Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol. 6 No. 1, 2017, pp.21-26

Evolutionary Algorithms Techniques Based on MET Heuristics of Accustomed Computing Performance

P. Senthil
Associate Professor in MCA Computer Science,
Kurinji College of Arts and Science, Tiruchirappalli, Tamil Nadu, India
Email: senthilkanapathy@gmail.com

Abstract **- In robotics manipulators, the aisle should be optimum, appropriately the torque of the apprentice can be minimized in adjustment to save power. This cardboard includes an optimal aisle planning arrangement for a automatic manipulator. Recently, techniques based on metaheuristics of accustomed computing, mainly evolutionary algorithms (EA), accept been auspiciously activated to a ample amount of automatic applications. In this cardboard the bigger BBO algorithm is acclimated to abbreviate the cold action in the attendance of altered obstacles. The simulation represents that the proposed optimal aisle planning adjustment has satisfactory performance.**
Keywords: **Biogeography based Optimization-Aisle planning-Obstacle detection-Robotic Manipulator**

I. INTRODUCTION

One of the important aspect of industrial robots is the motion planning. In designing process of the robot the first task is defining the geometry of the path. Two kinds of motions are introduced in robotics path planning literature: The point to point and the continuous path control motions. In the first one only the control point of the motion is considered and the rest of the path is found by the robot. In the second category, the speed and the position of the robot has to be estimated in each point of the path. In [1] the problem of the point-to point of redundant robot in presence of the obstacles is considered. GA is used to solve the problem.

In [2]GA is used to search the robot configuration space for the optimum path. Also, GA is used to generate the position the configuration of a mobile robot[3]

The direct kinematics and the inverse dynamics are adopted to design path planning method using genetic algorithms [4].

In [5]the problem of the point to point trajectory of flexible redundant manipulators is considered. The objective in [6] is to minimize the time ripple in the trajectory of the robot without colliding with any obstacles. The trajectory planning is studied in [7] for a two-degree-of freedom robot with point obstacles. The aim is moving the end-effectors of the robot from a initial starting point to a target avoiding the point obstacles. Online trajectory of robot arms are introduced in [8].

Recently a new optimization concept, based on biogeography has been introduced [9] .This method has good convergence properties on various benchmark functions. It represents the biology behavior of immigration and emigration of animal species among islands. One characteristic of BBO is that the original population is not discarded after each generation. Various versions of biogeography-based optimization modals were proposed in [9, 10]

Biogeography Based Optimization (BBO) is a population-based evolutionary algorithm (EA) and it adopts the migration operator to share information among solutions. This feature makes BBO applicable to the majority of problems, where GA and PSO are applicable.

Biogeography based optimization can be used to avoid obstacles for a robot manipulator in this paper. Each joint angle is obtained according to the inverse kinematics solutions of manipulator with it. Collisions between obstacles and manipulator area voided for each trajectory and the compliance of jointmotion is ensured.

The rest of this paper is organized as follows: In section 2 BBO algorithms is explained. In section 3 robotic tracking problems is formulated. In order to prove the effectiveness of the proposed controllers, simulation results have been presented in section 4.

II. BBO ALGORITHM

Different methods of recognizing the robot trajectory has been researched in recent years. BBO is a kind of heuristic algorithm developed for the general-purpose optimization which is based on the distribution of biological organisms. In this method, there are some unique specifications in compare with other heuristic optimization algorithms. Indeed, in BBO the feasible problem solutions are known as habitats or islands. And they are going to be solved via the context of migration. The manner of information sharing between the problem solutions is presented by migration context.

There are some main parameters which had introduced in BBO algorithm. Suitability index variable (SIV) shows a variable that denote habitability in a solution. Habitat suitability index (HSI), shows a variable that describes the goodness of the solution. The solutions with high HSIs are well suited as residences for biological species.

Immigration rate (λ), shows that acceptation features from other solutions is how likely to one solution. Emigration rate (μ), shows that sharing features with other solutions is how likely to one solution. Generally the number of species in each habitat (solution) is S, and when there are zero species in a habitat, maximum possible immigration rate occurs. There are some parameters introduced in BBO as following:

TABLE 1 PARAMETERS USED IN BBO ALGORITHM

Parameters	BBO parameters
	Description
λ	shows how likely a solution is to be came in.
μ	shows how likely a solution is to share its information
SIV	*suitability index variables* known as vector of integers
HSI	*habitat suitability index*, high *HSI* solutions show
S	the number of species in each habitat
S_{max}	the largest possible number of species
I	the maximum possible immigration to the habitat that happens when there are any species in it
E	the maximum rate that happens when the immigration had stoped

In BBO algorithm $\lambda_k$and $\mu_k$are defined as emigration rate and immigration rate. For each habitat as following:

$$\lambda_k = \frac{Ek}{p}$$

$$\mu_k = I\left(1 - \frac{k}{P}\right)$$

where *P* is individual number and *k* is the number of habitat.According to the mentioned description, the species can be modelled as follows:

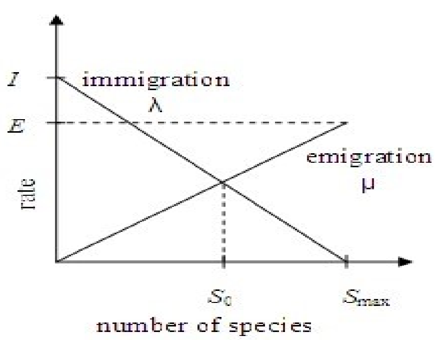


Fig.1 Species model of a single habitat [9]

III. MOTION PLANNING

Considering the beginning point to the destination of trajectory plan of a robot arm, the problem of finding the shortest way to be followed is handled in this paper. The problem includes detecting the shortest path toward the destination without happening object collision. To achieve this particular, the trajectory plan should be divided into segments by intermediate points known as “spline”.

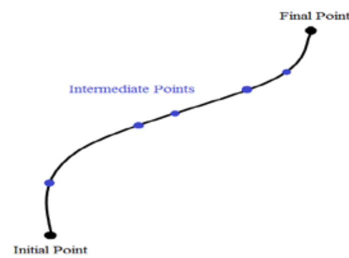


Fig.2 Intermediate points on the point-to-point trajectory

It’s obvious that the path can be consisting of several possible choices. This possibility is evaluated by *Violation.* The methodology is also executable to more complicated obstacles.

Suppose that the trajectory is coordinated. It's clear that all paths are not coordinated. This suppose infers the possibility of excluding on optimal path in the field with more complicated obstacles. As we know, coordinated paths are able to imply different trajectories.

The system concludes two sections:

1. Global path planning which works to detect the optimum path for the whole terrain. In this case, optimum path is the shortest way from the initial to final point without object collision.
2. The local motion planning which works to detect the optimum path and plans the controlling program in a promontory.

The primary population was generated at random and estimated by the estimation function. Splines are modelled as *“n”*, so in creating random solution, *n* solutions were taken between maximum and minimum ranges. We define *n=5* here in each iteration.

At the other hand, μ and λ are represented as follows:

$$\mu = \frac{nPop + 1 - (1{:}nPop)}{nPop + 1}$$

$$\lambda = 1\text{-}\mu$$

As it was stated in the probability of migration from X_i to X_j , the habitat ,which had the most probability, is allocated the large amount of the Roulette Wheel; so the possibility of the election gets higher.

We can formulate this action as follows:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

The Roulette Wheel Selection can be modelled as shown in figure 3.

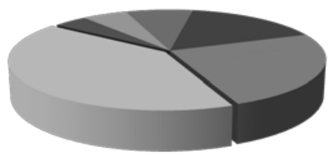


Fig.3 Roulette wheel selection Example

XS and *YS* are *3-1* matrixes consist of initial position, final position and spline coordinates. The function that we want to minimize is:

$$L = \sum \sqrt{dx^2 + dy^2}$$

According to above descriptions about violation, it should be calculated to avoid robot collide with obstacles. This function is described as follows:

$$d = \sqrt{(xx - xc)^2 + (yy - yc)^2}$$

$$V = max\ (1 - {}^{d}/_{r})$$

Violation is the mean value of the elements along different dimension of *V*. according to above methods, *Z* is depicted to exert *Violation* as following:

$$Z = L \times (1 + \beta \times Violation)$$

Where β is a positive integer.

Thus, the proposed steps of mentioned algorithm are as bellow:

1. Initialize the BBO parameters.
2. Generating the primary population and habitats.
3. Defining λ and μ as emigration and immigration rates for each habitat.
4. Selecting SIV_k of X_i .
5. Evaluating the emigration rate (λ_i) of X_i .
6. Selecting the destination by considering the rate of immigration (μ).
7. Replacing the new habitats (X_{ik}) with old ones (X_{jk}).
8. Applying mutation operation on X_{jk} with the probability of m_j .
9. Evaluating new habitats.
10. Replacing the new habitats and selecting the best, if the final population is matched by our target. Otherwise, going to step 4.

These steps can be summarized as the flowchart of fig 4.

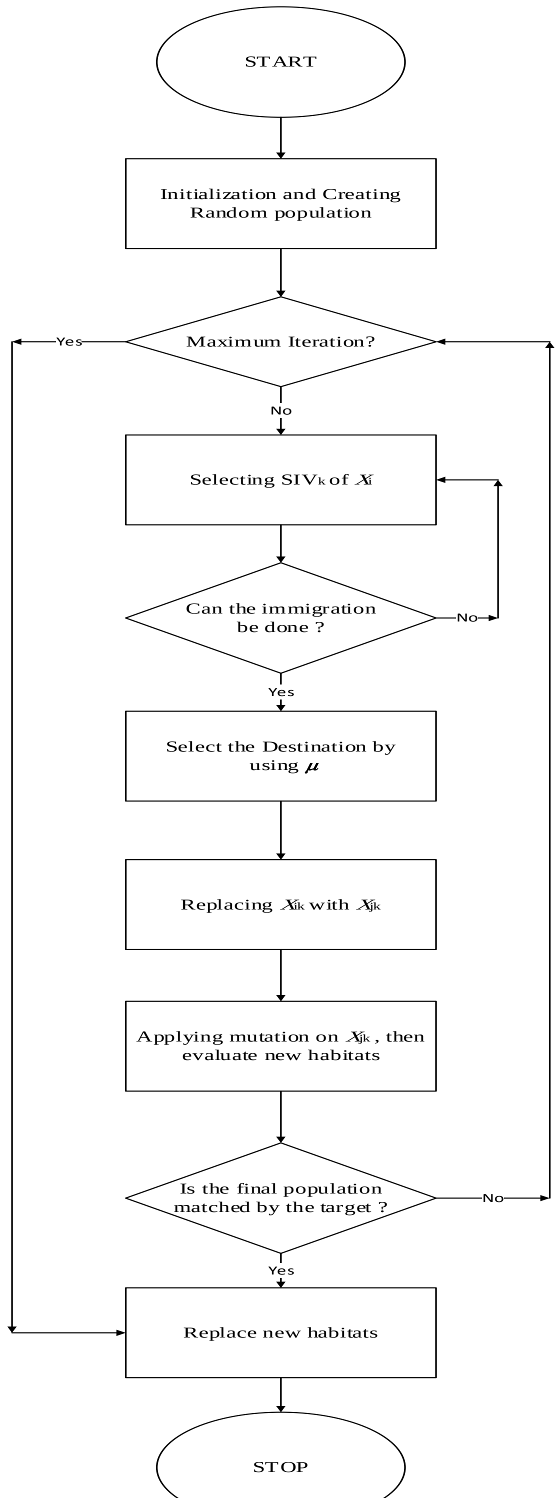


Fig.4 Flowchart of improved BBO algorithm

IV. SIMULATION

Simulations were done to indicate the efficiency of the proposed method. In this case, the 2 degrees of freedom manipulator is selected to be studied because of its plain and for the case of proving the esteem of the approaching proposed. It can be applicable to more complicated manipulators with more joints.
The conditions of initial and final boundaries for angular displacement and velocities are applied to two equations for each link.
It consists of two-ax model as it is shown in Fig.5.

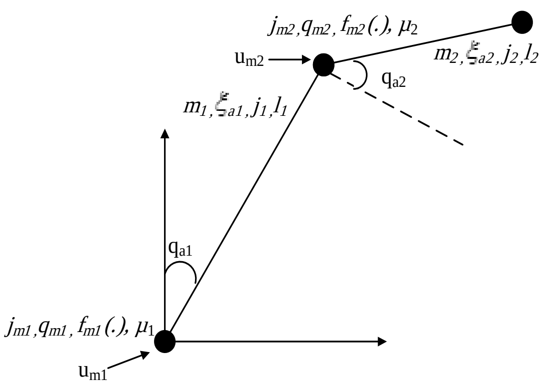


Fig.5 2DOF manipulator

Each link can be defined by mass mi, length l_i, center of mass m_i and inertia j_i due to the center of mass. The joints are modeled as a spring damping pair with nonlinear spring torque τ_{si} and linear damping d_i. The deflection in each joint is given by the arm angle q_{ai} and the motor angle q_{mi}. The motor characteristics are given by the inertia j_{mi} and a nonlinear friction torque f_i.

$$q = (q_{a1} \quad q_{a2} \quad q_{m1}/\mu_1 \quad q_{m2}/\mu_2)^T$$
$$u = (0 \quad 0 \quad u_{m1}\mu_1 \quad u_{m2}\mu_2)^T$$

which u_{mi} is known as the motor torque and μ_i is the gear ratio which can be defined as follows:

$$\mu_i = q_{mi} = q_{ai} > 1$$
$$\mu_i = q_{mi} = q_{ai} > 1$$

We assume two circles as obstacles of the simulation:
First obstacle: $\begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$ as coordinate of its center and *r = 1.3*
Second obstacle: $\begin{bmatrix} 2 \\ 4 \end{bmatrix}$ as coordinate of its canter and *r = 2* as the ray.

Parameter values for BBO algorithm in this paper are:

TABLE 2 VALUES USED IN SIMULATION

Values	**Parameter values of BBO**
	Description
nPop = 50	Population size
MaxIteration= 1000	Maximum number of iterations
var_{min} = -10	Lower bound of variables
var_{max} = 10	higher bound of variables

Splines are defined as n=5. Pop is as a nPop*1 matrix which each row is copied from the habitat matrix.
The new result is produced by the following formula:

$$newpop(i).position(k) = pop(i).position(k) + \alpha(pop(j).position(k) - pop(i).position(k))$$

Where $0 \prec \alpha \prec 1$ is defined as 0.9 and j is the result of the Roulette Wheel Selection function which is described in this paper.
Mutation is also checked after migration per iteration as bellow:

$$newpop(i).position(k) = newpop(i).position(k) + \sigma \times rand$$

Where σ=0.02×($\mathrm{Var_{max}}$-$\mathrm{Var_{min}}$) and rand is a random number in [0,1].

Now we have to calculate penalty function to avoid object collision with the following formula

$$Z = L \times (1 + \beta \times Violation)$$

which β=10.
By following these steps, the optimized path will be obtained.

As mentioned in this paper, BBO algorithm should be compared with PSO algorithm that has great outcome than the other heuristic algorithms. All of the parameters are the same as BBO algorithm simulation. Here, c1=c2=2 and ω = 1.
Velocity of the algorithm is calculated by the following formula:

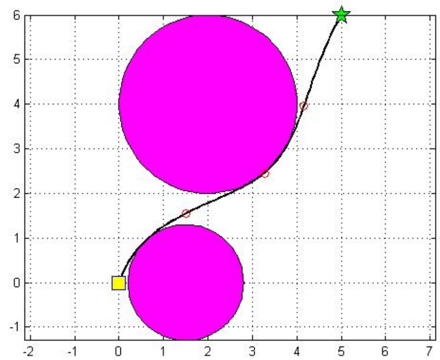
$$\mathrm{Velocity_{max}} = \sigma \times (\mathrm{Var_{max}} - \mathrm{Var_{min}})$$
$$\mathrm{Velocity_{min}} = -\mathrm{Velocity_{max}}$$

In this case, both position and velocity are being updated.

If either $Particle(i).position \prec \mathrm{var}_{\min}$ or $Particle(i).position \succ \mathrm{var}_{\max}$, replace $\mathrm{var}_{\min}$ and $\mathrm{var}_{\max}$ instead of the results obtained.

PSO algorithm is based on finding the best solutions like BBO algorithm. Thus, the personal and global best solutions are being updated in each iteration.

In figure6, we see the result of the BBO algorithm in the case of optimizing through this paper. Figure7 is also the result of the PSO algorithm in the case of optimizing as it is explained above .



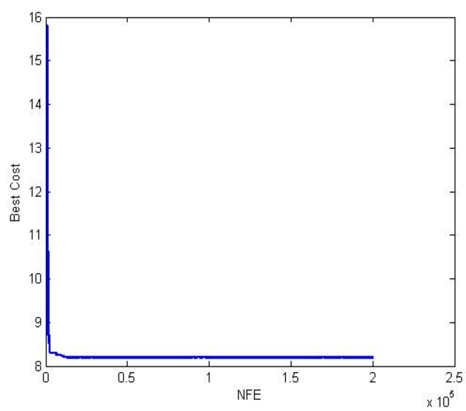
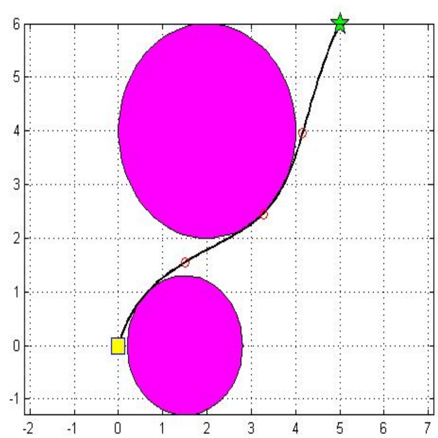


Fig.6 the results of PSO algorith



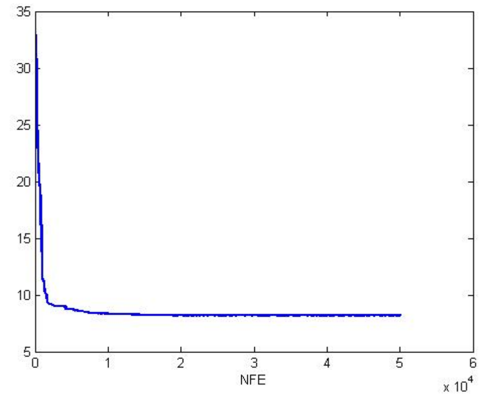


Fig.7 the results of BBO algorithm

Table 3 is drew to compare the results of the achieved optimization in simulations. It is obvious that BBO algorithm has better result than PSO algorithm.

TABLE 3 COMPARING THE RESULTS ACHIEVED IN BOTH ALGORITHMS

Comparison	
PSO	8.1911
BBO	7.9913

V. CONCLUSION

The paper proposed BBO algorithm for path planning and trajectory of a robotic manipulator in accosting with obstacles. Thus, the optimum trajectory gained, had minimum length without any collision with the obstacles in the route. The position of initial and final points are specified and 5 intermediate points are considered .The BBO is presented for a 2D path planning and it's performance is evaluated by simulation. The results of simulation that are reported in the paper indicate that BBO algorithm is able to approach a pre-specified goal with minimum length trajectory and in a sensible number of iterations.

As shown in the results, path planning by using BBO in compare with outcomes of path planning based on PSO algorithm, has less trajectory length, less mathematical complexity but relatively simple. It's concluded that the suggested method can significantly enhance the efficiency of robot manipulator.

REFERENCES

[1] M. Abbaszadeh and H. J. Marquez, "Nonlinear observer design for one-sided Lipschitz systems," in *American Control Conference (ACC), 2010*, 2010, pp. 5284-5289.

[2] A.Nearchou and N. Aspragathos, "Collision-free continuous path control of manipulators using genetic algorithms," *Journal of systems engineering*, vol. 6, pp. 20-32, 1996.

[3] M. Chen and A. M. Zalzala, "A genetic approach to the motion planning of redundant mobile manipulator systems considering safety and configuration," 1995.

[4] E. S. Pires and J. Machado, "A GA perspective of the energy requirements for manipulators maneuvering in a workspace with obstacles," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 1110-1116.

[5] S. Yue, D. Henrich, W. Xu, and S. Tso, "Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms," *Robotica*, vol. 20, pp. 269-280, 2002.

[6] E. S. Pires, J. T. Machado, and P. M. Oliveira, "An evolutionary approach to robot structure and trajectory optimization," in *Transactions of the 10th IEEE Conference on Advanced Robotics*, 2001, pp. 333-338.

[7] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, pp. 455-470, 2004.

[8] D. Chwa, J. Kang, and J. Y. Choi, "Online trajectory planning of robot arms for interception of fast maneuvering object under torque and velocity constraints," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, pp. 831-843, 2005.

[9] D. Simon, "Biogeography-based optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 12, pp. 702-713, 2008.

[10] A. Ghavifekr, S. Ghaemi, and R. Behinfaraz, "A Modified Biogeography Based Optimization (BBO) Algorithm for Time Optimal Motion Planning of 5 DOF PC-based Gryphon Robot."