

A Novel Bio-Inspired Distributed Coverage Controller for pollution monitoring

John Oyekan, Huosheng Hu and Dongbing Gu
School of Computer Science and Electronic Engineering,
University of Essex, Wivenhoe Park, Colchester CO3 4SQ, United Kingdom.
Email: jooyek@essex.ac.uk, hhu@essex.ac.uk, dgu@essex.ac.uk

Abstract—The Voronoi partition method can provide a coverage to an area of interest and has gained a lot of popularity when compared with other coverage schemes such as virtual springs and deterministic annealing. This paper presents the development of a novel bio-inspired algorithm that uses a behaviour based approach to solve the problem of coverage. It is shown that by combining a source seeking behaviour - bacteria chemotaxis and a group foraging behaviour - flocking, it is possible to provide coverage to an area of interest and obtain results similar to that obtained by using the Voronoi partition method. Experimental results show that the novel coverage algorithm can out perform the voronoi partition method when presented with some case study scenarios.

Index Terms—Voronoi Partition, Bacterial Algorithm, Environmental monitoring, Nature inspired Algorithms, Optimal Coverage.

I. INTRODUCTION

Recently, environmental protection has become an urgent task and researchers have created various useful and capable coverage algorithms, including deterministic annealing [1], virtual springs [2], Smoothed Particles Hydrodynamics [3], Voronoi partitions [4] and bacteria behaviour[5]. The deterministic annealing algorithm developed by kwok et al in [1] can provide coverage to the distribution of a spatiotemporal function, and its accuracy depends on obtaining the best heating and cooling cycle. The approach is also dependent on a heavy communication scheme in which commands are flood over the network of agents. Shucker et al used mathematical equations that describe the length of springs under a force and developed a mechanism to control the distribution of agents tracking a complex diffusing target [2]. This approach required long distance communication and might not be suitable for simple robotic agents with limited hardware.

In [3], Pac et al used ideas from computational fluid dynamics to generate a controller that enabled agents to move like a fluid in the environment and provide coverage to it. The behaviour of bacteria was used by Mesquita et al [5] to develop a controller capable of providing coverage of a spatiotemporal distribution without taking collisions into account. However, from all the above mentioned approaches, the use of the Voronoi partition technique has gained a lot of popularity and has been used in various forms as in

[6][7][8][9][10].

In [11], bacteria behaviour and flocking behaviour are combined to provide coverage to an environment containing a simulated invisible hazardous substance. It was shown that the approach was capable of providing a visual representation of the invisible hazardous substance no matter the distribution or shape of the pollution. In order to validate this approach further, it is compared with the Voronoi partition method in this paper by presenting both techniques with various simulated scenarios and then analysing the data of how well they performed.

The rest of the paper is organized as follows. Section II introduces the coverage controllers to be compared in this work. Section III discusses how both approaches were compared and the results were obtained. Section IV discusses both future work and conclusions obtained during the experimental work carried out in this paper.

II. THE COVERAGE CONTROLLERS

A. *BactFlock Controller*

The BactFlock controller is developed by using a behaviour based paradigm that combines bacteria and flocking behaviours using vector addition and gains to control the output of each behaviour. The advantages of the paradigm include reactivity to dynamic changes in the environment, flexibility to add other behaviours necessary to accomplish a task, ability to add a machine learning component if needed to tune the parameters of the behaviour to obtain more optimal coverage[12]. The reactive feature of the paradigm makes the approach capable of responding to an ever changing pollutant or spatiotemporal profile. We shall now discuss each of the two behaviours in detail.

1) *Bacteria Behaviour*: The bacteria behaviour is inspired by the chemotaxis foraging of bacteria. A bacteria forages for food in the environment by navigating up the gradients of the food's chemical signal. The motion of the bacteria is composed of a combination of straight runs and tumbles to a random angle. As the food's chemical signal gets stronger, the likelihood of a tumble reduces causing longer straight runs, i.e. causing the bacteria to exhibit a biased random walk towards the food source. In the absence of food chemicals,

the bacteria's motion becomes a random walk which has been proven to be equivalent to diffusion (Readers can look at [13] for the proof).

The diffusion capability of the bacteria in the absence of a food chemical signal makes it an ideal explorer of the environment. A large population of bacteria in an environment containing multiple food sources have a chance of finding each of the food sources due to their exploratory behaviour.

The foraging behaviour of the bacteria has been modelled by various robotic researchers mostly by using rule based approach such as [14] in their quest to develop a controller for a single robotic agent. Dhariwal et al [15] were able to develop a bacteria controller for a single agent by using the Keller Segel model. How this was done is not clear. In this work, we develop a bacteria controller for a single robotic agent based on the Berg and Brown model which describes the behaviour of a single bacterium.

One of the advantages of using biology models to develop controllers for agents is that their performance could be easily compared to their biological counterparts. In addition, by understanding the model and adjusting various parameters, various capabilities of the biological counterparts could be assessed and harnessed on robotic agents. The use of models also opens up the ability to analytically access the robotic agent's behaviour in response to changes in the biological controller parameters.

The Berg and Brown model is presented in equations 1 to 3 and has been used on a physical agent in our work of [16] to show that it is capable of finding the source of a spatiotemporal quantity in a diffusion based or low turbulent environment.

$$\tau = \tau_o \exp\left(\alpha \frac{\overline{dP_b}}{dt}\right) \quad (1)$$

$$\frac{\overline{dP_b}}{dt} = \tau_m^{-1} \int_{-\infty}^t \frac{dP_b}{dt'} \exp\left(\frac{(t' - t)}{\tau_m}\right) dt', \quad (2)$$

$$\frac{dP_b}{dt} = \frac{k_D}{(k_D + C)^2} \frac{dC}{dt} \quad (3)$$

where τ is the mean run time and τ_o is the mean run time in the absence of concentration gradients, α is a constant of the system based on the chemotaxis sensitivity factor of the bacteria, P_b is the fraction of the receptor bound at concentration C . In our work, C was the present reading taken by our Robotic agent. K_D is the dissociation constant of the bacterial chemoreceptor. $\frac{dP_b}{dt}$ is the rate of change of P_b .

$\frac{\overline{dP_b}}{dt}$ is the weighted rate of change of P_b , while τ_m is the time constant of the bacterial system. The above equations determine the time between tumbles and hence the length of runs between tumbles. During the tumble phase, the agent can randomly choose a range of angles in the set $\sigma \in \{0, \dots, 360\}$ by randomly choosing angles. This made it possible for our agents to backtrack if there is a favourable gradient behind

it. For details on how to tune a robotic agent using the Berg and Brown source controller, the reader is referred to [16].

2) *Flocking Behaviour*: Foraging in a group enables organisms to find richer food sources in the environment. Looking at this biological phenomenon from a technical point of view, one could say that flocking enables organisms escape from "food local maximas" in the environment. Flocking was first modeled by Reynolds in 1986 and the model has been used by various researchers in various forms as can be seen in [17]. In this work, the flocking behaviour is implemented by using the generalized Morse potential as follows.

$$F_{output}^i = G_G^i * [G_R^i * \exp(-r/20) - G_A^i * \exp(-r/20)] \quad (4)$$

where G_R , G_A are the repulsion and attractive terms respectively. The gain G_G controls how closely the agents get to each other for a set values of G_R and G_A .

In order to combine both the bacteria behaviour and the flocking behaviour, the equation 5 is used with gains of G_F and G_B applied to both the flocking controller and bacteria controller outputs respectively. In this work, each agent is able to buffer up 5 of its closest neighbour's cartesian position in a user defined communication radius in order to compute the most closest agent to it. This information is then used to maintain a comfortable distance from that neighbour.

$$Output = F_{output}^i * G_F + B_{output}^i * G_B \quad (5)$$

3) *Velocity Controller*: In the survey carried out by Tindal et al [18], it was discussed that one of the ways of achieving simulation accuracy of the chemotactic band observed by bacteria colonising a food source was by including a velocity function component. Depending on the velocity function used, the simulated bacteria bands change shape. As a result of this phenomenon, a velocity controller shown in Equation 6 was embedded in the bacteria controller in order to provide coverage.

$$\beta^i(t) = \frac{\beta_o * T}{(C(t))} \quad (6)$$

where T can be viewed as a system temperature constant and can be used for tuning the system using an adaptable scheme. By controlling this constant, the spread of the agents in the environment can be controlled to achieve different coverage levels. $\beta^i(t)$ is a dynamic velocity of agent i that depends on the present reading $C^i(t)$ of the environmental quantity, β_o is the standard velocity without any reading.

This velocity function was embedded in the bacteria controller. The present reading $C^i(t)$ of the agent adapts the velocity of the agent so that in an area of higher concentration, it moves slowly covering a smaller area and vice versa. From the above, it can be seen that the BactFlock method has a lot of gain values which can be adjusted to achieve various coverage levels depending on the user requirements.

B. Voronoi Partition

The use of Voronoi partitioning in robotics was pioneered by Cortes et al [4]. They showed how a group of robotic agents P could be controlled to achieve optimal coverage of a simulated spatiotemporal distribution $\phi(q)$ in an area Q . By dividing the area Q into Voronoi cells using the individual robotic agents positions- Equation 7, the mass density each Voronoi cell is calculated using the spatiotemporal quantity in the cell ρ - Equation 8. From the mass density, the position of the centre of mass of the Voronoi cell is calculated using Equation 9 and then the robot is moved to this position using equation 10 assuming Equation 11 dynamics.

By doing this, the agents navigate up the spatiotemporal distribution towards its source whilst providing coverage. This approach has the advantage of incorporating both the flocking controller and the bacteria controller discussed in sectionII-A above into one controller. It also saves a user of this approach from having to do endless tuning of the controller. However in order to be able to use this approach, the sensor used by the robotic agents must have a radius in which they sense the spatiotemporal quantity in their immediate vicinity.

$$V_i = \{q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i\} \quad (7)$$

$$M_V = \int_{V_i} \rho(q) dq \quad (8)$$

$$C_V = \frac{1}{M_V} \int_{V_i} q \rho(q) dq \quad (9)$$

$$u_i = -k_{prop}(p_i - C_{V_i}) \quad (10)$$

$$\dot{p}_i = u_i \quad (11)$$

This approach can be viewed as trying to minimize the cost function Equation 12 where $f(\cdot)$ could be any function used to simulate the cost of the robotic agent's sensor being far away from the position q . For information on the voronoi partition technique the reader is referred to [4].

$$M_V = \sum_{i=1}^n \int_{V_i} f\|q - p_i\| \phi(q) dq \quad (12)$$

Because of the computational demand in measuring M_V and C_V for every voronoi cell V_i and because of this is unrealistic in a real life scenario, we use a radius value to specify the range of the measurements q to collect for each region. Using this approach reduces the computational burden especially when a large area is to be monitored.

III. SIMULATION AND RESULTS

In this section, comparison is made between the two techniques of providing coverage to an environment under investigation using three test metrics- Convergence speed, Coverage of smooth functions, and Coverage of noisy functions where the approaches are presented with noisy spatiotemporal

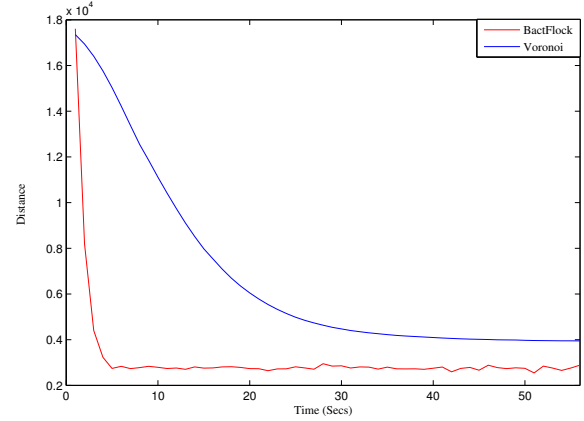


Fig. 1. Rate of convergence of the BactFlock coverage controller and Voronoi coverage controller.

distributions. For all the experiments in this section, 50 agents were used except otherwise stated and $k_{prop} = 1$ was used for the voronoi method.

A. Convergence Speed

The ability of an algorithm to converge quickly at the source of a spatiotemporal function could be very useful especially when the spatiotemporal quantity is hazardous. In order to compare the convergence speed of both approaches, a 2D Gaussian function having parameter values of $mean_x = 150$, $mean_y = 150$, $\sigma_x = 125$, $\sigma_y = 125$ and $Amplitude = 50$ was used while the agents were distributed at position $(x, y) = (350, 350)$ with a standard deviation of $(100, 100)$. The rate of convergence of the flock centre of agents to the mean position of the Gaussian function as in equation 13 was used as a measure of the rate of convergence for both algorithms.

$$swarm_{centre} = \frac{1}{N} \sum_{i=1}^N X_i \quad (13)$$

where N is the number of agents in the swarm and X is the position of the individual agents. The velocity β of the bacteria agents were set so that they did not go above the value of 1 so as to get a fair comparison with the voronoi partition method. $G_G = 10$, $k_d = 2$, $\alpha = 2$ and $\tau_o = 2$ were used with a communication radius and sensor coverage radius of 20 respectively used for the BactFlock and voronoi methods.

The BactFlock method as shown in Figure 1 was faster than the voronoi method. This could be as a result of the computations required by the voronoi method to obtain the centre of mass of the voronoi cells whereas the BactFlock method does not require this computation.

B. Ability to escape local maxima

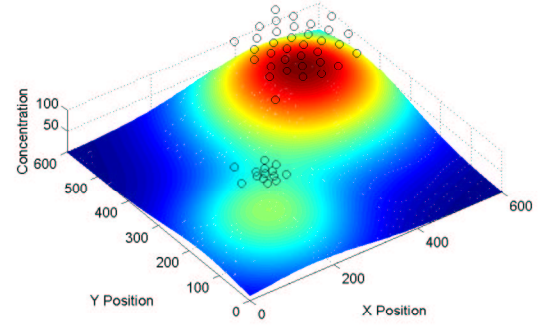
The ability to escape local maxima and find the global maxima is a challenge to most optimization algorithms. The ability to find the global maxima is very useful so that more sensors could be deployed to that area in order to capture more data. In order to test both approach's ability to escape local maxima, two Gaussian functions were placed in the simulated environment. The Gaussian functions had parameters values of $mean_x = 150$, $mean_y = 150$, $\sigma_x = 25$, $\sigma_y = 25$, $Amplitude = 50$ and $mean_x = 450$, $mean_y = 450$, $\sigma_x = 150$, $\sigma_y = 150$, $Amplitude = 100$. The agents were placed at positions of $(x, y) = (50, 50)$ with a standard deviation of $(100, 100)$. The G_G parameter of the BactFlock method was set to 10 as before. It was discovered that the agents using the voronoi method were trapped in a local maxima as shown in Figure 2 and did not change position even as time goes on to infinity.

However the agents using the BactFlock technique were able to escape the local maxima and still search the environment until they were finally able to distribute themselves in accordance to the distribution of the simulated spatiotemporal function in the environment. A Kullback-Divergence measure, Equation 14 (Where R is the number of grids the simulated environment was divided into. q is the concentration reading at the position of the agents and p is the Gaussian function.), was used to investigate the measure of coverage provided by both algorithms. As can be seen in Figure 2(c), the Voronoi method provided less coverage as a result of it getting stuck in the local maxima when compared to the BactFlock method. In addition, the graph shows that the rate of convergence for the BactFlock method was faster than that of the Voronoi method when two Gaussian functions were present in the environment. This could be due again to the exploration ability of the BactFlock method. In this experiment, a communication radius and sensor coverage radius value of 40 were also used.

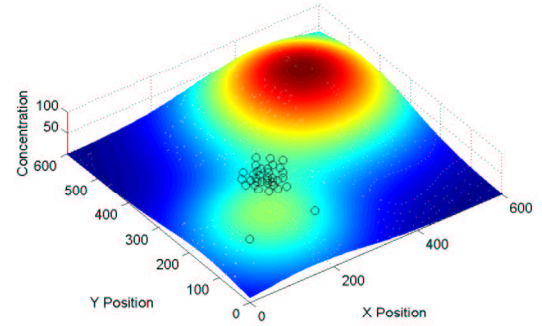
$$KL_{measure} = \sum_{i=1}^R \left[q_i \log \frac{q_i}{p_i} \right] \quad (14)$$

C. Coverage of smooth functions with similar peaks

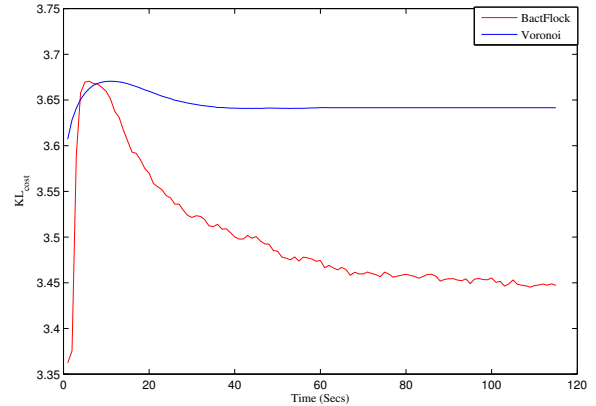
The ability of a coverage algorithm to distribute agents in the environment according to the spatiotemporal quantity distribution is very important especially if the spatiotemporal quantity is to be represented accurately visually. This is especially true if the spatiotemporal quantity as similar peaks but having different masses as a result of different spreads. In order to test this for both approaches compared in this paper, two Gaussian functions with similar peaks but different standard deviations were used. The parameters of the Gaussian functions were $mean_x = 150$, $mean_y = 150$, $\sigma_x = 125$, $\sigma_y = 125$, $Amplitude = 100$ and $mean_x = 450$, $mean_y = 450$, $\sigma_x = 50$, $\sigma_y = 50$, $Amplitude = 100$.



(a)



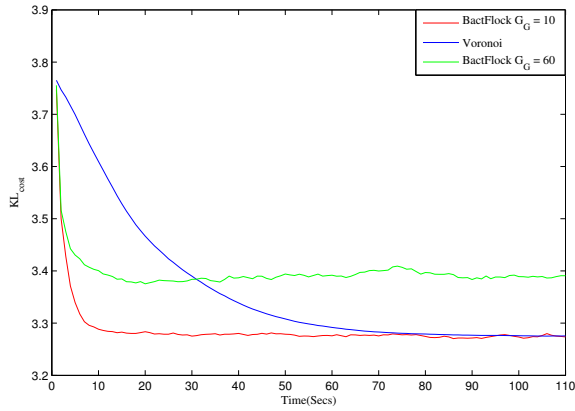
(b)



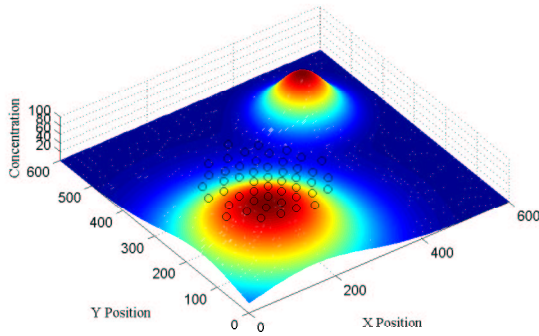
(c)

Fig. 2. Robotic agent coverage of two Gaussian functions using Fig. (a)- BactFlock Method, $G_G = 60$, $\beta_o = 32$, $k_d = 2$, $\alpha = 2$, Fig. (b)- Voronoi Method and Fig. (c) showing the difference in the Kullback Leiber cost obtained by both methods.

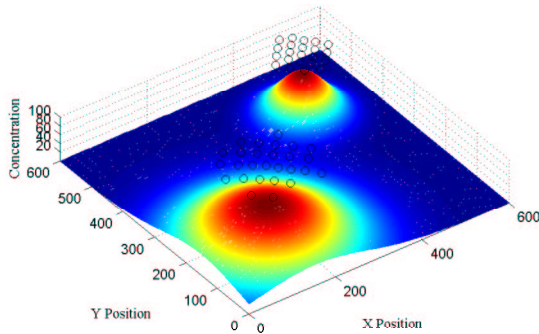
Agents were placed in between the Gaussian functions at $(x, y) = (300, 300)$ with a standard deviation of $(x, y) = (100, 100)$ so that the voronoi method does not get trapped in local



(a)



(b)



(c)

Fig. 3. Comparing coverage for the Voronoi and BactFlock method. Fig 3(a)-KL cost for Voronoi and BactFlock; Fig 3(b) and Fig 3(c) shows the distribution of agents for BactFlock and Voronoi respectively after approximately two minutes.

maxima.

It can be from the results in Figure 3 that BactFlock method was not able to distribute agents to cover the Gaussian function $mean_x = 450$, $mean_y = 450$, $\sigma_x = 50$, $\sigma_y = 50$, $Amplitude = 100$ but generated a slightly lower

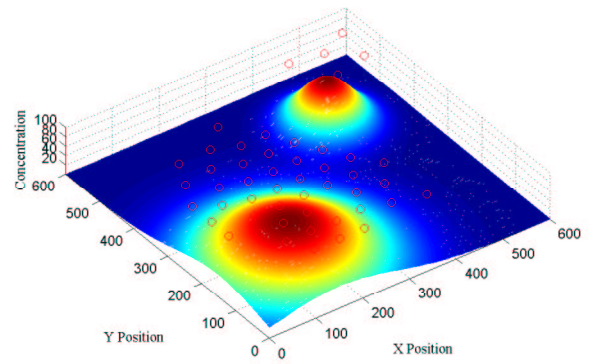


Fig. 4. Adjusting BactFlock coverage controller for better coverage.

cost when compared to the Voronoi method which was able to cover both functions. This is because most agents were distributed in the Gaussian function having the bigger amplitude thereby reducing the KL cost according to equation 14. The problem of not detecting the smaller Gaussian can be partly solved by increasing the G_G gain from 10 to a value of 60 as shown in Figure 4. By doing this the agents will be repelled more strongly from each other and hence explore the environment more. The KL cost of doing this as seen in Figure 3(a) is higher due to the spread of the agents in the environment resulting in less agents around the highest readings of the simulated spatiotemporal quantity. Focusing on the voronoi method results, it can be seen that even though it was able to distribute the agents to both Gaussian functions, the ratio of the agents might not be proportionate to the amount of spatiotemporal quantity under each Gaussian.

D. Coverage of noisy functions

In order to test both approaches in a more real life scenario, noisy functions were generated by randomly placing particles in the environment as shown in Figure 5 at location $(x, y) = (350, 350)$ while agents were placed at $(x, y) = (50, 50)$ at the start of the experiment. For the BactFlock approach agents were able to measure concentration at their position by counting the amount of particles across their body length. As each agent had a body length of 10 by 10 pixels, the highest concentration reading will be 100 whilst the lowest will be 0. The communication radius with flocking agents for the BactFlock approach was 40. Whereas for the voronoi partition method, the agents had a sensor radius of 40.

As can be seen in Figure 5(a), since the voronoi partition method did not have any exploration capability, this made it difficult for the sensors to take any reading from the environment and hence form a visual distribution whereas this was not a problem to the BactFlock method as seen in Figure 5(c) and 5(d). Having noticed this, agents for the voronoi method were then shifted to the fringes of the

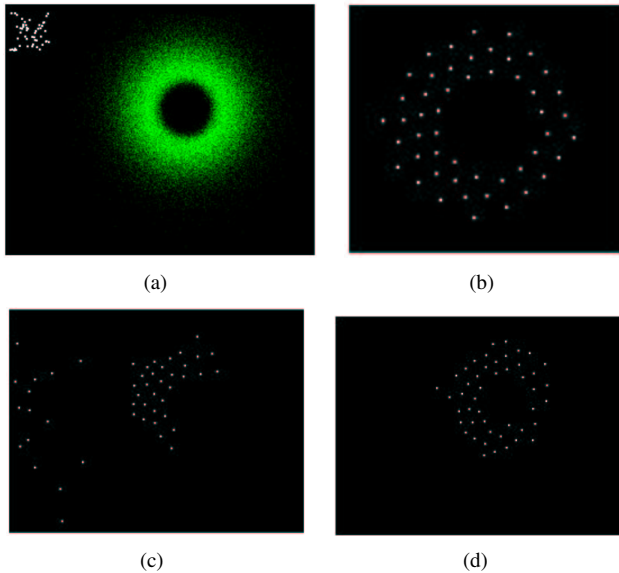


Fig. 5. Robotic agent coverage of a Doughnut function using Fig. (a)-Voronoi Method with agents at $(x, y) = (50, 50)$, Fig. (b)- Voronoi Method with agents at $(x, y) = (250, 250)$ after 8 minutes, Fig. (c)- BactFlock Method with agents at $(x, y) = (50, 50)$ after 8 minutes and Fig. (d) BactFlock Method with agents at $(x, y) = (50, 50)$ after 20 minutes.

noisy function at $(x, y) = (250, 250)$ and resulted in the distribution shown in Figure 5(b). The final results of the Voronoi partition method were visually similar to the results obtained by the BactFlock method.

IV. CONCLUSION AND FUTURE WORK

This paper presents a comparison between the Voronoi partition method of coverage and a novel behaviour based coverage algorithm. It is clear that the Voronoi partition method lacks the capability to explore its environment and presents a computational burden as a result of the need to compute the centre of mass of the voronoi cells for each agent as where as the BactFlock method has an embedded exploration capability that aids rich data collection in addition to using a computational efficient point measurement system. The Voronoi partition method has a higher KL cost in some test case scenarios when compared to the BactFlock method and also gets trapped in local maxima. This can be partly solved by using machine learning to learn the spatiotemporal function as in [7] leading to an additional increase in computation. Whereas, the local maxima problem can be easily solved by tuning gains in the BactFlock method.

Our future work is to use the sensory data collected by the BactFlock method to build a map of the spatiotemporal function for navigating deterministically in the environment. We will also investigate the ways of improving both schemes: (i) To use the BactFlock method to initially investigate the environment containing multiple functions and then switch to the Voronoi partition to form the final distributions after a set

time defined by the user; (ii) To use the BactFlock entirely and use a high system temperature initially and then slowly reduce the temperature β of the system as time progresses.

V. ACKNOWLEDGEMENTS

This research work is financially sponsored by European Union FP7 program, ICT-231646, SHOAL

REFERENCES

- [1] A. Kwok and S. Martinez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *Proc. of American Control Conference, St. Louis, Missouri, USA*, no. 1, pp. 1448–1453, 2009.
- [2] B. Shucker, T. Murphey, J. K. Bennett, and S. Member, "Convergence Preserving Switching for Topology Dependent Decentralized Systems," *IEEE Transactions on Robotics*, vol. 24, pp. 1–11, 2008.
- [3] M. R. Pac and A. M. Erkmen, "Control of robotic swarm behaviors based on smoothed particle hydrodynamics," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4194–4200, Oct. 2007.
- [4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [5] A. R. Mesquita, J. P. Hespanha, and K. Astrom, "Optimotaxis: A Stochastic Multi-agent Optimization Procedure with Point Measurements," *Hybrid Systems: Computation and Control*, vol. 4981, pp. 358–371, 2008.
- [6] M. Schwager, J. J. E. Slotine, and D. Rus, "Consensus learning for distributed coverage control," *In Proceedings of International Conference on Robotics and Automation, Pasadena, CA, May, 2008*.
- [7] M. Schwager, J.-j. Slotine, and D. Rus, "Decentralized, Adaptive Control for Coverage with Networked Robots," *IEEE Int. Conf. on Robotics and Automation*, pp. 3289–3294, 2007.
- [8] R. A. Cortez, H. G. Tanner, and R. Lumia, "Distributed Robotic Radiation Mapping," *Springer Tracts in Advanced Robotics*, vol. 54, pp. 147–156, 2009.
- [9] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. S. Pereira, "Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks," *Springer tracts in advanced robotics*, vol. 8, pp. 1–16, 2010.
- [10] J. McClurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," *Distributed Autonomous robotic systems*, Springer, vol. 6, no. 9, pp. 399–408, 2007.
- [11] J. Oyekan, H. Hu, and D. Gu, "Bio-Inspired Coverage of Invisible Hazardous Substances in the Environment," *International Journal of Information Acquisition*, vol. 07, no. 03, p. 193, 2010.
- [12] R. C. Arkin, *Behaviour-based Robotics*. The MIT Press, Cambridge, Massachusetts, London, England, 1998.
- [13] J. Shi, "Derivation of Reaction-Diffusion Equations," *Lecture Notes in Partial Differential Equations and Mathematical Biology*, no. Spring, 2006.
- [14] L. Marques, U. Nunes, and T. D. Almeida, "Olfaction-based mobile robot navigation," *Thin Solid Films*, vol. 418, no. 1, pp. 51–58, 2002.
- [15] A. Dhariwal, G. S. Sukhatme, and A. A. G. Requicha, "Bacterium-inspired robots for environmental monitoring," *Proc. of IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, April*, pp. 1436–1443, 2004.
- [16] J. Oyekan and H. Hu, "Bacteria Controller Implementation on a Physical Platform for Pollution Monitoring," in *Proc. of IEEE Int. Conf. on Robotics and Automation, Anchorage, Alaska, USA, May 3-8*, 2010, pp. 3781–3786.
- [17] R. Olfati-saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 03, pp. 401–420, 2006.
- [18] M. J. Tindall, P. K. Maini, S. L. Porter, and J. P. Armitage, "Overview of Mathematical Approaches Used to Model Bacterial Chemotaxis II: Bacterial Populations," *Bulletin of Mathematical Biology*, pp. 1570–1607, 2008.