Extending the RoboCup Rescue to Support Stigmergy: Experiments and Results

Gabriel R. C. Jacobsen, Carlos A. Barth, Fernando dos Santos

[1]Departamento de Sistemas de Informação
Universidade do Estado de Santa Catarina (UDESC) – Ibirama, SC – Brazil

gabrielrigoj@gmail.com, {carlos.barth, fernando.santos}@udesc.br

Abstract. *Social insects have inspired researches in computer sciences as well as 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 trough 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 and the improvements of performance that the agents can be led to. We extended the RoboCup Rescue environment to enable the use of stigmergy by the agents in it. Experimental results shown that the use of stigmergy leads to an improvement on agents' performance by 11.5% to 26%, 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 [16]. 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 and negotiation are available in the literature. Ideally, those approaches shall be evaluated on real world scenarios to check their effectiveness. The RoboCup Rescue[8] simulator was created with this purpose.

In the RoboCup Rescue the agents face 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 limited by a maximum number of bytes an agent can send trough a channel at once. Those restrictions impose the need of an effective use of the available resources. Despite the ability of agents to communicate among themselves via radio channels, if the communication system was broken (due to the catastrophe) there would be no way to interact.

Social insects have been 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 insects 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 study regarding the use of stigmergy in the RoboCup Rescue environment. The current version of the RoboCup Rescue does not allow indirect communication between agents through stigmergy. So, this paper investigates two hypothesis: a) it is possible to use stigmergy in the RoboCup Rescue environment; and b) the performance of agents is improved using stigmergy.

We extended the RoboCup Rescue simulator to enable the use of stigmergy by the agents. We performed a set of experiments to verify the feasibility of stigmergy and to compare the performance against agents that do not use stigmergy. The results showed that the use of stigmergy leads to an improvement on agents' performance by 11.5% to 26%, depending on the scenario.

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

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 construction of nests and the cooperative transport of prey. According to [1], the collective activities of social insects are self-organized. The complex collective behavior may emerge from interactions among individuals that exhibit simple behavior, in a flexible and robust way. These abilities inspired engineers and computer scientists to develop models that mimic the self-organized behavior of social insects to solve problems. These models are then 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 the stimulus.

Stigmergy is observed in the process of ant foraging, in which ants search for food. Initially there is no pheromone in the environment, which means that the ants take random paths to search for food. If an ant finds a source of food, this ant moves back to the nest, laying a pheromone trail while walking. When the nestmates sense the pheromone trail, they are stimulated to follow it to the food source. After a while, a lot of ants are engaged in the transportation of the food by 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 a path is probabilistic, and take into account the number of ants that already followed the 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 units 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 the value of A_i is, 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, a path with slightly more pheromone than the other will have a 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.

An ant that passes on a path already marked with pheromone also drops a certain amount of pheromone to reinforce the stimulus on its nestmates. Thus, the amount of pheromone present on the path depends on the number of ants which already passed 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 k that passed on the path A, as show in equation 3.

$$\Delta\tau_A(t, t + 1) = \sum_{k=1}^{m} \Delta\tau_A^k(t, t + 1) \tag{3}$$

The value of ρ represents the coefficient of pheromone evaporation. As the ants stop using a certain 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. 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 Simulation League [8, 15] is to provide a testbed for simulated rescue teams acting in situations of urban disasters. Currently the RoboCup Rescue 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, buried and/or injured civilians. The simulator incorporates some collaborative agents acting to mitigate the situation. Some issues such as heterogeneity, limited information, limited communication channel, planning, and real time characterize this as a complex multiagent domain [8]. RoboCup Rescue aims at being an environment where multiagent techniques that are related to these issues can be developed and benchmarked.

In the RoboCup Rescue, the main agents are fire brigades, police force, and ambulance teams. Agents have limited perception of their surroundings; can communicate only by radio channels, but are limited in the number and size of messages they can exchange. Regarding information and perception, agents have knowledge about the map. This allows agents to compute the paths from their current locations to given places. However this does not mean these paths are free since an agent has only limited information about the actual status of some path, e.g. whether or not it is blocked by debris.

At the implementation level, the simulator is composed of a kernel and a set of modules. The kernel controls the simulation, invoking every module as needed. Each module is responsible for simulate an aspect of the disaster scenario. For instance, the fire module simulates fire ignition and propagation on the buildings, and the traffic module is responsible for moving the agents in the map.

Each simulation follows a sequence of time steps. On each time step, every agent must decide what action will be taken, given its perception. The perception is composed of objects representing the environment (e.g. a building object, a road object, etc). For each object, the agent can access its properties (e.g. whether a building is on fire, amount of blockages over a road, etc). When a decision is taken, the agent sends a command to the kernel with the chosen action, e.g.: the agent sends a *move* command containing the path he wants to move on whenever the action is “to move on the environment”.

To measure the performance of the agents, the rescue simulator 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 [9] 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 with and without obstacles. The work of [14] focuses on the capability of the robots in perceiving the environment as a way of making decisions in a collaborative swarm of robots through stigmergy. In [12] the authors aim to implement the necrophoric behavior of the bees as a way to give the robots in the swarm the capability of recognizing and rescuing a disabled robot. Payton et al. [10] presents a swarm of robots acting in a rescuing scenario, where virtual pheromones are used as a strategy of communication and coordination. In [11], the authors conclude that information about damages in essential infrastructure is crucial to make decision in a critical scenario. 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.

Regarding RoboCup Rescue, an efficient coordination amongst agents is a critical factor given the characteristics of the scenario and limited capabilities of the agents. Swarm techniques have been applied to get coordination. Ferreira Jr. et al. [4, 5] presented Swarm-GAP, a multiagent task allocation algorithm based on the model of division of labor in insect colonies. Santos and Bazzan [13] 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 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 agent's coordination.

The current version of the RoboCup Rescue simulator[1] does not allow an agent to use indirect communication, via stigmergy (dropping and sensing pheromones). Thus, we have not found any work that report results about the performance of a team of agents which uses stigmergy in this environment. This paper investigates this issue. We improve the simulator to allow the stigmergy, an run a set of experiments to verify the performance.

5. Stigmergy in the RoboCup Rescue

In order to use stigmergy in the RoboCup Rescue, we need to extend the simulator to accomplish the following requirements:

(a) every road A must have a way to store an amount of pheromone τ_A;
(b) it must be possible to an agent k to perceive the pheromone τ_A stored on a road;
(c) it must be possible to an agent k to drop certain amount of pheromone $\Delta\tau_A^k$ on a road;
(d) the existing pheromone on a road must evaporate over a time, given an evaporation coefficient ρ .

To satisfy the requirement (a), we extended the implementation of the simulator to incorporate property τ_A on every road object A. The property stores the amount of pheromone virtually laid on each road. Once agents can access the properties set to a road object, these agents will detect the property value τ_A at every time step, satisfying the requirement (b).

In order to satisfy the requirement (c), we extended the implementation to provide to the agent a way it can informs the amount of pheromone $\Delta\tau_A^k$ it want to drop on every road of a given path. The traffic simulator, while processing the moving command, is responsible for increasing the amount of pheromone following the equation 2.

The last requirement (d) is accomplished by a change on the traffic simulator. At every time step, the traffic simulator applies the user-defined value of the evaporation coefficient ρ over the amount of pheromone τ_A present on every road, as given by equation 2.

6. Experiments and Results

To investigate our first hypothesis, which claims that is possible to use stigmergy in the RoboCup Rescue, we performed a series of experiments using the *Kobe4* map[2]. We enabled only fire brigade agents to accomplish the task of fire-fighting. We also disabled the simulation of blockages, ensuring all roads are free. Fig. 1 presents the map used in the experiments.

The decision of disabling all agents but the fire brigades was based on our perception of similarity between the two tasks of fire-fighting and food-searching (ants). 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 for some source; since some source is found a sign is sent to the other agents expecting some cooperation.

[1] available at www.robocuprescue.org

[2] The *Kobe4* map is used in competitions of the *RoboCup Rescue Simulation League*. It is available for download at the site of the RoboCup Rescuesimulator

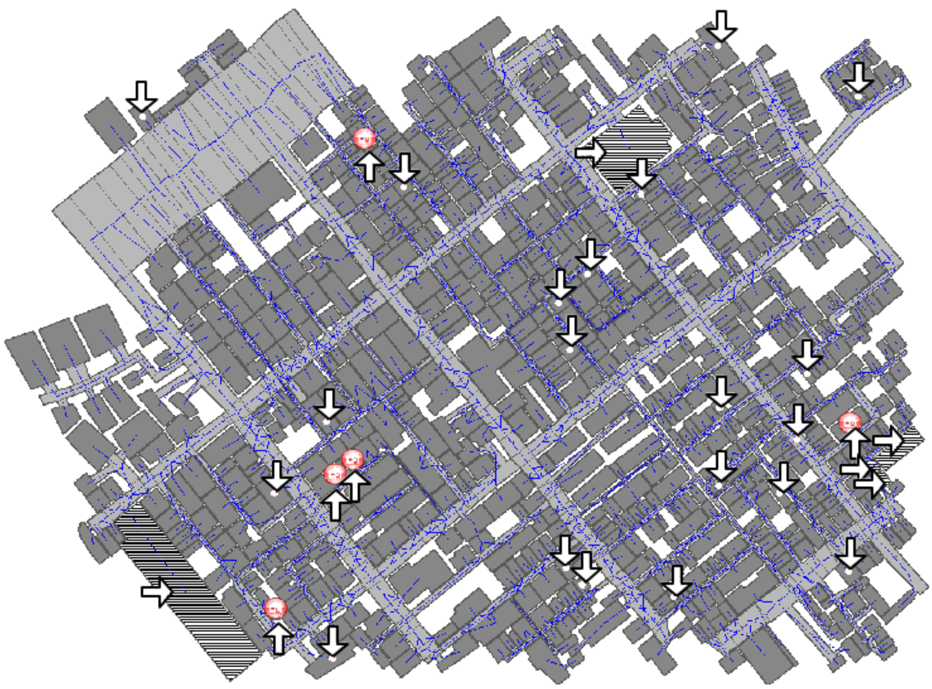

Figure 1. The *Kobe4* map used in the experiments with 5 refuges. The small circles (pointed out by up-to-down arrows) represent the initial position of the fire brigades. The big circles (pointed out by down-to-up arrows) represent the refuges, in which fire brigades can refill their water tanks. Striped rectangles represent the five initial fire spots (pointed out by left-to-right arrows).

Similar to ants searching for a food source, the fire brigade agents should explore the environment looking for buildings on fire. In the same way the ants transport a piece of food to the nest when they found it, fire brigades go back to some refuge to refill when they are running out of water. Since each agent has knowledge about the map and its roads, it can calculate the shortest path to the refuge. When traveling back to a refuge, each agent drops an amount of pheromone on the roads it passes. When the water tank is full, each agent that leaves the refuge is stimulated by the pheromone on the roads, according to 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 the nest and are stimulated to move to a food source.

The experiments were performed on two scenarios: 20 agents and 1 refuge (20_1 for short), and 20 agents and 5 refuges (20_5). On each scenario, the following values where used for $\Delta\tau_A^k$ (amount of pheromone dropped by each agent on a piece of road): 0 (no pheromone), 1, 5, and 10 pheromone units. The evaporation coefficients 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 left (e. g. after an earthquake followed by a fire and the intervention of the fire brigades). Figure 2 presents the results obtained. All data is 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.

As we can see, when we consider each scenario individually, there is no differ-

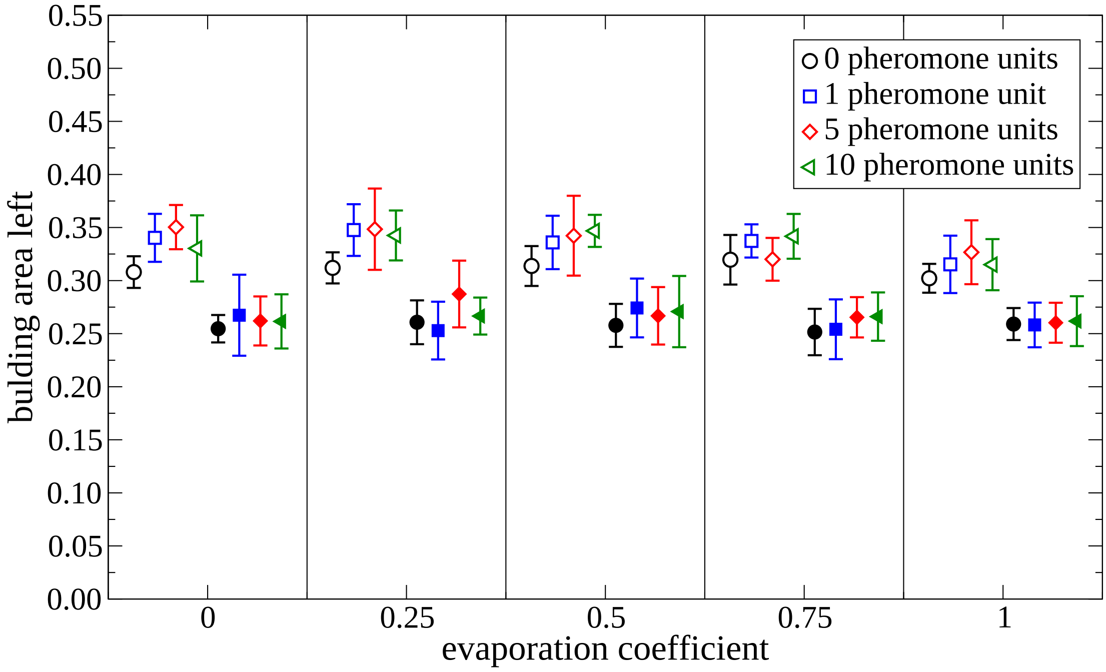

Figure 2. Results for each *Kobe4* scenario, showing the average building area left for each pheromone unit and evaporation coefficient. Unfilled symbols represent the 20_1 scenario, while filled symbols represent the 20_5 scenario.

ence that can be statistically significant between the performances, independently of the combination of evaporation coefficient and pheromone units. We believe this is due to the reduced set of values tested for those parameters. Given the complexity and dynamics of the environment, it is possible that a better performance is obtained with an unevaluated combination of evaporation coefficient and pheromone units.

From the experiments we can observe a relation between the use of stigmergy and the number of refuges where the fire brigades refill its tanks. Given 1 refuge enabled, the performance is higher (t test, 95% confidence) than enabling 5 refuges in all the scenarios. With one refuge enabled, all the agents move to it to refill its tanks, increasing the amount of pheromone in the paths. So, the stimulus on the agents which are leaving the refuge is increased. As a result, many agents move to the same fire spot, fighting the fire cooperatively and reducing the damages, leading to a improved performance.

Our second hypothesis, which conjectures that the performance of agents can be improved using stigmergy, is investigated comparing to agents that do not use stigmergy. For this comparison, we use the reference implementation of the sample fire brigade agent which comes in the RoboCup Rescue. 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 both sample agent and ours remains the same, i.e. when the water tank is empty, both types of agents go back to some refuge to refill. Thus, despite the simplicity of both types of 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.

Fig. 3 presents the results obtained for the sample fire brigade. The results showed for the stigmergy fire brigade are the best ones for each scenario, as highlighted on Table 1 (appendix A). As we can see, the average performance is improved in 26% in the scenario with 1 refuge, and by 11.5% with 5 refuges (t test, 95% confidence).

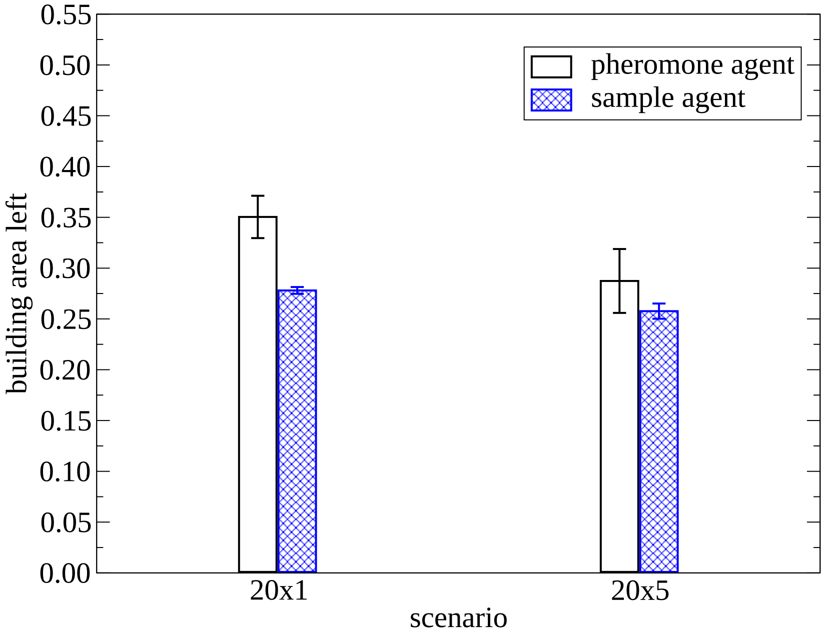

Figure 3. Results comparing fire brigades using stigmergy against sample fire brigades, showing how the performance is improved using stigmergy.

7. Conclusions and Future Work

In this paper we have addressed two hypothesis: the feasibility of using stigmergy in the RoboCup Rescue; and the improvement of the agent's performance as a result of using stigmergy. Previous works in robotics and multiagent systems [9, 6, 14, 12, 10, 11, 13, 5] do not consider stigmergy in an environment with the RoboCup Rescue characteristics.

We have extended the RoboCup Rescue to incorporate stigmergy. The agents can now drop and sense pheromones in the environment, enabling the formation of pheromone trails and thus, the use of stigmergy. The pheromone evaporation was also incorporated in order to minimize the existence of obsolete paths.

We have presented a set of experiments to demonstrates 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 11.5% to 26%, depending on the number of available refuges.

As future work we consider to investigate the literature about swarm intelligence to find out if there is a way of incorporate to the pheromone trails a notation of direction. Currently, there is no indication of the direction the pheromone trails points to. We believe this feature will improve the quality of the agent movement in the environment by avoiding the agents of getting stuck at some point (sometimes an agent keep going back and forward on some part of the path).

We also consider to perform experiments with all three types of agents using stigmergy (fire brigades, police forces and ambulance teams). However, we need to extend the RoboCup Rescue in a way that the agents can be able to distinguish types of pheromones on the roads (e.g. fire brigades must be stimulated by pheromones dropped by other fire brigades rather than ambulance teams).

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] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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	0.308 ± 0.015	0.312 ± 0.015	0.314 ± 0.019	0.320 ± 0.023	0.302 ± 0.014
	1	0.340 ± 0.023	0.348 ± 0.024	0.336 ± 0.025	0.337 ± 0.016	0.315 ± 0.027
	5	0.350 ± 0.021	0.348 ± 0.038	0.342 ± 0.038	0.320 ± 0.020	0.327 ± 0.030
	10	0.330 ± 0.031	0.343 ± 0.024	0.347 ± 0.015	0.342 ± 0.021	0.315 ± 0.024
20_5	0	0.255 ± 0.013	0.261 ± 0.021	0.258 ± 0.020	0.252 ± 0.022	0.259 ± 0.015
	1	0.267 ± 0.038	0.253 ± 0.027	0.274 ± 0.028	0.254 ± 0.028	0.258 ± 0.021
	5	0.262 ± 0.023	0.287 ± 0.031	0.267 ± 0.027	0.265 ± 0.019	0.260 ± 0.019
	10	0.262 ± 0.026	0.267 ± 0.017	0.271 ± 0.034	0.266 ± 0.023	0.262 ± 0.023

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