Inference algorithm for teams of robots using local interaction

Vitaly Vorobiev[1]
[1]*National Research Center "Kurchatov Institute", Moscow, Russian Federation*
Vorobev_VV@nrcki.ru

Abstract
One of the possible approaches to the technical implementation of logical inference in robot groups is considered in paper. The problem is that the usual implementations of the inference mechanism, for example, which are used in expert systems, are difficult to implement to robots that work in a team. It is due to the fragmented knowledge of each robot about the environment where they perform the tasks assigned, the need to exchange data during the inference and monitor this process, etc. In addition, the presence of an inference subsystem may be necessary for emergence of emergent properties in a group of robots. The output subsystem can be used to solve a variety of tasks, for example, choosing the most preferred strategy for the whole collective, building a general picture of the world, planning, etc. In this regard, the paper presents some mechanisms that allow the inference in the logic of predicates of the first order for a team of robots whose interaction with each other is exclusively local in nature. The inference procedure is carried out in the team of robots, which form a special structure, called a static swarm.

Keywords: inference, robot swarm, static swarm, local interaction

1 Introduction

The implementation of inference in groups of robots using local interaction is considered in this paper. The logical output subsystem is one of the robot subsystems, which presence can allow the collective of such robots to exhibit emergent properties, i.e. solve qualitatively more complex tasks (Karpov, 2013). In the simplest case, a robot with such subsystem can choose from the list of possible actions the most preferable from the point of view of the current situation, the goals it faces, and so on. (Stuart, 2006). More complex tasks are solved with its help: planning, building a picture of the world, etc. (Bratko, 1990).

If you don't consider the robotic aspect, then inference is widely used, for example, in expert systems. In addition, there is also a convenient tool that allows you to solve both similar problems and a number of others, implemented in the Prolog language (Bratko, 1990). However, using such mechanisms in real technical devices has its own specifics, especially for robot teams, where the organization of the inference subsystem is highly dependent on the organization of interaction between robots. For example, in (Karpov, 2013) is shown that the implementation of the inference on Prolog leads to a situation where all the resources of the team are aimed at inference and proposed using of linear inference based on the produce.

Indeed, if we consider the case with a single robot, then it is enough just to correctly interpret the sensor data and to make an inference using the search mechanism with a return and unification. A suitable example of a robotic system can be the robot Shakey (Fikes, 1971), where the STRIPS planner was implemented. Another example, described in (Jonsson, 2000), is a scheduler for a spacecraft.

In the case of a team of robots using the Prolog, the following problems can arise:

- The starting problem. There must be a special robot in the group that initiates the inference procedure. If you allow the possibility of initiating an inference to all members of the team, this can lead to large amounts of transmitted data between robots. This will result in a high load of communication channels and data loss during the exchange process. Thus, a mechanism is needed that makes it possible to single out such a robot from the whole team.
- Local character of interaction in a group of robots. The essential limitation is that each robot can exchange data only with a limited number of neighbor robots. This limitation is necessary, firstly, in connection with the high resource-consuming nature of the approach that provides the "all-to-all" connection. Secondly, if we consider biological systems as an object of imitation of collective robotics (Karpov, 2016), it becomes obvious that the connection "all-to-all" is impossible. Consequencely, channels implementing this type of communication will be low-speed, which will interfere with intensive data exchange between robots. This is another argument against initiating the withdrawal of all the robots of the team.
- Fragmentary knowledge base, i.e. each robot of the team most likely doesn't have all knowledge of the area of habitation. Moreover, the individual knowledge of one robot may conflict with the knowledge of another (Vorobiev, 2015), which also introduces certain features in inference . Besides, it is impossible to consider the collective of robots and as a distributed knowledge base. There is no robot, which has a structure that would describe which robots store this or that information (Karpov, 2013). Thus, it is problematic to use address requests to specific robots carrying the necessary data.

In this connection, in order for the inference to possess the completeness property in such a team of robots, it is necessary to search for subgoals in all elements of the collective, i.e. sending requests from the robot that initiates the logical inference to everyone else, using retransmission requests. It is important to remember the received requests within the output of one goal in order to avoid the situation of duplication, when the same multiple request is processed more than once.

- Since the search for subgoals must be performed in the whole team, the problem of processing the response arises. It is, the time required for an answer to reach the robot realizing the inference is inversely proportional to the number of retransmissions of this response. For example, a response that requires three retransmissions will arrive after a response that requires one retransmission. Thus, the order of their location in the addressee's database depends on the time of their arrival. In this connection, the situation described in (Stuart, 2006) in the ninth chapter or in (Bratko, 1990) may arise, where a different ranking order of the rules in the knowledge base leads to different output results. In certain situations this process can be infinite.
- Stopping problem. The procedure of inference shouldn’t end if the robot initiating it could not find in its database facts or rules comparable with the subgoal, since such facts and rules can be in the knowledge bases of other robots.

Thus, we can say that the organization of inference in a team of robots is reduced to the organization of distributed search, which takes into account the aspect of the technical implementation of interaction between robots. In other words, the task is reduced to solving the task of parallelism at the search level (Vagin, 2004).

2 Feasible solution

These problems can be solved as follows:

- Start problem. It is solved by assigning a special robot-leader inside the collective, which will deal with the logical conclusion. This can be done in advance by specifying a hierarchy within the team programmatically, in the process of the team's work by the operator (Couture-Beil, 2010) or directly by the team. The latter option is most preferable, since it does not require any external involvement. In addition, in this case, the collective becomes more resistant to situations when a predetermined leader is dropped for some reason. In this case, one of the existing algorithms for choosing a leader in a team of robots can be used, for example, (Santoro, 2007), (Karpov, 2015), (Karpov, 2015), (Vorobiev, 2017), etc. In the future, the robot leader initiating and executing the logical conclusion will be called the initiating node (IN).
- Local nature of interaction. This is more a technical limitation, which is superimposed on each robot in order to be able to create teams with a number of individuals within them of the order of hundreds. It entails the impossibility of direct communication of the IS with all the other robots of the collective. Therefore, communication is carried out by retransmission of messages received by the robot to its neighbors. In order for the communication channels not to be overloaded with duplicate messages, it is suggested to remember the received messages within the output of one target, compare the newly arrived messages with them and delete the last ones if a repetition was found or remembered if a repeat was not found. A similar mechanism is described in (Vorobiev, 2015b). Technically, a similar local interaction system can be implemented as described in (Karpova, 2016)
- Fragmentary nature of the knowledge base of robots. In this case, it is important to ensure access to the knowledge bases of knowledge of all other robots. This can be achieved by sending out information to its request neighbors with an sub-goal. Those, in turn, relay messages to their neighbors, etc. Thus, the query will reach all available robots at the moment. The received request is processed by the robot, i.e. He seeks in his knowledge base the rules or facts comparable with the sub-goal and sends them back (Vorobiev, 2015b), (Vorobiev, 2015a).
- Handling the answer. The problem can be solved by caching, the mechanism of which is described in (Stuart, 2006).
- The problem of stopping. The conclusion is considered unsuccessful when all robots have not found in their knowledge base comparable with the sub-goal expression, and the robot engaged in logical inference, received information about this (Vorobiev, 2015b).

3 Problem description and results

Task is: there is a team of four robots that explore a certain terrain, interpreting sensory data as some facts. In addition, each robot has predefined rules and facts, for example, **red (X): - dangerous (X)**. The terrain is divided into sectors, each of which has its own unique number. The number of sectors in the example is nine.

In the process of research, the robot knows exactly what sector it is in and forms a fact of the form: sector (**Data_1, Data_2, ..., Data_n, Sector_num**), where Data is the data of the corresponding sensor and Sector_num is the number of the observed sector. In this example, the number of sensor types is four: the temperature sensor, the "friend-of-foe" system, the range finder and the light sensor, and their data are interpreted as follows:

- Temperature. **low** if the temperature is below a certain threshold and **high** - if it is higher or equal to it.
- The system is "friend-of-foe". **many**, if the number of observed active objects with the identifier "foreign" is greater than a certain threshold, otherwise a **few**.
- Range finders. **close** - there are many objects in the sector, **open** - a little.
- Illumination. **light** - high degree of illumination, **dark** - low.

After a certain time, the robots form a static swarm, a structure that is a robot connection scheme obtained at some time (Karpov, 2013). Then the team elects the leader according to the algorithm presented in (Vorobiev, 2017). A priori predefined rules are the rules that describe which sectors are adjacent, which sectors are dangerous and alarming. Dangerous are sectors where there are many observed "alien" objects and which are dark. Alarming are sectors that are adjacent to at least two dangerous sectors.

The knowledge base of each robot is facts obtained on the basis of sensory data and a priori prescribed rules, for example:

```
%Relations from №1 to 3 – results obtained in the process of studying the terrain by robot facts.
%Relation 1. Description the first sector by robot
sector(-,many,-,dark,first).
% Relation 2. Description the second sector by robot
sector(low,few,close,light,second).
% Relation 3. Description the third sector by robot
sector(hight,-,close,light,third).
% Relations from №4 to18 – a priori preset rules and facts
% Relations from №4 to16 – determine the neighborhood of sectors with each other
near(sector(-,-,-,-,first), sector(-,-,-,-,second)).        %Relation №4
near(sector(-,-,-,-,first), sector(-,-,-,-,fourth)).        %Relation №5
near(sector(-,-,-,-,second), sector(-,-,-,-,third)).        %Relation №6
near(sector(-,-,-,-,second), sector(-,-,-,-,fifth)).        %Relation №7
near(sector(-,-,-,-,third), sector(-,-,-,-,sixth)).         %Relation №8
near(sector(-,-,-,-,fourth), sector(-,-,-,-,fifth)).        %Relation №9
near(sector(-,-,-,-,fourth), sector(-,-,-,-,seventh)).      %Relation №10
near(sector(-,-,-,-,fifth), sector(-,-,-,-,sixth)).         %Relation №11
near(sector(-,-,-,-,fifth), sector(-,-,-,-,eighth)).        %Relation №12
near(sector(-,-,-,-,sixth), sector(-,-,-,-,ninth)).         %Relation №13
near(sector(-,-,-,-,seventh), sector(-,-,-,-,eighth)).      %Relation №14
near(sector(-,-,-,-,eighth), sector(-,-,-,-,ninth)).        %Relation №15
near(X,Y):-near(Y,X).                                       %Relation №16
%Relation №17, describing the dangerous sector
danger(sector(-,many,-,dark,-).
%Relation №18, describing the warning sector
warning(S):-near(S,S1),danger(S1),near(S,S2),danger(S2).
```

It is necessary to determine the alarming sectors of a given terrain - warning (S). Since at the moment the system completes the logical conclusion, when all subgoals prove to be proven, the leader is given only one answer. The base of the received facts of each robot is presented in the form of this relations:

```
%Robot #1
sector(-,many,-,dark,second).
sector(low,few,close,light,first).
sector(high,-,close,dark,third).
sector(low,-,-,dark,forth).
%Robot #2
sector(high,few,-,dark,fifth).
sector(low,-,close,light,seventh).
sector(-,few,-,-,eighth).
sector(low,few,-,-,sixth).
%Robot #3
sector(-,many,open,-,fourth).
sector(-,many,-,-,third).
sector(-,few,-,dark,fifth).
sector(-,many,-,light,seveth).
%Robot #4
sector(-,many,open,dark,ninth).
```

sector(high,few,-,-,eighth).
sector(low,many,close,light,seventh).
sector(-,few,-,dark,sixth).

The leader is Robot №2, which as a result gives the sector(low, few, close, light, first) answer, because It is located near to sectors №2 and №4, which are dangerous.

4 Conclusion

The paper examines an example of the implementation of logical inference for a team of robots. The implementation takes into account the specificity of the interaction of robots in the team and determines the mechanisms that allow monitoring this process and avoid situations with congestion of communication channels and irrational use of the computing resources of the team. The principal possibility of using this mechanism for the tasks of collective robotics is shown.

5 Acknowledgements

The project was particularly supported by RSF 16-11-00018 grant (which funded first and second sections of this paper), and RFBR 16-29-04412 ofi-m (which funded third section of this paper and program realization of inference algorithm).

References

Couture-Beil A., Vaughan R. T., Mori G. (2010). Selecting and commanding individual robots in a multi-robot system. *CRV 2010 - 7th Canadian Conference on Computer and Robot Vision*, (pp. 159–166).

Fikes R. E., Nilsson N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, *2*(3-4), (pp. 189–208).

Jonsson A. K., Morris P. H., Muscettola N., Rajan K., Smith B. (2000). Planning in Interplanetary Space: Theory and practice. In Proceedings of *Aips*

Karpov, V., Karpova, I. (2015). Leader election algorithms for static swarms. *Biologically Inspired Cognitive Architectures*, *12*(0), (pp. 54–64).

Santoro, N. (2007). Design and analysis of distributed algorithms, John Wiley & Sons, 2006, 589 p.

Bratko, I. (1990). Prolog Programming for Artificial Intelligence, p. 560

Vorobiev, V. V. (2015a). Logicheskii vivod v staticheskom roe. In VIII International conference " *Integrated models and soft computing in artificial intelligence"* (pp. 539–548).

Vorobiev, V. V. (2015b). The searching task in static swarm . In open semaintic technologies and intelligece systems (pp. 489–494).

Vorobiev, V. V. *(2017). Leader Selection and Clusterization Algorithms in a Static Robot swarm*
In *Mechatr*onika, avtomatizatsiya, *upravlenie 18*(3), (pp. 166–172).

Vorobiev, V. V., Parshikova, E. A. (2015). **.** Application of Multisets for Assessment of the Situation by the Mobile Agent. *Information technologies 21*(6), (pp.421–426).

Karpov, V. E. (2013). Upravlenie v staticheskich roiach. Postanovka zadachi. In VII International conference " *Integrated models and soft computing in artificial intelligence"* (pp. 730–739).

Karpov, V. E. (2016). Models of social behavior in the group of robotics. *Large-scale system control*, (59), (pp. 165–232).

Karpova, V. E. (2016). Pseudo-Analog comminication in robot groups. In *Mechatr*onika, avtomatizatsiya, *upravlenie 17*(2), (pp. 94–101).

Stuart R., Norvig P. (2006). *Artificial intelligence. A modern approach.2nd edition p. 1408*