Evolutionary co-evolution of robot sensors and behaviour for multi-robot cooperation.

NAEEM GANEY, University of Cape Town - South Africa
NTOKOZO ZWANE, University of Cape Town - South Africa
JAE JANG, University of Cape Town

Contents

1. PROJECT DESCRIPTION

1.1. Introduction

Modern field of robotics focuses on creating intelligent robots [Wang and Tan 2006]. These robots are capable of perceiving the surrounding environment and acts autonomously to accomplish a given task. They are particularly useful in solving problems in environments that are unsuitable for human workers. Multi-robot system coordinates the behaviour of a team of multiple robots in a system to accomplish a cooperative task [Trianni 2008]. Multi-robot system has the benefit of providing redundancy and wider geographical coverage as well as providing solutions to problems that cannot be solved by single robot [Stone and Veloso 2000].

However designing these sophisticated multi-robot systems is not free of challenges. In order for robots to accomplish a task, robots need to be taught the relevant intelligence to accomplish the task (Artificial Intelligence) [Russel and Norvig 2010]. In order to build a robot and make it accomplish a task, we need to develop the body and the brain of the robot. We refer to this as the controller and morphology of a robot. Designing an effective controller and the morphology for a given task requires extensive knowledge regarding the robots, task and the environment involved. Furthermore the robots are designed for a very specific task and are not suitable to be used in wide range of applications [Nolfi and Floreano 2000].

Evolutionary robotics (ER) [Nolfi and Floreano 2000] was introduced to overcome these issues. This approach combines the field of robotics and machine learning (a subfield of Artificial Intelligence) to solve the above-mentioned design problem, by using evolutionary algorithms (EA) to adapt the behaviour and morphology of the robots rather than requiring a detailed hand design from a Robotics Engineer.

The unsupervised learning nature of EAs makes it possible to train robots in domains with low supply of training data. Ideally the training are done offline in a simulated environment to increase the speed of learning and avoid engineering problems and possible damage to robots during learning (crashing into walls etc.). After the evolution process completes the evolved controller and morphology are ported to robots in a real world environment.

There are abundance of researches done for evolving the controllers for an evolutionary multi-robot systems. However little research has been done on co-evolving the morphology along with the controller, even though there are papers emphasizing the importance of morphology on the performance of robot systems [Lichtensteiger and Pfeifer 2002]. This project aims to prove that co-evolution of controller and morphology is the best Evolutionary Robotics (ER) approach by comparing it with approaches that only evolve the controller or the morphology of an autonomous mobile robot.

A simulated multi-robot system will be developed to test and evaluate the robot teams evolved with different approaches. The robots will be an abstraction of the Khepera III robot designed by the K-Team Corporation, using an Artificial Neural Network (ANN) as a controller and multiple sensors of different types.

Robot Teams will be given two cooperative tasks to accomplish. The first task requires autonomous mobile robots to cooperate in a trash collection exercise. Robots navigate the simulated environment in search for trash, pick it up and return the trash to a home base. An element of cooperation is required because some trash objects cannot be transported by a single robot, hence the requirement for multiple robot agents to work together. The second task is to organise the trash in a pre-specified manner at the home base.

1.2. Problems

Accurately designing the controllers and morphologies of multi-robot system to carry out wide range of collective behaviour task is very difficult and time consuming. Evolutionary Robotics provides a solution to this problem by allowing the design process of robot teams to be automated.

Research in the field of Evolutionary Robotics also allows us to build an arbitrary robot system : a similar methodology and algorithm can be applied to other robotics applications. With the ability to build an arbitrary robot system, we can reduce the need for specialised skills in a particular robot system or task.

The same techniques can be applied to robots of different size and morphology, different tasks and and competition instead of cooperation between robots.

1.3. Importance

Finding a good design approach for multi-robot systems are important as well trained team of robots can be used to complete tasks that are too remote or hazardous for humans to attempt [Arkin 1998]. Modern applications includes collective construction, surveillance, search and rescue [Trianni 2008].

1.4. Difficulties

Apart from the inherent complexity of designing a robot, the key issues for multi-robot systems for collective tasks are the complex design process of the controllers and morphologies. Global behaviour of the robot team is the emergent behaviour resulting from dynamic interactions of individual robots with themselves and the environment. It is difficult to identify the individual behaviours of robots that would result in the desired global behaviour [Nolfi and Floreano 2000]. Different configuration of morphologies offers different advantages. Some configuration allows the robots to excel at a specific task but other configuration may allow the robots to be used in more general applications. Deciding on a configuration that allows wide range of application as well as maintaining sufficient performance requires extensive knowledge.

Another problem is designing an accurate simulation version of the real world. The real world is stochastic and noisy. A good quality robot design in a simulated environment may not work as well in the real world. Environment factors such as temperature, ground friction and power limitation may affect a robot's progress [Jakobi et al. 2011]. The gap between the simulation and the real world does not affect our project as we will be dealing with the simulated environment only.

Lastly, robot agent systems need to undergo large amounts of training. Finding a good solution is CPU intensive, requiring High Performance Computers and long periods of waiting time.

2. PROBLEM STATEMENT

Certain tasks are too complex to be solved with a solution that is written as a simple heuristic, the particular problem in this cases is a foraging and construction task, which requires co-operation between a team of robot agents. Our aim is to figure out which of the multiple approaches to the design of the controller and sensory configurations (morphology) of this team of homogeneous robots is best fit for a problem of this nature.

A comparison will be made between system agents whose ANNs have been trained using different approaches:

(1) Evolving the controller of the agents using a fixed robot morphology
(2) Evolving the morphology of the agents using a fixed robot controller
(3) Simultaneously evolving both the controller and morphology of the agents

We intend to use the first two approaches (evolving either only the controller or the morphology) as control cases to help support our hypothesis. That the systems with simultaneous controller and morphology evolution will outperform the other two cases.

3. PROCEDURES AND METHODS

We will be comparing three different approaches to adapting team of robots to complete a cooperative task. Controller-only, Morphology-only and co-adaptation of controller and morphology. Teams trained with different approaches will be tested on a foraging task on a common platform to compare the effectiveness of each approaches.

The MASON library will be used to create a system to simulate the multi-agent environment. System will be built with java, and will have a graphical user interface to allow the user to input various experiment parameters and be able to see the visual representation of agents as they train.

The environment will consist of team of homogeneous robots and various type of objects with different dimensions that either need to be collected or avoided. Quantity of robots and objects to be placed will be specified by the user. The Goal of the robot team is divided into two stages, stage 1 is to acquire and collect the target objects and bring it back safely to a designated area (home region / nest). Stage 2 requires the collected objects to be arranged in order of their dimensions. Each robot will have a neural controller defining its behaviour, sensory system (morphology) that provides the input informations to the controller and battery system powering the robot with varying consumption rate depending on the morphology of the robot. Based on the approach used, different component of robot will be evolved so that maximum amount of target objects can be retrieved and arranged in the shortest amount of time.

3.1. Controller evolution

For the controller evolution approach, the experimenter will decide on a fixed set of morphology to be used by each individual robots throughout the experiment. Then the neural network controller is evolved to adapt its behaviour for the pre-defined morphology.

This is the most common approach adopted by modern applications. Neuro-evolution method selected for this approach is Symbiotic Adaptive Neuro-evolution (SANE) [Moriarty and Mikkulainen 1996] SANE differentiates itself from other NE techniques by evolving population of neurons instead of population of networks. This has the advantage of maintaining diversity as well as avoiding the competing convention problem. The network is formed by randomly selecting individual neurons to construct the network and each neuron is evaluated by averaging the fitness of the whole network. This technique has the advantage of maintaining population diversity as well as avoiding the competing convention problem.

3.2. Morphology evolution

In this approach, the morphology will be evolved. Given a fixed controller (assumed to be already evolved), the evolutionary process will allow for changes in adapting the sensor types and placement on the robot, this will involve selecting the best configuration of the sensors on the robot, evolving the number of each of sensor type as well as the different sensor parameters (i.e: orientation, bearing and field of view).

In this implementation, the topology and weights of the links will be evolved, however, only the neurons and links corresponding to the sensors will be subject to evolution, the rest of the ANN will remained fixed. It has been shown [Lee et al. 1996] that the morphology of robots can be effectively evolved using a Genetic Algorithm to evolve a floating point string representing the different parameters of the robot morphology.

There are two different evaluation criteria that can be used. First is the performance of the robot team, i.e whether the were able to complete the task after specified training time. Secondly the training time required to reach the threshold performance, how much time was required to reach satisfactory performance.

3.3. Co-Evolution of Controller and Morphology

The Co-Evolution approach involves adapting both the morphology and controller of the robot agent. The Morphology and Controller will both be put under evolutionary control. Sensor types and their respective positions determine the morphology of a robot agent. The controller of a robot is represented by an Artificial Neural Network. The inputs to the artificial neural network are the sensory readings and the outputs drive the right and left wheels of the robot.

The controller and morphology will be put under evolutionary control. Controllers have been successfully trained using Neuro-Evolution techniques. Common Neuro Evolution techniques only adapt the topology and weights of Artificial Neural Network. Using existing Artificial Neural Network framework, we can add extra sensory parameters as input nodes to the Artificial Neural Network.

NEAT, a common method used to evolve the topology and weights of an Artificial Neural Network can be adapted to include the sensory configurations of a robot to be adapted in addition to the controller of a robot. Because the weights (and not the input nodes) are adapted by the Evolutionary Algorithms, recurrent network configurations will be made use of. A recurrent connection weight represents a characteristic of a sensor.

4. RELATED WORK

4.1. Khepera Robots

The K-Team Corporation have engineered a robot for use within the field of evolutionary robotics. [Spronck et al. 2004] uses the Khepera robot for a *box-pushing application* and has proven that the controller designed was able to direct a simulated and real-life version Khepera robot. The Khepera [Mondada 2005] robot has been used in many other applications for the development of Artificial Intelligence in Robot Technologies. The Khepera robot has many sensors that can be configured and has good processing power which makes it an attractive choice for research in Evolutionary Robotics.

4.2. ANN for Robot Controller

[Nolfi et al. 1994] explains the appropriateness of using an Artificial Neural Network for controller design. The weights can be adapted by using a common Evolutionary Algorithm, NEAT [Moriarty and Mikkulainen 1996]. An Artificial Neural Network is an attractive choice for a robot controller due its robustness to noise and fault tolerance.

4.3. Co-Evolution

The focus of co-evolution is to adapt the morphology and controller of the robot simultaneously. [Lund 2003] refers to the morphology and controller as the brain and the body of a robot and adapts the controller by using Genetic Programming and a Genetic Algorithm for adapting the Morphology of the robot.

[Lee et al. 1996] used a different approach to co-evolution, where the controller and morphology of the robots was evolved using different techniques. They also used Genetic Programming for the controller and a Genetic Algorithm for the morphology of the robot. This is key since we are able to apply the morphology evolution style they used to the adapt only the morphology of our agents for part of the project.

4.4. Multi Agent Systems

Multi-agent systems [Woodridge 2009] is a system that involves more than one individual agent (or robot). The agents can either be competitive or cooperative. In a cooperative system - the agents succeed and fail as a team. In competition, an individual agent benefits at the expense of his another robot. Cooperation is required when an individual robot is not able to complete a task without the assistance of a fellow robot in the same system. For example, the box-pushing application [Petrovic 1999] requires individual robots to pair up with one or more robots in order to be able to have enough force to push a box.

An application by Waibel et al. involves to cooperation between multi agent robots for a box pushing application which is very similar to ours [Waibel et al. 2009].

5. ANTICIPATED OUTCOMES

Our first outcome is to have a very good simulator for our robot application. We want to abstract the environment, the Khepera robot and the Khepera sensors as close as possible. We also want to be able to use the simulated environment to create an arbitrary robot, adapt the robot?s behavior and be able to test the fitness of a robot instance critically. We also want to be able to the simulation quickly so that we can find better solutions in our limited amount of time. We want to be able to make use of the High Performance cluster (HPC) and exploit parallel programming to make our training faster. By making it faster - we can run more tests.

We are hoping to find effective methods to train autonomous mobile robots. Our first outcome is that the robots inhibit the desired behavior that we wish for them to inhibit. We want to be able to show that our solution can be applied to many different applications.

We hope to outperform a heuristic model for the task. We also want our robot task to be done in the most efficient manner possible - in real life, robots have limited battery power and therefore the need for an efficient solution is important.

We want to be able to show some cooperation between our autonomous mobile robots. That robots can work together to work on tasks that they cannot accomplish alone.

Finally we want to be able to compare the results of the three different approaches and conclude that our hypothesis is valid, that the co-evolution approach outperformed the controller evolution and morphology evolution approach.

In summary, Our outcomes will be evaluated by the following criteria

(1) How good the simulated environment is.
(2) The speed of training the autonomous mobile robot.
(3) Successful completion of the task
(4) The level of multi-robot cooperation
(5) How efficient the robots are at doing the task.
(6) A list of results and conclusions about various techniques we have tried out to show which is the most appropriate approach for each task.We can conclude whether it is worthwhile adapting the morphology and the controller of an autonomous mobile robot.
(7) We can find the most appropriate morphology (effectiveness, robustness and efficient) for the foraging task application

5.1. Expectations

We expect that our robots work together to complete a task. We expect that our robots will do so efficiently. And we also expect to have a list of results and conclusions about various techniques and a sound understanding about which techniques work and which ones don?t. We also expect that the co-evolution of controller and morphology to outperform the other two approaches.

6. PROJECT PLAN

6.1. Deliverables

(1) Setup the simulated environment and have an accurate abstraction of the Khepera III Robot and its sensors.
(2) We need to do some form of basic controller in a robot (collision avoidance) to show that our simulated environment works and that we can train a robot to accomplish a basic collision avoidance task. This will also show that we can abstract the controller of a robot by using an ANN.
(3) We want our robots to be able to search for objects in the environment, pick them up and return them to the home base.
(4) We want to show some level of cooperation between robots by introducing bigger boxes so that robots need to work together in order to accomplish the task.
(5) We want to find methods to make the process faster. Mainly, we want to show that we are accomplishing better results. We will investigate various evolutionary computing techniques to see which method is most appropriate to the task.
(6) We want to start adapting the morphology, as well as the controller of the robot. We want to be able to have a search space for morphology configurations and find the morphology and controller combination which is most effective for the task. We will also be introducing a power penalty for each move - some sensors use more power than others.
(7) Task 2 is to be able to arrange the boxes in the home environment. This deliverable will show that the boxes are arranged in the manner specified.
(8) Further optimisation techniques on both Task 1 and Task 2 to show that are we investigating methods that can possibly enhance the quality of our solution. The

main outcome of this deliverable is to show that we are constantly investigating various machine learning techniques (possibly developing our own) to find the most efficient solution.

(9) Make use of the High Performance Cluster to speed up the process of our analysis. The speed up will allow us to test more techniques, train for longer iterations and search a wider search space.

(10) Final report, outline of our results and conclusions based on our results.

6.2. Risks

— **The risk of a team member dropping out**
This will be mitigated by created disjointness in the project tasks that each members needs to do. The risk will be managed by splitting the third individual's work between the other two members of the group.

— **There is the risk that the scope of the project is too big**
We will mitigate this risk by isolating tasks and should the risk become a reality, we will be able to cut out the task completely without affecting the rest of the project.

— **There is a risk of being unable to adapt the current framework to our application.**
We have made contact with the developer of the framework and have consulted with him. Should the current framework be too difficult to adapt - we will develop a simpler framework.

— **The risk of the Evolutionary training being too slow.**
This will be solved by making use of the High Performance Cluster.

— **'Complexity in configuring the high performance cluster.**
This has been mitigated as all members have taken the High Performance Computing module and further assistance can be obtained from Chris Laidler

— **We face the risk of the learning being too steep**
We plan to mitigate this risk by being in close contact with Dr. Geoff Nitschke to continuously obtain resources on the field of Evolutionary Robotics. We will also make use of books on the topic which can be obtained from UCT Main Library.

— **Time management and the risk of being overwhelmed by other modules/projects.**
This is being mitigated by working during the vacation when members are free of the responsibility of doing other courses. Furthermore, we have assigned dedicated slots in our schedule to work on this project.

6.3. Work Allocation

All work done in setting up the simulated environment and the High Performance Cluster will be done collaboratively. Thereafter, the 3 above mentioned approaches will be taken on individually by each group member. The three approaches : controller adaptation, morphology adaptation and co-evolution of morphology and controller will be completed by Jae Jang, Ntokozo Zwane and Naeem Ganey respectively. [Parisi et al. 1991]

Thereafter, the team will regroup to compare results on their respective approaches with the hope of proving our hypothesis.

REFERENCES

Ronald Arkin. 1998. Behaviour Based Robotics. *MIT Press* (1998).

Nick Jakobi, Phil Husbands, and Inman Harvey. 2011. Noise and the reality gap : The use of Simulation in Evolutionary Robotics. (2011).

Wei-Po Lee, John Hallam, and Henrik H Lund. 1996. A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 384–389.

Lukas Lichtensteiger and Rolf Pfeifer. 2002. *An Optimal Sensor Morphology Improves Adaptability of Neural Network Controllers*. Springer.

Henrik Hautop Lund. 2003. Co-Evolving Control and Morphology with LEGO Robots. (2003).

Francesco Mondada. 2005. Mobile Robots for a Synthetic Approach to Interdisciplinary Research. *LSA - STI - EPFL* (2005).

Moriarty and Mikkulainen. 1996. Efficient reinforcement learning through symbiotic evolution. (1996).

Stefano Nolfi and Dario Floreano. 2000. Evolutionary Robotics : The body, Intelligence, and Technology of Self-Organising Machines. *MIT Press* (2000).

Stefano Nolfi, Dario Floreano, Orazio Miglino, and Francesco Mondada. 1994. How to evolve autonomous robots : Different approaches in evolutionary robotics. *Proceedings on the Artificial Life IV Conference* (1994).

Domenico Parisi, Stefano Nolfi, and Federico Cecconi. 1991. Learning, Behavior and Evolution. *Toward a practice of autonomous systems* (1991).

Pavel Petrovic. 1999. Overview of Incremental Approaches to Evolutionary Robotics. *Norweigen University of Science and Technology* (1999).

Stuart J. Russel and Peter Norvig. 2010. *Artificial Intelligence : A modern approach*. Pearson Education.

Pieter Spronck, Ida G. Sprinkhuizen-Kuyper, and Eric O. Postma. 2004. Evolutionary Learning of a Neural Robot Controller. (2004).

Peter Stone and Manuela Veloso. 2000. Multiagent Systems: A Surv e y from a Machine Learning Perspecti v Multiagent Systems: A Surv e y from a Machine Learning Perspecti v Multiagent Systems: A Surv e y from a Machine Learning Perspecti v Multiagent Systems: A Surv e y from a Machine Learning Perspecti v Multiagent Systems: A Surv e y from a Machine Learning Perspecti v Multiagent Systems : A survey from a machine learning perspective. *Autonomous Robotics* (2000).

Vito Trianni. 2008. *Evolutionary Swarm Robotics Evolutionary Swarm Robotics : Evolving Self-Organising Behaviours in Groups of Autonomous Robots*. Springer.

Markus Waibel, Laurent Keller, and Dario Floreano. 2009. Genetic team composition and level of selection in the evolution of cooperation. *Evolutionary Computation, IEEE Transactions* (2009).

Lingfeng Wang and Kay Chen Tan. 2006. *Evolutionary robotics: from algorithms to implementations*.

Michael Woodridge. 2009. *An Introduction to Multiagent Systems*. John Wiley and Sons, Inc.