paths in different environments, in which we realise different constraints. The robot answer to constraint experimentation plan is often unforeseeable and the results are useful for planning the environment visiting.

Keywords:
emergent properties, robot behaviour, autonomous system, exploration paths.

1. Introduction

The field of Artificial Life investigates the fundamental properties of living systems and attempts to capture these in artificial media, like computers. One of the most important concepts that have been identified is *emergent behaviour*. This concept, and the study of it, is important because it provides an elegant way to create complex systems that are: (I) robust, (ii) Distributed, (iii) Extendable. These qualities are all desirable in most types of computer systems. Emergent properties, in the context of computer programs, are often defined as *being properties that are a consequence of the interactions of the behaviours programmed into the simpler parts of the system*. If these properties sum up to "more" than the sum of their parts, they are said to be *synergetic*. The terminology is useful for identifying fuzzy properties of complex systems that might be hard to describe formally. This paper will address the issue of emergent behaviour in simulation of mobile robotic trajectories. A formal definition of emergent properties was proposed by

i_1 ,

of applying interactions

primitives, ***S1*** ,
and the observable properties of the primitives ***Obs1*** (S^1)

(S^1 ***, Obs^1 (S^1) , Int^1).*** This means, that a Property P is emergent ***iff*** it is observable on S^2

property is said to be an emergent behaviour of the group. Literature describes various emergent properties in dynamic systems: they derive from interaction among parts of the system and among these parts and the environment and they are not explicitly implemented. The meaning of ‘explicitly implemented’ depends on type of the actual system. For instance in the case of gazelle herd this means that individuals do not pursuit a group target, but it interact and control only a limited part of the community and of the space. In the case of an artificial system, like a computer simulation or a robot team, this means that the named behaviour is not pre-programmed. Normally the property is named as ‘emergent’ when a macro-behaviour appears while the individuals have only local knowledge. Emergent properties may be found in many fields. This work concerns environments in which different robots operate with proper target. These targets may be: - Creating space structures of static or dynamic type; - *Foraging*, that is dislocating resources from one or plus zones to other one; - mapping, that is creating a model of unknown environment. Properties that may emerge in these systems may be: - stable space structures; - collective decisions, for instance concerning the *foraging* source choice; - *task* distribution among robots; - organisation in social layer. From the robotics point of view the interest for emergent properties depends on the possibility of obtaining complex behaviour by implementing, in a variable number of robots, very simple algorithms.

We can adopt more general approach to definition of emergent properties, and define them in terms of systems that contain a number of agents. These agents are able to modify the world where like the grouping near the dominant animal, or in general some segregation forms, are well described. His model consists [4] in a grid of world which is inhabited by two type of individuals: each type prefers to be surrounded by a minimum of identical individuals. This causes the migration of individuals so that the result is a series of homogeneous groups. The swarming patterns in ants emerge in ants, that constitute a big society: populations of warrior ants over 20 millions of people. These insects are in practice blind and for their work use the pheromone for signing their paths and for following the paths of the others. The warriors perform raid in-group of 200.000 people. Different species use different patterns of swarming: in the past the ethnologists attributed the difference in swarming to different inherited behaviours. But Denebourg shown, by using simulated ants, that different swarming patterns may be derived from the same system of pheromone delivering and path following: that is the ants execute ever the same algorithm and the variations concern only the food distribution in the environment. The collective ant behaviour depends on pheromone intensity: for instance the shortest path to the food depends on the fact that the pheromone intensity remains higher. The different patterns derive from interaction between ant-flow out (for food search) and the ant-flow in (with food). This realises a positive feedback: The higher level attracts more ants and so the pheromone intensity grows, and so on. Deneubourg and Goss, by extending the model, found another emergent property: the capacity of avoiding dangers. In fact, if along a path an ant is dead or come back with delay, the pheromone levels decreases and the other ant's choice alternative paths. Furthermore the task distribution is present in various social insects.

Emergent properties are evident also in artificial systems, for instance in *cellular automata*. These are dynamic systems, simulated structure, like walls, corridors, gates, obstacles, etc., and by other objects, that are interesting for the agents, for instance resources to displace or energy sources, etc. In this approach, named agent based modelling, the main element is the agent, in which we can detect the following properties: a) Interior representation of data (memory or *states*). b) Methods for modifying the interior representation of data (*perception*). c) Methods for interacting with the environment (*behaviours*). In many structures using ABM perception and behaviours are described by a standard programming language. This on one hand offers great flexibility and on the other hand the interactions, perceptions, behaviours may be very complex, For this reason agent-oriented language are studied. The advantage and drawbacks of ABM can be compared with the traditional dynamic system simulations, like differential equations, statistical approaches, etc. These have the following drawbacks: - wheel describe the properties of a known system do not explain their origin; - do not treat situations in which the hypotheses are invalid; - have problems with discontinuity; - have problems with heterogeneous populations, in which the individuals may learn. In practice the ABM is a system to integrate the conventional methods. ABM normally studies the agent group dynamic and in particular the variation of this in different environment. The agent may interact with a) direct space interaction; b) indirect space interaction (resource possession, resource exhaustion, pheromone dispersion, etc.); c) communication; d) transactions. ABM is used in many disciplines, like Robotics, Economy, Ecology, and Biology. In our work we will use the collective behaviours and in particular the co-operative behaviours: that is behaviours that attempt to reach a target by acting in-group. When, given a task, in a robot team the co-operation mechanism supplies a growth of the total system profit.

g. The robot rests on two wheels and on two Teflon balls; the wheels are linked to two DC motors with incremental encoder, that for each impulse gives 1/10 mm. of forwarding. The sensors are 8, six on the front end and 2 on the back; they supply a distance value from the obstacle of 10 bits. The detected distance depends on light and on the surface type and obstacle colour: an object is detected from 1.5 to 4 cm.

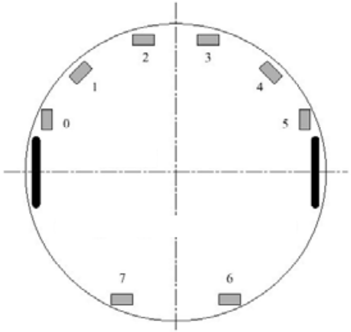


Figure 1.The robot sensors

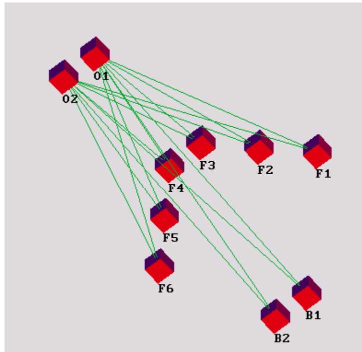


Figure 2. The control ANN

The controller of the simulated robot is realised in evolutionary form. An artificial neural network (ANN) constitutes the control system with the topology of figure 1 and in which it is possible to vary the interneural weights. The input to the controller is constituted by the values registered from the sensors, the output by the amplitude of the motor command (see figure 2). GNU/Linux is the used operating system C++ is the programming language. Figure 3 shows the graphic interface of system. The YAKS software is released under GPL license, then with source code that may be freely modified. This software was developed departing from consultation. In fact we call "record", in these tables, the values induced on sensors by all objects of the simulated world. The object sampling is realised in 3D, but the simulation in practice evolves in a plane and the robot has only two freedom degree for translation. Besides the gravity is not taken into account. The YAKS system allows simulating the real sensors of Khepera, as well as ad hoc sensors. It may simulate the following sensors: frontal IR of proximity (2,4 or 6); back IR sensors (2); array of light sensors (1,4,6, or 8), gripper, ground (for robot parking), compass (rotation angle); energy (parking on energy zone); rod (for recognition from another robot); rod sensor: gives 1 if a rod is detected in its vision field of 36 degrees. The ***environment*** objects are: *wall*, without thickness and described by co-ordinates of initial and final points; *zone*, that are circular, described by the centre co-ordinates and by ray R; *lights*, described by the centre co-ordinates and by the ray R; roundobs, or cylindrical obstacles of circular shape: described by the centre co-ordinates and by the ray R; sroundobs, cylindrical mobile obstacles described by the centre co-ordinates and by the ray R. An environment description my be:

```
# Box
wall 0.000000 0.000000 400.000000 0.000000
wall 400.000000 0.000000 400.000000 400.000000
wall 0.000000 400.000000 400.000000 400.000000
wall 0.000000 0.000000 0.000000 400.000000
# Diagonal wall
wall 300.0 100.0 100.0 300.0
# Walls 1
wall 100.0 0.0 0.0 100.0
wall 400.0 300.0 300.0 400.0
# Obstacles
sradius 12.0
sroundobst 300.0 100.0
sroundobst 100.0 300.0
```

TEST_SAME_START_POSITIO
N **0**

The file includes: number of robots involved (1); robot sensor (front 6 back); generations (1000); start generation, individuals of population (100); epochs (2), that is the number of time the robot acts; the timesteps (400); the parents (20), that is individuals generating new population; offspring's (5), that is number of sons of each parent; selection method (0); bit mutation (1) that is mutation percent; number of fitness function(20), number of individual to log(10), whose genetic code is to be saved; generations after which to save the genetic code (1);

4. Experiments

Emergence of space patterns related to forward movement.
In a first experiment we explore the environment in a systematic and iterative manner. The environment is shown in figure 4. The greater circle represents the robot, and the diameter represents the axis of the whiles. It is the presence of shortest wall that induces the circular path. The sensor number is 6 on front and two on back. The simulation file is the previous one. The adopted fitness function is the n.21 and, at the k simulation instant, is represented by:

$$\Phi = \left[\frac{|O_s| + |O_d|}{2}\right]\left[\frac{2 - |O_s - O_d|}{2}\right]\left[1 - S_{max}\right]$$

Where: O_s is the activation function of output neuron of the left motor. O_d is the activation function of output neuron of right motor. S_{max} is maximum value among the input proximity sensors. Each factor varies between 0 and 1. The first factor assumes high values experiment, to the low mutation rate (1-%): 128 bits (16 neural connections x 8 bits for coding the weights) constitute the genetic code; the inversion of few bits scarcely affects the movement direction. The fitness function is represented in figure 6

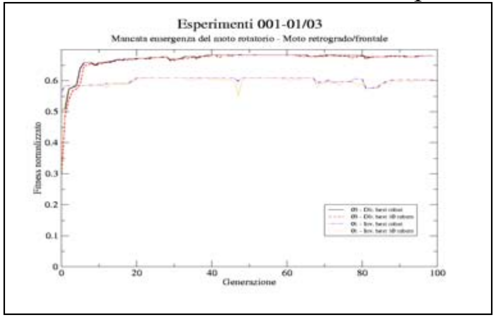


Figure 6 Fitness function of backward movement.

Note that the evolution toward good performance of robot is quite fast: after 10 generations the fitness is good and after 20 it is reached almost the maximum asymptotic value. It is evident in the figure that the fitness and the mean of best ten are monotone and growing. Instead the mean value of the population oscillates: this depends on worsening of performance due to casual mutation. The fitness is independent on movement type.

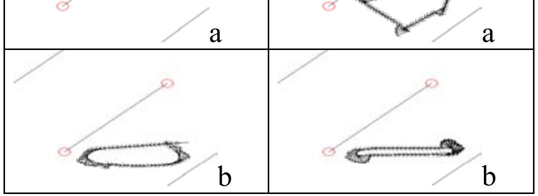


Figure 8. Patterns with forward (a) and backward (b) movement.

there. Instead they emerge interesting space patterns: in the forward movement the path adapts itself to trapezoidal shape of half-corridor; in the backward movement the patterns are smoother. (See figure 8)

Emergence of circular pattern. Noise influence

The values that the objects induce on robot sensors are derived from lookup tables and are deterministic.

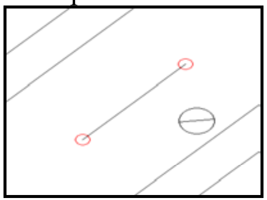


Figure 9. Environment in presence of noise

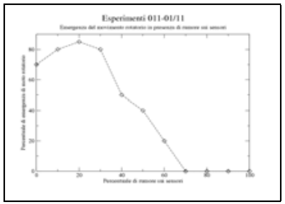


Figure 10.Emergence of circular movement in presence of noise

The simulation of a not full reproducibile experiment, requires the addition of noise to the input. In fact in this experiment we simulate the effect of noise on the emergence of circular movement. Figure 9 shows the environment. The noise percentage will be varied during

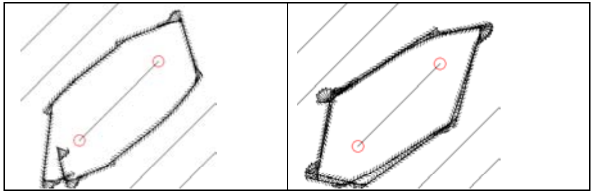


Figure 12. Pattern with 2 output sensors

evaluating the circular movement emergence and the different exploration patterns.. The environment is the same as in the previous experiment. The same is the fitness function. The emergence of circular pattern is present in both cases. The patterns in presense of 4 frontal sensors are similar to one with 6 frontal sensors. Less regular are the patterns in presense of 2 frontal sensors and in some cases attempt to assume an ellipsoidal aspect. By reducing the sensor number we reduce the resolution and robot detects the obstacle with higher path variability. Figures 11, 12 show the results.

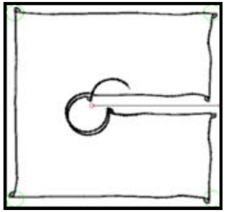
Exploration of an open space.

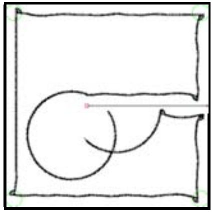
In a first experiment the robot explores a free environment: in the environment, with 1000 mm of diameter, there are five circular zone to explore, along a side. They are : 6 frontal, 2 back and 1 revealing transit on a zone,. sensors. The ANN neurons are: 9 on input and 2 on output and the fitness function is the sum of explored zones during the 4 epochs. They emerge two main behaviours: 1)in the first one robot go ahead in frontal mode and, when is near the wall, rotates of an angle (clockwise or not) independently on the fact that he passed trough a zone or not. *The amplitude of the angle is the main emergent property*: this allows robot to through the greater number of zones. (see figure 13a) 2)The second one is similar to the

Figure 14a.Open space. II° exp. I° patt.

Figure 14b. Open space. II°exp.II° patt.

Figure 14c. Open space. II° exp. III° patt.





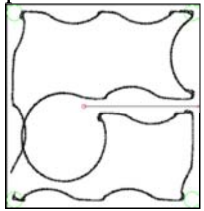


Figure 15a. 2-room env.. III° exp. I° patt.

Figure 15b. 2-room env.. III° exp. II° patt.

Figure 15c. 2-room env.. III° exp. III° patt.

movement disappears, and emerges three strategies: 1)Oscillating wall following; as in figure 14a. 2)Wall following; as in figure 14b. 3)Smooth wall following, as in figure 14c. In a Third experiment we make the environment more complex, transforming it in a 2-room environment with a central wall and 4 little zones in the 4 corners; at the end of the central wall a little obstacle to avoid invisibility of the wall. The adopted strategies rom the robot are shown in figures 15a, 15b, 15c: the first two patterns are similar: in presence of cylindrical obstacles the robot performs a border following, and when the sensors do not record activity, the robot turns in search of a new wall. The third pattern concerns a backward movement where the robot follows the wall with an oscillating circular movement, with a ray of curvature apparently equal to half of environment width. In a patt.

2)Wall following parallel to the wall, then little circular movement to reach the zones. Along a central wall, the circular movement allows passing through the central zones. After, the robot side, meeting the next wall, differs from the one near the wall, then the robot turns 90° and again turns with little curvature.Also the next wall will be near the side that produce the rotation of 90°. Finally the third wall is reached with the old side, the wall following and then the cycle begin again. This path is the more *complex* experimented and is shown in figure 16c.

5. Conclusion

The interaction of a robot with a complex environment gives complex robot behaviour. Then forecasting its behaviour may be very difficult and brings us to simulate it. The chosen approach, that uses a control system of the robot realised with a neural network and the genetic evolution of the behaviour allow to overcome the difficulty of detailed and often impossible projects. In fact the project complexity scales more rapidly then the number of involved parts: the complexity is related with number of interactions among parts. [8]. Another problem is the scarce a priory knowledge of interaction of the robot with the environment. An action influences the next stimuli and may have a long-term influence. This suggests using a procedure that allow to gradually varying the control to have useful behaviour. In our experiments we directly projected only the fitness function. The first series of experiments do not show new behaviour. Instead second series of experiments emerged complex paths, very difficult to preview in a project: to the gradual growing of complexity the robot answered with ever new techniques,

emergence of increasingly complex advantageous behaviours". In the special issue "Emergent Properties of Complex Systems, *International Journal of System Science* 31, pp.843-860, 2000.

[8] Cliff D.T., Harvey I., Husbands P., "Explorations in Evolutionary Robotics", *Adaptive Behaviour 2*, pp. 73-110, 1993.

[9]Dale K., *Evolving Neural Network Controllers for Task Defined Robots*. Technical report csrp355, The School of Cognitive and Computing Sciences, University of Sussex, UK.

[11] Deneubourg J.L., Goss S., "Collective patterns and decision making". *Etholgy, Ecology and Evolution,* 1, pp. 293-311, 1989.

[12] Floreano D., Mondada F., "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot", in *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*. MIT press/Bradford Books, 1994.

[13] Floreano D., Mondada F., "Evolution of Homing Navigation in a Real Mobile Robot"*IEEE Transactions on systems, man, and cybernetics*. 1994.

[14] Hamilton W.D., "Geometry for the selfish herd". *Journal of Theoretic Biology*, 31, pp. 295-31, 1971.

[15] Hemerlijjk C.K.,"Social phenomena emerging by self-organisation in a competitive, virtual world ('DomWorld'). In *Learning to behave. Workshop II: Internalising knowledge, pp.11-19*. Ieper, Belgium 2000.

[16] Hemerlijjk C.K.,"Computer Models of Social Emergent Behavior". *International Encyclopaedia of the Social & Behavioural Sciences*. Elsevier Science Ttd, 2001.

[17] Hogeweg P., Hesper B.,"The ontogeny of interaction structure in bumblebee colonies: a MIRROR model". *Behavioural Ecology and Socio-biology*, 12, pp. 271-283, 1983.

[18] Holland J.H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press 1975.

[19] Kaelbling L.P., Littman M.L., Moore A.W. 1996. "Reinforcement Learning: A Survey". *JAIR, Journal of AI Research*, vol. 4, 1996.

[20] Koza J.R., *Genetic Programming: on the programming of computers by means of natural selectio*n. MIT Press, Cambridge, Massachussets, 1992.

[21] Krink T. Vollrath F., "Emergent properties in the behaviour of a virtual spider robot". *Proc. Royal Society* London, 265, pp. 2051-2055, 1998.

.[22] Lund H.H., Hallam J., *Sufficient Neurocontrollers can be Surprisingly Simple*. Research Paper 824, Department of Artificial Intelligence, University of Edinburgh, 1996.