

# Collective Robotics, Neuroevolution and System Complexity

## Literature Review

Scott Hallauer

Department of Computer Science  
University of Cape Town  
Cape Town, South Africa  
[scott.hallauer@gmail.com](mailto:scott.hallauer@gmail.com)

### ABSTRACT

Evolutionary robotics has shown great promise in the automated design of autonomous, collaborative robotic systems. Through various techniques of neuroevolution, robotic neural controllers and morphologies can be evolved that are effective in achieving complex tasks. Strategies that have been proposed for the neuroevolution of artificial neural networks include single- and multi-objective variations of the NEAT approach. Recently, neuroevolution has been applied to the study of the costs of complexity in natural evolution, with useful applications in robotics. This paper reviews the state of the art in collective robotics and neuroevolution, and concludes by identifying gaps in the literature – namely the lack of standardisation in definitions of system complexity and the need for further research and development in multi-objective co-evolution of morphology and neural controllers in collective robotics.

### CCS CONCEPTS

• Computing methodologies~Multi-agent systems • Computing methodologies~Evolutionary robotics • Computing methodologies~Neural networks

### KEYWORDS

Collective Robotics, Evolutionary Robotics, Neuroevolution, Morphological Complexity

## 1 INTRODUCTION

Over the past three decades, the field of collective robotics (especially in combination with evolutionary robotics) has experienced immense growth. Numerous techniques have been developed for evolving neural controllers and morphologies for robotic agents which facilitate the emergence of group behaviour. This strategy draws inspiration from the problem-solving strategies of social insect colonies, which effectively divide work amongst individuals and collaborate to achieve complex tasks [20, 35]. Also taking insight from biological systems are the structures of artificial neural networks and the process of artificial evolution in evolutionary robotics [28, 38].

Besides enabling the automatic development of simple, inexpensive robots which can collectively execute complex actions [36], this field also provides a novel approach to studying

longstanding questions in the area of evolutionary theory [22]. Of specific interest is research into the evolutionary origin of morphological and neural complexity, and the related costs of complexity in development [15, 40].

This paper reviews the literature covering the fields of collective robotics, neuroevolution and system complexity. Firstly, a high-level explanation is given for how collective robotics systems are simulated and evolved. Thereafter, an in-depth analysis of the strategy and process of neuroevolution is provided. This includes an introduction to the concepts of artificial neural networks and evolutionary algorithms. Specific examples of state-of-the-art algorithms for evolving artificial neural networks through single- and multi-objective optimisation are also described. Finally, definitions for aspects of system complexity in robotics are outlined and previous work in the area is reviewed.

## 2 COLLECTIVE ROBOTICS

*Collective* or *swarm robotics* (SR) is the problem-solving approach whereby a homogenous, multi-robot system accomplishes specific task goals through collaborative *group behaviour* [35]. In this context, the term “homogenous” is referring to groups of robots with highly similar, if not identical, neural controllers and body plans. There are many real-world applications for SR systems, such as hazardous waste clean-up and oceanic environmental monitoring [50]. This section reviews previous work in the SR field and explores the connection with *evolutionary robotics*.

### 2.1 Group Behaviour

The 1990s saw a surge of interest in the application of social insect group behaviour patterns to the field of robotics [20, 34, 35, 37]. Similar to how ant colonies consisting of many simple individuals can achieve tasks far beyond the capabilities of any single ant, it was envisaged that simple robot controllers could collectively produce complex behaviour. The successful implementation of such a SR system requires a form of decentralised control similar to that employed by social insects [35]. This requires robot controllers to be designed in such a way that the local interactions produced by their behaviour result in global coordination [11]. Modelling such control architectures is not an elementary computational task and so the use of *artificial*

*neural networks* and *evolutionary algorithms* (elucidated in section 3), both inspired by biological systems, has been adopted as a standard approach [35, 37, 58].

## 2.2 Environment Simulation

There are a variety of software packages available for simulating collective robotics systems. SimbotCity is one such robot population simulator developed by Kube and Zhang for their related work in the early 1990s [34, 35]. In their implementation, a modified fixed-priority *subsumption* architecture [13] was used for managing behaviour arbitration.

Many modern alternatives have been developed in recent years. ARGoS was introduced in 2011 with the primary objective of providing a simulation environment for large heterogenous robot swarms [46]. In 2013, a C++ based simulator named RoboroBot! was released which specialises in evolutionary swarm robotics [12]. In 2015, Kilombo was announced as virtual simulator for the popular Kilobot robot [49] (traditionally used in physical simulations) which greatly expedites development and allows for pre-screening of potential controller algorithms [31]. Even more recently, in 2019, a paper was published detailing a massive multi-agent simulation environment, SCRIMAGE, for simulating collaborative robots [19].

## 2.3 Evolutionary Robotics

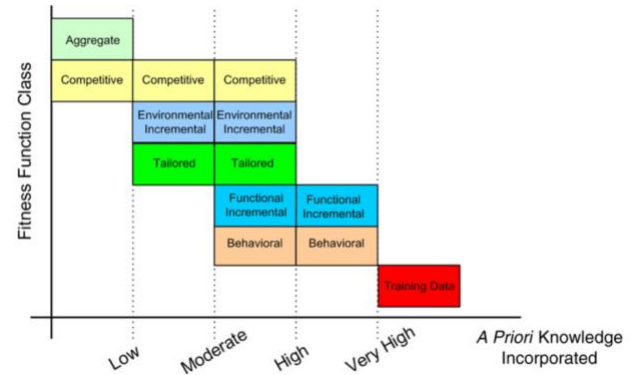
*Evolutionary robotics* (ER) is the field of research into the automatic design of autonomous robot controllers and morphologies through processes which mimic *natural evolution* [28]. The basic premise of natural evolution is that a population of genetically similar, yet unique, organisms compete in a common environment and, through the process of *natural selection*, the “fittest” individuals survive and reproduce [62]. In this manner, superior traits are inherited over successive generations and the population as a whole tends to become better adapted to its environment.

**2.3.1 Automated Robot Design.** Traditionally, robotic controllers and morphologies are manually, and often laboriously, designed by human engineers [43]. This is an expensive endeavour that has hindered potential niche applications of robotics which are not feasible without the economies of scale [36]. ER provides an alternative, cost-effective approach to robot design whereby variations are automatically assessed for functionality through simulation and iteratively improved over multiple cycles. Additionally, using this method of *artificial evolution*, new models of cognition can be developed that, through experimental analysis, may prove better than existing hand-designed solutions [45].

**2.3.2 Fitness Functions.** A key aspect of the artificial evolution process used in ER is the measurement of fitness for the developing robotic components. During each cycle, or *generation*, robots (consisting of the current state of the evolving morphology and/or controller) are tasked with engaging in an evaluation period. Each robot morphology and/or controller (whichever is being evolved) is then assessed based on its performance using a

*fitness function*. As the final step in each cycle, an *evolutionary algorithm* (see section 3.2) is applied which uses the results from the fitness function to selectively propagate the fittest robots for the next generation [43].

There are seven broad classes of fitness functions as defined by Nelson, Barlow and Doitsidis [43]. This classification system is based on the level of *a priori* knowledge incorporated by the fitness functions (Figure 1). Firstly, there are *training data fitness functions* which use data sets for comparing behaviour and scoring fitness. These are commonly used in mimetic learning, where a robot attempts to replicate the behaviour of a trainer (see [21]). Secondly, there are *behavioural fitness functions* which assess fitness based on how a robot goes about performing its tasks rather than what it ultimately achieves (see [8]). In contrast, there are also *aggregate fitness functions* which select based only on high-level ability to accomplish a task, without consideration of the behaviour involved (see [29]). Combining characteristics of both *behavioural* and *aggregate fitness functions*, there are *tailored fitness functions* which contain behaviour-measuring terms and aggregate terms in the fitness calculation (see [47]). Then there are *functional incremental fitness functions* which are used to iteratively select for more complex abilities by progressively adapting the fitness function over the course of robot evolution (see [61]). Similarly, there are *environmental incremental fitness functions* which gradually increase the difficulty of the environment in which the evolving robots interact (see [41]). Finally, there is *competitive* and *co-competitive fitness selection* which involves direct intra-population competition between individuals, where interactions influence other robots’ behaviours and their resulting fitness evaluation (see [14]).



**Figure 1: Chart relating classes of fitness functions to levels of incorporated *a priori* knowledge [43].**

**2.3.3 Evolving Group Behaviour.** ER enables the development of robot controllers which facilitate the self-organising, group behaviour patterns required in SR systems. This is possible by simulating environments which require collaboration between individuals for successful task completion and the subsequent selection and propagation of effective controllers.

Baldassarre, Nolfi and Parisi [7] demonstrated that simulated robots evolved for the ability to move together toward a light

target were able to display collective behaviours with interesting properties. Forms of *situated specialisation* were observed where robots with identical controllers expressed varying behavioural roles dependent on their specific circumstances in the group.

Similar forms of specialised behaviour (albeit in a heterogenous population) were specifically selected for in research by Nitschke, Schut and Eiben [44] where the performance of three cooperative co-evolution methods was compared. In this study, robot controllers were evolved for a simulated collective construction task. The results indicated that the method of *collective neuroevolution* (CONE) outperformed the other two tested methods: *multi-agent enforced sub-populations* (MESP) and *cooperative co-evolutionary algorithm* (CCGA).

Distinguishing it from the majority of studies in the field, research in 2002 by Quinn *et al.* [47] explored the evolution of controllers for real robots. Here, robots, minimally equipped with infrared sensors, were evolved for a formation movement task. It was observed that a team of robots successfully evolved to adopt individual, specialised roles in order to complete the assigned task.

More recent work by Duarte *et al.* [23] similarly investigated the evolution of collaborative control systems for a physical swarm of robots. The neural controllers were initially evolved, in simulation, for various swarm robotics tasks (e.g. homing, dispersion, clustering and monitoring) before being transferred to real aquatic surface robots. Results demonstrated that the controllers successfully achieved similar task performance in a real-world, uncontrolled environment as they did in simulation.

**2.3.4 Researching Natural Evolution.** Another interesting application of ER is as a novel approach to the study of evolutionary theory itself, through the new experimental technique of computer simulation [22, 45]. This allows us to investigate previously inaccessible questions, such as the role of a complexity cost in evolution (see section 4.3). Therefore, the combined use of ER and SR techniques may assist in shedding new light on research into the evolution of human brain size and the social brain hypothesis, which remains an area of open inquiry [24, 42].

A 2015 study by Ferrante *et al.* [26] delved into a further area of interest in evolutionary theory: the origin of self-organised *task partitioning* in populations. This is particularly common in insect communities where distinct behavioural roles are taken on by different individuals to accomplish a shared task [48]. To approach this research problem, ER and SR techniques were used to investigate task specialisation in populations of identical robots. The results indicated that task partitioning is favoured in environments with features that lower switching costs and increase net group efficiency.

These are just two of many examples for how ER has been utilised in the research of evolutionary theory. For a more in-depth analysis of other earlier studies in the field, the 2000 review by Webb [60] is a good source.

### 3 NEUROEVOLUTION

*Neuroevolution* is a broad term encompassing an array of approaches to training *artificial neural networks* (ANNs) for specialised behaviours, all of which draw on insights from neuroscience and evolutionary biology [27]. ANNs are used in many different problem spaces including pattern recognition, prediction, optimisation and associative memory, as well as robot control [30, 58]. This section introduces the concept of an ANN, explains their applicability in *evolutionary algorithms* (EAs) and describes a few common methods of neuroevolution in the field of ER.

#### 3.1 Artificial Neural Networks

ANNs are structured as weighted directed graphs with nodes representing artificial neurons and edges representing the connections between neuron inputs and outputs [30]. An example of the basic composition of an ANN is presented in Figure 2, where environmental information flows from *input nodes* (or *units*) to *output nodes* via weighted connections which may pass through *hidden nodes* within the network. The original inspiration for this architecture, as evidenced by the name, can be traced back to the fundamental structure and operation of biological nervous systems [38]. This alternative approach to computation makes ANNs far more effective in solving certain problems, such as pattern recognition, which traditional computational architectures struggle to handle efficiently.

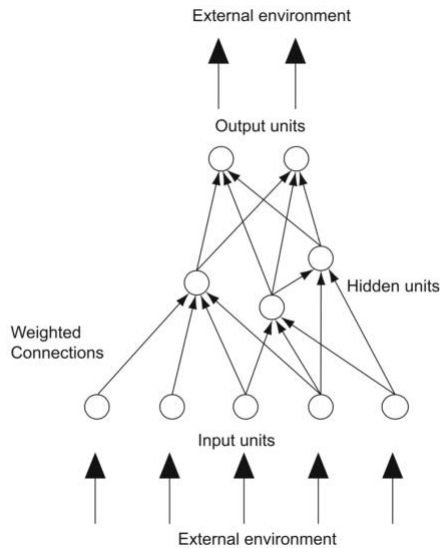
**3.1.1 Classification of ANNs.** There are two major types of ANNs: *feed-forward* networks and *recurrent* networks [30]. In feed-forward networks, the graph is acyclic, and data only moves from input nodes to output nodes. Alternatively, in recurrent (or *feedback*) networks, cycles occur in the graph whereby information from previous activations can feed back into the network and alter the inputs to each node. As a result of this operation, feed-forward networks are considered to be *static* (i.e. only produce a single set of output values for a given input value) whereas recurrent networks are considered to be *dynamic* (i.e. produce a sequence of output values for a given input value).

**3.1.2 Encoding of ANNs.** ANN architectures can be represented using either *direct encoding* or *indirect encoding* schemes [33, 63]. In the former approach, all the structural information (i.e. every node and connection) of the ANN is encoded in an unabridged form. This representation results in a one-to-one mapping from the *genotype* (or encoding) to the *phenotype* (or architecture). At the other extreme, with indirect encoding, only the most pertinent aspects of the network (e.g. the number of hidden layers) are encoded such that a variation of the phenotype can be derived from the genotype [63].

**3.1.3 Training Methods.** Training ANNs involves iteratively updating connection weights to achieve greater task performance. The ability of ANNs to automatically learn through experience of examples, rather than following prescriptive rules, is one of the major advantages over conventional expert systems [30]. Traditionally, ANNs are trained using methods such as *supervised* and *reinforcement* learning. Supervised learning techniques, such

as *backpropagation* [2], make use of labelled data sets (i.e. correct input-output pairs) to incrementally revise connection weights in the network so as to match the mapping between inputs and expected outputs. Reinforcement learning techniques, on the other hand, do not use labelled training data and are commonly implemented in situations where such data is difficult or impossible to compile. In the standard model, an agent (or robot) is placed in an environment where it can both sense its surroundings and control the actions of its body. The agent is then incentivised, through a system of reward and punishment, to favour behaviour that achieves specific goal-oriented behaviours [32].

An alternative approach, known as *unsupervised learning*, has proven successful in training ANNs. Unsupervised learning techniques provide none of the exemplar solutions or reward and punishment feedback used in supervised and reinforcement learning, respectively [9]. Instead, simply the correlations between input data are considered. Neuroevolution algorithms (see section 3.2) typically fall under this category.



**Figure 2: A generic neural network architecture. It consists of input units and output units which are connected to the external environment and hidden units which connect to other neurons but are not directly connected to the environment [27].**

### 3.2 Evolutionary Algorithms

*Evolutionary algorithms* (EAs) represent a class of probabilistic optimisation algorithms which are based on the model of natural evolution [6]. These entail a collective learning process whereby individuals in a population, representing potential solutions to a problem, undergo repetitive cycles of *selection*, *mutation* and *recombination* converging on an optimal solution [6]. When EAs are applied to training ANNs (especially for complex control tasks), the process is referred to as neuroevolution.

**3.2.1 Classification of EAs.** There are four main streams of EAs which are differentiated based on the data structures used in the encoding of candidate solutions: *genetic algorithms* (GAs), *evolution strategies* (ESs), *evolutionary programming* (EP) and *genetic programming* (GP) [6, 25]. GAs use strings over a finite alphabet (traditionally binary [63]) to represent genotypes, whereas ESs use real-valued vectors, EP uses finite state machines and GP uses tree structures [25]. One of the benefits to using GAs (with a binary alphabet) in neuroevolution is the inherent structural simplicity and the ease of applying *crossover* and *mutation* operators to binary strings [63]. However, it has been proposed that directly using real numbers to represent connection weights may be a better strategy [63]. Using ESs or EP to evolve real-valued vectors can counteract the negative impact of the *permutation problem* (i.e. having the same number of values as variables [51]), resulting in a more efficient execution of the evolutionary process [63].

**3.2.2 Neuroevolution of Augmenting Topologies (NEAT).** NEAT is a popular state-of-the-art neuroevolution strategy that uses a direct, genetic encoding scheme of ANNs to evolve both network topology and connection weights [57]. This genetic encoding is a linear representation of the network, with a set of *connection genes* which each indicate a connection between two *node genes*. Node genes consist of a set of inputs, hidden nodes and outputs. During mating, this format of encoding enables the easy lining-up and crossing-over of corresponding genes.

The original idea behind NEAT is based on three main concepts: *historical markings*, *speciation* and *incremental growth* [57]. Historical markings are used to track the origin of each gene in the network over the course of evolution. To achieve this, upon the introduction of a novel gene (through mutation), a unique *innovation number* is assigned to that gene. This information allows the algorithm to correctly identify and line up matching genes (i.e. those with the same innovation numbers) in separate *genomes* (i.e. the genotypes of different individuals) when crossing over, without the need for computationally expensive structural comparison through graph traversals.

*Speciation* is the process whereby the population is divided into different niches for each topological innovation. This initially protects new topologies from being competed out of consideration and gives them a chance to develop and optimise their structure through competition with others in only their niche [57]. Once again, innovation numbers can be used here to improve computational efficiency by avoiding topology matching.

Finally, the concept of *incremental growth* from minimal structure is contrary to conventional approaches which seed the starting population with many random topologies to encourage diversity. NEAT, in comparison, begins with a uniform set of simple topologies consisting only of input and output nodes (i.e. no hidden layers). Therefore, each candidate ANN starts with a minimal topology which increases in complexity through evolution. This approach minimises the dimensionality of the search space and gives NEAT a performance edge over other techniques. Furthermore, through this process of *complexification*,

NEAT is able to discover far more complex candidate ANNs than fixed-topology techniques [53, 56].

**3.2.3 Hypercube-based NEAT (HyperNEAT).** HyperNEAT is an extension of NEAT which, in contrast, employs an indirect encoding scheme called *connective compositional pattern producing networks* (connective CPPNs) [55]. The chief benefit of this approach over NEAT is that it is able to take advantage of the task's geometry during ANN evolution by mapping its regularities onto the network's topology. In this way, HyperNEAT can take geometric properties, such as the arrangement of sensors on a robot's body, into account while evolving a robot's controller; an ability that traditional NEAT lacks.

CPPNs are used to indirectly encode ANNs through a composition of functions which represent a high-level description of the structural relationships resulting from a process of development [53]. The data structure of CPPNs is highly analogous to that of ANNs, and so, with little customisation, algorithms used in evolving ANNs can be easily applied to evolving CPPNs. Since HyperNEAT works with CPPNs rather than directly with ANNs, it is an extension of the more general framework of CPPN-NEAT [53]. In a 2014 study by Cheney *et al.* [16], where robots composed of *soft* (as opposed to rigid) elements were selected for locomotion speed, it was found that using CPPNs evolved faster robots with more natural morphologies than when a direct encoding was used.

### 3.3 Single-objective Neuroevolution

In conventional *single-objective* neuroevolution, the optimisation of only one parameter is considered in the evolutionary process. For example, when evolving a robot controller, one might simply specify task accomplishment as the objective for optimisation. No other restrictions are placed on the evolutionary process and *any* developments which positively, or neutrally, impact achieving this objective are selected and propagated. As a result, the final evolved controller may incorporate unexpected or undesirable traits unrelated to the objective of focus. Although this may initially seem like an unequivocal disadvantage, the flexibility of this approach does allow for the discovery of creative solutions to challenging problems.

Most early work in the field of ER incorporated this approach to neuroevolution. NEAT, as described in section 3.2.2, is one such single-objective technique. Stanley *et al.* [54] used NEAT, in 2005, to evolve an warning system for impending car crashes. In this study, the single objective being optimised was the accuracy of accident prediction. Results showed that NEAT was able to successfully evolve a reliably accurate predictor for how far in the future a crash would occur. This demonstrates a useful, real-world application of single-objective neuroevolution.

However, there are other research problems which cannot be successfully approached using the inherent limitations of single-objective neuroevolution. For such problems, known as *multi-objective optimisation problems* (MOPs), a *multi-objective* approach has been developed [17].

### 3.4 Multi-objective Neuroevolution

Building on work in traditional neuroevolution, *multi-objective* neuroevolution offers a significantly more refined approach to evolving ANNs. Here, it is possible to specify multiple, often contradictory, parameters to be simultaneously optimised. There are many real-world problems of this nature. For example, consider the case of designing a consumer electronic device, where one might wish to minimise cost and power consumption while maximising performance [65]. A similar ER-related MOP is in the evolution of high-performing, low-complexity neural controllers for robots (see section 4). Many *multi-objective evolutionary algorithms* (MOEAs) have been developed over the years [17], of which two are reviewed below. Thereafter, we analyse two strategies which apply MOEAs to the process of neuroevolution.

**3.4.1 Non-dominated Genetic Sorting Algorithm II (NGSA-II).** NGSA was introduced by Srinivas and Deb [52] in 1994 as one of the first attempts at developing a MOEA. An updated NGSA-II [18] was later introduced in 2002 to improve upon the original algorithm. In NGSA, individual solutions undergo multiple rounds of classification. This process begins by classifying all non-dominated solutions into a separate category and assigning dummy fitness scores (proportional to population size) which give them all equivalent reproductive potential. To maintain diversity, this group of solutions is shared with the assigned fitness values. Then this group of solutions is ignored and another layer of non-dominated solutions is classified. This process repeats until, eventually, all solutions have been classified. Since the individuals in the earlier fronts have higher fitness, they are copied more frequently than the individuals in later fronts.

NGSA-II addressed three of the major criticisms of NGSA, namely (1) the high complexity of non-dominated sorting, (2) the lack of elitism and (3) the need for specifying the sharing parameter [18]. NGSA-II is far more computationally efficient than its predecessor, which earned the approach some renewed interest after being announced [17].

**3.4.2 Strength Pareto Evolutionary Algorithm II (SPEA-II).** Zitzler, Laumanns and Thiele [64] introduced SPEA-II in 2001 as an improvement to the original SPEA from 1999 [65]. SPEA is an approach which combines multiple MOEAs in one. There are two core components in SPEA: a primary population and a secondary population, called the *archive*. The archive represents a continuously updated set of *non-dominated* solutions from the primary population. Initially, the archive is empty and the primary population consists of a set of candidate solutions. At the end of each generation, the non-dominated solutions in the population are copied into the archive. If the archive surpasses a certain capacity threshold over the course of evolution, it is pruned by applying a clustering technique. A fitness score is then assigned to both the solutions in the archive and the population. For solutions in the archive, this score (known as the *strength value*) represents the number of solutions in the population which it dominates. For solutions in the population, it is calculated by summing the strength values of all the solutions in the archive which dominate

it. Thereafter, the mating phase is executed wherein the solutions are selected from the union of the population and the archive. The selected solutions then undergo mutation and crossover events before replacing the previous population. This process is iteratively repeated.

SPEA-II provides three main updates which improve on the algorithm used in SPEA: (1) a fine-grained fitness assignment strategy, (2) a density estimation technique and (3) an enhanced archive truncation method [64]. In implementing (1), fitness assignment was enhanced to factor in, for each candidate solution, both the number of solutions it dominates and the number of solutions which dominate it. Search guidance was improved with (2), where a nearest-neighbour density estimation technique was used. Finally, preservation of boundary solutions was guaranteed with (3).

**3.4.3 NEAT Pareto Strength (NEAT-PS).** Willigen, Haasdijk and Kester [59] introduced a custom multi-objective extension of NEAT, named NEAT-PS, in their 2013 work involving the evolution of controllers for autonomous vehicles. In this study, the multiple objectives considered by the algorithm consisted of various user preferences such as speed, comfort or fuel economy.

NEAT-PS incorporates the *Pareto Strength* (PS) approach from SPEA-II into the NEAT neuroevolution strategy. The benefit of using the PS approach is that it calculates just one fitness score for each candidate solution based on the varying fitness scores for each objective being optimised. This resulting, single fitness score can then be used in the standard NEAT approach with no changes. However, one of the disadvantages of scalarising the multi-objective fitness vector in this manner is that a monotonic evolution of all objectives cannot be guaranteed [1].

**3.4.4 NEAT Multi-Objective Diversified Species (NEAT-MODS).** In a recent 2016 paper, Abramovich and Moshaiyo [1] proposed a multi-objective variation of NEAT, which they termed NEAT-MODS. This implementation aims to address some criticisms of previously described multi-objective neuroevolution strategies, such as NEAT-PS. Specifically, a diversified population of species is maintained without the need for any supplementary objective.

Overall, NEAT-MODS follows the same fundamental steps as in NEAT but with some substantial changes to the selection process. Selection is divided into two stages: (1) selection of species and (2) selection of individuals. In the first stage, the union of the populations of parents and offspring is sorted in a similar manner to that employed by NSGA-II. A subset of species is then selected from this sorted list in a way that supports both the principle of elitism and the exploration of diversity. In the second stage, the best individuals are selected from the previously selected species. Here, the individuals of each species are once again sorted and selected by serial progression of the species. In a comparative simulation, NEAT-MODS was found to perform significantly better than NEAT-PS [1].

## 4 SYSTEM COMPLEXITY

The development of low-complexity robots is becoming a topic of increasing interest as ER is being applied to more and more complex task environments. There are notable performance benefits to having simple robot morphologies and controllers, which includes reduced computation time [15]. Besides the applications in robotics, the study of the origin of complexity in biological systems has long been an area of open inquiry [3]. This section attempts to define the concepts of morphological and neural complexity in relation to ER, and reviews previous research into the multi-objective neuroevolution of reduced complexity in robotics.

### 4.1 Morphological Complexity

*Morphological complexity* is related to the layout of the physical body of a robotic system, including body shape and the number and arrangement of sensors on the body.

In 2012, Auerbach and Bongard [5] investigated the relationship between a robot's morphological complexity and the complexity of the task environment in which it is evolved. Supporting the hypothesis, it was observed that increasing environmental complexity directly promoted the evolution of more complex robot morphologies. However, contrary to these results, further research in the same year on the relationship between environmental and mechanical complexity found a decrease in more complex task environments [4].

Later research by Auerbach and Bongard [3], in 2014, analysed the evolution of complexity in robots selected for locomotion ability. Firstly, it was seen that these robots tended to increase in complexity over evolutionary time. This supported the "arrow of complexity" hypothesis previously posited [10]. Secondly, as with the 2012 study, results indicated that more complex morphologies evolve in complex environments compared with simple environments.

### 4.2 Neural Complexity

*Neural complexity* is related to the structural nature of the neural controllers, or ANNs, involved in the coordination of a robotic system. Such neural controllers are responsible for defining the behavioural actions of a robot in response to sensory input from the environment. In this context, higher neural complexity is associated with ANNs which have a higher number of node-to-node connections and hidden layers.

As previously described, most neuroevolution strategies (namely NEAT) are designed to gradually increase ANN complexity over multiple generations. This approach is an attempt to successfully model complex patterns of behaviour. However, as mentioned in the introduction to this section, increased ANN complexity can also have undesirable consequences, such as increased computation time [15].

### 4.3 Multi-objective Evolution of Complexity

An ideal robot system would be one that is both minimally complex (in its morphology and neural controller) and maximally performant (in its ability to complete complex tasks). This has been a long-anticipated goal in ER which is becoming more relevant as the field expands to more complex problem spaces, such as autonomous vehicles. Recent research has focussed on applying new developments in ER, namely with the introduction of multi-objective neuroevolution, to the evolution of such ideal robot systems.

A 2008 study by Capi and Kaneko [15] investigated the application of multi-objective neuroevolution to the development of low-complexity neural controllers for robots required to perform multiple, simultaneous tasks. Results demonstrated that such an approach can successfully evolve low-complexity neural controllers which are effective in performing tasks. Two alternative encoding schemes, *connectionist encoding* and *node-based encoding*, were compared in this experiment. It was observed that node-based encoding resulted in evolved controllers with higher task performance and more robust neural performance.

Research by Furman, Nagar and Nitschke [40], in 2019, applied the strategies of single- and multi-objective neuroevolution of robotic systems (using variations of NEAT and NEAT-MODS) to the study of the cost of complexity in evolutionary theory. In this study, robots were evolved for a collective gathering task where the objectives of high task performance and low morphological complexity were optimised. This was in an attempt to simulate the hypothesised selective pressures in natural evolution, where larger (and thus more complex) brains carry an increased energy consumption requirement. Two independent simulations were run in environments of varying task difficulty, one with the low-complexity optimisation objective and one without. The experiment produced interesting results. In simpler task environments, both simulations produced robots equally effective in task performance. However, seemingly counterintuitively, in more difficult task environments, robots evolved with a cost on morphological complexity achieved higher task performance (with simpler morphologies) than those evolved without such a cost. Thus, the key insight gleaned from this study is that neuroevolution with a complexity cost can promote the evolution of simpler morphologies with higher performing task-accomplishing behaviours.

## 5 CONCLUSIONS

Overall, it is evident from the literature that the field of collective robotics has made great strides in progress since its advent in the early 1990s, especially in the area of evolutionary robotics. There has been a notable shift in focus from single-objective neuroevolution strategies for artificial neural networks to the development of multi-objective techniques for solving multi-objective optimisation problems. This has seen interesting applications in the evolution of robotic morphologies and neural

controllers. Furthermore, the combinations of evolutionary robotics techniques and advancements in neuroevolution have been applied to the study of evolutionary theory. Recently, interest has increased in the research of system complexity in the evolution of both robotic and natural agents.

Multi-objective optimisation, with algorithms such as NEAT-PS and NEAT-MODS, presents a promising new approach to the research of the cost of complexity (both morphological and neural) in evolution. As a result, we have witnessed multi-objective neuroevolution being successfully applied to the evolution of high-performing, low-complexity robot systems. There are, however, some weaknesses of current multi-objective approaches in evolutionary robotic systems. The reduction of a multi-objective fitness vector (i.e. the multiple parameters being optimised) into a scalar value, as in the NEAT-PS algorithm, cannot ensure monotonic evolution of all objectives.

In conclusion, there are some gaps that have been noted in the current state of the art. Firstly, it is apparent that there is a lack of consensus in the literature on standard definitions for morphological and neural complexity. Secondly, there appears to have been very little work in the co-evolution of morphology and neural controllers in the field of collective robotics [39]. Finally, there are limitations in current approaches to multi-objective neuroevolution.

Thus, possible directions for future research could include a review and standardisation of definitions for morphological and neural complexity. Also, further study is needed in the area of co-evolution of morphological and neural complexity in collective robotics. This research should consider additional algorithmic development to address the current limitations of multi-objective neuroevolution.

## REFERENCES

- [1] Omer Abramovich and Amiram Moshaiov. 2016. Multi-objective topology and weight evolution of neuro-controllers. In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC '16)*. IEEE, 670-677. DOI: 10.1109/CEC.2016.7743857
- [2] Shun-ichi Amari. 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing* 5, 4 (Jun. 1993), 185-196. DOI: 10.1016/0925-2312(93)90006-O
- [3] Joshua E. Auerbach and Josh C. Bongard. 2014. Environmental influence on the evolution of morphological complexity in machines. *PLOS Computational Biology* 10, 1 (Jan. 2014), e1003399. DOI: 10.1371/journal.pcbi.1003399
- [4] Joshua E. Auerbach and Josh C. Bongard. 2012. On the relationship between environmental and mechanical complexity in evolved robots. In *Proceedings of the 2012 Conference on Artificial Life (ALIFE '12)*. MIT Press, Cambridge, MA, 309-316. DOI: 10.7551/978-0-262-31050-5-ch041
- [5] Joshua E. Auerbach and Josh C. Bongard. 2012. On the relationship between environmental and morphological complexity in evolved robots. In *Proceedings of the 14th. annual conference on Genetic and evolutionary computation (GECCO '12)*. ACM, New York, NY, 521-528. DOI: 10.1145/2330163.2330238
- [6] Thomas Bäck and Hans-Paul Schwefel. 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation* 1, 1 (Mar. 1993), 1-23. DOI: 10.1162/evco.1993.1.1.1
- [7] Gianluca Baldassarre, Stefano Nolfi and Domenico Parisi. 2003. Evolving mobile robots able to display collective behaviors. *Artificial Life* 9, 3 (Jul. 2003), 255-267. DOI: 10.1162/106454603322392460
- [8] Wolfgang Banzhaf, Peter Nordin and Markus Olmer. 1997. Generating adaptive behavior using function regression within genetic programming and a real robot. In *Proceedings of the 2nd. International Conference on Genetic Programming*. 35-43.
- [9] Horace B. Barlow. 1989. Unsupervised learning. *Neural Computation* 1, 3 (Sep. 1989), 295-311. DOI: 10.1162/neco.1989.1.3.295



- [10] Mark A. Bedau. 1998. Four puzzles about life. *Artificial Life* 4, 2 (Apr. 1998), 125-140. DOI: 10.1162/106454698568486
- [11] Nicolas Bredeche, Evert Haasdijk and Abraham Prieto. 2018. Embodied evolution in collective robotics: a review. *Frontiers in Robotics and AI* 5, Article 12 (Feb. 2018), 15 pages. DOI: 10.3389/frobt.2018.00012
- [12] Nicolas Bredeche, Jean-Marc Montanier, Berend Weel and Evert Haasdijk. 2013. Roborobo! a fast robot simulator for swarm and collective robotics. *arXiv e-prints* (Apr. 2013), arXiv:1304.2888.
- [13] Rodney A. Brooks. 1986. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation* 2, 1 (Mar. 1986), 14-23. DOI: 10.1109/JRA.1986.1087032
- [14] Gunnar Buason, Nicklas Bergfeldt and Tom Ziemke. 2005. Brains, bodies, and beyond: competitive co-evolution of robot controllers, morphologies and environments. *Genetic Programming and Evolvable Machines* 6, 1 (Mar. 2005), 25-51. DOI: 10.1007/s10710-005-7618-x
- [15] Genci Capi and Shin-ichiro Kaneko. 2008. Evolution of low-complexity neural controllers based on multiobjective evolution. *Artificial Life and Robotics* 12, 1 (Mar. 2008), 53-58. DOI: 10.1007/s10015-007-0441-0
- [16] Nick Cheney, Robert MacCurdy, Jeff Clune and Hod Lipson. 2014. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution* 7, 1 (Aug. 2014), 11-23. DOI: 10.1145/2661735.2661737
- [17] Carlos A.C. Coello. 2006. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine* 1, 1 (Feb. 2006), 28-36. DOI: 10.1109/MCI.2006.1597059
- [18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (Aug. 2002), 182-197. DOI: 10.1109/4235.996017
- [19] Kevin DeMarco, Eric Squires, Michael Day and Charles Pippin. 2019. Simulating collaborative robots in a massive multi-agent game environment (SCRMIMAGE). In *The 9th. Springer Proceedings in Advanced Robotics (SPAR '19)* Springer, Cham, Switzerland, 283-297. DOI: 10.1007/978-3-030-05816-6\_20
- [20] Jean-Louis Deneubourg, Simon Goss, Nigel Franks, Ana Sendova-Franks, Claire Detrain and Laetitia Chrétien. 1991. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the 1st. International Conference on Simulation of Adaptive Behavior*. MIT Press, 356-363.
- [21] Cristian Dima, Martial Hebert and Anthony Stentz. 2004. Enabling learning from large datasets: applying active learning to mobile robotics. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA '04)*. IEEE, 108-114. DOI: 10.1109/ROBOT.2004.1307137
- [22] Stéphane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret and Agoston E. Eiben. 2015. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI* 2, Article 4 (Mar. 2015), 18 pages. DOI: 10.3389/frobt.2015.00004
- [23] Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho M. Oliveira and Anders L. Christensen. 2016. Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLOS ONE* 11, 3 (Mar. 2016), e0151834. DOI: 10.1371/journal.pone.0151834
- [24] Robin I.M. Dunbar. 1998. The social brain hypothesis. *Evolutionary Anthropology: Issues, News, and Reviews* 6, 5 (Dec. 1998), 178-190. DOI: 10.1002/(SICI)1520-6505(1998)6:5<178::AID-EVAN5>3.0.CO;2-8
- [25] Agoston E. Eiben and James E. Smith. 2003. *Introduction to Evolutionary Computing* (2nd. ed.). Springer-Verlag, Berlin, Germany.
- [26] Eliseo Ferrante, Ali Emre Turgut, Edgar Duéñez-Guzmán, Marco Dorigo and Tom Wenseleers. 2015. Evolution of self-organized task specialization in robot swarms. *PLOS Computational Biology* 11, 8 (Aug. 2015), e1004273. DOI: 10.1371/journal.pcbi.1004273
- [27] Dario Floreano, Peter Dürri and Claudio Mattiussi. 2008. Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1, 1 (Mar. 2008), 47-62. DOI: 10.1007/s12065-007-0002-4
- [28] Pablo Funes and Jordan Pollack. 1998. Evolutionary body building: adaptive physical designs for robots. *Artificial Life* 4, 4 (Oct. 1998), 337-357. DOI: 10.1162/106454698568639
- [29] Gregory S. Hornby, Seiichi Takamura, Osamu Hanagata, Masahiro Fujita and Jordan B. Pollack. 2000. Evolution of controllers from a high-level simulator to a high DOF robot. In *Proceedings of the 3rd. International Conference on Evolvable Systems (ICES '00)*. Springer Berlin Heidelberg, 80-89. DOI: 10.1007/3-540-46406-9\_9
- [30] Anil K. Jain, Jianchang Mao and K.M. Mohiuddin. 1996. Artificial neural networks: a tutorial. *Computer* 29, 3 (Mar. 1996), 31-44. DOI: 10.1109/2.485891
- [31] Fredrik Jansson, Matthew Hartley, Martin Hinsch, Ivica Slavkov, Noemí Carranza, Tjelvar S.G. Olsson, Roland M. Dries, Johanna H. Grönqvist, Athanasius F.M. Marée and James Sharpe. 2015. Kilombo: a kilobot simulator to enable effective research in swarm robotics. *arXiv e-prints* (Nov. 2015), arXiv:1511.04285.
- [32] Leslie P. Kaelbling, Michael L. Littman and Andrew W. Moore. 1996. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research* 4 (May 1996), 237-285. DOI: 10.1613/jair.301
- [33] Yohannes Kassahun, Mark Edgington, Jan H. Metzen, Gerald Sommer and Frank Kirchner. 2007. A common genetic encoding for both direct and indirect encodings of networks. In *Proceedings of the 9th. annual conference on Genetic and evolutionary computation (GECCO '07)*. ACM, New York, NY, 1029-1036. DOI: 10.1145/1276958.1277162
- [34] Claus R. Kube and Hong Zhang. 1992. Collective robotic intelligence. In *Proceedings of the 2nd. International Conference on Simulation of Adaptive Behavior*. MIT Press, Cambridge, MA, 460-468.
- [35] Claus R. Kube and Hong Zhang. 1993. Collective robotics: from social insects to robots. *Adaptive Behavior* 2, 2 (Sep. 1993), 189-218. DOI: 10.1177/105971239300200204
- [36] Hod Lipson and Jordan B. Pollack. 2000. Automatic design and manufacture of robotic lifeforms. *Nature* 406, 6799 (Aug. 2000), 974-978. DOI: 10.1038/35023115
- [37] Alcherio Martinoli and Francesco Mondada. 1997. Collective and cooperative group behaviours: biologically inspired experiments in robotics. In *Proceedings of the 4th. International Symposium on Experimental Robotics*. Springer Berlin Heidelberg, 1-10. DOI: 10.1007/BFb0035192
- [38] Warren S. McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5, 4 (Dec. 1943), 115-133. DOI: 10.1007/bf02478259
- [39] Jessica Meyer and Joachim Hertzberg. 2018. Co-evolution of morphology and behavior in self-organized robotic swarms. In *Proceedings of the 5th. BSC Severo Ochoa Doctoral Symposium*. Barcelona Supercomputing Center, 74-75.
- [40] Danielle Nagar, Alexander Furman and Geoff Nitschke. 2019. The cost of complexity in robot bodies. In *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC '19)*. IEEE.
- [41] Hiroshi Nakamura, Akio Ishiguro and Y. Uchikawa. 2000. Evolutionary construction of behavior arbitration mechanisms based on dynamically-rearranging neural networks. In *Proceedings of the 2000 IEEE Congress on Evolutionary Computation (CEC '00)*. IEEE, 158-165. DOI: 10.1109/CEC.2000.870290
- [42] Ana Navarrete, Carel P. van Schaik and Karin Isler. 2011. Energetics and the evolution of human brain size. *Nature* 480 (Nov. 2011), 91-93. DOI: 10.1038/nature10629
- [43] Andrew L. Nelson, Gregory J. Barlow and Lefteris Doitsidis. 2009. Fitness functions in evolutionary robotics: a survey and analysis. *Robotics and Autonomous Systems* 57, 4 (Apr. 2009), 345-370. DOI: 10.1016/j.robot.2008.09.009
- [44] Geoff S. Nitschke, Martijn C. Schut and Ágoston E. Eiben. 2012. Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm and Evolutionary Computation* 2 (Feb. 2012), 25-38. DOI: 10.1016/j.swevo.2011.08.002
- [45] Stefano Nolfi and Dario Floreano. 2002. Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences* 6, 1 (Jan. 2002), 31-37. DOI: 10.1016/S1364-6613(00)01812-X
- [46] Carlo Pinciroli, Vito Trianni, Rehan O' Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Timothy Stirling, Álvaro Gutiérrez, Luca M. Gambardella and Marco Dorigo. 2011. ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5027-5034. DOI: 10.1109/IROS.2011.6094829
- [47] Matt Quinn, Lincoln Smith, Giles Mayley and Phil Husbands. 2002. Evolving team behaviour for real robots. In *Proceedings of the 2002 EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics*. 14-16.
- [48] Francis L.W. Ratnieks and Carl Anderson. 1999. Task partitioning in insect societies. *Insectes Sociaux* 46, 2 (May 1999), 95-108. DOI: 10.1007/s000400050119
- [49] Michael Rubenstein, Christian Ahler and Radhika Nagpal. 2012. Kilobot: a low cost scalable robot system for collective behaviors. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA '12)*. IEEE, 3293-3298. DOI: 10.1109/ICRA.2012.6224638
- [50] Erol Şahin. 2005. Swarm robotics: from sources of inspiration to domains of application. In *Proceedings of the 1st. International Workshop on Swarm Robotics*. Springer Berlin Heidelberg, 10-20. DOI: 10.1007/978-3-540-30552-1\_2
- [51] Barbara M. Smith. 2000. Modelling a permutation problem (2000), 615-619.
- [52] Nidamarthi Srinivas and Kalyanmoy Deb. 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2, 3 (Sep. 1994), 221-248. DOI: 10.1162/evco.1994.2.3.221
- [53] Kenneth O. Stanley. 2007. Compositional pattern producing networks: a novel abstraction of development. *Genetic Programming and Evolvable Machines* 8, 2 (Jun. 2007), 131-162. DOI: 10.1007/s10710-007-9028-8
- [54] Kenneth O. Stanley, Nate Kohl, Rini Sherony and Risto Miikkulainen. 2005. Neuroevolution of an automobile crash warning system. In *Proceedings of the*



- 7th. annual conference on Genetic and evolutionary computation (GECCO '05). ACM, New York, NY, 1977-1984. DOI: 10.1145/1068009.1068340
- [55] Kenneth O. Stanley, David B. D'Ambrosio and Jason Gauci. 2009. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life* 15, 2 (Apr. 2009), 185-212. DOI: 10.1162/artl.2009.15.2.15202
  - [56] Kenneth O. Stanley and Risto Miikkulainen. 2004. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research* 21 (Feb. 2004), 63-100. DOI: 10.1613/jair.1338
  - [57] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10, 2 (Jun. 2002), 99-127. DOI: 10.1162/106365602320169811
  - [58] Spyros G. Tzafestas. 1995. Neural networks in robot control. In *Artificial Intelligence in Industrial Decision Making, Control and Automation*. Springer Dordrecht, 327-387. DOI: 10.1007/978-94-011-0305-3\_11
  - [59] Willem van Willigen, Evert Haasdijk and Leon Kester. 2013. A multi-objective approach to evolving platooning strategies in intelligent transportation systems. In *Proceedings of the 15th. annual conference on Genetic and evolutionary computation (GECCO '13)*. ACM, New York, NY, 1397-1404. DOI: 10.1145/2463372.2463534
  - [60] Barbara Webb. 2000. What does robotics offer animal behaviour? *Animal Behaviour* 60, 5 (Nov. 2000), 545-558. DOI: 10.1006/anbe.2000.1514
  - [61] Lee Wei-Po, Johnn Hallam and Henrik H. Lund. 1997. Applying genetic programming to evolve behavior primitives and arbitrators for mobile robots. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*. IEEE, 501-506. DOI: 10.1109/ICEC.1997.592362
  - [62] Sidney G. Winter. 1989. Natural Selection and Evolution. In *Allocation, Information and Markets*. Palgrave Macmillan, London, 214-222. DOI: 10.1007/978-1-349-20215-7\_22
  - [63] Xin Yao. 1999. Evolving artificial neural networks. *Proceedings of the IEEE* 87, 9 (Sep. 1999), 1423-1447. DOI: 10.1109/5.784219
  - [64] Eckart Zitzler, Marco Laumanns and Lothar Thiele. 2001. SPEA2: improving the strength pareto evolutionary algorithm. *TIK-Report* 103 (May 2001). DOI: 10.3929/ethz-a-004284029
  - [65] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (Nov. 1999), 257-271. DOI: 10.1109/4235.797969