Robotics **2013**, *2*, 165-184; doi:10.3390/robotics2030165

OPEN ACCESS

robotics

ISSN 2218-6581

www.mdpi.com/journal/robotics

Article

Hormone-Inspired Behaviour Switching for the Control of Collective Robotic Organisms

Tüze Kuyucu [1,⋆], Ivan Tanev [2,⋆] and Katsunori Shimohara [2]

[1] Engineering Department, Griffith Textile Machines, Washington NE38 8QA, UK

[2] Department of Information Systems Design, Doshisha University, Kyotanabe 610-0321, Japan; E-Mail: kshimoha@mail.doshisha.ac.jp

⋆ Authors to whom correspondence should be addressed; E-Mails: tuze@gtm-ltd.co.uk (T.K.); itanev@mail.doshisha.ac.jp (I.T.).

Received: 1 June 2013; in revised form: 15 July 2013 / Accepted: 16 July 2013 /
Published: 24 July 2013

Abstract: Swarming and modular robotic locomotion are two disconnected behaviours that a group of small homogeneous robots can be used to achieve. The use of these two behaviours is a popular subject in robotics research involving search, rescue and exploration. However, they are rarely addressed as two behaviours that can coexist within a single robotic system. Here, we present a bio-inspired decision mechanism, which provides a convenient way for evolution to configure the conditions and timing of behaving as a swarm or a modular robot in an exploration scenario. The decision mechanism switches among two behaviours that are previously developed (a pheromone-based swarm control and a sinusoidal rectilinear modular robot movement). We use Genetic Programming (GP) to evolve the controller for these decisions, which acts without a centralized mechanism and with limited inter-robot communication. The results show that the proposed bio-inspired decision mechanism provides an evolvable medium for the GP to utilize in evolving an effective decision-making mechanism.

Keywords: collective robotics; modular robotics; swarm robotics; hormone-inspired; XGP; Genetic Programming; reconfigurable robotics; pheromone-inspired

1. Introduction

Modular robots present potential robustness characteristics beyond the capabilities of wheeled vehicles, such as: the ability to traverse challenging terrain and insignificant performance degradation when partial damage is inflicted. On the other hand, they lack the emergent intelligence, manoeuvrability and flexibility of swarms. Thus, modular robotic movement and swarming can be viewed as two complementing behavioural traits rather than mutually exclusive ones. In this article, we present a hormone- and emotion-inspired control mechanism for a collective reconfigurable robotic system. The objective of the proposed control mechanism is to decide whether a robotic organism that is deployed for the exploration of an unknown environment will benefit from switching between states of swarming into moving as a single modular robot and *vice versa*. Although a single decision at its core, the aforementioned mechanism's result is a determining factor in a switch between different and conflicting behaviours. Furthermore, the behavioural switching is time-consuming and a determining factor in the success and survival of the robotic organism. Such a decision can not be easily made via a predetermined algorithm or predetermined protocol, because of the environment being unknown, unpredictable and only partially observed by the perception subsystems of the robotic modules. Moreover, the collective decision, of the robotic system as a whole, to switch from a certain type of behaviour into another one when each individual robot makes its own decisions is even more difficult to formulate. Rather, it could emerge as a higher-level property of the complex, non-linear robotic system, as a result of the interaction between the lower-level entities (modules) and the environment [1]. In terms of the practical application being considered, the work by Berend *et al.* is the most related: to design a system that is reliant on online evolution in achieving organism formation via a swarm of robots in order to increase survivability of the robots [2]. The model presented here relies on offline evolution and is aimed at addressing predetermined objectives. The adaptability of the system is obtained as an emergent property derived from both the interactions (i) between the robotic modules and (ii) between modules and the environment.

Hormone- and emotion-inspired decision models in collective robotics have been developed by various authors to address a range of tasks in the control of a group of robots. The two terms (hormone and emotion) are often used in highly related systems; thus, we consider models from both sources of inspiration relevant and use the two terms interchangeably. The hormone-inspired decision systems model the behavioural effects of emergent emotional changes, due to the hormonal fluctuations that can be observed in biological organisms.

The models developed differ in the style of implementation, as well as the point of view taken on how these hormone-inspired models function. Often, the difference in the modelling comes from whether a low- or high-level point of view is used. In a system where the low-level interactions of hormones and chemical pathways are modelled, the aim is to create a tightly-coupled control mechanism that can define detailed behaviours in a robot. From a different point of view, the fluctuations in hormones are associated with certain emotions leading to high-level behaviours (often used as a behaviour switching mechanism among pre-defined behaviours rather than as a control mechanism). Shen *et al.* [3] present a control model for multiple robot coordination that is based on detailed hormone interactions. Their mechanism, based on Turing's reaction-diffusion model [4], defines the coordinated movements of robots via hormone-messages passed between neighbouring robots. Another reaction-diffusion-based complex coordination mechanism is used by Hamann *et al.* [5]. These two mechanisms presented by Shen *et al.* and Hamann *et al.* behave as detailed coordination mechanisms inspired by the micro-level interactions of the hormones in biological organisms. These mechanisms are designed to be applied as controllers of low-level actions (such as controlling the actuators of wheeled robots) that lead to an emergent higher-level behaviour. Due to the gap between the low-level properties of the micro-entities and the emergent high-level properties of the system as a whole, these mechanisms would be difficult to configure (or evolve).

Utilizing a simpler hormone-inspired decision mechanism limits the flexibility and creativity of the underlying system. However, it drastically simplifies the control models and allows them to be easy to modify, as well as easy to integrate with other control models. Murphy *et al.* [6] achieved multi-robot coordination in a small team of heterogeneous robots using a simple emotion-based control with a limited amount of coding. Moioli *et al.* [7] showed that a simple hormone-inspired approach to task switching in a swarm of robots works well with tasks featuring conflicting objectives (such as exploration and energy preservation). The ALLIANCEarchitecture [8], as well as a similar approach by Walker and Wilson [9] also provides simple, yet efficient, behaviour switching mechanisms based on motivations, such as impatience and acquiescence, similar to the hormone-inspired model proposed in our work. Both of these models use continuous broadcasts between robots, since they require the robots to be aware of others' tasks. This presents problems with scalability, as well as reliability issues as the number of modules in the swarm grows. Although not comprehensive enough to cover all the related hormone-inspired control mechanisms, an analysis of a large number of emotion-inspired mechanisms can be found in [10]. The proposed hormone-inspired mechanism is similar to the simple behaviour switching mechanisms used by many other authors. The novelty of our approach is in its area of application, integration, as well as its implementation.

The proposed decision mechanism is inspired by the fluctuations in the hormonal signals of biological organisms that cause changes in the emotional states, which, in turn, determine the actions for many crucial decisions. Our work is inspired by the drastic behavioural fluctuations often observed in biological organisms, due to the changes in hormonal regulators [11]. We propose an automated design of the decision mechanism via simulated evolution, because we believe that an *a priori* hand-coded solution would not be adaptable, due to the unknown and unpredictable nature of the environment. Moreover, such a solution would not be an optimal one, due to the inherent complexity of the modular robotic system. Indeed, the analytical models for such complex, modular robotic systems do not exist, and often, the desired high-level properties of the system as a whole cannot be directly inferred from the hand-coded low-level behaviour and morphology of its entities. We use XML-based Genetic Programming (XGP) for the evolution of the control mechanism (Tanev [12]). The evolved model is intended to decide the timing of switching between the swarming and snake-like locomotion behaviours, and *vice versa*. Both behaviours were developed in our earlier works [13,14]. The task is for a group of robots that approach a corridor as a swarm to overcome the various obstacles presented and explore as much of the area as possible in the process. The experiments demonstrate the intuitive behaviour achieved by the robots as a group by switching among the main behaviours (swarming, modular robot reconfiguration and movement) via the use of the presented hormone-inspired decision mechanism.

This article is organized as follows. In Section 2, we describe the adopted robotic modules and their simulated environment. Section 3 details the algorithms executed for all three behaviours that the robotic system can perform. Section 4 explains the decision mechanism designed for switching among various behaviours (described in Section 3). In Section 5, we describe the experimental setup and present the empirical results obtained from the hand-tuned version of the hormone-inspired algorithm. In the same section, we provide the values of the major parameters of XGP and the results obtained by the simulated evolution. Section 6 provides a comparative analysis on the results, presented in the preceding section, and Section 7 draws a conclusion of our work.

2. Robots and Their Environment

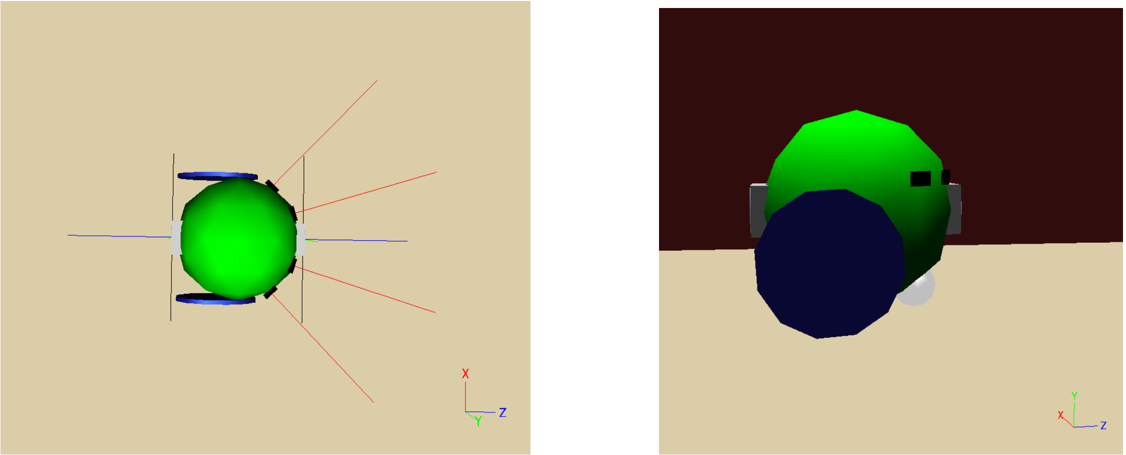
We use the commercially available robot simulation platform, Webots, in our experiments, which realistically models the physics of the adopted mobile robots, their interactions with each other and with the environment [15]. Table 1 details the parameter settings used for the Webots simulation platform. The robots are spherically shaped two-wheeled robots with a differential drive and $7\,cm$ in size. Robots feature a set of a simulated radio emitter/receiver and four infra-red (IR) sensors in front, as shown in Figure 1. The radio emitter and receiver set allow the robots to establish direct one-to-one and one-to-many communication with other robots. The radio communication models a simple RFmodule with a custom communication protocol, *i.e.*, without a known communication protocol, such as WiFi or Zigbee. The robots are equipped with two actuators—one in front (vertically oriented) and one at the back (horizontally oriented)—which have magnetic connectors attached that allow them to connect with other robots. The front actuator has an "active" connector (an electromagnet) that can be activated or de-activated by the robot. The rear actuator has a "passive" connector (a permanent magnet), which is always in the same state and can only be connected with an "active" type connector (since two passive connectors have the same polarity). Thus, the robot that is connecting with its "active" (front) actuator has the initiative in establishing a robot-to-robot connection.

Table 1. Webots related parameters of the simulated robots.

World Properties		**Robot Physics Properties**	
CFM	0.00001	density	-1
ERP	0.2	mass	0.1
basicTimeStep	32	centerOfMass	0 -0.03 -0.031
bounce	0.5	bounce	0.5
bounceVelocity	0.01	bounceVelocity	0.01
coulombFriction	0.5	coulombFriction	1
forceDependentSlip	0.004	forceDependentSlip	0
runRealTime	true		

The robots are designed to reflect a realistic size of a robot with similar capabilities (e.g., epuck, Symbrion robot [16]). They are aimed at being capable of forming a swarm of wheeled robots and achieve common tasks, such as exploration of unknown environments. The robots are also designed to substitute the spherical modules used in the sidewinding Snakebots in our earlier work [17]. The latter is the reason why the robots have a spherical casing. The robots here do not use ball joints as before (due to the limitations of the Webots simulation platform), but utilize two separate joints.

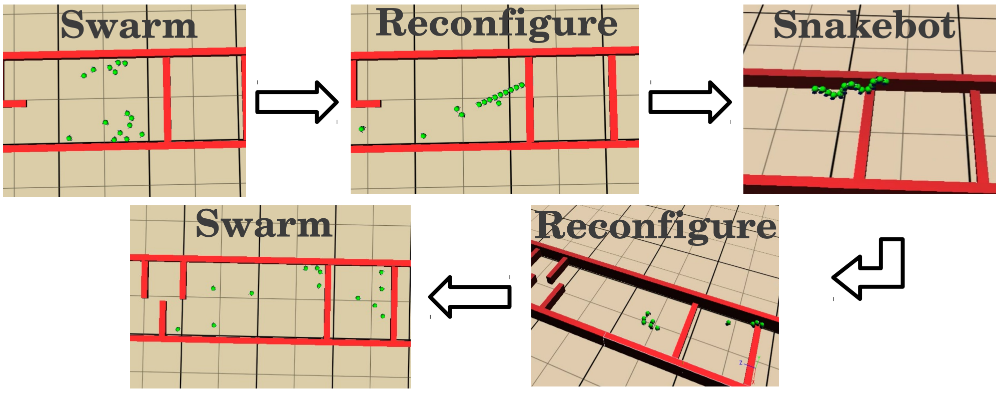
Figure 1. The differential-wheeled robot used in simulation experiments with four infra-red (IR) sensors of $7\,cm$ range pointing forward and two connection points that allow docking with other robots.



3. Collective Behaviours

To behave as a collective robotic organism, the robots need to be able to achieve two behavioural states: (i) swarming; and (ii) modular robot. These two states require completely different behaviours from the individual robots, and thus, they are coded as separate control mechanisms. In order to achieve a transition between these two states, the robots need to be able to "reconfigure" themselves, which is the third behavioural state: (iii) reconfiguration. All three behavioural states are illustrated in Figure 2 to demonstrate an example case where these three behaviours would be highly beneficial if used correctly. All these behaviours are previously developed and are not the main focus of this work. Therefore, in the following subsections, we will only provide a brief explanation of these two behaviours.

Figure 2. An example scenario of the collective robotic organism utilizing all three behaviours.



3.1. Swarming

Swarm intelligence is an emergent behaviour from the decentralized interactions between many systems. In robotics, it is the coordinated (explicitly or implicitly) movement and actions of many robots that contribute to achieving tasks via the emergent behaviour. We use the pheromone-inspired swarm control [14], which does not require any direct interaction within the robots to achieve the multi-robot coordination required. Pheromones can be used to provide a stigmergic medium of communication, which influences the future actions of a single individual or a group of individuals via changes made to the environment. A term first introduced by Grasse [18], stigmergy as a communication mechanism allows the history of an individual's actions to be tracked without the need to construct a model of the environment within the individual's own memory; making the emergence of higher complexity behaviours from a group of simple individuals possible.

The swarm control algorithm used here is a pheromone-inspired mechanism that builds environmental information by chemical gradients. The environment is assumed to handle the storing and diffusion of chemicals; thus, the robot controllers do not store any chemical information, except the sampled concentrations within the immediate vicinity of the robot. The algorithm used is developed specifically for achieving optimum exploratory behaviour via a large number of real robots in unknown environments. Such environments include areas of high devastation (e.g., earthquake or tsunami stricken settlements) or distant and dangerous missions. Exploring an unknown area quickly is a mission-critical objective in rescue operations. Such operations can face a list of limitations, such as the lack of a terrain map, the failure of previously established communication networks and the lack of reliable GPS tracking. In such missions, the first task is to search the area in question as quickly as possible and locate targets. The robots would be required to be capable of various functionalities other than area exploration; therefore, it is desired that the integration to a swarm and the ability to explore are seamless and do not consume a large amount of the robot's resources. Utilizing a real stigmergic communication would be an efficient method of achieving such emergent behaviour with low overhead.

Robots that utilize stigmergic trails to communicate with each other have been shown to effectively coordinate and quickly explore a given terrain [19,20]. The use of stigmergic pheromone-based communications in robotics has a range of other potential advantages, such as the possible ability to adjust the range and persistence of a pheromone, not being limited to line-of-sight, the ability of pheromones to propagate through the environment (while forming gradients) and freeing the individual robots from the burden of communication management and processing—usually involved with other forms of commonly used communications (such as radio, IR, visual, audio, *etc.*). On the other hand, the use of physical substances for pheromone-based communication within robots is problematic and poorly understood. However, there is undergoing work in improving their use with promising results, and it is predicted that with improvements in sensing technology, it may be possible that a robot could carry a lifetime supply of chemicals [21]. The use of olfactory sensors in robotics is a developing area with promising demonstrations of distinguishing multiple odours by mobile robots, e.g., [22], and recent development of high resolution olfactory sensors for robotic systems [23]. With these developments, the use of real chemicals for inter-robot communication can become as convenient as any other conventional method, such as infra-red or radio.

The swarming behaviour used here is similar to the "node counting" algorithm described in [20]; however, in our case, the goal is to achieve a quick "survey" of an area by visiting key locations as quickly as possible, instead of "sweeping" by visiting all the available locations in the environment. The developed algorithm assumes that a more probabilistic approach to exploration is highly beneficial in realistic situations and that the common strategy of sweeping of the whole area for the exploration of unknown environments is highly inefficient and unrealistic. The algorithm is implemented as a three-layered subsumption architecture, where each layer implements one basic behaviour: *random walk* ("exploration") is the behaviour corresponding to the lowest priority layer, while *pheromone-based coordination* is the middle layer, and the *wall avoiding* behaviour is realized by the highest priority architectural level. In the pheromone-inspired algorithm, the robots explore a given environment by dropping artificial pheromones in their environment (to mark the visited locations) and sampling the pheromones dropped by other robots. Each pheromone has the ability to diffuse and evaporate, and they are simulated in a grid-like environment, where each grid point is the size of a robot.

The pheromone-inspired swarming (a completely separate mechanism from the hormone-inspired mechanism, which is the main focus of this article) is explained in detail in [14].

3.2. Modular Snakebot

Due to its simple chain-like geometry, the automated formation and configuration of a modular Snakebot is relatively easy. Some useful features of snake-like robots include the smaller size of the cross-sectional areas (in comparison to other modular robots), stability, ability to operate in difficult terrain, good traction and complete sealing of the internal mechanisms [24,25]. Moreover, due to the modularity and homogeneity of their design, the snake-like robots have high redundancy, which, in turn, provides it with inherent fault tolerance and adaptability properties [17].

In our model, once the robots connect to each other to form a long chain, they can use the actuators between the neighbouring robots to achieve locomotion. To achieve coordinated movement amongst the modules, the robots synchronize their internal timers and assign themselves individual IDs depending on their location in the Snakebot. The head of the snake being zero, the tail, n − 1 (in a Snakebot with n modules). It was noted in an earlier work that a sidewinding locomotion is the fastest and most efficient form of locomotion for a snake-like robot [17]. Even though the results obtained in those early studies are used here, we implement a rectilinear locomotion instead. The current robots achieve a stable rectilinear locomotion, due to the presence of wheels. In fact, the addition of wheels also makes it difficult to use the previously evolved sidewinder controls, since the wheels can change the geometry of the module. Equation (1) (adapted from the evolved solutions in [13]) is used for the movement of vertical actuator in each robot, which successfully achieves a moving sine-wave within the Snakebot. The horizontal actuators are locked at their initial perpendicular position. Although the motion gait, considered in this work, is the rectilinear one (which is also similar to the locomotion gait of caterpillars) instead of sidewinding, as exhibited in our previous implementations of the Snakebot, we use the term Snakebot to refer to the specific morphology of the modular robot (rather than particular locomotion gait) in order to provide a clear connection to the source and inspiration of this work [17]. A rectilinear locomotion is preferred in these preliminary experiments, because, compared to sidewinding, it offers the advantages of (i) simplicity of implementation, (ii) simplicity of steering and (iii) a much reduced cross-sectional area of the moving Snakebot.

$$V_p = -0.1 \times (id - 16) \times \pi sin(id + 3t - 1) \tag{1}$$

Equation (1) is used with a maximum of 15 modules in the experiments. V_p is the position of the vertical actuator in radians. This equation is only used for achieving locomotion within a modular robot and does not have an impact on the hormone-inspired mechanism described here.

3.3. Reconfiguration

In order to switch from a swarm to a single, yet modular, robot (and *vice versa*), the robotic modules need to coordinate their motion and assemble together (or disassemble, respectively). This process is referred to as reconfiguration, and it can be initiated by one or more robots. The remaining robots then have the freedom to join or not join the initiating robot. When there are no robots willing to join the modular robot, the connected robots (if more than one) may then start to move as a single, snake-like modular robot. In a reverse situation, the robots simply decide to disconnect, while moving as a Snakebot, and, once separated, to start moving as entities of a robotic swarm. For the experiments presented in Section 5, the initiating robot first turns towards east, which is the direction that the experimental corridor stretches. This is done to make sure that the Snakebot is facing the correct direction once it is formed. Reconfiguration is the only stage the presented robots communicate directly via wireless messages. These wireless messages involve broadcasts of a robot's request for potential partners to dock and the positive responses that the other robots may give.

Self-assembly and automatic reconfiguration can be quite a challenging task on its own depending on the number of the degrees of freedom of the modular robot (e.g., [16]), and sophisticated solutions that rely on evolutionary computational techniques have been developed, e.g., [26]. The focus of our work, however, is not in the self-reconfiguration of modular robots; hence, the complexity of the reconfiguration scenario is kept to a minimum. Since the considered modular robots have only two points of connection (front and back), the only shape that can be achieved is a snake-like shape, which simplifies the possible control mechanisms required for the locomotion of the formed modular robot. The tasks of robotic modules in achieving a reconfiguration of a swarm of robotic modules into a single modular Snakebot involve:(i) finding robots in the swarm that want to dock; (ii) locating and docking to each other without collisions; (iii) generating the correct IDs (used for Snakebot locomotion) for each robot; and (iv) deciding when the reconfiguration is complete. The initial decision to initiate a reconfiguration or to become available for reconfiguration participation is decided by a separate mechanism. This mechanism is the primary focus of our work and will be elaborated upon later in Section 4.

Once a robot decides to initiate a reconfiguration process, it broadcasts a message (via radio communication) to all other robots to signal its decision. The robots that are available for docking respond with messages addressed to the initiating robot. If no responses are received (implying that no robots are available for docking), the initiating robot terminates the reconfiguration process and returns back to the swarm. Otherwise, the initiating robot picks the closest one of the responding robots by estimating the distances from the signal strength of the received responses. Once chosen, the initiating robot waits for the other to dock, occasionally exchanging messages in order to: (i) maintain connection;

(ii) provide the docking robot with the direction the waiting robot is facing; and (iii) allow the docking robot to have a rough estimation of the distance between them. If the connection between the two robots is lost or the docking robot takes a long time to accomplish the procedure, the waiting robot terminates the procedure and looks for a different robot to connect with. If the docking process completes successfully, the docked robot takes the role of the initiating robot and looks for other robots to connect with. However, if in this case, the waiting robot is unable to get a response from other robots for docking, it then signals all the other robots that are part of the same modular robot to switch into modular robot mode (by synchronizing timers) and start moving as a Snake.

Disassembly is a much simpler process, which does not require any coordination among the robots. Again, the decision is made by the mechanism explained in detail in Section 4. The disassembly results in the deactivation of the front ("active") connector of the corresponding robot.

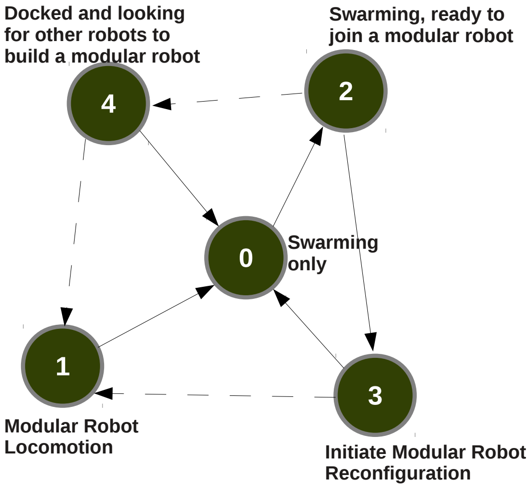
4. Hormone-Inspired Behavioural Switching Based on Patience

The decision to switch states (disconnect from a modular robot and join a swarm or start forming a modular robot while swarming) is made on an "impatience" value that the robots increase or decrease depending on the environmental factors. This is modelled after the biological organisms, namely animals, which maintain emotions, such as anxiety, tolerance, restlessness and eagerness, that contribute to large changes in behaviour via hormonal feedback in their bodies. Hormones in biological organisms are known to achieve coordination amongst different members and are a determining factor for social behaviours [11]. We are inspired by this duty of hormones in biology and believe that a similar decision mechanism could provide intuitive changes in robotic behaviour. Here, we provide a simplistic model of a hormone, where we do not try to provide a biologically-plausible implementation. The main source of inspiration is not how the hormonal chemical networks work in biology, but the situational uses (*i.e.*, behavioural switching) of slow, but smooth, chemical gradients instead of fast and sharp logical decisions. The model presented provides some of the basic functions of a hormonal network in biological organisms, namely: (i) storage and secretion of hormones; (ii) recognition and processing of the hormone; and (iii) degradation of the hormone. The hormones in our model are not transported to other "cells", since they are used in the regulation of a single-cellular entity (*i.e.*, the individual robot) that does not share its hormonal state with other robots. However, the resulting behavioural shifts in a robot are shared with other members of the group, which affects the status of the other individuals. Thus, the hormones produced indirectly control the behavioural state of the whole group.

The impatience value used to model the behaviour of hormones has a certain range, and it constantly degrades over time (multiplied by 0.95 every time step in the presented experiments), even when not in use. The production (secretion) of the impatience value is determined by a separate logic (hand-coded or evolved), which only controls the incremental changes in the value of the impatience value, as elaborated in Section 5. We define a threshold level for the impatience value, and if this value is reached, an action takes place depending on the robot's state. If the impatience value is high enough to initiate an action, it is reset to zero (*i.e.*, it is "consumed"). The three major behaviours described in Section 3 define the states that a robot can be in. We spread these three behaviours to a total of five states: **State 0** is swarming only, **State 1** is Snakebot locomotion only, **State 2** is swarming, but ready to join a Snakebot,

State 3 is initiate reconfiguration for a Snakebot and **State 4** is docked and looking for other robots to form a Snakebot. The three behaviours are spread over five states in order to simplify the reconfiguration process and the integration of the three distinct behaviours and encourage gradual change in behaviours rather than abrupt switching. For the experiments presented in Section 5, the robots are initialized in State 2. The decision mechanism determines the changes between these states that place the robot in a different behavioural zone. When a change in state is triggered, the state transitions are as follows: State 0 –> State 2; State 2 –> State 3; states 1, 3 or 4 –> State 0. Figure 3 illustrates the state machine built to accomplish the desired behaviours, with the aforementioned five states.

Figure 3. The states and the possible transitions in the main controller of the robots. The solid lines illustrate the state transitions, due to the decisions made by the hormone-inspired behaviour switching, whereas the dashed lines represent the state transitions, due to changes in the environmental conditions.


Docked and looking for other robots to build a modular robot
Swarming, ready to join a modular robot
4
2
0
Swarming only
1
3
Modular Robot Locomotion
Initiate Modular Robot Reconfiguration

These states and the state transitions define the behaviours and the order of change between these behaviours, which are easy to define. The most difficult task, however, is to decide when to initiate these state transitions. Table 2 shows a list of variables that each robot has access to and we believe are sufficient for the robots to analyse their environments and adjust their internal states. All these variables are either acquired without any communication or via the existing communication taking place among the robots during the reconfiguration procedure.

Table 2. Perception information available to each robot relevant for determining the change in the impatience value.

Perception	**Description**
lowestPH	Value of lowest pheromone conc.requests
No. largestModuleSize	No. of modules in the largest active Snakebot.
moduleImpact	$largestModuleSize - currentSnakebotSize$

The pheromone levels, which are stored by the environment, provide information to individual robots about the environment and whether the goal of exploration is being satisfied [14]. The $requests$, $largestModuleSize$ and $moduleImpact$ are affected by the behavioural state of the other robots in the environment. These variables inform the individual robots about the state of the other robots and create the means for social monitoring and pressure. Thus, the hormonal changes within the other robots have the possibility to cause hormonal changes within.

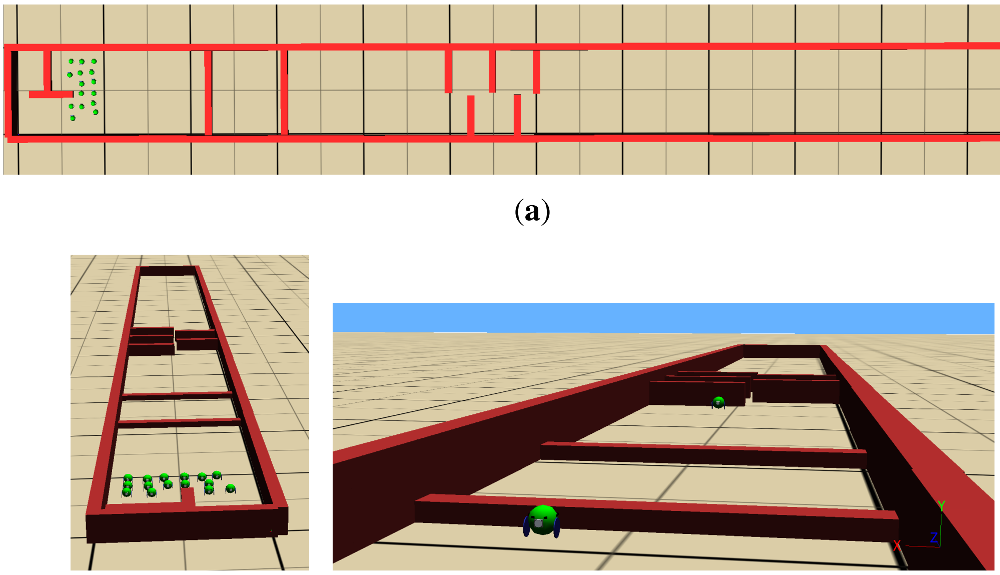
5. Experiments

An experimental environment is designed to develop and test the decision mechanism for its ability to provide efficient and intuitive switching between behaviours in order to accomplish quick exploration of unknown areas. The environment is designed to test the ability of the robotic group to demonstrate all three behavioural states to successfully explore a given environment without any information about the area. The environment is a long corridor ($24\,m$) with two types of obstacles: low continuous walls and high walls with gaps. The low walls present a challenge to individual robots, which are too small to overcome these obstacles, but these low walls can be climbed over by a Snakebot. The high obstacles cannot be overcome by either the individual robots or a Snakebot. However, the high obstacles are arranged to have small offset gaps that give individual robots the opportunity to circumnavigate. Figure 4 provides different views of the environment.

Fifteen robots are initialized at the left end of the corridor as a swarm. The goal for the robots is to clear all the obstacles in the way and explore as much of the environment as they can. The robots are expected to explore this section as well as they can before joining together to overcome the low obstacles blocking their way. Once formed, the modular robots that overcome the low obstacles are expected to partially disassemble after the first low obstacle, while the rest disassemble when they reach the high obstacles, where they can no longer move forward. The disconnected individual robots that reach the high obstacles are then expected to find their way to the other side of these high obstacles. This requires the robots to form a Snakebot and then go back to being a swarm at least once.

We use a simple hand-coded algorithm shown in Algorithm 1 to successfully clear all the obstacles shown in Figure 4 with appropriate task switching in approximately 7.5 min. Different behaviour control patterns can be obtained by adjusting the constants used for the impatience increment, as well as the constant values, $PHEROMONECEILING$, $REQUESTCEILING$ and $NHEADMODULES$. The values used for the latter three in our experiments are 600, 100 and 1, respectively. The constants are picked and optimized via numerous trial and error runs in the experimental environment. For example, $PHEROMONECEILING$ is set to 600, as this is the commonly encountered minimum pheromone concentration surrounding a robot in an already explored area. In cases where a smaller value is detected, the area is likely to be only partially explored, and for higher values, there is a chance of long delays in behaviour switching, even after the area is fully explored (thus, 600 is picked as a sub-optimal compromise). The value of the impatience increment in the case of high pheromone levels is set to 2, which is not a large enough increase to bring the impatience levels high enough (50) to trigger a change in behaviour, since the impatience value is multiplied by 0.95 every time step (thus, the maximum impatience value attainable is $\frac{2}{0.05} = 40$). Most of the other sub-conditions under the initial condition (of high pheromone values or requests) increase the impatience value with larger increments of 5.0. In these cases, a quick change in behaviour is required, and we aim to bring them about with large increments. The latter is not true for the first condition, which checks if the robot is part of a moving modular robot. In this case, the increments are made in significantly smaller values (0.5001, which means that the maximum value impatience can reach is $\frac{2.0+0.5001}{0.05} = 50.002$) in order to allow the Snakebot the opportunity to move when it is first formed. By the time the newly formed modular robot can start moving, the impatience value is already high (*i.e.*, 40.0); thus, using a large increment value would cause the Snakebot to disassemble before it can start to move.

Figure 4. The experimental environment viewed from three different positions.



(a)

(b) **(c)**

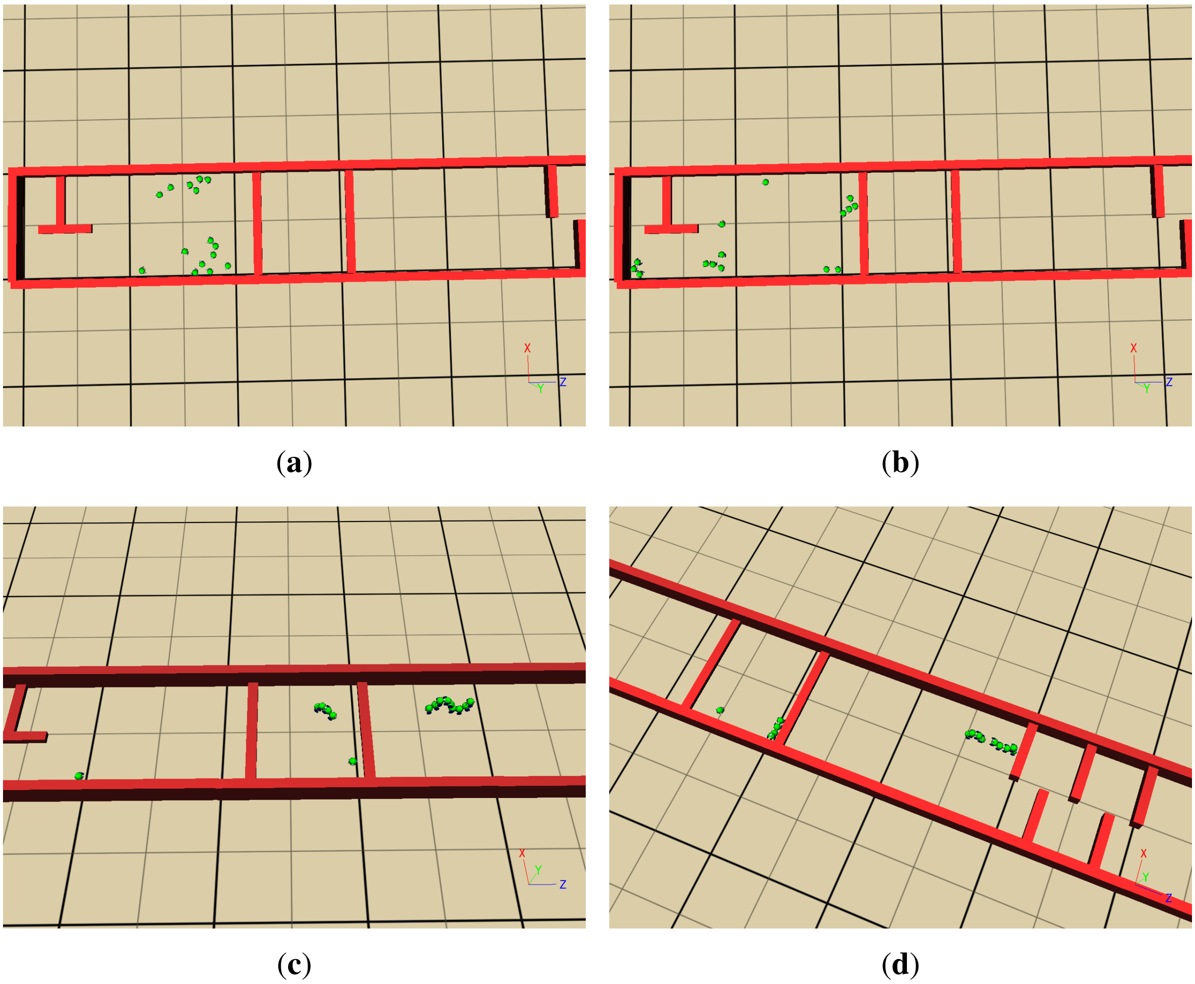
Algorithm 1 The hand-coded algorithm for controlling behaviour changes by adjusting the impatience value.

```
if $lowestPH > PHEROMONECEILING$ or $requests > REQUESTCEILING$ then
    $impatience \leftarrow impatience + 2.0$
    if $state = 1$ and $impatience >= 39.9$ then
        $impatience \leftarrow impatience + 0.5001$
    else if $(state = 1$ or $state = 3$ or $state = 4)$ and $moduleImpact > 0$ then
        $impatience \leftarrow impatience + 5.0$
    else if $state = 0$ and $largestModuleSize < 6$ then
        $impatience \leftarrow impatience + 5.0$
    else if $state = 2$ and $largestModuleSize < NHEADMODULES$ then
        $impatience \leftarrow impatience + 5.0$
    end if
end if
```

An example run in the environment shown in Figure 4 with 15 robots is illustrated in Figure 5.

As mentioned earlier, it takes a total of 7.5 min for the first robot to clear all the obstacles. The algorithm is written to ensure that there is only one robot initiating the configuration of a Snakebot. Although the latter ensures that the Snakebot formed is as large as it can be to have the best chances of overcoming the obstacles, it is time-consuming, and the configuration process takes more than three minutes. The amount of time it takes for the first robot to clear all the obstacles can be reduced by allowing two robots to initiate the Snakebot configuration process (by setting $NHEADMODULES$ to 2). In the latter case, the first robot clears all the obstacles within 5 min. However, neither of the Snakebots disassemble between the two low obstacles; thus, that area remains unexplored. If, however, the number of robots that can initiate the Snakebot configuration process is set to 3 or more, no Snakebots that can cross the low barriers form (they are too small); thus, none of the robots can clear the obstacles.

Figure 5. Snapshots of a run using the hand-crafted Algorithm 1. (**a**) 10 s; (**b**) 1 min 18 s; (**c**) 4 min 20 s; (**d**) 4 min 35 s; (**e**) 5 min 3 s; (**f**) 7 min 33 s.



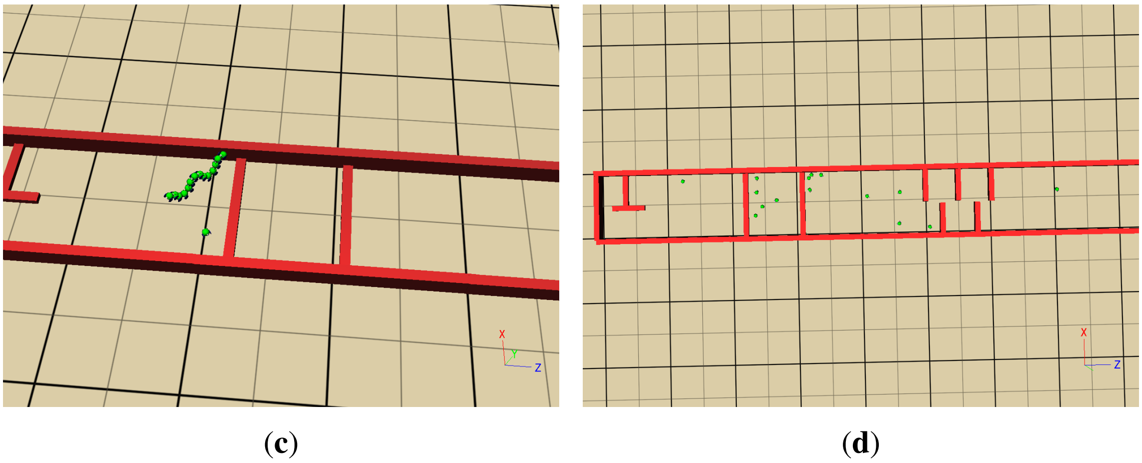
(**a**)

(**b**)

(**c**)

(**d**)

Figure 5. *Cont.*



(c) **(d)**

We believe that this is a well-fitting problem for GP, as it only involves 4 perception inputs and a single output; yet, it is a difficult problem to solve using hand-coded logic without any map-specific information. The mechanism of incrementing the impatience value that yields a desired behaviour of the robotic system as a whole is not obvious. The changes in behaviours need to be well synchronized amongst the robots in order to prevent fruitless oscillations between their respective states. By utilizing XGP, we intended to evolve the optimal mathematical model of the conditions that can trigger the transitions among a large group of robots to overcome the obstacles in order to successfully explore their environment. The population of XGP includes 200 individuals with an elite size of 10 individuals. To create the remaining 190 individuals of a new generation, we employ a binary tournament selection: two individuals are picked at random. Ninety-percent of the time, a new individual is created via single point crossover (reproduction), and 10% of the time, the fittest of the two is chosen to be passed on to the next generation. The crossover point is randomly selected within the genotype. The mutation randomly alters 2% of the newly created individuals (all except the elites). Each run lasts 40 evolutionary generations. Table 3 illustrates the main parameters of XGP. The set of terminal symbols of XGP consists of the four perception values (as shown in Table 2), the randomly generated floating point constants of [0..1], and the integer constant of [0..100]. The function set consists of the mathematical operations, addition, subtraction, multiplication and division.

The genotype of the individuals in XGP is represented as parse trees. These threes are evolved to increment the impatience value in order to trigger beneficial state changes in various environmental conditions. Each individual is evaluated for 1,000 s in the Webots simulation platform, which roughly corresponds to 5 min of a real-time run of the simulation platform on average. Each experiment involves 15 robots, with a homogeneous breeding strategy in that each robot is controlled by the same individual of XGP being evaluated. The fitness of each individual is determined by the number of checkpoints visited and the number of robots that clear all the obstacles. Checkpoints are placed every 25 cm in the environment, and they are meant to encourage exploratory behaviour, as well as prevent an over-fit solution to overcoming the obstacles in the corridor (*i.e.*, filter out the solutions that start reconfiguration to form a modular robot before any exploration is done). The fitness value is evaluated according to Equation (2).

$$fitness = 10 \times NoRobotsClearObstacles + checkPointsVisited \tag{2}$$

Table 3. Main parameters of XML-based Genetic Programming (XGP).

Category	**Value**
Genotype	Impatience increment (parse tree)
Population Size	200 individuals
Selection	Binary tournament selection; ratio: 0.1 Reproduction ratio: 0.9
Elitism	10 individuals
Mutation Rate	0.02
Trial Interval	1,000 simulated seconds
Termination Criteria	No. of generations = 40

After a total of 20 independent evolutionary runs, various successful control mechanisms evolve that can achieve the desired behaviour. Figure 6 shows the average fitness convergence over the 20 evolutionary runs. In 10 of the runs, the robotic system, controlled by the evolved mechanism of reconfiguration, was able to clear all the obstacles in the experimental environment. Out of the 10 successful runs, 6 provided robust solutions, where the re-runs could achieve good results. The reason for the latter is the over-fitting of the environmental conditions created during the evolutionary runs. Although half of the evolved controllers were able to cross both the low and high obstacles, only two got stuck at the tall obstacles, 1 got stuck in between the low obstacles and the rest could not cross the low obstacles. We can conclude that the successful formation of a Snakebot in time to cross the low obstacles was the major crux of the problem for evolution.

We expected to obtain a control mechanism that can form a single Snakebot from all 15 modules to carry them over the two low obstacles and then disassemble when the high obstacles are encountered. Although such control mechanism emerged, the most common evolved behaviour was the formation of multiple Snakebots (mostly two or three Snakebots) to overcome the obstacles. The latter was implicitly favoured by evolution, due to the gains in speed in carrying the robots over the obstacles, which allowed them a longer time to explore the much larger area beyond the high obstacles. By having multiple Snakebots configured at the same time, not only the reconfiguration process was parallelised, but also the problem with narrowing spaces behind the large Snakebots that slow down the docking robots were alleviated. Furthermore, the formation of multiple robots is a more robust approach, since the Snakebots are prone to falling over (which immobilizes the Snakebot) when using the rectilinear locomotion. In the case of a single Snakebot, this meant that the robots are unable to cross the low obstacles if the Snakebot falls over shortly after they are formed, whereas, in the case of multiple Snakebots, there is a smaller chance of all of the modular robots toppling over. Figure 7 shows snapshots from some of the successful runs, all of which demonstrate different behaviour and utilize different environmental information in making the decisions.

Figure 6. The fitness convergence of the runs over the evolutionary generations. The dashed lines at fitness 60 and 80 roughly indicate the fitness values when the first two low obstacles are crossed. The thick dashed line is the average fitness convergence of all the runs. The standard deviation starts at around 10 for the first generation and reaches and stays constant at 90 after the 5th generation.

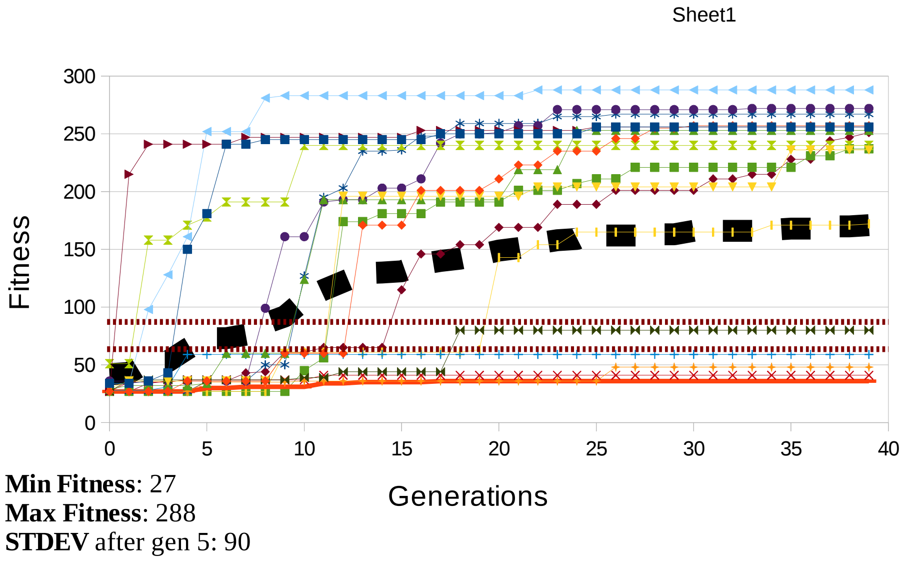


Figure 7. Snapshots of three different control mechanisms evolved using XGP. Runs 1 and 2 illustrate the formation of two separate Snakebots, and Run 3 shows a large Snakebot being formed to cross the low obstacles. The controllers evolved for the first and third runs dismantle the Snakebot when an unexplored area is encountered, whereas the evolved controller used in the second run utilizes a strategy of disassembling the Snakebot after some time passes from the discovery of an unexplored area. In the latter case, the Snakebots cross the low obstacles and get stuck at the high obstacles for a while until there is a high enough pheromone concentration nearby. (**a**) Run 1; (**b**) Run 1; (**c**) Run 1; (**d**) Run 2; (**e**) Run 2; (**f**) Run 2; (**g**) Run 3; (**h**) Run 3; (**i**) Run 3.

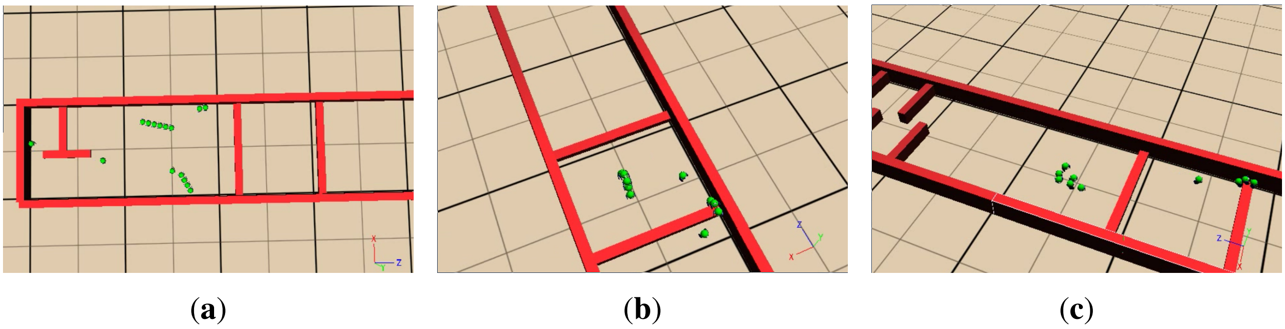
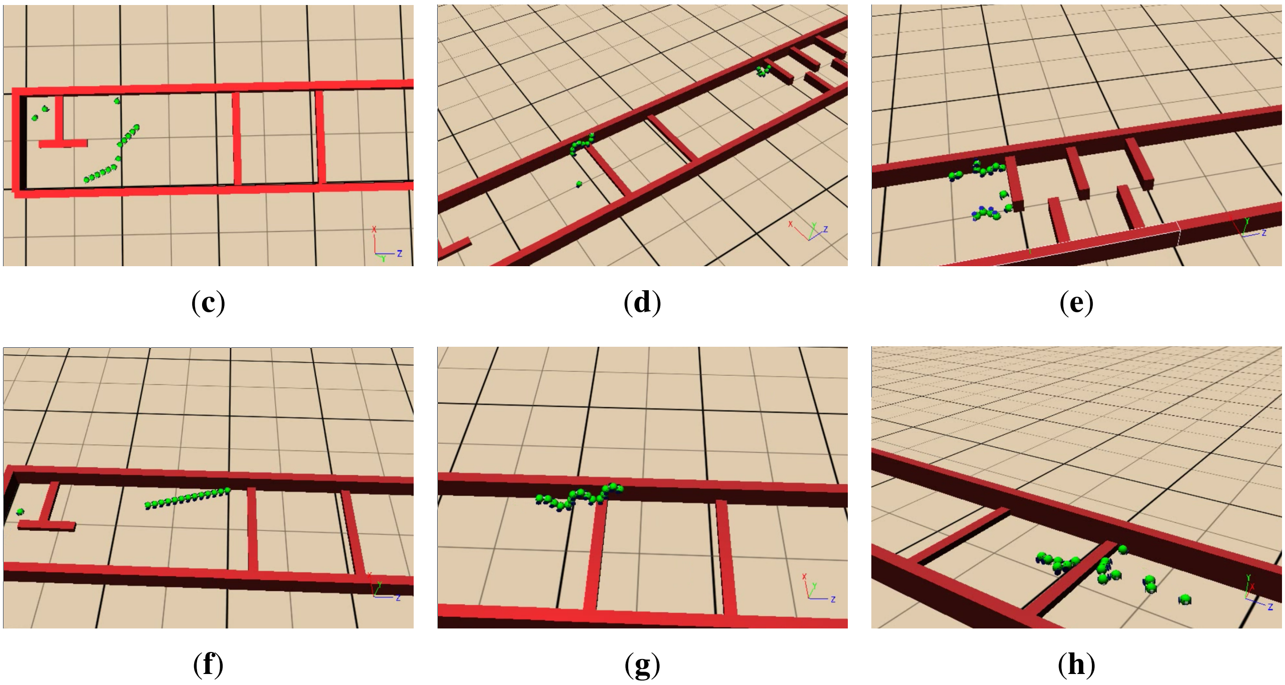

(**a**) (**b**) (**c**)

Figure 7. *Cont.*



(**c**) (**d**) (**e**)

(**f**) (**g**) (**h**)

6. Discussion

In our observations, the resulting genetic programs mainly utilize the $lowestPheromone$ variable to initiate state changes in the robots. The remaining inputs are used to limit the frequency and number of robots changing into particular states (such as initiating modular robot reconfiguration). In all cases, we observed that when the $lowestPheromone$ level reached above a certain threshold, the robots under the influence would start reconfiguration (if they were swarming). On the other hand, the variables, $largestModuleSize$ and $moduleImpact$, were used for preventing the robots from initiating too many reconfigurations. The use of randomized inputs were rare. The initial idea of using random variables as an input was to encourage specialization among robots and avoid switching to same states at the same time. In our attempts to construct some simple hand crafted controllers prior to the evolution runs, the most common problem was preventing global state changes (for example, when all the robots want to initiate reconfiguration). The inputs listed in Table 2 seem to be sufficient in preventing this; thus, a random input may not be beneficial.

The procedure for disassembling from a modular robot was significantly different among the various controllers evolved. One of the two common solutions involved detecting low pheromone concentrations. This proved to be a good approach to ensuring that all the newly discovered areas can be efficiently explored by a swarm. This, however, caused two problems: (i) early disassembly, where some of the modules might be left behind or on top of an obstacle (e.g., Figure 7(c)); or (ii) the inability of some robots to disconnect from the Snakebots once the pheromone levels increase in the environment. The latter could be solved by not allowing the robots to drop pheromones while they are part of a modular robot. This may, however, lead to other unanticipated problems and needs experimenting. The release of pheromones by robots in modular mode help them keep the other robots away during reconfiguration. This helps to reduce collisions and to speed-up reconfiguration process. The second solution to disassembly of Snakebots involved using pheromone values within a certain range, which made sure that the Snakebots do not disassemble the moment they start moving (high pheromone concentration area), as well as cover some new ground, which ensured that all robots are carried as far as they can be via the Snakebot (no disassembly in the low concentration areas). Although this solution scored quite high, due to the large number of robots that could clear all the obstacles, it meant that the small area between the two low obstacles remains partially explored. In this case, there were also some robots that could never disassemble, due to the pheromone levels becoming too high, which left them defunct.

7. Conclusions

In this work, we presented a simple decision mechanism for behaviour switching in a collective robotic organism. The decision mechanism uses simple rules and is based on the accumulation, as well as decay of an "impatience" value inspired by the hormonal regulation of emotions in biological organisms. The decision mechanism is used to orchestrate a range of previously developed behaviours (such as swarming, modular robotic locomotion and reconfiguration) in accomplishing a challenging task for a group of 15 robots. The control of the decision mechanism is achieved via an evolved Genetic Program (GP), which utilizes various information that is readily available to the robots from their interactions with the environment or other robots. The controllers developed for the behaviour switching relied heavily on the information gathered via the pheromones dropped by swarming robots, which show that the use of simple chemical gradients in the environment can be useful for coordinating behaviours other than swarming.

The evolved solutions present multiple ways of achieving the target goal of exploration of the whole corridor. The robots are able to clear the obstacles quickly ($\approx$4 min when three Snakebots are formed and $\approx$6 min when a single Snakebot is formed). The presented hormone-inspired behaviour switching mechanism provides a gradual, yet constrained, way to control the changes in the emergent behaviours. The use of XGP yielded quick results in achieving the desired behaviours, demonstrating that the underlying mechanisms are evolvable. The evolved solutions outperformed the hand-coded controller by a large margin and provided a greater diversity of strategies. The solutions indicate a few remaining uncertainties, such as the need for a Snakebot to release pheromones, and the use of random variables, which require further investigation.

Acknowledgments

The presented work is part of a fellowship funded by JSPS.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Jin, Y.; Guo, H.; Meng, Y. A hierarchical gene regulatory network for adaptive multi-robot pattern formation. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2012**, *42*, 805–816.
2. Weel, B.; Haasdijk, E.; Eiben, A. The Emergence of Multi-Cellular Robot Organisms through On-Line On-Board Evolution. In *Applications of Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7248, pp. 124–134.
3. Shen, W.; Will, P.; Galstyan, A.; Chuong, C. Hormone-inspired self-organization and distributed control of robotic swarms. *Auton. Robots* **2004**, *17*, 93–105.
4. Turing, A. The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond.* **1952**, *237*, 36–73.
5. Hamann, H.; Stradner, J.; Schmickl, T.; Crailsheim, K. A Hormone-Based Controller for Evolutionary Multimodular Robotics: From Single Modules to Gait Learning. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10), Barcelona, Spain, July 2010.
6. Murphy, R.; Lisetti, C.; Tardif, R.; Irish, L.; Gage, A. Emotion-based control of cooperating heterogeneous mobile robots. *Robot. Autom. IEEE Trans.* **2002**, *18*, 744–757.
7. Moioli, R.C.; Vargas, P.A.; Husbands, P. A Multiple Hormone Approach to the Homeostatic Control of Conflicting Behaviours in an Autonomous Mobile Robot. In Proceedings of the Eleventh Conference on Congress on Evolutionary Computation (CEC'09), Trondheim, Norway, 18–21 May 2009; IEEE Press: Piscataway, NJ, USA; pp. 47–54.
8. Parker, L. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *Robot. Autom. IEEE Trans.* **1998**, *14*, 220–240.
9. Walker, J.; Wilson, M. A Performance Sensitive Hormone-Inspired System for Task Distribution Amongst Evolving Robots. In Proceedings of the IROS 2008, IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, September 2008; pp. 1293–1298.
10. Rumbell, T.; Barnden, J.; Denham, S.; Wennekers, T. Emotions in autonomous agents: Comparative analysis of mechanisms and functions. *Auton. Agents Multi-Agent Syst.* **2012**, *25*, 1–45.
11. Adkins-Regan, E. *Hormones and Animal Social Behavior*; Princeton University Press: Princeton, NJ, USA, 2005.
12. Tanev, I. DOM/XML-based portable genetic representation of the morphology, behavior and communication abilities of evolvable agents. *Artif. Life Robot.* **2004**, *8*, 52–56.
13. Kuyucu, T.; Tanev, I.; Shimohara, K. Incremental Genetic Programming via Genetic Transpositions for Efficient Coevolution of Locomotion and Sensing of Simulated Snake-Like Robot. In Proceedings of the European Conference on Artificial Life, Paris, France, August 2011; pp. 439–446.
14. Kuyucu, T.; Tanev, I.; Shimohara, K. Evolutionary Optimization of Pheromone-Based Stigmergic Communication. In *Applications of Evolutionary Computing, EvoApplications 2012*; Springer Verlag: Malaga, Spain, 2012; Volume 7248, pp. 61–70.
15. Michel, O. Webots: Professional mobile robot simulation. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 39–42.

16. Liu, W.; Winfield, A.F.T. Autonomous Morphogenesis in Self-Assembling Robots Using IR-Based Sensing and Local Communications. In Proceedings of the 7th International Conference on Swarm Intelligence (ANTS'10), Brussels, Belgium, September 2010; Springer-Verlag: Berlin/Heidelberg, Germany; pp. 107–118.
17. Tanev, I.; Ray, T.; Buller, A. Automated evolutionary design, robustness and adaptation of sidewinding locomotion of simulated snake-like robot. *IEEE Trans. Robot.* **2005**, *21*, 632–645.
18. Grassé, P.P. La reconstruction du nid et les coordinations inter-individuelles chez Bellicosi-termes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs (in French). *Insectes Sociaux* **1959**, *6*, 41–83.
19. Beckers, R.; Holl, O.E.; Deneubourg, J.L.; Bielefeld, Z.U.; Bielefeld, D. From Local Actions to Global Tasks: Stigmergy and Collective Robotics. In *Artificial Life IV (1996)*; MIT Press: Cambridge, MA, USA, 1994; pp. 181–189.
20. Svennebring, J.; Koenig, S. Building terrain-covering ant robots: A feasibility study. *Auton. Robot.* **2004**, *16*, 313–332.
21. Purnamadjaja, A.; Russel, R. Bi-directional pheromone communication between robots. *Robotica* **2010**, *28*, 69–79.
22. Distante, C.; Indiveri, G.; Reina, G. An application of mobile robotics for olfactory monitoring of hazardous industrial sites. *Ind. Robot Int. J.* **2009**, *36*, 51–59.
23. Misawa, N.; Mitsuno, H.; Kanzaki, R.; Takeuchi, S. Highly sensitive and selective odorant sensor using living cells expressing insect olfactory receptors. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 15340–15344.
24. Dowling, K. Limbless Locomotion: Learning to Crawl. In Proceedings of IEEE International Conference on Robotics and Automation, Detroit, MI, USA, May 1999; Volume 4, pp. 3001–3006.
25. Hirose, S. *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*; Oxford University Press: Oxford, UK, 1993.
26. Meng, Y.; Zhang, Y.; Jin, Y. Autonomous self-reconfiguration of modular robots by evolving a hierarchical mechanochemical model. *IEEE Comp. Int. Mag.* **2011**, *6*, 43–54.