Bacteria Swarm Foraging Optimization for Dynamical Resource Allocation in a Multizone Temperature Experimentation Platform

Mario A. Muñoz, Jesús A. López, and Eduardo Caicedo

Grupo de Investigación Percepción y Sistemas Inteligentes
Universidad del Valle, Cali, Colombia, South America
{andremun,jesuslop,ecaicedo}@univalle.edu.co

Abstract. In this work, an algorithm based on the Bacteria Swarm Foraging Optimization was used for the dynamical resource allocation in a multiple input/output experimentation platform. This platform, which mimics a temperature grid plant, is composed of multiple sensors and actuators organized in zones. The use of the bacteria based algorithm in this application allows the search the best actuators in each sample time. This allowed us to obtain a uniform temperature over the platform. Good behavior of the implemented algorithm in the experimentation platform was observed.

1 Introduction

The natural selection tends to eliminate animals with poor "foraging strategies" and favor the propagation of the animals with successful ones, since they have more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or redesigned. This evolutionary principle has led the scientists in the field of "foraging theory" to model the foraging activity as an optimization process. The balance between the energy intake and the time spent in its search has been "engineered" into what is called an "optimal foraging policy". Optimization models are also valid for "social foraging" where groups of animals cooperate to forage [1].

The E. Coli bacterium is one of this individuals that form groups for social foraging. This bacterium is probably the best understood microorganism, whose entire genome has been sequenced. Also, the E. Coli is capable of reproduce by division and occasionally transfer gene sequences from one to another. The bacterium moves through its medium by two states, the tumble and the run, that allows it to search for food and avoid noxious substances. The knowledge of the characteristics and behavior, known as chemotaxis, of the E. Coli and its interactions between each other and the environment, has allowed the development of the *Bacteria Swarm Foraging Optimization (BFSO)* algorithm [2].

In this paper, an algorithm based on the Bacteria Swarm Foraging Optimization is used to dynamically allocate the time of ignition of an actuator in a Multizone Temperature Experimentation Platform (MuTEP) and to achieve a uniform temperature over a particular area. The MuTEP, presented in [3], is a multiple input / output plant that emulates the workings of a system designated to control the temperature over a surface.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 427–435, 2007.
springerlink.com

The paper is organized as follows. First we show the main concepts of the BSFO algorithm. Next we describe the experimentation platform used to proof our algorithm. Later we continue with the explanation of the bacteria algorithm for dynamical task allocation in the experimentation platform. Finally we show some results when the algorithm was applied to the platform.

2 The Bacteria Swarm Foraging Optimization

Suppose that we need to find the minimum of $J(\theta), \theta \in R^p$, when we do not have a deterministic description of $J(\theta)$ or its gradient. This problem becomes a non gradient optimization problem, where the ideas from bacteria foraging can be used. Suppose that θ is the position of the bacteria and $J(\theta)$ represent the environment conditions, with $J(\theta) < 0$, $J(\theta) = 0$ and $J(\theta) > 0$ represents that the bacteria location is a nutrient rich, neutral or noxious environment, respectively. Basically, the chemotaxis is a foraging behavior where bacteria tries to climb up the nutrient concentration, avoid noxious substances and search for ways out of neutral media by a random walk.

A chemotactic step *j* is defined as a tumble followed by a tumble or a tumble followed by a run, a reproductive step *k* is defined as the selection of the fittest in the population and its splitting, and a elimination–dispersal event *l* as the selection of random individuals and its relocation in a new random position. Then,

$$P(j,k,l) = \theta_i(j,k,l) \mid i = 1,2,\ldots S \tag{1}$$

are the positions of each member of the *S* bacteria population at *j-th* chemotactic step, *k-th* reproductive step and *l-th* elimination and dispersion event. Then $J(i,j,k,l)$ is the location cost of the *i-th* bacteria $\theta_i(j,k,l) \in R^p$, and N_c as the bacteria's life time in chemotactic steps. To represent a tumble, a length unit in a random direction y (j) is generated:

$$\theta_i(j+1,k,l) = \theta_i(j,k,l) + C(i) \cdot \psi(j) \tag{2}$$

where $C(i)$ is the size of the step at the direction $\psi(j)$. If in $\theta_i(j+1,k,l)$, the value of $J(i,j+1,k,l)$ is less than in $\theta_i(j,k,l)$, then a new step is taken until a maximum of N_s, making this cycle a chemotactic step.

After N_c chemotactic steps, a reproduction step is taken. For the reproduction, the healthiest bacteria are split and the others are eliminated, maintaining a constant population. After N_{re} reproduction steps, a dispersion and elimination event is made, where each bacteria is subject to relocation with a probability p_{ed}. After N_{ed} dispersion and elimination, the algorithm ends. The population size *S* is restricted to an even number, so the population can be easily kept constant.

3 The Multizone Temperature Experimentation Platform

The Bacteria algorithm was tested in a Multizone Temperature Experimentation Platform (MuTEP) [3], which was composed of two parts, a process stage, and a data acquisition stage.

The process stage, shown in the figure 1, is an emulation of a planar temperature grid. This is a system that exhibits effects that are difficult to model, especially strong interactions between zones. Therefore, it requires the use of particular control strategies. This type of system is mainly used in the semiconductor industry for the elaboration of crystals and the generation of photo resistive layers, processes that require maintaining a constant surface temperature [4, 5]. The shape of the grid was selected because of its symmetry and the difficulty in raising the edges temperature.

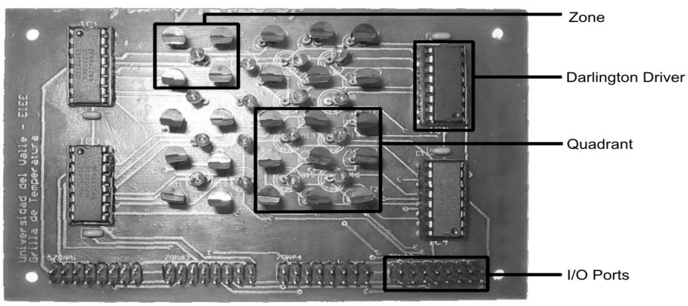

Fig. 1. Process stage of the Multizone Temperature Experimentation Platform (MuTEP)

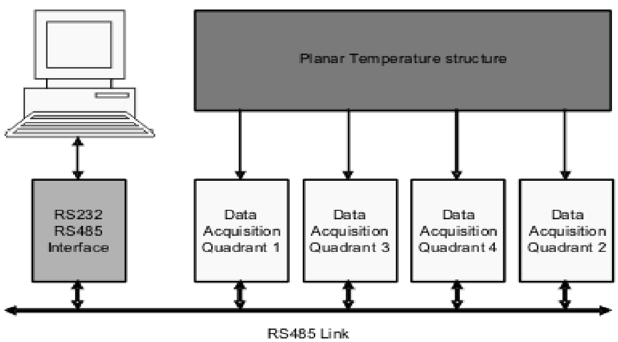

Fig. 2. System Architecture of the Multizone Temperature Experimentation Platform (MuTEP) for management form a single computer

The data acquisition system used was composed of four modules based in a low cost micro controller. Each module controlled a quadrant of the process and enabled communication with a master computer, which contained the management and control algorithm. The basic structure of the system is shown in Fig. 2.

4 Bacteria Algorithm for Dynamical Resource Allocation

We design an algorithm using the BSFO algorithm as base to develop an strategy similar to the shown in [6, 7]. This strategy finds the zone with the lowest temperature and assigns the resource to that zone. To assign the resource implies to turn on the bulb of the zone and, as consequence, the temperature of the zone is increased. This is a centralized approach were a single computational agent takes a destination based in global knowledge.

The searches done by the BSFO are in continuous spaces. In this application, the search space corresponds to the surface defined by the platform, and the bacteria will move between the actuators depending of its food concentration.

The objective function used in our proofs was to obtain a maximum uniform temperature over the process surface using a limited amount of actuators (four maximum in our proofs) over a period of time. The amount of actuators depends of the numbers of bacteria (agents) used in the algorithm. We did proofs using two and four bacteria, because of the limitation of the population size to even numbers. This produce that in each sample time are turn on two or four bulbs.

In our algorithm the bacteria are placed at random at the beginning of the experiment. Then it will have a limited number of attempts per chemotaxis step to find a lower cost position, and will be able to move one position in a sample time. The surface cost is directly related to the surface temperature, making the food sources the location with lower temperature. If a bacteria is unable to find a better position after its attempts, can be eliminated or left in place with a probability of $p_{ed} = 0.1$ (low to avoid random search), to avoid the lockup of the bacteria in one position. Only the nutrients action was used, and there is a restriction in the location of the agents, so they do not use the same position. With these considerations, the developed algorithm can be presented as follows [8]:

1. The bacteria are located at random inside the search space in the first generation.
2. The current cost surface is calculated by the actual value obtained from the sensors in the platform.
3. A random tumble is generated and the next position is find. If the next position exceeds the edges, it will be bounded to the nearest position.
4. A run cost is calculated by the difference between the new position in the current surface and the current position in the previous surface.
5. If the difference value is negative, the bacteria are displaced to the new position and its health is updated. If the difference value is positive, steps 3 to 5 are repeated until the search attempts are exhausted.
6. If no better position than the current is found, a dispersion event is performed. If $rand < p_{ed}$, where *rand* is a random number using uniform distribution, the bacteria is relocated in a random place in the surface and its health reset, otherwise, the bacteria stays in its position and its health stays the same.
7. Steps 3 to 6 are repeated for all the bacteria.
8. If the number of samples per reproductive step are achieved, a reproduction step is performed where the healthiest bacteria are split in two and the offspring is placed in the lowest temperature position beside the parent.

5 Results

For the execution of our algorithm different sizes of the population were used. First, taking in account that each bacteria used turns on an actuator, the maximum size of the population is four bacteria. The experiments were carried on using populations of two and four bacteria. The results for two bacteria are shown in the figure 3 and for four bacteria in the figure 4, during a 1800 samples experiment, with $N_{re} = 10$. The figures show that with four agents a higher temperature was achieved than with two agents. The actuators use pattern show that the corner and edge positions where preferred over the central ones. Nevertheless, with the use of the random dispersion, the central positions where used. The higher actuator use in the central part of the structure indicates that there are variations in the way the structure behaves.

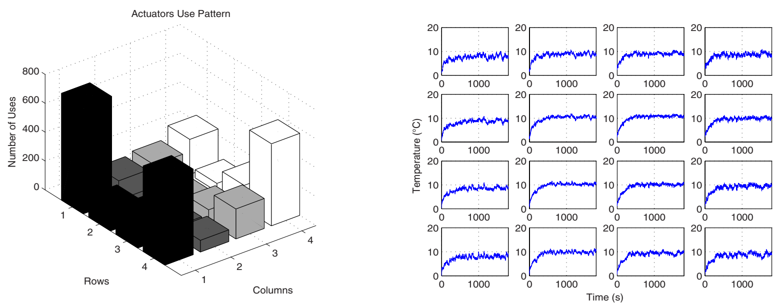

Fig. 3. Results of the experiment obtained with two bacterias: (a-left) shows the number of times an actuator is used in the experiment (b-right) shows the zones' temperatures over the experiment

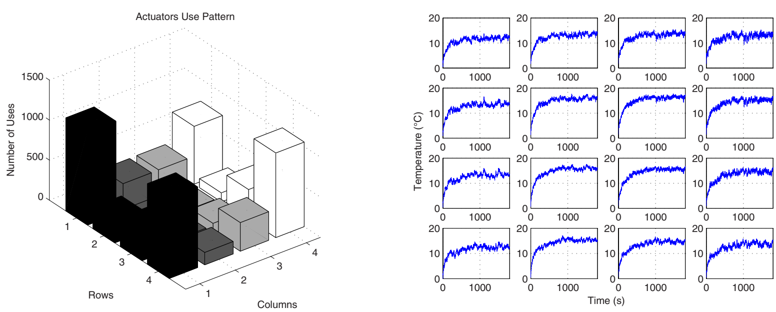

Fig. 4. Results of the experiment obtained with four bacterias: (a-left) shows the number of times an actuator is used in the experiment (b-right) shows the zones' temperatures over the experiment

For the results evaluation, a series of parameters where used to check if the obtained temperature surface gets the control goal [8], during a 5 experiment series. These are:

- Maximum average temperature ΔT: Corresponds to the maximum variation of the average surface temperature. With $t_p(t)$ the average temperature at 0 and $t_p(t_{fin})$ the average temperature at the end of the experiment ΔT is defined by the equation 3 The results are shown in the figure 5.

$$\Delta T = t_p(t_{fin}) - t_p(t) \tag{3}$$

- Settling time t_{est} : Corresponds to the settling time taken to achieve the maximum average temperature. The results are shown in the figure 5.
- Settling temperature spreading σ: Correspond to the error of the surface to the average temperature. The results are shown in the figure 6.
- Spreading percentage %σ: The comparision of the sperading and the achived average tempeature is allowed. The results are shown in the figure 6.
- Control action average *CA*: Corresponds to the average number of actuators used in a sample. The identification of the control effort used to raise the temperature is allowed and calculated by the equation 4, where $u_j(i)$ corresponds to the control action value of the *j–th* actuator at the time *i*. The results are shown in the figure 7.

$$CA = \frac{1}{t_{fin}} \sum_{i=0}^{t_{fin}} \sum_{j=1}^{L} u_j(i) \tag{4}$$

The results showed that, while the temperature spreading percentage is lower with the bacteria algorithm than with a simple random and sequential selection, and even with an ant algorithm presented in [9], is not better than the simple strategies shown in [6, 7]. The achieved average temperature stays around the values for the number of

Table 1. Evaluation Parameters of the Experiment

Method	ΔT	t_{est}	σ	%σ	CA
Sequential	5,1808	699	1,2579	24,28	1,0000
Random	4,8320	388	1,1621	24,05	0,9339
Simple	5,2275	609	0,3568	6,83	1,0000
Distributed	12,4800	482	0,5748	4,61	3,1189
1 Ant	4,4483	534	0.7532	16.93	1,0000
2 Ants	8,2744	831	1,2089	14,61	1,9872
4 Ants	13,5954	448	1,6402	12,06	3,7740
2 Bacteria	8,3389	482	0,8289	9,94	1,9456
4 Bacteria	13,5789	797	1,3190	9,71	3,6989

agents used. Because of the elimination–dispersal of the agents, some may fall in a filled position, making the *CA* decrease. Finally, the settling time varies depending in the environmental conditions, making this value not very useful in the comparison.

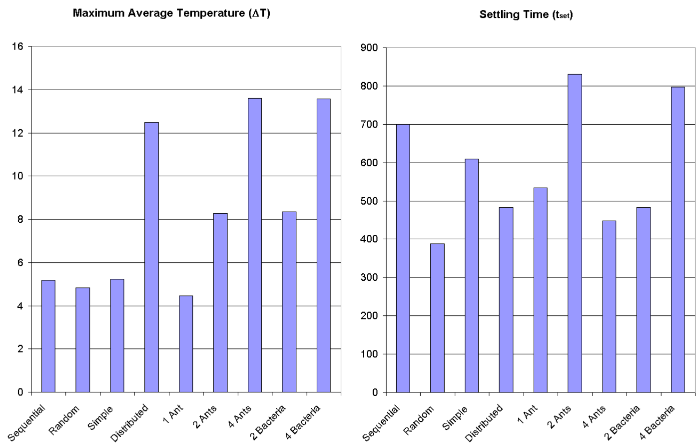

Fig. 5. Maximum average temperature ΔT (left) and Settling time t_{est} (right) with the bacteria algorithm and the algorithms described in [3, 9]. The graphics shows that an increase of the population achieves a higher temperature, while the settling time stays around a determinated range.

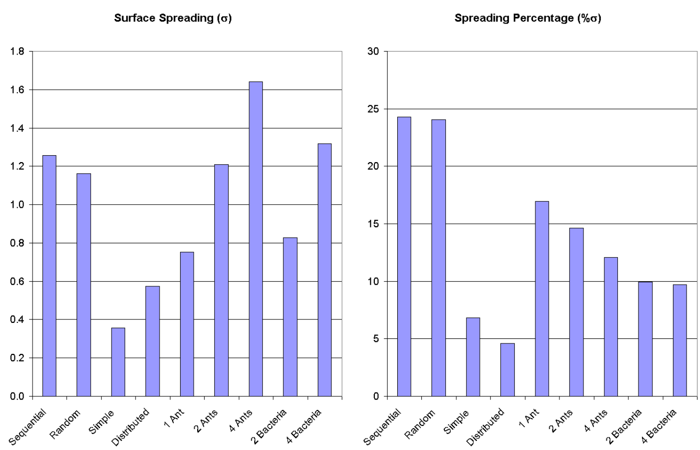

Fig. 6. Settling temperature spreading σ (left) and Spreading percentage %σ (right) with the bacteria algorithm and the algorithms described in [3, 9]. The graphic shows that an increase of the bacteria population increases the temperature spreading.

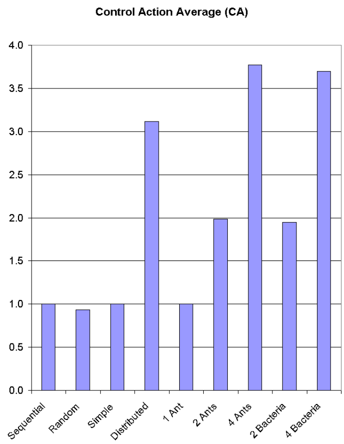

Fig. 7. Control action average *CA* with the bacteria algorithm and the algorithms described in [3, 9]

6 Conclusions

In this paper, the implementation and test of a bacteria based algorithm for dynamical resource allocation in the MuTEP platform was presented. This work represents the first step toward the implementation of a complex intelligent controller. The simulated bacteria presented an intelligent behavior that allowed the achievement of good results, although they are not optimal, they use the most adequate actuators.

The experiment showed that at a large population the temperature variation is increased. The analysis of the results showed that the bacteria algorithm allowed a shorter establishing time than the algorithms showed in [3].

Further work will include the construction of a model based controller for each zone or for the whole system, using a bacteria based algorithm. The development of this controller will include restrictions in the number of actuators used, magnitude of the control action, and continuous operation different from the on-off approximation used in this experiment.

Acknowledgments

We thank Professor Kevin Passino for inspiring this work and his graduate students Nicanor Quijano, Jorge Finke, and Alvaro Gil for their valuable discussions. Part of this work has been supported by Colciencias and Universidad del Valle through a graduate research fellowship awarded to Jesús A. López, and a graduate research scholarship awarded to Mario A. Muñoz, through grant No. 1106–11–17707.

References

[1] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3):52–67, 2002.
[2] K. M. Passino. Distributed optimization and control using only a germ of intelligence. 2000.
[3] M. A. Muñoz, J. A. López, and E. F. Caicedo. Implementation of a distributed control experimentation platform. In IEEE 2005 Conference on Industrial Electronics and Control Applications, 2005.
[4] C. D. Schaper, K. A. El-Awady, and A. E. Tay. Spatially programmable temperature control and measurement for chemically amplified photoresist processing. In K. Dang A. J. Toprac, editor, Procedures SPIE – Process, Equipment and Materials Control in Integrated Circuit Manufacturing V, volume 3882, pages 74–79, 1999.
[5] K. El-Awady, C. D. Schaper, and T. Kailath. Temperature cycling and control system for photosensitive materials processing. Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures, 21(4):1459–1465, 2003.
[6] N. Quijano. Experiments and technologies for decentralized temperature control. Master's thesis, The Ohio State University, 2002.
[7] N. Quijano, A. E. Gil, and K. Passino. Experiments for dynamic resource allocation, scheduling and control. IEEE Control Systems magazine, pages 63–79, 2005.
[8] M. A. Muñoz. Asignación dinámica de recursos con técnicas bio inspiradas para un sistema de control de temperatura mimo. Technical report, Universidad del Valle, 2005.
[9] M. A. Muñoz, J. A. López, and E. F. Caicedo. Ant colony optimization for dynamical resource allocation in a multizone temperature experimentation platform. In 2006 Electronics, Robotics and Automotive Mechanics Conference, volume 1, pages 137–142, 2006.