ARTICLE IN PRESS

Biologically Inspired Cognitive Architectures (2015) **xxx**, xxx–xxx

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Leader election algorithms for static swarms

Valery Karpov *, **Irina Karpova**

National Research University Higher School of Economics, Moscow, Russia

Received 24 January 2015; received in revised form 2 April 2015; accepted 2 April 2015

KEYWORDS
Swarm robotics;
Voting procedure;
Group leader selection;
Roles distribution;
Local interaction;
Pack-hunting

Abstract
Work solutions are proposed for problems of leader definition and role distribution in homogeneous groups of robots. It was shown that transition from a swarm to a collective of robots with hierarchical organisation is possible using exclusively local interaction. The local re-voting algorithm is central to the procedure for choice of leader while distribution of roles can be achieved by a wave method. The basis for this approach is the static swarm model characterised by the absence of a set control centre; it represents the network fixed at some time interval as a set of locally interacting agents. A task of cooperative hunting by distributed mobile robots based on local interaction was considered. Two strategies were used for the hunting task solution: individual hunting and pack-hunting. Simulation results showed that symbiosis of leader election and role distribution procedures has advantages over the individual strategy.

Introduction

Active research into the creation of systems of interacting robots has been ongoing for nearly a quarter of a century. Approaches such as collective, swarm and flocking robotics have been prominent in modern robotics and the theory of multi-agent systems, but the overwhelming tendency of research in this area remains at a theoretical, model level. According to Yogeswaran and Ponnambalam (2010) and Shi, Tu, Zhang, Liu, and Wei (2012), it is apparent that this absence of valid, significant results is not least connected with the relative neglect of a number of important tasks. Research in the field of group robotics (to coin a generalised name for collective, swarm, flock etc. robotics) has a very fragmentary character.

One warning sign is the obvious transfer of the enter of gravity of research to swarm robotics, which can be viewed as a simpler, more basic level of group robotics model.

The reason seems in the following. A swarm as a homogeneous set of agents is capable of solving tasks of a very limited class – so-called simple tasks. We will try to explain this thesis. If we cannot explain the behaviour of the entire

* Corresponding author.

http://dx.doi.org/10.1016/j.bica.2015.04.001
2212-683X/

Please cite this article in press as: Karpov, V, & Karpova, I (2008), Leader election algorithms for static swarms, *Biologically Inspired Cognitive Architectures* (2015), http://dx.doi.org/10.1016/j.bica.2015.04.001

system, knowing its properties and the principles of functioning of its components (emergent properties), such a system is called difficult. Otherwise, we deal with a simple system. The swarm is not a complex system since the principles and algorithms of functioning of its entities are identical. In this sense, we speak about simplicity of the tasks solved by a swarm.

The solution of complex challenges means existence of functional heterogeneity. Differentiation of functions, realisation of levels of abstraction at information processing, planning, etc. are the tasks demanding existence of some structure. Let us consider a system consisting of identical elements (agents) with various roles. From this viewpoint, the mechanism of defining a leader can be considered a possible method to initiate formation of structures, as a certain analogue of formation of the centre of crystallisation. So, we consider cognitive abilities growth of a swarm using a mechanism of functional heterogeneity. Further, we consider the mechanisms at the basis of this functional heterogeneity.

We elaborate two among the many problems in swarm robotics that remain insufficiently studied. The first is the problem of leader definition in a homogeneous group of robots and the second is role distribution among members of the group under conditions of exclusively local interaction.

Leadership

One of the basic features of swarm robotics is the local character of interaction of robots, with each other and with their environment (Shi et al., 2012). This kind of interaction is called implicit communication according to Yogeswaran and Ponnambalam (2010), which means that each robot in the group directly interacts only with neighbours within some limited visibility range.

In such systems, it usually follows that robots make decisions independently on further actions, guided by some simple rules of local interaction. However, the overwhelming majority of examples of task solution in the field of swarm robotics concern the coordinated movement of a swarm. For instance, in the obvious and rather simple task characterised as the 'Leader—Follower' method in Song and Zhao (2014), it is considered that there is an a priori leader in the group who sets this movement.

There are many variants of local interaction rules, from the formalistic in Pavlovsky, Kirikova, and Pavlovsky (2010) to the very exotic. For instance, Dewi, Risma, and Oktarina (2012) describe the virtual spring-damper model of flocking mobile robots' behaviour. The 'spring' component of the model defines the force of attraction to the leader (not follower to follower), and the 'damper' component defines the repelling force.

Another especially technical approach has been proposed in Gigliotta, Mirolli, and Nolfi (2014), in which one of the robots must become the 'leader' of the group by maximising the value of its communicative output. The main issue of this method is that it leads to the appearance of a set of leaders, and a number of them depend on the topology of the swarm. In a very interesting project described in Nissan EPORO (2009), autonomous robots used the rather simple 'Fish Behaviour' interaction rules for collision-free driving, in which all robots were equipped with a set of complex sensors. The general direction of robot movement is set from the outside. Therefore, we deal mainly with the a priori set leader, or with techniques that avoid defining the leader under the conditions of some specific objective.

However, in a number of works, a swarm leader is elected. In Kim, Shin, Woo, Eom, and Lee (2008), group leader selection is based on optimising power consumption, making it necessary to know the distances between robots and the power consumed by transfer of the message from one robot to another. In Yu, Jian, and Wang (2008), swarm traffic control using the centre of masses or the geometrical centre of a swarm is described. This method requires that coordinates of agents, their speed and their direction of movement must be known. Another paper, for example, Loukas, Woehrle, Glatz, and Langendoen (2012) describes a localised mechanism for determining the information potential on each node, based on local process information and the potential of neighbouring nodes. In that instance, the node with the minimum potential was considered to be the leader. A similar technical approach was offered in Karpov (2012), in which the agent with the greatest weight was appointed as leader. The difference between the leader and other members of the group (or a flock) was that the leader did not use the rule 'move to the nearest neighbour'. This approach provided a solution to the problem of coordinated movement, turning a swarm into a flock.

In this paper, we are interested in the problem of leader identification in a more general case, in terms of mechanisms of local information interaction.

Role distribution

As a solution to the basic problem of coordinated movement of robots, it is sufficient that there is a leader. However, more complex challenges solved by a group of robots require differentiation of their functions and, generally speaking, distribution of tasks between robots; this is perhaps one of the most problematic concepts of swarm robotics. In reviews mentioned above Yogeswaran and Ponnambalam (2010) and Shi et al. (2012), task distribution in robot groups was considered rather declaratively. At best, some physical models, methods of distributed planning, optimisation and other general mechanisms are mentioned in Kalyaev, Kapustjan, and Ivanov (2011). In practice, one usually deals with either the centralised control systems or with homogeneous groups without functional differentiation. For example, Kukushkin, Katalinic, Cesarec, Zdyb, and Kettler (2012) describe an assembly system model with self-organising behaviour (Bionic Assembly System —

BAS). The behaviour of each mobile robot in this system depends on its internal state and on the state of the system. A central computer plans the global production of BAS, synchronising the supply of parts and so on. Another example is a system organised in a two-level linear hierarchy. Facing the robot swarm's heading direction, the leftmost robot in the front line is assigned as the global leader automatically. The rest of the robots in the front line work as the group leader, and the ones after them are the followers (Wu, Qu, Xu, & Chen, 2014).

So, it is possible to draw the conclusion that differentiation of functions and distribution of tasks is not considered an actual problem by swarm robotics. Instead, it is usually taken that the swarm has to solve only simple, mass problems like coordinated movement. This certainly reduces the importance of the swarm approach and goes straight to the main declared thesis of swarm robotics as an approach to the solution of complex behavioural tasks using a set of simple technical devices — robots.

In the present case, it is considered that the problems of leader formation and role distribution are extremely important for the development of swarm robotics. This paper considers some ways in which these tasks might be solved, such as modelling the organisation of a group of robots as a static swarm.

The remainder of the paper is organised as follows. Sections 'Task definition', 'Static swarm' and 'Voting task' are devoted to the static swarm definition. Section 'Voting task' describes a set of voting algorithms that solve model a task of leader election. The experimental results are presented in Section 'Experiments'. Sections 'Task distribution' and 'Pack-hunting task' describe a variant of the role distribution task solution and demonstrate a pack-hunting task solution based on procedures of leader election and roles distribution. Finally, Section 'Conclusions' summarises the conclusions and suggestions for future work.

As a starting point, we consider the question of agent's structure and then turn to the main object of this work, namely static swarm.

Task definition

The task is formulated as follows. Consider a set of simple devices (robots or agents) capable of directing local interaction between neighbours. The question is whether it is possible to formulate conditions under which it will be possible for such a system to solve more complex problems, both at the behavioural level and at the levels of information processing, decision-making, and so on. In other words, it is necessary to define the conditions of emergence of synergetic effects or emergent properties.

There are two main classes of swarm models: micro and macro. The first of is based on the behaviour of individuals, while the second class is based on the description of the swarm as a whole. For example, at the micro-level, models of finite-state machines are widely used, and at the macro-level, hydrodynamics models are common. Hybrid models combining both approaches are less often applied. For example, Berman et al. describe a model of the dynamics of environment state in Berman, Halasz, Kumar, and Pratt (2007) that defines the behaviour of members of a swarm (agents). The reasoning in the present study generally belongs to the micro-level class because our interest is in the mechanism of local interaction of robots in a swarm (as in Stefanuk, 2004).

To begin, consider the structure of an agent, where the main objective is a set of some simple devices. Simplicity here means some principal limitation of cognitive abilities (sensors, calculations and memory).

Further, some serious restrictions will be placed on the robots' communication opportunities, in which each robot can communicate with no more than some limited number of its neighbours. It will further be assumed that the robots have a fixed number of communication ports (i.e. contact points that form information channels). For example, robots must be connected physically to each other's communication ports for communication as organised in Fig. 1.

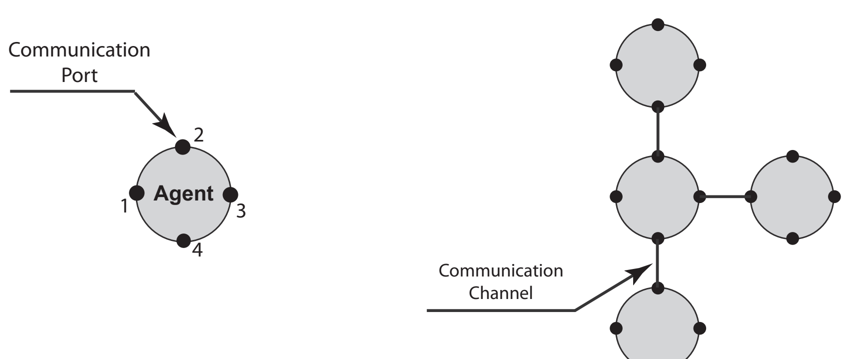

Fig. 1 Robots with four communication ports and their links.

Static swarm

One of the models describing the organisation of a great number of locally interacting agents or robots is the so-called *static swarm*. This arrangement is characterised by the absence of a control centre and represents the given network fixed at some time interval as a set of agents (Karpov, 2013).

It is a rather natural scheme. Majority of works with a decentralised focus have a fundamentally static view: the structures that describe the degree of 'local centralisation' are fixed during the assignment process (Liu & Shell, 2012).

The main feature of a static swarm is that at some moment, instead of a set of separated agents, the computing structure is completely defined, enabling the solution of difficult calculation and data processing tasks. It follows that a static swarm can be considered as an object with qualitative properties other than a simple set of agents.

The main properties of a static swarm are activity, locality of interactions and functional heterogeneity.

Activity

Certainly, unlike the computer network, the network of agents has to be capable of perceiving signals from the environment, and of producing some effector functions (such as motion, for example) in order to have an impact on the outside environment.

Locality of interaction

An important feature of a static swarm is essentially the local nature of interaction: agents communicate only with their neighbours.

Functional heterogeneity

The solution of complex tasks (i.e. manifestation of emergent properties of a system) assumes heterogeneity exists in the group in terms of differentiation of functions carried out by agents: strategic and tactical management, gathering and information processing, realisation of effector functions, and so on. The organisation of the mechanism of this functional heterogeneity is therefore an important question.

From the practical point of view, the static swarm is a particularly temporary structure. Naturally, the movement of agents changes links between elements and leads to structural changes in the swarm. Therefore, functioning of the entire system is reduced to some timepoint agents form the static structure for the joint solution of a task, for example, exchange of information, logical conclusion and so forth. The agents gather for some kind of 'conference', after which the static swarm breaks up and its elements function according to the received roles before the 'conference'.

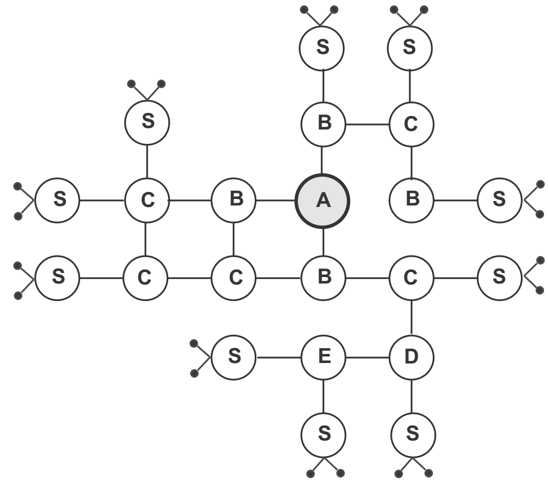

Fig. 2 An example of the network organisation.

Let us consider the following task. Let there be a set of agents (robots) capable of local information exchange between nearest neighbours. Further, at some timepoint, the static swarm must realise a certain procedure for role distribution: some individual must act as a control centre, another must serve as the data processing function, another has to collect information from the external environment, and so on.

The general principles of role distribution can be based on the following obvious reasoning: a node (agent) with a maximum number of links (neighbours) becomes a candidate for the control centre role. Its nearest neighbours are analysers of information; they prepare information for decision-making. Nodes located on the periphery of a network are responsible for gathering information. An example of such a network of agents is shown in Fig. 2.

Here, node *A* becomes a main control centre; its nearest neighbours (*B*) are assigned the role of analyser; and peripheral nodes (*S*) act as external sensors. Labels *C*, *E* and *D* designate other roles. The main question, then, is how do these node-agents choose the central, main node. There are several possible ways to organise such a vote.

Voting task

Consider the following formulation of the voting task. Let there be a set of agents with limited communication opportunities, so that agents are capable only of directing local interactions between neighbours. The task requirement is that agents must choose a leader by voting.

All further assumptions are based on the fact that the topology of a network is unknown and that all reasoning must have an especially 'local' character (i.e. considered from the point of view of the agent taking part in voting).

At some timepoint, agents receive a global signal that the vote is beginning. At this moment, each agent establishes communication channels with its neighbours, and a

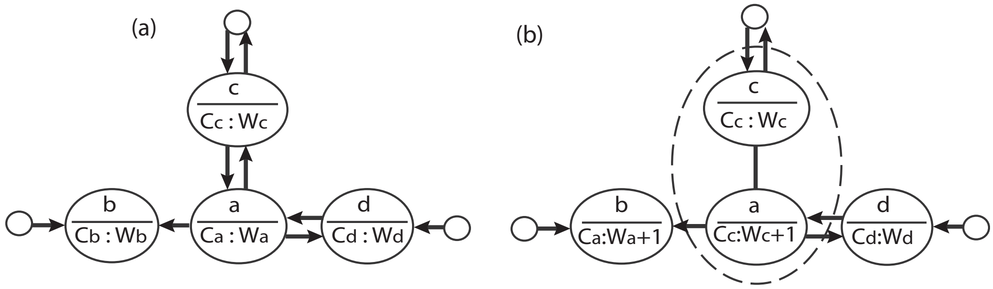

Fig. 3 One voting step: (a) initial state, and (b) votes final distribution.

generally directed graph is formed. The graph's node is an agent; incoming edges are interpreted as the ability to receive data from source nodes. In this way, communication channels are formed.

Let a static swarm be fixed (i.e. assume that its topology will not change further).

Each agent is described as

$$A = (\alpha, L, C, W)$$

where α = agent's identifier or name; L = a list of agent-neighbours from whom agent α can receive information (incoming edges); C = a candidate's identifier, for whom agent α votes; W = weight of candidate C, i.e. the number of votes which, in the agent's opinion, should be given to the candidate.

The essence of a voting procedure is that each agent defines for whom its neighbours vote. Depending, then, on the weight of the candidate for whom the neighbour votes, the agent can change its opinion and vote for the same candidate.

Fig. 3 presents one step of this voting scheme. The node labels indicate the following: the agent's identifier α is the 'numerator', and values C_a and W_a represent the candidate's identifier and weight.

Assume that agent α votes for candidate C_a and agent ***c*** votes for candidate C_c. If weight W_a is less W_c then agent α can change its opinion and revote, adding one more voice to the weight of the new candidate.

The probability that agent i will change their opinion under the influence of the opinion of agent j (an opponent) can be defined as follows:

$$p_{ij} = \frac{W_i}{W_i + W_j}$$

That is, the tendency to change opinion naturally depends on the degree of conviction or weight of the candidate.

The distribution of voices of candidates and their weight at an initial timepoint is also implemented quite naturally; each agent votes for itself (declares itself the candidate), and the weight of this decision is equal to the number of this agent's neighbours. The number of links is not the only possible variant of determination of weight of the agent. Here, we can use other reasons.

Algorithms for the agent's voting behaviour are given below.

Algorithm G1(α). Agent's decision-making—1
α — agent
C_a — the candidate for whom agent α votes
W_a — candidate's weight
L_a — list of agent's candidates
Procedure G1(α)
  To choose among neighbours of the opponent with maximum weight A_{op}:
    $A_{op} \in L_\alpha,\ C_{op} \neq C_\alpha$
    $W_{op} = \max_{i \in L_\alpha} W_i$
  To calculate value of probability of change of opinion:
    $p_\alpha = \frac{W_{op}}{W_\alpha + W_{op}}$
  To change opinion with probability p_a:
    $C_a \leftarrow C_{op}$
    $W_a \leftarrow W_{op} + 1$
end procedure G1(α)

It is possible to 'roughen' the decision-making algorithm, forcing the agent to change its opinion on the candidate if there is a stronger opponent in its environment. If parity is observed between scales of opinion of the agent and the strongest opponent, the choice of decision can be carried out probabilistically.

Algorithm G2(α). Agent's decision-making—2
Procedure G2(α)
  To choose among neighbours of the opponent with the maximum weight A_{op}:
    $A_{op} \in L_\alpha,\ C_{op} \neq C_\alpha$
    $W_{op} = \max_{i \in L_\alpha} W_i$
  if $W_{op} > W_a$ then – The opponent is 'stronger'. We change the opinion:
    $C_a \leftarrow C_{op}$
    $W_a \leftarrow W_{op} + 1$
  else
  if $W_{op} = W_a$ then – Forces are equal. We change the opinion – with probability 0.5
    $p \leftarrow$ rand() – Random value from 0 to 1
      if $p > 0.5$ then
        $C_a \leftarrow C_{op}$
        $W_a \leftarrow W_{op} + 1$
    end if
  end if
end procedure

Please cite this article in press as: Karpov, V, & Karpova, I (2008), Leader election algorithms for static swarms, *Biologically Inspired Cognitive Architectures* (2015), http://dx.doi.org/10.1016/j.bica.2015.04.001

A common voting scheme might look like this:

```
Algorithm V(A). Voting
A — a set of agents
α — agent
C_a — the candidate for whom agent α votes
W_a — candidate's weight
L_a — list of agent's candidates
Eoj — flag of end of a voting procedure
procedure V(A)
  eof ← false
  for all α ∈ A do      – Agents initialisation
    C_a ← α
    W_a ← dim(L_a)
  end for
  while not eoj do      – Main cycle of vote
    for all α ∈ A do      – Cycle on all agents
      G1(α)
    end for
    'Definition of conditions of completion of voting
procedure eoj'
  end while
end procedure V(A)
```

In this algorithm, the biggest problem is the item 'Definition of conditions of completion of voting procedure eoj'. In the absence of global information on a network's state, the agent has to make the decision for itself that voting is finished. Information received from the immediate environment is obviously not sufficient for this purpose, and two variants of agent behaviour are therefore possible:

1. To consider that voting must be completed in at most some certain number of steps, involving top assessment of the number of voting algorithm steps.
2. To realise some procedure for an exchange of messages defining that voting is completed, and no agent further changes their decision.

The first variant must prove convergence of iterative voting procedures. Some reasoning can be based by analogy with a schema for reaching a consensus. DeGroot (1974) defines consensus as mutual agreement on a subject among a group of people (agents in our terminology). The main issue is that with DeGroot's schema convergence can be proved only for some partial situations.

Another interpretation of the voting process is the well-known Polya urn model (a Polya urn scheme). The main issue in the leader election is an emergence of cycles of re-voting, when agent A changes its opinion and agrees with the opinion of agent B, and agent B, in turn, agrees with A. The role of one agent A or B can affect a group of agents, i.e. the group of agents voting for candidates A and B. However, the essence of the voting process is that the change in the agent's opinion results in the opponent increasing its weight. We can interpret a pair of agents with opposite opinions as a Polya's urn. The number of white and black balls in the urn corresponds to the weight of the agent's opinions. When the agent's changes agree with the opponent, the weight of the new opinion increases. This means that the number of balls in the urn increases too. So, the voting process is equal to the procedure of a ball's choice from the urn. A Polya scheme converges, but in practice we have one issue. The voting procedure produces cyclic processes with time delay. Unfortunately conditions for convergence of Polya scheme with time delays are not investigated.

The second variant also implies the existence of some assessed number of voting steps prompting the agent to send a request defining voting procedure completion. The realisation of procedures of this sort also presents a number of highly technical difficulties — in particular, an increase in network traffic, as each agent must realise this procedure irrespective of the others.

Justification of these algorithms requires answers to two main questions: (1) convergence of the algorithms to one solution and (2) estimation of the number of voting steps. Unfortunately, these questions currently remain open, as we can speak only about the results of modelling, according to which the process of voting converges. Clearly, the number of voting steps does not exceed the number of robots in the group.

Centralised voting

There are problems with the above result, particularly from the local nature of the agents' decision-making. If each agent knew the graph structure, definition of the leader would be quite a routine task. In fact, it is possible to provide a rather simple scheme for exchange of messages between agents, which would allow describing the full graph structure. The number of steps does not exceed the number of robots N in a group. To achieve this, it is enough for the robots to report to each other everything they know about the structure of the graph at the present time, as follows:

At an initial timepoint, information about the structure of the graph for each agent is limited to knowledge of its neighbours. This incomplete graph is represented, for example, by the list of edges L^i_0 sent by agent i to its neighbours. Having received such a list, each agent combines it with their existing list for a fuller picture in the form of a new list:

$$L^i_t = \bigcup_{k \in Z} L^k_{t-1}$$

This is a combination of the lists received from all neighbours from some area Z at the previous timepoint.

Through no more than N steps, then, each robot will have all the information about the graph's structure. Further, all of them elect the only leader, proceeding, for example, from reasons of maximum connectivity, equidistance and so on. An obvious and ineradicable deficiency of such a scheme is a very big flow of information, which should be reported to robot neighbours. The practicality of this scheme in real systems is rather doubtful.

Experiments

Two types of series of experiments were carried out. In the first type, some fixed topology structures were investigated — a line, a ring and a square. The main idea was to check the convergence of the algorithms in these special (degenerate) cases. The second type of carried a statistical orientation.

As a further example of a voting procedure, Fig. 4 shows three steps in the voting procedure for a solid group of robots.

In the first step, each agent votes for itself, so that the number of cells designating 'borders' of distribution of voices for the corresponding candidates is equal to the number of robots. The second step (re-voting) shows an integration voting areas for chosen candidates. Finally, in the sixth step, all votes are assigned to a single candidate, and the voting procedure comes to an end. An example of the process of voting in extremely adverse conditions is shown in Fig. 5, involving two clearly expressed zones connected by two isthmuses.

In this situation, a cyclic process of distribution of voices can be observed. At the 20th step of the vote, two stable areas are formed, each of which votes for their own candidate, and a process of cyclic re-voting begins. It is clearly visible how preference areas actually trade places in the 37th step. The process continues only as far as step 51, when all oscillations stop and a single candidate remains.

The second type of series carried out a statistical orientation. Planar graphs of 50—200 agents were generated. For each value of N ($N = 50, 75, 100, \ldots, 200$), one hundred experiments were carried out. Therefore, the total number of experiments was 700. We varied the average number of agent's links (neighbours), parameter L. The value of L changed from 3 to 7. Fig. 6 shows the distribution of the L value per number of robots (N). This diagram illustrates all experiments show a similar distribution of L. This implies that we should work with equally distributed topologies of robot groups.

In these experiments, a voting procedure finished after election of the single leader (normal exit) or after 500 voting steps (failure). Only 25 of 700 experiments or 3.5% ended in failure where two leaders were left after 500 steps. The simulation results are presented in Fig. 7.

These results show that failure does not depend on the number of robots in a group. On the hand, the number of voting steps increases with the group size. The dependence of the average number of voting steps on the number of robots is shown in Fig. 8.

Therefore, we can claim that the voting procedure converges in a statistically significant number of experiments, and the convergence rate weakly depends on the number of robots.

It is necessary to underline that at the end of the voting procedure we have only one leader. However, we cannot

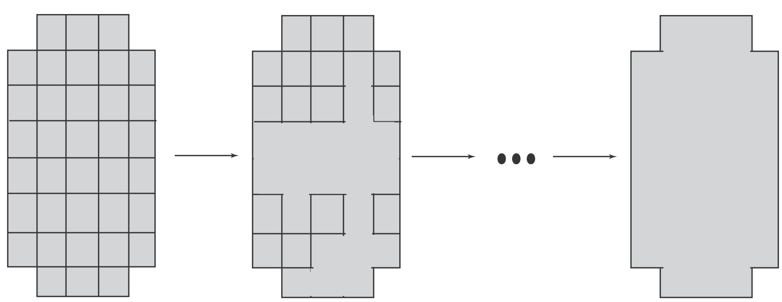

Fig. 4 Voting procedure in a solid group. Steps 1, 2 and 6.

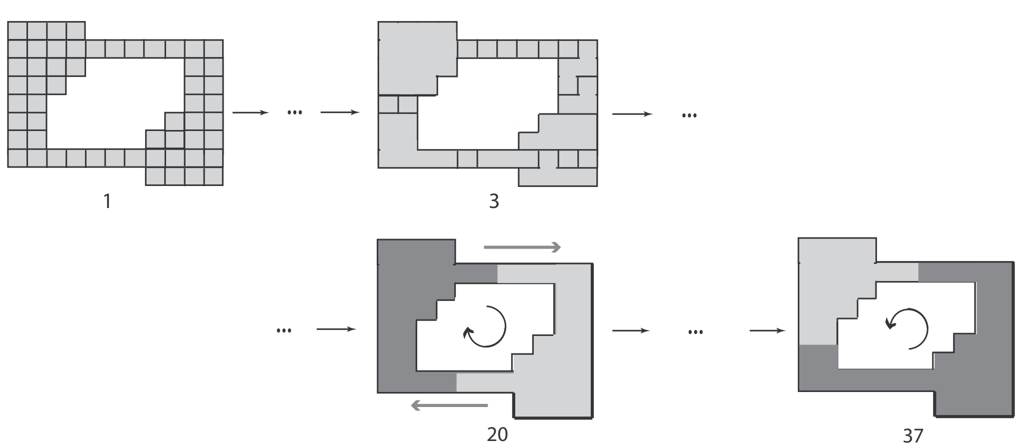

Fig. 5 Cyclic voting procedure. Steps 1, 3, 20 and 37.

Please cite this article in press as: Karpov, V, & Karpova, I (2008), Leader election algorithms for static swarms, *Biologically Inspired Cognitive Architectures* (2015), http://dx.doi.org/10.1016/j.bica.2015.04.001

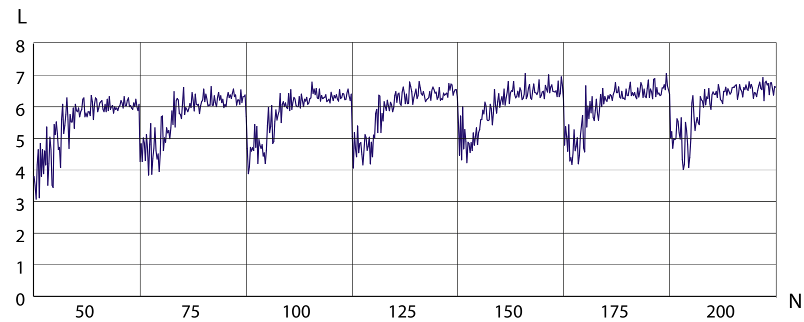

Fig. 6 Distribution of the average number of links L (Y-axis) per different number of robots N (X-axis).

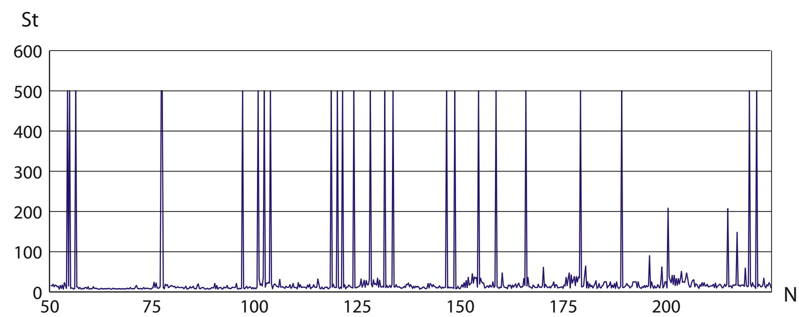

Fig. 7 Dependence of the number of voting steps St (Y-axis) on the number of robots N (X-axis).

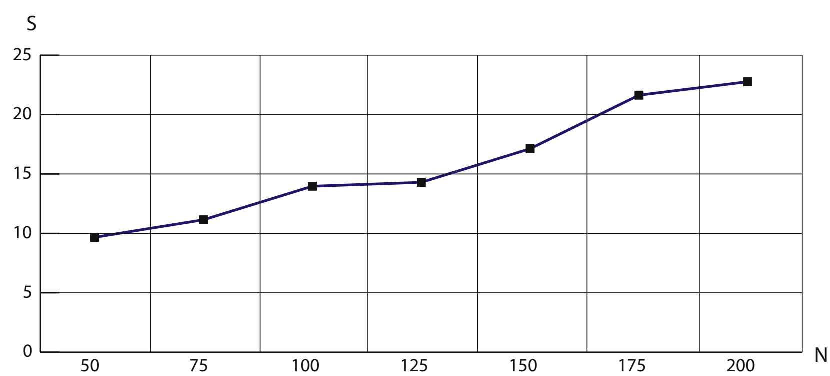

Fig. 8 Dependence of the average number of steps in voting (Y-axis) on the number of robots N (X-axis).

predict, which node will be a leader. So, we can only reach the solution from a set of possible variants.

Task distribution

In the absence of morphological distinctions between agents, role distribution in a static swarm is defined exclusively by the current topology of the system. The distribution process is presented as a well-known procedure of control wave distribution. The initiator of distribution is the leader, whose role is designated as R_0. Direct neighbours of the leader receive an initial message, according to which a role R_1 is assigned to them, and so on. Thus, the role of robot i is defined by the roles in its environment:

$$R_i = \max_{k \in Z} R_k + 1$$

The wave distribution of roles is realised exclusively by local interaction, but there is one essential problem. For successful functioning of the system, M roles are required, with the process of distribution of a wave consisting of L steps (Fig. 9).

If $M = L$, there is no problem. If $M < L$, there are too many performers with role R_M. This is not a good situation, but it is not fatal because in a static swarm we are not interested in role distribution optimisation (like Kalyaev et al. (2011)). If $M > L$, however, the situation is worse, as there is a deficiency of performers, which is extremely undesirable. Agents playing several roles at once (combination of specialisations) can cover the performance deficit, so that

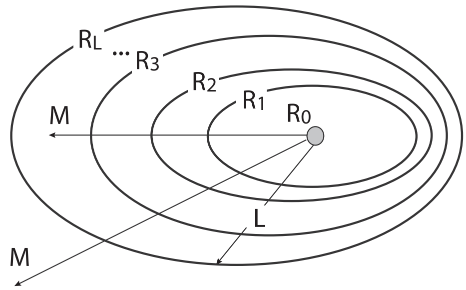

Fig. 9 Actual (L) and required role number (M).

definition of the procedure whereby an agent should assume additional functions seems simple. For example, if an agent with a role number L has no neighbours with roles bigger than L, it means that this agent is at the periphery. Further, if we know that M roles are needed, this agent has to assume roles from L to M. The prevention of any deficiency can also be defined in advance. If there is a group of N agents with maximum connectivity to each agent s (the maximum number of neighbours), it is possible to estimate the minimum number of roles M. Estimation of the M value is

$$M \sim \log_s N$$

Pack-hunting task

We consider the following task. Let there be a field on which there are some agents of two kinds, predators and their victims. The field has a limited size and forms a toroidal surface, i.e. the edges of the field are closed. The task consists of defining rules of predator behaviour so that they are victims for minimum time. In some sense, we have a variant of the pursuit-evasion problem here.

Agents can move and they are equipped with four sensors. Every sensor can detect another agent in front, behind, at the left and to the right of it. The field of vision of the agent is limited. The agent can send a broadcasting message. This message can be accepted only its close neighbours.

A victim is an agent with very primitive behaviour. When a predator appears in its field of vision, it victim runs away in the opposite direction. They are pure individualists. The victim perishes when the predator appears near it in the next field.

We assume that the speed of the victim is twice that of the predator.

For a toroidal surface and low predator speed, it becomes a very complex challenge to catch up with the victim. Thus, formation of groups of hunters, i.e. packs, can be useful.

The formation of a pack is not a primitive process. If predators have a behavioural rule 'IF (sensor detects a predator) THEN (move to it)' then after some time all predators will form compact motionless groups. In this situation, predators need a leader to lead this pack. This leader can be named as Local Leader.

As mentioned above, there is a simple technical escape. Each agent has its individual and unique parameter, an identifier ID. This ID can play the role of the agent's weight. The agent with the greatest weight becomes the Local Leader. The Local Leader does not use the rule 'move to the nearest neighbour'. It tries to find a victim and uses a wandering strategy. All other members of the group follow it.

Fig. 10 shows a set of wandering packs with Local Leaders (red colour).

This compact (packing) motion is not enough for successful hunting. The pack does not catch up with a fast victim. The pack has to surround it. This means that it is necessary to cast beaters. So, we have a situation with role distribution.

A schema of pack-hunting is given below and consists of two stages: Search for Victim and Hunting Procedure.

Search for Victim:

1. A pack formation. Hunters follow Local Leaders.
2. If some agent detects a victim, then start Hunting Procedure.

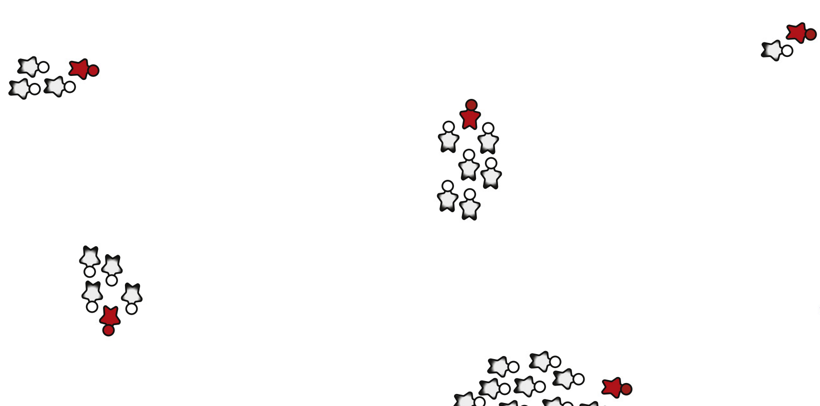

Fig. 10 A set of packs. Local Leaders are red. The others predators (white) follow them.

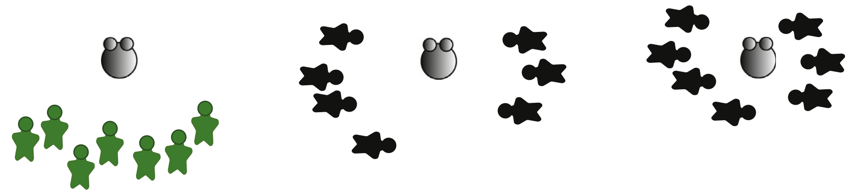

Fig. 11 (a) Hunters detect a victim. (b) Roles are distributed. Hunters bypass a victim. (c) Hunters 'attack' a victim.

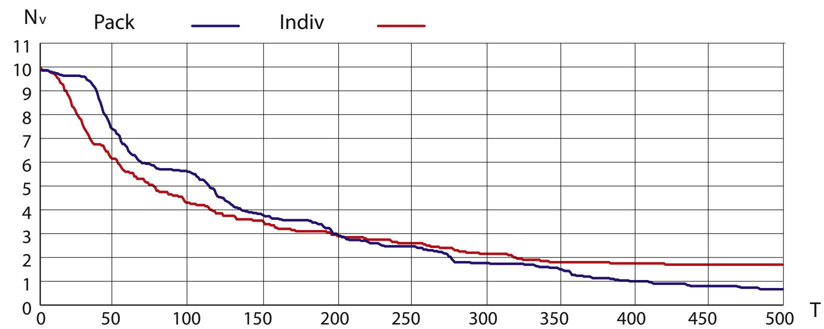

Fig. 12 Dependence of number of victims number on time. 'Pack' shows a pack-hunting strategy, and 'Indiv' shows individual hunting.

Hunting Procedure:

1. Leader election. The initial weight of a candidate is neither its ID number nor the number of neighbours. Initial weights are determined by proximity to the victim.
2. Role distribution. In this task, predators have two roles: the left and right beaters. The main objective is to bypass the victim from two sides and not to allow it to escape.

Fig. 11 shows an example of three stages of the hunting procedure.

A series of experiments was carried out. Parameters of experiments are:

Agent number: 30 hunters (N_h), 10 victims (N_v).
Size of field: 100×100 cells (a toroidal surface).
Hunter's speed: 1 (one step per timepoint).
Victim speed: 2.
Modelling time, T: 500 steps.

Two strategies of hunting were estimated: individual hunting and pack-hunting.

Fig. 12 shows the averaged results of 50 experiments.

It is clear that an individual strategy is more preferable in a situation when there is a lot of 'food'. It is easy to hunt. A pack-hunting strategy gets advantages when 'food' becomes scarce. In general, this strategy is more successful.

We want to underline that the cornerstone of this solution is a symbiosis of the leader election and role distribution procedures.

Conclusions

Simple and effective methods have been proposed for the solution of important problems of swarm robotics such as leader definition and role distribution in a group of agents. Efficiency is understood as the acceptability of robots with limited cognitive abilities (insufficiency of sensory abilities, computing capacities, communications channels etc.—in short, all that is peculiar to a swarm).

The simulation results showed that 93–98 per cent of simulations give full consensus in leader election, and this consensus is reached in a relatively low number of steps. This confirms the validity of the proposed algorithms.

The leader definition and role distribution introduce a differentiation of functions in homogeneous groups of robots, which provides a growth of cognitive abilities of a swarm and transition to complex task solutions. Advantages of this approach are, for example, the task of cooperative hunting by distributed mobile robots. In some sense, we can say that the cognitive abilities of a pack of hunters in a static swarm are higher than those of an 'ordinary' swarm (a homogenous set of individuals without functional differentiation).

Despite its simplicity, realisation of these mechanisms confirms the basic possibility of the formation of very complex structures in the organisation of homogeneous groups, and again confirms that distinctions between a swarm, flock and collective of robots are somewhat artificial.

The static swarm model is a convenient way of looking at swarm robot organisation. While it is limited to exclusively local interaction between agents, it offers all the advantages of a system understood as a network of connected agents, allowing solutions to problems such as storage and data processing, coordinated movement and so on (Karpov, 2013).

In future work, we hope to investigate the mechanism of logical consequence in static swarms, hypothesising that logical consequence procedures can be implemented by exclusively local interaction methods.

Acknowledgements

We gratefully acknowledge the support of RFBR grants Nos.14-01-00817 and 15-01-07900.

References

Berman, S., Halasz, A., Kumar, V., & Pratt, S. (2007). Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *2007 International conference on robotics and automation* (pp. 2318–2323). IEEE, 2007.

DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association, 69*(345), 118–121.

Dewi, T., Risma, P., & Oktarina, Y. (2012). Wedge formation control of swarm robots. *The 14th industrial electronics seminar, electronic engineering polytechnic institute of Surabaya (EEPIS), Indonesia* (pp.294–298).

Gigliotta, O., Mirolli, M., & Nolfi, S. (2014). Communication based dynamic role allocation in a group of homogeneous robots. *Natural Computing, 13*(3), 391–402.

Kalyaev, I., Kapustjan, S., & Ivanov, D. (2011). Decentralized control strategy within a large group of objects based on swarm intelligence. *Proc. of the 2011 IEEE 5th international conference on robotics, automation and mechatronics, RAM 2011* (pp. 299–303).

Karpov, V. E. (2012). Partial mechanisms of leadership and self-consciousness in swarm robotics. *Proc. of the 13th National conf. on artificial intelligence KII-2012, Belgord, 2012, (Belgorod State Technical University, Belgorod, 2012)*, Vol. 3 (pp. 275–283) (in Russian).

Karpov, V. E. (2013). Control in static swarms. Problem definition. *Proc. of the VII International science-practical conference on Integrated models and soft computing in artificial intelligence, Kolomna. (Fizmatlit, Moscow, 2013)*, Vol. 2 (pp. 730–739) (in Russian).

Kim, S.-C., Shin, K. H., Woo, C.-W., Eom, Y. S., & Lee, J. M. (2008). Performance analysis of entropy-based multi-robot cooperative systems in a MANET. *International Journal of Control, Automation and Systems, 6*(5), 722–730.

Kukushkin, I. K., Katalinic, B., Cesarec, P., Zdyb, D., & Kettler, R. (2012). Modeling of mobile robot behavior for line-less Bionic Assembly System. In: B[ranko] Katalinic (Ed.), *Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium* (pp. 0865–0870). Vienna, Austria: Published by DAAAM International, ISBN 978-3-901509-91-9, ISSN 2304-1382.

Liu, L., & Shell, D. A. (2012). Large-scale multi-robot task allocation via dynamic partitioning and distribution. *Autonomous Robots, 33*(3), 291–307.

Loukas, A., Woehrle, M., Glatz, P., & Langendoen, K. (2012). On distributed computation of information potentials. In *Proceedings of the 8th ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing, FOMC'12*. Portugal: Madeira, 19 July 2012, Article number 5.

Nissan EPORO (2009). Nissan EPORO Robot Car 'Goes to School' on Collision-free Driving by Mimicking Fish Behavior – Advanced Robotic Concept Debuts at CEATEC JAPAN 2009. <http://www.nissan-global.com/EN/NEWS/2009/_STORY/091001-01-e.html>.

Pavlovsky, V. E., Kirikova, E. P., & Pavlovsky, V. V. (2010). Modeling of behavior of big groups of robots in the environment with obstacles. *Proc. of the scientific seminar 'Control in distributed network-centric and multi-agents systems'* (pp. 10–13). St.Petersburg: 'Koncern CNII «Jelektropribor»' (in Russian).

Shi, Z., Tu, J., Zhang, Q., Liu, L., & Wei, J. (2012). A survey of swarm robotics system. *Proc. of the Third Intern. Conf. on Advances in Swarm Intelligence* Vol. 1 (pp. 564–572). Shenzhen, China, 2012.

Song, Y., & Zhao, W. (2014). Multi-agent system rendezvous via refined social system and individual roles. *WSEAS Transactions on Systems and Control, 9*(1), 526–532.

Stefanuk, V. L. (2004). Local organization of intellectual systems. Fizmatlit, Moscow, 328 p. (in Russian).

Wu, H., Qu, S., Xu, D., & Chen, C. (2014). Precise localization and formation control of swarm robots via wireless sensor networks. *Mathematical Problems in Engineering*.

Yogeswaran, M., & Ponnambalam, S. G. (2010). Swarm robotics: An extensive research review, advanced knowledge application in practice. *InTech, 5*, 259–278.

Yu, H., Jian, J., & Wang, Y. (2008). Flocking motion of mobile agents with a leader based on distance-dependent adjacency matrix. Lecture Notes in Computer Science. *1st International Conference on Intelligent Robotics and Applications, ICIRA 2008* (pp. 1165–1174). Wuhan; China; 15 October 2008 through 17 October 2008; Code 74343. Volume 5314 LNAI, Issue PART 1, 2008.

Please cite this article in press as: Karpov, V, & Karpova, I (2008), Leader election algorithms for static swarms, *Biologically Inspired Cognitive Architectures* (2015), http://dx.doi.org/10.1016/j.bica.2015.04.001