FSTT Algorithm: Can Tides Assist Bio-Inspired Gradient Taxis?

Joshua Cherian Varughese[1,2], Ronald Thenius[1], Franz Wotawa [2] and Thomas Schmickl[1]

[1]*Artificial Life Lab, Department of Zoology, Karl-Franzens-Universität Graz,*
e-mail: joshua.varughese@uni-graz.at
[2]*Institute for Software Technology, Techische Universität Graz*

Abstract. In this article we introduce a variation of the Firefly-Slime mold-Taxis (FSTaxis) algorithm, which is an emergent gradient ascent solution using external environmental influences such as tides, wind among others. Such external environmental influences are useful sources of energy for movement. If utilized, this results in substantial energy saving compared to robots relying solely on propulsion. Assistance using external factors can be adopted by various types of service robots depending on their environment of operation (for example, rescue robots, robotic underwater exploration). The variant of the FSTaxis algorithm we present in this paper combines bio-inspired communication strategies to achieve gradient taxis purely based on neighbor-to-neighbor interaction and tidal movements for mobility. In this article, we discuss the modified algorithm in detail and further introduce first simulation results obtained using a multiagent simulation environment.

Key words: gradient taxis, tides, self organization, swarm robotics, bio-inspiration

1 Introduction

In swarm robotics, a set of rules are employed in a swarm of robots to solve complex problems. Much research has been done in the recent past on swarm robotics to show how complex problems can be solved by a swarm of simple robots [10][12][2]. Swarm robotics has drawn inspiration from nature to mimic complex behaviors by using simple rules. Swarm intelligence has found applications in optimization problems in various fields[9] [8]. For example, multi-modal optimization [18] has taken inspiration from fireflies and such efforts have shown promising results. In project subCULTron [14], the vision is to develop an underwater society of learning, adapting, self-sustaining robots which can be used for various applications. Given the challenges of robotic systems underwater such as limited availability of classical communication systems, limited mobility etc., there need to be stable but simple solutions. Swarm robotics is an especially attractive solution for subCULTron and similar projects as it imposes minimum economic and computational resource demands while yielding solutions to complex problems[7].

Many algorithms have been presented in the past about possible emergent gradient taxis algorithms based on swarm behavior [11][2]. Since swarm intelligent solutions present a gradient taxis solution based on simple rules, the robots executing these tasks can be inexpensive and simple. Such algorithms are excellent solutions for robotics where mobility is independent and recharging does not pose an issue. However, for robots that need to explore inaccessible niches autonomously, (for example, rescue robots, robots in nuclear reactors, underwater robots where payload should be low) saving energy by any means is of paramount significance. With the context of such application areas, we will explore whether robots can use environmental influences for mobility.

In the past, a gradient taxis solution has been presented called the FSTaxis algorithm. This algorithm is a feasible bio-inspired solution for gradient taxis based on swarm behavior for underwater robotics. The FSTaxis algorithm is applicable to robots that have a means for independent movement. In this paper, we will formulate a variant of the FSTaxis algorithm which will enable the robots to utilize an external influence for movement. The following sections will explain the algorithm, the method used for testing the algorithm and corresponding results. This algorithm is an example of how emergent solutions can be made possible using an external assistance without using complex computation, large amount of memory and with minimum power consumption.

The objectives of this paper are as follows:

1. Describe a variant of the FSTaxis algorithm.
2. Test the algorithm in a gradient to show its gradient taxis capability.
3. Discuss strengths and weaknesses of the algorithm.

2 Inspiration

As mentioned in the introduction, the FSTaxis algorithm and its variant presented in this paper draws inspiration from nature. Before presenting the algorithm, we will discuss briefly the bio-inspiration for the parent algorithm. This section briefly discusses bio-inspired communication strategies used by slime mold and fireflies and also the motion strategy using an external environmental influence.

2.1 Biological Inspiration

Communication strategy of the FSTaxis algorithm draws inspiration from the aggregation phase of slime mold. Aggregation in slime molds occur in case of scarcity of food. During the aggregation phase, some slime mold cells (centers) release Cyclic Adenosine Mono Phosphate (cAMP) into the environment thus creating a temporary chemical concentration spike around them [13]. Concentration of cAMP diffuses rapidly into the environment and therefore the chemical spike is transitory. This chemical spike attracts other slime mold cells towards the centers. When surrounding cells perceive this chemical signal, they release cAMP themselves, thereby relaying the signal and then move towards the “recruiting centers”. This will start a cumulative process of the centers attracting slime mold cells far removed from the recruiting centers. The periodicity of the release of cAMP by the cells is 12-15 seconds [1]; in between releases, individual cells are insensitive to cAMP pulses in the environment. This interval can be understood as the refractory phase of the amoeba and forms the basis for refraction time in the FSTaxis algorithm. The signal relaying mechanism described above forms the basis for spatiotemporal patterns known as scroll waves [13]. The emergence of scroll waves enable the amoeba to move towards the recruiting centers for successful aggregation.

Another source of inspiration for the FSTaxis algorithm is fireflies. Fireflies have been a subject of elaborate studies in the past [3] for their spectacular bioluminescence and cooperative behavior for attracting mate or prey. Fireflies cooperatively blink in unison in order for the swarm to have a higher chance of attracting mates or prey [3]. A simple physical mechanism known as phase coupled oscillation synchronization is responsible for the cooperative blinking in fireflies. Initially, the individual fireflies blink randomly; when it perceives a blink in its surrounding, it blinks again and then resets its own frequency to match the other [4]. Eventually, the fireflies achieve complete synchronization.

2.2 External Environmental Influence

Forces naturally present in the environment like tides, wind, waves etc have been utilized by humans in different ways to harvest energy [15]. Many animals and plants also utilize such external influences for migration, reproduction etc. [5] [6]. If utilized, the external influences can save massive amounts of energy. Since this paper is written as part of research in aquatic robotics, focus will be on tides as the external influence. Tides produce complex water movement depending on several factors such as shape and composition of the shoreline, depth, etc. Water movement can be as simple as bidirectional to complex swiveling movements. For the sake of simplicity, we will model the tides to be bidirectional with a constant speed, v_t with heading changing periodically, p_t. Randomness in the heading of the tides is applied to simulate a scenario closer to the actual water movement.

3 The FSTaxis and Tide algorithm

The Firefly-Slime mold-Taxis and Tides (FSTT) algorithm makes use of the bio-inspired communication strategy of the FSTaxis algorithm. The behavior of agents in the FSTT algorithm can be broadly classified into *Ping behavior* and *Motion* *behavior*. The ping behavior of the FSTT algorithm corresponds exactly with that of the FSTaxis algorithm. Motion behavior, on the other hand, will differ considerably from the FSTaxis algorithm since FSTT algorithm uses only tidal movements for mobility. The only difference between the FSTaxis algorithm and its variant, the FSTT algorithm, is in its procedure for movement as seen in the Algorithm **??**. In the FSTaxis algorithm, an agent is self propelled and in the FSTT algorithm, the agent relies solely on external influences such as tides.

3.1 Ping behavior

Each agent has three communication states: “active”, “refractory” and “inactive” as shown in the state transition diagram Figure 1. Initially, all agents are in the inactive mode and have an internal countdown timer. The timer value is associated with its position in the environmental gradient. In the inactive mode, the agent monitors incoming pings. If the agent receives a ping, it itself goes into the pinging state where it broadcasts a ping for a period of time, t_p. During t_p, the agent is said to be pinging and after t_p, the agent enters the refractory mode for a period of time, t_r. In the refractory state, the agent ignores all incoming pings. After the refractory time, the agent sets itself back to inactive mode.

Each agent has an inherent cycle time determined by the environmental gradient at its position. If the internal timer(value associated with the environmental value at its physical position) of any agent counts to zero before a ping is received, the agent enters pinging mode and broadcast a ping. The agent sets its own ping frequency, f_p , by associating it with the gradient value at its position, g_p. This ping (”original ping”) is further relayed by the neighboring agents as per the ping behavior explained above. The agent that triggers the original ping (the agent whose f_p counted to zero) is referred to as the “leader” hereafter in this paper.

In order to scale ping frequencies to meaningful values, two preset maximum and minimum are selected for the gradient under consideration. Let these values be g_{max} and g_{min}. Equation 1 demonstrates the relation between ping frequency of agents and the environmental value under consideration. In equation 1, α and ω are constants that can be modified to achieve meaningful values of ping frequency.

$$f_p = \alpha + \frac{((g_p - g_{min})}{(g_{max} - g_{min})} * \omega \tag{1}$$

3.2 Motion behavior

Motion behavior in FSTT algorithm is dependent on the ping mode of the agent. An agent in inactive mode does not move. As shown in Figure 1, motion is initiated in the active mode. When any agent receives a ping it sets itself to active mode, sets its own heading towards the received ping and checks if the direction of external influence such as the tide is favorable. In order to check if the tide is favorable, the agent compares the direction of the incoming ping and the tide direction. If the tide has a component in this direction, the agent releases itself and moves with the tide at tide speed v_t for fixed distance β. A ping can only be perceived within the limited sensor range, s_r, of the robot, therefore it limits the number of agents that are able to affect any particular agent. In the scenario described above, it is possible that each agent receives multiple pings from different directions, h_n, where n is the number of agents pinging. In such as case, the agent will calculate the mean heading, h_{mean} , and set its heading towards this mean. Another scenario which will occur is that the agent finds that the tide is not suitable for movement. In this case, the agent remains fixed to the water body bed (by fixing itself on the ground) and continues to monitor incoming pings and storing their incoming direction. Once the tide is favorable, the agent will release itself and move in the mean direction of all incoming pings received while monitoring. If an agents internal clock triggers, an "original" ping based on the environmental value is broadcast; then, this agent labels itself the leader and does not move in that particular cycle.

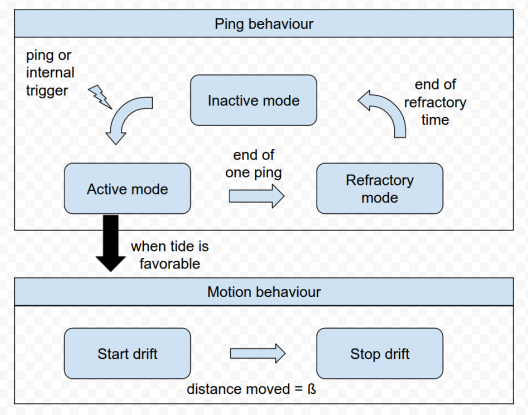

Fig. 1: The state transition diagram of the FSTT algorithm is shown here. The algorithm has two behaviors - ping behavior and motion behavior. There are three states in ping behavior: active, refractory and inactive. An agent transitions from inactive to active state when it receives a ping from another agent or when its own internal clock triggers. After the ping duration, the agent transitions into a refractory mode. After the refractory time, the agent transitions back into the inactive mode. While in active mode, if the tide is favourable, motion behavior is triggered and the agent moves a preset distance in the direction of the tide at tide speed.

When a swarm of agents execute the FSTT algorithm as per description above, scroll waves of pings similar to that in slime mold (as mentioned in Section 2.1) propagates through the swarm. Since the swarm is not free to move in every direction, the agents will move as close as possible to the most frequently received ping, which is the agent at the maximum environmental gradient value. This is due to the fact that the internal timer of the robot at highest gradient value will count to zero first, the direction of the wave will be from the higher to lower gradients. Therefore, the agents will want to move towards the agents with highest ping frequency but has to wait till the tide is favorable. Since a bi-directional tide model in Section 2.2 is considered, the agents will use the forward and backward movement to get as close as possible to "leader". When the agents are in their new position, their internal clock takes the values of the environmental factor (gradient value). Whichever agents internal clock triggers first becomes the leader and the swarm then gathers around this agent. This repeated choosing of leaders and gathering around the leader will draw the swarm towards areas with higher gradient value and in essence emerges into a gradient ascent.

4 Method

To demonstrate the gradient ascent capability by using tides, we choose a gradient with one maxima at the center. The equation of the test gradient is shown in Equation 2 and here, γ representing the radius of the gradient circle. Figure **??** shows the initial formation with 120 agents in netlogo. As previously mentioned, randomness has been added to the motion of both the heading and speed of tides in order to make the simulation as close to the real scenario as feasible. A circular gradient has been used to demonstrate the gradient (depth) ascent capability of the FSTT algorithm. In Figure **??**, shades of black has been used to depict the local depth value. Dark colored areas are deeper and the light colored areas are shoallower.

As mentioned before, the frequencies are scaled according to Equation 1 and Table 1 shows all the constants used in this experiment. The simulation environment used is Netlogo 4.3.1 [17]. In Netlogo, the test area is divided into "patches" (spatial units) and the agents are called "turtles". For the purpose of this experiment, depth is the physical quantity associated with each patch. The sensor radius of each agent is measured in patches and in this experiment it is taken to be three patches.

It is of merit at this point to describe the characteristics of turtles used for simulation. Each turtle is capable of choosing whether to move with the tide or not. In underwater robotics, this will mean that the robots have a mechanism to restrict themselves from moving with the water, for example, by fixing themselves to the water body bed. Each agent has a sensor to detect direction of tides in simulation. In a real scenario, this will translate to the underwater robots having a flow sensor. In addition to these capabilites, the turtles are able to sense the value of the gradient at its location, a sensor to detect incoming pings and also the ability to broadcast pings .

$$f(x) = \sqrt{x^2+y^2} - \gamma \qquad (2)$$

Algorithm 1 The FSTaxis algorithm

```
repeat
    procedure PING BEHAVIOR($t_p$, $t_r$,$v_a$,$t_f$)
        for all agents do
            if pingmode = refractory mode then
                count down $t_r$
                if $t_r = 0$ then
                    set state ← inactive mode
                    set $t_f \leftarrow 1/f_p$ - $Equation 1$
                    set leader status ← “OFF”
                end if
            end if
            if pingmode = active state then
                count down $t_p$
                if $t_p = 0$ then
                    set state ← refractory mode
                end if
            end if
            if pingmode = inactive mode then
                if any ping received? then
                    set state ← active mode
                    set move agent ← “ON”
                end if
            end if
            count down $t_f$
            if $t_f = 0$ then
                set state ← active mode
                set leader status ← “ON”
            end if
        end for
    end procedure
until $forever$


repeat
    procedure MOVEMENT(move agent, leader status)
        for all agents with movement = “ON” and leader status ! = “ON” do
            while $distancecovered < \beta$ do
                Create empty list, $l$
                for $i \leftarrow 1, no : ofpingsreceived$ do
                    append list $l \leftarrow h_i$
                end for
                calculate $h_{mean}$ of list, $l$
                set agent heading $h_a \leftarrow h_{mean}$
                if $h_{mean}$ - heading of current < 90 then
                    dive up
                    drift with the tide at speed $v_t$ for distance $\beta$
                    dive down
                end if
            end while
            set move agent ← “OFF”
        end for
    end procedure
until $forever$
```

Parameters							
Variable	ω	g_{max}	g_{min}	s_r	β	α	γ
Value	0.1	150	5	3	4	0.008	29.06
Units	-	m	m	patches	m	-	-

Table 1: Table showing all parameters used for demonstrating the FSTT algorithm

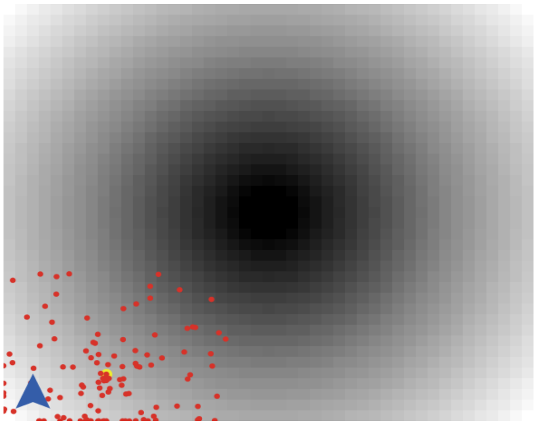

Fig. 2: Figure shows the initial formation with 120 agents in the simulation environment, netlogo. Here, the dark dots are agents and the gradient shown to the center of the arena represents the depth gradient. The bright circle shows the mean position of the entire swarm. The blue arrow on the bottom left corner represents the tide direction.

5 Results

Initially, as shown in Figure **??**, the swarm displays a linear formation purely by drift. Figure **??** shows how the robots further aggregate into small sub-swarms. Each of these sub-swarms is then able to navigate individually to the goal as seen in Figure **??**. The stages in aggregation and navigation towards the goal is shown in Figures **??**, **??**, **??**. After 50,000 iterations, the swarm reaches the deepest point of the environment (converges) to the goal and then oscillates around the goal as shown in Figure **??**. The bright marking shows the trajectory of the mean of all 120 agents. The back and forth motion is seen in the trajectory due to the bidirectional nature of the tide.

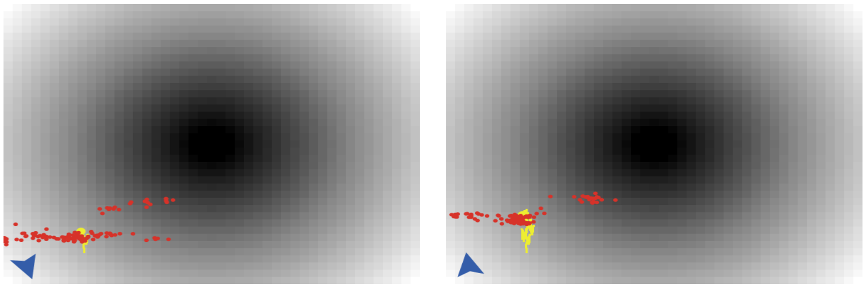

Fig. 3: The figures show intermediate formations by agents executing the FSTT algorithm in simulation. Initially, the agents form a linear formation. Here, the dark dots are agents and the gradient shown to the center of the arena represents the depth gradient. The bright marking shows the mean trajectory of the entire swarm. The blue arrow on the bottom left corner represents the tide direction. The dark areas shows the deeper zones and the light colored areas depict low depth.

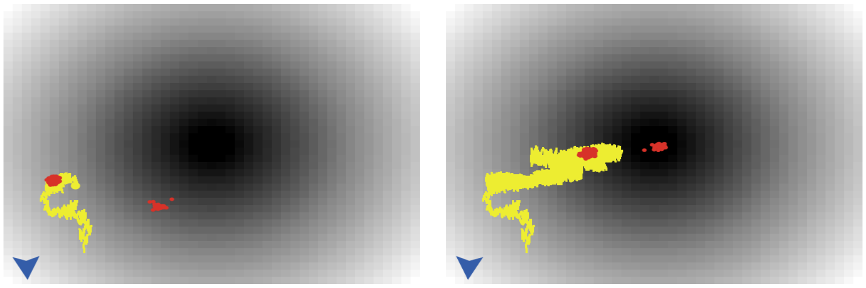

Fig. 4: The figures show intermediate formations by agents executing the FSTT algorithm in simulation. After the initial linear formation, the agents split into aggregates of smaller swarms. Here, the dark dots are agents and the gradient shown to the center of the arena represents the depth gradient. The bright marking shows the mean trajectory of the entire swarm. The blue arrow on the bottom left corner represents the tide direction. The dark areas shows the deeper zones and the light colored areas depict low depth.

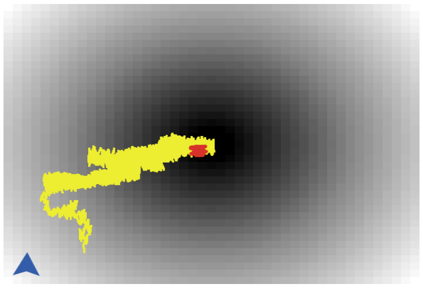

Fig. 5: This figure shows the final formation of agents executing the FSTT algorithm with all the agents having converged to the goal.

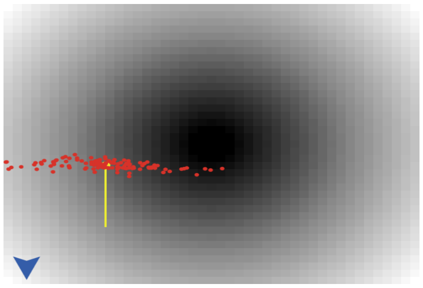

Fig. 6: This figure shows the formation of agents executing the FSTT algorithm with purely bidirectional tide. The agents form a line in the middle of the arena

It is also worth mentioning the effect on formation in case of the total absence of randomness in the tide. If tides were purely bidirectional without any randomness, it would mean that the agents have only two directions to move which are: forward and backward. In such as case, the expectation is that tides will bring the agents as close as possible to the goal. In this scenario, the agents will move from the initial position and form a line as shown in Figure **??**. The trajectory of the swarm in this case is a straight line due to absence of any sidewards movement by the agents.

6 Discussion

As seen in the Section **??**, the FSTT algorithm is able to successfully navigate a swarm of agents to the goal (highest environmental gradient value) by repeatedly using the water movement due to tides. It is also important to note that the FSTT algorithm uses no global knowledge for navigation. While being significantly slower than the FSTaxis algorithm , the energy saved is massive since it depends solely on tides for movement. This approach is valuable in salvaging all possible ways of saving energy.

Since the ping behavior requires the agents to move towards the mean ping, the swarm splits into sub-swarms. After splitting into sub-swarms, each sub-swarm is able to navigate individually to the goal and reunite with the there with the other sub swarms as per figures **??** and **??**.

Although tides in the real world are more complicated than a bidirectional tide, this simple tide model enables the demonstration of the idea presented in this paper. The assumption that tides are bidirectional does not hinder the implementation of the algorithm in more complicated scenarios. In case of more complicated water movements, the agents will still be capable of determining if the tides are favorable by determining the difference between the direction of the incoming ping and the direction of water movement.

As per Figure **??**, we see that randomness in water movement enables the agents to have more degrees of freedom than purely bidrectional freedom. Even though such a case is idealistic, results associated with purely bidirectional tides show that the agents are able to inch closer to the goal.

7 Conclusion

The FSTaxis algorithm forms the basis for the FSTT algorithm by taking advantage of tide movement. It is evident that FSTT algorithm is able to guide a swarm of robots to the highest point in a gradient with a single maxima. In the future, there is opportunity to validate the algorithm by adding noise to the gradient and investigating boundary conditions. Multi-modal gradients can be used to observe the behavior of the FSTT algorithms in presence of multiple maxima. It is expected that FSTT algorithm will perform similar to the FSTaxis algorithm .

It is also worth mentioning that the tide model assumes a period of 50 iterations. In reality, tides reverse only twice in a day. Convergence at around 50,000 iterations mean that the convergence time we are looking at runs into several months. This long convergence time is possibly the result of overly strict simulation environment of tides. While the basic capability of the algorithm for gradient ascent is evident from simulation, the test of feasibility of the hardware requirements such as ping broadcasting technology and water current measurement are still a focus of ongoing research. In the future, feasibility tests of this algorithm will be done with real world hardware. Nevertheless, the FSTT algorithm based on the results presented in Figure **??** is a novel, innovative and energy saving solution for underwater robots.

By learning the nature of tides or other external influences present in the environment, agents will be able to move closer to the goal. If such a capability is combined with independent motion capability, it will result in massive energy savings and enable long term autonomy of robots in difficult environments. FSTT algorithm implementation can be also considered as a partial solution for independent mobility instead of a sole mobility solution.

Acknowledgements This work was supported by EU-H2020 Project no. 640967, subCULTron, funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No 640967.

References

1. Alcantara, F., Monk, M.: Signal propagation during aggregation in the slime mould dictyostelium discoideum. Microbiology **85**(2), 321–334 (1974)
2. Bjerknes, J.D., Winfield, A., Melhuish, C.: An analysis of emergent taxis in a wireless connected swarm of mobile robots. In: IEEE Swarm Intelligence Symposium, pp. 45–52. IEEE Press, Los Alamitos, CA (2007)
3. Buck, J., Buck, E.: Biology of synchronous flashing of fireflies. Nature **211**, 562–564 (1966)
4. Camazine, S., Denenbourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Synchronized flashing among fireflies. pp. 143–166. Princton University Press, Princton (2001)
5. Chapman, J.W., Nesbit, R.L., Burgin, L.E., Reynolds, D.R., Smith, A.D., Middleton, D.R., Hill, J.K.: Flight orientation behaviors promote optimal migration trajectories in high-flying insects. Science **327**(5966), 682–685 (2010). DOI 10.1126/science.1182990. URL http://science.sciencemag.org/content/327/5966/682
6. Friedman, J., Barrett, S.C.: Wind of change: new insights on the ecology and evolution of pollination and mating in wind-pollinated plants. Annals of botany (2009)
7. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks. IEEE Press, Los Alamitos, CA (1995)
8. Nakagaki, T.: Smart behavior of true slime mold in a labyrinth. Research in Microbiology **152**, 767–770 (2001)
9. Schmickl, T., Crailsheim, K.: A navigation algorithm for swarm robotics inspired by slime mold aggregation. In: E. Şahin, W.M. Spears, A.F.T. Winfield (eds.) Swarm Robotics - Second SAB 2006 International Workshop, Lecture Notes of Computer Science, pp. 1–13. Springer-Verlag, Berlin, Heidelberg, New York (2007)

10. Schmickl, T., Hamann, H.: BEECLUST: A swarm algorithm derived from honeybees. In: Y. Xiao (ed.) Bio-inspired Computing and Communication Networks. CRC Press (2011)
11. Schmickl, T., Hamann, H., Stradner, J., Mayet, R., Crailsheim, K.: Complex taxis-behaviour in a novel bio-inspired robot controller. In: H. Fellermann, M. Dörr, M.M. Hanczyc, L.L. Laursen, S. Maurer, D. Merkle, P.A. Monnard, K. Støy, S. Rasmussen (eds.) Proc. of the ALife XII Conference, pp. 648–655. MIT Press (2010)
12. Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., Crailsheim, K.: Get in touch: Cooperative decision making based on robot-to-robot collisions. Autonomous Agents and Multi-Agent Systems **18**(1), 133–155 (2008)
13. Siegert, F., Weijer, C.J.: Three-dimensional scroll waves organize dictyostelium slugs. PNAS **89**(14), 6433–6437 (1992)
14. subCULTron: Submarine cultures perform long-term robotic exploration of unconventional environmental niches (2015). http://www.subcultron.eu/
15. Turner, J.A.: A realizable renewable energy future. Science **285**(5428), 687–689 (1999). DOI 10.1126/science.285.5428.687. URL http://science.sciencemag.org/content/285/5428/687
16. Varughese, J.C., Thenius, R., Wotawa, F., Schmickl, T.: Fireslime algorithm: Bio-inspired emergent gradient taxis. In: Proceedings of the Fifteenth International Conference on the Synthesis and Simulation of Living Systems. MIT Press. Submitted
17. Wilensky, U.: Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL (1999). URL http://ccl.northwestern.edu/netlogo/
18. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Stochastic algorithms: foundations and applications, pp. 169–178. Springer (2009)