We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

20

Evolutionary Strategies Combined With Novel Binary Hill Climbing Used for Online Walking Pattern Generation in Two Legged Robot

Lena Mariann Garder and Mats Høvin
University of Oslo
Norway

1. Introduction

Evolutionary algorithms (EA) has often been proposed as a method for designing systems for real-world applications (Higuchi et al., 1999). Developing effective gaits for bipedal robots is a difficult task that requires optimization of many parameters in a highly irregular, multidimensional space. In recent years biologically inspired computation methods have been employed by several authors. For instance, Hornby et al. used genetic algorithms (GA) to generate robust gaits on the Aibo quadruped robot (Hornby et al., 2000). GA applied to bipedal locomotion was also proposed by Arakawa and Fukuda (Arakawa & Fukuda, 1996) who made a GA based on energy optimization in order to generate a natural, human-like bipedal gait. One of the main objections to applying EA's in the search for gaits is the time consuming characteristic of these techniques due to the large fitness search space that is normally present. For this reason most approaches have been based on offline and simulator based searches. To reduce the time spent searching large search spaces with EA, various techniques for speeding up the algorithms have been presented. With the increased complexity evolution schema introduced by Torresen (Torresen, 1998), Torresen has shown how to increase the search speed by using a divide and conquer approach, by dividing the problem into subtasks in a character recognition system. Haddow and Tufte have also done experiments with reducing the genotype representation (Haddow & Tufte, 2000). Kalganova (Kalaganova, 2000) has shown how to increase the search speed by evolving incremental and bidirectional to achieve an overall complex behavior both for the complex system to the sub-system, and from the sub-system to the complex system. For an exhaustive description of other approaches readers may refer to Cantú-Paz (Cantú-Paz, 1998).

The robot presented in this paper is a two-legged biped with binary operated pneumatic cylinders. The search space in our experiments was set up to describe the forward speed of the robot given the different gaits, and the goal was to find the most efficient gait with respect to speed. To enable efficient gaits the search space needed to be quite large as the accuracy of the pause lengths between the different leg positions is outmost critical, especially for gaits dominated by jumping movements. The focus has not been on evolving a balancing system as there have been no other sensory feedback than the forward position of the robot. The main goal for our work was to find a search algorithm fast enough to enable real-time gait generation/adaptation where the fitness is provided by the mechanical robot

Source: Bioinspiration and Robotics: Walking and Climbing Robots, Book edited by: Maki K. Habib
ISBN 978-3-902613-15-8, pp. 544, I-Tech, Vienna, Austria, EU, September 2007

without the need for an offline simulator model. In real-time evolution, challenges like explosive pneumatic movements and vibrations, effects the feet-to-floor friction. This vibration makes the robot shoe soles occasionally slip during kick-off and make the system very unpredictable as the robot occasionally may stumble instead of jump even for seemingly optimal patterns.

Figure 1. The two legged robot, "Henriette"

The search space in our experiments was set up to describe the forward speed of the robot given the different gaits, and the goal was to find the most efficient gait with respect to speed.

2. The Robot Hardware

The robot skeleton is made of aluminum and is provided with two identical legs. The height is 40 cm. Each leg is composed of an upper part (i.e. the thigh) connected through a cylindrical joint to the lower part (i.e. the calf). Pneumatic cylinders are attached to the thigh and the calf used for controlling the movements of the calf and the thigh separately. As shown in Fig. 1 and Fig. 3, the rear cylinder in each foot actuates the calf whereas the front cylinder actuates the thigh. The cylinders can either be fully compressed or fully extended (binary operation), and the pneumatic valves are located on top of the robot. The valves are electrically controlled by 4 power switches connected to a PC I/O card (National Instruments DAQ-pad) and the different searching algorithms are implemented in the programming language C++ on the PC. The pneumatic air pressure was set to 8 bar and provided by a stationary compressor. The robot was attached to a balancing rod at the top (Fig. 1 and Fig. 2) making the robot able to move in two dimensions. The other end of the rod was attached to a rotating clamp on a hub. The robot walks around the hub with a radius of 2 meter. In addition to being a balancing aid, the rod supplies the robot with air pressure and control signals from the DAQ-pad. The hub has a built in optical sensor representing the rod angle in 13 bit Gray code.

3. Genetic Algorithm

A genetic algorithm is based on representing a solution to the problem as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic operators to evolve the solutions in order to find the best one(s). In the simple GA approach (Goldberg, 1989), (Torresen, 2004) the chromosomes are randomly initiated and the only genetic operators used are mutation and crossover. The selection process is done by roulette wheel selection.

Figure 2. The entire system containing the robot, the balancing rod and the hub

3.1 The Chromosome Coding

In our experiments each gait is coded by a 30 bit chromosome. The chromosome represents three body positions each followed by a variable pause. A body position is composed of the positions of the 2 legs (4 cylinders) and represented by four bits (Fig. 3) each describing the status of the corresponding cylinder (compressed or extracted). A complete gait is then created by executing 3 body positions with 3 appropriate pauses in between. Each pause length is represented by 6 bits. The pause length is represented as a binary number corresponding to pauses from 50ms to 300ms. Various simulations have shown no GA search speed improvement by representing the pauses in Gray code. Two cylinders can move a single leg to 4 different positions. Two legs with four cylinders can hold 16 different positions, and three following positions with 6 bits pauses in between make a search space of 2^{30} = 1 073 741 824 different gaits. Although the search space can be made slightly smaller by representing each gait by a cyclic coding (Parker, 2001) our experiments have shown no noticeable difference in search speed for cyclic/non cyclic coding for this robot. The size of this search space clearly requires a more efficient search algorithm than simple GA in order to enable real-time gait development in hardware.

3.2 Pauses

A gait is composed of leg positions and pauses. In our robot evolution we have found that the most efficient gaits with respect to forward speed are gaits dominated by jumping

movements. In a jumping movement the pause length between each leg kick is outmost critical as the robot may stumble if the timing of the leg kick is just slightly wrong.
Measurements show that a pause length deviation in the magnitude of 20ms can make the difference between a relatively useless and a highly effective gait. It is however a trade-off between the desire to represent the pause lengths with a high number of bits and the exponential decrease in search speed for each extra bit used due to the increased size of the search space.

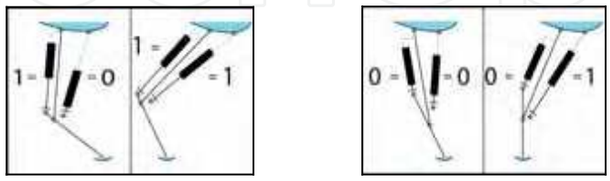

Figure 3. The four leg positions

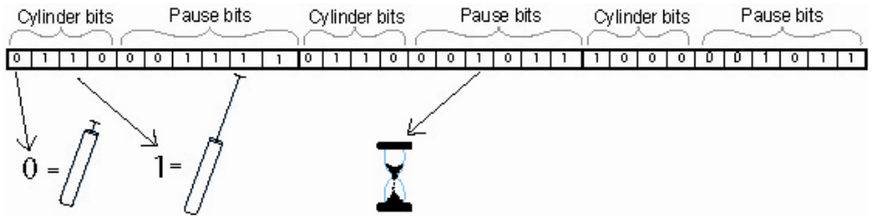

Figure 4. Chromosome representation 1

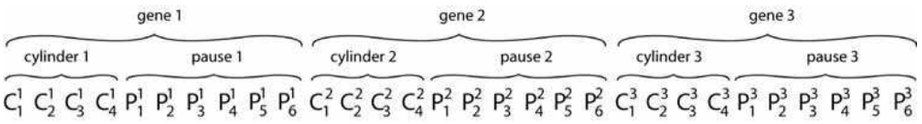

Figure 5. Chromosome representation 2

4. Simulated Results

To compare the efficiency of the different search algorithms against each other the robot was first simulated in software.

4.1 The Simulator

A simple mechanical chicken-robot simulator has been implemented in C++. This simulator models the robot with exact physical dimensions and a weight of 3 kg. The centre of gravity is located at the hip joint. It was found very difficult to model the feet-to-floor friction force exactly as this force is heavily modulated by large vibrations in the robot body and supporting rod during walking/jumping. The feet-to floor friction force is a very important factor for developing efficient jumping patterns and the lack of an exact model for this effect is assumed to be the main weakness of the simulator. The fitness of each chromosome (gait) is a function of the forward speed of the robot caused by the corresponding chromosome. Each gait is repeated 3 times in sequence to reduce the impact caused by the initial leg positions. A movement in the backward direction causes the fitness to be zero.

4.2 Search Space Topology

The optimal search algorithm for a given problem depends heavily on the topology of the search space. For the chromosome coding described in chapter 3 and the chosen software robot model we have tried to get an overview of this topology by separating the search space in two parts, one part generated by the pause bits and one part generated by the leg position bits. Fig. 6 shows a plot of the fitness landscape for all possible leg positions in a single chromosome (gait) were all 3 pause lengths are fixed at 100ms. The size of this search space is $24^3 = 4096$ leg positions. This plot indicates that the part of the overall search space generated by the leg positions is very chaotic although there may be some repetitive phenomena. A similar topology has been found for other choices of constant pause lengths. The different leg positions are sorted by the Gray value of their corresponding bits to keep the bit difference between neighboring chromosomes in the plot as low as possible, but even so the landscape is chaotic with many narrow peaks. In Fig. 7 the fitness landscape is plotted for different pause lengths where the leg positions are kept constant. To make the fitness landscape visually informative one of the 3 pause lengths are also kept constant at 70 ms resulting in a three dimensional plot. As this plot indicates the part of the overall fitness landscape generated by the pause lengths is smooth and will typically contain a few numbers of maxima. In this type of landscape a hill climbing search will normally be more efficient than a genetic algorithm.

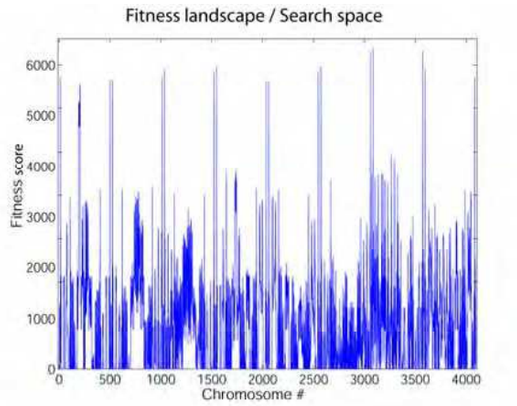

Figure 6. Plot of the fitness landscape generated by different body positions

4.3 Simple GA/ES Simulations

The focus for this real-time application has been to find a search algorithm capable of finding an optimal gait in less than 20 generations. The first search approach was to perform a search for an optimal chromosome (gait) in the global search space consisting of 2^{30} different chromosome values. Simple and more advanced genetic algorithms were tested against different evolutionary strategies (ES) (Goldberg, 1989). ES's showed to be favorable for this particular application. In all our simulations 5% noise is added to the fitness function to model practical effect such as variable foot friction, vibrations, variable air pressure and pause length deviations caused by non-ideal real-time behavior of the XP operating system.

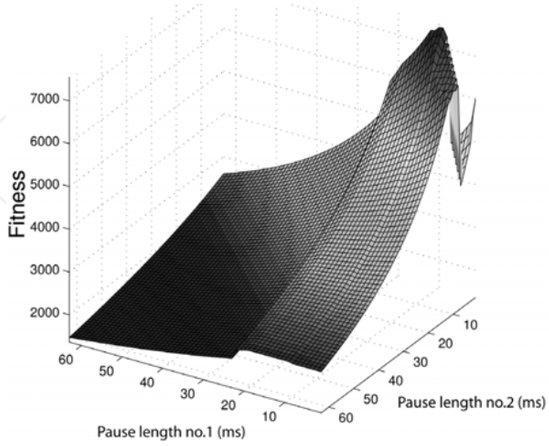

Figure 7. Plot of the fitness landscape generated by different pause values

An evolutionary strategy with roulette wheel selection, elitism, a population size of 10 chromosomes, no crossover but with as high as 20% mutation probability for each bit was found to be the most effective. The high mutation probability indicates that the ES is struggling with the topology in this global search space. This result is not surprising as the global search space is assumed to be dominated by the chaotic and complex phenomena shown in the partial search space shown in Fig. 6. In Fig. 11 we see that ES produces slightly less than twice as effective gaits compared to a stochastic search after 15 generations. In all plots each graph shows the mean result from 1000 simulations with randomly initiated populations. 5 different graphs are shown to illustrate the consistency of the simulations. By optimizing the simple GA with different types of selection models, parameter tuning and pause variation, the gaits still did not evolve fast enough for real time evolution.

4.4 The Incremental ES Approach

The next approach was to evolve the partial search spaces shown in Fig. 6 and Fig. 7 separately by an incremental evolutionary genetic algorithm. Incremental ES differs from regular ES and GA because the search space is divided into smaller parts and evolved separately (Torresen, 1998) ,(De Jong & Potter, 1995). By gradually evolving each task in series increased complexity can be achieved (Floreano & Mondada, 1998), (Arakawa & Fukuda, 1996). The first incremental approach was to first evaluate the leg position bits, with fixed pause lengths. After obtaining gaits with sufficient fitness the leg position bits are fixed and the pause bits are evolved separately. From Fig. 8 we see that this approach is not successful as the fitness is never found to be higher than the fitness provided by simple ES. Leg position bits are evolved up to generation 11 and pause bits are evolved from generation 12. The next incremental approach was to divide the search in to 7 increments. First the leg position bits were evolved, then the most significant pause bits were evolved, then the next most significant pause bits were evolved until the least significant pause bits were evolved in the last increment. Even this approach was not found to provide better results than simple ES.

4.5 The ESBH Algorithm

The third and more successful incremental approach was to combine ES and binary hill climbing (BH) in the ESBH algorithm. From Fig. 7 we notice that the fitness landscape is smooth with few maxima. In a practical application disturbances will be added to this landscape due to variable foot friction, vibrations, variable air pressure and pause length deviations caused by non-ideal real time behavior of the operating system. However, the main characteristic of this landscape indicates that a hill climbing algorithm may be more efficient than a ES based search. In the ESBH algorithm the leg position bits are first evolved by evolutionary strategies up to generation 8. All pause length bits are fixed corresponding to pause lengths of 150 ms.

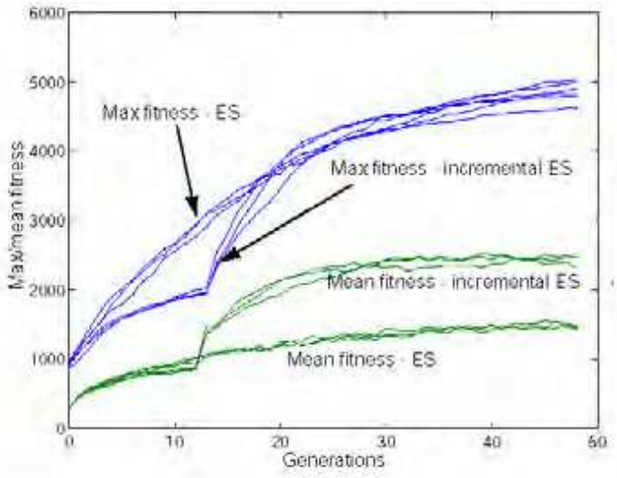

Figure 8. Fitness development for simple ES and incremental ES

In generation 8 ES have normally found a decent leg position pattern. From generation 9 all leg position bits ($C^x{}_1C^x{}_2C^x{}_3C^x{}_4$) are fixed. In generation 9 all possible combinations of the most significant pause length bits are tested (coarse search) where all other bits are kept fixed. With 3 pauses in a chromosome there are 8 possible combinations of the most significant pause bits to be tested. The chromosome with the highest fitness containing the most successful most significant pause bits is kept. 8 copies of this chromosome are then made forming generation 10. In generation 10 all combinations of the next most significant pause bits are tested keeping the other bits fixed. The chromosome with the highest fitness containing the most successful next most significant pause bits are then kept. 8 copies of this chromosome are then made forming generation 11 and so on until the least significant pause bits are found in generation 14. The search is then terminated. In this way the search space given by pause lengths is searched in a coarse to fine sequence.

To fully understand the operation of the binary hill climbing algorithm one may look at a simplification where the pause in gene 3 is kept constant and the algorithm is applied only to the pauses in gene no.1 and gene no.2. When the pause bits $P_1{}^1$and $P_1{}^2$ are varied and the rest of the pause bits are fixed at 0, there are 4 different pause combinations. This is illustrated in Fig. 10 where the four corners of the largest square represent all four pause

combinations. Suppose that the algorithm evaluate the fitness of all 4 corners in the largest square and selects the combination $P_1^1 = 1$ and $P_1^2 = 0$. In the figure this is illustrated by point A. When $P_1^1 = 1$ and $P_1^2 = 0$ and the pause bits P_2^1 and P_2^2 are varied where the rest of the pause bits are fixed at 0, there are 4 new pause combinations illustrated by the four corners of the next largest square in the figure. Suppose that the algorithm evaluate the fitness of all these 4 corners and selects the combination $P_1^1 = 1$, $P_1^2 = 0$ and $P_2^1 = 1$, $P_2^2 = 1$. In the figure this is illustrated by point B. By proceeding with less significant pause bits the algorithm continues to evaluate new squares where each side is half the size of the previous, hence the name "binary hill climbing". In Fig. 11 the ESBH algorithm is compared to simple ES and stochastic search.

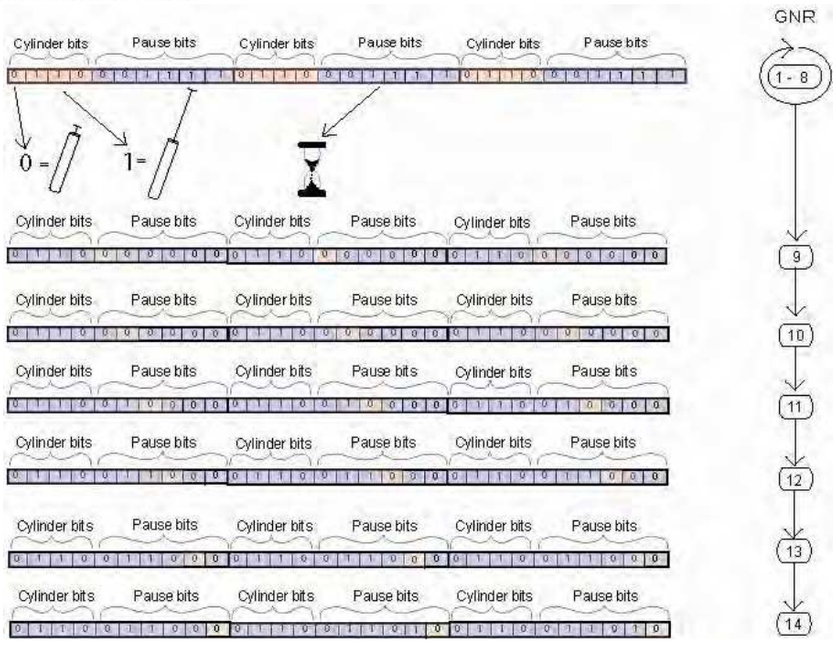

Figure 9. The ESBH algorithm

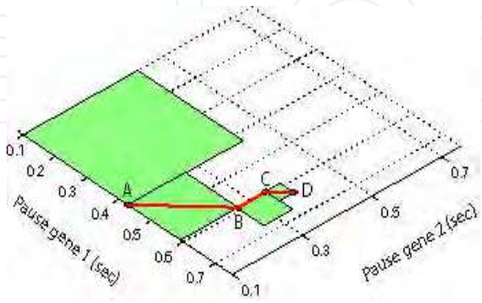

Figure 10. An incremental approach

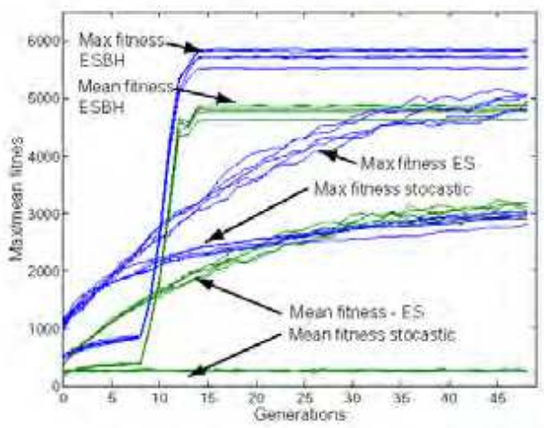

Figure 11. Fitness developments for the different approaches

As each graph represents the average fitness development over 1000 simulations, we see that the ESBH algorithm is in average superior to the others in this application where the focus is fast learning in less than 20 generations. A possible objection to the proposed ESBH algorithm is that heavy noise in the fitness calculations may cause the algorithm to derail and search in a non optimal region of the search space. To make the algorithm more robust an improvement could therefore be to let the algorithm run each increment over more than 1 generation and select the optimal chromosome based on fitness averaging.

4.6 Gaits Obtained

The gaits obtained can be divided into three categories, two sub optimal gaits and one optimal gait. In Fig. 12-14 these gaits are illustrated. The optimal gaits were based on synchronous jumping where both legs are kicking at the same time. By kicking both feet at the same time the most power was available causing the longest jumps. Other suboptimal gaits were based on one-leg jumping or asymmetric jumping where one foot was slightly delayed with respect to the other.

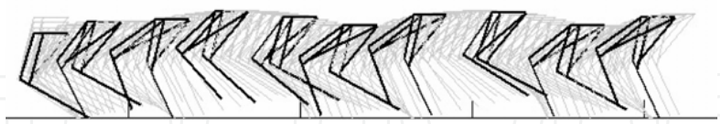

Figure 12. Illustration of a suboptimal gait based on asymmetric jumping. It is similar to the fastest horse gait called gallop

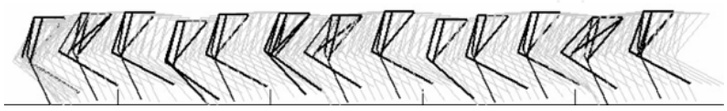

Figure 13. Illustration of a suboptimal gait based on every other one-leg jumping, similar to the movements made in the sport pole vault

Figure 14. Illustration of the optimal gaits based on jumping. This is the most efficient gait obtained, but in real life this gait has many drawbacks. E.g. the slippery effect in the floor to feet friction when the robot kicks hard

5. Practical Challenges

This section focuses on some of the practical challenges that arose while evolving directly on the robot. The first challenge was the foundation and floor in the laboratory. The floor was too hard for optimal robotic gait evolution and when the robot was expected to jump, it slipped. The robot became worn, due to the hard foundation and vibrations in the balance rod. As a solution, the robot was provided with rubber shoes, illustrated in Fig. 15.

Figure 15. The rubber shoes

The result was less tear and rod vibrations. Furthermore, the robot began to walk more springier, and started to evolve more efficient gaits based on jumping. The jumping-based gaits turned out to be the most effective. Due to sound propagation, a carpet was needed as a base underneath the robot. The carpet resulted in less noise, but again the slippery effect became an issue. This problem was solved by pasting sand paper underneath the rubber shoes. Other contributing factors were variations in the air pressure that influenced the performance and the real time qualities. The floor in the laboratory has a slight incline, resulting in a small variety in the fitness measure when evolving on the robot. These descriptions are some of the problems faced when evolving gaits on a real robot.

6. Measured Results

The ESBH algorithm has been tested on the pneumatic robot in an attempt to verify the theory. It was found very difficult to verify the theory accurately due to various practical side effects. One major problem was time consumption and mechanical wear out, particularly of the sandpaper shoe sole which affected the system significantly. When the robot moved, the whole system was vibrating heavily due to the quick contraction/ expansion movement of the pneumatic pistons. In Fig. 16 two typical fitness developments are shown for the ESBH algorithm. In these examples the binary hill climbing starting point was set to the 7th generation. From the measurements we notice an improvement in fitness after this point. However, the algorithm was found to produce proper gaits in less than 10 generations in almost all our experiments. From these few measurements it is difficult to conclude that the algorithm is working significantly better than simple GA in real life. The

only conclusion one can make so far from these measurements is that the algorithm itself is working quite well in this very noisy environment.

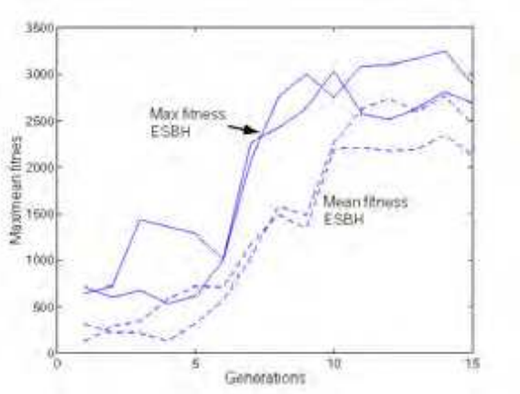

Figure 16. Measured fitness development for the ESBH algorithm

7. Conclusion

In this chapter an incremental search algorithm combining ES and binary hill climbing has been presented. The ESBH algorithm is compared to simple GA and ES, and stochastic search. We see that the ESBH algorithm is in average superior to the other approaches in this application where the focus is fast learning in less than 20 generations. A possible objection to the proposed ESBH algorithm is that heavy noise in the fitness calculations may cause the algorithm to derail and search in a non optimal region of the search space. Although various simulations has shown that the ESBH algorithm develop proper gaits significantly faster than standard GA/ES based algorithms, practical side effects in a physical environment, such as highly unpredictable shoe sole friction due to vibrations, varying pneumatic air pressure and wear out makes it difficult to prove that this algorithm is better than standard GA based algorithms. The algorithm itself, on the other hand was found to perform quite well in a very noisy environment.

8. Further Work

A possible improvement for future work could be to incorporate an “a priori knowledge library” of good patterns earlier evolved. One of the drawbacks in genetic algorithms and related programming methods are the possibility to end up in local optima without finding optima with higher fitness. A possible way to expand this work would have been to make a library of chromosomes that have been found favorable. If the ES gets stuck in local optima with low fitness, new chromosomes from the library could have replaced some of the chromosomes in the population. This routine can be a "control method" that runs in the background replacing individuals if the mean fitness does not exceed a certain level. There are great opportunities to further develop the ESBH algorithm as well. It could last for more generations by representing the pauses with more bits. This would of course make the search space larger as well.

9. References

Arakawa, T. & Fukuda, T. (1996) Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization, In: *Proceedings of the 1996 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1495–1500

Cant´u-Paz, E. (1998) A survey of parallel genetic algorithms. In: *Calculateurs Paralleles, Reseaux et Systems Repartis*, pp. 141–171, Paris

Floreano, D. & Mondada, F. (1998) Hardware solutions for evolutionary robotics. In: *Proceedings of the First European Workshop on Evolutionary Robotics*, pp. 137–151, Springer-Verlag, London

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Boston

Haddow, P. C. & Tufte, G. (2000) An evolvable hardware FPGA for adaptive hardware. In: *Proceedings of the2000 Congress on Evolutionary Computation*, pp. 553–560, IEEE Press, California

Higuchi, T.; Iwata, M.; Keymeulen, D.; Sakanashi, H.; Murakawa, M.; Kajitani, I.; Takahashi, E.; Toda, K.; Salami, N.; Kajihara, N. & Otsu, N. (1999) Real-world applications of analog and digital evolvable hardware, In: *IEEE Transactions on Evolutionary Computation*, pp. 220–235

Hornby, G.; Takamura, S.; Yokono, J.; Hanagata, O.; Yamamoto, T., & Fujita. M. (2000) Evolving robust gaits with Aibo, In: IEEE International Conference on Robotics and Automation**,** pp. 3040–3045

De Jong, K. & Potter, M. A. (1995) Evolving complex structures via cooperative coevolution, In: *Proceedings on the Fourth Annual Conference on Evolutionary Programming*, pp. 307–317, MIT Press, Cambridge

Kalganova, T. (2000) Bidirectional incremental evolution in extrinsic evolvable hardware, In: *Proceedings of the 2nd NASA/DoD workshop on Evolvable Hardware*, pp. 65–74, IEEE Computer Society , Washington DC

Parker, G. B. (2001) Evolving cyclic control for a hexapod robot performing area coverage, In: *Proceedings of the 2001 IEEE Computational Intelligence in Robotics and Automation*, pp. 555–560, Canada

Torresen, J. (1998) A divide-and-conquer approach to evolvable hardware, In: *Proceedings of the Second International Conference on Evolvable Systems*, pp. 57–65, Springer-Verlag, London

Torresen, J. (2004) An evolvable hardware tutorial, In: Proceedings of the 14th International Conference on Field Programmable Logic and Applications, pp. 821–830, Belgium

Bioinspiration and Robotics Walking and Climbing Robots
Edited by Maki K. Habib

ISBN 978-3-902613-15-8
Hard cover, 544 pages
Publisher I-Tech Education and Publishing
Published online 01, September, 2007
Published in print edition September, 2007

Nature has always been a source of inspiration and ideas for the robotics community. New solutions and technologies are required and hence this book is coming out to address and deal with the main challenges facing walking and climbing robots, and contributes with innovative solutions, designs, technologies and techniques. This book reports on the state of the art research and development findings and results. The content of the book has been structured into 5 technical research sections with total of 30 chapters written by well recognized researchers worldwide.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen