Efficient Multi-foraging in Swarm Robotics

Alexandre Campo and Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
`{acampo, mdorigo}@ulb.ac.be`

Abstract. In the multi-foraging task studied in this paper, a group of robots has to efficiently retrieve two different types of prey to a nest. Robots have to decide when they leave the nest to forage and which prey to retrieve.
The goal of this study is to identify an efficient multi-foraging behaviour, where efficiency is defined as a function of the *energy* that is spent by the robots during exploration and gained when a prey is retrieved to the nest. We design and validate a mathematical model that is used to predict the optimal behaviour. We introduce a decision algorithm and use simulations to study its performance in a wide range of experimental situations with respect to the predictions of the mathematical model.

Key words: swarm robotics, multi-foraging, mathematical modelling.

1 Introduction

Foraging is a classical metaphor used in swarm robotics [1]. In foraging, a group of robots has to pick up objects that are scattered in the environment. The foraging task can be decomposed in an exploration sub-task followed by a transport sub-task. Foraging can be applied to a wide range of useful tasks. Examples of applications are toxic waste clean-up, search and rescue, demining and collection of terrain samples. Central place foraging is a particular type of foraging task in which robots must gather objects in a central place. Borrowing the terminology from biology, the central place is also called the *nest* and the objects *prey*.

Multi-foraging is a variation of the foraging task in which different types of objects to collect are considered [2]. These different types of objects can be concurrently and independently collected by the individuals and can have different properties. Multi-foraging adds a level of complexity with respect to the traditional foraging task as it may be necessary for the individuals to choose which prey to take, and when.

The study of the efficiency of foragers has first been the concern of biologists. In his seminal article [3], Charnov exposes the fundamental hypothesis that gives birth to the field of optimal foraging. The hypothesis is that evolution has shaped individual behaviours of foraging animals so as to maximize the net energy intake. Three decades later, roboticists try to identify how robots should cooperate in order to forage efficiently. Efficiency has been defined in several ways: in biology, researchers use the term *energy* and measure weights of

F. Almeida e Costa et al. (Eds.): ECAL 2007, LNAI 4648, pp. 696–705, 2007.

animals before and after tasks to quantify energy spent and gained. In robotics, the vocabulary is less well established. Terms such as *reward, income, benefit* have been used [4,5,6,7]. For the sake of simplicity, we will use the term *energy*, as in biology. Foraging efficiently is thus a quest to maximize the energy of a group of foraging robots. Robotics researchers often consider that energy is spent when robots move during exploration and is gained when a prey is successfully retrieved to the nest [7].

We focus on a specific case of multi-foraging in which there are only two types of prey that have to be retrieved to the nest. The spatial aspect of the task is negligible as the prey have random locations in the environment. The exploration mechanism used by the robots to find prey is a random walk. Hence, robots discover prey in the environment at a given rate. Robots have to decide whether they rest at the nest and in this way spare energy, or forage in the environment. If they choose to forage, they have to decide which prey to take so as to maximize the energy of the group.

Our objective is to identify an individual behaviour that leads the group of foraging robots to have an efficient collective behaviour.

To achieve this objective, we first design and validate a mathematical model of multi-foraging. Mathematical modeling of robotic experiments is a methodology [8] [9] [10]. Mathematical models are opposed to individual based models (IBMs) [11]. In IBMs, each robot is represented as well as the environment. Differently, mathematical models are analytic description of the evolution of a system, in which the individuals of a system are not represented separately. Mathematical models are faster than IBMs because their computation time does not depend on the number of individuals. They can be used as optimization tools: Ijspeert *et al.* [12] have used a stick pulling experiment as an example to demonstrate how the behaviour of the robots could be made efficient. Within the limits of the mathematical tools available, it is also possible to draw conclusions on the dynamics and intrinsic properties of the system.

The mathematical model we devise predicts with a good confidence the optimal behaviour of the robots, and can therefore estimate the maximum amount of energy that a group of robots can accumulate during an experiment. We use the model as a yardstick to evaluate the performance of the group of robots and test different behavioural rules. Based on simplified equations, we introduce a decision algorithm to control the behaviour of the robots. To evaluate the performance of the algorithm, we run simulations using a large set of 2160 different experimental configurations, obtained by varying the parameters of the experiment.

In Section 2 we detail the task, the experimental setup and the controller of the robots. Section 3 is devoted to the description and validation of the mathematical model. Section 4 presents the decision algorithm and the evaluation of its performance using the predictions of the mathematical model. Section 5 concludes the paper with a discussion of the results and some ideas for future work.

2 Methods

The Task of Multi-foraging. The task studied in this paper is the search and retrieval of objects to the nest by a group of robots. Objects can be retrieved by a single robot. Searching and retrieving objects is energy consuming, but retrieved objects yield energy to the group. There are two types of objects that differ in energy required to be carried to the nest and energy yield. Robots have to decide when to search for objects and which objects are to be retrieved so as to maximize the energy accumulated by the group. Robots have no *a priori* knowledge of the properties of objects and should adapt online to the environment.

Experimental Setup. All the experiments in this paper are simulated. The environment is a circular arena of 1.20 meters of radius. Robots are initially randomly scattered on it. A circular nest is located in the center of the arena. Robots can locate the nest thanks to a lamp suspended above it. The nest has a structure of three concentric rings with different grey levels. The innermost ring defines where robots can rest. The second ring defines where a robot can safely release a prey, with good confidence that it is inside the nest. Finally, the outermost ring defines the boundary of the nest.

Prey are introduced in the environment at random locations around the nest, at a fixed distance from it. New prey appear and disappear with constant rates per time unit. They are removed when they fall inside the nest. Prey have a weight and friction that define the time required for being retrieved. An amount of energy is associated to a prey and is attributed to the group of robots once it is delivered in the nest. Prey of a same type share all their characteristics. We use only two different types of prey in the experiments.

The simulated robots have the same characteristics as *s-bots* from the swarm-bots project [13]. We rely on ground sensors to perceive the structure of the nest. Infrared sensors are used for collision avoidance. The camera is employed to determine the location of the nest and discriminate the type of prey thanks to their colour. Last, the robots use the camera to perceive if a nearby prey is already being retrieved by another robot.

Controller of the Robots. The controller used is the same for all the robots. The architecture of the program is a finite state machine (FSM). The scheme in Figure 2 represents the possible states, with arcs denoting the possible transitions between states. Robots are initialized in the *Explore* state.

- **Explore.** The robot performs a random walk in the environment. An obstacle avoidance subroutine is triggered when needed. The robot switch to the *Rest* state with a probability constant over time. If a prey is close enough and no green colour is perceived the robot enters the *Grasp* state.
- **Grasp.** The robot has detected a prey and may decide to perform a physical connection. This decision depends on the probabilities π_1 and π_2. If grasping is successful the robot enters the *Retrieve* state, otherwise it enters the *Ignore* state.

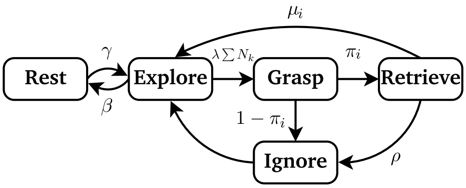


Fig. 1. Finite state machine representing the robot's controller. Transitions between states are triggered according to the probabilities described in Table 1. The control parameters (probabilities β, π_i, $i = 1, 2$) can be modified by a decision mechanism as described in Section 4.

- **Retrieve.** The robot becomes green. This colour is used to keep away other robots. The robot heads toward the nest. When the robot reaches the nest, it releases the prey and enters the *Explore* state. During retrieval, the robot has a constant probability to give up and enter the *Ignore* state.
- **Ignore.** The robot performs a random walk with collision avoidance, neglecting any encountered prey. After a delay of five seconds (enough to move away from a prey) the robot enters the *Explore* state.
- **Rest.** The robot heads back to the nest to rest there. With a constant rate per time unit, the robot can decide to leave the nest by entering in the *Explore* state.

3 Mathematical Model

Description. Partial differential equations are devised to model the flows of robots among five main states. We neglect the modeling of the *Grasp* and *Ignore* states because they occur rarely and their duration is relatively short. In order to describe how energy is gained, we model the retrieval process in two distinct parts, one for each type of prey. In addition, we noticed that the time required to go back to the nest before resting is not negligible and has to be modelled. To this extent, we introduce the *Back* state. We end up with five main states among which flow of robots are exchanged. The meaning of all the variables and parameters is explained in Table 1.

A set of differential equations is used to model the flows of robots exchanged among the states. In the following, we provide a detailed explanation of the first equation. As explained in Section 2, several transitions lead robots to enter or leave the *Explore* state. Each right-term of the differential equations is an amount of robots per second doing a specific transition.

$$\begin{aligned}
\tfrac{\mathrm{d}E}{\mathrm{d}t} &= -\beta E + \gamma I + \textstyle\sum_{i=1}^{2} \left(-\pi_i E N_i \lambda + \mu_i R_i + \rho R_i\right) \\
\tfrac{\mathrm{d}B}{\mathrm{d}t} &= +\beta E - \kappa B \\
\tfrac{\mathrm{d}I}{\mathrm{d}t} &= +\kappa B - \gamma I \\
\tfrac{\mathrm{d}R_i}{\mathrm{d}t} &= \pi_i E N_i \lambda - \mu_i R_i - \rho R_i \qquad \forall i \in [1,2] \\
\tfrac{\mathrm{d}N_i}{\mathrm{d}t} &= \varphi_i - \pi_i E N_i \lambda - \xi_i N_i + \rho R_i \qquad \forall i \in [1,2]
\end{aligned}$$

Table 1. Summary of variables and parameters of the experiment.

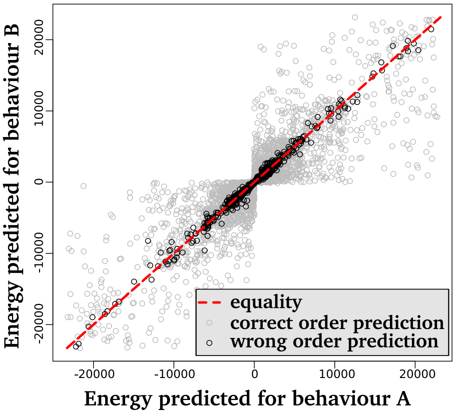
Variable	**Description**
E	the number of robots in *Explore* state
B	the number of robots in *Back* state, going to the nest to rest
I	the number of robots in *Rest* state (or inactive robots)
R_i	the number of robots in *Retrieve* state (prey of type i)
N_i	the number of prey of type i in the environment
Parameter	**Description**
T	the total number of robots in the experiment
λ	rate of objects per second found in the environment by a single robot
κ	probability for a single robot to find the nest
En_i	energy associated to a prey of type i
En_p	energy lost during one second spent outside the nest for one robot
φ_i	incoming rate per second of prey of type i
ξ_i	probability constant over time for a prey of type i to disappear
μ_i	inverse of the average time required to retrieve a prey of type i
ρ	probability to give up an ongoing retrieval
β	probability for a robot to return to nest
γ	probability for a robot to leave the nest and look for prey
π_i	probability to take a prey of type i upon encounter

- First, robots can decide to rest at nest with a probability β. On average βE robots leave the *Explore* state and enter the *Back* state.
- Conversely, robots in *Rest* state have a probability γ to come back in *Explore* state. Thus there are on average γI robots entering the *Explore* state.
- Robots may find a prey and decide to retrieve it. The probability to find a single object being λ, the average number of exploring robots that find a prey of type i is $EN_i\lambda$. As robots decide to retrieve the prey with probability π_i, the average number of robots that leave the *Explore* state to retrieve a prey of type i is $\pi_i EN_i\lambda$.
- We consider that a robot has a probability of μ_i to achieve the retrieval of a prey of type i. Hence, there are on average $\mu_i R_i$ robots that achieve a retrieval and come back in *Explore* state.
- Last, during the retrieval of a prey of type i, robots have a probability ρ to give up and come back in *Explore* state. On average their are ρR_i robots that give up retrieval of prey of type i.

Validation. To evaluate the quality of the model and determine to which extent we can rely on it to have a good prediction, it is mandatory to carry out a validation process. This phase involves the comparison of the results obtained in simulation against those of the model for a collection of typical experimental situations. We define a range of reasonable values for each parameter of the experiment (see Table 2), except for the control parameters of the robots π_1, π_2 and β. A *configuration* of the experimental setup is defined by selecting one value for each parameter from its range. There are 2160 possible configurations that define a set P. We denote $C_i \in P, i \in [1, 2160]$ one particular configuration.

Simu. / Math.	$A < B$	$A > B$
$A < B$	43.22%	7.64 %
$A > B$	6.97 %	42.13 %

(a)



(b)

Fig. 2. **(a)** Comparison table of predicted orders by the mathematical model with respect to simulation results. The notation $A <,> B$ signifies that the energy accumulated using behaviour A is lower, respectively higher than using B. **(b)** Energy predicted for behaviours B in function of energy predicted for behaviours A. The gray circles show when both the mathematical model and the simulation agree on which behaviour performs better. The black circles show cases of disagreement. They lie along the dashed line ($r^2 = 0.98$, slope $a = 1.00$), indicating that disagreements arise mainly when the predicted energy of two behaviours are very much alike.

In the following, we present a test that is meant to assess the ability of the model to compare the outcome of two different behaviours. The test consists of selecting randomly two behaviours A and B from the control space $(\pi_1, \pi_2, \beta) \in [0, 1]^3$. We compare the accumulated energy predicted for A and B. The same comparison is carried out using one single run of simulation for each behaviour.

We use each configuration $C_i \in P$ to parameterize an experiment of one hour. For each configuration C_i we generate 5 pairs of random behaviours $(A_{ij}, B_{ij}), j \in [1, 5]$. The table shown in Figure 2(a) summarizes the frequencies of all possible comparison results for the 10800 tests performed. The table indicates that in 85.35% of the tests, the mathematical model and the simulations agreed on the ranking of the behaviours. The table is almost symmetric and shows no better performance of the model if A superseeds B or the opposite.

Moreover, we have studied the conditions in which disagreement between the mathematical model and the simulations occurs. We plotted the predicted energy for behaviour B in function of the energy predicted for behaviour A. Figure 2(b) shows as black circles the pairs of behaviours that lead to disagreement. The regression performed on the black circles returns a correlation coefficient $r^2 = 0.98$ and a regression slope $a = 1.00$. The wrong predictions of the model occur mainly if the two behaviours are supposed to yield very similar energy. Given that we use only one run of simulation without averaging, an error caused by the noise in simulation is more likely to appear for these pairs of behaviours.

Table 2. Each parameter of the experiment is given a range of reasonable values. By associating to each parameter one value, we define an experimental configuration. In total, there are 2160 possible configurations.

Parameter	**Range of values tested**	**Unit**
T	$1, 2, 3, 5, 10, 15$	robot
$N_1(0)$, $N_2(0)$	5	prey of type 1 or 2
λ	1/159.4	probability
κ	1/19.51	probability
En_1	$-100, -10, -1, 1, 10, 100$	energy
En_2	1	energy
En_p	$-0.001, -0.01, -0.1$	energy
φ_1	$1/15, 1/30, 1/60, 1/120, 1/180$	prey / second
φ_2	1/60	prey / second
ξ_1, ξ_2	0.002	probability
μ_1	$1/90, 1/40, 1/30, 1/60$	second^{-1}
μ_2	1/60	second^{-1}
ρ	0.0111	probability
β	**control parameter**	probability
γ	1/400	probability
π_1, π_2	**control parameter**	probability

4 Efficient Multi-foraging

Decision Algorithm. The decision algorithm is a piece of code plugged in the controller of the robots that modifies their individual behaviour through the three control parameters π_1, π_2 and β. The algorithm relies on an equation that permits robots to individually estimate the instantaneous amount of energy EI that can be obtained by the group. In the following we briefly expose the steps that lead to this equation.

Rate of prey grasped by robots: $preyRate = E\lambda \sum_{i=1}^{2} N_i \pi_i$,
Proportion of prey of type i grasped: $prop_i = \frac{E\lambda N_i \pi_i}{E\lambda \sum_{j=1}^{2} N_j \pi_j}$,
Mean time of retrieval of a prey: $retTime = \sum_{i=1}^{2} 1/\mu_i prop_i$,
Mean time to grasp & retrieve a prey: $preyToNest = 1/preyRate + retTime$.

$$\begin{aligned} EI &= E \cdot En_p + \textstyle\sum_{i=1}^{2} En_i \cdot prop_i \cdot 1/preyToNest, \\ &= E \cdot En_p + E\lambda \textstyle\sum_{i=1}^{2} \frac{En_i N_i \pi_i}{1+E\lambda \sum_{j=1}^{2} 1/\mu_j N_j \pi_j}. \end{aligned} \tag{1}$$

Equation 1 can be used by each robot to estimate the rate of energy currently gained by the group. All the parameters of this equation, except for λ, are either control parameters, or can be estimated by the robots during the exploration of the environment.

Indeed, each robot can estimate the density of robots or prey of any type in the environment, which are respectively λE, λN_1 and λN_2. This estimate is a direct measure of the number of objects encountered per second. We use a classical discount factor method to limit the impact of past observations in the estimate of objects density. However, to compute EI robots need to know λ. This parameter may be estimated by measuring the time to go back to the nest, even though the collisions with other robots may diminish the quality of such an estimate. In the following, the robots are given the parameter λ that characterizes the size of the environment.

Based on Equation 1, the decision algorithm estimates parameters of the experiment using the observations of the robot. It then estimates the impact on the rate of energy EI of four triplet of parameters (π_1, π_2 and E): $\{$ $(1, 1, E+1)$, $(0, 1, E+1)$, $(1, 0, E+1)$, $(\epsilon, \epsilon, \epsilon)$ $\}$, where $\epsilon = 0.05$. The ϵ is a lower bound that guarantees adaptivity by forcing robots to grasp sometimes the prey or explore the environment and update their estimate of the situation. Notice also that E is a global variable that can not be directly changed by a single robot but only influenced if that robot changes its β parameter. Thus, control parameters (π_1, π_2, β) of the robot are updated to converge towards the triplet that maximizes EI.

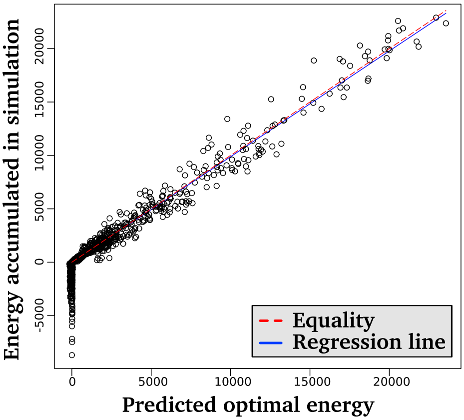


Fig. 3. The decision algorithm performs on average 99% as well as the predicted optimal behaviour (see Section 4).

Performance. We assess the performance of the decision algorithm by carrying out a systematic comparison of the energy accumulated in simulation with the energy obtained by the predicted optimal behaviour. For each configuration $C_i \in P$, we use the mathematical model to find out the predicted optimal behaviour OB_i. We use a single run of simulation to determine the energy gain $E_{pred}(C_i, OB_i)$ associated to OB_i. We also run a single simulated experiment with the decision algorithm used by each individual.[1] The control parameters are initially set to $(1, 1, 1)$ so that robots start by exploring the environment. The energy accumulated with a configuration C_i and the decision algorithm plugged in the robot's controller is denoted $E_{dec}(C_i)$.

In Figure 3, the energy $E_{dec}(C_i)$ is compared to the predicted optimal energy $E_{pred}(C_i, OB_i)$. A linear regression is applied to data such that $E_{pred}(C_i, OB_i) > 0$. The correlation coefficient ($r^2 = 0.98$) indicates that the linear relationship hypothesis holds (p-value < 0.001). The slope of the regression line is $a = 0.99$ and the bias is $b = -23.73$, which means that the decision algorithm performs on

---

[1] According to [14], this is the optimal sampling strategy.

average 99% as well as the predicted optimal behaviour. If $E_{pred}(C_i, OB_i) = 0$, it means that robots should always stay in the nest. However, the algorithm imposes a minimum exploration to get information about the environment which produces some cases with negative energy as seen in the plot.

5 Conclusions

Achievements and Contributions. We have designed a mathematical model of the experiment. Its validation has shown that the model can be used to rank successfully two different behaviours in 85% of the cases tested. In addition, we found that the errors in the remaining 15% arose only in ambiguous cases in which the energy yield by the two compared behaviours are very similar. This new tool, previously unavailable in the multi-foraging literature, makes possible the evaluation of robots performance on a relative scale.

An equation to calculate the average instantaneous reward gained by the group of robots has been devised. It has been used to implement a decision algorithm for the robots. The tests have shown that robots using the decision algorithm manage to accumulate on average 99% of the energy that can possibly be gained.

Perspectives and Future Work. In our work, we neglected on purpose collisions among robots. Lerman *et al.* [15] emphasized the impact of interferences on the efficiency of a group of robots. It is likely that robots may perceive a drop of performance and cope with the phenomenon automatically. In the future, we intend to study how well a group can adapt to a situation in which collisions may happen at a high rate and so impact strongly on the performance of the robots. The number of types of prey has been deliberately limited to two in this study. However, the mathematical model and Equation 1 can be extended to handle any number of type of prey and the decision algorithm can be adapted to comply with the extended equations. New tests will have to be carried out to measure the performance of this new implementation. Adaptivity to a dynamic environment is a recurrent property in swarm robotics. The decision algorithm has been implemented in a way that enables this aspect. Future work will report a detailed analysis of the collective behaviour of the robots facing sudden changes in the environment. It may also be possible to have robots foraging efficiently without knowledge of the λ parameter, although that would probably degrade the performance of the robots. We will work in this direction, so to make the behaviour of the robots totally free of any *a priori* knowledge of the environment. The recruitment of foraging robots could improve the performance of the robots, as reported in [7]. We did not implement this feature in the behaviour of the robots for the present study but it seems to be a promising direction to enhance the group foraging capabilities. Last, to validate our approach and assess the realism of our simulations we plan to carry out a number of experiments with real robots.

Acknowledgements. This research has been supported by COMP^{2}SYS, a Marie Curie Early Stage Research Training Site funded by the European Community's

Sixth Framework Programme under contract number MEST-CT-2004-505079. Alexandre Campo and Marco Dorigo acknowledge support from the Belgian FNRS of which they are respectively Research Fellow and Research Director.

The authors would like to thank Jean-Louis Deneubourg for help in designing the mathematical model and for fruitful discussions. The authors are also indebted to Mauro Birattari and Thomas H. Labella for relevant comments and suggestions.

References

1. Dorigo, M., Şahin, E.: Guest editorial. Special issue: Swarm robotics. Autonomous Robots 17(2–3), 111–113 (2004)
2. Balch, T.: Reward and diversity in multirobot foraging. In: Workshop on Agents Learning About and with Other Agents (IJCAI-99), Stockholm, Sweden (1999)
3. Charnov, E.L.: Optimal foraging, the marginal value theorem. Theoretical Population Biology 9(2), 129–136 (1976)
4. Ulam, P., Balch, T.: Using optimal foraging models to evaluate learned robotic foraging behavior. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 12(3-4), 213–222 (2004)
5. Labella, T.H., Dorigo, M., Deneubourg, J.-L.: Efficiency and task allocation in prey retrieval. In: Ijspeert, A.J., Murata, M., Wakamiya, N. (eds.) BioADIT 2004. LNCS, vol. 3141, pp. 274–289. Springer, Heidelberg (2004)
6. Lerman, K., Jones, C., Galstyan, A., Matarić, M.J.: Analysis of dynamic task allocation in multi-robot systems. The International Journal of Robotics Research 25(3), 225–241 (2006)
7. Liu, W., Winfield, A., Sa, J., Chen, J., Dou, L.: Strategies for energy optimisation in a swarm of foraging robots. In: Şahin, E., Spears, W.M., Winfiel, A.F.T. (eds.) Swarm Robotics. Second International Workshop, SAB 2006, Rome, Italy, September 30-October 1, 2006. LNCS, vol. 4433, Springer, Heidelberg (2006)
8. Sugawara, K., Sano, M.: Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. Physica D: Nonlinear Phenomena 100(3-4), 343–354 (1997)
9. Sugawara, K., Sano, M., Yoshihara, I., Abe, K.: Cooperative behavior of interacting robots. Artificial Life and Robotics 2, 62–67 (1998)
10. Kazadi, S., Abdul-Khaliq, A., Goodman, R.: On the convergence of puck clustering systems. Robotics and Autonomous Systems 38(2), 93–117 (2002)
11. Martinoli, A., Easton, K.: Modeling swarm robotic systems. In: Experimental Robotics VIII. Springer Tracts in Advanced Robotics, vol. 5, pp. 297–306. Springer, Heidelberg
12. Ijspeert, A.J., Martinoli, A., Billard, A., Gambardella, L.M.: Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. Autonomous Robots 11(2), 149–171 (2001)
13. Dorigo, M., Tuci, E., Groß, R., Trianni, V., Labella, T.H., Nouyan, S., Ampatzis, C., Deneubourg, J.-L., Baldassarre, G., Nolfi, S., Mondada, F., Floreano, D., Gambardella, L.M.: The swarm-bots project. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics. LNCS, vol. 3342, pp. 31–44. Springer, Heidelberg (2005)
14. Birattari, M., Dorigo, M.: How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs? Optimization Letters 1(3), 309–311 (2007)
15. Lerman, K., Galstyan, A.: Mathematical model of foraging in a group of robots: Effect of interference. Autonomous Robots 13(2), 127–141 (2002)