Exploring Grammatical Evolution for Horse Gait Optimisation

James E. Murphy, Michael O'Neill, and Hamish Carr

University College Dublin, Belfield,
Dublin 4, Ireland
{james.murphy,m.oneill,hamish.carr}@ucd.ie
http://ncra.ucd.ie

Abstract. Physics-based animal animations require data for realistic motion. This data is expensive to acquire through motion capture and inaccurate when estimated by an artist. Grammatical Evolution (GE) can be used to optimise pre-existing motion data or generate novel motions. Optimised motion data produces sustained locomotion in a physics-based model. To explore the use of GE for gait optimisation, the motion data of a walking horse, from a veterinary publication, is optimised for a physics-based horse model. The results of several grammars are presented and discussed. GE was found to be successful for optimising motion data using a grammar based on the concatenation of sinusoidal functions.

Key words: Grammatical Evolution, physics-based animation, gait optimisation, quadrupedal locomotion, Fourier analysis

1 Introduction

A well-constructed physics-based animal model can produce physically realistic animations given good quality motion data. This data can be expensive to measure and unreliable to estimate. We propose to take some potentially inaccurate motion data and optimise it using an evolutionary algorithm approach.

Animal motion data is sometimes published in biomechanical and veterinary literature. Data can also be gleaned from sequential high-speed photographs of an animal in motion [1]. Although this data can be extracted and formatted for use with a physics-based model, it will not automatically produce stable locomotion. The data must be optimised for use with a specific model.

In this paper we present an approach to gait optimisation which utilises the Grammatical Evolution (GE) evolutionary computation technique. Rather than simply performing a parameterised optimisation on the data, which is known to work well, we propose a Fourier analysis based approach. By representing an animal's gait as a summation of sinusoidal functions, we optimise a more minimal set of parameters, as we successively concatenate sinusoidal functions of differing amplitude, phase and frequency. This representation is compact and mimics the sinusoidal nature of muscle movement. It also gives the evolutionary process more freedom to evolve than a parameterised optimisation, including the potential to retarget gait cycles from one animal to another.

We discuss how GE is used to optimise a walk gait for a physics-based horse model. Using a horse simulation application as the fitness function, gait cycles are generated and assessed. The gait cycle representation and the manner in which the simulation application acts as fitness function is described in Section 3. We compare the results of a variety of grammar types and speed-up strategies in Section 4. Section 5 concludes the paper with a brief discussion of the presented results. First of all, we present a brief overview of some related work.

2 Related Work

A gait is a pattern in which an animal moves using its limbs. The four main natural gaits of a horse are the walk, trot, canter and gallop. During locomotion, a sequence of muscles, determined by the current gait pattern, are contracted periodically to produce movement in the bones. The rotating bones cause the hoof to push off the ground surface, thrusting the animal forwards. The cyclical nature of a gait allows it to be quantified and described in terms of gait cycles.

A gait cycle begins when a foot contacts the ground and ends when that same foot contacts the ground again. The fraction of the cycle in which the foot is in contact with the ground is called the duty factor. The most distinguishing feature of a gait cycle is its footfall sequence, as the order in which an animal's feet impact the ground differs between gaits. Gaits are also described in terms of stride length and stride frequency [2].

The gait an animal will utilise, when travelling at a particular velocity, can be predicted based on the dynamic similarity principles [3]. This theory is based on the dimensionless Froude number, which is a function of the animal's velocity, height of hip from the ground and acceleration due to gravity. The dynamic similarity hypothesis states that when different mammals are travelling at equal Froude numbers, their gait patterns will be dynamically similar. This is exploited in our fitness function as will be described in Section 3.2.

Reproducing animal locomotion through animation is a well-studied topic in computer graphics and other disciplines such as robotics. Physics-based animal models can realistically reproduce animal motion given a well-constructed model and motion controller. An animal model is constructed as a series of interconnected rigid bodies. The rigid bodies represent the animal's bones and the connections between these bones are the joints. A physics engine ensures that the bones react in a physically realistic manner to gravity, friction, collisions, applied forces and torques [4]. Forces and torques applied to the bones produce linear and angular accelerations. As the bones are connected together by joints, forces applied to a single rigid body may propagate throughout the rigid body system. To produce motion in the model, a motion controller applies torques of specific magnitude about each of the joints in a limb, with precise timing, to produce a desired bone rotation. The development of animal models and motion controllers has been studied in the computer animation field for many years.

Genetic Algorithms are used to generate quadruped robot gaits in [5–7]. The fitness function in [5] is of interest as it is concerned with finding a gait that uses minimal energy, while covering a required distance at a specified speed. This can be applied to gait generation for quadruped animations as they state that the optimal gaits produced for the robots are comparable to those expected of a real-life animal travelling at the same Froude number. Each flavour of evolutionary algorithm will vary in performance depending on the problem domain. We apply a relatively new type of evolutionary algorithm called Grammatical Evolution (GE) to gait optimisation. GE is one of the most popular forms of grammar-based Genetic Programming (GP) due to the convenience by which a user can specify and modify the grammar, whilst ignoring the task of designing specific genetic search operators. It has been successfully employed to financial prediction [9], but has never been applied to gait generation and optimisation.

The GE search adapts principles from molecular biology. GE is distinct from other evolutionary algorithms as it uses a variable length binary or integer string to derive solutions from a Backus Naur Form grammar. In GE, the evolutionary algorithm's genetic operators are applied to the strings (genotypes) rather than problem domain solutions (phenotypes). Potential complexities of the phenotype are inconsequential. This makes GE a good choice for gait optimisation because the horse gait problem domain is large and complex, as will be discussed in the following section. For further information on GE please refer to [10].

3 GE for Gait Optimisation

Our goal is to optimise gait data and explore how motion variations in a joint affect an animal's gait. GP allows us to explore these motion variations as a gait is optimised. The advantage of GP over other model/structure learning methods, such as Neural Networks, is that output structures are generally in a human-understandable format. We can examine motions and identify patterns which may allow us to generate motion for morphologies for which we have no data.

Measured data, and experience of manually tuning motion data for animation purposes, provide a general template of how each joint should move for realistic motion of the model. Information on an animal's musculature and joint limits can be implicitly included through constraints imposed on the evolving structures. We wish to compare results of multiple optimisations utilising differing styles and levels of constraint on evolutionary freedom. Grammar-based GP, specifically GE, is ideal for this purpose as various constraint methods can be easily incorporated into a grammar and rapidly performance tested.

During the GE process, each phenotype produced from a grammar is passed to a simulation application as motion data. This application, acting as fitness function, assesses each phenotype for a few gait cycles. The GE process proceeds until an optimal solution is found. Technical details of the GE set-up are beyond the scope of this paper however, the GE parameters used for each run are presented in Table 1. All results in Section 4 are generated using GEVA [11].

Table 1. GE parameters used for every run.

Parameter	**Value**
Generations	50
Population	75
Max. wrapping	3
Replacement	generational
Elite size	7
Selection	Tournament (3)
Initialisation	RampedFullGrow
Max. depth	10
Grow prob.	0.5
Crossover prob.	0.9
Crossover point	fixed
Mutation prob.	0.02

Sequential screen-shots of the simulation application, implemented using Open Dynamics Engine [8], are shown in Figure 1. Simulation application details are beyond the scope of this paper however, the most important aspect of the simulation application is the gait cycle representation. It is based on the observation that, as muscles tend to relax and contract in a sinusoidal manner, real-life joint-angle data can be decomposed into a sum of sinusoidal functions through Fourier analysis. Motion data for each joint is therefore represented as a sum of these sinusoids, referred to as the sum-of-sines representation. It is a compact and elegant way of representing cyclical data such as a single cycle of a gait, which is used repeatedly to produce sustained locomotion. Details of our grammars and fitness function follow in Section 3.1 and Section 3.2 respectively.

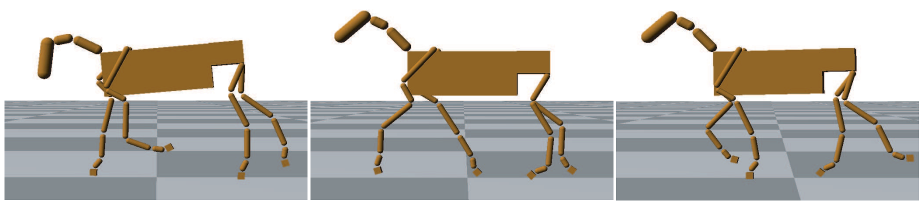

Fig. 1. Sequential screen-shots of the physics-based horse model walking.

3.1 Gait Optimisation Grammar

The grammars must allow for construction of syntactically correct motion data in the sum-of-sines format. If the goal is to optimise real-life animal gait data, that data must be incorporated into the grammar. The greatest consideration in the grammar construction is the degree to which phenotypes are allowed differ from the seed data. The grammar should produce motion data for each joint in the model, represented as a summation of sinusoidal functions, in the form:

$$\langle amp \rangle * sin(\langle freq \rangle * 2 * PI * time + \langle phase \rangle) \tag{1}$$

Motion data, extracted from plots in [2], is decomposed into its component sinusoidal functions through Fourier analysis. Fourier terms whose amplitude is below some arbitrarily chosen threshold value are discarded, leaving a more compact summation of sinusoidal functions approximating the motion data.

This representation could be simply optimised by manipulating the variable values of its constituent sinusoidal functions. The values of the *amp* and *phase* parameters in each function can be optimised within some defined range. As the extracted motion, or seed, data dictates the frequencies of the functions we can optimise, the range of potential solutions is constrained. To provide greater flexibility, our GE grammars produce phenotypes by concatenating sinusoidal functions to the seed data from a fuller range of frequencies, not just those dictated by the minimal form of the Fourier analysis. Depending on the grammar, phenotypes can be constrained to remain close to the seed data or allowed to deviate. Each of the generated phenotypes must be assessed in the animal simulation application by means of a fitness function.

3.2 Gait Optimisation Fitness Function

Our fitness function is based on energy efficiency and gait predictions based on the dimensionless Froude number, which is calculated as follows:

$$Fr = v/\sqrt{g * h} \tag{2}$$

Where *Fr* is the Froude number, v is velocity, g is gravity and h is height of the animal's hip from the ground. Dynamic similarity theory states that an animal travelling at a particular Froude number will share gait characteristics with other animals travelling at the same Froude number. This implies that if we have gait information for a single animal moving at a range of Froude numbers, we can predict the gaits of other animals. Such data is published in [3].

A gait is optimised to move the model at a particular Froude number. From that Froude value we calculate velocity and predict the phase difference between limbs, stride frequency, stride length and duty factors.

These predictions are used in the fitness function to score the phenotypes. The phase difference and stride frequency are set in the application based on the Froude number argument. Only the joint-angle motion data is generated. An optimal generated gait moves the model with the velocity, duty factor and stride length values predicted by the dynamic similarity theory. Energy efficiency of a gait is also a factor in the fitness score. In nature, animal morphology and joint-angle motion has generally evolved to use the minimum energy to travel a desired distance at a desired velocity. The fitness function therefore rewards those phenotypes that use minimal energy.

Each of the fitness components has an associated weight. Each component's contribution to the fitness score is a function of its weight and a measure of the error from its predicted value. The better the gait, the lower the score. Using this technique, a perfect score of 0 should never occur. The energy component is a weighted sum of the model's total energy use, averaged per cycle. As the model must expend energy to move, even the most optimal gaits will have a positive fitness value, as will be seen in the following section.

4 Experiments and Results

To explore the use of GE for gait optimisation, grammars which utilise seed data are investigated as well as free-style grammars which do not. In the case of these unconstrained, free-style grammars, the goal is not to produce aesthetically realistic gaits, but rather create novel movement and test the capabilities of our multivariable fitness function. It is apparent from experimentation, that grammars providing parameterised optimisation of seed data perform well. A future goal of our research is to retarget optimal gait data from one animal to another as outlined in [12]. Our experiments have shown that a simple parameterised optimisation of the seed data does not have the scope to alter motion data significantly enough to allow retargeting to animals with different morphology to that of the source. This motivates our exploration of grammars which provide the flexibility to evolve gaits from one animal to another.

The results presented in this paper are divided into three categories. In Section 4.1, we describe a grammar which optimises data by concatenating sinusoidal functions to the seed data. This grammar is compared with parameterised optimisation approaches. In Section 4.2, the investigation of the concatenating functions grammar continues with two attempts to speed-up the evolutionary process through generational manipulation of the grammar and fitness function. Finally, in Section 4.3, grammars that do not use seed data are presented.

4.1 Parameterised Optimisation Comparison

For our parameterised optimisation, each term in the compact summation of sinusoidal functions data approximation is represented as a triple (amplitude, frequency and phase). Each of these values is optimised within a range of 25% of itself. While this approach performs well on data which has been measured from an animal with the same morphology as our model, as in the presented case, it is constrained to produce a limited set of motion, unsuitable for interspecies gait retargeting. We aim to develop a grammar which can at least match and hopefully surpass the parameterised approach, while having the scope to completely diverge from the seed data during our retargeting experiments.

The grammar presented in Figure 2 optimises the seed data by adding (or subtracting) sine and cosine functions of differing amplitude and frequency to the seed data, which is itself in the sum-of-sines form. Generated motion may deviate from the seed data through addition of functions of differing frequency.

```
<prog> ::= <fcurve0> <newline> <fcurve1> <newline> ... <fcurve11>

<fcurve0> ::= <curve0> | <curve0> + <funcs>
...
<fcurve11> ::= <curve11> | <curve11> + <funcs>

<funcs> ::= <funcs> <op> <funcs>
          | <function>
          | <med_amp_var>

<op> ::= + | -

<function> ::= <low_amp_var> * sin( <low_freq_var> * 2 * PI * t )
             | <low_amp_var> * cos( <low_freq_var> * 2 * PI * t )
             | <med_amp_var> * sin( <med_freq_var> * 2 * PI * t )
             | <med_amp_var> * cos( <med_freq_var> * 2 * PI * t )
             | <hi_amp_var> * sin( <hi_freq_var> * 2 * PI * t )
             | <hi_amp_var> * cos( <hi_freq_var> * 2 * PI * t )

<low_freq_var> ::= 1 | 2
<med_freq_var> ::= 3 | 4
<hi_freq_var> ::= 5 | 6 | 7 | 8

<low_amp_var> ::= 0 | 0.25 | 0.5 | ... | 20
<med_amp_var> ::= 0 | 0.1 | 0.2 | ... | 4
<hi_amp_var> ::= 0 | 0.05 | 0.1 | ... | 1

<curve0> ::= 6.97+7.7*sin(1*2*PI*t+-1.07)+2.56*sin(2*2*PI*t+2.97) ...
...
<curve11> ::= ...
```

Fig. 2. An illustrative example of a grammar based on the concatenation of sinusoidal functions to the seed data. Note the seed data at the bottom of the grammar. (Omitted terms represented by '**...**'.)

The frequencies of the appendable sinusoidal functions and the seed data summations have a range of 1 to 8Hz. Fourier analyis of the seed data shows that higher frequency functions have amplitudes less than our arbitrarily chosen threshold value of 1. These functions are considered less influential to the overall movement and are discarded for compactness. The grammar also constrains appended functions of a particular frequency to have an amplitude of a specific range. This range is based on observations from the Fourier analysis.

The best fitness plot in Figure 3, shows the numerical (parameterised) optimisation starting off worst. Gradually it improves and achieves a similar optimal solution score to the concatenating functions grammar. The overall winner in terms of best fitness is a grammar which uses a combination of parameterised optimisation and concatenating functions. The sinusoidal functions are added to the seed data, whose parameters are optimised in parallel.

4.2 Concatenating Functions Speed-Up Attempts

To improve the performance of the concatenating functions grammar, two tests are presented in which the fitness function and grammar dynamically changes.

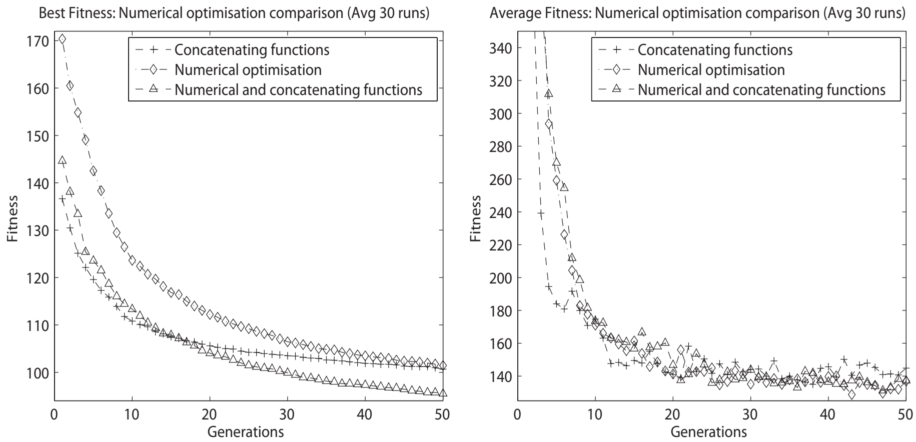

Fig. 3. A numerical optimisation approach is contrasted with the concatenating functions grammar and a hybrid of the two approaches. Best and average fitness (averaged, 30 runs) are presented on the left and right respectively. *(Note difference in scale.)*

For the varying fitness function test, the fitness weights are changed from generation to generation, as shown in Table 2. It was hoped that this would speed up the evolutionary process, prevent the process from becoming stuck at local minima and produce a more well rounded solution, i.e. one which optimises aspects of velocity, duty factor and stride length equally.

The results presented in Figure 4 do not show any speed-up. The change in fitness function does seem to drive the evolution forward in some situations however, in this instance, the change in fitness function causes a plateau in the best fitness score from generation 10-20. The large spike in the corresponding average fitness plot indicates that the model has a high distance error value at generation 10. The increase in the distance-scalar's weight causes temporary chaos. The process recovers and quickly proceeds to an optimal solution.

Table 2. Fitness function weights during the fitness function variation run.

Fitness measure	**Gen. 0-10**	**Gen. 11-20**	**Gen. 21-30**	**Gen. 31-40**	**Gen.41-50**
Distance	1	2	1	1	1
Duty factor	1	1	2	1	1
Stride length	1	1	1	2	1
Energy	1	1	1	1	1

The second test involves restricting joint motion on a generational basis. The sequence in which joints are given freedom is presented in Table 3. While again this approach does not speed up the evolutionary process, it is clear from the average fitness plots in Figure 4 that the varying joint freedom grammar produces

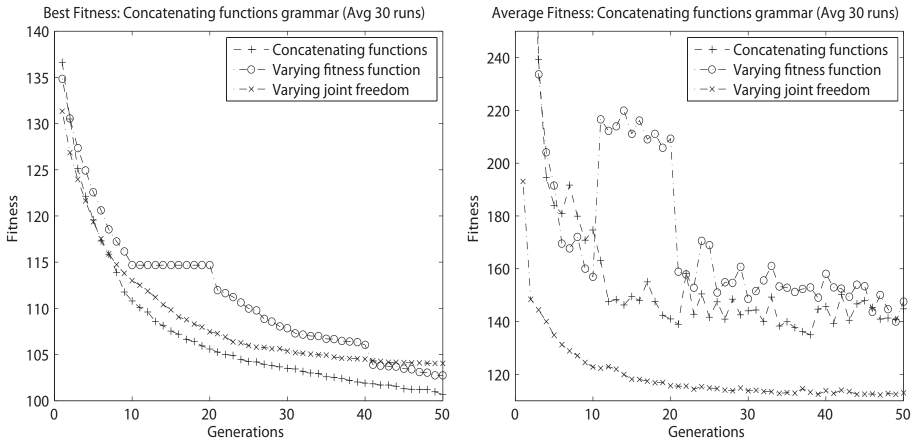

Fig. 4. The concatenating functions grammar with alternating fitness and grammar strategies. Best and average fitness (averaged, 30 runs) are presented on the left and right respectively. *(Note difference in scale.)*

very stable gaits from the earliest generations. The large starting values apparent in most of the average fitness plots are the result of the unviable phenotypes passed to the simulation application, usually at the start of the evolutionary process. The motion data can cause the model to wildly gyrate its limbs or provide such a boisterous gait that the model flips over. These bad phenotypes are awarded a very high score (corresponding to the worst fitness possible). By initially restricting the model's degrees of freedom, production of the bad phenotypes appears minimised. The results are summarised in Table 4.

4.3 Free-Style Grammars

In contrast to previous grammars which contain seed data, the use of free-style grammars is also investigated. As the grammar, shown in Figure 5, does not contain constraints based on the animal's joint limits and muscle distribution, the motions resulting from this free grammar vary greatly across the 30 runs completed. In some instances, the model moves utilising only its front or hind limbs. Other runs exhibit a sequence of sudden hops to move the model. On a few occasions, motion is produced by placing the limbs squarely under the animal's body and using a high-frequency, small amplitude, back and forth motion to "vibrate" the animal along the surface. The fact that these very different gait cycles achieve similar fitness scores demonstrates a flaw in the multivariable fitness function. It appears that improvements in one aspect of the fitness score can overshadow other components, which suggests a more sophisticated fitness function is required. Currently, the ultimate shape of a solution using a free-style grammar may be randomly determined early in the evolutionary process.

Table 3. Generational joint freedom. Only joints with a ✓ are free to move and evolve motion data for each generation range. All other joints remain static.

Joint	Gen. 0-10	Gen. 11-20	Gen. 21-30	Gen. 31-50
Scapula (fore)	✓	✓	✓	✓
Shoulder (fore)	✓	✓	✓	✓
Elbow (fore)	-	✓	✓	✓
Carpal (fore)	-	-	✓	✓
Fetlock (fore)	-	-	-	✓
Hip (hind)	✓	✓	✓	✓
Stifle (hind)	-	✓	✓	✓
Tarsal (hind)	-	-	✓	✓
Fetlock (hind)	-	-	-	✓
Proximal (neck)	-	✓	✓	✓
Mid (neck)	-	-	✓	✓
Atlas (head/neck)	-	-	-	✓

Table 4. Best and average fitness with standard deviations (averaged, 30 runs), of the concatenating functions grammar with the fitness function and grammar variations.

Strategy	Avg. Best	Std. Dev.	Avg. Avg.	Std. Dev.
None	100.333	8.5231	140.9203	43.1542
Varying fitness weights	102.7	5.3572	154.5484	50.3666
Varying joint freedom	104.0333	7.8892	112.2726	8.4968

An interactive evolutionary computation technique could be employed in the early generations to guide the process towards a realistic motion.

A free-style sum of sinusoidal functions technique is also tested. The grammar is similar to the concatenating functions grammar in Figure 2, except that the functions are not concatenated to any seed data. Our knowledge that animal gait data can be decomposed into a summation of sinusoidal functions, each having parameters which fall within particular frequency, amplitude and phase values, is incorporated into the grammar. In contrast to the free grammar in Figure 5, information about the animal's musculature is implicitly included through the specific frequency and amplitude ranges of the sinusoids. While a small number of generated gaits are visually unrealistic, the majority are comparable to the real-life motion of a horse. While not as good as those grammars which include seed data, there is potential for improvement given a more sophisticated fitness function and increased population and generation values.

Figure 6 shows the free and sinusoidal grammar scoring comparably to the concatenating functions grammar in terms of fitness. This illustrates the pitfalls of using a multivariable fitness function and few motion constraints. Out of the free grammar's 30 runs, very different "optimal" solutions score similarly. It demonstrates that if realism is the goal, seed data, or constraints built into the grammar based on observations of animal motion, are required.

```
<prog> ::= <fcurve> <newline> ... <fcurve>

<fcurve> ::= <expr>

<expr> ::= <expr> <op> <expr>
         | (<expr> <op> <expr>)
         | <pre-op> (<expr> * t)
         | <var>

<op> ::= + | - | / | *
<pre-op> ::= sin | cos
```

Fig. 5. An illustrative example of a free grammar. The t variable is required by our simulation application so that generated motion data may be a function of time. The *fcurve* terms, for each joint presented in Table 3, are omitted and represented by '**...**'.

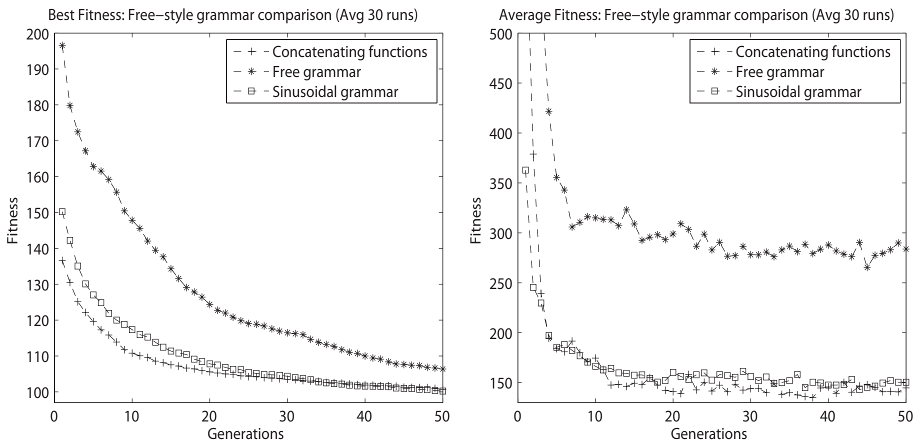

Fig. 6. Free and free-style sinusoidal grammars compared with the concatenating functions grammar. Best and average fitness (averaged, 30 runs) are presented on the left and right respectively. *(Note difference in scale.)*

5 Discussion and Conclusions

GE allows us to rapidly explore different approaches to gait generation. Phenotypes are produced in a human-readable format which assists understanding of gait motion. Grammars also allow us to construct gait representations other than the presented sum-of-sines, necessary for non-cyclical gait transition motions.

Table 5 shows that the concatenating functions grammar improves upon the parameterised approach in terms of fitness score, albeit by a small margin. The overall winner is a combination of the two. The concatenating sinusoidal functions approach is found to be a compact method of representing and optimising gait data. By optimising seed data whilst appending new sinusoidal functions, we have the flexibility to retarget to other morphologies whilst maintaining realism.

Table 5. Overall best and average fitness scores (averaged, 30 runs) achieved by each grammar alongside their respective standard deviations.

Grammar	Avg. Best	Std. Dev.	Avg. Avg.	Std. Dev.
Numerical & concatenating functions	95.3	5.2729	138.8273	57.2904
Concatenating functions	100.333	8.5231	140.9203	43.1542
Sinusoidal grammar	100.4333	6.0039	150.2595	44.2394
Numerical only	101.3333	8.5715	132.6506	37.6833
Free	106.3	19.05	279.1248	81.3519

The multivariable nature of our fitness function allows for significant motion variance between similarily scoring phenotypes. It may be benificial to use a multi-objective optimisation approach in future. With further refinement of the grammars and fitness function, this GE method of gait optimisation can be utilised in our gait retargeting solution. In future we also hope to use a grammar-based approach to produce sophisticated balance and directional systems.

Acknowledgments. Thanks to IRCSET and IBM.

References

1. Muybridge, E.: Horses and Other Animals in Motion. Dover Publications (1985)
2. Back, W., Clayton, H. M.: Equine Locomotion. Harcourt Publishers (2001)
3. Alexander, R. M., Jayes, A. S.: A Dynamic Similarity Hypothesis for the Gaits of Quadrupedal Mammals. In: Journal of Zoology (London), volume 201, pages 135-152. The Zoological Society of London (1983)
4. Parent, R.: Computer Animation: Algorithms and Techniques. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann (2002)
5. Kiguchi, K., Kusumoto Y., Watanabe, K., Izumi, K., Fukuda, T.: Energy-Optimal Gait Analysis of Quadruped Robots. In: Artificial Life and Robotics, volume 6, pages 120-125. Springer Japan (2002)
6. Garder, L.M., Høvin, M.E.: Robot Gaits Evolved by Combining Genetic Algorithms and Binary Hill Climbing. In: Genetic and Evolutionary Computation Conference '06, pages 1165-1170. ACM (2006)
7. Xu, K.,Chen, X., Liu, W., Williams, M.: Legged Robot Gait Locus Generation Based on Genetic Algorithms. In: Proc. International Symposium on Practical Cognitive Agents and Robots, volume 213, pages 51-62. ACM (2006)
8. Smith, R.: Open Dynamics Engine v0.5 User Guide. (2006)
9. Brabazon, A., O'Neill, M.: Biologically Inspired Algorithms for Financial Modelling. Springer (2006)
10. O'Neill, M., Ryan, C.: Grammatical Evolution. Genetic Programming Series. Kluwer Academic Publishers (2003)
11. GEVA: Grammatical Evolution in Java, `http://ncra.ucd.ie/geva`
12. Murphy, J. E., Carr, H., O'Neill, M.: Grammatical Evolution for Gait Retargeting. In: Proc. Theory and Practice of Computer Graphics '08, pages 159-162. Eurographics (2008)