Morphological Evolution of Dynamic Structures in a 3-Dimensional Simulated Environment

Gary B. Parker (*Member, IEEE*), Dejan Duzevik, Andrey S. Anev, and Ramona Georgescu

Abstract— **The results presented in this paper are a part of the second phase of a body of research with the goal of co-evolving the mind and morphology of dynamic robots. We use a 3-Dimensional simulated gravitational environment to evolve LEGO structures. The focus of the current research is evolution of locomotion. The genetic algorithm uses a fitness function that encompasses the structure's movement, stability, and tension. Ten evolution tests were performed and all successfully yielded moving robots.**

I. Introduction

THE dominant research in evolutionary robotics is concentrated on the evolution of cognitive processes for robots with pre-defined morphologies. The logic behind these approaches is that evolutionary computation allows optimal development of robot's mind with minimal human influence. Yet, to confine the mind to a predetermined body is to create unintended constraints for the efficiency of a robot. Recently there has been a growing interest in the simultaneous co-evolution of the mind and the body of robots.

Some efforts targeted architectural aesthetics. O'Reilly incorporated evolutionary-strategy modules in CAD systems [1]. Her methods targeted partial transformations of pre-designed frames. The shortcoming of these attempts is the lack of an embedded fitness function, and the system's reliance on the feedback of the designers.

Karl Sims achieved significant virtual results creating a virtual environment to simultaneously evolve the morphology and controllers of creatures [2]. The evolution was successful in the simulated environment, yet his results have limited practical value, as real construction of his model seems implausible.

A team headed by Jordan Pollack at Brandeis University implemented a simulated gravitational environment to evolve static structures. The system used basic laws of physics and no human influence regarding physical characteristics to create motionless objects [3,4].

Henrik Lund developed dynamic robots using LEGO Mindstorms [5]. The system used pre-designed modules, simple structures composed of wheels, axles, and motors. The evolution combined these pre-designed parts to evolve a robot suited for a certain type of movement.

Manuscript received 31 Jan 2007.

G.B. Parker is the Director of Computer Science at Connecticut College, New London, CT, USA (e-mail: parker@conncoll.edu)

D. Duzevik, A. Andrey, and R. Georgescu are graduates from Connecticut College, New London, CT, USA.

Our research follows the logic of these approaches, but goes beyond their assumptions by introducing two novel fundamental concepts. First, we use a detailed chromosome structure that allows the system to exploit multiple connection points when building the structures, as well as during crossover and mutation. Second, we neither apply predetermined modules nor set bias towards specific connections. These two aspects of our system allow a greater degree of freedom and uninhibited evolution driven only by the designed fitness function. In addition, use of LEGO allows us to re-create the simulated structures from affordable material.

The result of this stage of the research is a system that evolves structures that successfully move in a straight line. The system creates diverse dynamic robots using pieces of different dimensions, shapes, and functions of a LEGO Mindstorms kit. The greatest initial hurdle in designing the system was to transpose the real LEGO pieces' properties in the simulated environment. The consequent challenge was to ensure a crossover of chromosomes that represent LEGO pieces in a virtual environment governed by the laws of physics. The final conundrum was the design of an intelligent fitness function that would enable efficient learning by the genetic algorithm. The solutions to these design problems are discussed throughout this paper.

The properties of the programmable control brick (RCX) enable downloading controls onto the robot itself, which creates an autonomous entity. With this research we show that pre-constructed modules are not necessary for evolution. We undertake an atomic approach where the fundamental constituent parts are pieces whose functions are not clearly defined until they are placed in the context of the entire unit. A list of frames that describe all available pieces allows easy construction of evolved structures.

Our research entails three stages: evolution of the morphology, evolution of the controller, and the combination of the two. The morphological evolution is broken down to two phases: creation of stable structures and the evolution of movable robots. During the first phase we used only bricks and successfully developed an algorithm that generated cohesive and stable tower-like structures [6]. This paper reports on the second phase of the morphological evolution that adds wheels, axles, and motors to the system. The goals of the future research phases will be an evolution of the controller and its incorporation with the evolution of the morphology described here.

II. Environment Simulation

In order to generate multiple generations that contain populations of 70 and more individuals, a building environment that would emulate the real world was necessary. Therefore, we developed a simulation environment that reflects the constraints of gravity and stress on the objects.

The program is composed of six interrelated modules (Figure 1): Builder, Physics Module, Locomotion, Stability, Fitness Evaluation, and Reproduction.

The Physics module evaluates the aggregate tension between all joints in the structure. If the structures created in the Builder break due to their own weight, the structure is returned to the Builder with the point of fracture. The Builder deletes the fractured part and sends the chromosome for re-evaluation to the Physics module. The process is reiterated until a rigid structure is obtained. The Locomotion module determines whether the individual is able to move, its speed, angle of movement, and friction, while the Stability module determines the structures stability. Individuals that show less inhibited movement and greater stability get better ranking during the Fitness Evaluation and are included in the pool for reproduction.

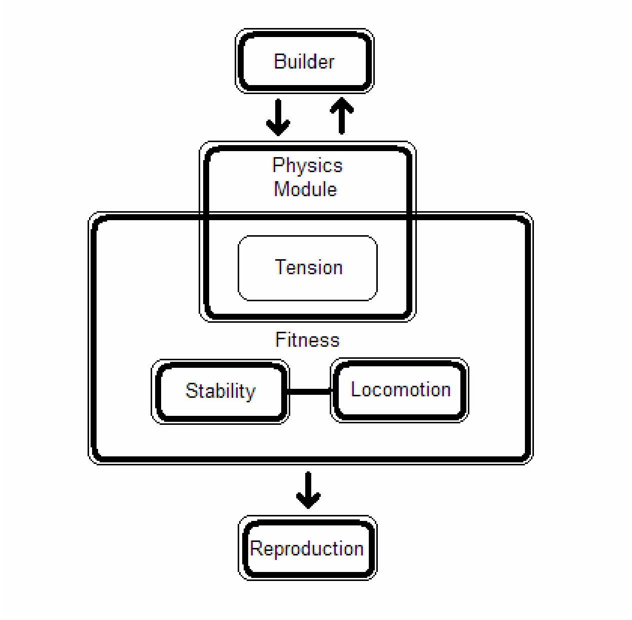


Fig. 1. Diagram of the System's Modules and their interaction.

A. Piece Representation

The constituent elements of our structures are pieces from a LEGO Mindstorms set. For all 139 different parts of the set we developed frames that represent their functional and physical properties. The frame specifies the type of the piece, spatial dimensions, weight, joint types, and joint coordinates. Considerably the heaviest and the largest piece of the pool is the RCX, only comparable to the motors and followed by the rest of the LEGO pieces. Bricks connect using knobs; wheels and axles connect through axle-ends and axle-holes; motors are bricks with an axle-extension, which allows connections with all pieces. These specific properties of the LEGO bricks led to interesting results as the system learned to balance the dynamic structures.

Every frame (an example is shown in Figure 2) plays the role of abstract knowledge; once used in a structure, it is altered to represent reality (only free joints are left and their positions are updated according to the coordinates of the piece in the builder). The system, however, can always reference the frame to retrieve the original properties of the piece. The length and width of the LEGO pieces are measured in units that represent the length of a one-knob, square piece. The height unit is one half of the length unit, according to LEGO standards. The weight of pieces is measured in grams. Each connection point coordinate shows its distance to the center of gravity of the structure in the x, y and z axis. The first free connection-point shown in Figure 2 is (3/2 -3/2 5), these three numbers represent the distances from the motor's center of gravity in the x, y, and z axis.

```
'((motor)
  (dimensions
    (length 5) (width 3) (height 10) (weight 0.04147))
  (connections
    (free
      (3/2 -3/2 5) (5/2 -3/2 5) ... ... (-5/2 1/2 -5) (-3/2 1/2 -5))
    (maleKnob
      (-5/2 -3/2 5) (-3/2 -3/2 5) ... ... (-1/2 3/2 5) (1/2 3/2 5))
    (femaleKnob
      (-5/2 -3/2 -5)(-3/2 -3/2 -5)... ...(-1/2 3/2 -5)(1/2 3/2 -5))
    (axel-end
      (3/2 -1/2 0) (5/2 -1/2 0) (3/2 -1/2 -1) (5/2 -1/2 -1))))
```

Fig. 2. A frame representing a motor. The actual frame of a motor has 32 coordinates representing possible connection points.

B. Builder Module

The Builder module is responsible for creating individuals by connecting pieces from the piece pool in the virtual building space. The building space is a three-dimensional pixel matrix, which provides computationally efficient bonding and intersection detection. Each pixel is labeled according to its current state: *free* (when not connected), *solid* (when unusable), or according to the existing connection: (*joint-type piece-reference*) or (*point-of-connection piece-reference*$_1$ *piece-reference*$_2$). As pieces are added the relevant pixels are inspected and if the joint is not free, placement is rejected. In the case of a free joint the new piece is examined for availability of a complementary joint at the coordinate. If the piece is placed all its connections are reflected not only in the builder but also in the chromosome of the unit. The builder is restricted to a finite piece-pool (for instance a single LEGO set) that determines the types and number of pieces available, consequently limiting the yielded structures to be comprised solely of pieces available to the user. The products of the Builder module are structures of interconnected pieces.

C. *Physics Module*

The Physics Module tests the overall integrity of the created structure. It distributes torques that the pieces exert on one another and evaluates the tension of the structure. Funes used a greedy generalized network flow algorithm [3,4] to evaluate the stability of a structure. Our system uses an algorithm that traverses the structure and classifies pieces as supports and supported in all connections.

The force that a single piece exerts is spread throughout its supports to the ground. The force may be decreasing or increasing the strength of the affected joints depending on the direction of the torque exerted. The possibility of fracture exists when the sum of forces applied by the weights of supported pieces prevails over the sum of the binding capacities of the knobs. Fractures do not exist in the structure if the capacities of all joints are larger than zero after the torques throughout the system have been applied to all pieces. In a case of a structural fissure, the broken part is removed and the remaining structure is re-evaluated. For a more detailed explanation of the algorithm that determines the cohesion of the individual refer to the description of the first phase of our research [6].

The initial capacities of the knobs and the weights of all pieces were derived experimentally. The final capacities of all joints are calculated for each individual by following a network of support links that ends with the ground. The result of this calculation is later used in the fitness function to represent the structure's tension.

D. *Stability Module*

The Stability Module evaluates the stability as defined by the ground supports of a structure. In contrast with the Physics Module, which assesses the overall sturdiness of the structure, the Stability module is concerned only with its balance. An algorithm finds the outer edges of the supports on the ground. The distance between a projection of the structure's center of gravity (COG) and the peripheral lines formed by the edges of pieces located on the ground level is used to determine the degree of stability of a given individual. The further the COG is from the support edges towards the center, the more stable the structure, and hence the better the fitness score. If the COG falls out of the boundary of the ground supports, the structure is determined to be unstable. The centralization of the COG is inversely adjusted for the height; higher structures call for better ground stability. This aspect of the stability module is crucial for the dynamic structures described in this paper, as movement requires even greater stability.

E. *Locomotion Module*

The Locomotion Module analyses three properties of the structure, which are used in the fitness function: direction, speed, and angle of movement. In our initial considerations we intended to allow movement to originate from any piece that is connected to a motor and that can touch the ground when rotated. The bricks used in this part of the research have axle-holes along their sides, through which axles can be attached. This would enable primitive movement ("flopping") even when wheels are not present in the system; yet, the weight of the fundamental part, the RCX, is to great for the force of a LEGO motor to overcome it using a brick on an axle. Therefore, movement is only possible when a ground wheel is connected to a motor.

The first task of the Locomotion Module is to assess if a combination of a motor-wheel-ground connections exist in the structure. Throughout this paper the wheels of this connection are referred to as *active wheels*. If no active wheels exist, the structure is deemed static, even if the structure has wheels that touch the ground. If active wheels exist the system uses the following steps to estimate the structure's ability to move.

First, the module estimates the most optimal direction of movement. The line of movement is simply derived from the most common alignment of wheels on the ground. For example, two wheels aligned at 0 or 180 degrees and one wheel at 90 or 270 degrees would yield 0 or 180 as a line of movement. The third wheel would add to the tilt and the friction of the structure.

Next, the module chooses the better direction of movement. Since dynamic friction is greater when the touching points on the ground are in front of the wheels or the center of gravity of the moving objects, the module estimates the direction in which the structure faces less frontal friction. As the LEGO motors are programmable in two directions with equal speed, this estimation emulates decision making, in which a human designer would be engaged.

Once the direction is set, the module estimates the speed and the angle of movement of the structure. An active wheel exerts force on the structure and creates a certain speed. The speed is not increased when more wheels are connected to the same motor, but the speed increases nominally when two motors power one wheel, as commonly occurs in later stages of the evolution. Bricks on the ground exert friction that reduces the speed and affect the angle of movement of the structure. The friction exerted on the structure is calculated on both sides of its center of gravity; further, the module computes the positioning and the distance between the wheels and the bricks on ground, as well as the wheels facing a different direction. The module uses this data to deduce the angle, or the tilt, at which the structure moves. The two values, the speed and the angle, are the crucial factors determining the fitness value of the structure in combination with the tension and stability estimates.

III. Learning

We employ evolutionary computation in this research to find an optimal solution. In accordance with theories behind evolutionary methods our genetic algorithm is designed to preserve and promote beneficial traits in the individual and guarantee that the system is capable to overcome local peaks in its search for the best possible structure.

After successful completion of the algorithm that created tower-like structures [6] the goal of the current research is to alter the system to create moving robots. A population of 70 individuals was used for all runs. The evolution was allowed to run for 300 generations. Larger populations and longer evolutions only exerted a greater burden on processing memory and time, due to the complexity of the pieces and the nature of the crossover.

A. Chromosome Structure

Since our research entails several phases with changing goals of the evolution, we designed a genetic algorithm to handle learning at each step. The chromosome contains the constituent pieces. Each gene of an individual's chromosome represents a piece of the LEGO set and its connections with other pieces in the structure. The addition of wheels, axles, and motors to the piece-pool did not require changes to the chromosome structure, although the individuals adjusted to the changing tasks and grew in complexity.

We apply variable chromosome length, which does not constrain the actual size of the final product. All pieces that compose the structure are represented as a vector. The genotype of each entity describes the phenotype showing the type, positioning in space, physical properties, and the connections of the pieces. A piece representation in chromosome is shown in Figure 3.

```
(piece-name (space-coordinates piece-orientation)
   (free-joints
      (joint-type (coordinates) (coordinates) …)
      (joint-type (coordinates) …))
      …
   )
   (index-of-contiguous-piece
      (joint-coordinates) (joint-coordinates) …)
   (index-of-contiguous-piece
      (joint-coordinates) …)
      …
)
```

Fig. 3. The chromosome is made up of a list of pieces in the structure. The representation is in Scheme.

Seventy randomly created structures compose the initial population. The only constraint to the process is that the RCX is always a part of any structure, because an individual without the control-brick would be un-programmable. Pieces are chosen randomly from the piece pool. Each piece is assigned an arbitrary orientation and is connected to an arbitrary available connection to another piece of the structure. The binding can be unsuccessful if the piece is in conflict with another piece already in the structure; in such case the piece is returned to the piece pool and the process is repeated once again. Initially all individuals have the same size (number of pieces). The mating produces structures with variable size.

B. Fitness Evaluation

The goal of our genetic algorithm is to create a successful, uninhibited evolution of autonomous, moving structures. The quest for the most effective fitness calculation ended with an approach that exploits three main parameters: locomotion, stability, and tension.

Combining the three properties was challenging, mostly as a result of the LEGO pieces' specific characteristics. The great utility of the LEGO pieces was offset at this stage with the constraints that their design put on the system. For instance, the Mindstorms kit has only four types of LEGO wheels, three of which are much higher than any brick, including the RCX. Our initial tests were performed with all four types of wheels. The larger wheels increased the speed, and the system learned quickly to exploit exclusively the largest wheel. This posed problems, however, as the largest wheel would raise and tip the structure, leading to detrimental stability finesses. The possibility that two wheels would counterbalance each other in the early stage of the evolution is very low; hence, most structures that possessed ability to move did not survive after a couple of generations. To solve this problem we kept only the smallest wheel in the kit, which does not always tip the structure when used in the early generations.

The second issue was to adjust the stability and the locomotion modules in a way that maximizes the two properties, but prioritizes locomotion. In opposite cases, when stability prevailed over locomotion, the evolution would quickly favor stable and unmoving structures and disregard ones with potential locomotion. To avoid this problem, we favored the locomotion fitness over stability. The speed of an active wheel always outweighs the stability estimate of the structure. In this way, less stable, yet movable structures survive. Subsequent generations improve its stability.

The locomotion fitness is an estimate of the movement of the robot due to its configuration. This value increases as a learning system and adds pieces that are in contact with the ground and produce thrust (wheels on motors) and/or reduced drag (wheels alone, oriented in the direction of movement). As a result of the adjustments discussed in the previous paragraphs, each additional motor-wheel-ground connection creates a jump in the resulting fitness function. Therefore, when a new active wheel is added in the direction of movement the system preserves it regardless if the structure is unstable. The system amends this situation in the generations that follow when it maximizes stability and eventually creates stable, movable objects.

The stability fitness uses the center of gravity and the ground pieces returned by the stability module to assign a stability value for each structure. Structures with wider ground base and center of gravity close to the ground receive better stability. We use proportional symmetry to assign the same stability value to pieces with identical form, but different dimensions.

In addition to the stability and locomotion values, the tension of the structure calculated during the physics module affects the fitness function. Structures with less total tension between the joints of the piece yield a better fitness score; conversely, pieces that exert greater stress to the overall structure diminish its fitness. The tension is especially important for movable objects, as inertia and additional stresses due to locomotion increase the possibility of fissures among high-tension structures.

The individual's fitness is determined by adding the three values: locomotion, stability and tension. We put the first priority on locomotion and favor stability over tension. Stability has a natural advantage over tension, as maximum stability is set to 1000, while maximum tension to 200. Locomotion priority is achieved by setting the minimum value of an active wheel to always exceed the maximum stability value. In this way we ensure that even the slightest possibility of locomotion will always raise the fitness value more than a perfectly stable, but unmoving object. It's better to move and fall than to never move at all.

The elite (top 30%) of the population are preserved into the next generation. All individuals of the population participate in stochastic selection for reproduction. Individuals are chosen randomly, but the mating probability is larger for individuals with greater fitness. Once mating is complete the two structures are returned to the population, which gives them the possibility of being selected for mating once again. The offspring created through reproduction supply the remaining 70% of the population.

C. Crossover

The reproduction of individuals occurs during the crossover phase. Two parents (parent A and parent B) are chosen from the elite population of the previous generation. The parent with a greater fitness is biased to provide more genetic material. For our example we assume it is parent A. Crossover starts by following links between pieces of parent A and re-creates the structure in a new simulated environment up to the crossover point.

The system then chooses a piece on parent B and attempts to attach it to the crossover point. If pieces from parent A conflict with the piece, then the system attempts three solutions. It tries different orientations which might adjust the piece to its new location. If the piece intersects with another piece in all orientations, the system invokes mutation to make it fit. If mutation fails, the system looks for free connection points in the vicinity of the original crossover point.

When the three methods fail, forced crossover occurs and pieces from parent A are used again. The new piece from A becomes the new crossover point and the process is repeated until pieces from parent B are successfully attached to the initial pieces from parent A. The multiple attempts to crossover, and the growing intricacy of structures presents the greatest demand for processing power and time. The result of the crossover is a new structure that has qualities of both parents.

D. Mutation

Mutation occurs randomly or after failed crossover, when the system tries to avoid resorting to forced crossover. Initially the system uses a proximity table that indicates how close in structure different pieces are. The proximity table is determined by the parameters of the piece: length, width, height, and number of knobs. Even though selection of the mutated piece is random, those that are closer in structure to the original piece have higher probability of being chosen. Wheels are close to both bricks and axles so that bricks on an axle can mutate into wheels on axles. Similarly, axles connected to motors can mutate into wheels connected to motors. The mutation that goes the other way is often harmful for the individual, so the system favors mutation that creates more motor-wheel connections.

Three types of mutation can occur: structural, limited positional, and general positional. *Structural mutation* is invoked when the genetic operator is attempting to place a piece that is not in the piece pool (if all of such pieces had been used). The *limited positional* leads to small, random changes in a single piece's position and orientation. The *general positional* mutation exerts a domino effect on all subsequent pieces by shifting them according to the shift in the mutated piece, and in that way this type of mutation creates a global impact on the structure.

IV. Results

The goal of our tests was the evolution of movable structures. Because of the narrow definition of our objective, the fitness function was evaluated using three properties: stability, locomotion, and tension. Better stability, increased locomotion, and lower tension were, therefore, key factors to raise the fitness of an individual. Our evolution provided solutions by balancing the three properties.

We show the results of 5 trials done with a population of 70 individuals to demonstrate the learning progression in Figure 4. The mutation level was set to 15% but forced mutation may increase this probability. An elite group of 30% of the population (21 individuals) was chosen to reproduce and persist in the next generation. The curves in Figure 4 show the improvement of the top fitness over three hundred generations. In all cases, there was fast initial improvement that decreased with the generations. Improvements in stability and tension are responsible for the parts of the curves that show steady rise, while the locomotion fitness contributed to the sharp increases of the top fitness.

Figure 5 shows the progression of the robot evolved during Test 3 (one of the mid-level performers from Figure 4). The pictures show the robot from the line of movement and from its side after 100, 200, and 300 generations. The dark squares represent LEGO bricks and axles, lighter squares show motors, and circles depict wheels. The center of gravity of each structure is shown with a white circle. In the early generations, the system preserves movable structures even if they would tip after some initial movement.

After 100 generations, the system designs a movable structure with a very low stability (the robot would fall forward and to the side). In addition, at this point of evolution the angle of movement is such that the robot would progress to a side (often spin in circles) rather than proceed a straight line.

During the consequent one hundred generations the system increases the stability and straightens the angle of movement by placing active wheels on both sides of the center of gravity. Moreover, the system avoids forward tipping by distributing wheels along the side of the structure. Motors are second-heaviest elements, and as such, the system uses them to balance the overall structure even if they do not connect to wheels. As a result, most motors from the piece pool are used by the time the evolution reaches the final generation.

At generation 300, the system discards unnecessary elements, such as wheels that do not touch the ground and other extensions. The system further distributes wheels on sides of the robot, maximizing the fitness by strengthening speed, direction, tension, and stability.

V. Conclusions

We successfully created a system that evolves movable robots without the use of pre-designed modules, with minimum human input, and with the option of building the simulated robots as actual robots. The work reported in this paper is a step toward our goal of co-evolving morphologies and controllers for LEGO Mindstorms robots. The next step is the evolution of controllers for the robots produced in this phase of the research.

References

[1] O'Reilly U.M. and Ramachandran G. (1998) "A Preliminary Investigation of Evolution as a Form of Design Strategy." *Proceedings of the Sixth International Conference on Artificial*. pp. 443 - 447.

[2] Sims, K. (1994) "Evolving 3D Morphology and Behavior by Competition." *Artificial Life IV: Proceedings of the Sixth International Conference on Artificial Life.* pp. 28-39

[3] Funes, P. and Pollack, J. (1998) "Evolutionary Body Building: Adaptive physical designs for robots." *Artificial Life 4*. pp. 337-357.

[4] Funes, P. and Pollack, J. (1999) "Computer Evolution of Buildable Objects." *Evolutionary Design by Computers*. pp 387-403.

[5] Lund, H. H. (2001) "Co-evolving Control and Morphology with LEGO Robots." Hara and Pfeifer (eds.) *Morpho-functional Machines*, Springer-Verlag, Heidelberg.

[6] Parker, G. B., Anev, A. S., and Duzevik, D. (2003) "Evolving Towers in a 3-Dimensional Simulated Environment." *Proceedings of the 2003 Congress on Evolutionary Computation*. pp 1137-1144.

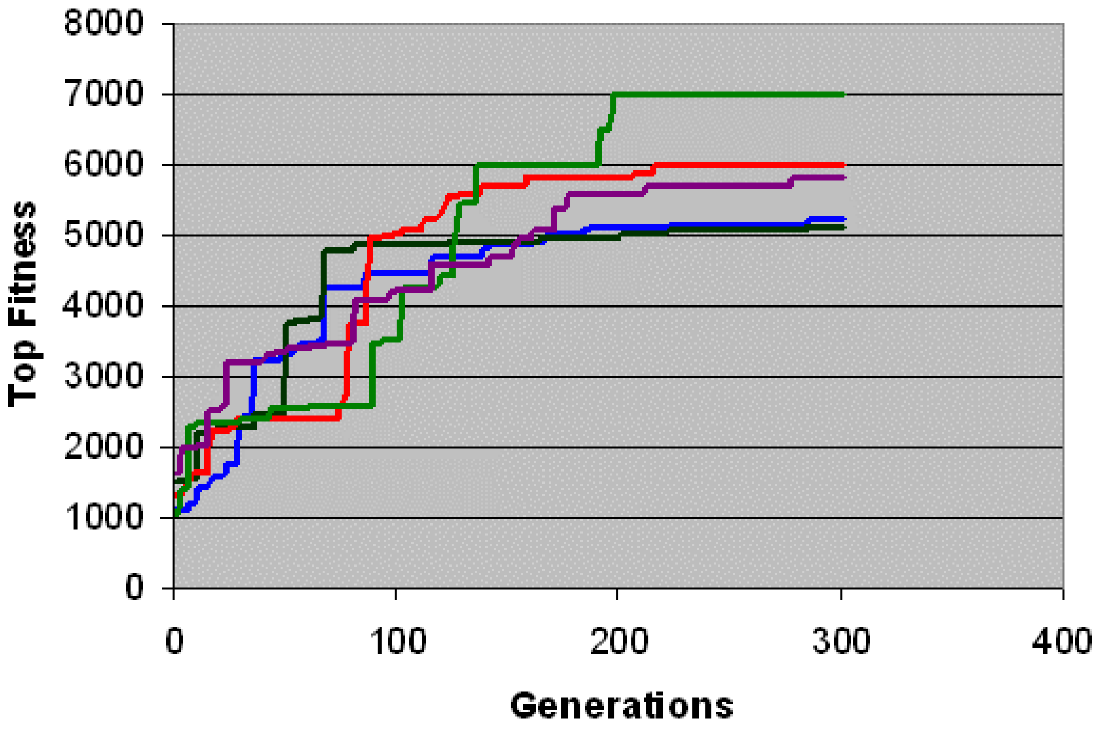
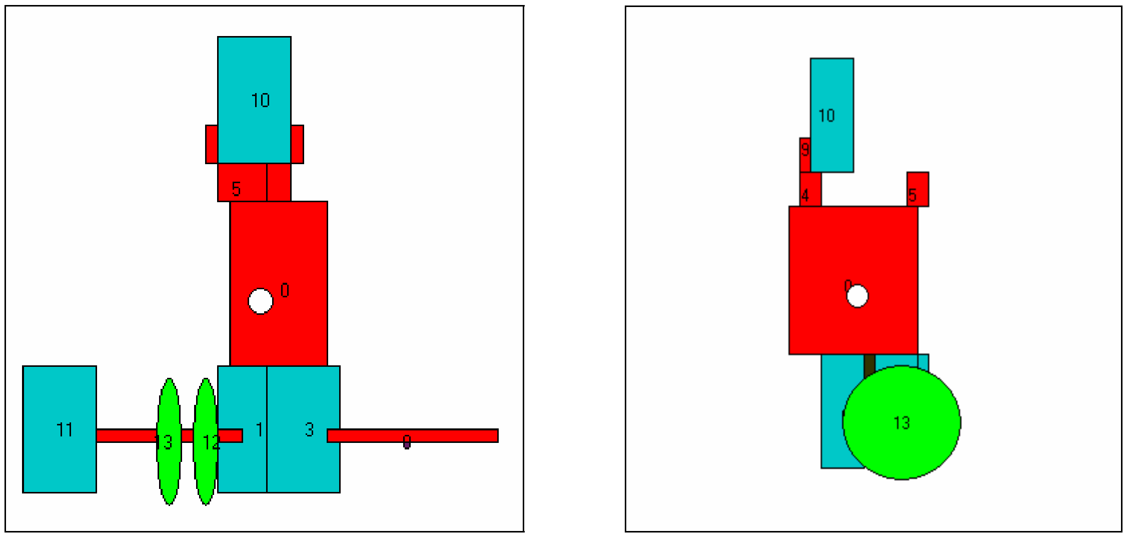
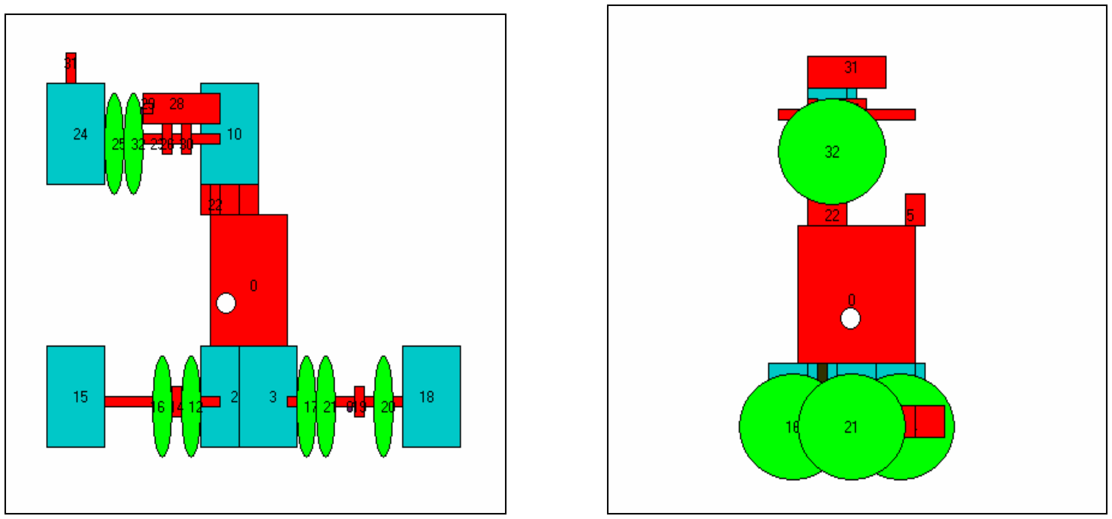


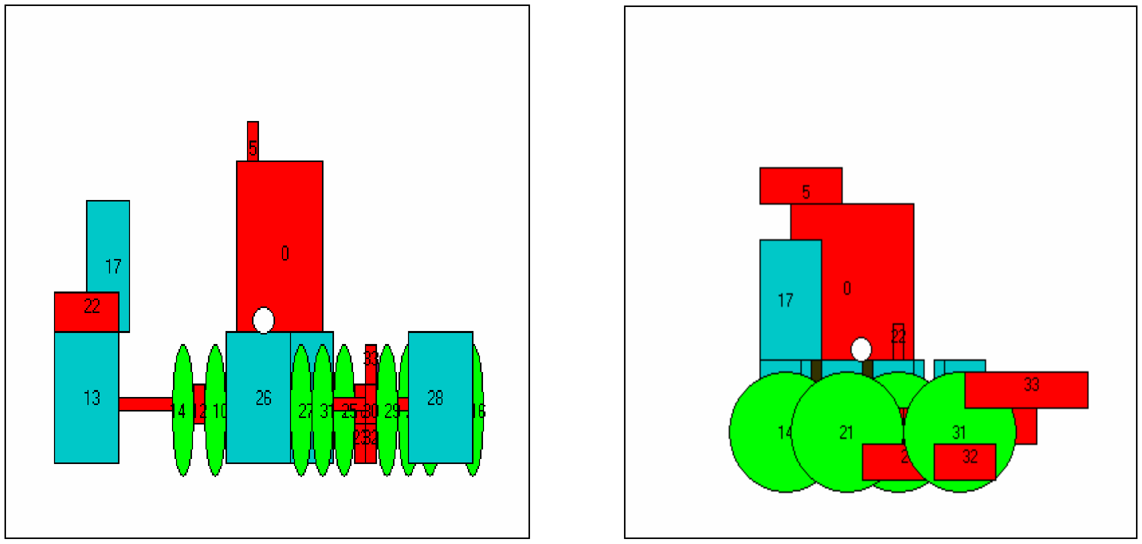
Fig. 4. Graph of the evolutionary progress of five tests using population sizes of 70 individuals, mutation rate of 15%, and 30% elite selection. The curves show the improvement of the top fitness over 300 generations. The stability fitness represents the gradual increases of the curves, while the locomotion fitness (formed largely by active wheels) causes the sudden jumps. The combination of the two fitness functions leads to the step-like curves.



After 100 generations



After 200 generations



After 300 generations

Fig. 5. Three stages of the evolution. The structures on the left show the best structures from their line of movement; the structures on the right show the side view. The dark squares are LEGO bricks and axles, light squares are the motors, and circles are wheels.