Gene regulatory network to control and simulate virtual creature's locomotion

Ahmed Tibermacine, Noureddine Djedi
LESIA laboratory / Department of Computer Science
Biskra University
PB 145 RP, 07000 Biskra, Algeria.
{ahmed.tibermacine, n.djedi}@univ-biskra.dz

Abstract: - This paper explores the application of evolutionary computation techniques for evolving behaviours of virtual creatures inhabiting a realistic virtual environment. Our approach uses a computational model of gene regulatory network, which is inspired of cell control mechanism of real cells. Usually used to control virtual cells in developmental models, recent works showed that gene regulatory networks are also capable to control various kinds of agents. This paper details how a gene regulatory network is evolved to control range of articulated virtual creatures. To do so, the inputs and outputs of the network are directly mapped to the creatures' sensors and actuators. To evaluate this approach, we have compared its performance to one of the most recent successful evolutionary methods, the NEAT algorithm. The results show the gene regulatory network model may possibly be a viable solution for evolving control solutions for physical machines.

Key-Words: - Gene regulatory network, virtual creature, behaviour control, physical simulation, Evolutionary computation

1. Introduction

Gene regulatory networks are usually used to control cell behaviours in developmental models through the production of internal and external proteins. Over the past few years, regulatory networks have been proved efficient as a central system for cell based developmental models such as established in [1-6] Recently, many modern computational models of these networks have been used in a wide range of problem and have proven their capacity to simulate biologically plausible gene regulatory networks [7-9] and to control agents [10-12]. More specifically, we think that Gene regulatory networks can successfully learn how to generate behaviours in 3D virtual creatures acting in a physically realistic environment to fulfil a specific task, forming a large space of behaviors and more complex control task than in domains where GRN was previously tested. In the literature, evolution of autonomous creatures was pioneered by Karl Sims in 1994 [13, 14]. Sims was the first to introduce the idea of using evolutionary methods to automatically generate 3D virtual creatures. His creatures inhabit a three-dimensional world with simulated physical laws and are controlled by a series of neural networks distributed along the body of the creatures. Since these publications, several researchers have re-implemented Sims' work in controlling creatures adopting different evolutionary approaches [15-29].

In this work, we investigate the possibility of using Genetic regulatory networks to evolve three-dimensional self-controlling virtual creatures; as a step toward devising an evolutionary approach for the emergence of locomotive behaviours that utilizes technique inspired by biological evolutionary systems. In this paper, the subjects of our experiments are virtual articulated creature placed in a virtual 3D environment with simulated physics using rigid-body dynamics. Controller of the creature is subject to optimization by evolution in four independent experiments: crawling, arm-based moving, somersaulting and running. We demonstrate whether the GRN improve the performance of evolutionary search by comparing it to the one of the most recent successful algorithm for the evolution of neural networks: NeuroEvolution of Augmenting Topologies (NEAT [30]), a well-known method for optimizing both structure and weights of a neural network. NEAT outperforms the best fixed-topology neuroevolution method on a challenging benchmark reinforcement learning task [30]. NEAT algorithm has been shown to be effective in many applications such as pole balancing [31], robot control [32], vehicle control [33], and character control in a video game [34-36], etc. To our knowledge, this work is the first attempt to use a gene regulatory network to generate locomotion behaviours for virtual creatures inhabiting within a physical 3D environment.

The rest of the paper is organized as follows. Section 2 introduces the computation model we have used in this work. Section 3 describes how the GRN is connected to the creature sensors and actuators and how the GRN is trained with a genetic algorithm to produce a basic controller. Following which, section 4 provides background about NEAT. Section 5 then describes the setup of our experiments, fitness evaluation and discussing the details of the creature's morphologies. Results of experiments are provided in section 6. We then bring some discussion by comparing the effectiveness of both methods on experiments using in our work, and present an idea to improve this model. Finally, section 8 concludes with a summary of this work together with some possible future works.

2. Our Gene Regulatory Networks model

The gene regulatory network used to develop locomotive strategies in this paper is a simplified model based on Banzhaf's model [9]. It has already been successfully used in other applications. It is capable of developing modular robot morphologies [1], producing 2-D images [37], controlling cells designed to optimize a wind farm layout [38], controlling reinforcement learning parameters in [39] and driving virtual racing car [40]. This model has been designed for computational purpose only and not to simulate a biological network.

This model is composed of a set of abstract proteins. The protein pr is composed of three tags:

- The protein tag id_{pr} that identifies the protein,
- The enhancer tag enh_{pr} that defines the enhancing matching factor between two proteins.
- The inhibitor tag inh_{pr} that defines the inhibiting matching factor between two proteins.

These tags are coded with an integer in [0, p] where the upper bound p can be tuned to control the precision of the network. In addition to these tags, a protein is also defined by its concentration that will vary over time with a particular dynamics described later. A protein can be of three different types:

- *Input*, a protein whose concentration is provided by the environment, which regulates other proteins but is not regulated,
- *Output*, a protein with a concentration used as output of the network, which is regulated but does not regulate other proteins, and
- *Regulatory*, an internal protein that regulates and is regulated by others proteins.

With this structure, the dynamics of the GRN are computed by using the protein tag. They determine the productivity rate of pairwise interaction between two proteins. For this, the affinity of a protein pr_a for another protein pr_b is given by then enhancing factor $u^+_{pr_a pr_b}$ and the inhibiting factor $u^-_{pr_a pr_b}$ calculated as follows:

$$u^+_{pr_a pr_b} = p - |enh_{pr_a} - id_{pr_b}|;$$

$$u^-_{pr_a pr_b} = p - |inh_{pr_a} - id_{pr_b}| \quad (1)$$

The proteins are then compared pairwise according to their enhancing and inhibiting factors. For a protein pr_a, the total enhancement g_{pr_a} and inhibition h_{pr_a} are given by:

$$g_{pr_a} = \frac{1}{N}\sum_{pr_b}^{N} c_{pr_a} e^{\beta(u^+_{pr_a pr_b} - u^+_{max})} \quad ;$$

$$h_{pr_a} = \frac{1}{N}\sum_{pr_b}^{N} c_{pr_b} e^{\beta(u^-_{pr_a pr_b} - u^-_{max})} \quad (2)$$

Where N is the total number of proteins in the network, c_{pr_a} is the concentration of the protein pr_b, u^+_{max} is the maximum observed enhancing factor, u^-_{max} is the maximum observed inhibiting factor and β is a control parameter which will be detailed hereafter. At each time step, the concentration of a protein pr_a change with the following differential equation:

$$\frac{dc_{pr_a}}{dt} = \frac{\delta(g_{pr_a} - h_{pr_a})}{\emptyset} \quad (3)$$

where $\emptyset$ is a normalization factor to ensure the sum total of the output and regulatory protein concentration is equal to 1. β and δ are two scaling factor that influence the reaction rates of the network. β affects the importance of the matching factors and δ is used to modify the production level of the proteins in the differential equation. In summary, the lower both values are, the smoother the regulation is; the higher the values are, the more sudden the regulation is.

3. Using a GRN for evolving behaviourisms in virtual creatures

3.1. Linking the GRN to the creature sensors and actuators

The creatures have sensors to collect data from the environment and feed the data to the GRN which return values to solve the problem it is applied to. Table 1 presents the variety of sensors employed in this simulation as well as the range of values that the sensors are able to detect.

Table 1 Representation of sensory data

Sensor Type	Output Range	Sensor Type	Output Range
Hinge	[-pi,pi]	Angular Velocity	[-100, 100]
Slider	[-1,1]	Height	[0, 1000]
Ball/Socket	[-pi,pi]	Direction	[0,2pi]
Linear Velocity	[-300,300]	Touch	on / off

To use the gene regulatory network for evolving behaviourisms in virtual creatures, we convert the input and output signals into normalized concentration values and to find a proper way of connecting the creature actuators to the GRN. Before being computed by the GRN, the environmental data collected by the sensors is normalized to [0,1] with the following formula:

$$norm(v_{sensor_i}) = \frac{v_{sensor_i} - min_{sensor_i}}{max_{sensor_i} - min_{sensor_i}} \quad (4)$$

where v_{sensor_i} is the value of sensor (i) to normalize, min_{sensor_i} is the minimum value of the sensor and max_{sensor_i} is the maximum value of the sensor.

Table 2 Representation of effector data

Actuator/ Joint	Max Force/ Torque	Input Range
Hinge	800 g-cm	[-10,10] rad/sec
Slider	1200 g-cm	[-100,100] cm/sec
Ball/socket	600 g-cm	[-10,10] rad/sec

Once the GRN input protein concentrations are updated, the GRN's dynamics are run one time in order to propagate the concentration modification to the whole network. The concentration of the output proteins are then fed directly to the inputs of actuators which convert it into a desired speed by linearly scaling the effectors inputs to the inputs range as shown in table 2. The maximum force/torque is predetermined for each type of actuator, and is designed to be similar to several common types of DC electric motors.

Depending on the type of joint used, the actuators exert a pulling/pushing force or a flexion/extension force in any of the degree of freedom of joint. Therefore, two proteins, O^+ for pushing or flexion force and O^- for pulling or extension force, are necessary to determine the final values of desired motor velocity provided to the creatures' actuators that computed as follows:

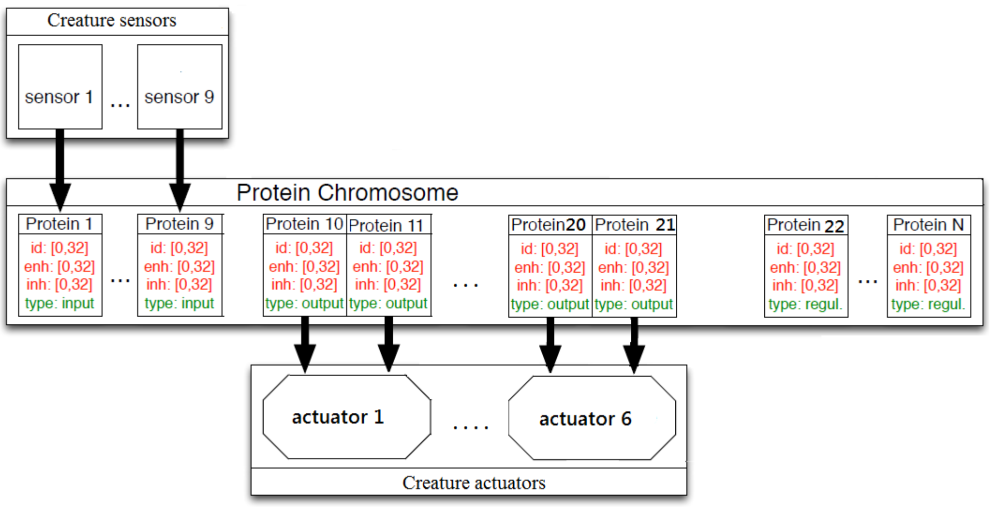


Fig. 1 Organization of the protein chromosome and link to the creature sensors and actuators.

$$velocity = \frac{c(o^{+}) - c(o^{-})}{c(o^{+}) + c(o^{-})} \quad (5)$$

where $c(o^{*})$ is the concentration of the output protein o^{*}. Figure [1] shows the connection of the GRN to the virtual creature.
The use of two output proteins to control one motor's velocity also allows for the GRN to produce an output compatible with the virtual creature's effectors, which also require data to be presented in a bipolar format. The continual cycle of input and output to and from the GRN is the mechanism that can produce intelligent behaviours.

3.2. GRN's genetic encoding

The GRN's genetic encoding scheme is designed to allow corresponding protein genes to be easily lined up when two genomes are crossed over during mating. Each genome includes two independent chromosomes. The first one is defined as a variable length chromosome of indivisible proteins. Each protein is encoded with three integers between 0 and p that correspond to the three tags. In this particular work, p is set at 32 and the genome proteins are organized with the input proteins first, followed by the output proteins and then regulatory proteins. The inputs and outputs presented in the previous section will be always being linked to the same protein, as represented in Figure 1. The second chromosome is encoded as two double precision floating point values that correspond to the two scaling factor presented in the previous section.

Before it can control the creatures, the regulatory network needs to be optimized. In this work, we use a genetic algorithm GA with particular crossover and mutation operators represented in Figure 2 to optimize the first chromosome:
Crossover can only occur between two proteins and never between two tags of the same protein; this ensures the integrity of both sub-networks when the GRN is subdivided into two networks. When assembling another GRN, local connections are kept with this operator and only new connections between the two networks are created.

A mutation in this GRN model can change both tags within a randomly selected protein and network structures. Protein tags are mutated as in standard mutation, with one tag randomly chosen either perturbed or not at each generation. Structural mutation occurs in two ways each mutation expands or shrinks the size of the genome by adding or removing protein(s). In the add protein mutation, a single new regulatory protein with random identifier, enhancing and inhibiting factors values is added to the first chromosome, in the remove protein mutation, an existing regulatory protein is removed from the first chromosome.

4. Neuroevolution of Augmenting Topologies (NEAT)

This type of neuroevolution methods uses direct encoding to describe the network structure and connection weights. NEAT's networks provide three key features: it marks all genes with a unique identifier, allowing tracing where the genes come from, thus carry out effective crossover. The NEAT's network uses the speciation of its individuals and protects innovation. NEAT starts from a population of simple networks with no hidden neurons and increasingly growing its complexity by the gradual addition of neurons and connections over the evolution to match the complexity of the problem which results in significant performance gains.

The structure of a NEAT genome consists of list of neuronal genes which describes the function of the neuron in the network that is an input neuron, an output neuron, a hidden neuron, or a bias, and a list of connections genes which contains information on two connected neurons, the weight associated with this connection, a flag to indicate

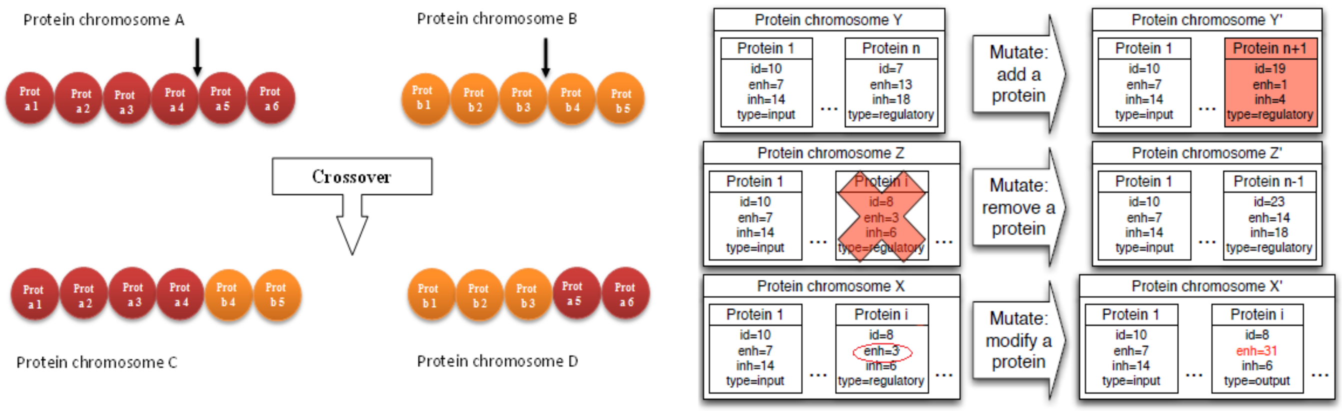


Fig. 2 Crossover and mutation operators applied to the protein chromosome

whether the link is activated, a flag to indicate if the link is recurrent, and an unchangeable innovation number that allows to find the corresponding genes.

In crossover, NEAT's network performs alignment of genes of two different genomes to produce new offspring; genes that have the same innovation number are randomly selected from one of the parents. Genes of the fit parent with innovation numbers not found in the other parent are inherited by the offspring as well. Innovation numbering tackles the competing conventions problem by avoiding the inheritance of the same genes [30].The mutation in Neural networks NEAT can change both connection weights and network topology. The connection weights mutate as in any system of neuroevolution, while structural changes are provided by two special mutation operators. The first consists to add a new connection between two previously unconnected neurons and the second to add new neuron to network. Adding new structures usually initially decreases the fitness of the network, thus newly created structures are less likely to survive long enough to be optimized. This problem is solved by speciation, which divides the population into separate niches of similar genomes. The individuals' fitness of each species must be adjusted before any selection occurs. This ensures that the similar individuals in the population are punished to maintain diversity. Speciation allows new innovations to be optimized without facing competition from individuals with different structures. A full description of the NEAT method is presented in [30].

5. Experiments

In this particular work, the creatures' morphology is completely predetermined. Four different morphologies are introduced into simulation; each one is used in independent experimentation. The results of the four experiments allow us from one hand to evaluate the generalization capabilities of the GRN's process in its ability to evolve efficient locomotive behaviours when it is applied on a variety of morphologies, and from a second hand to prove that the system is able of controlling a variety of joints and structures that exist in the natural world.

5.1. Virtual environment and physics engine

Virtual environments play a crucial role in the evolution of autonomous virtual creatures. The environment must accurately model the physical laws that exist in nature to ensure that the creatures existing in the virtual environment are unable to generate movements or forces that are impossible to produce in the real world. To ensure this realism, the virtual environment employs a physics engine, open dynamic engine (ODE), to accurately simulate certain physical system, such as rigid body dynamics, joint, contacts/collisions, friction, inertia and gravity. The world in which the creatures live within is 3D physically simulated environment whose its ground is flat terrain completely barren with coefficient of friction similar to that of asphalt. This ground upon which the virtual creatures can use their members to generate force.

5.2. Creature morphology

The different morphologies we have used in this work are:

5.2.1. Crawler

The first type of creature introduces a morphology that is composed of two appendages protruding from the front of the main body part. These appendages are attached via 1-DOF hinge joints as shown in Figure 3a. This morphology was designed so that the creature would use its appendages to crawl forward. This morphology was introduced to show that the gene regulatory network is an effective method to train simple creatures to perform tasks that involve manipulating a simple control system.

5.2.2. Arm-based creature

The second creature design consists of two complex "arm-like" structures protruding from the top of its main body, as shown in Figure 3b. Each complex "arm-like" is composed of three rigid bodies interconnected with hinge joints and connected to the main body with a hinge joint. The "hands" are shaped like paddles to provide greater contact area with the ground. This creature was designed so that it would learn how to alternate their "arm" movements to allow each "arm" to pull the torso forward and time their movements. This morphology was introduced to see how well the gene regulatory network copes with a more morphologically complex creature and to determine if it would be able to learn how to effectively manipulate complex articulated members composed of multiple rigid bodies and joints.

5.2.3. Hopper

The third creature is designed to elicit somersault behaviours which represent the most complex movement behaviour of all species. This creature would require at least some knowledge of projectile motion and be able to determine where is will land after each successive jumping motion. *Hopper* creature is composed of two legs that attach to the underside of the main body part via hinge joints. Each "leg" is composed of two rigid bodies interconnected via a slider joint, as shown in Figure 3c. The slider joint gives the legs the ability to quickly extend. This morphology was introduced to determine if the gene regulatory network would be able to cope with a more complex behaviour and learn how to effectively manipulate lot of information from its environment.

5.2.4. Runner

The forth creature morphology is composed of two "arm-like" appendages protruding from either side of the main body, as shown in Figure 3d. The two appendages are connected to the torso via ball and socket joints that provide three degrees of freedom for each "arm". Due to the two additional degrees of freedom, this creature is required to process significantly more information than the *Crawler* creatures utilizing hinge joints with only one degree of freedom. This morphology was introduced to demonstrate the effectiveness of the GRN algorithm at evolving efficient behaviors in virtual creatures employing complex joints.

In each experiment, the performance of GRN was compared to the performance of the NEAT neuroevolution method in order to show whether GRN improves the performance of the evolutionary search. For each experiment, the two algorithms have been tested. The fitness graphs are representative of the average of 20 simulation runs per morphology.

5.3. Fitness Evaluation

In order to assess the performance of GRN and NEAT on the problem of evolving locomotive strategies for a virtual creature, the creatures are evaluated by the total distance travelled in feet, minus a penalty for traveling off course, in the time period of 7.5 seconds.

$$dis(Cr_i) = \varphi(-y - y_0) - (|x - x_0|^{\vartheta}) \quad (6)$$

where x_0 and y_0 represent the initial starting position of the creature on **X** and **Y**, respectively, φ the multiplication factor for the distance travelled

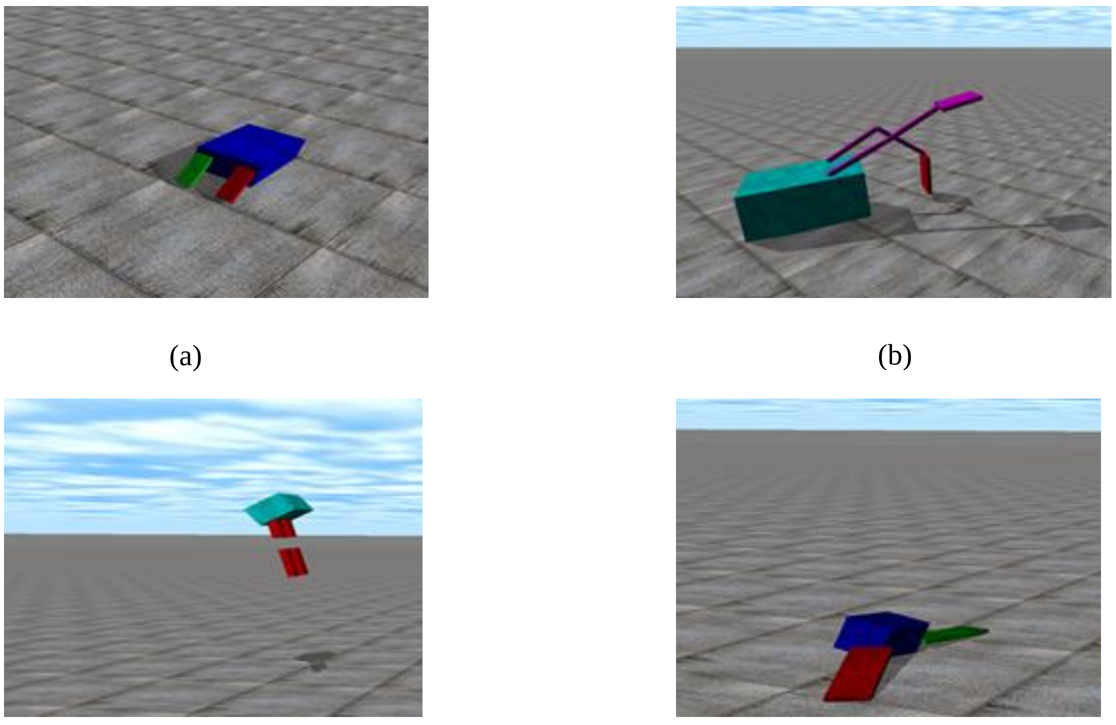


Fig. 3 Virtual creature morphology: Crawler (3a), Arm-based creature (3b), Hopper (3c) and Runner (3d).

and ϑ is the penalty factor. Therefore, the fitness function of each creature is defined as follows:

$$Fitness = \begin{cases} dis(Cr_i) & if \quad dis(Cr_i) > 0 \\ 0 & if \quad dis(Cr_i) = 0 \end{cases} \quad (7)$$

5.4. Parameter settings

The same experimental settings for each method are used in all experiments; they were not tuned specifically for any particular experiment. These values are chosen empirically through a series of test cases. Each configuration was tested in 20 runs. Each run was stopped after 300 generations. Significance levels were computed using Student's t-test. This section presents the experimental setting we have used in this work.

5.4.1. GRN parameter settings

Population size of 80 GRNs was used. In each generation, the probability of mating two genomes was 70% and mutation was 80%. The minimum and maximum number of regulatory proteins in each chromosome is 4 and 20 respectively. Both scaling factor β and δ are evolved in the interval [0.5; 2]. Values under 0.5 produce unreactive networks, whereas values over 2 produce very unstable networks.

5.4.2. NEAT parameter settings

The four experiments used a population of 80 NEAT networks. The multipliers used to tweak the final score of measuring compatibility were C_1= 1.0, C_2= 1.0, and C_3= 0.4. In all experiments, compatibility threshold δ_t= 0.26. If the maximum fitness of a species did not improve in 15 generations, the networks in the stagnant species were not allowed to reproduce. The leader of each species transferred to the new population without mutation; this provides per species elitism. In each generation, the probability of mating two genomes was 80%. In each species, the probability of a new node mutation was 3% and the probability of adding a new link was 7%. There was a 10% chance of a genome having its connection weights mutated, in which case each weight had a 10% chance of being assigned a new random value and a 90% chance of being uniformly perturbed. We used a bipolar sigmoid transfer function:

$$f(x) = \frac{2}{1+e^{-x}} - 1. \quad (8)$$

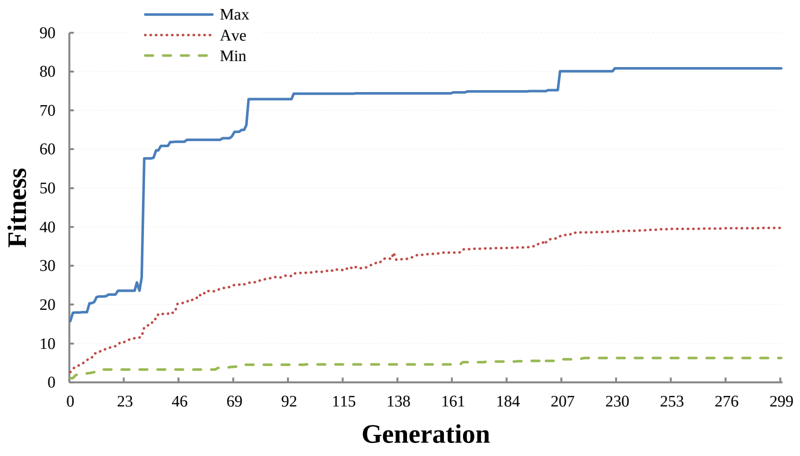
By employing this transfer function, the NEAT network is able to effectively process the bipolar sensory data and generate bipolar output necessary for the correct operation of the virtual creatures' actuators.

6. Results and measuring performance

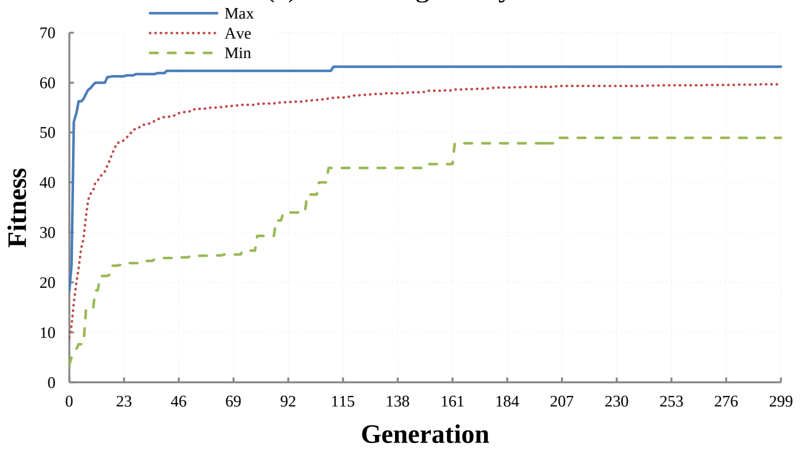
In the crawling experiment, both GRN and NEAT were able to consistently find effective solutions (see Figure 4a and 4b) allowing the *Crawler* creature to learn how to use its two front appendages to pull itself forward. GRN achieved a score of 80.85 within 300 generations of evolution and NEAT achieved a score of 63.20 within 300 generations of evolution. GRN thus outperformed NEAT by 21.78% (student's t-test p-value < 0.005), confirming that training simple virtual creatures to perform tasks that involve manipulating a simple control system is favorable to gene regulatory networks.

With the arm-based creature, both methods failed to simultaneously learn how to alternate their "arm" movements to allow each "arm" to pull the torso forward and time their movement. Controller evolved by GRN and NEAT allow to the arm-based creature to use only one "arm" to pull itself forward by digging their "hand" into the ground and pulling their torso forward. The performance of the NEAT with a score of 202.20 was in this case better than GRN which achieved a score of 147.42 (although the difference was statistically significant; p-value < 0.0001). Illustrations (5a) and (5b) show us the scores achieved by Arm-based creatures produced by GRN and NEAT.

In the somersaulting experiment, both GRN and NEAT were able to consistently find good solutions enabling the *Hopper* to learn how to control their summersaults and performing numerous summersaults in a row. GRN achieved a score of 303.69 within 300 generations of evolution and NEAT achieved a score of 221.97 within 300 generations of evolution– GRN performed 26.9% better than NEAT during the 300 generation. However, differences between GRN and NEAT were not statistically significant (p-value > 0.2) due to a large standard deviation in this particular experience but the GRN outperforms NEAT out of the 20 runs.



(a) Gene Regulatory Network

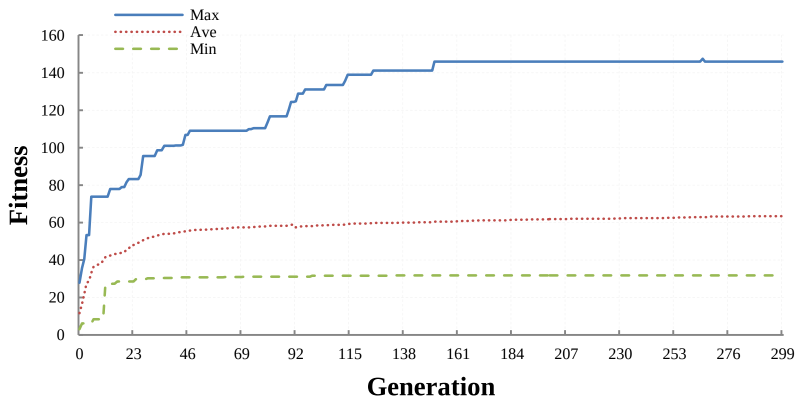


(b) NeuroEvolution of Augmenting Topologies

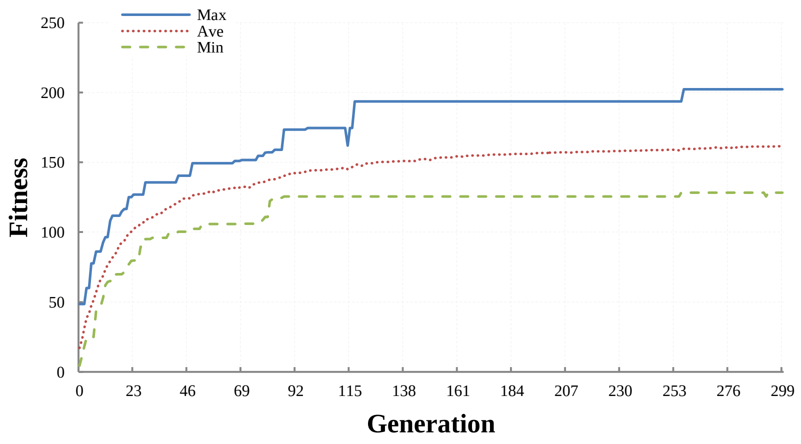
Fig. 4 Fitness graph of the first variety "crawler". The evolution of the population's best, worst and average fitness for the GRN experiment (4a) and for the NEAT experiment (4b) averaged over 20 run.

The results, as evidenced in Figure (6a) and (6b), demonstrate a type of evolution known as punctuated equilibrium. In punctuated equilibrium, the population undergoes long periods of stasis, with short and dramatic increases in fitness between these longer periods of stasis.
Once again, GRN have successfully demonstrated its ability to evolve complex locomotive behaviours. This creature was able successfully learn how to apply force to its actuators in order to precisely time the rotation of its body perfectly to allow the feet to impact upon landing, so that it can immediately perform another summersault sequence and control its trajectory.

In the Running experiment, both GRN and NEAT were able to consistently find effective solutions (see Figure 7a and 7b) enabling the *Runner* creature to learn how to manipulate both arms by simultaneously swinging them forward, placing them on the ground and thrusting them back to produce forward motion. GRN achieved a score of 85.82 within 300 generations of evolution and NEAT achieved a score of 146.59 within 300 generations of evolution– NEAT thus outperformed GRN by 41.45%. Differences between the two methods are extremely statistically significant (p-value < 0.0001). Although, this simulation demonstrates that neural networks NEAT are more efficient than GRN at evolving behaviours in autonomous creatures employing complex joint with up to 3 degrees of freedom. GRN show its ability to generate behaviour in a reasonable amount of time for morphologies involving appendages connected via joint, very much how human arms or legs are attached.



(a) Gene Regulatory Network

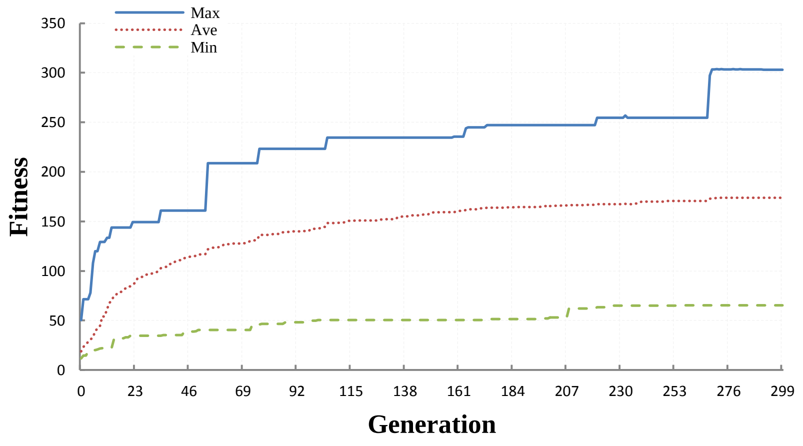


(b) NeuroEvolution of Augmenting Topologies

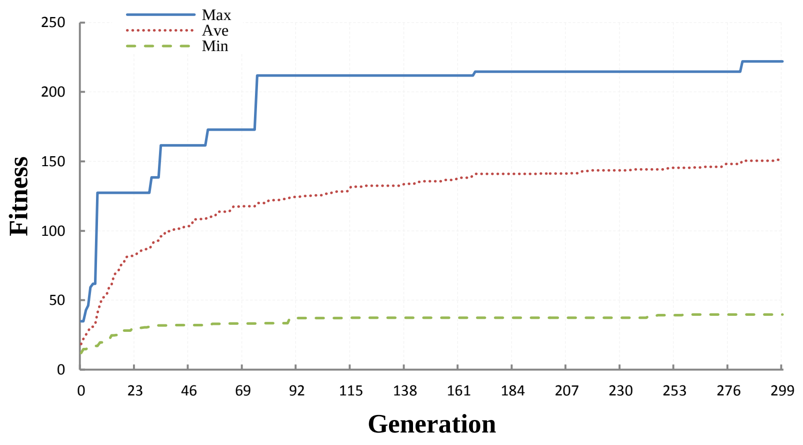
Fig. 5 Fitness graph of the second variety “Arm-based creature”. The evolution of the population's best, worst and average fitness for the GRN experiment (5a) and for the NEAT experiment (5b) averaged over 20 run.

7. Discussion

The results of the experiments show that using GRN to evolve behaviours in a virtual environment is feasible. Creatures based on GRN controller display the ability to process and make use of the sensory data coming back from the joints and input sensors to produce effective locomotion solutions, and in doing so, they are able to learn to adapt to their environment. The results from the simulation show that GRN is an effective method of training simple virtual creatures to perform tasks that involve manipulating a simple control system. The most surprising results were obtained with the hopper morphology, which turned out to be the most difficult morphology to evolve a locomotion solution for. GRN look more efficient when trait examples that require lot information from its environment. Crawling experiment demonstrates that when the joint is highly complex, GRN fails to provide significant benefit over NEAT. In other hand, Arm-based experiment also shows that when the creature’s morphology is composed of a complex structure with similar part where the behavioural system is required to coordinate its movement to generate the desired behaviour, GRN fails to provide efficient locomotive solution. This drop in performance can be explained by the genotype-phenotype mapping that usually works with simple problems, but cannot process domains with very large phenotypic solutions that contain similar parts. This problem could be handled by finding an appropriate generative encoding for GRN’s genome that enables it to compress the description of the solution such that information can be reused and allowing the final solution to contain more components than the description itself. Based



(a) Gene Regulatory Network



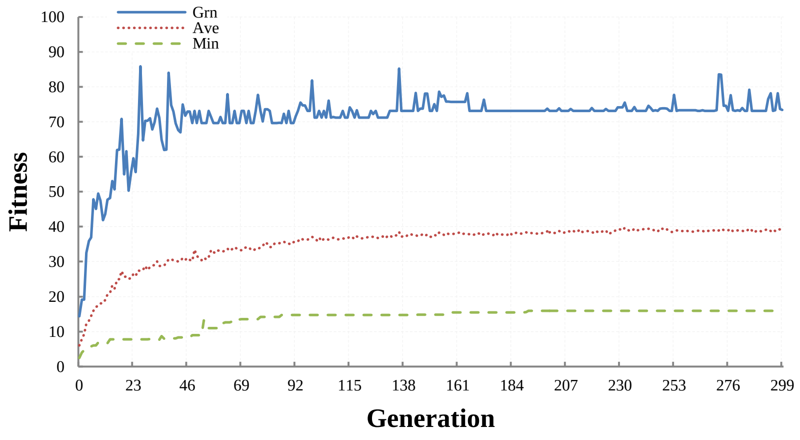
(b) NeuroEvolution of Augmenting Topologies

Fig. 6 Fitness graph of the fourth variety "Hopper". The evolution of the population's best, worst and average fitness for the GRN experiment (6a) and for the NEAT experiment (6b) averaged over 20 run.

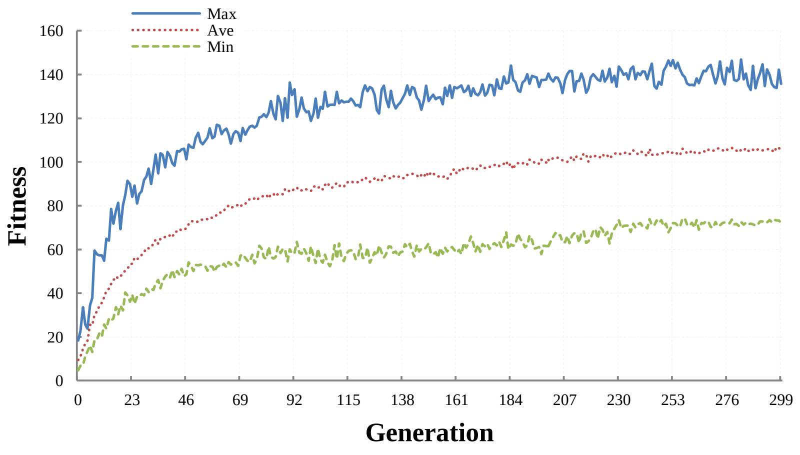
on four experimentations, we concluded that GRN improves upon NEAT in control creature domain for a relatively simple creature's morphology, but that the benefit disappeared as the morphology is complicated.

8. Conclusion and future work

The contribution presented in this paper is the proposal of a novel approach for the evolution of autonomous articulated creatures. The proposed algorithm is inspired by GRN – an algorithm for controlling cell behaviours in developmental models. We demonstrated the advantages and disadvantages of GRN on four tasks: crawling, arm-based moving, somersaulting and running. Results from the arm-based moving and Running experiment have confirmed that GRN does not provide an advantage for tasks where creature consists of complex structures with similar parts and employing complex joints. However, crawling and somersaulting experiments have shown that if the morphology of creature is composed of simple structure and the joint that was used to attach this structure are simpler than the GRN algorithm significantly outperforms NEAT even whether the behavior to be generated are complex or need a lot of information about its environment. The results of our experiences using the GRN as a creature's controller also emphasize that this approach can be used in many agent-based applications, the only requirement is to be able to convert the input and output signals into normalized concentration values and finding a proper way of linking the agent sensors and actuators to the GRN. The major contribution of this work is not simply a good result, but rather a new research direction for evolving effective methods of locomotion through direct



(a) Gene Regulatory Network



(b) NeuroEvolution of Augmenting Topologies

Fig. 7 Fitness graph of the fourth variety "Runner". The evolution of the population's best, worst and average fitness for the GRN experiment (7a) and for the NEAT experiment (7b) averaged over 20 run.

encoding. Future developments should take into consideration new improvements in this GRN model in order to handle the problem of controlling complex morphologies and enhancing its performance to control complex joints which will allow exploring the possibility of evolving ever more complex behaviours such as path following and competition amongst the creatures.

References:

[1] S. Cussat-Blanc and J. Pollack, "A cell-based developmental model to generate robot morphologies," in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, 2012, pp. 537-544.

[2] J. Disset, S. Cussat-Blanc, and Y. Duthen, "Self-organization of Symbiotic Multicellular Structures," in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, pp. 3-5.

[3] R. Doursat, "Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering," in *Organic computing*, ed: Springer, 2008, pp. 167-199.

[4] M. Joachimczak and B. Wróbel, "Evo-devo in silico-a Model of a Gene Network Regulating Multicellular Development in 3D Space with Artificial Physics," in *ALIFE*, 2008, pp. 297-304.

[5] M. Joachimczak and B. Wróbel, "Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension," in *Advances in Artificial Life. Darwin Meets von Neumann*, ed: Springer, 2011, pp. 35-43.

[6] J. Knabe, M. Schilstra, and C. L. Nehaniv, "Evolution and morphogenesis of

differentiated multicellular organisms: autonomously generated diffusion gradients for positional information," *Artificial Life XI*, 2008.

[7] T. Reil, "Dynamics of gene expression in an artificial genome—implications for biological and artificial ontogeny," in *Advances in Artificial Life*, ed: Springer, 1999, pp. 457-466.

[8] P. E. Hotz, "Combining developmental processes and their physics in an artificial evolutionary system to evolve shapes," *On Growth, Form and Computers*, pp. 302-318, 2003.

[9] W. Banzhaf, "Artificial regulatory networks and genetic programming," in *Genetic programming theory and practice*, ed: Springer, 2003, pp. 43-61.

[10] S. Cussat-Blanc, S. Sanchez, and Y. Duthen, "Controlling cooperative and conflicting continuous actions with a Gene Regulatory Network," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 2012, pp. 187-194.

[11] H. Guo, Y. Meng, and Y. Jin, "A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network," *BioSystems*, vol. 98, pp. 193-203, 2009.

[12] M. Nicolau, M. Schoenauer, and W. Banzhaf, "Evolving genes to balance a pole," in *Genetic Programming*, ed: Springer, 2010, pp. 196-207.

[13] K. Sims, "Evolving 3D morphology and behavior by competition," *Artificial life*, vol. 1, pp. 353-372, 1994.

[14] K. Sims, "Evolving virtual creatures," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994, pp. 15-22.

[15] N. Chaumont, R. Egli, and C. Adami, "Evolving virtual creatures and catapults," *Artificial life*, vol. 13, pp. 139-157, 2007.

[16] G. S. Hornby and J. B. Pollack, "Evolving L-systems to generate virtual creatures," *Computers & Graphics*, vol. 25, pp. 1041-1048, 2001.

[17] P. Krčah, "Evolving virtual creatures revisited," in *GECCO*, 2007, p. 341.

[18] P. Krčah, "Towards efficient evolutionary design of autonomous robots," in *Evolvable Systems: From Biology to Hardware*, ed: Springer, 2008, pp. 153-164.

[19] N. Lassabe, H. Luga, and Y. Duthen, "A new step for evolving creatures," *IEEE-ALife*, vol. 7, pp. 243-251, 2007.

[20] T. Miconi and A. Channon, "An improved system for artificial creatures evolution," *Proceedings of Artificial Life X*, pp. 255-261, 2006.

[21] A. Tibermacine and N. Djedi, "NEAT neural networks to control and simulate virtual creature's locomotion," in *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*, 2014, pp. 9-14.

[22] M. S. Saad, H. Jamaluddin, and I. Z. Darus, "PID controller tuning using evolutionary algorithms," *Wseas transactions on Systems and Control*, vol. 7, pp. 139-149, 2012.

[23] H.-C. Chen, D.-H. Guo, and H.-M. Feng, "Fuzzy embedded mobile robot systems design through the evolutionary PSO learning algorithm," *Wseas Transactions on Systems*, vol. 10, pp. 259-269, 2011.

[24] S. Vassileva and F. Neri, "An introduction to the special issue: modeling and control of integrated bio-systems," *Wseas Transactions on Systems*, vol. 7, pp. 221-222, 2012.

[25] A. S. Staines and F. Neri, "A Matrix Transition Oriented Net for Modeling Distributed Complex Computer and Communication Systems," *Wseas Transaction on Systems*, vol. 13, 2014.

[26] M. Camilleri, F. Neri, and M. Papoutsidakis, "An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques," *Wseas Transactions on Systems*, vol. 13, pp. 242-251.

[27] M. Papoutsidakis, D. Piromalis, F. Neri, and M. Camilleri, "Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control," *Wseas Transactions on Systems*, vol. 13, pp. 242-251, 2014.

[28] T. Ababsa, N. Djedi, Y. Duthen, and S. C. Blanc, "Decentralized approach to evolve the structure of metamorphic robots," in *Artificial Life (ALIFE), 2013 IEEE Symposium on*, 2013, pp. 74-81.

[29] N. Ouannes, N. Djedi, H. Luga, and Y. Duthen, "Modeling a bacterial ecosystem through chemotaxis simulation of a single cell," *Artificial Life and Robotics*, pp. 1-6, 2014.

[30] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through

augmenting topologies," *Evolutionary computation*, vol. 10, pp. 99-127, 2002.

[31] K. O. Stanley and R. Miikkulainen, "Efficient Evolution of Neural Network Topologies," *Network (Phenotype)*, vol. 1, p. 3, 2002.

[32] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Evolving competitive car controllers for racing games with neuroevolution," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 1179-1186.

[33] N. Kohl, K. Stanley, R. Miikkulainen, M. Samples, and R. Sherony, "Evolving a real-world vehicle warning system," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1681-1688.

[34] M. Wittkamp, L. Barone, and P. Hingston, "Using NEAT for continuous adaptation and teamwork formation in Pacman," in *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, 2008, pp. 234-242.

[35] K. O. Stanley and R. Miikkulainen, "Evolving a roving eye for go," in *Genetic and Evolutionary Computation–GECCO 2004*, 2004, pp. 1226-1238.

[36] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the NERO video game," *Evolutionary Computation, IEEE Transactions on*, vol. 9, pp. 653-668, 2005.

[37] S. Cussat-Blanc and J. Pollack, "Using Pictures to Visualize the Complexity of Gene Regulatory Networks," in *Artificial life*, 2012, pp. 491-498.

[38] D. Wilson, E. Awa, S. Cussat-Blanc, K. Veeramachaneni, and U.-M. O'Reilly, "On learning to generate wind farm layouts," in *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, 2013, pp. 767-774.

[39] K. I. Harrington, E. Awa, S. Cussat-Blanc, and J. Pollack, "Robot coverage control by evolved neuromodulation," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1-8.

[40] S. Sanchez and S. Cussat-Blanc, "Gene regulated car driving: using a gene regulatory network to drive a virtual car," *Genetic Programming and Evolvable Machines*, pp. 1-35.