Learning and Measuring Specialization in Collaborative Swarm Systems

Special issue on Mathematics and Algorithms of Social Interactions,
C. Anderson & T. Balch (Eds.), *Adaptive Behavior*, 12(3–4):199–212, Dec. 2004.

Ling Li
Learning Systems Group, California Institute of Technology

Alcherio Martinoli
Swarm-Intelligent Systems Research Group, EPFL

Yaser S. Abu-Mostafa
Learning Systems Group, California Institute of Technology

This paper addresses qualitative and quantitative diversity and specialization issues in the framework of self-organizing, distributed, artificial systems. Both diversity and specialization are obtained via distributed learning from initially homogeneous swarms. While measuring diversity essentially quantifies differences among the individuals, assessing the degree of specialization implies correlation between the swarm's heterogeneity with its overall performance. Starting from the stick-pulling experiment in collective robotics, a task that requires the collaboration of two robots, we abstract and generalize in simulation the task constraints to k robots collaborating sequentially or in parallel. We investigate quantitatively the influence of task constraints and types of reinforcement signals on performance, diversity, and specialization in these collaborative experiments. Results show that, though diversity is not explicitly rewarded in our learning algorithm, even in scenarios without explicit communication among agents the swarm becomes specialized after learning. The degrees of both diversity and specialization are affected strongly by environmental conditions and task constraints. While the specialization measure reveals characteristics related to performance and learning in a clearer way than diversity does, the latter measure appears to be less sensitive to different noise conditions and learning parameters.

Keywords collaborative swarm systems, distributed learning, specialization, diversity

Correspondence to: Ling Li, Learning Systems Group, California Institute of Technology, Caltech 136-93, Pasadena, CA 91125, USA.
E-mail: ling@caltech.edu
Tel.: +1-626-3953887, *Fax*: +1-626-7950326.

1 Introduction

Artificial swarm systems based on swarm intelligence consist of relatively simple autonomous agents. They are truly distributed, self-organized, inherently scalable, and exploit an adequate balance between explorative and exploitative behavior for robustly facing changes in the environment or task conditions (Bonabeau, Dorigo, & Theraulaz, 1999).

Swarm systems can be homogeneous or heterogeneous. A homogeneous system usually consists of physically identical entities with the same hardware and software capabilities. Depending on the metric used for assessing the swarm performance, different entities might also generate a homogeneous swarm if each of them performs the same function and contributes in the same way to the swarm performance. A heterogeneous system may differentiate at different levels: at the hardware level, at the controller software level, at the function/behavior level, etc. In this paper, we use software agents emulating real robots that differ exclusively at the controller level: they can be endowed with different control parameters.

Several degrees of heterogeneity are possible: from homogeneous swarms (one caste) to fully heterogeneous ones (the number of castes equal to the number of agents in the swarm) via multi-caste swarms. Depending on environmental and task constraints, a homogeneous solution may not achieve the best results. Learning, as an automatic way to adjust control parameters or select behavioral rules without a priori assuming the degree of swarm heterogeneity, represents an effective tool to explore not only homogeneous solutions (Hayes, Martinoli, & Goodman, 2003) but also heterogeneous ones (Murciano, Millán, & Zamora, 1997; Li, Martinoli, & Abu-Mostafa, 2002). In this paper, we are interested in comparing *distributed learning*, where the adaptation process can occur at the level of a single robot or a caste, with *centralized learning*, where a homogeneous swarm adapt as a whole and the same set of control parameters characterizes each individual.[1] In order to perform a fair comparison, we have chosen a simple algorithm which can be easily adapted to the different types of reinforcement signal and swarm diversity.

The *global reinforcement signal*, often equivalent to the swarm performance, is stable and meaningful for the goal of learning. However, when a global reinforcement signal is used in a distributed learning process, the credit-assignment problem has to be solved in some way. That is, the global signal measuring the swarm performance has to be decomposed or translated so that it can be interpreted at the agent or caste level (Murciano et al., 1997). In case of a potentially heterogeneous swarm, agents might contribute in different ways to the swarm performance. If their contribution is a function of their control parameters, a clear mapping between individual control solution and swarm performance could be derived. Defining this global-to-local mapping is equivalent to solving the credit-assignment problem. For certain types of tasks and software agents without communication or memory limitations, this problem has been solved in elegant ways (e.g., Wolpert & Tumer, 2001). However, if agents are characterized by limited communication and/or information processing capabilities, the credit-assignment problem becomes much more difficult. An elegant way to bypass the credit-assignment problem proposed in the literature for such systems is to exploit a global communication network to broadcast the adaptation results to each agent, using some centralized learning process, in combination with an enforcement of the swarm homogeneity (Hayes et al., 2003; Versino & Gambardella, 1997). As a matter of fact, if each agent contributes (on average) in the same way to the swarm performance, it should get the same share of the merit achieved at the swarm level.

Depending on the availability of a signal assessing the swarm performance and the communication capabilities of real agents, a centralized or distributed solution based on the global reinforcement signal might be too expensive, if not infeasible. An alternative solution is to use a *local reinforcement signal* (Matarić, 1998; Parker & Touzet, 2000). In this case, a distributed learning algorithm rewards a single agent based on the local assessment of its contribution to the swarm performance. Although this type of reinforcement signal is immediate and exploits the inherent parallelism of the swarm, it often provides only a noisy estimation of the swarm performance. The more limited and local the communication and perception capabilities are (e.g., in extreme cases no communication at all and very short-range sensors), the noisier the local assessment is, due to partial perception. In other words, for an individual agent it is extremely difficult to assess whether the success of its actions has been achieved because of its own merit or because the other agents around it created a favorable situation.

A sort of solution between a fully heterogeneous swarm and a homogeneous one is represented by a caste-based swarm. In this scenario, distributed, caste-centered learning can be implemented where each caste interprets a reinforcement signal (either global or collected by its members, i.e., *caste-centered*) and adapts its control parameters or rules in a coordinated way. Further information about variation on multi-robot rewards function can be found in Balch (2002b).

In this paper, we consider different task constraints and types of reinforcement signals, and quantitatively measure the performance, diversity, and specialization of a swarm. If agents do not belong to the same caste, they are not allowed to communicate among themselves but can receive global reinforcement signals broadcast from a central unit. Within a caste, agents are networked and coordinate their adaptation process so that caste homogeneity is maintained.[2] We support the discussion first with a concrete collaboration experiment concerned with pulling sticks out of the ground, and then with its generalized versions, where the collaboration is extended to k sequential or parallel operations within a certain time window—analogous to pulling longer or heavier sticks. We show that specialization can arise in all versions of experiments as a function of task constraints on the environment and the swarm, no matter which type of reinforcement signal is used. As long as the diversity in agents brings advantage to the swarm performance, learning can drive the system to become specialized.

Finally, as we will further outline in Section 6, the specialization analysis and synthesis presented in this paper could be applied to completely different experiments using distributed, embedded systems whose units are supposed to solve a task in a collaborative way. For example, self-locomoted sensor nodes characterized by pseudo-random movement patterns, endowed with local communication capabilities, and engaged in a monitoring task over a well-delimited area, could be represented in the same abstracted way as the robots engaged in the stick-pulling experiment. In this case, the metric used to assess the swarm performance could be related to the number of successful event detections reported by the swarm to a base-station knowing that, before emitting an alarm signal, at least k nodes of the swarm should collectively agree to have detected the same event.

2 Diversity and Specialization

Traditionally, swarm systems have been classified on a bipolar scale as either heterogeneous or homogeneous, depending on whether any of the agents differ from the others. This view is limiting because it does not permit a quantitative comparison between heterogeneous systems. In this section, we introduce metrics to quantitatively measure the diversity and specialization of a swarm system, which can enable the investigation of issues such as the impact of diversity on swarm performance and the impact of task constraints on specialization.

2.1 Diversity Measure

The essential idea behind the diversity measure is to cluster similar agents according to a problem-specific difference measure d and look at the pattern they form in the feature space.

An intuitive and straightforward way is to use the number of clusters the agents form in the feature space as the diversity measure. Each cluster is believed to be a group of specialists, and essentially, the more clusters we have, the more diverse the system is. However, in general cases where the difference measure is not binary, finding the "right" number of clusters for a data set is often ill-posed. Depending on different criteria, one number may or may not be better than another. Li (2002) used a heuristic criterion (maximizing the difference between the inter-cluster and the intra-cluster distances) to select the "optimal" clustering.

Instead of just using the number of clusters, Balch (2002a) introduced Shannon's information entropy to measure the diversity and named it the *simple social entropy*. For a system $\Re$ with m clusters of agents, we use p_i to denote the portion of agents in the i-th cluster (i.e., the cluster size divided by the swarm size). The simple social entropy is defined as

$$H(\Re) = -\sum_{i=1}^{m} p_i \log p_i. \tag{1}$$

Input: a swarm system $\Re = \{r_1, r_2, \ldots, r_n\}$ of size n; a difference measure d.
For different level h, the C_u clustering algorithm does:

1. Initialize n clusters with cluster $c_i = \{r_i\}$;
2. For each c_i: for each r_j: If $d(r_j, r_k) \leq h$ for all r_k in c_i, add r_j to cluster c_i;
3. Discard redundant clusters;
4. Calculate p_i and the entropy $H(\Re, h)$. Note that when r_j belongs to s clusters including c_i, its contribution to p_i is $1/sn$.

Return $\int_0^\infty H(\Re, h)\, dh$ as the hierarchic social entropy.

Figure 1: Hierarchic clustering and social entropy.

Here we still have to circumvent the problem of "optimal" clustering since even the same clustering algorithm may give different sets of p_i when different algorithmic parameters are used. A solution to this problem is to select a priori the clustering algorithm and average the entropy over all possible parameters characterizing the chosen algorithm. Balch combined numerical taxonomic classification techniques and the C_u clustering algorithm (Jardine & Sibson, 1971) to order agents hierarchically according to the spatial structure of the system and a taxonomic level parameter h (see Figure 1). The simple social entropy can be then transformed in the so-called *hierarchic social entropy* by integrating over the parameter h:

$$D(\Re) = \int_0^\infty H(\Re, h)\, dh. \tag{2}$$

It is worth noting that the C_u algorithm allows for cluster overlapping. As a consequence, considering the specific way for calculating p_i illustrated in Figure 1, when clusters overlap the entropy may increase when h increases.[3]

After some preliminary tests, we have verified that the hierarchic social entropy achieves a stabler and more meaningful diversity measure than that based on the number of clusters, especially when the swarm size is small. This is probably due to the fact that the social entropy encodes much more details of the feature space (such as the spatial distribution of the clusters). Thus we use the hierarchic social entropy (2) as our diversity measure in this paper.

2.2 Specialization Measure

Specialization means more than just being diverse. While diversity means difference among individuals no matter whether the difference is good or bad in respect to the swarm performance, specialization, with the definition "structural adaptation of a part to a particular function," also means adaptation in order to fit. When diversity is obtained via an iterative process such as learning or evolution, other reasons (e.g., noise in the replication mechanism) can also cause the system to become diverse. However, a system becomes specialized when, given specific constraints of viability or survival at the agent level, its diversity is evoked for better swarm performance. In short, *specialization is the part of diversity that is demanded for better performance*. Accordingly, a specialization metric should measure the part of diversity that enhances the performance.

When looking at a swarm system statically, it is impossible to identify the part of diversity that corresponds to the performance improvement. We have to couple the system with a dynamic process where its performance and diversity can change and interact. If the performance generally increases with higher diversity, the system benefits from being more diverse than the initial status, and the degree of specialization should increase accordingly; otherwise, if the greater diversity does not help the performance, the degree of specialization should decrease. That is, specialization can be measured along a dynamic process as a result of the correlation between the diversity and the perfor-

Figure 2: Physical set-up for the original stick-pulling experiment. Collaboration between two robots is necessary to pull a stick out of the ground.

mance. If we assume the system starts from a homogeneous setting with no diversity or specialization, and the diversity D and the swarm performance R change with time as correlated random variables, the correlation coefficient between D and R naturally acts as the percentage of specialization in diversity. More formally, the degree of specialization can be defined as

$$S = \mathrm{corrcoef}(D; R) \times D. \tag{3}$$

Note that our specialization measure S is negative when D and R are negatively correlated.

We should note that there are other definitions of specialization. O'Donnell and Jeanne (1990) defined specialization of an individual as the entropy of the proportions of its activity. Low entropy means that an individual focuses on less activities, i.e., it is specialized. Gautrais, Theraulaz, Deneubourg, and Anderson (2002) considered a similar metric that is the complement of the frequency of an individual changing tasks. The lower the frequency of changing tasks is, the higher the individual is considered specialized. However, their definitions focus on the *specialization of individuals*, while we consider specialization as a quantity of the whole swarm, which is more coherent with the diversity measure we use in this paper.

3 Stick-Pulling Experiments

Martinoli and Mondada (1995/1997) and later Ijspeert, Martinoli, Billard, and Gambardella (2001) investigated collaboration in non-communicating robots engaged in a stick-pulling experiment (Figure 2). We call their experiment the original one since we will abstract and generalize it later in Subsection 3.2.

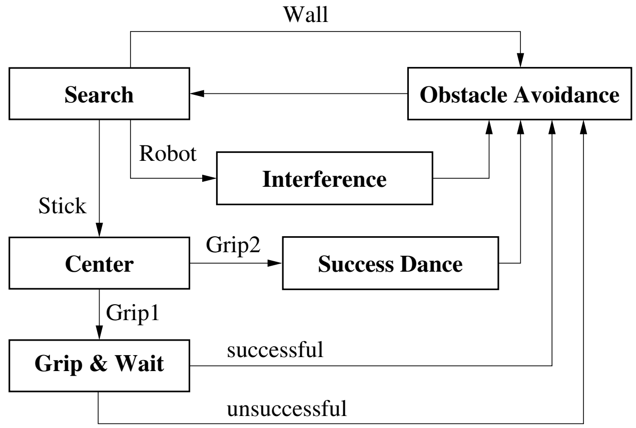


Figure 3: FSM representing the robot's controller. Transitions between states are triggered by sensory measurements.

3.1 Original Stick-Pulling Experiment

In the original experiment, robots equipped with gripper turrets and proximity sensors search a circular arena and pull sticks out of the ground. The stick length has been chosen so that a single robot is incapable of pulling a stick out completely on its own, but collaboration between two robots is sufficient for solving this task. Each robot is characterized by a *gripping time parameter* (GTP) which is the maximal length of time that a robot waits for the help of another robot while holding a stick.

The behavior of a robot is determined by a simple program (Figure 3). The default behavior is searching for sticks, i.e., wandering in the arena until an object is detected. If a stick is detected, the robot pulls it up and determines whether another robot is already holding it by measuring the elevation speed of the gripper arm. If the elevation is fast, there is no other robot holding the stick and we call such a grip *grip1*. Otherwise, the robot assumes that another robot is already holding the stick and therefore "braking" the elevation. Such a grip is named *grip2*.

After a robot makes a grip1, two cases can occur: either a second robot helps the first one before the GTP expires (we call this a *successful collaboration*) or the first robot times out and resumes the search for sticks. The specific GTP values play a crucial role in the overall *stick-pulling rate* (defined as the number of sticks pulled out per unit time) which is the metric adopted in previous papers[4] (Ijspeert et al., 2001; Li et al., 2002; Martinoli, Easton, & Agassounon, 2004) and this one for the swarm performance. To ensure the stick-pulling rate is reliably measured, experiments usually take a long time and sticks are inserted back by the experimenter after they have been completely pulled out.

We use the microscopic model (Ijspeert et al., 2001) as the simulation platform, which represents agents as separate probabilistic finite-state machines (PFSM). The flowchart of a PFSM is based on the blueprint of the corresponding real robot controller and its transition probabilities are computed using simple geometric considerations and systematic experiments with one or two real robots. Unlike macroscopic models (see Martinoli et al., 2004, for the same experiment) which intrinsically assume agents can be clustered into certain castes, microscopic models allow us to study all levels of swarm diversity, from homogeneity to full heterogeneity, since each agent is represented by a separate PFSM. Furthermore, this model allows for quantitatively accurate predictions while being four or five orders of magnitude faster than other popular simulation tools such as sensor-based embodied simulations.[5] Therefore, although we have not tested our results using real robots or realistic simulations, we believe that their validity is not limited to abstract agents.

3.2 Generalized Stick-Pulling Experiments

The strict collaboration property of the stick-pulling task has a major influence on swarm diversity and specialization. In order to emphasize this effect, we abstract and generalize the original experiment so that a successful collaboration requires now k (> 2) robots instead of just two.

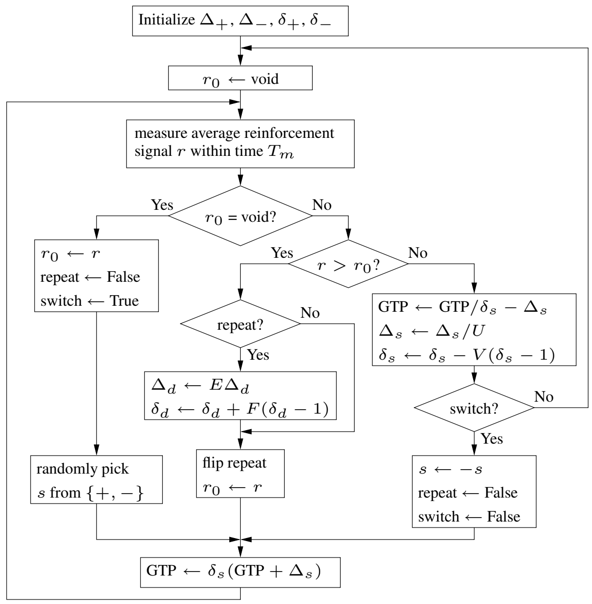
Sequential Collaboration: Pulling Longer Sticks One way to extend the original experiment is to assume longer sticks so that one robot can only pull a stick up by $1/k$ of its length. k consecutive grips, which may be called grip1, grip2, …, and grip$k$, respectively, are thus needed for pulling out a stick entirely. If the robot currently holding the stick times out, it will drop the stick so that further robots will have to start over from grip1. We call this type of collaboration required for pulling longer sticks *sequential collaboration*. Note that we do not really need more than two robots in order to complete the task. Theoretically, two robots with very large GTPs are able to pull out sticks of any length but inefficiently, if they help each other alternately.

Parallel Collaboration: Pulling Heavier Sticks Another way to extend the original experiment is to suppose the sticks are shorter but heavier so that one robot is too weak to lift a stick up. Exactly k robots are needed simultaneously to lift a stick and pull it out. When a robot finds a stick, it grips the stick until timing out or until there are enough robots to lift the stick, whichever comes earlier. Robots do not reset their timers when a new robot joins the pulling. In contrast with the sequential case, unless all the robots currently holding the same stick time out, the pulling process does not need to be restarted from scratch. We call this type of collaboration *parallel collaboration*.

4 Learning Algorithm

We proposed and tested an adaptive line-search algorithm and found that it could achieve near-optimal performance under different conditions in the original stick-pulling experiment (Li et al., 2002). In this paper, we use the same learning algorithm for both the original and the generalized stick-pulling experiments. As mentioned before, the algorithm has been designed to work properly in both centralized and distributed learning scenarios. It is worth recalling that we are not trying to solve a global optimization problem off-line based on our probabilistic model but rather to emulate what would happen with different realistic scenarios, including that where each agent is endowed with simple learning but no communication capabilities at all.

The algorithm is depicted in Figure 4. An agent (or a caste, or the swarm as a whole) first tries to update its GTP in a randomly chosen search direction s. It then keeps the new GTP for a small period T_m and monitors the performance by averaging the reinforcement signal it receives. If the performance is better than before, the agent continues in that direction; otherwise, it undoes the last GTP update and switches to the other direction. In addition to adapting the GTP, the search step (Δ_s and δ_s) also varies. When the same direction has been selected twice, the search step



Algorithmic variables:

	Range	Description
s	$\{+,-\}$	search direction
Δ_+	$[2, 60]$	GTP offset (sec)
Δ_-	$[-60, -2]$	GTP offset (sec)
δ_+	$[1.1, 5]$	GTP factor
δ_-	$[0.2, 0.9]$	GTP factor
r_0	$\{\text{void}\} \cup \mathbb{R}$	previous performance
r	$\mathbb{R}$	current performance
repeat	Boolean	reinforcement flag
switch	Boolean	direction flag

Algorithmic parameters:

	Value	Description
T_m	2400	averaging period for reinforcement signal (sec)
E	1.9	GTP offset enlarge factor
F	0.3	GTP factor enlarge ratio
U	2	GTP offset shrink divider
V	0.5	GTP factor shrink ratio

Figure 4: Flowchart of the learning algorithm and its parameters.

for that direction is increased (controlled by parameters E and F) in order to speed up the convergence. When the observed performance oscillates, which implies that the current GTP is close to the optimal one, the search step is accordingly decreased (controlled by parameters U and V) to stabilize the performance.

Although at first glance this algorithm may appear similar to a gradient-descent method, it has been designed in a way that it neither requires the derivative to be calculated nor assumes continuity in the search space. Furthermore, the algorithm works well in a noisy environment, since the monitoring of performance implies an averaging process which low-pass filters noise in the reinforcement signal, and the occasional picking of a random search direction helps to escape from unfavorable situations caused by the noise. Finally, we did not implement a classical momentum term since the fitness landscape of the stick-pulling dynamics (see Figure 5) does not present any local maxima other than those created by noise. Their time-variable characteristics, still present after filtering, usually help the algorithm to avoid deadlock situations.

We use both types of reinforcement signals with the learning algorithm. The local reinforcement signal rewards an agent when it makes a successful collaboration, i.e., when it completely pulls out a stick or passes the stick to another agent. The global reinforcement signal is the swarm performance. The two types of reinforcement signals "align" well in the original experiment as well as its parallel extension since a successful collaboration means exactly a stick pull-out and vice versa. However, in sequential cases, a successful collaboration only contributes to but may not finally result in a stick pull-out, and without the central unit broadcasting or other explicit communication forms among agents, an agent never knows its true contribution unless it does the final grip. Thus the local reinforcement signal in sequential cases is not aligned with the global one, which leads to a clear system bifurcation in terms of specialization (see Subsection 5.3).

5 Results

All the experiments we conducted started from a homogeneous system, i.e., the same initial GTP for all agents. The experiments lasted long enough for the learning to stabilize. Swarm performance and diversity were recorded periodically along the experiments so that specialization could be measured via Equation (3). We simulated 50 runs for each initial GTP and plotted the mean absolute or relative swarm performance, diversity, and specialization over the runs. The error bars represent one standard deviation per run in Figure 5, and the standard error in all other figures.

Li (2002) suggested using a difference measure of logarithmic form since both the performance and the logarithm are less sensitive to GTP changes when GTP is large. That is, for two agents with GTPs g_1 and g_2 respectively, the difference between them is $d = |\log g_1 - \log g_2|$. This difference measure is used in all of our experiments.

5.1 Validation of the Learning Algorithm

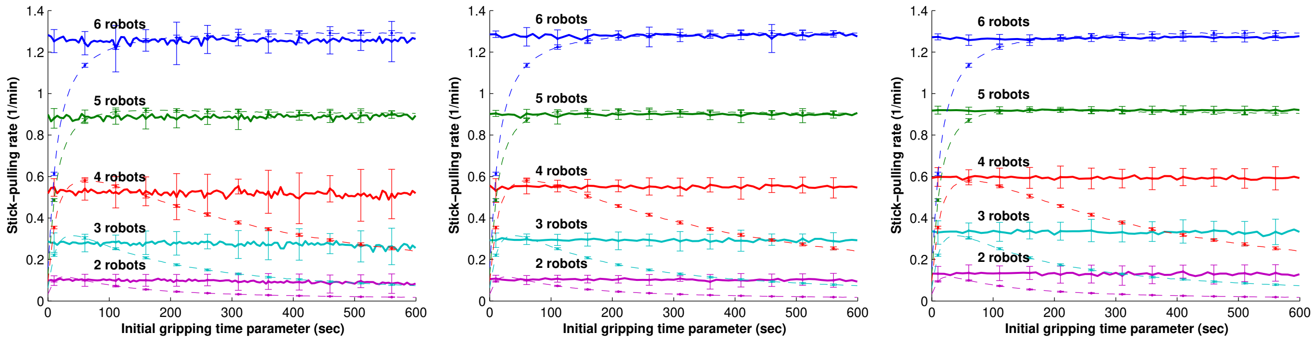
Before introducing results in diversity and specialization, we would like to compare the performance of hardwired agents with learning agents engaged in the original stick-pulling experiment. The experimental settings are the same as in some previous papers (Ijspeert et al., 2001; Li et al., 2002; Martinoli et al., 2004), i.e., 2 to 6 robots and 4 sticks in an arena of 40 cm in radius. Figure 5 contrasts the learned performance with that obtained by systematically varying the GTP value for this particular stick-pulling scenario. The performance of the hardwired swarms (dashed curves) is compared with that of a homogeneous learning swarm (i.e., a single GTP parameter characterizes the whole swarm) using global reinforcement signal (Figure 5a and b), and a heterogeneous swarm (i.e., each agent is characterized by its own GTP) using local reinforcement signal (Figure 5c). Figure 5a has been generated by using a shorter averaging period ($T_m = 600$ s) than the default one ($T_m = 2400$ s, see Figure 4). We can immediately see that the characteristics of the low-pass filter embedded in the algorithm plays a relevant role in the learning performance: a noisier reinforcement signal translates into a higher standard deviation and a lower mean of the learned swarm performance.

For these specific experimental constraints, the system characterized by a homogeneous swarm with a fixed GTP exhibits quite different dynamics depending on the robot/stick ratio. Roughly speaking, when there are more robots than sticks, the stick-pulling rate increases monotonically with the GTP until reaching a plateau corresponding to the optimal rate for homogeneous swarms. In other words, with the specific system constraints of the original stick-pulling experiment, since there are always robots "free" to help, waiting very long is a good strategy for robots holding sticks. On the other hand, when the number of robots is equal to or smaller than that of sticks, waiting in vain for a very long time may generate *deadlock* situations where every robot holds a different stick and waits for help. We thus define the *deadlock threshold* as the maximal number of robots that could still incur into a deadlock situation.[6] Previous research showed that specialization was desired particularly in this situation (Ijspeert et al., 2001; Li et al., 2002), i.e., when the swarm size is below the deadlock threshold.

The stick-pulling rate of the learning swarms instead consistently achieved the same level of performance independent of the initial GTP. Depending on their low-filter parameter setting, the homogeneous learning swarms almost reached the maximal performance levels obtained by the hardwired agents, while the heterogeneous ones often outperformed those levels, especially for swarm sizes equal to or smaller than four. As we will see in the following subsection the increased performance of the heterogeneous swarms is due to the specialization of their agents.

5.2 Specialization in the Original Stick-Pulling Experiment

In this subsection and the following one, we will systematically analyze and compare the swarm performance, diversity, and specialization for the same system constraints. Since this paper focuses on specialization and its relationship to diversity and performance, rather than plotting the absolute swarm performance, we have considered the relative performance in comparison to a homogeneous learning swarm. Therefore, for each swarm size, we normalized the performance after learning of a given heterogeneous swarm by dividing it by that of the corresponding homogeneous swarm. Homogeneous swarms, of course, do not appear on diversity and specialization plots since their diversity and specialization are null. In all the plots we compare three different learning conditions under which the heterogeneous swarms become specialized: fully heterogeneous swarms under the local reinforcement signal, fully heterogeneous swarms under the global reinforcement signal, and swarms



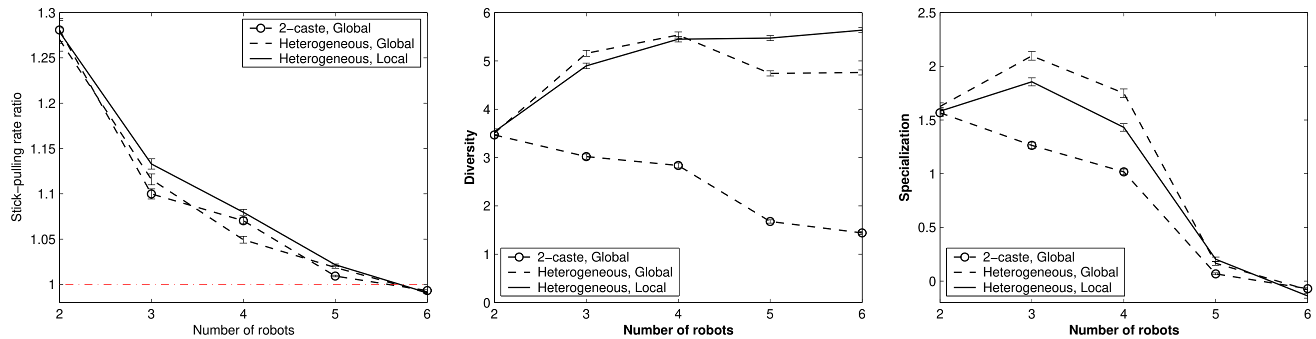
(a) (b) (c)

Figure 5: Performance after learning in the original stick-pulling experiment. The dashed curves represent the performance of homogeneous swarms with a fixed GTP (no learning); the solid curves show that of (a) homogeneous swarms using the global reinforcement signal and a shorter period for averaging the reinforcement signal ($T_m = 600\,\mathrm{s}$), (b) homogeneous swarms using the global reinforcement signal and default $T_m = 2400\,\mathrm{s}$, and (c) heterogeneous swarms after learning under the local reinforcement signal.

partitioned in castes under the global reinforcement signal.[7]

Figure 6a confirms the results in Figure 5: 2-caste and fully heterogeneous swarms achieve better performance for smaller swarm sizes. Diversity has a reduced impact on performance when there are more robots than sticks and heterogeneous swarms achieve even slightly worse results after learning than homogeneous ones with 6 (or more) robots. The explanation is quite simple: the advantages of specialization fade out with more robots than sticks and the increased search space for heterogeneous swarms in combination with noise and/or credit-assignment problem reduces the optimization power of the learning algorithm. While the diversity measure (Figure 6b) produces roughly flat curves, the specialization measure (Figure 6c) outlines clearly this phenomenon: further diversity does not necessarily improve the performance with more robots than sticks, and it is even negatively correlated with 6 (or more) robots.

By comparing the different learning constraints, we can also formulate the following remarks. First, the fully heterogeneous swarms achieve at least the same level of performance as the 2-caste ones. This is an interesting result since it tells us that constraining the learning space using a priori information, either collected with systematic studies (Ijspeert et al., 2001) or based on intuitive reasoning, does not necessarily help a swarm to perform better. Indeed, although with these particular experimental settings the difference in performance between 2-caste and fully heterogeneous swarms is minimal, it is interesting to note that for odd swarm sizes the performance of the 2-caste swarms are lower than those of the fully heterogeneous swarms under both types of reinforcements (see Figure 6a, dashed curves). In other words, the learning algorithm has a harder time compensating unbalanced castes with a reduced number of GTPs than distributing the appropriate individual GTPs to all the members of the swarm. Second, constraining the learning process reduces also the possible degree of specialization of a swarm, as shown in Figure 6c. In other words, we could consider the a priori caste partitioning as a "low-pass filter" for the swarm specialization. Namely, a larger search space does not prevent fully heterogeneous swarms from increasing or reducing their level of specialization when the system constraints prompt it in one direction or the other (see Figure 6c). Third, since the local reinforcement signal is noisier than the global one, we expect that under the global reinforcement signal truly specialized robots generate a larger portion of the diversity. This is validated in Figure 6c since the diversity (and specialization) under the global reinforcement signal dropped



(a) (b) (c)

Figure 6: Results of the original stick-pulling experiments for different team sizes and learning constraints (homogeneous and 2-caste teams under the global reinforcement signal, heterogeneous teams under the global and local reinforcement signals). (a) Relative performance after learning. (b) Swarm diversity. (c) Swarm specialization.

faster than under the local reinforcement signal when the specialization was less relevant.

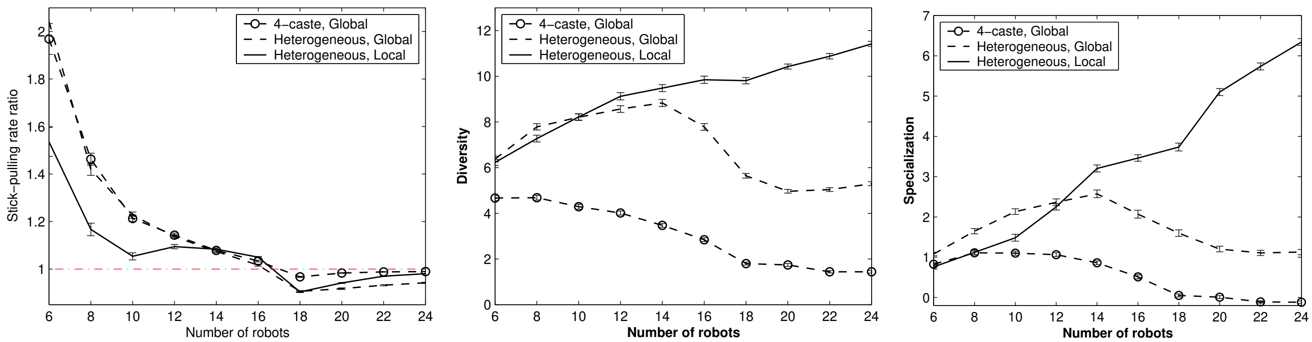
5.3 Specialization in the Generalized Stick-Pulling Experiments

In order to accommodate more robots required by the generalized experiments, we used a larger arena of 80 cm in radius, 16 sticks, and 6 to 24 robots. We simulated the generalized experiments with k from 3 to 5 and selected the case $k = 4$ for the plots reported in this paper (see Figures 7 and 8). We also increased the averaging period T_m to 36,000 s since the probability of successful collaboration under such experimental and learning conditions becomes smaller.

We first consider the results of the generalized sequential experiment (Figure 7). First, while in the original stick-pulling experiment the local and global reinforcement signals were aligned, this is no longer the case here. As a consequence, results in performance, diversity and specialization diverge more clearly as a function of the reinforcement signal used. Second, the deadlock threshold, which is now set at 16 robots, appears again to play a major role in the system dynamics. Diversity brings advantages to the swarms in terms of performance with swarm sizes smaller than 16. However, the local reinforcement signal, due to more limited local perception in a larger arena and a bigger swarm size as well as the non-alignment between this signal and the swarm performance, is much noisier under these conditions than in the original stick-pulling experiments (compare Figure 6a and Figure 7a). This makes distributed learning with a local reinforcement signal more difficult—the performance is not as good as that obtained using the global reinforcement signal (see Figure 7a). Finally, since we have many more robots for the generalized experiment, constraining the learning process with 4 castes reduces the search space dramatically. Learning benefits from the smaller search space even though it cannot explore the fully heterogeneous solutions, and the performance of the 4-caste swarms is actually the best overall: when the swarm size is smaller than the deadlock threshold, it is similar to that of the fully heterogeneous swarms with global reinforcement signal, and when the swarm size is larger than the deadlock threshold, it is close to that of the homogeneous swarms.

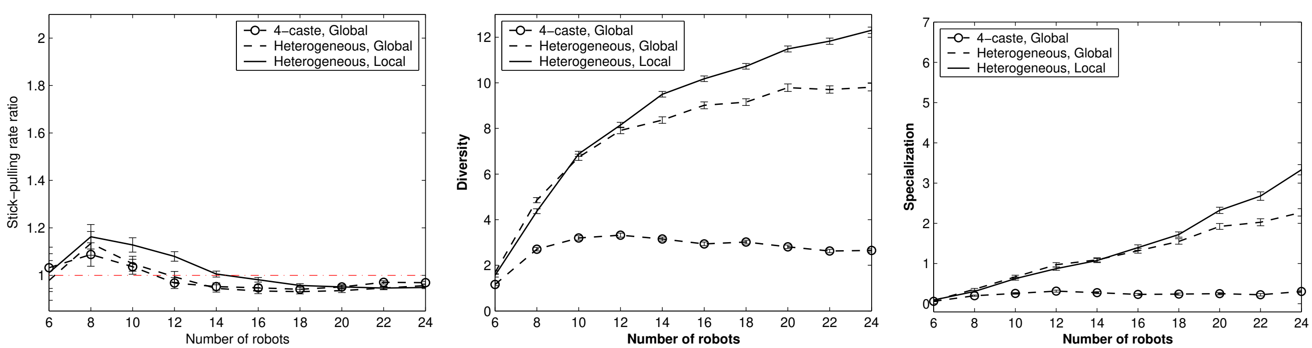
Before looking at the results for parallel experiments, we expected that the specialization in parallel cases would be higher than that in sequential cases. Our arguments were:

- In sequential cases, the requirements for robots doing grips before grip$k$ are similar—their GTPs must be large enough to allow the next robot to grip the stick and take over the extraction process. In parallel cases, k different GTP values may be instead required—robots doing grip1 need the largest GTP and robots doing



(a) (b) (c)

Figure 7: Results of the sequential stick-pulling experiments with $k = 4$ for different team sizes and learning constraints. (a) Relative performance after learning. The horizontal line at level one represents the learned performance of homogeneous teams. (b) Swarm diversity. (c) Swarm specialization.



(a) (b) (c)

Figure 8: Results of the parallel stick-pulling experiments with $k = 4$ for different team sizes and learning constraints. (a) Relative performance after learning. (b) Swarm diversity. (c) Swarm specialization.

grip$k$ need the smallest GTP.

- With the same number of sticks, the parallel collaboration essentially requires more robots working simultaneously. We know from the original stick-pulling experiment that specialists are more likely to arise if there are insufficient robots compared to the number of sticks.

However, our predictions were confirmed only with large swarm sizes (say, 18 robots or higher) and the global reinforcement signal. An investigation of the learned GTPs shows that when the number of robots is small in parallel cases, all robots have similar GTPs ($\sim 300\,\mathrm{s}$) and the diversity is low. This gives us hints about the seemingly weird phenomenon.

Note that if there are t sticks in the arena, the deadlock threshold is $(k-1)t$ in parallel cases, larger than the threshold in sequential cases if $k > 2$. Our experience with previous experiments made us believe that specialization is high any time the number of robots is less than the deadlock threshold, which is not always true. Just as in a company having many more jobs than employees, when the deadlock threshold is much higher than the number of robots, each robot tends to have multiple roles, as every employee has to take multiple jobs. Since a robot has only one GTP value, trying to specialize into too many directions just makes all GTPs similar and results in a low diversity, especially when k is large in parallel cases. Under these conditions, the much noisier local reinforcement signal achieves better performance than the global reinforcement signal (see Figure 8a) since it helps to break the equilibrium among possible different specialization directions and drives the system away from homogeneity.

For both generalized experiments, when the number of robots is larger than the deadlock threshold, the decreasing of specialization was observed with the global reinforcement signal.[8] It was, however, difficult to precisely predict what swarm sizes would have maximally benefited from specialization under specific system and learning constraints. Finally, as was the case in the original stick-pulling experiment, the 4-caste solution exerts its low-pass filtering function on specialization in all generalized stick-pulling experiments.

6 Discussion

In this section we discuss some advantages and drawbacks of our method, in terms of synthesizing and analyzing specialization in a collaborative swarm system. In particular, instead of the specific stick-pulling case study, we would like to consider the usefulness of this method from a general system optimization perspective. We summarize our method using the following four steps.

1. Given an embedded swarm system, generate a quantitatively faithful microscopic model that captures all the system parameters of interest.
2. Select the subset of parameters which can differ on individuals and might play crucial roles in the total swarm performance.
3. Let a given machine-learning algorithm search in the space defined by this reduced set of parameters.
4. Analyze results according to the diversity and specialization metrics presented in Section 2.

As every other method, the one reported here also shows strengths and limitations. A first strength of our method is that it is fast and selective. Thanks to the modeling abstraction, only parameters of interest are considered in the search space. From a computational cost point of view the gain is double-sided: the search is focused on a smaller parameter space and the evaluation of candidate solutions is achieved several orders of magnitude faster than other popular simulation tools used in mobile robotics (see also Martinoli et al., 2004). A further strength is that the method is flexible and generalizable. We have shown in this paper how easy it is to transform a given concrete physical experiment into more generalized ones when the description is sufficiently abstracted. We therefore believe that the methods introduced for learning and measuring specialization can be easily applied to completely different tasks and experimental scenarios. For instance, a collaboration constraint similar to that of the generalized parallel stick-pulling experiment could arise in a monitoring experiment where self-locomoted sensor nodes have to detect events happening in a given region of space. The swarm performance could be assessed as the number of successful event detections reported by the swarm to a base station knowing that, before emitting an alarm signal, at least k nodes of the swarm should have detected the same event. Other examples of tasks which could benefit from the methods proposed here could be concerned with foraging (Labella, Dorigo, & Deneubourg, 2004), collective transportation (Kube & Bonabeau, 2000), and aggregation (Agassounon & Martinoli, 2002).

Among the limitations of our method, we can certainly mention that the optimization and specialization quality depends on the modeling and the learning algorithm. If the underlying model is quantitatively correct, we can easily transport the optimization results to the original real embedded system. Otherwise a further optimization process might be required on real hardware. Furthermore, the learning algorithm presented in this paper is certainly a good baseline but it could be replaced with more powerful ones according to the size of the search space and the amount of noise characterizing the optimization problem. As we mentioned before, the choice of the learning algorithm depends also on the type of optimization one is interested in (i.e., on-line or off-line) and whether or not, in case of on-line optimization, the algorithm is suited for the computational and communication capabilities of the real embedded agents. Furthermore, the learning algorithm (including its algorithmic parameters) influences the quantitative specialization results. Although we did not carry out a full sensitivity analysis on the robustness of our specialization measure, we have noticed that certain specialization results in the presence of noise (see for instance Figure 7c, the increase of specialization after 18 robots with heterogeneous swarms and local reinforcement signal) are quantitatively mitigated or stressed by the choice of, for instance, the length of the time windows used for measuring specialization or the total learning time. Future work should assess precisely how much dependence our specialization measure shows on the underlying adaptation process. In contrast, the diversity measure, being a static one, did not show any significant quantitative changes as a function of the parameter of the learning algorithm. Finally, as we have seen in the generalized stick-pulling experiments, specialization results are not always easy to interpret. They appear to be more meaningful than those obtained via a diversity measure in order to decide whether or not a swarm should be heterogeneous under certain system constraints. Results also appear to confirm the fact that our intuition might constrain the optimization process in negative ways, for instance by introducing a fixed number of castes. We believe that it is premature at this point to draw any general conclusions about the use of pre-established castes instead of letting an adaptation process discover the number of castes and their corresponding parameters for us. We need to apply the method presented in this paper to other case studies and try to pin down general principles characterizing the influence of the number of castes on the quality of the final solution.

7 Conclusion

This paper presented our initial effort to measure specialization in collaborative swarm systems. Specialization is a mixed concept of both diversity and adaptation. We define specialization as the part of diversity that is induced by the need of performance improvement. Our experiments with the original and generalized stick-pulling experiments showed that specialization, while being more sensitive to the underlying adaptation process, was more meaningful than diversity when properties related to performance and learning were under study. Our results validated some of our intuitions about specialization in these collaborative experiments but also revealed some properties that we did not see at first.

In the discussion, we have outlined some of the general properties of our method since we believe that it can be applied to other tasks and other swarm systems. Future work will explore further the robustness of this method in other experimental scenarios and the dependence of the specialization measure on the dynamic process in which diversity and swarm performance interact.

Notes

[1]Notice that it is only the adaptation process that happens in a centralized way. Agents still act autonomously.

[2]This is just a possible illustration of how we could implement a caste-centered adaptation. In principle, the adaptation algorithm could also run on the central unit (which in turn will keep track of the different caste parameters) or on a caste leader.

[3]For example, there are three agents r_1, r_2, and r_3 in the swarm system. $d(r_1, r_2) = 6$, $d(r_1, r_3) = 4$, and $d(r_2, r_3) = 3$. When h is 3, the clustering algorithm gives $c_1 = \{r_1\}$ and $c_2 = \{r_2, r_3\}$. Thus the entropy is $H(\frac{1}{3}, \frac{2}{3})$. When h is increased to 4, r_3 is also included in c_1, and the entropy becomes $H(\frac{1}{2}, \frac{1}{2})$, which is larger than the entropy at $h = 3$.

[4]The *collaboration rate* (the number of successful collaborations per unit time) was in fact used in the previous papers. It is equivalent to the stick-pulling rate when exact one successful collaboration is required for a stick pull-out.

[5]Some environmental conditions, such as the stick positions and the initial placement of the robots, have been abstracted in the microscopic model. However, considering the probabilistic nature of trajectories and collisions, such abstraction does not result in statistically significant differences between the microscopic model and the realistic simulation or real robots. See Martinoli et al. (2004) for a detailed discussion on the impact of such abstracted conditions, as well as different experimental and modeling levels.

[6]As demonstrated in Martinoli et al. (2004) the exact location of the bifurcation point on the "number of robots" axis depends not only on the robot/stick ratio but also on a physical interaction parameter regulating the collaboration mechanism. Since we kept this physical interaction parameter constant in all the experiments of this paper, for sake of simplicity, we neglect such detailed explanation.

[7]We also carried out experiments with teams partitioned in castes in combination with a caste-centered reinforcement signal. Since the results achieved under this learning condition were very similar to those obtained with the same teams and a global reinforcement signal, we omit them in the plots for sake of clarity.

[8]For parallel cases, since the threshold is much higher, we verified this with 2 sticks, 4 to 9 robots, and $k = 4$.

Acknowledgments

We would like to thank Carl Anderson, Tucker Balch, Christopher Cianci, and the anonymous reviewers, for their valuable input and suggestions. This work has been principally supported by the Caltech Center for Neuromorphic Systems Engineering under the US NSF Cooperative Agreement EEC-9402726 and the Northrop Grumman Corporation Foundation. Alcherio Martinoli is currently sponsored by a Swiss NSF professorship.

References

Agassounon, W., & Martinoli, A. (2002). Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In C. Castelfranchi & W. L. Johnson (Eds.), *Proceedings of the first international joint conference on autonomous agents and multiagent systems* (pp. 1090–1097). New York: ACM Press.

Balch, T. (2002a). Measuring robot group diversity. In T. Balch & L. E. Parker (Eds.), *Robot teams: From diversity to polymorphism* (pp. 93–135). Natick, MA: A K Peters.

Balch, T. (2002b). Taxonomies of multirobot task and reward. In T. Balch & L. E. Parker (Eds.), *Robot teams: From diversity to polymorphism* (pp. 23–35). Natick, MA: A K Peters.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press.

Gautrais, J., Theraulaz, G., Deneubourg, J.-L., & Anderson, C. (2002). Emergent polyethism as a consequence of increased colony size in insect societies. *Journal of Theoretical Biology*, *215*, 363–373.

Hayes, A. T., Martinoli, A., & Goodman, R. M. (2003). Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, *21*, 427–441.

Ijspeert, A. J., Martinoli, A., Billard, A., & Gambardella, L. M. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, *11*, 149–171.

Jardine, N., & Sibson, R. (1971). *Mathematical taxonomy*. New York: Wiley.

Kube, C. R., & Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, *30*, 85–101.

Labella, T. H., Dorigo, M., & Deneubourg, J.-L. (2004). Efficiency and task allocation in prey retrieval. In A. J. Ijspeert & M. Murata (Eds.), *Proceedings of the first international workshop on biologically inspired approaches to advanced information technology* (pp. 32–47). Berlin: Springer.

Li, L. (2002). *Distributed learning in swarm systems: A case study*. Unpublished master's thesis, California Institute of Technology, Pasadena, CA.

Li, L., Martinoli, A., & Abu-Mostafa, Y. S. (2002). Emergent specialization in swarm systems. In H. Yin, N. Allinson, R. Freeman, J. Keane, & S. Hubbard (Eds.), *Lecture notes in computer science: Vol. 2412. Intelligent data engineering and automated learning — IDEAL 2002* (pp. 261–266). Berlin: Springer.

Martinoli, A., Easton, K., & Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *International Journal of Robotics Research*, *23*, 415–436.

Martinoli, A., & Mondada, F. (1997). Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In O. Khatib & J. K. Salisbury (Eds.), *Lecture notes in control and information sciences: Vol. 223. Experimental robotics IV* (pp. 3–10). Berlin: Springer. (Original work published 1995)

Matarić, M. J. (1998). Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical Artificial Intelligence*, *10*, 357–369.

Murciano, A., Millán, J. del R., & Zamora, J. (1997). Specialization in multi-agent systems through learning. *Biological Cybernetics*, *76*, 375–382.

O'Donnell, S., & Jeanne, R. L. (1990). Forager specialization and the control of nest repair in *Polybia occidentalis* Olivier (Hymenoptera: Vespidae). *Behavioral Ecology and Sociobiology*, *27*, 359–364.

Parker, L. E., & Touzet, C. (2000). Multi-robot learning in a cooperative observation task. In L. E. Parker, G. Bekey, & J. Barhen (Eds.), *Distributed autonomous robotic systems 4* (pp. 391–401). Berlin: Springer.

Versino, C., & Gambardella, L. M. (1997). Learning real team solutions. In G. Weiß (Ed.), *Lecture notes in artificial intelligence: Vol. 1221. Distributed artificial intelligence meets machine learning: Learning in multi-agent environments* (pp. 40–61). Berlin: Springer.

Wolpert, D. H., & Tumer, K. (2001). Optimal payoff functions for members of collectives. *Advances in Complex Systems*, *4*, 265–279.