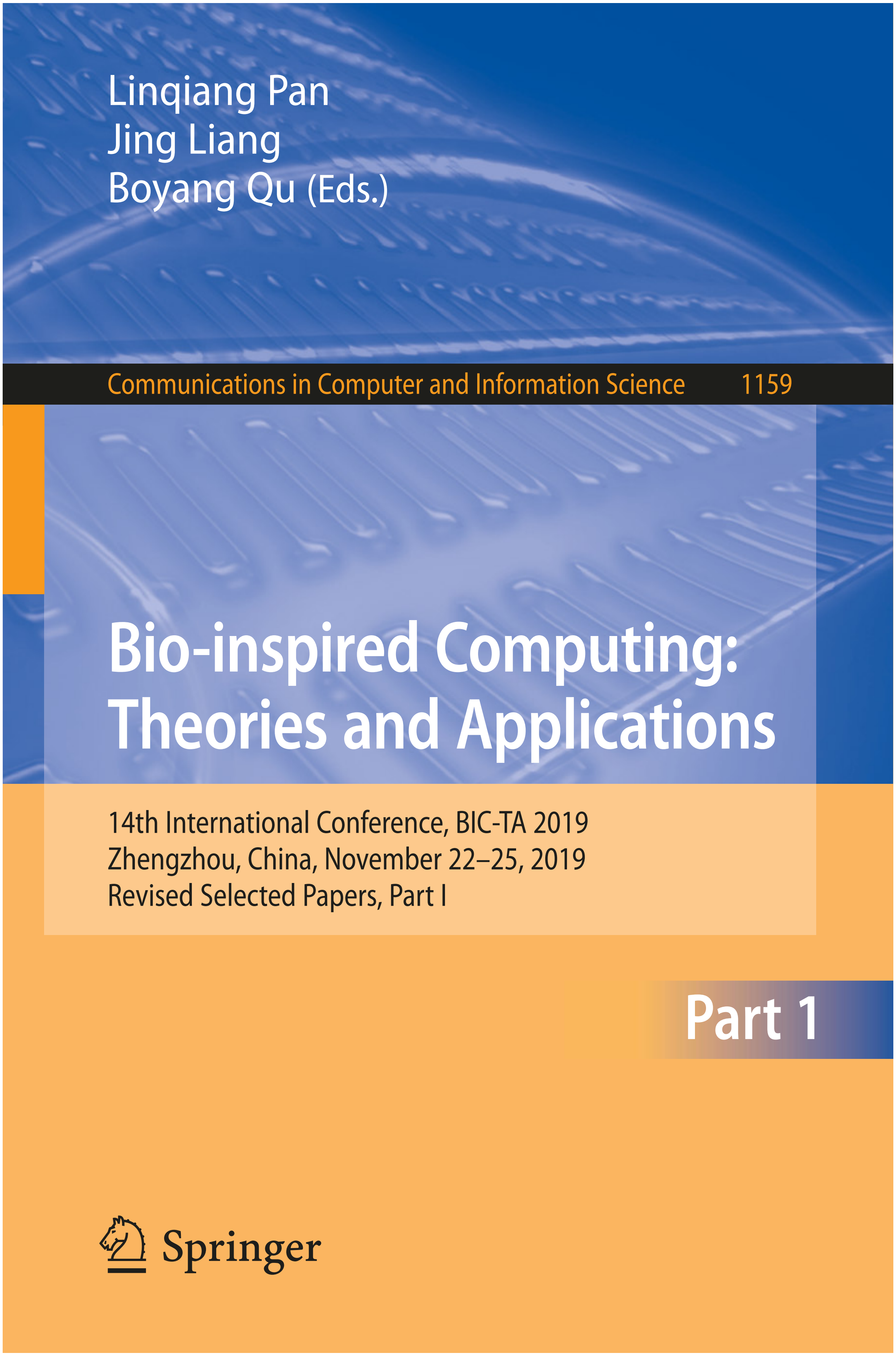

Linqiang Pan
Jing Liang
Boyang Qu (Eds.)

Communications in Computer and Information Science 1159

Bio-inspired Computing: Theories and Applications

14th International Conference, BIC-TA 2019
Zhengzhou, China, November 22–25, 2019
Revised Selected Papers, Part I

Part 1

Springer

Communications in Computer and Information Science 1159

Commenced Publication in 2007

More information about this series at http://www.springer.com/series/7899

Linqiang Pan · Jing Liang · Boyang Qu (Eds.)

Bio-inspired Computing: Theories and Applications

14th International Conference, BIC-TA 2019
Zhengzhou, China, November 22–25, 2019
Revised Selected Papers, Part I

Springer

Editors
Linqiang Pan
Huazhong University of Science
and Technology
Wuhan, China

Jing Liang
Zhengzhou University
Zhengzhou, China

Boyang Qu
Zhongyuan University of Technology
Zhengzhou, China

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-15-3424-9 ISBN 978-981-15-3425-6 (eBook)
https://doi.org/10.1007/978-981-15-3425-6

© Springer Nature Singapore Pte Ltd. 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

11. Liang, J., Yue, C., Qu, B.: In 2016 IEEE Congress on Evolutionary Computation. CEC 2016 (Institute of Electrical and Electronics Engineers Inc.), pp. 2454–2461 (2016)
12. Liang, J., Yue, C., Qu, B.: Multimodal multi-objective optimization: a preliminary study. In: Evolutionary Computation, pp. 2454–2461 (2016)
13. Liu, Y., Yen, G., Gong, D.: A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies. IEEE Trans. Evol. Comput. **23**(4), 660–674 (2019)
14. Pan, L., Cheng, H., Ye, T.: A region division based diversity maintaining approach for many-objective optimization. Integr. Comput. Aided Eng. **24**(3), 279–296 (2017)
15. Pan, L., Li, L., He, C.: A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization. IEEE Trans. Cybern (2019, in press)
16. Rudolph, G., Naujoks, B., Preuss, M.: Capabilities of EMOA to detect and preserve equivalent pareto subsets. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 36–50. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_7
17. Santander-Jimnez, S., Vega-Rodrguez, M.A.: Performance evaluation of dominance-based and indicator-based multiobjective approaches for phylogenetic inference. Inf. Sci. **330**, 293–314 (2016)
18. Shelokar, P., Quirin, A., Cordn, S.: A multiobjective evolutionary programming framework for graph-based data mining. Inf. Sci. **237**, 118–136 (2013)
19. Tanabe, R., Ishibuchi, H.: A niching indicator-based multi-modal many-objective optimizer. Swarm Evol. Comput. **49**, 134–146 (2019)
20. Yi, R., Luo, W., Bu, C., Lin, X.: A hybrid genetic algorithm for vehicle routing problems with dynamic requests, pp. 1–8 (2017)
21. Yue, C., Qu, B., Yu, K., Liang, J., Li, X.: A novel scalable test problem suite for multimodal multiobjective optimization. Swarm Evol. Comput. **48**, 62–71 (2019)
22. Yue, C., Qu, B., Liang, J.: A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems. IEEE Trans. Evol. Comput. **22**(5), 805–817 (2018)
23. Zhang, Q., Hui, L.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
24. Zhang, W., Li, G., Zhang, W.: A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization. Swarm Evol. Comput. **50**, 100569 (2019)
25. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

A Bacterial Foraging Framework for Agent Based Modeling

Mijat Kustudic and Niu Ben(✉)

College of Management, Shenzhen University, Shenzhen 518060, China
`Mijat.k.ntc@gmail.com`, `drniuben@gmail.com`

Abstract. Swarm optimization algorithms and agent based modeling (ABM) are two closely related research areas, parts of the multi agent system field, but they are traditionally not combined. Swarm optimization, in this case the bacterial foraging optimization (BFO), searches for an optimal solution while the ABM searches for a conclusion which resembles the real world, and it can be far from optimal. To bridge the gap, the overall goal this paper is to propose a new paradigm in the form of an architecture and operation procedures, thus creating a BFO-ABM hybrid. The other goal is to create a method which enables 3D visualization of the BFO algorithm. Firstly, an environment is created together with bacteria which physically perform all operators of the BFO. Secondly, a way of seamlessly embedding the bacteria from the BFO into the ABM environment is described. The bacteria are then manipulated and motivated with food and toxicity to act in a certain agent-like way. Simulation results prove that the agents can be effectively used as an ABM tool to present agents of all sizes and behaviors resembling numerous things, from companies, vehicles to people.

Keywords: Swarm optimization · Agent based modeling · Evolutionary computation

1 Introduction

1.1 A Subsection Sample

Computational field of Multi agent systems (MAS) is an interesting field of research which encompasses numerous subfields all based on individual and intelligent agents. These agents are autonomous, decentralized and have only their local view. Swarm optimization algorithms and agent based modeling approaches are subfields of MAS but they are not traditionally combined. At first glance they seem to be based on different principles but actually their logic is quite similar. The main difference is the goal which is ought to be made: in optimization there needs to be an optimal solution and in ABM a conclusion that resembles the real world, the result can be far from optimal.

Optimization algorithms have been used to solve different real world optimization and engineering problems. They are usually based on behaviors of certain animals or natural phenomena employed to search for an optimal solution,

L. Pan et al. (Eds.): BIC-TA 2019, CCIS 1159, pp. 727–738, 2020.
https://doi.org/10.1007/978-981-15-3425-6_58

to name a few of them: Particle swarm optimization [1], Hydrologic Cycle Optimization [2], Differential evolution [3], Water Cycle Algorithm [4], Artificial Bee Colony [5], Genetic Algorithm [6], Ant Colony Optimization (ACO) [7], Bacterial Foraging Optimization Algorithm (BFOA) [8] and numerous others which have proven themselves in different areas. These algorithms are always being updated through consideration of new strategies; for example within the BFO change of the chemotactic step length [9], population change [10] and the algorithm being adaptive [11]. Even though these algorithms are based on the real world, only a few tackle the changing habitat where the microorganisms are living [12]. This dynamic relationship makes them more similar to ABMs. They are defined as dynamic optimization tools which are an important focus point in research [13–15]. They consider that surroundings affects organisms but, in turn, they also affect the surroundings making them an important research point [16].

On the other hand, agent based modeling (ABM) is based on analyzing behaviors, emergence and adaption of complex systems with the notion that these systems are built from the bottom up. Their beginnings can be traced to cellular automata [17] and based on those simple principles numerous ABM usages have spawned, ranging from sociology [18], economics [19] and political science [20]. Even though both approaches are a part of MAS, both have intelligent agents and their results are based on their interactive behavior, combining or hybridizing them has not been done.

Bacteria within the BFO move around the environment while searching for food, consume it and move on to other areas. While this phenomena is interesting from the optimization standpoint, ABM is focused on the bacterial behavior and movement patterns. The primary goal of this paper is to create an architecture, mathematical representation and operation procedures in order to bridge the gap between optimization algorithms, namely the BFO, and agent based modeling, we will refer to it as BFO-ABM. Secondary goal is to create a method to enable 3D visualization of the BFO algorithm. To construct it, certain changes to the original are proposed: Creation of a 3D environment where the bacterial agents move and forage. Harnessing the environment (crucial element of the ABM) which the optimizing bacteria can inhabit (crucial element of the BFO). Creation of a methodology which can help in visualizing BFO operators and actions previously described only in writing. Employing the BFO's bacteria, through the ABM prism, as agents representing anything from companies to people.

To show performance and test the modeling capabilities of the BFO-ABM hybrid, we conducted two experiments on an explicit objective function which presents the environment. The first experiment is based on observing the bacterial agents' movement and behavior, disregarding the decline in nutrients and toxicity levels. Second experiment is based on observing bacterial agents' behavior until all the nutrients are completely gone while avoiding the toxic areas. In these two scenarios bacterial health is observed together with the speed of convergence and their effect on the objective function (the environment). This paper is organized as follows: Sect. 2 presents the formation of the environment around which the bacteria move and interact with. Section 3 describes the

BFO-ABM hybrid along with different bacterial actions. Section 4 presents results of two experimental studies together with discussions and potential uses of further modeling, followed by the conclusion in Sect. 5.

2 Environment

Optimization algorithms are usually not visualized, but if we wish to understand their movement we should transfer them to 2 or 3 dimensions. This is environment creation is the crucial step of the BFO-ABM hybrid since it enables the first element to be implemented into the second one. The algorithm envisions a space for activity in the form of a classical grid topology, called the Moore neighborhood, Fig. 1. Bacterial agents are grounded on the grid and are able to follow its topological change in 3D.

The algorithm is consisted out of two basic types of elements: habitat grid and the bacteria. They exist together and effect one another - they are in sync at every moment. As the environment possesses certain positive elements (food, nutrients) and negative ones (toxins and/or no food) the organisms are forced to adapt to it. The change comes from them eating the available food, where it exists and leaving it foodless and uninteresting. As the current is left empty they search for the next best place to forage, which contains the highest quantity of available food.

Grid. The grid covers a 3D space of X × Y × Z through which movement is possible but only across the X × Y grid surface, while their action and implications take effect and are noted on the Z axis. This can be represented as:

$$Grid_t = \{\beta, X, Y, Z \tag{1}$$

Where the $Grid_t$ depicts the habitat over which the agents move at a certain moment t while β is the amount of food which is distributed around it. X And Y are dimensions of the grid which are previously defined and are stationary while Z is a dynamic grid term, which is being changed by the foraging process.

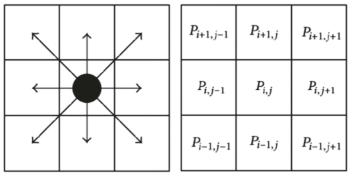

Fig. 1. Moore neighborhood movement possibilities [21]

Bacteria. Each bacteria can be defined as a set of its current characteristics. This means that a bacterium B at a certain time point t is defined and identified by its current position on the grid in the form θ_{xyz} coordinates but also the current iteration step.

$$B = \{\theta_{xyz}, iter \tag{2}$$

All food is randomly distributed across the grid with coordinates x and y and is consumed over t time instances. In both tested versions of the algorithm the food is not created over the course of iterations.

3 Employing the BFO as an ABM

Formation of the Habitat. When constructing the habitat multiple factors need to be taken into account. Analyzing these factors is key for understanding it and extracting some interesting waypoints from it. Surroundings can be presented as a combination of terrain elements, toxic areas, but also areas which are filled with food. We can present the movement space as the Z axis together with its parts:

$$Z = Z_{terrain} + Z_{toxicity} + Z_{food} \tag{3}$$

The bacteria need to consider all of these factors and navigate through them. They will choose to visit and interact with areas which possess the easiest way and the most food – this implies that the minimum of the function is there. To form the Z axis all of the layers are combined into one by addition of their values.

This means that there are certain tradeoffs which need to be considered. Such as: if there is a lot of food at a local area but the terrain is inhospitable, there will be no interest in going there; if there is food but the area is mildly dangerous the overall success needs to be compared with other areas to see if pursuing that area is the best option; and finally if the area is easy to get to and there is a lot of food it will be very interesting to visit. Mathematically we can present it as different points $(A^{Z1}_{x,y}, A^{Z2}_{x,y}, A^{Z3}_{x,y})$ which have the same coordinates but are on different axes, their impact on the Z axis will be calculated as follows:

$$A^{Z1}_{x,y}, A^{Z2}_{x,y}, A^{Z3}_{x,y} = \begin{cases} if sum > 0, interest\ is\ low \\ if sum = 0, interest\ is\ neutral \\ if sum < 0, interest\ is\ high \end{cases} \tag{4}$$

After having this calculation for all points on all axes the summary of them in the form of Z axis is created (Fig. 2). This is also done after each iteration step since the consumption of nutrients reduces their availability in certain areas after each step.

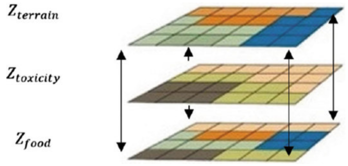

Fig. 2. Z axis formation

Initialization. Bacteria move through and across space which can be created implicitly, using a formula, or explicitly by directly creating it in Excel, for example. Major advantage of the explicit finite method is its relative simplicity and computational ease, on the other hand the implicit environment creates better solutions. In the case of this experiment an explicit one with 20×20 fields is used. Amount of food β is distributed across the environment grid thus forming the Z axis minima. Initial positions of agents at $\mathrm{t} = 0$ are randomly generated.

Operators. Bacteria move around their habitat by using their flagella (a whip-like part of their body) with the goal of escaping hazardous and foodless areas and finding ones with food. This process is called chemotaxis but the BFO-ABM considers a Migration term, which will be described further in the text. If a bacterium does not "consume" enough food or is not healthy enough it will die. To keep the population constant the organisms will *reproduce* by dividing into two identical copies which are positioned at the same location. This keeps the population diversified and eliminates some bacterium which are trapped in local minimum. If greater numbers of them are concentrated around a certain food source they will consume it food faster. *Dispersion* is employed to make the population more dynamic and to stimulate exploration and foraging in different areas.

Migration. Migration is defined as moving across the physical and dynamic space whose action requirements are presented by pseudocode in the Table 1. First task is to find the closest minimum to each bacterium, the minimum is defined through its X_{min} and Y_{min} coordinates. After this, the comparison between the current coordinates of the bacterium (X_{Bac},Y_{Bac}) starts. If the coordinates of the food and the bacterium match, then the bacterium is at the right place and the feeding can begin. Important thing to note is that in this algorithm *chemotaxis* operator helps agents move around a local area, migration considers covering of greater - global distances. To conserve energy the bacteria will choose locations where food is plentiful. This implies a greedy selection and helps with focusing on potential global minima.

Agent Waypoints. In the presented case, visualization of nutrient abundance is presented using lower values of the Z axis while the higher values do not possess any food, maybe because it has already been consumed, thus that area

Table 1. Pseudo code for the migration function

```
For I = 1:S
  If $X_{Bac}$ is different than $X_{min}$
      Move it towards it by 1 step
  elseif $Y_{Bac}$ is different than $Y_{min}$
      Move it towards it by 1 step
  End
End
```

is not interesting. That is how the BFO-ABM harnesses the bacterial interest and uses it as an agent modeler. The agents consider these low locations as waypoints or areas of interest to move about. The pseudo code is based on the current location (X_{Bac},Y_{Bac}) and the memorization of the *History* - where the agents have already been. Also important is the $History^{end}$ which memorizes the whole historical path of them but without the last position. This means that the bacterium influences the environment only if it has spent some time at the location, if it has not (meaning that the coordinates of the historical paths and the current position do not match) the bacterium is just passing by and has visited that place for the first time. This comparison and exclusion is made because we want to find out what are the areas of interest and at the same time exclude the path, because we presume it does not possess food. If the coordinates match then a *speed of consumption* value is added to the Z axis point reducing the interest which the bacteria have for that point, in other words the food located at that point is reduced by that amount by a single bacterium foraging activity.

Stopping Criteria. The algorithm stops because of two reasons. First one being the iteration counter reaches its maximum and all the operators have been used. Second reason is that all food in the environment has been consumed (Table 2).

Table 2. Pseudo code for the consumption function

```
For I = 1:S
    $History_X$(I)=$X_{Bac}$ && $History_Y$(I)=$Y_{Bac}$
    $History_X^{end}$=length($History_X$)-1 && $History_Y^{end}$=length($History_Y$)-1
      If $History_X$ - $History_X^{end}$ == 0 && $History_Y$ - $History_Y^{end}$ == 0
        Z($X_{Bac}$,$Y_{Bac}$) == Z($X_{Bac}$,$Y_{Bac}$) +1
      End
End
```

4 Experimental Studies

To demonstrate the algorithm two scenarios are taken into account. As a control in the experiment the grid considered will be the same size and characteristics, both will have the same number of minima which are corresponding to the amounts of nutrients. It is created using an explicit method based on Excel tables, each corresponding to a specific value of terrain characteristics, toxicity location and food distribution; they are summarized into a single value of the Z axis which size is 20×20 fields. Figure 3 shows the initial shape of the environment grid. The minima are marked as lower values than 7 while values equal to 7 present foodless area which are of no interest. Bacteria will search for food located in the minima. Corresponding values can be found in the Table 3. After the initialization the agents start their foraging activity and their actions take effect.

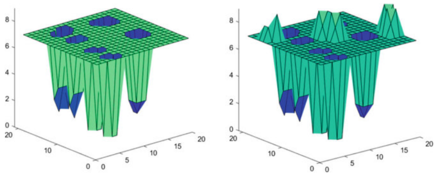

Fig. 3. Initial values of the Z axis for the two experiment scenarios.

4.1 Simulation I

Population for the first experiment is set to 5. Speed of nutrient consumption is set to 0.001. Stopping criterion of the algorithm, relies on the number of iterations and is set to 100. Figure 4 shows a moment at the 30^{th} iteration; bacteria are converging to the absolute minimum which is closest to them. From the moment of entering the minimum they begin foraging for food. This influence and interaction is also visible in the table through the increase of mean, standard deviation and minimum values. Since there are no toxic areas around the environment the bacteria can take the shortest (optimal) route.

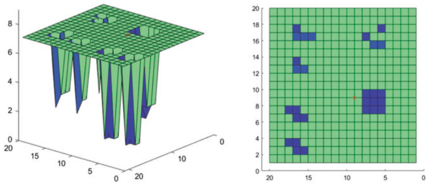

Fig. 4. Z axis situation at iteration 30

On the Fig. 5 we can see gradual warping of the terrain due to the increased foraging. The first minimum has been foraged enough (it is no longer a minimum because of terrain increase) so there is no more food there and other sources are being searched for. Compared to Fig. 4 we can see food sources are getting exhausted and bacteria need to move to other new ones. Note that the all bacteria are positioned on the same coordinates making them a more effective swarm.

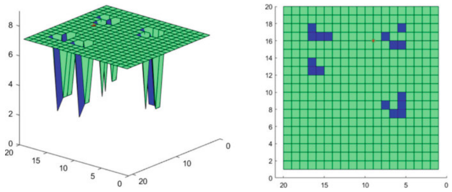

Fig. 5. Z axis situation at iteration 60

Figure 6 continues to show the causal relationship between the bacteria. We see the increasing effects of the process through space warping. Also visible is the gradual change of the minima's color, from dark blue to lighter shades. This is because the lack of food is presented as greater numbers, compared to the global minimum which are darker colors.

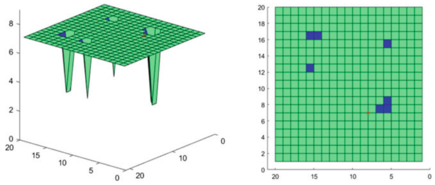

Fig. 6. Z axis situation at iteration 90 (Color figure online)

Final result of the interplay process can be seen on the Fig. 7. We see a very different situation than the initial one (left side of Fig. 3) all minima have been affected because their values have been changed – increased numerically due to the consumption of food which was located in them.

4.2 Simulation II

The second simulation features an increased number of bacteria which is set to 10 together with the incorporation of toxic areas. Initial Z axis shape and values can be seen on the right part of the Fig. 3. Since lower levels of the axis present positive elements, food, negative elements are presented as higher values; bacteria tend to evade them while foraging. The simulation shows how more subjects act faster thus having a greater impact and achieving exploration and exploitation of minima in a shorter time period, regarding the number of iterations. This is due to more competition and the same food level making

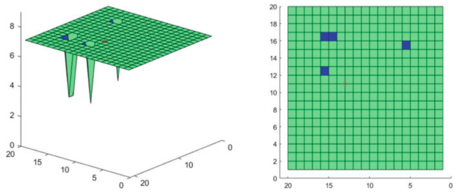

Fig. 7. Z axis situation at the final iteration

dynamics and behaviors different. Toxic levels need to be avoided which changes the path of the bacteria, making it a bit longer.

In the second simulation the stopping criteria was not the iteration limit but the food depletion in the habitat. That is why there are no numbers lower than 7, which exist in the first experiment. As in the final result of the first simulation we can see a great difference and effects which were left by the process (Fig. 8). Also noticeable is the rate and convergence speed which is much greater than it the first experiment; it can be seen on the Z axis. The experiment shows that more agents act quicker and also that their efficiency is increased by swarming and taking effect on a single location.

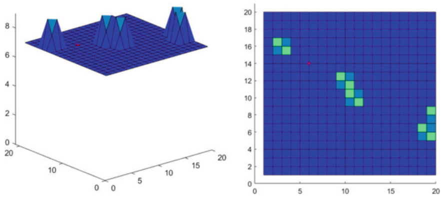

Fig. 8. Z axis situation after all food is consumed

4.3 Discussion

Figure 9 shows the explicit objective function which is being optimized by the interacting bacteria. From it we can observe consumption to movement ratios. Namely, if the line is horizontal it shows the bacteria are moving to the next minimum and during that movement they are not consuming. If the line is steeper than the horizontal one it shows food consumption happening, steeper line means more consumption per unit of time. Left side of the figure shows 5 and the right one 10 bacteria interacting. We can conclude that greater impact is being achieved with 10 bacteria since the conversion rate is faster, even though sometimes they take a longer route to avoid the toxic areas. The line would be also much steeper if food sources were close together or in other words, if the objective function (the environment) was of different shape and value.

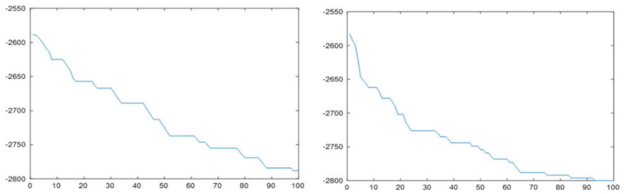

Fig. 9. Objective function from the two simulation scenarios

Table 3 summarizes all values of the Z axis from different experiment steps. In the first experiment the mean and standard deviation are being increased through the foraging process. Mean value has been increased from 6.475 to 6.959. During the course of the experiment standard deviation is being steadily lowered due to the foraging process, 0 will signal that there is no more food available. Final value from the first experiment presents the global minimum which has been increased, from 1 to 3. This is because all minima have been affected by the foraging.

Table 3. Numerical values from the simulation

Result	First value	Fig. 3	Fig. 4	Fig. 5	Final value I	Final value II
Mean	6.475	6.642	6.817	6.922	6.959	7.075
Std	1.008	0.914	0.710	0.519	0.374	0.271
Max	7	7	7	7	7	9
Min	1	1	2	2	3	7

Second simulation has a different stopping criteria than the first one food depletion in the habitat and also toxicity levels are visible. We see that the mean has increased to 7.075 and standard deviation has been lowered to 0.271 signaling deficiency of food but also that the toxicity levels are still active. The overall maximum is 9 which also points to existing dangerous areas for the bacteria. On Fig. 10 we can see how different bacteria have different fitness levels and how it changes over time. It is interesting to note in both experiments that there are several leaders forming, they come to the food source first and forage the most, thus the others do not have the opportunity to eat.

4.4 Bacteria as Agents

It is important to grasp how useful can this BFO-ABM approach be. For example we can consider these agents as companies which need to position themselves on the market to have the best access to the customer base. The first company to set up its business at a market quadrant usually wins, but the classical question

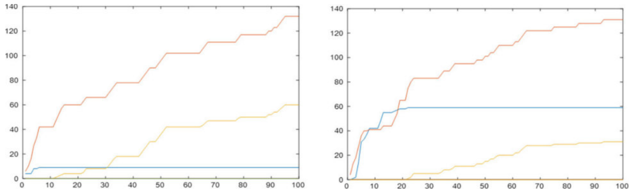

Fig. 10. Individual bacterial fitness from the two simulation scenarios

is whether it is better to be focused only on certain customers or be ready to move about and be more flexible; similarly done in [22].

Other consideration for the bacterial agents can be simulating the movement of people around a certain space, they cannot move through walls and barriers (presented as high toxicity levels); similar simulation is done in [23]. To simulate a fire escape simulation we place the food outside of the room, bounded by "toxicity" serving as walls. Then we can check how bacterial agents act depending on the danger and the environment, as seen in [21]. Apart from these applications the BFO-ABM approach can consider and simulate numerous other agents and scenarios no matter the size or intention.

5 Conclusion

Use of swarm optimization algorithms and agent based modeling is usually kept separate. This is understandable since behind them are different paradigms of operation. Apart from these differences, there are numerous similarities, because they are both from the multi agent systems field. Based on those similarities, this paper has presented a new architecture and framework which is used for modeling bacterial behavior within a 3D environment. It also showed how to seamlessly incorporate bacteria to act according to the ABM rules which makes this approach a hybrid one. The whole environment is presented as an objective function which needs to be optimized by bacterial consumption of food while at the same time avoiding the toxic areas.

Different simulation scenarios showed how bacteria can be manipulated and motivated with food and toxicity to act in a certain agent-like way. Even though the presented environment and scenarios are relatively simple, it shows that they can be used as an effective agent modeling tool. Our future work will be based on using these bacteria in simulating behaviors of companies, vehicles or people and applying them to real world situations.

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)

2. Xiaohui, Y., Niu, B.: Hydrologic Cycle Optimization Part I: Background and Theory, Advances in Swarm Intelligence, pp. 341–349 (2018)
3. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
4. Eskandar, H., Sadollah, A., Bahreininejad, A., et al.: Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput. Struct. **110–111**(10), 151–166 (2012)
5. Dervis, K., Bahriye, A.: A comparative study of Artificial Bee Colony algorithm. Appl. Math. Comput. **214**(1), 108–132 (2009)
6. Holland, J.: Genetic algorithms. Sci. Am. **267**(1), 66–72 (1992)
7. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2006)
8. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. **22**, 52–67 (2002)
9. Niu, B., Fan, Y., Wang, H., Li, L., Wang, X.: Novel bacterial foraging optimization with time-varying chemotaxis step. Int. J. Artif. Intell. **7**(11), 257–273 (2011)
10. Fernandes, C., Ramos, V., Rosa, A.C.: Varying the population size of artificial foraging swarms on time varying landscapes. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 311–316. Springer, Heidelberg (2005). https://doi.org/10.1007/11550822_49
11. Chen, H.N., Zhu, Y.L., Hu, K.Y.: Adaptive bacterial foraging algorithm. Abstract Appl. Anal. **2011** (2011). Article ID 108269
12. Tang, W.J., Wu, Q.H., Saunders, J.R.: Bacterial foraging algorithm for dynamic environments. In: IEEE Congress on Evolutionary Computation (2006)
13. Morrison, R.W., De Jong, K.A.: A test problem generator for non - stationary environments. In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation, pp. 2047–2053. IEEE Press (1999)
14. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Massachusetts (2002)
15. Tang, W.J., Wu, Q.H., Saunders, J.R.: A novel model for bacterial foraging in varying environments. In: Gavrilova, M., et al. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 556–565. Springer, Heidelberg (2006). https://doi.org/10.1007/11751540_59
16. Daas, M.S., Batouche, M.: Multi-bacterial foraging optimization for dynamic environments. In: International Conference of Soft Computing and Pattern Recognition (2014)
17. Conway, J.: The game of life. Sci. Am. **223**(4), 4 (1970)
18. Bruch, E., Atwell, J.: Agent-based models in empirical social research. Sociol. Meth. Res. **44**(2), 186–221 (2015)
19. Tesfatsion, L., Judd, K.L. (eds.): Handbook of Computational Economics: Agent-Based Computational Economics, vol. 2. Elsevier, Amsterdam (2006)
20. Cederman, L.E.: Computational models of social forms: advancing generative process theory. Am. J. Sociol. **110**(4), 864–893 (2005)
21. Wang, C., Wang, J.: A modified floor field model combined with risk field for pedestrian simulation. Math. Probl. Eng. **1**(10) (2016)
22. Van, L.E., Lijesen, M.: Agents playing Hotelling's game: an agent-based approach to a game theoretic model. Ann. Reg. Sci. **57**(2–3), 393–411 (2015)
23. Tecchia, F., Loscos, C., Conroy-Dalton, R., Chrysanthou, Y.L.: Agent Behavior Simulator (ABS): a platform for urban behavior development. In: First International Game Technology Conference and Idea Expo (GTEC 2001) (2001)