Learning and adaptive behavior in autonomous robots
and
Multi-robot applications

2008-03-07

Lecture 14

Literature for this lecture:

- **Wahde, M.** An introduction to adaptive algorithms and intelligent machines, p. 89-94 (distributed in the lecture)

- Additional reading: **Scherffig, L.** (2002): *Reinforcement learning in motor control.*
  http://www-lehre.inf.uos.de/~lscherff/bachelor/rlimc.pdf

- **Labella T.H., Dorigo M., Deneubourg J.-L.** (2006): *Division of Labour in a Group of Robots Inspired by Ants' Foraging Behaviour.*
  http://www.swarm-bots.org/index.php?main=2

Part I: Learning and adaptive behavior in autonomous robots

- Characteristic of autonomous robots: **self-development** and **learning** through interaction with its environment

- Algorithm(s) for a robot's "mental development":
  - **Reinforcement learning**, **Q-learning**

Learning

- **Supervised learning:**
  - Teaching through *examples*
  - *States* of the environment: *s*
  - Availible *actions*: *a*
  - Set of training examples: {*s*, *a*}-pairs
- **Unsupervised learning:**
  - Biological organisms learn by *trial-and-error*
  - Unknown situation: try *some* action, and observe the resulting state of the environment

RL motivation

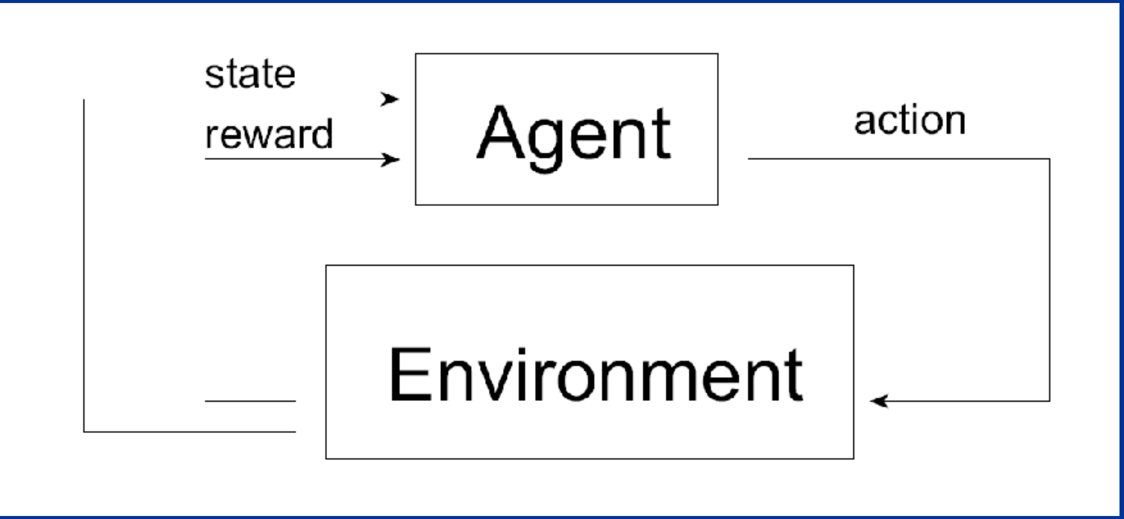
- Thorndike, 1911: **Law of effect**:
  - Behaviors in animals which lead to *reward* are strengthened
  - Behaviors that result in *punishment* or discomfort are weakened
  - The amount of strengthening or weakening is proportional to the amount of reward or punishment

Reinforcement learning

- **Reinforcement learning** is an intermediate method, between *unsupervised* and *supervised* learning:
  - The agents action *a* in a given state *s* gives rise to a reinforcement signal *r*
  - Thus, during reinforcement learning the information given by the triplet {*s, a, r*} must be availible to the **agent**

Reinforcement learning

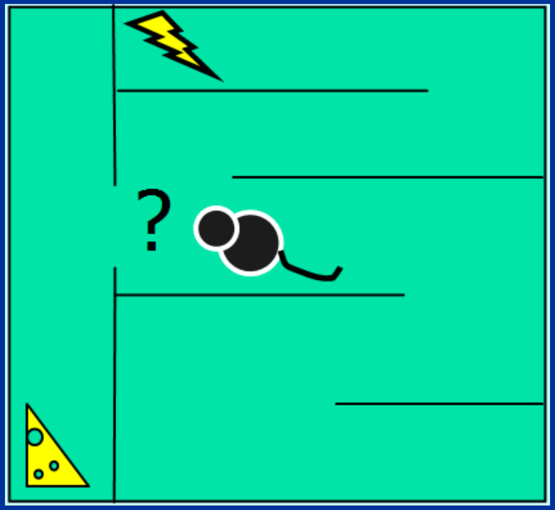
- The agent seeks to learn an *association* between **situations** (states) and **actions** to be taken given the environment in this situation:



- The agent's *goal* is to try to *maximize* the **cummulative reward**

Reinforcement learning

- Example: A rat moving around in a maze
  - If it finds *food*, it receives a *positive* reinforcement
  - If it takes a *wrong turn*, a *punishment* is received:



Q-learning

- Basic version of reinforcement learning:
  - the set of *states* $\{s_i\}$ and the set of *actions* (for each state) $\{a_i\}$ are finite.
- Consider an agent (robot) which is embedded in an environment:
  - the agent determine the current state by taking measurements of the environment
  - by taking actions, it can modify the state
  - States: $S = (s_1, s_2, \ldots, s_n)$
  - Actions: $A = (a_1, a_2, \ldots, a_m)$

Q-learning

- The agent receives a *reward r* for each action taken
- **Objective:** to find a method (policy), P, that maximizes the *total cumulative reward*:

$$R_P(s(t)) = r(t) + r(t+1) +$$

- Rewards obtained in the future is considered *less important* than *immediate rewards*:

$$R_P(s(t)) = r(t) + \delta r(t+1) + \delta^2 r(t+2) + \ldots$$

- Thus, **discount factor** $\delta < 1$ is introduced

Q-learning

- An optimal policy $P_{opt}(s)$:
  - a policy which maximizes $R_p(s(t))$ for all states s.
- A **quality function** $Q(s,a)$ is introduced:
  - **Q(s,a)**: the sum of the immediate reward when performing action $a(t)$ and the value R_{Popt} obtained by acting according to the optimal policy thereafter:

$$Q(s(t), a(t)) = r(t) + \delta R_{P_{\mathrm{opt}}}(s(t+1)).$$

Q-learning

- The task of *maximizing the cumulative reward* can now be reduced to the task of maximizing Q:

$$R_{P_{\mathrm{opt}}} = \max_{a} Q(s(t), a).$$

- However, only the immediate reward *r(t)* can be computed directly:

$$Q(s(t), a(t)) = r(t) + \delta R_{P_{\mathrm{opt}}}(s(t+1)).$$

- Computation of the *second term* would require knowledge of the optimal policy...

Q-learning

- A *recursive equation* for *Q* can now be obtained:

$$Q(s(t), a(t)) = r(t) + \delta \max_{\alpha} Q(s(t+1), \alpha).$$

- An iterative learning method for Q which uses the *present estimate* $\tilde{Q}$ of Q, is given by:

$$\tilde{Q}(s(t), a(t)) \rightarrow \tilde{Q}'(s(t), a(t)) = r + \delta \max_{\alpha} \tilde{Q}(s(t+1), \alpha).$$

Obtaining Q:

1. The elements of the matrix $\tilde{Q}(s,a)$ are set to zero.
2. The state *s(t)* is sensed, and an action *a(t)* is taken: With probability *p*, the action that maximizes Q(s(t),a(t)) is taken (**exploitation**). With probability *1-p*, a random action is taken (**exploration**).
3. When the new state has been reached, the estimate of Q is is updated according to:

$$\tilde{Q}(s(t),a(t)) \rightarrow \tilde{Q}'(s(t),a(t)) = r + \delta \max_a \tilde{Q}(s(t+1),a).$$

Convergence

- It can be shown that the iteration defined by

$$\tilde{Q}(s(t), a(t)) \rightarrow \tilde{Q}'(s(t), a(t)) = r + \delta \max_{a} \tilde{Q}(s(t+1), a).$$

causes the *estimate* to *converge to Q*.

- When the learning process has been completed, Q(s,a) generates the optimal action a to be taken in any state s (namely the action associated with the highest Q-value).

Q-learning

- Learning is a *trade-off* between **exploitation** and **exploration**:
  - If the action that is perceived as being optimal is always chosen (*greedy policy*) other actions cannot be discovered
  - If an *extreme exploration* policy is used, not much reward will be obtained...

Modified Q-learning

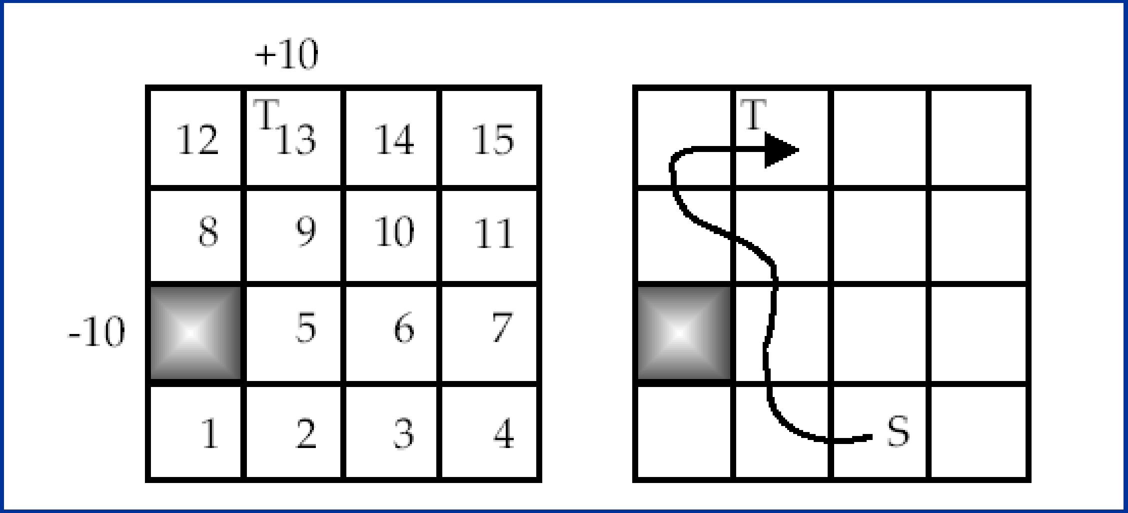
- A *modified version* of the learning algorithm is given by

$$\begin{aligned}\tilde{Q}(s(t), a(t)) &\rightarrow \tilde{Q}'(s(t), a(t)) \\ &= (1-\eta)\tilde{Q}(s(t), a(t)) + \eta(r + \delta \max_{\alpha} \tilde{Q}(s(t+1), \alpha),\end{aligned}$$

where η (0< η <1) is a learning rate parameter: the smaller the value of η, the smaller the incremental modification of $\tilde{Q}$.

Q-learning (example)

- Consider a robot moving on the discrete grid shown in the figure:



- Immediate rewards: +10 if the goal is reached, -10 if an attempt is made to enter the blocked square.

Q-learning (example)

- Initially, all $\tilde{Q}$-values are zero
- The robot move at random until the target T is reached or the robot tries to enter the blocked square.
- The robot started at state s=3 and the training episode was completed when state s=13 was reached, by moving to the right from state 12. The $\tilde{Q}$-value of the previous state will then be updated according to:

$$\tilde{Q}(12, \text{right}) \to \tilde{Q}'(12, \text{right}) = r + \delta \max_{a} \tilde{Q}(13, a) = 10 + 0 = 10.$$

- No other modifications of $\tilde{Q}$ occur during this episode

Q-learning (example)

- Consider Q(1,up):
  Immediate reward is -10
- Optimal path is then
  (in 5 steps):
  1 -> 1 -> 2 -> 5 -> 9 -> 13
- Therefore: Q(1,up)=
  $-10 + 0.9^4 10 = -3.4390$
- (In the example, $\delta = 0.9$ was used).

State	Right	Up	Left	Down
1	7.2900	-3.4390	—	—
2	6.5610	8.1000	6.5610	—
3	5.9049	7.2900	7.2900	—
4	—	6.5610	6.5610	—
5	7.2900	9.0000	-1.9000	7.2900
6	6.5610	8.1000	8.1000	6.5610
7	—	7.2900	7.2900	5.9049
8	9.0000	9.0000	—	-1.9000
9	8.1000	10.0000	8.1000	8.1000
10	7.2900	9.0000	9.0000	7.2900
11	—	8.1000	8.1000	6.5610
12	10.0000	—	—	8.1000
13	—	—	—	—
14	8.1000	—	10.0000	8.1000
15	—	—	9.0000	7.2900

Q-learning (example)

- This simple kind of reinforcement learning can be generalized to more **realistic (continuous) cases**. In such cases, the states and actions cannot normally be enumerated. Thus, instead of a matrix, Q can then be estimated using e.g. a **neural network**.
- Examples of applications: system identification, mechanics (balancing an inverted pendulum), game playing (backgammon) etc.

Part II: Multi-robot applications

- Example:
  - Division of Labour in a Group of Robots Inspired by Ants Foraging Behavior.
- Biologically inspired approach to robot control:
  - Insects can co-operate efficiently:
    - termites, bees, and ants.
  - Model based on ants' foraging behavior.

Collective insect behavior

- Insects have limited knowledge:
  - No direct communication
  - Only locally available information
  - No internal map of the environment
  - No sense of any "global plan"
- Still, insect behavior is amazingly robust in their natural environment!

Collective insect behavior

- Result of collective insect behavior goes beyond that of individual insects.
  - Key mechanism: Self organization!
- Why look at insects?
  - Inspiration for robotics researchers.
  - Multi robot systems experimental tool for biologists.

Collective robot behavior

- An object search and retrieval task
  - control algorithm inspired by a model of ants' foraging behavior.
- Division of labour:
  - robots co-operate in order to increase the efficiency of the group.
- Selection mechanism:
  - robots more suited to a task are more likely to carry out the task, than less capable robots.

Test application

- Prey retrieval task:
  - look for objects, *prey*, retrieve objects to the *nest*.
- Similar to behavior observed in real ants.
- Used as model ´for real-world applications:
  - search and rescue missions
  - demining
  - collection of terrain samples

Performance

- Since the task can be accomplished by a single robot, is there an actual performance gain in using more than one robot?
- Are more robots more efficient, than a single one?
- Efficiency = performance of the group:

$$\eta = \frac{income}{costs}$$

Efficiency

- Income:
  - prey retrieved to the nest.
- Cost:
  - interferences among robots
  - dangers in the environment
  - energy
- Income and cost depend on the number of robots in the environment.
- What is the optimal number of robots?

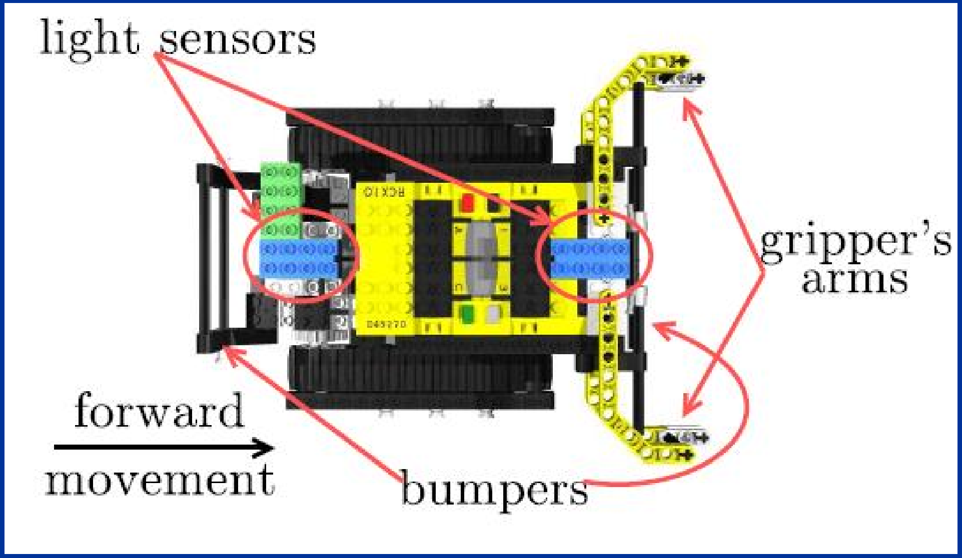
Ants' foraging behavior model

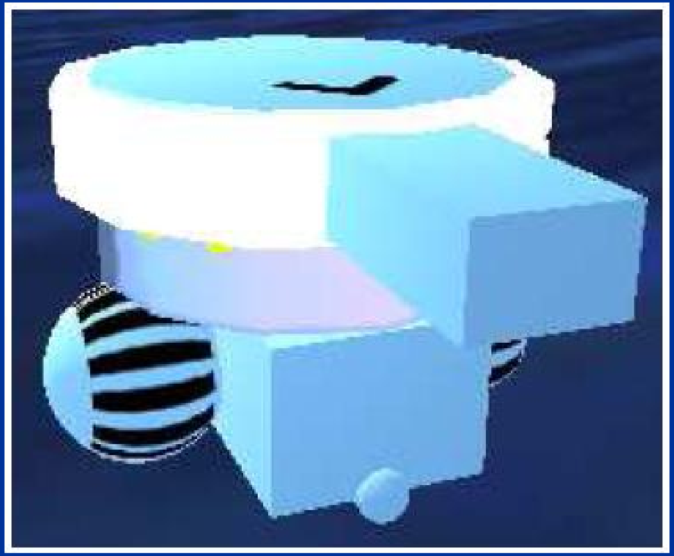
- Ants randomly explore the environment until one of them finds a prey:
  - pull it to the nest;
  - cut it;
  - recruitment;
- The prey is pulled straight to the nest
- Ant returns directly to the prey location, after retrieval.
- Learning and adaptation migth play a key role:
  - probability P_1 to leave the nest for new search
  - changes with a constant Δ, according to previous successes or failures.

Methods

- Real robots
  - validate a theoretical model
- Simulated robots
  - more data can be produce in shorter time: speeds up the analysis.
- Leads to more general conlusions!

Robots

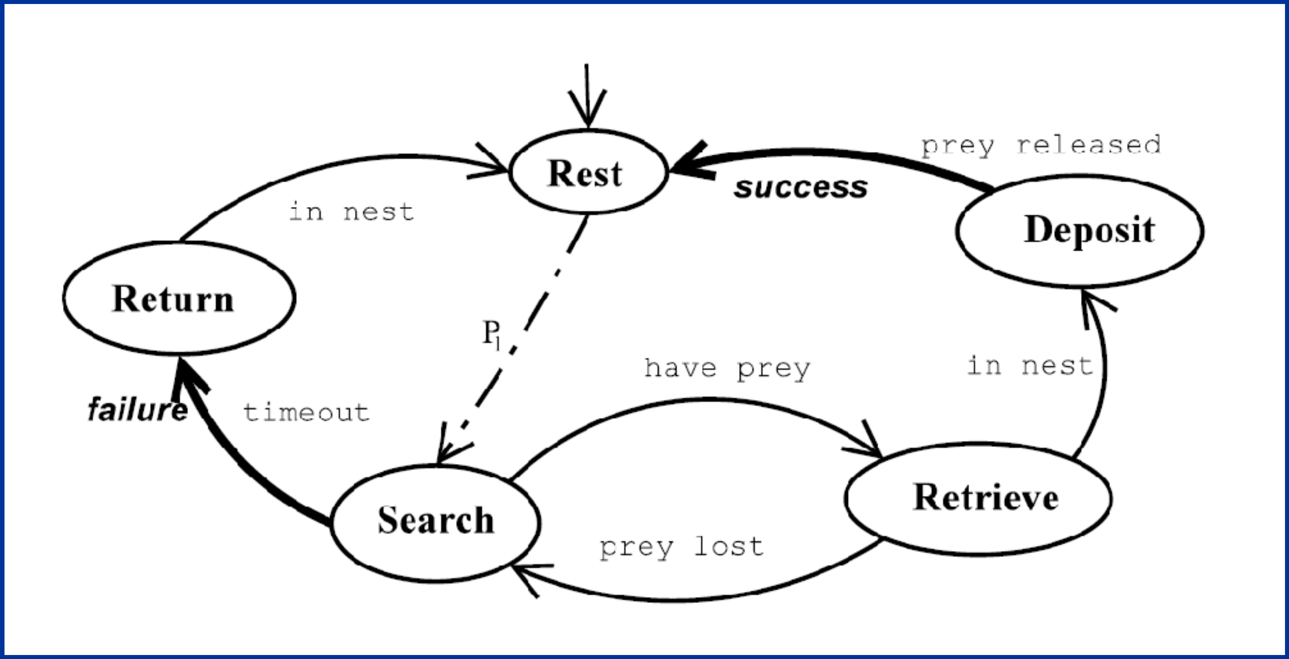




MindS-bot

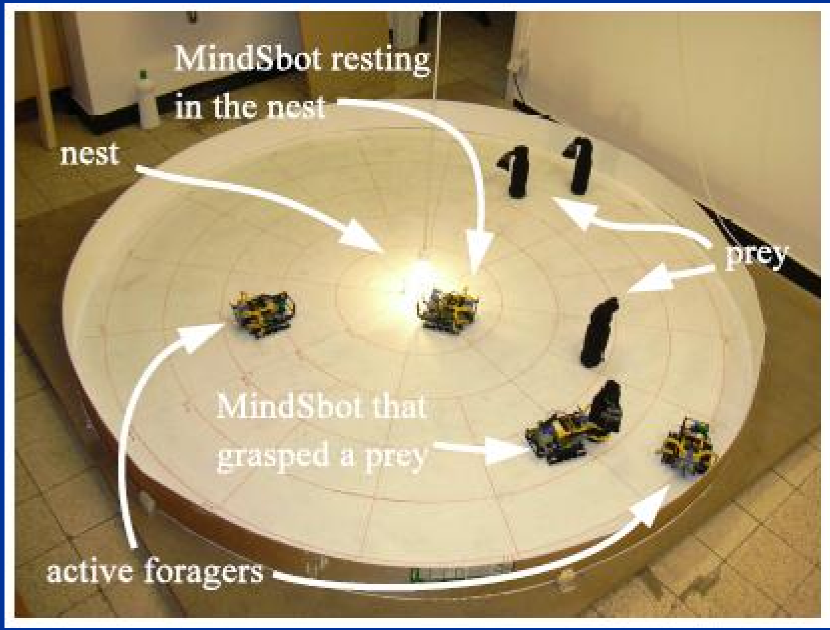
s-bot

Control: finite state machine



- Cond. state transitions:
  - When "label" is TRUE
  - With prob. P_1 once every second (+ Δ)

Experimental set-up



- Prey appear randomly in the environment
- Single experimental parameter: adaptation

Efficiency index

- Costs cannot easily be quantified.

$$\nu = \frac{performance}{\sum_{robots} duty\ time}$$

- performance = # retrieved prey
- duty time = time spent in "search" or "retrieve"

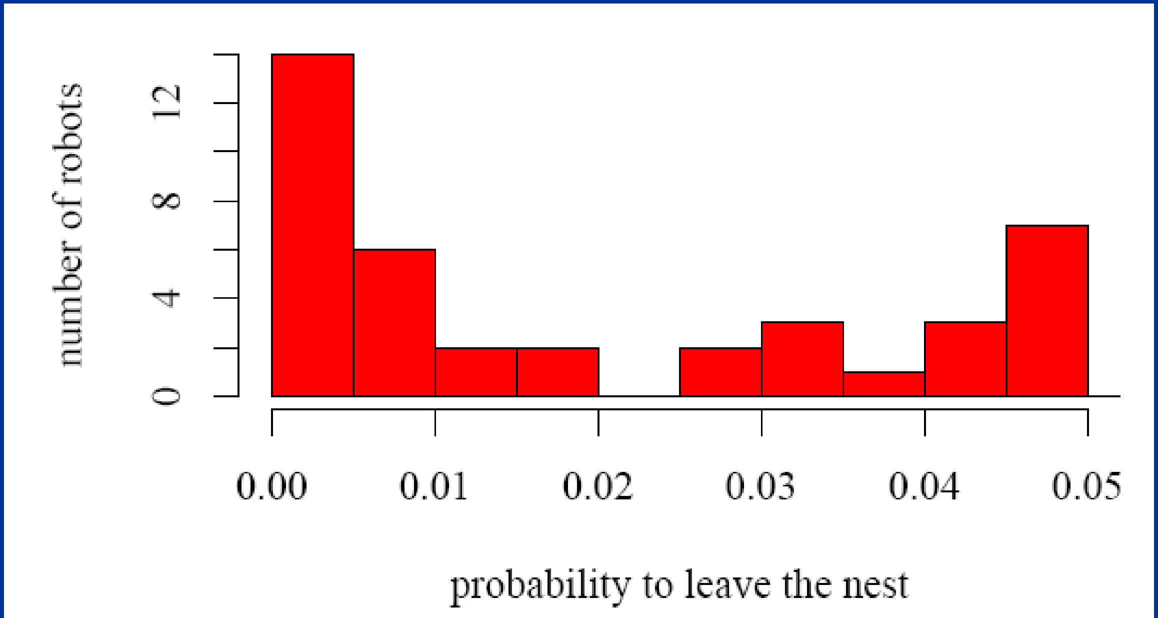
Experiments and results

- Efficiency (real and simulated robots):
  - Increased significantly when using adaptation.
  - No difference in *performance* obtained

    => improvement is due to decrease of group *duty time*.

Experiments and results

- Division of labour occured:



  – Two peaks in P_1 indicate two distinct groups of robots: *active foragers* have high P_1, and others have low P_1 value.

Conclusion

- Individual adaptation, which uses only locally availible information, can improve the efficiency of a group of robots by means of division of labour.

About the exam

- Friday, 20080314, 08.30-12.30, V-building
- Allowed to bring a calculator, provided that it cannot store any text: Can be bought at Cremona (Chalmers' bookstore).
- It is allowed to bring mathematical tables (such as e.g. Beta), as long as no text has been added.
- It is ***NOT*** allowed to bring any course material e.g. lecture notes, or to use other tools such as computers, cell phones etc.
- Make sure to bring a VALID ID!!

About the exam

- The maximum score on the exam will be 25 points.
- The exam will contain both mathematical problems and questions concerning the various topics covered in the lectures. You *may* be asked to derive (and use!) equations etc.
- **No** programming-related questions in the exam, i.e. you will **not** be asked to write program code.
- The problems can be based on **all** the material rated as *important* in the *Reading guidance* files.

Next quarter...

- The robot construction part starts (finally :-)) on **April 1st** in ET-lab (Fundamental physics building)