IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X
PP. 28-34
www.iosrjournals.org

Bacterial Foraging Optimization Algorithm for Job Shop Scheduling

P.Parvathi Srutha Keerthi[1]
[1]*Asst. Prof., Department of Mechanical Engineering, St. Martin's Engineering College, Hyderabad, Telengana, India*

Abstract: *Scheduling, one of the most important decision making process in the operation of manufacturing systems, is defined as a series of activities where it is required to allocate available resources to jobs by determining the exact production schedules for achieving a set of objectives. The dynamic nature of manufacturing systems makes the real life scheduling environment extremely complicated. To ensure the sustainability in this environment, generating effective production schedules in real time is the key in the operational aspect. Job shop scheduling problems (JSSPs) have been studied extensively and most instances of JSSP are NP-hard, which implies that there is no polynomial time algorithm to solve them. As a result, many approximation methods have been explored to find near optimal solutions within reasonable computational efforts. Swarm Intelligence techniques are one of the metaheuristics that provide near optimal solutions for JSSP. In this paper, Bacterial Foraging Optimization (BFO), a new comer in Swarm Intelligence techniques, is applied to solve JSSP. A MATLAB code for solving JSSP using BFO is developed and the bench marking instances are solved using the codes. The obtained results are compared with the best known solutions and Particle Swarm Optimization (PSO). PSO has given satisfactory results for JSSP when implemented individually and also as a variant when hybridized with other meta heuristics.*

Keywords: *Bench marking Instances, BFOA, JSSP,PSO, Make span*

I. Introduction

Manufacturing facilities are complex, dynamic, stochastic systems. Many manufacturing organizations generate and update production schedules, which are plans that state when certain controllable activities (e.g., processing of jobs by resources) should take place. Production schedules coordinate activities to increase productivity and minimize operating costs. A production schedule can identify resource conflicts, control the release of jobs to the shop, ensure that required raw materials are ordered in time, determine whether delivery promises can be met, and identify time periods available for preventive maintenance.

The two key problems in production scheduling are "priorities" and capacity.[1] Research in scheduling theory has evolved over the past forty years and has been the subject of much significant literature with techniques ranging from unrefined dispatching rules to highly sophisticated parallel branch and bound algorithms and bottleneck based heuristics. One of the most popular models in scheduling theory is that of the job-shop, as it is considered to be a good representation of the general domain and has earned a reputation for being notoriously difficult to solve. Job shop scheduling have been defined as providing scheduling for a shop floor with n jobs, m machines .If the completion time of J_i on M_k is C_{ik} then the duration in which all operations for all jobs are completed is referred to as the makespan C_{max}. In the optimisation variant of Π_J the objective of the scheduler is to determine starting times for each operation, $t_{ik} \geq 0$, in order to minimise the makespan while satisfying all the precedence and capacity constraints. That is, the goal maybe expressed as determining C^*_{max}, where ,

$$C^*_{max} = \min(C_{\max}) = min_{feasible\ schedules}(\max(t_{ik} + \tau_{ik}) : \forall J_i \in J, M_k \in M)$$

Each job follows its own sequence. The Job Shop Scheduling Problem(JSSP) belongs to the class of NP-Hard, which means that it is hard to solve the problem in polynomial time.

In this paper, a metaheuristic approach for solving JSSP has been presented. Swarm Intelligence techniques have gained importance in this field over the past few years. Following the same trend of swarm-based algorithms, Passino proposed the Bacterial foraging optimization algorithm BFOA. Application of group foraging strategy of a swarm of E.coli bacteria in multi-optimal function optimization is the key idea of the new algorithm.[1] This algorithm have been used in this paper to provide a near optimal solution for this problem. A few bench marking JSSP instances have been taken as a case study and the BFOA has been applied to the bench marking instances. The obtained results are compared with the best known solutions. Another Swarm

Intelligence technique Particle Swarm Optimization(PSO) has also been used to solve these Bench marking instances and a comparison has been made between the performance of these two algorithms .

II. Literature review

Foraging Optimization Algorithm (BFOA), proposed by Passino, is a new comer to the family of nature-inspired optimization algorithms. Passino has explained hoe the BFOA mimics the foraging behavior of the E.Coli Bacteria over a landscape of nutrients to perform parallel non gradient optimization in a paper presented in the International Journal of Swarm Intelligence research in 2005.[3]Since then, many researchers have applied BFOA and its variants for solving different problems.

Gautam Mahapatra and Soumya Banerjee in their paper A Study of Bacterial Foraging Optimization Algorithm and its Applications to Solve Simultaneous Equations showed how the bio-inspired Bacteria Foraging Optimization Algorithm (BFOA), can be used to solve such system of equation with rank less than or equal to n. [4]

In 'enhanced bacterial foraging algorithm for permutation flow shop scheduling problems' Shivakumar B. L.and Amudha.T dealt with one of the significant types of scheduling problems, the permutation flow shop scheduling problem. The competence of bacterial problem solving and a proposed hybrid bacterial swarming technique were analyzed by applying them to benchmark problems of permutation flow shop. A comparative analysis of the results indicates the improvement in scheduling efficiency in terms of reduced cost through the application of bio-inspired techniques.[5]

In Bacterial Foraging for Engineering Design Problems: Preliminary Results, Efr´en Mezura-Montes and Betania Hern´andez-Ocana have presented an approach based on the bacterial foraging optimization algorithm (inspired on bacteria moving in their environment looking for high-nutrient areas) to solve engineering design problems. [6]

Narendhar.S and Amudha. T in their research work, 'A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems', hybridized Bacterial Foraging Optimization was with Ant Colony Optimization and a new technique Hybrid Bacterial Foraging Optimization for solving Job Shop Scheduling Problem was proposed. [7]

Divya. P, Narendhar. S and Ravibabu. V in An Enhanced Bio-Stimulated Methodology to Resolve Shop Scheduling Problems symbolized the efficiency of Customized Bacterial Foraging Optimization algorithm. In this research work, Bacterial Foraging Optimization was combined with Ant Colony Optimization and a new technique Customized Bacterial Foraging Optimization for solving Job Shop Scheduling, Flow Shop Scheduling and Open Shop Scheduling problems were suggested. The Customized Bacterial Foraging Optimization was tested on the Benchmark instances and randomly created instances. [8]

Vipul Sharma S.S. Pattnaik Tanuj Garg in their paper 'A Review of Bacterial Foraging Optimization and Its Applications' presented an application based review of such variants .[9]

In "Synergy of PSO and BFO-A Comparative Study on Numerical Benchmarks",, the paper introduces a hybrid approach involving PSO and BFO for optimization of multi-modal and high dimensional functions. The algorithm combines PSO based mutation operator with bacterial chemotaxis in order to make judicious use of exploration and exploitation abilities of search space and to avoid false and premature convergence. The algorithm is tested on five standard functions Rosenbrock, Rastrigin, Griewank, Ackley, Shekel's Foxholes and also on spread spectrum radar poly phase code design. It is found that the overall performance of the hybrid algorithm is better than stand alone BFO and at least comparable to PSO and its variants.

In paper " Bacterial Foraging Oriented By Particle Swarm Optimization Strategy for PID Tuning", the Korani has applied BFO oriented by particle swarm optimization (PSO), called BFPSO, for tuning of PID (proportional derivative integral) controller kp, kI and kd. Conventional BFO depends on random search directions which may lead to delay in reaching global solution while PSO is prone to be trapped in local minima. In order to get better optimization, the new algorithm combines advantages of both the algorithms i.e. PSO's ability to exchange social information and BFO's ability in finding new solutions by elimination and dispersal. Simulation results demonstrated that overshoots of PID controller are reduced considerably with faster convergence and thus the hybrid algorithm outperforms conventional BFO and PSO.

The paper "Intelligent Bacterial Foraging Optimization Technique to Calculate Resonant Frequency of RMA" presents an hybrid approach involving BFO and PSO. The authors utilized faster convergence property of PSO for finding corresponding positions of bacteria in the predefined domain i.e. search space. By doing so they made the search space narrowed and thus reducing computational time. The change in resonant frequencies of different patches are calculated first and placed randomly in search space of PSO and then searching starts using PSO to nearest experimental values. Then theses values are placed in search space of BFO. Root mean square value (RMSE) is taken as fitness function in BFO. The results show promising improvement in accuracy and drastic reduction in time. [10]

Mahmoud in "Design Optimization of A Bow-Tie Antenna For 2.45 GHz RFID Readers Using A Hybrid BSO-NM Algorithm" used bacterial foraging oriented by particle swarm optimization (BF PSO or simply BSO) hybridized with Nelder-Mead algorithm to design bow-tie antenna for RFID readers. The BSO-NM algorithm is integrated with Method of Moments (MoM) to optimize the antenna. The algorithm optimizes bow tie antenna parameters i.e. height, neck width and flare angle to make it resonant at desired frequency. Reflection coefficient has been taken as cost function to be minimized. It is found that BFO-NM algorithm produced results better than those generated by individual BFO or BSO. [11]

K. M. Bakwad1, S.S. Pattnaik1, B. S. Sohi2, S. Devi1, B.K. Panigrahi3, Sanjoy Das4, M. R. Lohokare in their paer Hybrid Bacterial Foraging with Parameter free PSO presents fusion of Bacterial Foraging with parameter free Particle Swarm Optimization (HBFpfPSO). The proposed technique is used to enhance quality of global optima of multimodal functions.[12]

III. Bacterial Foraging Optimization Algorithm For Job Shop Schedulin

III.1 BFOA

BFOA and PSO belong to the group of swarm intelligence(SI) techniques which involves the study of the computational systems inspired by the collective intelligence of the homogenous agents in the environment. BFOA mimics the foraging strategies of the E.Coli bacteria in nutrient gradient environment. There are four important processes in BFOA:

- Chemotaxis- swim and tumble movements of the bacteria in search space to reach the nutrient rich environment avoiding noxious environment.
- Swarming- signaling among the bacteria to collective reach the nutrient rich space which is achieved by the release of attractants and repellents.

The cell to cell signaling between the bacteria is given by

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^{S} J_{cc}^{i}\left(\theta, \theta^{i}(j,k,l)\right)$$

$$= \sum_{i=1}^{S}\left[-d_{attract}\exp\left(-w_{attract}\sum_{p=1}^{m}(\theta_m - \theta_m^i)^2\right)\right]$$

$$+ \sum_{i=1}^{S}\left[-h_{repellent}\exp\left(-w_{repellent}\sum_{p=1}^{m}(\theta_m - \theta_m^i)^2\right)\right]$$

where

$\theta=[\theta_{1,\ldots,}\ \theta_p\ ]^T$ is a point on the optimization domain and θ^i_m is the m^{th} component of the i^{th} bacterium position θ^i.

$d_{attract}$ =0.1,be the depth of the attractant released by the cell (a quantification of how much attractant is released) and

$w_{attract}$ =0.2be a measure of the width of the attractant signal (a quantification of the diffusion rate of the chemical).

$h_{repellent}$ =$d_{attractant}$ be the height of the repellant effect (magnitude of its effect) and $w_{repellant}$ =10

- Reproduction- a process in which the healthy bacteria split in two and unhealthy bacteria die thereby improving the healthy population
- Elimination and dispersal- here the bacteria get dispersed all over the search space thereby increasing the chance of finding the global optimum.

BFOA to JSSP:

The BFOA algorithm in this work has been modified to suit the Job Shop Scheduling problem. Each parameter in BFOA have been taken like this:

- Objective of JSSP: makespan minimization
- Total number of operations o=n*m(n- number of jobs, m-number of machines)
- Population S=o
- Location of each bacteria=(t_{ij},k)[operation number for the i^{th} job on the j^{th} machine is k ($1 \leq k \leq n$) and the processing time for that job on that machine is t_{ij},
- dimension of the search space p is taken as 2
- Intial position of all bacteria (0,0)

- N_{re}=2
- P_{ed}=0.25

The algorithm obtains a schedule for each machine, j, (1≤j≤m), i.e., all jobs 1 to i are scheduled on j and once the schedule for the j^{th} machine is obtained, then the sequence proceeds to the next machine j+1.

Flow Chart for BFOA for JSSP:

- Initialize parameters p, S, Nc, Ns, Nre, Ned, Ped, C(i)(i=1,2…S),θ^i. t=time matrix and m=machine matrix
- Elimination-dispersal loop: l=l+1
- Reproduction loop: k=k+1
- Chemotaxis loop: j=j+1

```
[a] For i =1,2…S take a chemotactic step for bacterium i as follows.
[b] Compute fitness function, J (i, j, k, l).
temp=[t(1:i,j),mac(1:i,j)];
b1=0;b2=1;
r=(b2-b1).*rand(i,2)+b1;
 theta=temp.*r
for i=1:n
 jcc=[(0.1*[(-exp(-0.2*(theta(i,1).^2)))+(exp(-10*(theta(i,1).^2)))])+(0.1*[(-exp(-
0.2*(theta(i,2).^2)))+(exp(-10*(theta(i,2).^2)))])];
 [y k]=min(jcc);
jbest=jini+jcc;
 jini=jbest;
 pnext=pini+(ci*b);
 pini=pnext;
 seq=[seq,y];
end reproducuction step;
ed step
end
end
```

Thus the final sequence is obtained.

III.2 Particle Swarm optimization Algorithm(PSO)

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995, based on the swarm behavior such as fish and bird schooling in nature. PSO has been applied to almost every area in optimization, computational intelligence, and design of scheduling applications. The movement of a swarming particle consists of two major components: a social component and a cognitive component. Each particle is attracted toward the position of the current global best g∗ and its own best location x∗ i in history, while at the same time it has a tendency to move randomly. Let x_i and v_i be the position vector and velocity for particle i, respectively. The new velocity and location updating formulas are determined by

$V^{(t+1)}{}_i = v^{(t)}{}_i + \alpha v_1[g^* - x^t{}_i] + \beta v_2 [x^*{}_i - x^t{}_i]$.

$x^{t+1}{}_i = x^t{}_i + V^{(t+1)}{}_i$ where v_1 and v_2 are two random vectors, and each entry taking the values between 0 and 1. The parameters α and β are the learning parameters or acceleration constants, which can typically be taken as, say, α ≈ β ≈2. There are at least two dozen PSO variants which extend the standard PSO algorithm, and the most noticeable improvement is probably to use inertia function θ(t) so that vt i is replaced by θ(t)vt i where θ ∈ [0,1]. This is equivalent to introducing a virtual mass to stabilize the motion of the particles, and thus the algorithm is expected to converge more quickly.[13]

PSO for JSSP:

Each postion for a particle is given by the time and operation number. An initial population with some schedules is generated to find the p_{best}.The population is updated based on the makespan obtained for each sequence by retaining the best sequences and changing the inferior sequences. Best sequences are those with low makespan and the inferior ones usually have high make span value.

The flow chart for implementing PSO for JSSP is as follows:

1. Initialize maximum velocity V_{max} =0.5, number of jobs (n),number of machines(m), number of operations o=n*m; population size r=2*n; X- position vectors, V-velocity vectors
2. Initialize local best positions, global best positions, learning parameters, initial population
3. find the makespan for each sequence

4. updating the p_{best} value for i^{th} particle
if pbest(i) > makespan(i)
pbest(i)=makespan(i);
5. finding the g_{best}
6. updating the velocity and position vectors
7. finding the final makespan
8.end

III. Current Work

In the present work, 3 benchmarking instances ABZ6, ORB10, MT10 have been solved using BFOA and PSO. A comparison has been made between the obtained between the obtained results and the best known solutions.

Each instance consists of a line of description, a line containing the number of jobs and the number of machines, and then one line for each job, listing the machine number and processing time for each step of the job. The machines are numbered starting with 0[14]

1.ABZ6

Adams, and Zawack 10x10 instance (Table 1, instance 6)
10 10
7 62 8 24 5 25 3 84 4 47 6 38 2 82 0 93 9 24 1 66
5 47 2 97 8 92 9 22 1 93 4 29 7 56 3 80 0 78 6 67
1 45 7 46 6 22 2 26 9 38 0 69 4 40 3 33 8 75 5 96
4 85 8 76 5 68 9 88 3 36 6 75 2 56 1 35 0 77 7 85
8 60 9 20 7 25 3 63 4 81 0 52 1 30 5 98 6 54 2 86
3 87 9 73 5 51 2 95 4 65 1 86 6 22 8 58 0 80 7 65
5 81 2 53 7 57 6 71 9 81 0 43 4 26 8 54 3 58 1 69
4 20 6 86 5 21 8 79 9 62 2 34 0 27 1 81 7 30 3 46
9 68 6 66 5 98 8 86 7 66 0 56 3 82 1 95 4 47 2 78
0 30 3 50 7 34 2 58 1 77 5 34 8 84 4 40 9 46 6 44

2. ORB 10

Instance from George Steiner (GES2)
10 10
9 66 8 13 0 93 7 91 6 14 5 70 3 99 2 53 4 86 1 16
8 34 9 99 0 62 7 65 5 62 4 64 6 21 2 12 3 9 1 75
9 12 8 26 7 64 6 92 4 67 5 28 3 66 2 83 1 38 0 58
0 77 1 73 3 82 2 75 6 84 4 19 5 18 7 89 8 8 9 73
0 34 1 74 7 48 5 44 4 92 6 40 3 60 2 62 8 22 9 67
9 8 8 85 3 58 7 97 5 92 4 89 6 75 2 77 1 95 0 5
8 52 9 43 6 5 7 78 5 12 3 62 4 21 2 80 1 60 0 31
9 81 8 23 7 23 6 75 4 78 5 56 3 51 2 39 1 53 0 96
9 79 8 55 2 88 4 21 5 83 3 93 6 47 7 10 0 63 1 14
0 43 1 63 2 83 3 29 4 52 5 98 6 54 7 39 8 33 9 23

3. MT 10, Fisher and Thompson 10x10 instance, alternate name (mt10)
10 10
0 29 1 78 2 9 3 36 4 49 5 11 6 62 7 56 8 44 9 21
0 43 2 90 4 75 9 11 3 69 1 28 6 46 5 46 7 72 8 30
1 91 0 85 3 39 2 74 8 90 5 10 7 12 6 89 9 45 4 33
1 81 2 95 0 71 4 99 6 9 8 52 7 85 3 98 9 22 5 43
2 14 0 6 1 22 5 61 3 26 4 69 8 21 7 49 9 72 6 53
2 84 1 2 5 52 3 95 8 48 9 72 0 47 6 65 4 6 7 25
1 46 0 37 3 61 2 13 6 32 5 21 9 32 8 89 7 30 4 55
2 31 0 86 1 46 5 74 4 32 6 88 8 19 9 48 7 36 3 79
0 76 1 69 3 76 5 51 2 85 9 11 6 40 7 89 4 26 8 74
1 85 0 13 2 61 6 7 8 64 9 76 5 47 3 52 4 90 7 45

The results obtained for 10 iterations using each algorithm are as follows:

Instance	Best known solution	Bfoa	Pso
Abz6	1043	1293	1277
MT10	1051	1150	1149
ORB10	944	1583	1316

Table: 1 Solutions for JSSP instances using BFOA and PSO

IV. Conclusion

The effect of number of iterations on the make span of ORB10 for each algorithm are tabulated below:

Instance	Number of iterations	BFOA	PSO
ABZ6	10	1283	1277
ABZ6	25	1295	1468
ABZ6	100	1221	1324
ABZ6	200	1321	1221

Table: 2 Effect of number of iterations on makespan for ORB10 using BFOA and PSO

The convergence of the solution with number of iterations for PSO is as follows:

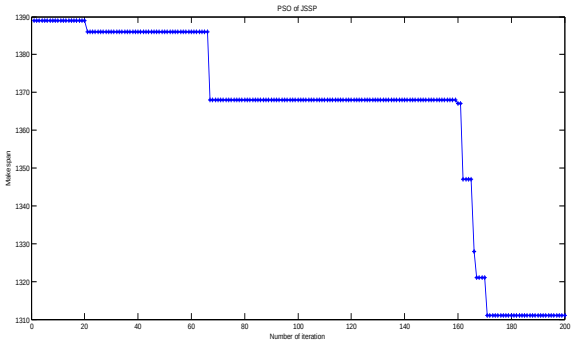

Fig 1: convergence of make span using PSO

The convergence of the solution with number of iterations for BFOA is as follows

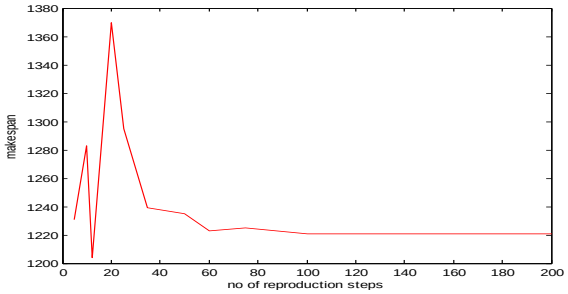

Fig 2: convergence of make span using BFOA

V. Conclusions:

- A new swarm intelligence technique Bacterial Foraging Optimization Algorithm has been applied to job shop scheduling problems of size n*m and also on some bench marking instances. The obtained results have been compared with the best known solutions and another swarm intelligence technique, Particle Swarm Optimization Algorithm.
- Quick convergence is observed and the results are almost instantaneous, thus proving the method to be a suitable method for solving real time JSSP, which are otherwise difficult to solve by conventional computational methods.
- The algorithm can be further improved to obtain solutions nearly equal to the best known solutions by fine tuning.

References

[1]. drum.lib.umd.edu/bitstream/1903/7488/4/25813_cov.pdf
[2]. ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/.../**jain**98**stateart**.pdf
[3]. Kevin M. Passino, International Journal of Swarm Intelligence research, 1(1),1-16, January-March,2010
[4]. International Journal of Computer Applications (0975 – 8887) ,Volume 72– No.5, May 2013,A Study of Bacterial Foraging Optimization Algorithm and its Applications to Solve Simultaneous Equations: Gautam Mahapatra, Soumya Banerjee
[5]. VOL.8,NO.2,FEBRUARY2013, ISSN 1819-6608, ARPN Journal of Engineering and Applied Sciences, Enhanced bacterial foraging algorithm for permutation flow shop scheduling problems: Shivakumar B. L.and Amudha.T
[6]. rajanand.blog.com/files/2010/10/**Bacterial-Foraging-for-Engineering**.pdf
[7]. International Journal of Programming Languages and Applications (IJPLA) Vol.2, No.4, October 2012, DOI: 10.5121/ijpla.2012.2401 1 A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems: Narendhar. S and Amudha. T
[8]. International Journal of Ambient Systems and Applications (IJASA) Vol.1, No.2, June2013, DOI : 10.5121/ijasa.2013.1203 An Enhanced Bio-Stimulated Methodology to Resolve Shop Scheduling Problems: Divya. P, Narendhar. S and Ravibabu. V
[9]. National Conference on Future Aspects of Artificial intelligence in Industrial Automation (NCFAAIIA 2012) Proceedings published by International Journal of Computer Applications ® (IJCA) A Review of Bacterial Foraging Optimization and Its Applications :Vipul Sharma S.S., Pattnaik Tanuj Garg
[10]. Global Journal of Computer Science and Technology Hardware & Computation Volume 12 Issue 10 Version 1.0 A Hybrid Bacterial Swarming Methodology for Job Shop Scheduling Environment By Shivakumar B L & Amudha T
[11]. IEEE Transactions On Power Systems,Vol.24,No.2,May2009 Maiden Application of Bacterial Foraging-Based Optimization Technique in Multi-area Automatic Generation Control: Janardan Nanda, Fellow, IEEE, S.Mishra, Senior Member, IEEE, and Lalit Chandra Saikia
[12]. 978-1-4244-5612-3/09/ c 2009 IEEE, Hybrid Bacterial Foraging with Parameter free PSO K. M. Bakwad, S.S. Pattnaik, B. S. Sohi, S. Devi1, B.K. Panigrahi, Sanjoy Das, M. R. Lohokare
[13]. Swarm-Based Metaheuristic Algorithms and No-Free-Lunch Theorems, Xin-She Yang National Physical Laboratory United Kingdom
[14]. http://www.eii.uva.es/elena/Elena