Ball Chasing Coordination in Robotic Soccer using a Response Threshold Model with Multiple Stimuli

Efren Carbajal and Leonardo Garrido

Intelligent Autonomous Agents Research Group,
Department of Computer Science, ITESM, Monterrey, Mexico
{A00937789,leonardo.garrido}@itesm.mx

Abstract. In any system made of several robots task allocation is an indispensable component in order to achieve coordination. We present two different approaches commonly found in literature to solve the dynamic assignment of the ball chasing task, i.e. multi robot task allocation and division of labour. In particular, we explore both approaches in a 3D simulation robotic soccer domain called Robotstadium. Moreover, we evaluate and compare four controllers representing different coordination implementations belonging to either of the two approaches. We show that by formulating the problem as one of division labour and then employing a response threshold model with multiple stimulus as arbitration mechanism, we provide an efficient algorithm with respect to a proven benchmark solution. We also present evidence indicating that this communication-less algorithm result in an emergent team behavior which *indirectly* also address the positioning problem of robots within the soccer field, keeping physical interference low.

Keywords: Task allocation, response threshold, division of labour, robotic soccer.

1 Introduction

In the last two decades, researchers have given more and more attention to Multi-Robot Systems (MRS). Besides an important reduction in hardware prices, there are some other advantages of MRS like performance benefits, and the accomplishment of inherently complex tasks, which explain its growing use by the scientific community. An example of this can be found in RoboCup, a robotic soccer competition which foster research in robotics and Artificial Intelligence (AI) [1].

Task allocation, the process of assigning individual robots to sub-tasks of a given system-level task, is a very important component of any MRS [13]. In robotic soccer a team of robots have to work together in order to win a match against other team of robots. This entails that robots have to cooperate and coordinate with each other. Where the global task of playing soccer require different roles or sub-tasks like defense, goalkeeper, passing, and shooting be carried out in order to achieve a team behavior.

Robotic soccer researchers have labeled this problem also as dynamic assigning roles or role allocation [10, 14]. Solutions found in literature often involve the construction of a model of the environment and the use of explicit communication to assign roles. A intuitive idea to solve this problem it is to following a centralized approach, however, the environment is only partial observable to each robot, and no single individual has enough information to make correct decisions [14]. Another attempt is to follow a well known approach in Multi-Agent Systems (MAS): negotiation. But, there are some issues concerning to communication in real time environments which make this technique unsuitable [15].

To overcome those issues some researchers had combined local and global information about ball, using global information only when more reliable local information is not available [14].

Given the difficulties experienced and the importance of local information, it seems suitable the use of a Swarm Intelligence (SI) approach to this problem. SI gives special importance to self-organization, local information, implicit communication to achieve emergent coordination. According to Lerman *at al.* [13]:

> "Emergent coordination algorithms for task allocation that use only local sensing and no direct communication between robots are attractive because they are robust and scalable"

In this paper we propose the use of a response threshold model as a SI task allocation mechanism for the ball chasing task in Robotstadium, a robot soccer simulation based on the Standard Platform League (SPL) of RoboCup and its rules [2].

Section 2 introduce the response threshold model and how this is extended in order to cover some issues present in the domain.

Section 3 shows the arquitecture of the generic controller (i.e. the controller used as base for all the coordination strategies). Section 4 describes the coordination strategies. Section 5 describes the experimental setup. Section 6 shows the results and offers an interpretation. Section 7 concludes.

2 Response Threshold

There are different classes of mathematical models which try to explain division of labour in social insects. According to Beshers *et al.* [5] response threshold models is one them, and in their work they described this class of model. In this model every individual has its own internal response threshold for every task,and engaging in this when the level of the stimulus related to such task exceeds their threshold [6]. This kind of models have been successfully implemented by researchers in MAS and robotics. Examples of these applications are: artificial mail retrieval system [8] in MAS domain, and clustering objects [3] and foraging [11] in MRS.

2.1 Fixed Threshold Model with One Task and One Stimuli

Bonabeau *et al.* [7] have developed a simple model of division of labour in insect societies. This model is describe by the authors as following. Suppose X is the state of an individual (where $X=0$ indicate that the task is not been performed, and $X = 1$ indicates the task is been performed), and θ_i the response threshold of individual i ($i = 1, 2... ... n$). Then the probability that an inactive individual i starts performing the task P_i per unit time is:

$$P_i(X = 0 \rightarrow X = 1) = \frac{s^2}{s^2 + \theta_i^2} \ . \tag{1}$$

From the eq. 1 we know that the probability that an individual will perform a task depends on s which is the magnitude of the stimulus related task. Similarity, an individual will become inactive with a probability p as describe in eq. 2. Where $\frac{1}{p}$ is the average time spend by an individual before stops performing the task, therefore, p can be found experimentally.

$$P_i(X = 1 \rightarrow X = 0) = p \ . \tag{2}$$

2.2 Fixed Threshold Model with One Task and Multiple Stimuli

In this paper, we propose a slightly different model a little less stochastic. In this model, the probability $P_i(X = 1 \rightarrow X = 0)$ is not independent of stimulus, so it is not a constant p. Instead, we model this probability as the inverse of $P_i(X = 0 \rightarrow X = 1)$ as it is describe in eq. 3. However, this can't be done without some sort of mechanism which allows to decrease the stimulus while performing the task independently of the execution's duration. Here is where the reference to multiple stimuli comes into play. Basically, we employ different types of stimuli that can be classify either as a excitatory or inhibitory, thus the intensity and direction of the stimulus can be calculated as the difference between the sum of excitatory and the sum of inhibitory stimulus as shown in eq. 4.

$$P_i(X = 1 \rightarrow X = 0) = 1 - \frac{s^2}{s^2 + \theta_i^2} \ . \tag{3}$$

$$s = \sum s_{excitatory} - \sum s_{inhibitory} \ . \tag{4}$$

2.3 Stimulus

Table 1. List of stimuli names and types

Stimulus	Type
Distance to the ball	Excitatory
Number of teammates closer to the ball	Inhibitory
Presence of obstacles (sonar)	Inhibitory

This model differs from the original in two aspects: the number of different stimuli related to the task, and how the probability to stop performing the task is computed. Generally in literature the stimulus is only one straightforward measurement of the environment. However, we believe there are several factors involved in the decision making of whether or not approaching to the ball could derive in a good output, specially when there is not direct communication among the individuals. Those factors and their nature are shown in table 1. Taking into account both type of stimulus: excitatory and inhibitory, make possible for the individual to engage in performing the task when excitatory stimulus predominates, or stop performing the task when inhibitory stimulus outweigh excitatory one. The result is a less stochastic response which adapts more quickly to dynamic environment, making the controller better suited for a real-time, coarse-grained MRS competition.

3 Controller

The control algorithm is based on a simple finite state machine (FSM) as it is depicted in figure 1. Different states represent the different phases of the soccer game, that is, the subtasks in which the overall game is decomposed. These sub-tasks are as follows:

Search The robot look for the ball just by moving its head. If ball has already been seen and there is no perception gained about a teammate robot, then the controller change Chase state. Otherwise, If ball is seen but a teammate perception is received then, it makes a transition to Wait state.

Chase Robot moves toward the ball and remains in this state until ball is close enough, that is when the Shoot state takes place. While going after the ball, this may gets outs of sight, in that case the controller returns to the Search state.

Wait In this state robot is stand up tracking the ball until no perception of other robot is received, alternatively switch to Chase state. Other possibility is to lose out the ball, in which case the controller is forced to return to search state.

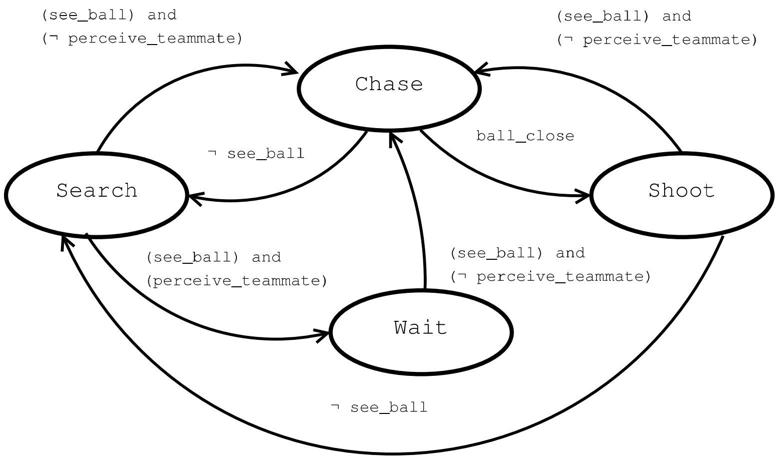

Fig. 1. Finite State Machine diagram.

Shoot Robot hit the ball in goal direction, then as ball moves along, robot either lose it and return to Search state, or start to chase it again, switching back to Chase state.

Transitions between states occur on the base of events that are external (e.g. Perception of another robot) to the robot. The edge labels between states (as shown in figure 1 represent the conditions (also called predicates) that must be true for the transition to occur. The complete list of the predicates, their meanings and how they are evaluated is given in table 2. Their truth values are evaluated from sensor readings at every control cycle.

Table 2. Definition of constants and predicates used by the control algorithm

B_D	Maximum distance between robot and ball to try to kick the ball
`see_ball`	Ball is detected using visual information obtained from the camera
`perceive_teammate`	A teammate is communicating that it is closer to the ball
`ball_close`	Distance to ball $< B\_D$

4 Coordinations Strategies and Evaluation Metrics

In order to coordinate robots, allocation strategies addressed in this work employ two types of communication mechanisms. The first refers as “public” emphasizes in the existence of an explicit collaborative information flow between teammates through a emitter/receiver device. Whereas in the latter, which we are call “private”, there is no explicit information sharing but intead robot recognition algorithms are employed to communicate indirectly by *sensing* the presence of others teammates. On the other hand, the keyword "utility" refers to the use of utility values for the purpose of allocate the chaser role. While the the keyword "reactive" indicate the existence of simple rules which rely solely in perceptions to determine the execution of the ball chasing task.

In threshold-based systems, the *propensity* of any agent to act is given by a response threshold. Basically, if the demand is above the agents threshold then that agent continues to perform the task, conversely, if the demand is below its threshold then the agent stops performing that particular task. In the algorithm presented in this paper the visual perception of the ball, teammates and opponents, represents the agent estimation of the demand or stimuli associated with the ball chasing task.

Thus, in what follows we present four different role allocation algorithms which can be described by some of the previous characteristics.

Public, Utility, Single Robot Allocation Algorithm (PuUS) In this strategy each robot consult the messages of the rest of the team, the message from each teammate consist in the estimation of distance (utility) from its own perspective to the ball. By comparing this messages to its own perception, the individual is able to determine whether or not is the closest robot to the ball, in which case, start chasing the ball. This approach ensures only one robot chase the ball at any given time and it is also the more frequent used among researchers in RoboCup. This path is suitable to formulate the assignation of tasks to robots as a task allocation problem.

Public, Utility, One or Less Robots Allocation Algorithm (PuUOoL) This strategy is based on the previous one, and supported on the fact that there are scenarios when reallocation of the chase task among robots could derive to collisions. One special case is when the closest robot to the ball isn't seeing it, giving rise to more collisions. Therefore this strategy is proposed as an alternative version of PuUS with the additional rule that no robot chase the ball until every robot is stand up and looking to the ball. This rule ensures that the average number of robots chasing the ball at any moment be of one or less.

Private, Reactive, Multiple Robot Allocation Algorithm (PrRM) In this implementation every robot has been programmed to chase the ball without previous communication; therefore, it will be times when more than one robot is in chasing mode. To avoid a grievous situation of undesired collisions, a reactive behavior that interrupt chasing activity is used. The reactive mechanism consist in the recognition of a teammate closer to the ball, in which case the robot stops and waits. This strategy is more in tune with the the division of labour approach follow by bio inspired researchers. Notice that this allocation strategy allows more than one chaser at the same time.

Private, Multiple-Stimulus-Threshold, Multiple Robot Allocation Algorithm (PrMSTM) This algorith is a modified version of PrRM but instead of reactive rules, robots have a threshold model to force action in a stochastic fashion. As explained in section 2, the intention of this mechanism is to ponder several key aspects (i.e. multiple stimulus) of the environment in order to assess the *demand* to execute the ball chasing task.

In order to evaluate the performance of such implementations, we make use of the direct output of every soccer robotic controller: goals (i.e. goals scored as well as goal conceded). But we also propose the use of complementary metrics making emphasis in the intention we have in this work to produce results that are interesting to continue with and validate beyond simulation level. Additional metrics proposed in this work are physical interference and efficiency. Both of them been of much interest in MRS [12, 9].

Performance The main indicator of performance in soccer is the goal difference (i.e. goals scored minus goals conceded). Because depending the team you choose as reference team lose of win the match, this number can take negative values (when reference team lose). To avoid confusion we are going to use as reference the team running implementations of strategies to test. In addition, those results were normalized in order to ensure only zero or positive numbers. The reason behind applying normalization to the data is because positive numbers are required to estimate the efficiency as it is discussed further in efficiency description. To summarize, for performance we mean the output of a normalization process that employs the goal differenceas the only input to generate zero or positive numbers. The normalization consist in adding all numbers with the absolute of the most negative (minimum value of goal difference) such that the most negative one will become zero and all other number become positive.

Efficiency Lets start by defining efficiency, we refer to efficiency as the capability to convert some valuable resource into another new valuable resource. In the context of robotic soccer, we define as the input resource the energy spent by the robot moving around, and as output resource,the total number of goals scored and conceded. However, the energy spent by a robot depends on too many factors, known and unknown, and a precise measurement it is far beyond the reach of this work. Conveniently, we know that the displacement of a robot is proportional to the energy it uses. Thus, we decide to use the displacement as an approximation of the energy spent. This way we end up with equation 5.

$$\eta_i = \frac{\Delta_i}{\epsilon_i} \ . \tag{5}$$

Where:

η_i = Team's efficiency during match i
Δ_i = Normalized goal difference during match i
ϵ_i = Displacement in meters during match i

Interference Another important issue related to the performance of a generic MRS are collisions, also known as *physical interference* [4, 9]. According to Goldberg at al. [9] physical interference arises in competition for space. They show with experiments that the measurement of it can be an effective tool for system design and evaluation. In Robotstadium's case, space competition are present mostly in a particular task: chasing the ball. The number of collisions by members of the tested team was tracked down with help of an emulated GPS device in the supervisor's code (i.e. referee code that along with code of controllers and world complete the simulation).

5 Experimental Setup

The main objetive of this study is to challenge two ideas widely established in practice. The first is related to the number of robots which should be chasing the ball at any given time during the match. So far this number has remained fix to one by roboCup researchers. As a consequence, two out of the four implementations evaluated in this section represent a division of labour approach which inherently allocate tasks without restrictions in the number of individuals that engage in them. The second idea is about the need of make use of a spatial model, autolocalization and explicit commmunication in order to distribute robots, improve performance and reduce collisions among robots. A way to challenge this idea is by contrasting it with bio inspired implementations, which produce an emergent team behavior that result in a self-organized distribution of robots in the field without direct communication or a concrete spatial model of the environment.

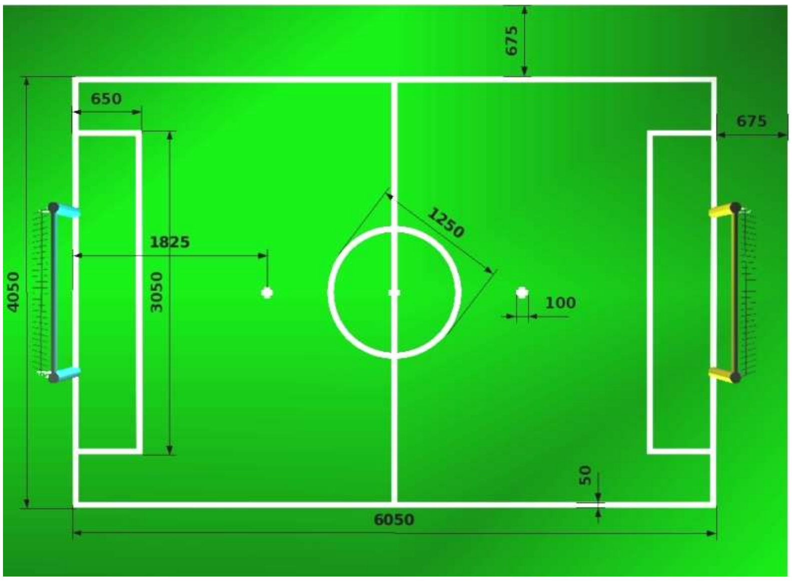

Fig. 2. Robotstadium field measures in millimeters taken from the top orthographic projection of the simulation in Webots

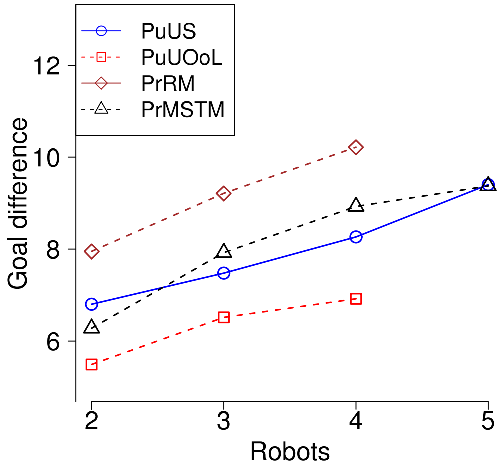

Fig. 3. Comparison of performance as group size increases

An experimental trial consist in running complete match divided in two half periods of 10 minutes, where in one side of the field there is a team representing one of the four strategies, while in the other side, there is a single robot. In this manner, experimental trials have been performed with each of the four strategies: PuUOoL, PuUS, PrRM and PrMSTM; and for each strategy forty trials were performed with two, three, and four robots. An extra run of experiments with five robots for the two better performed strategies were added as an attempt to shed light to further conclusions. As a result, a total of 560 experiments were carried out.

6 Results and Discussion

Data obtaining from all the experiments have been analyzed with a two-way ANOVA statistic test. Where performance, interference and efficiency were explained using coordination strategy as a categorical variable and team size (number of robots) as numerical variable. In all three tests both variables, strategy and team size, were found to be statistically significant with p-values < 0.05.

Beforehand we hypothesized that PrRM would produce the higher number of interferences among all strategies, thus, its performance would be outperformed by that of strategies PuUS and PuUOoL. Surprisingly, this was not the result as shown in fig. 3, despite the high level of interference. We believed this is due a higher chance to get first to the ball and to handicap the other team's robot. However, when looking to interference, as fig. 4 shows it, the performance gains of PrRM become overshadow by exponential growth in interferences.

On another hand, performance of PuUOoL was lower among all implementations. And performance of PuUS and PrMSTM was similar between them with no statistically significant difference observed. A slightly better performance was presented in the threshold strategy when the team size is 4, but with virtually the same performance when the team size reach 5 robots. Besides the fact of both implementations achieved almost the exact performance when team size was fixed to five, it is worth to note as well that PrMSTM strategy tendency to improve along the team size increases, ends abruptly when arrives to the five robots per team, resulting in both strategies performance curves crossing each other.

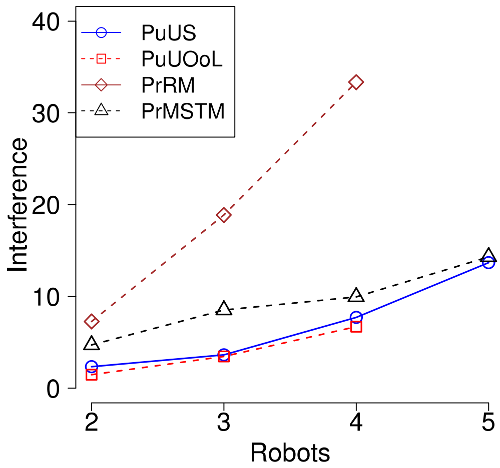

Fig. 4. Comparison of interference level as group size increases

Could these two seemingly independent issues be related?, we suspect they are indeed, we believe the performance of both strategies is bound by the field size, which may happens to be too small to allow an equal and active participation from all the robots. Outcome from efficiency, seems confirm this. Even when there was no apparent gain in performance in PrMSTM when passing from four to five robots, there was a gain in efficiency, and since efficiency is affected either by performance or energy spent, this gain must come mostly from a reduction in the latter. This reduction in energy spent is more pronounced than in others team size configurations, which can be explain by a good coverage of whole field by the five robots. Note that if this hypothesis were correct, may imply that PrMSTM strategy is also good at adapting or showing robustness to environmental changes. However, more evidence is needed to support this. So far we can only conclude that both controllers performed at a similar level.

As mentioned before, we hypothesize that the presence of interference of PrRM would be greater than found in PuUO, and the latter greater than found in PuUOoL. As figure 4 shows it, the guess was correct. However, we didn't know what to expect about the magnitude of those difference, neither of the level of interference in PrMSTM. The outcome is that the amount of collisions in strategy PrRM is, to a large extent, bigger than the rest of implementations.

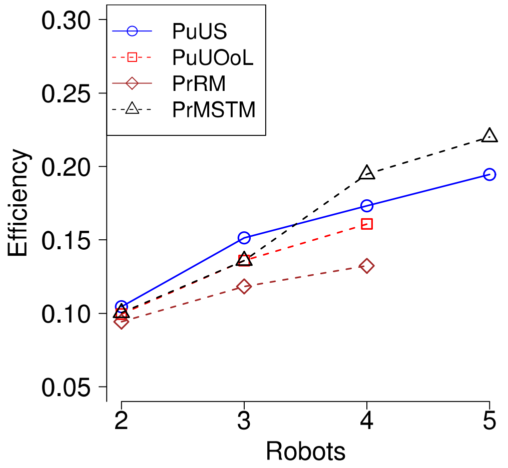

Fig. 5. Comparison of efficiency results as group size increases

Unlike PrRM, in PrMSTM the interference level is not a grievous problem, in fact, this is comparable to those in PuUOoL and PuUS when four or five robots are set per team.

Another hypothesis we were interested to proved is related to the efficiency. Due to the restriction of one at most, we were expecting a more efficient use of energy in PuUS and PuUOoL. But as shown in figure 5 this was not exactly the case. For three robots per controller PuUS is the most efficient, however, as the number of robots increase to four and up to five, PrMSTM takes the credit as the most efficient.

Is interesting how PrMSTM and PrRM both representing a division of labour approach are at the same time the most efficient and inefficient. This indicate the influence of the threshold model in regulating the number of active chasers. A exception of PrMSTM, what happened with the rest of implementations was that efficiency holds almost steadily from the start only gaining a little with every increase in team size. From the scalability perspective, we can observe in figure 5 that efficiency curves for all strategies, except for PrMSTM, resembled logarithmic functions, this fact indicate marginal returns while adding more robots.

7 Conclusions

In this paper, we have presented a comparative study of four distributed, multi robot allocation mechanisms that allow a team of autonomous, embodied agents to dynamically allocate the fittest individual(s) to a given task. These coordination algortihms can be classified in two approaches, task allocation with PuUS and PuUOoL as representatives, and division of labour with PrRM and

PrMSTM as representatives. We compared their performance and efficiency in a robotic soccer case study concerned with a ball chasing task.

We showed that framing the ball chasing task assignment as a division of labour problem and using a multiple stimulus threshold model to address it, the system performs as well as the benchmark solution (i.e.PuUS for been the most widely used among researchers) while at the same time arrive to additional benefits as a significant increment in efficiency when the team size is set to four or five robots without any sort of direct communication.

References

1. Robocup objective: Pushing the state of art. `http://www.robocup.org/about-robocup/objective/`
2. Robotstadium: online robot soccer competition. `http://robotstadium.org`
3. W Agassounon and A Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 3 AAMAS 02*, pp. 1090–1097 (2002)
4. R C Arkin and T Balch. Cooperative multiagent robotic systems. *Artificial Intelligence and Mobile Robots*, pp 277–296 (1998)
5. S N Beshers and J H Fewell. Models of division of labor in social insects. *Annual Review of Entomology*, 46(413-440):413–440 (2001)
6. E Bonabeau, M Dorigo, and G Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press (1999)
7. E Bonabeau, G Theraulaz, and J L Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings Biological Sciences*, 263(1376):1565–1569 (1996)
8. Eric Bonabeau, Andrej Sobkowski, Guy Theraulaz, and J L Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. *Bio Computation and Emergent Computing*, pp. 36–45 (1997)
9. D Goldberg and M J Matarić. Interference as a Tool for Designing and Evaluating Multi-Robot Controllers. *AAAIIAAI*, 8:637–642 (1997)
10. Eric Henry Work, Chown, Tucker Hermans, and Jesse Butterfield. Robust Team-Play in Highly Uncertain Environments (Short Paper). (Aamas) (2008)
11. M J B Krieger and J B Billeter. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84 (2000)
12. Thomas H. Labella, Marco Dorigo, and Jean-Louis Deneubourg. *Division of labor in a group of robots inspired by ants' foraging behavior.* PhD thesis (2006)
13. Kristina Lerman, Chris Jones, Aram Galstyan, and Maja J Mataric. Analysis of Dynamic Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research*, 25(3):225–241 (2006)
14. Michael J Quinlan, Steven P Nicklin, Stephen R Young, Timothy G Moore, Stephan K Chalup, and Richard H Middleton. The 2005 NUbots Team Report. *Electrical Engineering* (2006)
15. Thomas Rofer, Michael Weber, Hans dieter Burkhard, J Matthias, G Daniel, Jan Hoffmann, and Bastian Schmitz. German Team: RoboCup 2005 (2005)