Genetic Transposition Inspired Incremental Genetic Programming for Efficient Coevolution of Locomotion and Sensing of Simulated Snake-like Robot

Tüze Kuyucu[1], Ivan Tanev[1], and Katsunori Shimohara[1]

[1]Department of Information Systems Design, Doshisha University, Kyotanabe, Japan
{tkuyucu, itanev}@mail.doshisha.ac.jp

Abstract

Genetic transposition (GT) is a process of moving sequences of DNA to different positions within the genome of a single cell. It is recognized that the transposons (the jumping genes) facilitate the evolution of increasingly complex forms of life by providing the creative playground for the mutation where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome. In this work we investigate the effect of a GT-inspired mechanism on the efficiency of genetic programming (GP) employed for coevolution of locomotion gaits and sensing of the simulated snake like robot (Snakebot). In the proposed approach, the task of coevolving the locomotion and the sensing morphology of Snakebot in a challenging environment is decomposed into two subtasks, implemented as two consecutive evolutionary stages. At first stage we employ GP to evolve a pool of simple, sensorless bots that are able to move fast in a smooth, open terrain. Then, during the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control and sensing in a more challenging environment. For the second phase the seed is used as it is to create only part of a new individual, and the rest of the new individual's genetic makeup is created by a mutant copy of the seed. Experimental results suggest that the proposed two-staged GT inspired incremental evolution contributes to significant increase in the efficiency of the evolution of fast moving and sensing Snakebots.

Introduction

Snake-like robots feature potential robustness characteristics beyond the capabilities of most wheeled and legged vehicles, such as: the ability to traverse challenging terrain and insignificant performance degradation when partial damage is inflicted. Some useful features of snake-like robots include smaller size of the cross-sectional areas, stability, ability to operate in difficult terrain, good traction, and complete sealing of the internal mechanisms (Dowling, 1999; Hirose, 1993). Moreover, due to the modularity of their design, the snake-like robots feature high redundancy and fault tolerance (Tanev et al. 2005). Robots with such properties can be valuable for applications that involve exploration, reconnaissance, medicine and inspection.

Designing a controller that can achieve optimal locomotion of a modular Snakebot is a challenging task due to the large number of degrees of freedom in the movement of segments of a Snakebot. The locomotion gait of such bots is often seen as an emergent property; observed at a higher level of consideration of complex, nonlinear, hierarchically organized systems, comprising many relatively simply-defined entities (morphological segments). In such complex systems the higher-level properties of the system and the lower-level properties of comprising entities cannot be directly induced from each other (Morowitz, 2002). Therefore even if an effective incorporation of sensing information in fast and robust locomotion gaits might emerge from intuitively defined sensing morphology and simple motion patterns of morphological segments, neither the degree of optimality of the developed code nor the way of how to incrementally improve this code is evident to the human designer (Koza et al. 2000). The previous research demonstrates that the control for a fast moving modular robotic organism could be automatically developed through various nature-inspired paradigms, based on models of learning and evolution. The earlier work demonstrates the use of GP (Koza, 1994) for evolution of sensorless sidewinding Snakebots in various environmental conditions (Tanev et al. 2005). Furthermore, the coevolution of active sensing and the control of the locomotion gaits of Snakebots was achieved (Tanev and Shimohara, 2008). The morphology of the sensors, attached to each of the segments of the bot, coevolve with the way to incorporate the sensory readings into the control of locomotion of the bot. The genetically optimized morphological traits of the bot include the initial orientation, the timing of switching on, and the range of the simulated laser range finders (LRF) attached to each of the segments of the bot. The emergent features of the evolved gaits include both the contact and contactless wall-following navigation accomplished via adaptive, sensory-controlled differential steering of the fast moving sidewinding bot. Despite the abovementioned evidence of the feasibility of coevolution of active sensing and the locomotion, the resulting wall-following behavior is achieved in an environment that is too simplified, and therefore too distant from the real-world applications: a simple curved corridor with a plain, smooth surface.

In this work we further investigate the coevolution of the active sensing and locomotion control of sidewinding Snakebot in a more complex environment that, in addition to a narrow corridor, features several large obstacles and many randomly placed small obstacles constituting a rugged terrain within this challenging environment. The sensors on the Snakebot used in this paper follow the same model as proposed in (Tanev and Shimohara, 2008): each segment of the Snakebot is provided with a fixed, immobile LRF with evolvable initial orientation, range and timing of firing. Thus the evolutionary task is not only to determine the time patterns of turning angles and the incorporation of sensor values for effective sensing and locomotion, but also to optimize the initial orientation, effective range and the timing of activation of module sensors. Hence, the Snakebot genotype is represented as a triple consisting of a linear chromosome containing the encoded values of the three relevant parameters of LRF, and two parse trees corresponding to the algebraic expressions of the temporal patterns of the desired turning angles in horizontal and vertical directions (further detailed in Section "Algorithmic Paradigm"). The most efficient locomotion gaits of Snakebot are not necessarily associated with the forward, rectilinear motions (and sidewinding might emerge as a fast and robust locomotion). Therefore, the eventual fusion of the readings of many sensors mounted in all the segments of the bot would provide Snakebot with the capability to perceive the features of surrounding environment along its whole body. In addition to the widening of the area of the perceived surroundings, multiple sensors offer the potential advantages of robustness to damage of some of them, dependability of the sensory information, and an ability to perceive the spatial features of the surrounding environment due to the motion parallax.

The poor scalability is a common problem in the simultaneous evolution of multiple features of simulated creatures, as the search space of evolution increases faster than linearly with the increase of the number of simultaneously evolved features. The considered case of Snakebot implies that the size of evolutionary search space can be seen as a multiplication of the sizes of the search spaces of the following interdependent evolutionary subtasks:

- Evolution of *control of locomotion*: the time patterns of turning angles of actuators that result in a fast locomotion of the bot,
- Evolution of the *morphology of the active sensing* – initial orientation of the sensors, their range, and timing of their activation, and
- Evolution of the *incorporation of the sensor signals* into the control of locomotion of the bot.

The large search space of the evolution of the considered Snakebot results in an intractable computational effort. Therefore, we propose an approach of decomposing the initially defined task into two subtasks, implemented as two consecutive evolutionary stages. As the first stage we employ GP to evolve a pool of simple, generic sensorless bots that are able to move fast in a smooth, plain terrain. During the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control, sensing morphology, and the method of incorporating the sensor signals into the locomotion of the bot in the given environment.

In this paper we propose an incremental evolution through the elaborated two stages, interfaced by a new approach to seeding. Inspired by genetic transposition (GT), we use the seed from the bots evolved during the first stage to create only a part of a new individual in the second stage. The rationale for proposing such an approach is based on the observations that the evolved fast moving Snakebots with sensory abilities exhibit some emergent locomotion traits that are pertinent to the generic, sensorless sidewinding locomotion (Tanev and Shimohara, 2008). We speculate that a better computational efficiency of evolution can be achieved if we first allow these generic features to evolve in sensorless bots moving in a smooth, plain terrain (with the task featuring a narrow evolutionary search space), and then–incorporating the genotypes of these bots into the evolution of the morphologically more complex bots (with sensors) in a challenging environment. The proposed mechanism of incorporation of these generic features of locomotion is based on seeding the initial population of GP (employed for the evolution of the bot with sensors) via the GT-inspired mechanism. Using the proposed mechanism of GT, the seed does not form the whole genome of an individual Snakebot, but only a part of it. We believe that, similar to the nature, the latter would offer the opportunity to preserve the genetic makeup of the generic locomotion features intact, while incrementally "upgrading" it with the new sensing abilities.

From another perspective, our work is inspired by the discoveries in the neurobiology suggesting that the complex navigation behaviors of species in nature can be achieved through an appropriate real-time modulation, controlled by the sensory inputs, of the generic neural signals produced by sensorless central pattern generators (CPG) (Levitan and Kazczmarek, 2002). Within this context, we would like to investigate whether (i) the separation of the genotype into two parts, mimicking the natural CPG and its modulation via sensory processing, respectively, and (ii) evolving these two parts in two consecutive stages would contribute to the improvement of the efficiency of evolution of the Snakebot.

In the remaining of this document we will provide a brief background related to GT, followed by a section elaborating on both the evolutionary and the experimental frameworks used in this paper. Next, we will discuss the obtained experimental results, and finally draw the conclusions of the work presented and detail future work.

Genetic Transposition in GP

Discovered by Barbara McClintock in maize (*Zea mays*), the *transposons* (jumping genes) are sequences of DNA that can move around to different positions within the genome of a single cell, in a mechanism called *transposition* (McClintock, 1950). In the process, they can cause mutations and change the amount of DNA in the genome. It is recognized that the transposons, facilitate the evolution of increasingly complex forms of life by providing the creative playground for fast mutations where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome (Nowacki et al. 2009; Strand and McDonald, 1985).

The related transposition-inspired research in evolutionary computation (EC) started by the work of Simoes and Costa (Simoes and Costa, 1999; Simoes and Costa, 2000) on the favorable effect of transposition on the performance of genetic algorithms (GA). The first of their methods is intended to enhance the crossover operation in GA by exchanging only the genetic material that is specifically marked as a transposon (Simoes and Costa, 1999). Their second approach, (termed "asexual transposition") models the mutation of GA as a "cut and paste" operation observed in biological GT (Simoes and Costa, 2000). Chan et al demonstrate a successful implementation of a GT inspired mechanism in multi-objective optimization, which is shown to have superior performance in achieving pareto optimal solution in comparison to multi-objective optimization without the GT mechanism (Chan et al, 2008). Liu et al employ a similar GT inspired mechanism in a clonal selection algorithm, which is shown to provide improved performance in automatic clustering problem (Liu et al, 2009). In a related research, McGregor and Harvey use a mechanism similar to transposition which they termed as "plagiarism" (McGregor and Harvey, 2005). The "plagiarism" copies one part of the genotype into another, replacing the latter completely. The authors demonstrated that the proposed mechanism improves the performance of the evolution of solutions to the Boolean logic problems. Spirov et al. also develop an original implementation of artificial transposition, used as a form of mutation operator for the simulated evolution of evolving a finite state machine as a solver of the artificial ant problem (Spirov et al. 2009).

In these aforementioned works, as well as in the biology, GT can occur frequently during the evolutionary cycle (just like other common evolutionary operations, such as crossover). In the approach we propose, however, GT occurs only once for each "seeding phase" (which, in turn, is only once per evolutionary run – at the stage of creating the initial population), and not invoked during the evolutionary run. Therefore, although the source of inspiration is the same, the implementation of the proposed model differs significantly from the previously developed GT-inspired mechanisms in EC. However, for the rest of the paper, we will refer to the GT-inspired mechanism introduced here as genetic transposition (GT) for simplicity and succinctness.

In our work we are especially interested in achieving higher efficiency in GP for coevolution of locomotion gaits and sensing of the simulated Snakebot. At the initial stage of the proposed approach, we evolve a pool of generic fast-moving sidewinding bots in a flat, smooth terrain. Then, during the second stage, we use these Snakebots to seed the initial population of the bots that are further subjected to coevolution of their locomotion control and sensing in a more challenging environment. During the seeding process the generic, fast moving, sensorless bots are subjected to genetic retro-transposition (i.e., duplicated within the same genome). The resulting transposon (connected with the seeding genome via a randomly initialized "control gene") is subjected to 100% random mutation in order to allow for the incorporation of the sensing information into the locomotion control of the bot. The schematic parse tree of the genotype of Snakebot, created during the initialization of GP via the GT-inspired mechanism is illustrated in Figure 1.

Seeding of the initial population by means of including the previously evolved successful (or partially successful) solutions has been shown to be an effective way of improving the efficiency of simulated evolution. For example, Nolfi et al. (Nolfi et al. 1994) evolve the controller of simulated robot and then re-evolve (or, adapt) the obtained results on real robots to accelerate the evolutionary process. Other examples of successful seeding include the work of Vassilev et al. (Vassilev et al. 2000) on the optimization of the existing digital circuit design; Thomsen et al. (Thomsen et al. 2002) on the use of solution obtained from a domain-neutral algorithm as a seed to evolve an even better performing solution; Langdon et al. (Langdon and Nordin, 2000) on seeding the evolutionary population with hand-coded solutions that allow a better generality of the evolved results.

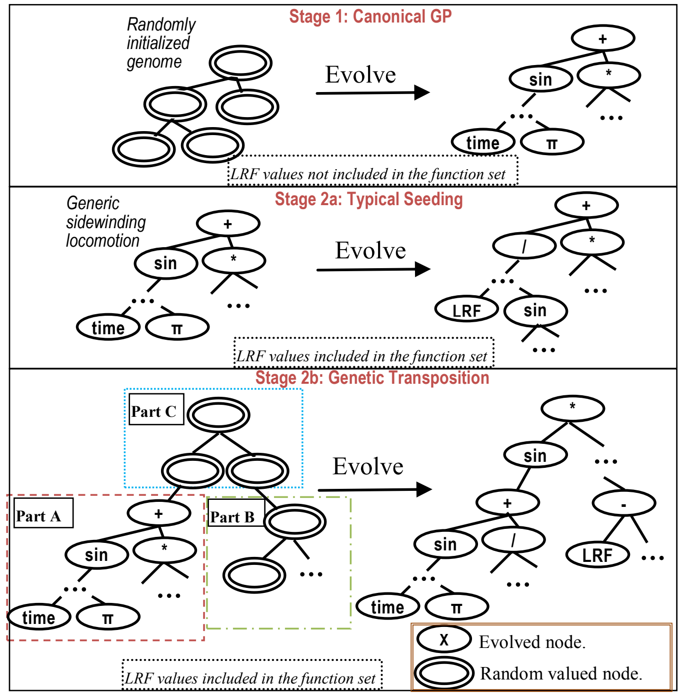

Figure 1: The mechanism of proposed genetic transposition in GP (Stage 2b) and the typical seeding process (Stage 2a). Both of these cases need to make use of a preliminary seed, and in the proposed approach this seed comes from a previously evolved sensorless Snakebot (Stage 1) that achieves fast locomotion on a smooth open terrain. In either of the Stages 2a and 2b, the resulting genome from Stage 1 is used as a seeding individual and further evolved, with additional sensory abilities (illustrated by the terminal symbol LRF) in a more challenging terrain. For Stage 2a the seed from Stage 1 makes up the whole genome of the initial Snakebot. For Stage 2b, the seed from Stage 1 is only a part (Part A) of the initial genome of the Snakebot. The rest of it contains a clone of the seed that has gone 100% mutation (Part B), and a randomly initialized group of control gene (Part C) which connects Parts A and B.

In addition, by utilizing the previously evolved solutions, seeding has also been applied to improve the performance of evolution of solutions from scratch. This technique, termed by Perry as "population enrichment" (Perry, 1994), has been demonstrated to be more efficient in discovering solutions in GP. "Population enrichment" is a form of seeding that is closest to the GT technique described in this paper. The main difference in these methods is the form of initialization, where in the "population enrichment" the seed is used to create the complete individual (see Stage 2a in Figure 1), while in GT the seeded genotype only forms a part of the genetic makeup of the newly created individual in the initial evolutionary population (see Stage 2b in Figure 1).

Evolutionary Framework and The Simulation Environment

In the experiments presented in this work we employed a DOM/XML-based implementation of GP (Tanev, 2004). The benefits of representing the genetic programs as DOM-parse tree featuring text-based XML-representation of genetic programs are (i) fast prototyping of GP by using standard built-in API of DOM-parsers for traversing and manipulating genetic programs, (ii) generic support for the representation of grammar of strongly-typed GP using W3C-standardized XML-schema; and (iii) human-friendly, text-based representation of the evolved solutions.

Representation of the Snakebot

We employ open dynamics engine (ODE) as a simulation platform for the Snakebot. ODE is a free, industrial quality software library for simulating articulated rigid body dynamics (Smith, 2004). It is fast, flexible and robust, and it has built-in collision detection. Therefore, ODE is suitable for a realistic simulation of the physics of an entire Snakebot when applying actuating forces to its segments. The ODE related parameters of the simulated Snakebot are same as elaborated in (Tanev et al. 2005).

Snakebot is simulated in ODE as a set of 15 identical spherical morphological segments (“vertebrae"), linked together via universal (Cardan) joints (Figure 2). All joints feature identical angle limits and each joint has two attached actuators (“muscles"). A single LRF sensor, with a limited range is rigidly attached to each of the segments.

The functionality of the LRF can be defined by the values of the following set of parameters: (i) orientation, measured as an angle between the longitudinal axis of the sensor and the horizontal axis of the joint, (ii) range of the sensor (in cm), and (iii) the timing of activation, expressed as a threshold value of the turning angle of the horizontal actuator. The reading of LRF is a scalar value which corresponds inversely to the distance between the sensor and an object (if any within the sensor's range), measured along the longitudinal axis of the LRF. In the initial standstill position of Snakebot the rotation axes of the actuators are oriented vertically (vertical actuator) and horizontally (horizontal actuator) and perform rotation of the joint in the horizontal and vertical planes respectively.

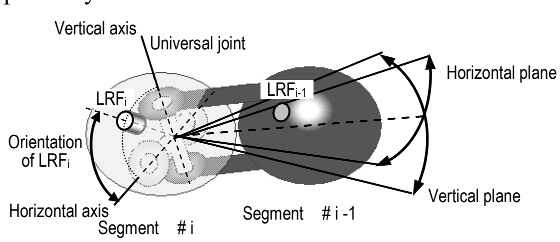

Figure 2: Horizontal and vertical actuators attached to the joint perform rotation of the segment #i-1 in vertical and horizontal planes respectively.

Considering the representation of Snakebot, the task of designing the fastest locomotion can be rephrased as developing temporal patterns of desired turning angles of horizontal and vertical actuators of each segment that result in fastest overall locomotion of Snakebot. The proposed representation of Snakebot as a homogeneous system comprising identical morphological segments is intended to significantly reduce the size of the search space of the GP. Since the size of the search space does not necessarily increase with the number of morphological segments of the Snakebot, the proposed approach offers a favorable scalability.

Algorithmic Paradigm

For the evolution of the Snakebot, the genotype is represented as a triple consisting of a linear chromosome containing the encoded values of the three relevant parameters of LRF, and two parse trees corresponding to the algebraic expressions of the temporal patterns of the desired turning angles of both the horizontal and vertical actuators, respectively (Figure 3).

The encoding of the parameters of LRF is as elaborated in Figure 3. The same figure also illustrates the function set and the terminal set of the GP, employed to evolve the control sequences of both actuators. Because the locomotion gaits by definition are periodical, the periodic functions sine and cosine are included in the function set of GP in addition to the basic algebraic functions. Terminal symbols include the variables time, segment_ID, an automatically-defined function (ADF), the reading of the sensor (LRF), and two constants: Pi, and a random constant within the range [0, 2]. The incorporation of the terminal symbol segment_ID (a unique index of morphological segments of Snakebot) allows GP to discover how to specialize (by phase, amplitude, frequency etc.) the
genetically identical motion patterns of actuators of each of the morphological segments of the Snakebot.

The rationale of employing ADFs is based on the observation that the evolvability of straightforward, independent encoding of desired turning angles of both horizontal and vertical actuators is rather poor. Even without ADFs, GP is able to adequately explore the potentially large search space and ultimately discover the areas that correspond to fast locomotion gaits in the solution space. However, it was observed in the previous work of Tanev et al (Tanev et al. 2005) that not only the motion patterns of adjacent segments are correlated, but the motion patterns of horizontal and vertical actuators of each segment in fast locomotion gaits are highly correlated too. Moreover, discovering and preserving such correlation by GP is associated with enormous computational effort. ADFs, which provide a way of introducing modularity and reuse of code in GP (Koza, 1994), are employed in our approach to allow GP to explicitly evolve the correlation between motion patterns of horizontal and vertical actuators as shared fragments in algebraic expressions of desired turning angles of both actuators. Furthermore, we observed that the best results are obtained by; (i) allowing the use of ADF as a terminal symbol in algebraic expression of desired turning angle of vertical actuator only, and (ii) evaluating the value of ADF by equalizing it to the value of currently evaluated algebraic expression of desired turning angle of horizontal actuator. The main GP (hence the EA) parameters are summarized in Table 1.

Genetic Operations. We employ a binary tournament selection and a single point crossover. The crossover point is randomly selected between the three components of the genotype (as shown in Figure 3). The mutation randomly alters either a value of an allele in the linear chromosome representing the parameters of LRF, or a sub-tree in one of the two parse tress that correspond to the temporal patterns of the control sequences of actuators.

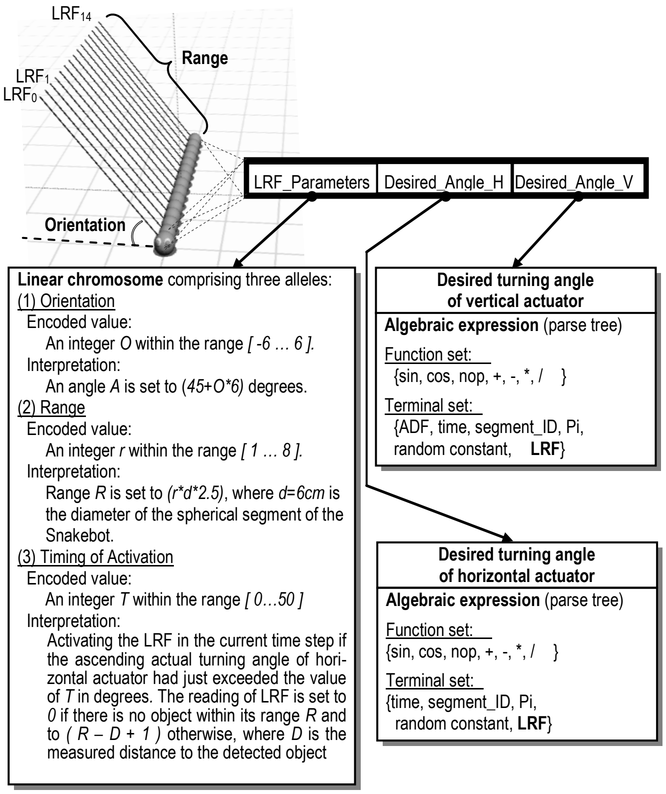

Figure 3: Genotype of the Snakebot, represented as a triple containing the values of the parameters of LRF and two algebraic expressions of the temporal patterns of the desired turning angles of horizontal and vertical actuators, respectively.

Fitness Evaluation. The fitness function is based on the average velocity of Snakebot, which is estimated from the distance traveled during the trial. As we shall elaborate later in the "Experimental Setup" section, the confined environment used in the trial is a narrow corridor covered with obstacles of various sizes (Figure 4). The velocity of locomotion needed to clear the final obstacles towards the end of the corridor for the given time of the trial (16s) corresponds to a fitness value of 100. The evolution is terminated if the bot reaches the fitness of more than 120 (fitness required to clear the whole corridor) or if the maximum accumulative number of 80 generations is reached. 80 generations was set as the cumulative maximum as a result of the experience from earlier experiments. Earlier experiments used in achieving locomotion of modular Snakebot had used 40 generations, which was a sufficient limit for the evolution of locomotion. Ideally, addition of a new feature should not require a much larger computational effort. Therefore, 40 generations per individual feature of the Snakebot was decided to be an acceptable cost.

Experimental Cases. In order to investigate comparatively the efficiency of proposed approach, we used three methods to evolve the locomotion of Snakebot with sensors:

Category	Value
Population Size	200
Selection	Binary Selection ratio: 0.1 Reproduction ratio: 0.9
Elitism	4
Mutation Rate	1%
Trial Interval	16s (400 time steps of 40 ms per step)
Termination Criterion	(Fitness=120) or (Num. of Generations=80)

Table 1: The GP-related parameters.

I. *Canonical GP* (single stage approach): In this case the evolution of the Snakebot is done from scratch; i.e. evolution starts with a population of randomly created individuals and optimizes these individuals to satisfy the target fitness. The limit of the evolutionary generations of GP is set to 80.

II. *Typical seeding* (two-staged approach): The genotypes of six best sensorless Snakebots that have already been evolved to achieve fast sidewinding locomotion in a plain, smooth terrain (Figure 1, Stage 1), is used to create the initial population. This evolved genotype is used as an elite individual to seed the initial population, where the exact copies of these six sensorless bots are used to form a small part (6 bots) of the initial evolutionary population. The remaining part of the population (194 bots) is randomly generated. This seeded population is then evolved to fully satisfy the target fitness (Figure 1, Stage 2a). The limit of the generations of both stages of evolution is set to 40.

III. *GT* (two-staged approach): The first stage of the proposed approach is identical to that of the typical seeding method. Similarly, the six best sensorless genotypes are used as elite individuals in the initial population of the second evolutionary stage. To create the remaining 194 bots of the initial population, however, we use the evolved best sensorless genotypes to form only part of these newly created individuals. The remaining parts of these individuals are created randomly, as elaborated in the section titled "Genetic Transposition in GP". These 194 partially seeded and partially random individuals and the six fully seeded individuals are used to form the initial evolutionary population (Figure 1, Stage 2b). The created population is evolved to fully satisfy the target fitness. Similar to the typical seeding, the limit of the generations of both stages of evolution is set to 40.

Experimental Setup

The experimental environment (Figure 4a and 4b) is formed of a straight narrow corridor (the width is the same as the length of the Snakebot) that has two groups of tall boxes that protrude to about 40% of the width of the corridor. In addition, part of the corridor is covered by many, randomly located and sized, small boxes that are designed to create a rough terrain and noisy environment for the sensors. The length of the corridor is set to seven times the length of the Snakebot. Starting from one end of the corridor, the aim of the bot is to reach the other end within the given time-span. We designed this environment with the intention to encourage the evolving bot to develop the following abilities: (i) fast locomotion (long enough corridor), that is (ii) not hindered by rugged terrain (small boxes), (iii) following of obstacles that cannot be overcome (walls), and (iv) circumnavigating obstacles that cannot be overcome (tall boxes).

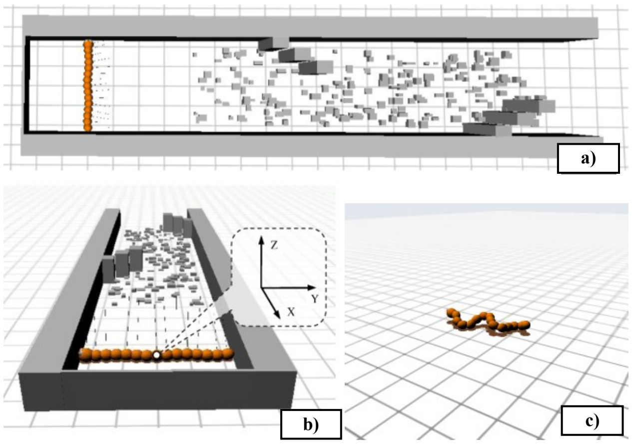

Figure 4: The experimental setup of the scenes.

For experimental case I and Stage 2 of experimental cases II and III, the Snakebot is initialized with 15 modules, on full stretch at the dead end of the corridor with its longitudinal axis perpendicular to the intended direction of movement (Figure 4a, and 4b). Initially, the rough terrain is not present to facilitate the evolution of basic locomotion on smooth terrain. After a fitness value of 60 is reached (i.e. the first set of large obstacles can be cleared by the Snakebot), a large portion of the corridor is filled with randomly initialized boxes (random size and location). The initial orientation of the Snakebot and the corridor is influenced by the previous work suggesting that sidewinding is the fastest and most robust locomotion gait for a Snakebot. Therefore, the Snakebot would be expected to enter a corridor featuring a similar orientation.

For Stage 1 of experimental cases II and III, a plain surface with no obstacles is used as the environment (Figure 4c), and the LRF is excluded from the GP function-set.

Experimental Results

The Snakebot is evolved applying the three different evolutionary approaches as described in the "Experimental Cases" section, and under the experimental conditions as outlined in the "Experimental Setup" section. For each approach we executed 38 independent runs. The fitness convergence characteristics of these runs are shown in Figure 5. As Figure 5a depicts, the canonical GP features average fitness (over all independent runs) of about 40, which corresponds to the 40% of the length of the corridor, which also corresponds with the position of the first set of tall obstacles. The pace of the improvement of the fitness is rather slow, with average value of about 30 at generation 40. These results suggest that the bot is struggling to discover the generic locomotion gaits that can result in a fast enough locomotion even in the absence of obstacles. The large search space of the evolution, caused by the need to additionally evolve the sophisticated morphology sensing (LRF), and the way to properly incorporate the sensing signals into the locomotion control is one of the reasons for the poor efficiency. Another reason is the challenging environment–the walls and various obstacles, which implies that the fitness landscape of evolution features fewer (compared to the previously tested cases [Tanev et al. 2005; Tanev and Shimohara, 2008]) and narrower optimal areas. Indeed, even if fast locomotion emerges during the initial stages of evolution, its survival value could be easily "underestimated" by evolution because the bot gets stuck at the first obstacle. Hence the large difference between the progression of the results displayed in Figures 5a and 5b.

	Canonical	Seeding	GT
Average Fitness	43	69.3	91.2
Median Fitness	37	67	91
Std Dev. of Avg. Fitness	23.5	27.2	19.4
Runs with Fitness >100	1 (2.6%)	3 (7.9%)	8 (21%)

Table 2: Statistics of the experimental results.

Conversely, the results of the first stage of both the typical seeding and GT (Figure 5b) indicate that the evolution of the locomotion of a Snakebot is more efficient, when relieved from the burden of dealing with the sensors and the sophisticated environment. The velocity of 100, which would be sufficient to clear the obstacles, is now easily achievable within 10 to 36 generations.

Then, when six of these best moving generic bots are incorporated via typical seeding into the initial population in the second stage of evolution (Figure 5c), and allowed to evolve for additional 40 generations, the average fitness value is 1.6 times higher than the result obtained by canonical GP (Table 2). However, the best efficiency of evolution is achieved when GT is used–the average fitness is more than 90 with 8 successful runs, and a smaller deviation in the fitness values achieved (Figure 5d and Table 2).

The proposed approach of employing GT allows the evolution to experiment with the way of processing the sensory signals without the risk of damaging the already evolved, fast locomotion control. Therefore, the transposition could facilitate the protection of the already evolved beneficial building blocks from the destructive effects of genetic operations. Conversely, since the locomotion control comprises 100% of the genotype of the bots created via typical seeding, any incorporation of the sensing information as a result of the genetic operation would most likely result in damage of this control. Indeed, the genotypes of the successful results achieved via GP with GT, the resulting genotype always had a portion in the form of Equation 1. Equation 1 (with C1 and C2 being constants) is the general form of the controllers achieved via the evolution experiments for the locomotion of a Snakebot on a smooth, empty terrain, i.e. Figure 5b. The resulting genotypes were simply the modulation of these controllers via the LRF signals. On the other hand the successful solution achieved by canonical GP and typical seeding runs did not have an exclusive part of their genotype that resembles Equation 1; instead a large, complicated equation that is hard to comprehend was the result.

$$C1 * \sin(ID + time + C2) \qquad (1)$$

In fact, when re-run on maps with differently arranged obstacles (to that of the environment present during evolution), the most robust Snakebots are observed to be from the GP runs using GT. We believe that the following are the reasons for the significant improvement in the efficiency (computational effort) of evolution due to GT:

- A wider spread of the initial seed into the population (than the typical seeding) of genotype that features generic ability to move,
- A better value of the initial fitness of the bots as they already feature the generic ability to move in their genotypes, and
- A separation of the sensing and locomotion parts of the genome, which may create a more efficient control mechanism for the bot.

We would like to point out that latter of the above mentioned arguments might provide a further insight into the design of robotic control systems and their sensorimotor control. The locomotion property of the Snakebot can be viewed as a continuous process that needs to be applied regularly under normal conditions, and the sensing property of the Snakebot can be viewed as a reflex that only needs to affect the actions of the bot when an event occurs. Such a concept might be seen as analogous to the reactive behavior related to the reflexes observed in biological organisms. For example, the collision-free flight of locusts in a crowded swarm is recognized to be achieved by direct, real-time input of the sensory signals into the wings muscles. The latter serve as a mediator for both the (i) "default" oscillating signals (generated by CPG) and (ii) the visual sensors (Uvarov, 1977).

From another viewpoint, our results can be seen as an evidence of the computational benefits of mimicking the neurobiological concept of achieving complex navigation behaviors of species in nature through sensory-controlled modulation of CPG. The moving trajectory of a sample best of run bot (Figure 6) illustrates the emergence of the following abilities of the bot: (i) fast locomotion (clearing the corridor), that is (ii) not hindered by rugged terrain (overcoming small boxes), (iii) following obstacles that cannot be overcome (walls), and (iv) circumnavigating obstacles that cannot be overcome (two groups of tall boxes).

The successful Snakebots from the results presented demonstrate the incorporation of sensor information within the control mechanism of the Snakebots for steering the Snakebot away from the large obstacles. The evolved Snakebots use the sensor signals as repulsive forces on the individual modules, which gradually change the course of the whole Snakebot.

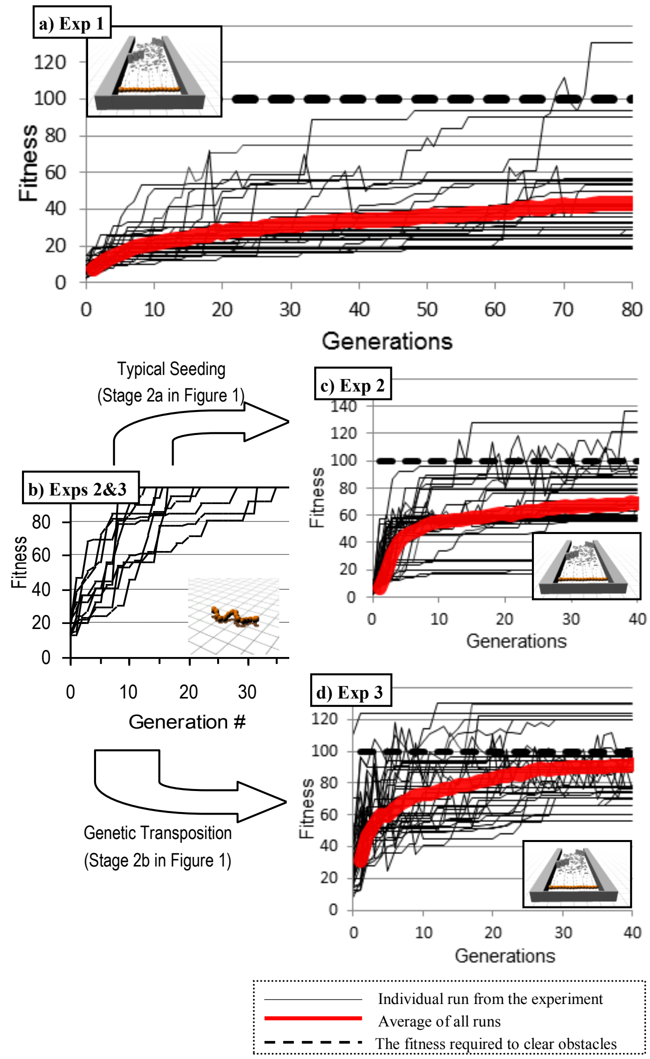

Figure 5: Fitness convergence characteristics of the three approaches used to evolve the Snakebot in the confined environment: single-staged canonical GP (a), and incremental two-staged typical seeding (b then c) and GT (b then d), respectively. The graphs show the fitness convergence of all 38 runs from each experiment.

Conclusions

We demonstrated that the evolution of a modular sidewinding Snakebot in a challenging environment with multiple forms of obstacles is a computationally demanding task. Dividing this task into two subtasks, implemented as two consecutive evolutionary stages, contributes to the significant improvement in the efficiency of evolution.

We introduced a genetic transposition inspired seeding technique to further improve both the quality of the bots and the computational effort required to evolve them. The proposed technique offers a significant improvement over typical seeding when applied to the evolution of an active sensing of fast moving Snakebot. The presented technique could be seen as a promising approach to incremental coevolution of multiple features of morphologically and behaviourally complex bots situated in challenging environments.

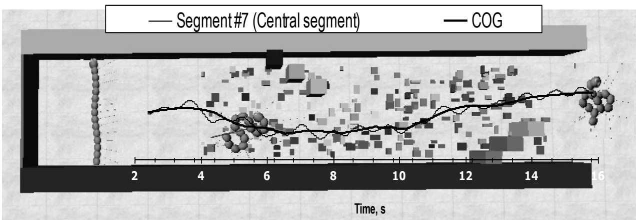

Figure 6: The moving trajectory of the central segment and the center of gravity (COG) of a sample best-of run Snakebot, evolved by incremental GP with GT.

In biology it is recognized that the transposons (the jumping genes) facilitate the evolution of increasingly complex forms of life by providing the creative playground for the mutation where the latter could experiment with developing novel genetic structures without the risk of damaging the already existing, well-functioning genome. The results shown in this paper demonstrate that this biological occurrence is also applicable to EC, and the proposed genetic transposition inspired seeding mechanism also facilitates the artificial evolution of increasingly complex systems.

As part of future work, we aim to analyze the Snakebots evolved in detail in order to gain an understanding of how the sensory signals are integrated with locomotion, and to infer the definition (understandable by human designers) of the control mechanism of the Snakebot. Furthermore, we plan on studying and designing mechanisms that can accompany genetic transposition in bringing a more efficient and robust evolution of complex robotic systems with multiple evolving features. Finally, we aim to generalize the proposed technique and define the properties of the tasks in evolutionary robotics that can be efficiently solved via this approach.

References

Chan, T.; Man, K.; Tang, K. & Kwong, S. (2008). A Jumping Gene Paradigm for Evolutionary Multiobjective Optimization Evolutionary Computation, IEEE Transactions on, 12, 143-159

Dowling, K. (1999) Limbless locomotion: learning to crawl In Robotics and Automation. Proceedings. 1999 IEEE International Conference on, volume 4, pages 3001 –3006 vol.4, 1999

Hirose, S. (1993) Biologically Inspired Robots: Snake-like Locomotors and Manipulators. Oxford University Press.

Koza, J. R. (1994) Genetic programming II: automatic discovery of reusable programs. MIT Press, Cambridge, MA, USA.

Koza, J. R., Keane, M.A., Yu J., Bennett, F.H., and Mydlowec W. (2000) Automatic creation of human-competitive programs and controllers by means of genetic programming. Genetic Programming and Evolvable Machines, 1:121–164. 10.1023/A: 1010076532029.

Langdon, W.B. and Nordin, P. (2000) Seeding genetic programming populations. In Proceedings of the European Conference on Genetic Programming, pages 304–315, London, UK. Springer-Verlag.

Levitan, I.B. and Kazczmarek, L.K. (2002). The Neuron—Cell and Molecular Biology. Oxford University Press.

Liu, R.; Sheng, Z. & Jiao, L. (2009). Gene Transposon Based Clonal Selection Algorithm for Clustering. Proceedings of the 11th Annual conference on Genetic and evolutionary computation, 1251-1258.

McClintock, B. (1950) The origin and behavior of mutable loci in maize. Proc Natl. Acad. Sci. USA, 36:344–355.

McGregor, S. and Harvey, I. (2005). Embracing plagiarism: Theoretical, biological and empirical justification for copy operators in genetic optimisation. Genetic Programming and Evolvable Machines, 6:407–420.

Morowitz, H.J. (2002). The Emergence of Everything: How the World Became Complex. Oxford University Press.

Nolfi, S., Floreano, D., Miglino, O. and Mondada, F. (1994). How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics. In 4th International Workshop on Artificial Life. MA: MIT Press, 1994.

Nowacki, M., Higgins, B.P., Maquilan, G.M., Swart, E.C., Doak, T.G., and Landweber, L.F. (2009). A functional role for transposases in a large eukaryotic genome. Science, 324 (5929): 935–938.

Perry, J. (1994). The effect of population enrichment in genetic programming. In Evolutionary Computation. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, pages 456 –461 vol.1, June 1994.

Simões, A. and Costa, E. (2000). Using genetic algorithms with asexual transposition. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'00), pages 323–330. Morgan Kaufmann.

Simões, A. and Costa, E. (1999). Transposition: A biologically inspired mechanism to use with genetic algorithms. In the Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms (ICANNGA'99), pages 612–619. Springer-Verlag.

Smith, R. (2004). Open Dynamics Engine.

Spirov, A.V., Kazansky, A.B., Zamdborg, L., Merelo, J.J., and Levchenko, V.F. (2009). Forced evolution in silico by artificial transposons and their genetic operators: The John Muir Ant Problem. Technical Report, Oct 2009.

Strand, D.J. and McDonald, J.F. (1985). Copia is transcriptionally responsive to environmental stress. Nucleic Acids Research, 13(12):4401–4410.

Tanev, I., Ray, T., and Buller, A. (2005). Automated evolutionary design, robustness and adaptation of sidewinding locomotion of simulated snake-like robot. IEEE Transactions on Robotics, 21:632–645.

Tanev, I. and Shimohara, K. (2008). Coevolution of active sensing and locomotion gaits of simulated snake-like robot. In Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO '08, pages 257–264, New York, NY, USA. ACM.

Tanev, I. (2004). Dom/xml-based portable genetic representation of the morphology, behavior and communication abilities of evolvable agents. Artificial Life and Robotics, 8:52–56.

Thomsen, R., Fogel, G., and Krink, T. (2002). A clustal alignment improver using evolutionary algorithms. In Evolutionary Computation. CEC '02. Proceedings of the 2002 Congress on, volume 1, pages 121 –126, May 2002.

Uvarov, B. (1977). Grasshoppers and Locusts, volume 2. 1977.

Vassilev, V.K., Job, D., and Miller, J.F. (2000). Towards the automatic design of more efficient digital circuits. In EH'00: Proceedings of the 2nd NASA/DoD workshop on Evolvable Hardware, page 151, Washington, DC, USA. IEEE Computer Society.