulator component, which is responsible for moving the agents in the map. The execution of this simulator follows a sequence of steps. Initially, the traffic simulator initializes the traffic parameters (i.e. velocity of agents). After that, it starts a loop that is repeated until the end of the simulation. On each step of that loop, the simulator processes each locomotion command received from all the agents, taking two actions: (1) moving the agent through the map and (2) updating position properties in the worlds' representation.

To measure the performance of the agents, the RCR defines a score. The score takes into account a relation between the building area left undamaged and the initial building area. When there are civilians to be rescued, the score also considers a relation between the health condition of all civilians at the beginning and end of the simulation.

4 Related Work

Regarding the use of swarm intelligence and stigmergy in robotics and multiagent systems, the work of [10] applies swarm intelligence to a multiagent system in a real constrained world. The use of stigmergy (through virtual pheromones) proved to be a useful strategy for reducing the communication overhead between robots. Hoff et al. [6] use stigmergy as a message protocol, measuring the performance of the swarm in environments that can present some sort of obstacles to the robots or not. The work of [15] focuses on the capability of the robots in perceiving the environment as a way of making decisions, considering collaborative swarm of robots through stigmergy. In [13] the authors intend the implementation of the necrophoric behavior of the bees as a way to give the robots of the swarm the capability of recognizing and rescuing a disabled robot. Payton et al. [11] present a swarm of robots acting in a rescuing scenario, on which virtual pheromones are used as a strategy of communication and coordination. In [12], the authors use a swarm of robots for checking emergency infrastructure in order to anticipate the restoration of this infrastructure as well as the community functions. They propose that a decision support system can receive feedback of a swarm of robots specialized in inspecting infrastructure in a disaster scenario using stigmergy, collaborating to life maintenance efforts. In the work of [9], a swarm of robots serves as guidance to human firefighters in large critical scenarios where visibility is poor due to the smoke. The purpose is to give support when the navigation of the human being is compromised in some level.

Regarding RCR, an efficient coordination amongst agents is a critical factor given the scenario characteristics and agents' limited capabilities. Swarm techniques have been applied to get coordination. Ferreira Jr. et al. [4, 5] presented Swarm-GAP, a multiagent algorithm for task allocation based on the model of division of labor in insect colonies. Santos and Bazzan [14] proposed *eXtreme-Ants*, also a multiagent task allocation algorithm which uses both the model of division of labor and the model of recruitment for cooperative transport. The recruitment model is used in *eXtreme-Ants* to deal with tasks which need a specific number of agents engaged simultaneously to be accomplished. A drawback of both Swarm-GAP and *eXtreme-Ants* is the use of the communication channel to establish the agents' coordination.

The available version of the RCR simulator does not allow an agent to use indirect communication, via stigmergy (dropping and sensing pheromones). Thus, to the best of our knowledge, there is no results about the performance of a team of agents which uses stigmergy in this environment. This paper investigates this issue. The simulator was improved to allow stigmergy and a set of experiments were performed to verify some changing in agents performance.

5 Stigmergy in the RoboCup Rescue

In order to use stigmergy in the RCR, we extended the simulator to meet the following requirements:

(i) every road A must have a way to store an amount of pheromone τ_A;

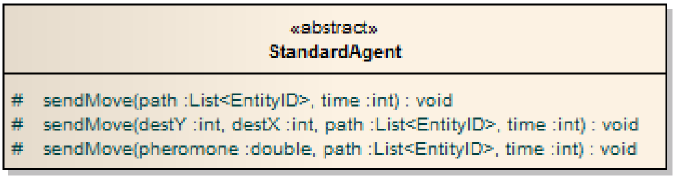
(ii) it must be possible to an agent i to perceive the pheromone τ_A stored on a road;

(iii) it must be possible to an agent i to drop certain amount of pheromone $\Delta\tau_A^i$ on a road;

(iv) the stored pheromone on a road must evaporate gradually, given an evaporation coefficient ρ .

In the available version of the RCR simulator every road A has a set of properties which give information about itself. In the extended version, a property called *pheromone* stores the amount of accumulated pheromone τ_A on the whole road. The agents can access the entire set of the road properties, by that way the requirements (i) and (ii) were accomplished.

In the RCR, the `StandardAgent` is the base agent class that must be extended in order to develop an agent. The base class agent provides a set of methods that will allow the agent to interact with the environment (e.g. to fight a fire, to rescue a civilian, etc). One of these methods is the `sendMove`, which moves the agent through some path. In order to meet the requirement (iii), we extended the base agent, overloading the `sendMove` method with an additional parameter *pheromone*. This parameter is the amount of pheromone $\Delta\tau_A^i$ the agent intends to drop on some road while moving. Figure 2 presents a simplified class diagram with the overloaded `sendMove` method.



```
#   sendMove(path :List<EntityID>, time :int) : void
#   sendMove(destY :int, destX :int, path :List<EntityID>, time :int) : void
#   sendMove(pheromone :double, path :List<EntityID>, time :int) : void
```

Figure 2. Simplified class diagram of the base class agent. An overloaded `sendMove` method was created in order to enable the extended agent to drop pheromones on the roads.

The last requirements (iii) and (iv) are accomplished by making changes in the traffic simulator. The actions performed by the extended simulator are represented in the diagram shown in Figure 3. The original actions of the base traffic simulator (as described in the section 3) are represented with rounded edges. The actions that were added are the ones with straight edges. When the iteration starts, the first action performed is the pheromone evaporation. The evaporation is based on an user-defined evaporation coefficient ρ obtained from the *traffic3.cfg* configuration file. The extended traffic simulator decreases the amount of pheromone τ_A for all *Road* object, as given by the equation 2. Another feature in the extended traffic simulator is the capability to process pheromone dropping events. Locomotion commands are received from the agents and the increasing amount of pheromone $\Delta\tau_A^i$ changes the total pheromone τ_A in the paths according to the equation 2. By doing these modifications, the requirements (iii) and (iv) were accomplished.

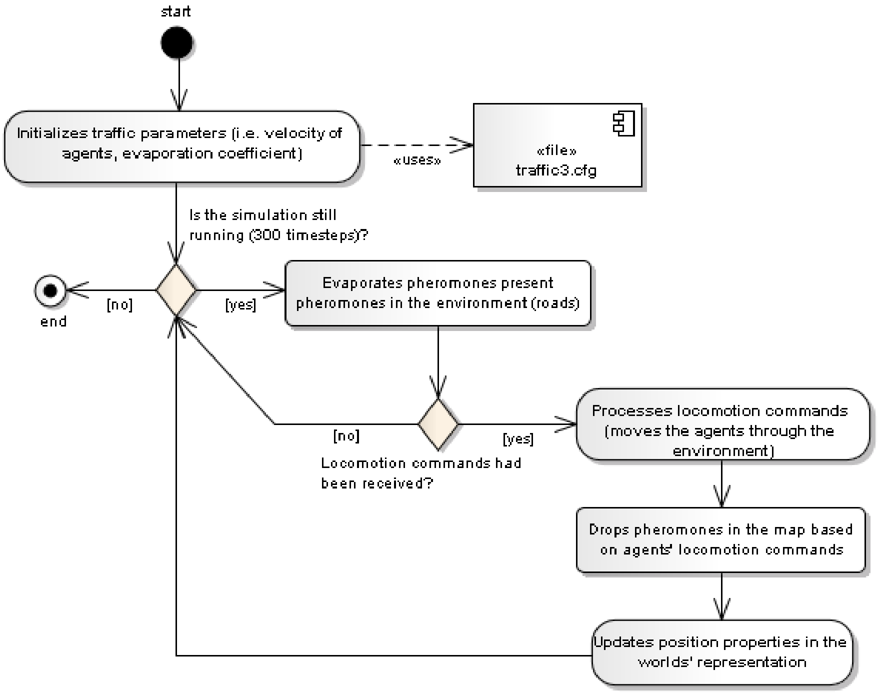

start
Initializes traffic parameters (i.e. velocity of agents, evaporation coefficient)
«uses»
«file»
traffic3.cfg
Is the simulation still running (300 timesteps)?
[no]
end
[yes]
Evaporates pheromones present pheromones in the environment (roads)
[no]
[yes]
Locomotion commands had been received?
Processes locomotion commands (moves the agents through the environment)
Drops pheromones in the map based on agents' locomotion commands
Updates position properties in the worlds' representation

Figure 3. Diagram showing the changes done in the traffic simulator to give support to pheromone dropping. The original actions of the base traffic simulator are represented with rounded edges. The actions that were added to support pheromone dropping are the ones with straight edges.

6 Experiments and Results

To investigate our first hypothesis, which claims that it's possible to use stigmergy in the RCR, we performed a set of experiments using the *Kobe4* map[2]. Figure 4 presents the map used in the experiments. The map is composed by 736 buildings and 1515 roads.

In the performed experiments, we decided to disable all types of agents except for the fire brigade agents. This decision is due to the similarity between the fire-fighting actions and the ant-foraging. So, in the sense of this research, the behavior of that sort of agent has high cohesion to our purpose. Regarding the movement of the group of agents in the environment, both tasks are based on the same main lines: first the agents explore the environment looking

[2] The *Kobe4* map is used in competitions of the *RoboCup Rescue Simulation League*.

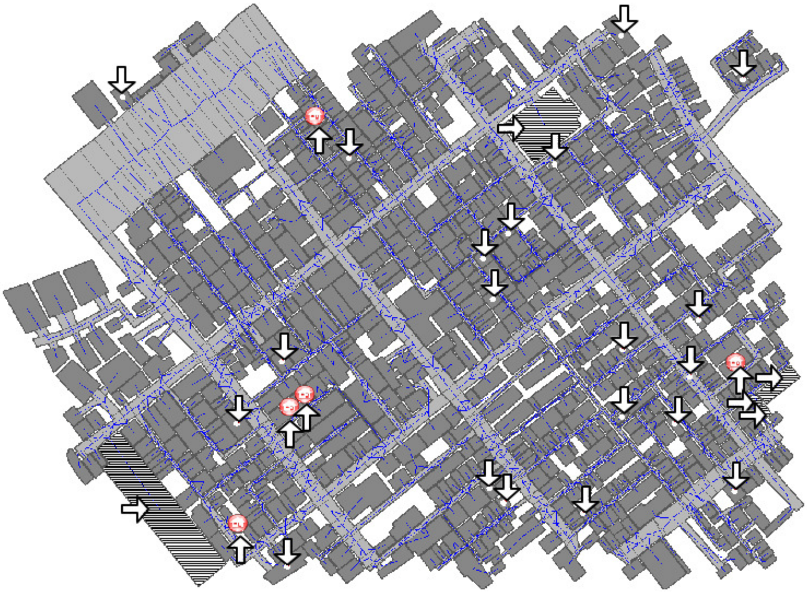


Figure 4. The *Kobe4* map used in the experiments with 5 refuges. The small circles (pointed out by up-to-down arrows) represent the initial position of the fire brigades. The big circles (pointed out by down-to-up arrows) represents the refuges, in which fire brigades can refill their water tanks. Striped rectangles represent the five initial fire spots (pointed out by left-to-right arrows).

for some source; if some source is found, a sign is sent to the other agents expecting some cooperation.

Very alike to ants looking for a food source, the fire brigade agents should explore the environment looking for buildings on fire. If the water of its tank ends while fighting a fire, the fire brigades should go back to a refuge to refill, in the same way the ants transport the food to the nest. Since each agent has knowledge about the map and its roads, it becomes possible to calculate the shortest path to the refuge. When traveling back to a refuge, each agent drops an amount of pheromone on the roads while passing. When the water tank is full, each agent that leaves the refuge is stimulated by the pheromone trail, according to the equation 1. This fact increases the tendency of the agent to move to a fire and continue fighting it, similarly to the ants that leave their nest and are stimulated to move to a food source.

To prevent the agents from getting lost, it was developed the following strategy: when an agent is walking for some time without sensing any pheromone trail, the agent must go back to some refuge. Since the agents drop pheromone while going back to a refuge, the probability of finding paths with unevaporated pheromone around the refuge is high. The experiments were made disabling blockages on the roads and civilians. The blockages were disabled because it would be necessary to use the police force agents to clear the roads, and they were disabled for our experiments. The civilians were disabled to measure the score based only in the fire brigade actions.

The experiments were performed on four scenarios: 20 agents plus 1 refuge (20x1), 20 agents plus 5 refuges (20x5), 40 agents plus 1 refuge (40x1), and 40 agents plus 5 refuges (40x5). On each scenario, the following values were used for $\Delta\tau_A^i$ (amount of pheromone dropped by each agent on a piece of road): 0 (no pheromone), 1, 5, and 10 pheromone units. The evaporation coefficients ρ used were 0 (no evaporation), 0.25, 0.50, 0.75, and 1 (total evaporation). To measure the performance of the agents, we use the building area that was left (e. g. after an earthquake followed by a fire and the intervention of the fire brigades). The parameter n of the equation 1 was set to 2, accordingly to the best value proposed by [2]. The parameter k of the same equation 1 was set in 0 to give total emphasis to paths with unevaporated pheromone. All data were averaged over 10 runs of the simulator. For the sake of completeness we show averages (and standard deviations) for all scenarios in appendix A (Table 1), where grey cells indicate the best score in each scenario.

Figure 5 shows the results obtained with 20 agents. As we can see, when the scenario has only one refuge, the score is improved when compared to a scenario with five refuges. With one refuge, all the agents move back to refill their tanks, increasing the amount of pheromone in the environment. So, the stimulus on the agents which are leaving the refuge is increased. This leads to a big concentration of agents surrounding the refuge and cooperatively fighting a fire spot, improving the performance. With five refuges, the agents are divided into small groups surrounding each refuge, decreasing the cooperative effort.

Figure 5. Results comparing fire brigades using stigmergy in the 20x1 (transparent) and 20x5 (filled) scenarios using different values of pheromone and evaporation.

In both 20x1 and 20x5 scenarios, there is no statistical difference (t test, 95% confidence) in the scores using 1, 5, or 10 pheromone units. This means that all of these scores were equivalent independently of the coefficient of evaporation (e.g. the scores with 1 pheromone unit are equivalent in all of the evaporation coefficients for each scenario).

In the 20x1 scenario, it is possible to see that when the agents drop some amount of pheromone, and the evaporation is lesser than 0.75, the results are better than using no pheromone (i.e. with an evaporation coefficient of 0.5, the use of 10 pheromone units gives a higher score than 0 pheromone units). Since the use of an evaporation coefficient of 1 (total evaporation) is equivalent to using no pheromone, this improvement was not observed in the case of total evaporation. In the 20x5 scenario the results were all equivalent. A possible explanation for that behavior is the fact that the agents were very spread over the map. Therefore, the agents weren't able to handle the situation.

Figure 6 presents the results obtained with 40 agents. The conclusions regarding the number of refuges were not observed here. In the most of the scores, there is no statistical difference when comparing 1 and 5 refuges. A possible cause is the small quantity of tests that were performed, given that the standard deviation is too high in this map. However, in the scenario with one refuge, there is a promising behavior with evaporations 0.25 and 0.5. As we can see, the result using some amount of pheromone is better than using no pheromone. We believe it is possible to obtain a better performance by the use of an unevaluated value of evaporation coefficient between 0.25 and 0.5.

Figure 6. Results comparing fire brigades using stigmergy in the 40x1 (transparent) and 40x5 (filled) scenarios using different values of pheromone and evaporation.

Our second hypothesis, whose conjecture is that the performance of agents can be improved using stigmergy, is investigated comparing our model against agents that do not use stigmergy. For this comparison, we use the reference implementation of the sample fire brigade agent which comes in the RCR. While our fire brigades uses stigmergy to decide which fires to extinguish, the sample fire brigade adopts a greedy strategy, choosing to extinguish the closest fire in its perception based on the euclidean distance between the agent and the fire spots. Except for this decision strategy, the other aspects of the sample fire brigade and pheromone fire brigade remains the same, i.e. when the water tank is empty, both agents go back to a refuge to refill. Thus, despite the simplicity of both agents, we have a fair comparison between them, taking into account only the aspect of interest to our hypothesis: the effect of stigmergy on the performance of agents.

The third hypothesis claims that the stigmergy model can be as efficient as a direct message approach. To investigate this hypothesis, we developed a multiagent system that interact via direct messages. When an agent perceives a fire sport, it broadcasts a message through a radio channel to all other agents. The message contains an integer (4 bytes) that identifies the building on fire. By the time the messages are received by other agents, they can come to cooperate in the fire-fighting task.

Figure 7 presents the comparison among the three approaches of multiagent systems earlier mentioned (stigmergy, direct messages, and greedy algorithm). The scores showed for the pheromone agents where the better ones from the previous experiments, namely 20x1: evaporation 0.5 and 5 pheromone units; 20x5: evaporation 0.5 and 5 pheromone units; 40x1: evaporation 0.25 and 10 pheromone units; 40x5: evaporation 0 and 5 pheromone units. As we can see, the performance of agents using stigmergy were always better than the greedy agent (t test, 95% confidence), with improvements of 36.55%, 9.01%, 38.60%, and 37.07% in the 20x1, 20x5, 40x1, and 40x5 scenarios respectively. The sample agent was not affected by the number of refuges, as occurs with the pheromone agent.

Figure 7. Comparison among three approaches for fire brigade agents: stigmergy, greedy, and direct messages.

Regarding the multiagent system that interact via direct messages, we can observe the following issues (t test, 95% confidence). In the 20x1 and 40x1 scenarios, the performances were equivalent. In these cases, the results of the multiagent system which uses stigmergy were as good as the results of the system which uses direct communication. In the 20x5 and 40x5 scenarios, the direct message system has a better performance. We noticed that the higher is the quantity of agents, the higher is the score, due to the quantity of agents fighting fire spots. When the scenario has only 1 refuge and the agents are using pheromone, the result is better than 5 refuges, which was explained in the hypothesis of groups that were formed through pheromone to fight fire spots cooperatively (described in the previous experiments). So, the 40x1 scenario would be the best amongst the others because it presents two main issues that affect the performance and cooperation: a) a bigger number of agents and, b) the consequent growth of the number of individuals per group.

Despite the fact that the direct message system appears to be better than the pheromone system in some cases, it must be taken into account a trade-off between the performance and the use of the communication channel. Figure 8 presents the use of the communication channel per timestep for the message agent to produce the results previously presented in Figure 7.

(a) (b)

(c) (d)

Figure 8. Use of the communication channel, measured by the average of bytes exchanged among all the agents per timestep.

We can see that in configurations with the same number of agents, the use of communication is initially similar. This occurs because the agent's start position are the same for all executions, as well as the location of the initial fire spots. The peaks that can be seen are formed when a large amount of agents are simultaneously fighting the same fire spot, since each agent continues to broadcast messages to all other agents. The higher peaks at the beginning of the graphs are due to a combination of the following factors: the agents start at the same position, and eventually engage themselves to fight the same fire spot.

Another behavior observed is caused by the number of refuges. When the map contains only one refuge, the use of the communication channel shows more dips. We suggest that this happens due to the simultaneous action of a large amount of agents in the same fire after refilling their tanks at the available refuge. When the fire spot is extinguished, the agents go back to search for other fire occurrences, meanwhile, they do not use the communication channel. When the next fire spot is found by an agent, all the other agents receive a message and start again to simultaneously fight the new fire occurrence. In the case of five refuges, the peaks and dips were not so accentuated since the agents have more options to refill their tanks, causing the formation of small groups constantly fighting fires.

Differently from the system that uses direct messages, our multiagent system that implements stigmergy does not require the communication channel. The communication is done only by laying down pheromones in the environment. To verify the use of pheromone in different situations, we measured the amount of present pheromone on each timestep. Figure 9 presents the used pheromone for each scenario.

(a) (b)
(c) (d)

Figure 9. Use of pheromone, measured as the sum of the existing pheromones over all the roads per timestep.

As can be seen in the scenarios 20x1, 20x5 and 40x1, there is an initial peak of pheromone. This can be explained because the agents initially extinguish fires that are far from a given refuge, causing a longer travel to refuel and consequently increasing the amount of dropped pheromone. After this peak, the agents start seeking fire spots to extinguish with high probabilities of being closer to a refuge, making the distance to refuel shorter and so decreasing the amount of dropped pheromone. In the case of 40x5 this behavior is not observed since the best result among the experiments had 0% of evaporation rate, accumulating the dropped pheromone over all the simulation.

Given the use of the communication channel and given the performance of the multiagent system which uses communication (which is better than the use of pheromones in the configurations with 5 refuges and equivalent with 1 refuge), we can observe that in a disaster scenario with a broken communication system, the use of stigmergy would be a reasonable solution. Another approach could use a hybrid system, mixing direct and indirect communication. If the direct communication weren't running or no agents were sending messages, the indirect communication would be useful. In a real application of this approach, the agents could stick electronic flags representing the pheromone on the road. Thus, other agents could pass through the flags and extract information about the pheromone intensity and calculate the evaporation based in the elapsed time since the flag had been sticked.

7 Conclusions and Future Work

In this paper we have addressed three hypotheses: the feasibility of using stigmergy in the RoboCup Rescue (RCR); the improvement in the performance of the agents as a result of using stigmergy and the efficiency of the stigmergy when compared to a multiagent system based on direct messages. Related work in robotics and multiagent systems [10, 6, 15, 13, 11, 12, 14, 5] do not consider stigmergy in an environment with the RCR characteristics.

We have extended the RCR to incorporate stigmergy. The agents can now drop and sense pheromones in the environment, enabling the formation of pheromone trails and so, the use of stigmergy. The pheromone evaporation was also incorporated in order to minimize the existence of obsolete paths. The component based architecture of the RCR has provided high flexibility for changes in the simulator.

We have presented a set of experiments to demonstrate the use of stigmergy by fire brigade agents. The results obtained from the experiments show that the use of stigmergy is feasible. Moreover, the use of stigmergy leads to an improvement in the performance of agents from 9.02% to 38.6% when compared to a system without communication and can be as efficient as a system which uses direct communication, depending on the number of available refuges.

As future work we consider to perform experiments enabling all sorts of agents (fire brigades, police forces and ambulance teams). However, we need to extend the RCR in a way where the agents will be able to distinguish different types of pheromones in the environment (e.g. fire brigades must be stimulated by pheromones dropped by other fire brigades rather than ambulance teams). Another possibility for future work would be the adaptation of the algorithm of Santos and Bazzan [14] to support the use of stigmergy. Also, it is considered the idea of comparing the stigmergy-based approach to multiagent systems used in the official competition.

References

[1] E. Bonabeau, G. Theraulaz, and M. Dorigo. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, USA, 1999.

[2] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3:159–168, 1990.

[3] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Positive feedback as a search strategy. Technical Report 91–016, Politecnico di Milano, 1991.

[4] Paulo R. Ferreira, Jr., Felipe Boffo, and Ana L. C. Bazzan. A swarm based approximated algorithm to the extended generalized assignment problem (E-GAP). In *Proceedings of the 6th International Joint Conference on Autonomous Agents And Multiagent Systems (AAMAS)*, pages 1231–1233, May 2007.

[5] Paulo Roberto Ferreira, Jr., Fernando dos Santos, Ana L. C. Bazzan, Daniel Epstein, and Samuel J. Waskow. Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. *Autonomous Agents and Multi-Agent Systems*, 20:421–443, May 2010.

[6] N.R. Hoff, A. Sagoff, R.J. Wood, and R. Nagpal. Two foraging algorithms for robot swarms using only local communication. In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pages 123 –130, 2010.

[7] N. Jennings, K. Sycara, and M. Wooldrige. A roadmap of agent research and development. *Autonmous Agents and Multi-Agent Systems*, (1):7–38, 1998.

[8] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: search and rescue in large-scale disasters as adomain for autonomous agents research. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, volume 6, pages 739–743, Tokyo, Japan, October 1999. IEEE.

[9] A. Marjovi, L. Marques, and J. Penders. Guardians robot swarm exploration and firefighter assistance. In *Workshop on NRS in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.

[10] Yan Meng, O. Kazeem, and J.C. Muller. A hybrid aco/pso control algorithm for distributed swarm robots. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 273–280, 2007.

[11] David Payton, Regina Estkowski, and Mike Howard. Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems*, 44(3–4):229–240, 2003.

[12] Vagelis Plevris, Matthew G. Karlaftis, and Nikos D. Lagaros. A swarm intelligence approach for emergency infrastructure inspection scheduling. In Kasthurirangan Gopalakrishnan and Srinivas Peeta, editors, *Sustainable and Resilient Critical Infrastructure Systems*, volume 1, pages 201–230. Springer Berlin, Berlin, 2010.

[13] A.H. Purnamadjaja and R.A. Russell. Pheromone communication: implementation of necrophoric bee behaviour in a robot swarm. In *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, volume 2, pages 638–643, 2004.

[14] Fernando dos Santos and Ana L. C. Bazzan. Towards efficient multiagent task allocation in the robocup rescue: a biologically-inspired approach. *Autonomous Agents and Multi-Agent Systems*, 22:465–486, May 2011.

[15] Olivier Simonin and François Charpillet. Indirect cooperation between mobile robots through an active environment. In *5th National Conference on Control Architecture of Robots*, pages 71–80, 2010.

[16] C. Skinner and M. Barley. Robocup rescue simulation competition: Status report. In Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 632–639. Springer-Verlag, Berlin, 2006.

[17] M. J. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, 2002.

Appendix A

scenario	pherom. units	evaporation coefficient				
		0	0.25	0.5	0.75	1
20_1	0	0.304 ± 0.014	0.309 ± 0.021	0.306 ± 0.014	0.321 ± 0.029	0.315 ± 0.012
	1	0.350 ± 0.025	0.351 ± 0.040	0.347 ± 0.032	0.328 ± 0.012	0.325 ± 0.041
	5	0.335 ± 0.024	0.359 ± 0.027	0.364 ± 0.026	0.333 ± 0.026	0.325 ± 0.024
	10	0.328 ± 0.023	0.344 ± 0.012	0.363 ± 0.035	0.325 ± 0.020	0.329 ± 0.027
20_5	0	0.260 ± 0.014	0.256 ± 0.011	0.263 ± 0.025	0.257 ± 0.012	0.255 ± 0.014
	1	0.261 ± 0.027	0.248 ± 0.053	0.268 ± 0.024	0.257 ± 0.022	0.264 ± 0.024
	5	0.269 ± 0.019	0.265 ± 0.042	0.271 ± 0.032	0.264 ± 0.017	0.263 ± 0.018
	10	0.261 ± 0.018	0.263 ± 0.024	0.261 ± 0.031	0.248 ± 0.027	0.249 ± 0.017
40_1	0	0.413 ± 0.016	0.412 ± 0.030	0.419 ± 0.027	0.437 ± 0.023	0.430 ± 0.023
	1	0.448 ± 0.032	0.446 ± 0.028	0.448 ± 0.029	0.418 ± 0.043	0.432 ± 0.018
	5	0.432 ± 0.009	0.458 ± 0.025	0.456 ± 0.018	0.443 ± 0.019	0.441 ± 0.033
	10	0.427 ± 0.013	0.475 ± 0.031	0.472 ± 0.027	0.457 ± 0.028	0.450 ± 0.021
40_5	0	0.405 ± 0.035	0.387 ± 0.028	0.390 ± 0.027	0.389 ± 0.027	0.405 ± 0.031
	1	0.419 ± 0.038	0.409 ± 0.060	0.419 ± 0.048	0.427 ± 0.044	0.394 ± 0.033
	5	0.447 ± 0.043	0.444 ± 0.049	0.416 ± 0.075	0.406 ± 0.050	0.399 ± 0.035
	10	0.404 ± 0.056	0.420 ± 0.030	0.435 ± 0.090	0.417 ± 0.033	0.385 ± 0.037

Table 1. Results for each *Kobe4* scenario, showing the average score for each pheromone unit and evaporation coefficient (grey cells indicate the best score).