
Path Planning Optimization for Mobile Robots Based on Bacteria Colony Approach

Cezar Augusto Sierakowski and Leandro dos Santos Coelho

*Pontifícia Universidade Católica do Paraná, Grupo Produtrônica
Programa de Pós-Graduação em Engenharia de Produção e Sistemas
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR – Brazil
E-mail: cezars@globocom; leandro.coelho@pucpr.br*

Abstract. Foraging theory originated in attempts to address puzzling findings that arose in ethological studies of food seeking and prey selection among animals. The potential utilization of biomimicry of social foraging strategies to develop advanced controllers and cooperative control strategies for autonomous vehicles is an emergent research topic. The activity of foraging can be focused as an optimization process. In this paper, a bacterial foraging approach for path planning of mobile robots is presented. Two cases study of static environment with obstacles are presented and evaluated. Simulation results show the performance of the bacterial foraging in different environments in the planned trajectories.

1 Introduction

Swarm intelligence is an emerging research area with similar population and evolution characteristics to those of genetic algorithms. However, it differentiates in emphasizing the cooperative behavior among group members. Swarm intelligence is used to solve optimization and cooperative problems among intelligent agents, mainly in computer's networks, mobile robotics [11] and cooperative and/or decentralized control [2], [7], [10], [16]. Swarm intelligence, in nature, may be composed of three main principles: evaluation, comparing and imitation. Evaluation is the capacity to analyze what is positive or negative in nature, attractive or repulsive. Even the smaller life forms have these abilities, in the case of bacteria, they are able to notice if the environment in which they are located is noxious or not. Learning won't happen unless beings are capable of evaluate the attractive and repulsive characteristics of the environment. Comparison is the way living beings use other beings as a standard to evaluate themselves, results of these comparisons may become a motivation to learning and/or modification. Imitation is an effective form of learning. However, very few animals, in nature, are capable of imitating, in fact, only human beings and some species of birds are capable of such action [10]. These three basic principles may be combined, in a simplified version, in computer programs, opening possibilities

for them to adapt to complex problems. Animals, or groups of animals, when foraging, act looking for maximizing the amount of energy obtained per unit of time spent foraging, considering the biological and environmental limitations.

In the context of biologically inspired optimization methods, several models of bacterial chemotaxis algorithm based on pioneered work of Bremermann [4] have been proposed in literature for applications in biology [1],[13], genetics [5], communication networks [20] and robotics [6]. In this paper, the foraging theory is applied to bacteria, adopting the bacteria colonies nomenclature. The fact that bacteria are one of the simplest living beings existing in Earth and they use the forage theory to benefit the group motivated this study. Bacteria own a control system that allows the foraging control and avoidance of noxious substances. In this context, the cooperation activity in a bacteria colony may be used in an optimization procedure, based in the forage strategy, as proposed by Passino [15].

This paper's contribution is to present a bacteria colony study for planning mobile robots trajectory. In this context are evaluated simulated cases of an environment composed by several obstacles. The next sections of the paper are organized as follows. Section 2, fundamentals and equations for bacteria colonies algorithm are discussed. Section 3 presents analysis of mobile robots trajectory optimization problem and cases. In Section 4 is discussed the results obtained with the bacteria colonies application. Conclusions and future perspectives are shown in Section 5.

2 Bacteria Colony

Natural Selection tends to eliminate animals with poor foraging strategies and to favor gene propagation of those with good foraging strategies, once these have higher chances of succeeding in reproduction. These evolutionary principles have taken scientists to develop the foraging strategies, turning it appropriate to optimization models [15]. The presence of flagellum allows the bacteria to move, the movement is acquired through the flagellum rotation in the same direction, at a rotating speed of 100 to 200 rotations per second. Bacteria may move in two different forms: they might run (swim for a period of time), movement achieved by the flagellum rotation counterclockwise, or they can tumble, achieved by the flagellum rotation clockwise. Bacteria switch between these two modes of operation during its entire lifetime (rarely the flagellum stops rotating).

After a run period, a bacterium tumbles, the tumble interval is about $0,14 \pm 0,19s$, according to Passino [15]. After the tumble, the bacterium is pointed in a random direction. When the flagellum are rotated counterclockwise, the bacterium will move towards the direction it's turned, at an average speed of $10 - 20 \mu m / s$, meaning, about 10 times its length by second, for a mean interval of $0,86 \pm 1,18s$. The local environment where bacteria live might change, either gradually or suddenly. So bacteria can suffer a process of elimination, through the appearance of a noxious substance, or to disperse, through the action of another substance, generating the effects of elimination and dispersion.

A bacterium position, in a time instant, can be determined through equation (1), where the position in that instant is calculated in terms of the position in the previous instant and the step size $C(i)$ applied in a random direction $\phi(j)$, generated in the bacterium tumble,

$$\theta'(j+1, k, l) = \theta'(j, k, l) + C(i) * \phi(j, k, l) \quad (1)$$

To adapt such strategy to optimization problems, an equation to determinate the cost of each position is needed to enable the comparison between the position and the environment. The cost is determined by the equation,

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta'(j, k, l), P(j, k, l)) \quad (2)$$

Through equation (2) is noticed that the cost of a determined position $J(i, j, k, l)$ is also affected by the attractive and repulsive forces existing among the diverse bacteria of the population $J_{cc}(\theta'(j, k, l), P(j, k, l))$.

After a determined number of chemotactic steps (steps comprehending the movement and the cost determination of each bacterium position), a reproductive step occurs. In this reproductive step the bacterium are sorted decreasingly by their cumulative cost. The lower half of the list die, these are the bacteria that couldn't gather enough nutrients during the chemotactic steps, and the upper half divide themselves into two new bacteria, located in the same position.

Summarizing, the term *taxis* refers to the locomotory response of a cell to its environment. In a taxis, a cell responds such that it changes both direction and the duration of the next movement step. The tactic response requires some directional information from the environment that bacteria obtain by comparing an environmental property at two different time steps. If the tactic response is related to information about chemical concentrations (that may be either attractants or repellents), it is called chemotaxis [14].

In Figure 1 (shown at the end of the article) the algorithm is presented in pseudo code. As seen in the pseudo code, the bacteria colony algorithm is basically composed by an elimination and dispersal loop, inside this loop, there is another one, who is responsible for the bacteria reproduction. Inside this one, there is a third loop, responsible for generating the direction in which each bacterium will run, determining the period the bacterium will move and, as a consequence, determining it's position after the loop execution, and calculating the fitness of these positions. The reproductive loop is responsible for determining which of the bacteria must reproduce and which must be exterminated after the movements executed in loop 3, through a cost analysis of their positions along their movement. The first loop is responsible for eliminating some bacteria; it's ruled by an elimination probability, repositioning them into another random position of the search space.

```

DO{  $l = l + 1$ ;
DO{  $k = k + 1$ ;
DO{  $j = j + 1$ ;
FOR EACH bacterium  $i$  {
  Calculate  $J(j, k, l)$ ; Assume  $J(j, k, l) = J(j, k, l) + J_{cc}(\phi'(j, k, l), P(j, k, l))$ ;
  Save  $J_{last} = J(i, j, k, l)$ ;

  Generate a random vector  $\Delta(i) \in \mathfrak{R}^p$ , with real numbers, within  $[-1, 1]$ ;

  Move  $\theta'(j+1, k, l) = \theta'(j, k, l) + C(i) * \frac{\Delta(i)}{\sqrt{\Delta^\tau(i) * \Delta(i)}}$ ;

  Calculate  $J(i, j+1, k, l)$ ;

  Assume  $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{cc}(\theta'(j+1, k, l), P(j+1, k, l))$ ;
  Assume  $m = 0$ ;
  DO{
    Assume  $m = m + 1$ ;
    IF  $J(i, j+1, k, l) > J_{last}$  (optimization problem) THEN
      Assume  $J_{last} = J(i, j+1, k, l)$ ;

      Calculate  $\theta'(j+1, k, l) = \theta'(j+1, k, l) + C(i) * \frac{\Delta(i)}{\sqrt{\Delta^\tau(i) * \Delta(i)}}$ ;

      Calculate  $J(i, j+1, k, l)$ ;
      Assume
         $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{cc}(\theta'(j+1, k, l), P(j+1, k, l))$ ;
    ELSE Assume  $m = N_s$ ;
  } WHILE  $m < N_s$ ;
}
} WHILE  $j < N_c$ ;
FOR EACH bacterium  $i$  {
  Calculate  $J_{Health}^i = \sum_{j=1}^{N_s-1} J(i, j, k, l)$ ;

  Sort the bacteria, according to the valor of  $J_{Health}$ ;
  Kill the bacteria with the smaller value of  $J_{Health}$ ;

  Duplicate the bacteria with the higher values of  $J_{Health}$ ;
} WHILE  $k < N_{re}$ ;
FOR EACH bacterium  $i$  {
  Eliminate and disperse bacterium with a probability of  $p_{ed}$ ;
} WHILE  $l < N_{ed}$ .

```

Fig. 1. Pseudo code of the foraging theory applied to a bacteria colony.

3 Trajectory Planning of Mobile Robots

Literature is rich in boardings to solve mobile robots trajectory planning in presence of static and/or dynamic obstacles [18], [3], [12]. One of the most popular planning methods is the artificial potential fields [17]. However, this method gives only one trajectory solution that may not be the smaller trajectory in a static environment. The main difficulties in determining the optimum trajectory are due to the fact that analytical methods are extremely complex to be used in real time, and the searching enumerative methods are excessively affected by the size of the searching space.

Recently, the interest in using evolutionary algorithms has increased, genetic algorithms are used in mobile robots trajectory planning, generally when the search space is large [8], [19], [9]. The trajectory planning is the main aspect in the movement of a mobile robot. The problem of a mobile robot trajectory planning is typically formulated as follows: given a robot and the environment description, a trajectory is planned between two specific locations which is free of collisions and is satisfactory in a certain performance criteria [19].

Seeing the trajectory planning as an optimization problem is the boarding adopted in this article. In this case, a sequence of configurations that moves the robot from an initial position (origin) to a final position (target) is designed.

A trajectory optimizer must locate a series of configurations that avoid collisions among the robot(s) and the obstacle(s) existing in the environment. The optimizer must also try to minimize the trajectory length found, in order to be efficient. The search space is the group of all possible configurations.

In the present study, it's considered a bidimensional mobile robot trajectory planning problem, in which the position of the mobile robot R is represented by Cartesian coordinates (x,y) in the xy plan. The initial and destination points of the robot are (x_0, y_0) and (x_{np}, y_{np}) , where n_p is a project parameter. The initial point is always $(100,100)$.

Only the trajectory planning problem is empathized in this paper, the robot control problem is not the focus of it this paper. However, details of the robots movement equations can be found in Fujimori *et al.* [8]. It's assumed that the obstacles are circular in the robot's moving plan. Besides, the hypothesis that the free bidimensional space is connected and the obstacles are finite in size and they do not overlap the destiny point is true.

The optimization problem formulated consists of a discrete optimization problem, where the objective function $f(x,y)$, which is the connection between the bacteria colony and the environment, aims to minimize the total trajectory travel by the mobile robot is given by

$$f(x, y) = \alpha d_{obj} + \lambda n_o \quad (3)$$

$$d_{obj} = \sum_{i=0}^{n_p} \sqrt{(x(i+1) - x(i))^2 + (y(i+1) - y(i))^2} \quad (4)$$

where α and λ are weighted factors, d_{obj} represents the Euclidian distance between the initial and the destiny point, n_o denotes the number of obstacles hitten by the robot movement following the planned trajectory, and n_p is the number of points where a trajectory change occurs (project parameter in this article). It's noticed by the equation (3) that an λ term exists, it's a weighted (penalty) term for unfeasible solutions, meaning, the trajectory that intercepts obstacles. In this case, the fitness function to be evaluated by the bacteria colony aims to maximize

$$fitness = \frac{K_c}{f(x,y) + \varepsilon} \quad (5)$$

where K_c and ε are scale constants.

4 Simulation Results

In order to start the system, the parameters p (optimization problem's dimension), S (population size), N_c (number of chemotactic steps), N_s (maximum number of steps that a bacterium can swim in a turn), N_{re} (number of reproductions), N_{ed} (number of elimination-dispersals events), p_{ed} (elimination-dispersal probability) and $C(i), i=1,2,\dots,S$ (speed of the movement taken in one step). The starting values for $\theta, i=1,2,\dots,S$ (positions) where chosen randomly with normal distribution. In this paper is adopted, $S = 50, N_c = 15, N_s = 4, N_{re} = 4, N_{ed} = 2, p_{ed} = 0.3$ and $C(i) = 2.5, i=1,2,\dots,S$.

The environment used for the trajectory planning is a 100x100 meters field. The search interval of the parameters using the bacteria colony is $x_i \in [0,100]$ meters and $y_i \in [0,100]$ m, where $i=1,\dots,np$. About the fitness it's adopted $\alpha=1, \lambda=200, K_c=100$ and $\varepsilon=1 \times 10^{-6}$. Afterwards are presented two simulated cases and the results analysis after 10 experiments with the bacteria colony algorithm.

As a comparative study a genetic algorithm is also applied in the simulated cases, the genetic algorithm is configured with the selected parameters, population size 50, crossover probability 0.85, mutation probability 0.15, maximum number of generations 300, the selection operator is the roulette wheeling, and the size of each chromosome is set to 16.

4.1 Case study 1: Environment with 4 obstacles

In Table 1 are presented the positions of the centers (x_c, y_c) of the circular obstacles and their respective radius (in meters) of case 1. The results obtained with the bacteria colony are restricted to $p=3$. In Tables 2 and 3 the achieved solutions are presented.

Table 1. Definition of obstacles for the case study 1.

Obstacle n ^o	Radius	Position (x, y)
1	10	(40, 15)
2	10	(20, 35)
3	20	(75, 60)
4	15	(35, 75)

Table 2. Results for an environment with 4 obstacles for the bacteria colony.

Experiment	Mean fitness	Maximum fitness	Minimum fitness	Standard deviation
1	0.2141	0.6950	0.0513	0.1044
2	0.2047	0.6902	0.0566	0.1099
3	0.2166	0.6916	0.0572	0.1036
4	0.2089	0.6906	0.0527	0.1059
5	0.2134	0.6912	0.0525	0.0986
6	0.2054	0.6934	0.0559	0.0869
7	0.2107	0.6889	0.0567	0.0910
8	0.2243	0.6915	0.0577	0.1213
9	0.2082	0.6908	0.0522	0.1032
10	0.2154	0.6952	0.0570	0.0932

The best (higher) fitness that the bacteria colony has achieved, for $p=3$, has been obtained with the solution: $(x_1, y_1) = (64.8631; 79.6103)$, $(x_2, y_2) = (37.2277; 45.1039)$ and $(x_3, y_3) = (28.3488; 28.6159)$. In Figure 2 the best result of the experiments is presented. In case study 1, the best of bacteria colony obtains a distance total of path of 143.8403. This distance is 98.31% of optimum path without obstacles. From 10 repeated simulations, it is shown that a robust convergence of path planning is obtained after about 120 generations (see Figure 2).

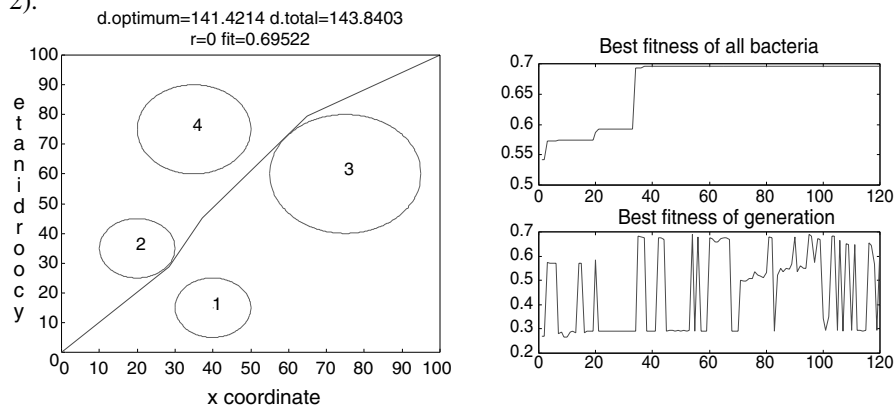
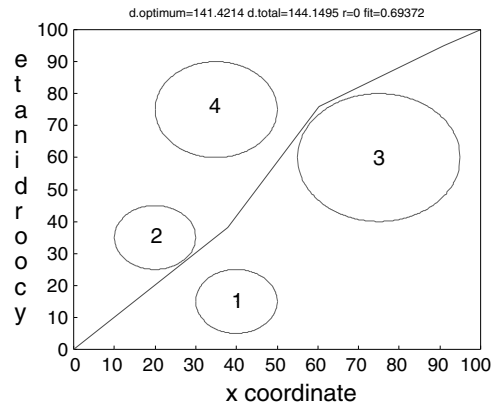


Fig. 2. Best result achieved for the bacteria colony in case study 1.

Table 3: Results for an environment with 4 obstacles for the genetic algorithm.

Experiment	Mean fitness	Maximum fitness	Minimum fitness	Standard deviation
1	0.1507	0.6915	0.0579	0.1317
2	0.1470	0.6937	0.0681	0.1047
3	0.1491	0.6838	0.0654	0.0975
4	0.1318	0.6788	0.0632	0.1011
5	0.1113	0.6868	0.0579	0.1373
6	0.1490	0.6930	0.0594	0.0963
7	0.1706	0.6933	0.0665	0.1441
8	0.1473	0.6881	0.0647	0.0970
9	0.1668	0.6912	0.0674	0.1207
10	0.1360	0.6915	0.0580	0.0936

The best (higher) fitness that the genetic algorithm has achieved, for $p=3$, has been obtained with the solution: $(x_1, y_1) = (37.9751; 38.1659)$, $(x_2, y_2) = (60.2136; 75.8572)$ and $(x_3, y_3) = (90.9789; 94.9493)$. In Figure 3 the best result of the experiments is presented. In case study 1, the best of genetic algorithm obtains a distance total of path of 144.1495. This distance is 98.10% of optimum path without obstacles.

**Fig. 3.** Best result achieved for the genetic algorithm in case study 1.

As seen above, bacteria colony achieves a better solution than genetic algorithms, but the difference between them is small in simple environments.

4.2 Case study 2: Environment with 12 obstacles

In Table 4 are presented the center positions (x_c, y_c) of the circular obstacles and their respective radius (in meters) for case 2. The results obtained for the bacteria colony are restricted to $p=4$. In Tables 5 and 6, the results for the case study 2 are summarized.

Table 4. Obstacles for case study 2.

Obstacle n ^o	Radius	Position (x, y)
1	05	(50, 50)
2	10	(75, 75)
3	10	(50, 70)
4	05	(20, 20)
5	10	(40, 15)
6	10	(70, 10)
7	08	(65, 40)
8	10	(20, 60)
9	10	(30, 40)
10	08	(85, 50)
11	05	(60, 90)
12	08	(20, 80)

In Figure 4, the best result of the experiments is presented. In this case, the best result of the experiments was: $(x_1, y_1) = (85.2011; 70.9190)$, $(x_2, y_2) = (76.2639; 62.0575)$, $(x_3, y_3) = (56.6595; 44.1203)$ and $(x_4, y_4) = (23.09478; 13.0078)$.

Table 5. Results for an environment with 12 obstacles with bacteria colony.

Experiment	Mean fitness	Maximum fitness	Minimum fitness	Standard deviation
1	0.1405	0.6810	0.0300	0.0830
2	0.1360	0.6942	0.0290	0.0808
3	0.1409	0.6866	0.0329	0.0816
4	0.1403	0.6438	0.0331	0.0681
5	0.1274	0.6673	0.0294	0.0561
6	0.1424	0.6805	0.0169	0.0853
7	0.1503	0.6833	0.0294	0.0821
8	0.1416	0.6907	0.0321	0.0710
9	0.1360	0.6942	0.0290	0.0808
10	0.1405	0.6810	0.0300	0.0830

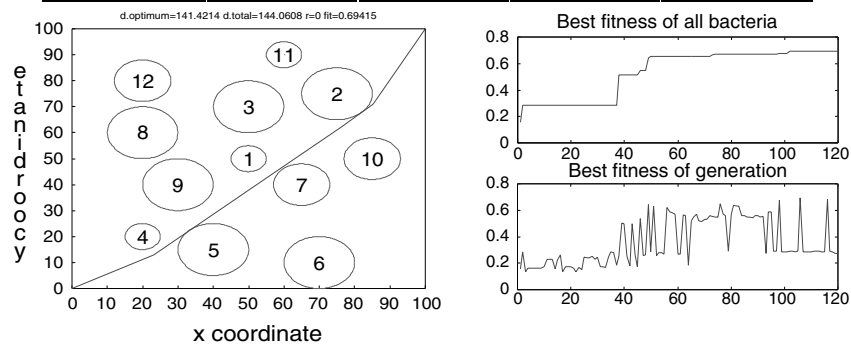


Fig 4. Best result achieved for the bacteria colony in case study 2.

In case study 2, the best of bacteria colony obtains a total distance of 144.0608. This distance is 98.17% of optimum path without obstacles. The average performance would be useful as an indication of the robustness of the configuration of bacteria colony. In Table 5 is observed that the bacterial colony responded well for all the simulations attempts. From 10 repeated simulations, it is shown that the results of bacterial colony were significant for path planning about 120 generations (see Figure 4) for the case study 2.

In Figure 5, the best result of the experiments is presented. In this case, the best result of the experiments was: $(x_1, y_1) = (2.8168; 24.3320)$, $(x_2, y_2) = (9.1478; 67.0436)$, $(x_3, y_3) = (88.9937; 93.6584)$ and $(x_4, y_4) = (94.6136; 88.3787)$.

In case study 2, the best of genetic algorithm obtains a total distance of 172.3573. This distance is 82.05% of optimum path without obstacles. As seen in the comparative study, the robustness of the bacteria colony is higher than the one of the genetic algorithms when dealing with complex environments.

Table 6. Results for an environment with 12 obstacles with genetic algorithm.

Experiment	Mean fitness	Maximum fitness	Minimum fitness	Standard deviation
1	0.0976	0.5560	0.0313	0.0797
2	0.0876	0.5802	0.0304	0.0796
3	0.0779	0.5413	0.0368	0.0705
4	0.1036	0.5567	0.0256	0.1052
5	0.1010	0.5521	0.0369	0.0828
6	0.0897	0.5462	0.0402	0.0763
7	0.0816	0.5565	0.0352	0.0733
8	0.1042	0.5616	0.0410	0.1041
9	0.0909	0.5605	0.0396	0.0836
10	0.0890	0.5544	0.0370	0.0757

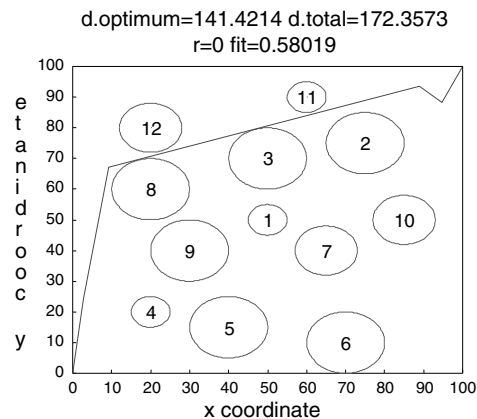


Fig. 5. Best result achieved for the genetic algorithm in case study 2.

5 Conclusion and future works

A research area with special relevance to mobile robot systems is devising suitable methods to plan optimum moving trajectories. There exist many approaches within the area of evolutionary computation and swarm intelligence to solve the problem of optimization of path planning in mobile robotics. In this paper the application of the forage theory of a bacteria colony in form of an optimization algorithm is explored for this purpose.

Must be emphasized that the bacteria own a control system that allows it is foraging capabilities and avoiding noxious substances. In this paper, the possibilities of exploring the bacteria colony efficiency are successfully presented, as shown in two simulated cases study. The results of these simulations are very encouraging and they indicate important contributions to the areas of swarm intelligence and path planning in robotics.

However, in future works, more detailed studies related to population size, number of chemotactic steps, number of swimming steps, number of elimination-dispersal events and movement speed are necessary. In this context, a comparative study of bacteria colony with evolutionary algorithms and others swarm intelligence methodologies will be done.

References

1. Alon, U.; Surette, M.G.; Barkal; Leibler, S. (1999) Robustness in bacterial chemotaxis, *Nature* 397, 14 January 1999, pp. 168-171
2. Baras, J.S.; Tan, X.; Hovareshti P. (2003) Decentralized control of autonomous vehicles, *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, pp. 1532-1537
3. Bennewitz, M.; Burgard, W.; Thrun, S. (2002) Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots, *Robotics and Autonomous Systems* 41, pp. 89-99
4. Bremermann, H.J. (1974) Chemotaxis and optimization, *Journal of Franklin Institute* 297, pp. 397-404
5. Charon, N.W.; Goldstein, S.F. (2002) Genetics of mobility and chemotaxis of a fascinating group of bacteria: the spirochetes, *Annual Review of Genetics* 36, pp. 47-73
6. Dhariwal, A.; Sukhatme, G.S.; Requicha, A.A.G. (2004) Bacterium-inspired robots for environmental monitoring, *Proceedings of the IEEE International Conference on Robotics & Automation*, New Orleans, LA, pp. 1496-1443
7. Dorigo, M.; Di Caro, G. (1999) The ant colony optimization meta-heuristic, in D. Corne, M. Dorigo, and F. Glover (editors), *New Ideas in Optimization*, McGraw-Hill, pp. 11-32
8. Fujimori, A.; Nikiforuk, P.N.; Gupta, M.M. (1997) Adaptive navigation of mobile robots with obstacle avoidance, *IEEE Transactions on Robotics and Automation* 13(4), pp. 596-602

9. Gemeinder, M.; Gerke, M. (2003) GA-based path planning for mobile robot systems employing an active search algorithm, *Applied Soft Computing*, 3, pp. 149-158
10. Kennedy, J.F.; Eberhart, R.C.; Shi, R.C. (2001) *Swarm intelligence*. San Francisco: Morgan Kaufmann Pub
11. Liu, Y.; Passino, K.M. (2004) Stable social foraging swarms in a noisy environment, *IEEE Transactions on Automatic Control* 49(1), pp. 30-44
12. Melchior, P.; Orsoni, B.; Lavaialle, O.; Poty, A.; Oustaloup, A. (2003) Consideration of obstacle danger level in path planning using A* and fast-marching optimization: comparative study, *Signal Processing* 83, pp. 2387-2396
13. Mello, B. A.; Tu, Y. (2003) Perfect and near-perfect adaptation in a model of bacterial chemotaxis, *Biophysical Journal* 84, pp. 2943-2956
14. Müller, S.D.; Marchetto, J.; Airaghi, S.; Koumoutsakos, P. (2002) Optimization based on bacterial chemotaxis, *IEEE Transactions on Evolutionary Computation* 6(1), pp. 16-29
15. Passino, K.M. (2002) Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Systems* 22(3), pp. 52-67
16. Silva, L.N.C.; Timmis, J.I. (2002) Artificial immune systems: a new computational intelligence approach, Springer-Verlag, London
17. Tsuji, T.; Tanaka, Y.; Morasso, P.G.; Sanguineti, V.; Kaneko, M. (2002) Bio-mimetic trajectory generation of robots via artificial potential field with time base generator, *IEEE Transactions on Systems, Man and Cybernetics - Part C* 32(4), pp. 426-439
18. Tu, J.; Yang, S.X. (2003) Genetic algorithm based path planning for a mobile robot, *Proceedings of the IEEE International Conference on Robotics & Automation*, Taipei, Taiwan, pp. 1221-1226
19. Xiao, J.; Michalewicz, Z.; Zhang, L.; Trojanowski, K. (1997) Adaptive evolutionary planner/navigator for robots, *IEEE Transactions on Evolutionary Computation* 1(1), pp. 18-28
20. Ward, M. (2001) BT ponders bacterial intelligence, *BBC News Online Technology*, 13 September 2001