International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, September 2015
www.ijates.com
ijates
ISSN 2348 - 7550

IMPLEMENTATION OF BACTERIAL FORAGING OPTIMIZATION ALGORITHM IN LEAF SPRING CUTTING STOCK PROBLEM

P.Murugan[1], R.Karthikeyan[2], K.Pandiaraj[3]

Department of Mechanical Engineering, P.A.C.RamasamyRaja Polytechnic College,
K.R.Nagar (Post), Rajapalayam (India)

ABSTRACT

Although the latest enterprise resource planning software provides efficient scheduling of production processes, they have less impact on material marking process, since they focus on the utilization of materials and machines with respect to time. The classical optimization procedures are found to be inefficient in finding the optimal material usage, when the real world complex problems become exponential. Recent publications of many researchers prove the successful implementation of the bio inspired optimization algorithms in such domains. In this article, the chief mechanism of bacterial foraging optimization algorithm and their search parameters are exploited to provide optimal solutions for the cutting stock problem faced by the leaf spring assembly manufacturing industries. Minimization of the trim loss during the cutting of twelve leaves in batches is considered for optimization in order to maximize the material utilization. It is found that the optimal values evolved out of the BFOA are superior when compared with Tabu Search Method. The search characteristics of the E.coli bacteria under consideration is found to be excel in producing the optimal results closer to null values for trim loss. This work also highlights the easy implementation and adoptability of the developed optimization procedure to similar kind of domains.

Keywords: Bacterial Foraging Optimization Algorithm, Leaf Spring Assembly, Tabu Search Method, and Trim Loss

I. INTRODUCTION

Marker planning is a procedure with an objective of determining optimal patterns and how many times each pattern is used. It is applicable for several industries like wood, glass, steel and paper industries. One dimensional marker planning are non-deterministic polynomial complete type problems of complex nature. There is no general procedure available to solve these types of Cutting Stock Problem (CSP) instances in a polynomial time. The use of exact methods is limited to the problems of very small size. Use of approximation methods, evolutionary methods and the meta-heuristics play a vital role in solving these optimization problems [1]. These problems are having dimensionality and are a kind of assignment with an assortment of large objects and small objects. The solution methods are item oriented and pattern oriented approaches. Recently many publications can be seen for solving the optimization of several industrial processes [2].

Among the non-traditional optimization algorithms the bacterial foraging optimization algorithm (BFOA) has been proved to be one of the powerful competitors. BFOA has been shown to be successful for the optimization of a wide range of continuous functions and has huge applications on discrete and combinatorial optimization domains [3]. In this article, it is tried to develop a basic BFOA structure to produce optimal results in the CSP domains. Another trajectory based meta-heuristic known as Tabu Search Method (TSM) is also found to be very effective in optimizing cutting stock problems. TSM has vast applications in almost all the fields of engineering. The flexible memory of TSM exhibits the adaptability and suitability of this algorithm for these types of minimization problems [4]. The main objective of the work is to minimize the trim loss and stock usage in the manufacturing of leaf spring assembly. A procedure based on the BFOA principles has been proposed to achieve this goal. This article explains the leaf spring specifications along with the constraints in the real time manufacturing. A mathematical model has been developed to solve this problem and same has been evaluated using the developed optimization procedures. Finally the results obtained out of this BFOA based procedure is compared with the results of TSM based procedure.

II. EARLIER RESEARCH WORK

BFOA is inspired by the social foraging behavior of Escherichia coli living in the human intestine. BFOA has already drawn the attention of researchers because of its efficiency in solving real-world optimization problems arising in several application domains. The underlying biology behind the foraging strategy of E.coli is emulated in an extraordinary manner and used as a simple optimization algorithm [5]. Senthilkumar M, et.al. had explained that Bacterial Foraging Algorithm is based on a computational intelligence technique which is not largely affected by the size and non-linearity of the problem and has converged to the optimal solution to many problems [6]. Riya M T., reviewed that Bacterial Foraging Optimization has been broadly accepted as a global optimization algorithm for distributed optimization and control. This paper presents a broad overview on the formalization of works contributed by Bacterial Foraging Optimization Algorithm to the field of grid scheduling [7]. Soft computing methods such as BFOA have proved their excellence in giving better results by improving the steady state characteristics and performance indices for the design of electrical drive controller [8]. Germ intelligence based bacterial foraging optimization algorithm has attracted a lot of attention as a high performance optimizer because of its faster convergence and global search approach. Since its inception in 2001, many variants of BFOA have come up leading to even faster convergence with higher accuracy [9].

III. PROBLEM ENVIRONMENT

Leaf springs are widely used for automobile and rail road suspensions. Its assembly consists of a series of flat plates, usually of semi-elliptical shape. The leaves are held together by means of two U-bolts and a centre clip. Rebound clips are provided to keep the leaves in alignment and prevent lateral shifting of the plates during the operation. The longest leaf, called the master leaf is bent at both ends to from the spring eye. At the centre, the spring is fixed to the axle of the vehicle. The multi-leaf springs are provided with one or more extra full length leaves in addition to the master leaf. These extra full length leaves are stacked between the master leaf and the graduated length leaves. The extra full length leaves are provided to support the transverse shear force. The simple construction of the leaf spring assembly is shown in the Fig.1.

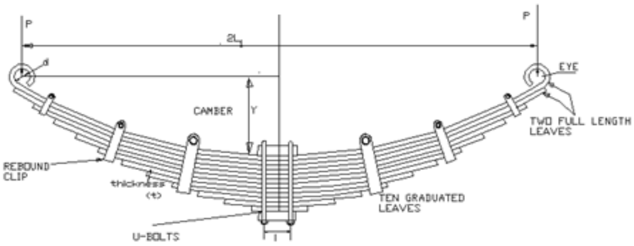

Fig.1. Leaf Spring Assembly

Each leaf has its own specifications and unique processing sequence. The cross section, length of each leaf and the quantity required in the assembly for a bench mark problem is given in the Table 1.

Table1. Leaf Spring Specifications

Part No.	PDF 4510810			
Leaf No.	Cross Section 'mm'		Cutting Length 'mm'	Quantity Required
	Width	Thickness		
1	76.2	13.5	1721	100
2	76.2	13.5	1780	100
3	76.2	13.5	1200	100
4	76.2	13.5	1070	100
5	76.2	13.5	936	100
6	76.2	12.7	792	100
7	76.2	12.7	672	100
8	76.2	12.7	552	100
9	76.2	12.7	418	100
10	76.2	11.11	276	100
11	76.2	11.11	180	100
BPP	76.2	11.11	140	100
TPP1	76.2	12.7	140	100
TPP2	76.2	12.7	140	100

Where, BPP – back pressure plates and TPP – tension pressure plates.

The cross section and length of each and every leaf varies from each other. The stock length is given as a standard value of 5500 mm. It becomes very difficult to cut the stock length to any one of the leaf so that the trim loss is less. Hence a combination of leafs can be cut in the stock length to have a minimum trim loss. Here, leafs of same cross section can be grouped to reduce the material wastage during the marker planning. A batch size of 100 units of leaf spring assembly is considered for this problem.

IV. OBJECTIVE FUNCTION

The main objective of the work is to minimize the trim loss during the marker planning of raw materials in the manufacturing of leaf spring assembly and is given by:

Minimize,

$$: f(x) = 1 - \left\{ \left[\sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{a} (x(i,j,k)) \right] \div \left[\sum_{k=1}^{s} L(k) \right] \right\}$$

-- (1)

Subjected to constraints,

(i) $L_{max} = 5500$

(ii) $t \langle l_j$; where j = 1 to n

(iii) $\mathrm{mod}|a| = 0$

Where, s = number of stocks available; n = number of leaves in the assembly; a = number of assembly sets required; L = length of stock; L_{max} = maximum stock length; t = trim loss; l_j = length of leaf and i, j & k =iteration counts

V. BFOA

Bacterial Foraging Optimization Algorithm (BFOA) proposed by Kevin Passino [10] is a biological inspired algorithm. BFOA mimics the behaviour of bacteria adopted (i) to find the nutrient food in an effective manner and (ii) the way of sending signals to other members in the particular swarm for an effective communication. BFOA proved its consistency in group foraging strategy based optimization applications. Especially the characteristics of E.Coli bacterial were found to be superior in finding solutions for he multi objective optimal functions. These kinds of bacteria searches for nutrients in an manner to maximize the energy obtained per unit time. The individual bacterium communicates with other members of the population by means of signals. These two factors are considered to be very important for the bacteria to take decisions about foraging. The process of bacterium movement in small steps is search of nutrients is called as chemotaxis. This biological movement of bacterium is conceptualized and implemented to find the optimal solution from the problem search space.

Foraging is a phenomenon of swarms to find nutrient sources at an optimal time to maximize their energy intake. The process of foraging is different for different species. In general, herbivorus find food easily which are low in energy value and hence they have to be consumed in large quantities. But the carnivorus find it very difficult to get the food, the foods consumed by them are of high energy value and can be consumed in lesser quantities. The availability of high energy nutrients with in a shorter range, the quantity and quality of nutrient available, the risk of survival, and the physiological characteristics of the forager may be considered as the important parameters for the successful implementation of the BFOA.

5.1 BFOA Methodology

The BFOA theory can be easily understand by the four steps adopted by the bacteria, i.e., Chemotaxis, Swarming, Reproduction, and Elimination-Dispersal.

ijates
ISSN 2348 - 7550

5.1.1, Chemotaxis

The movement of E.Coli bacteria in small steps for the search of high nutrient is done in two ways, i.e. swimming and Tumbling. In simple, swimming is a small movement of bacteria in a predefined direction and tumbling can be said as the movement of bacteria in a random direction.

5.1.1.1, Swimming

The movement of the bacteria for a period of time in the same direction is known as swimming i.e. the rotation of the bacteria will be in the clockwise direction. The process can be mathematically expressed as follows.

Assume, suppose $x^i(j,k,l)$ represents i^{th} bacterium at the j^{th} chemotactic, k^{th} reproductive, and l^{th} elimination-dispersal step, then the swimming of the bacterium is,

$$: x^i(j+1,k,l) = x^i(j,k,l) + \Delta x(i)$$

-- (2)

Where $\Delta x(i)$ – is the size of swimming step taken by the i^{th} bacterium per unit time, i.e. run length unit.

5.1.1.2, Tumbling

Here, the movement of i^{th} bacterium is defined by the size of the step taken in a random direction, i.e. the bacteria is assumed to be rotating in the counter clock wise direction. The tumbling process can be represented as,

$$: x^i(j+1,k,l) = x^i(j,k,l) + \Delta x(i) \times \frac{1}{2}\log\left(\Delta x^T(i) \cdot \Delta x(i)\right)$$

-- (3)

Where the $\Delta x(i)$ – indicates a secure vector in the random direction between the limits [-1,1].

5.1.2, Swarming

E.coli bacteria are found to form spatio-temporal patterned swarms in the semi-solid mediums. These swarms form as a travelling ring which moves in concentric patterns to have high bacterial density. The formation of swarms is continued until they reach richest high food resource locations. This movement rapidly converges the search nature by the attractive force applied by the forager on the neighbouring bacteria in the swarm. In order to frame a mathematical model a penalty function added to the fitness function. The penalty function is calculated based upon the relative distance of each bacterium from the fittest bacterium during that search process. This penalty function equals zero when all the bacteria in the swarm amalgamate into that solution point. At the end of swarming process, the resulted swarms will emerge as the solution point with high bacterial density.

The cell-to-cell signaling in the swarms can be represented as:

$$: J_{cc}(\Delta x, P(j,k,l)) = \sum_{i=i}^{S} J_{cc}(\Delta x, \Delta x^i(j,k,l))$$

-- (4)

$$: \left[= \sum_{i=i}^{S} [-D_{attract} \exp(-W_{attract} \sum_{m=1}^{p} (\Delta x_m - \Delta x_m^i)^2)] + \sum_{i=i}^{S} [H_{repellent} \exp(-W_{repellent} \sum_{m=1}^{p} (\Delta x_m - \Delta x_m^i)^2)] \right]$$

-- (5)

Where $J_{cc}(\Delta x, P(j,k,l))$ is the penalty function added to the actual objective function to present a time varying objective function.

5.1.3, Reproduction

According to the law of survival to the fittest, the least healthy bacteria survived after some chemotaxis stages will die. While the healthier bacteria asexually split into two groups which will replace the eliminated weaker bacteria to maintain a constant swarm size during the evolution process. The main purpose of this reproduction process is to replace the least healthy bacteria having poorer foraging ability with the healthier half to make the population size of the bacteria as a constant one.

For the given k, and l, and for each i = 1,2,…S, the health bacterium "i" is given by,

$$: J^{i}_{health} = \sum_{j=i}^{N_c+1} J(i, j, k, l)$$

-- (6)

Bacteria and chemotactic parameters are sorted in an ascending order for the selection process.

The S_r bacteria with the highest J_{health} value dies and the remaining S_r bacteria with the best values split.

5.1.4, Elimination-Dispersal

During the evolution process, gradual or sudden changes occurred in the local environment may cause the elimination of a set of bacteria, i.e. all the bacteria in the particular region may be either killed or may be dispersed to a new environment. Over the period of time, this unpredictable event may place a newer set of bacteria nearer to nutrients having high energy value. To implement this behaviour of bacteria in the algorithm, some bacteria are randomly selected and liquidated with a small probability while the remaining replacements are randomly initialized over the search space. This step helps in reducing the behaviour of stagnation and improves the long distance motile behaviour of selected bacteria at the particular population level. Though this elimination-dispersal event seems to destroy the improvement happened in the previous stages, actually there are possibilities of placing the reinitialized bacteria nearer to the nutrients having high energy values.

The control parameters of the BFOA along with their experimental values are shown in the Table 2.

Table 2. Control Parameters of BFOA and their Values

Parameters	Variable	Trial Values	Parameters	Variable	Trial Values
No. of bacterium	S	50	Size of the step	$\Delta x(i)$	0.1
Maximum No. of steps	N_s	4	BFOA Coefficient	$W_{attract}$	0.04
No. of Chemotactic steps	N_c	100	BFOA Coefficient	$D_{attract}$	0.01
No. of Reproduction steps	N_{re}	4	BFOA Coefficient	$H_{repellent}$	0.01
No. of Elimination-Dispersal steps	N_{ed}	2	BFOA Coefficient	$W_{repellent}$	10
Elimination-Dispersal Probability	P_{ed}	0.25			

5.2 Pseudo Code of BFOA

The general principles adopted for the BFOA based optimization procedure is given below.

{

Generate initial population;

Assignment operations;

Do Elimination-Dispersal ()

```
    Do Reproduction ()
        Do Chemotaxis ()
        Swarming ()
            Tumble;
            Swim ()
                Evaluation and selection;
            Next Swim;
        Next Swarm;
        Next Chemotaxis;
    Next Reproduction;
Next Elimination-Dispersal;
Stop;
}
```

VI. TABU SEARCH METHOD

Tabu search method (TSM) has its antecedents in methods designed to cross boundaries of feasibility or local optimality standardly treated as barriers, and to systematically impose and release constraints to permit exploration of otherwise forbidden regions. The philosophy of tabu search is to derive and exploit a collection of principles of intelligent problem solving. A fundamental element underlying in tabu search is the use of flexible memory [11]. From the standpoint of tabu search, flexible memory embodies the dual processes of creating and exploiting structures for taking advantage of history. The memory structures of tabu search operate by reference to four principal dimensions, consisting of recency, frequency, quality, and influence. These dimensions in turn are set against a background of logical structure and connectivity.

TSM operates under the assumption that a neighborhood can be constructed to identify “adjacent solutions” that can be reached from any current solution [12]. Pair wise exchanges are frequently used to define neighborhoods in permutation problems, identifying moves that lead from one solution to the next. Associated with each swap is a move value, which represents the change on the objective function value as a result of the proposed exchange. Move values generally provide a fundamental basis for evaluating the quality of a move.

A chief mechanism for exploiting memory in tabu search is to classify a subset of the moves in a neighborhood as forbidden. The classification depends on the history of the search, particularly as manifested in the recency or frequency that certain move or solution components, called attributes, have participated in generating past solutions. As a basis for preventing the search from repeating swap combinations tried in the recent past, potentially reversing the effects of previous moves by interchanges that might return to previous positions, we will classify as tabu all swaps composed of any of the most recent pairs of such modules [13]. Also, Tabu restrictions are not inviolable under all circumstances. When a tabu move would result in a solution better than any visited so far, its tabu classification may be overridden. A condition that allows such an override to occur is called an aspiration criterion.

In TS, each solution $x \in X$ has an associated set of neighbors $(H, x) \subset X$, called the neighborhood of x. Each solution $x' \varepsilon N(x)$ can be reached directly from x by a move. History determines which solutions may be reached by a move from the current solution, selecting x_next from N (H, x_now). The essence of the method depends on how the history record H is defined and utilized, and on how the candidate neighborhood Candidate_N (x_now) and the evaluation function c (H, x) are determined [14].

6.1 Memory Parameters

6.1.1, Short Term Memory: The move attributes of the solutions recently visited during the search pass are maintained. These solutions are removed from the memory when the corresponding Tabu tenure elapses.

6.1.2, Long Term Memory: A history of elite solutions (attributes) with their corresponding cumulative number of repetition in the whole of search pass (frequency) is maintained.

6.2 Decisive Factors

6.2.1, Choice criterion: The solution with the minimum COF value among the neighboring solutions of the current solution

6.2.2, Aspiration criterion: Tabu restrictions are lifted for the solutions under tabu classification, with the value of the COF, 10 percent or more, less than that of the current solution

6.2.3, Restarting criterion: When the solutions out of 10 consecutive iterations yield poorer and poorer solution, the procedure is repeated with a new initial solution.

6.2.4, Termination criteria: Reaching a predefined minimum value of COF or 1000 numbers of iterations whichever occurs first.

6.3 Methodology

6.3.1, Initialization: An initial set of parameters is selected at random among the feasible set of parameters. The objective function value for the solution is computed and defined as best value. The history record is initialized with empty record.

6.3.2, Evaluation and Updation: A set of all the non-tabu neighborhood solutions is generated and the objective function values are calculated and the one with the minimum value is chosen by choice criterion. The selected value is less than the best value, and then a move is performed with an updation in the history. Else, the best ten elite move attributes from the long term memory are chosen as per restarting criterion to yield a better move value with a re-initialization of history.

6.3.3, Termination: If the termination criterion is not met, step (6.3.2) is repeated otherwise the procedure is stopped.

VII. RESULTS AND DISCUSSIONS

For the designed procedures based on the bacterial foraging optimization algorithm, and tabu search method software is developed to conduct experiments for the benchmark problem. The optimal results obtained by this developed procedures are plotted in a chart as shown in the Fig.2. Fig.3. and Fig.4.

ijates
ISSN 2348 - 7550

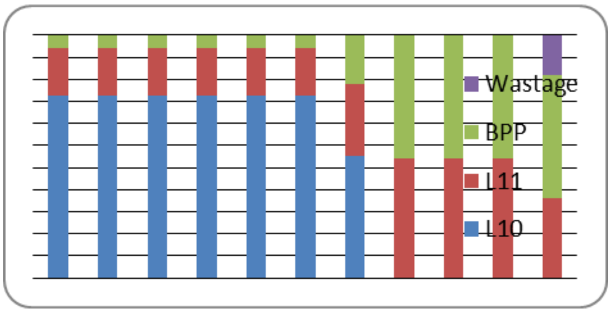

Fig.2. Outcome of the BFOA Based Procedure (For Leaf Thickness 11.11 mm)

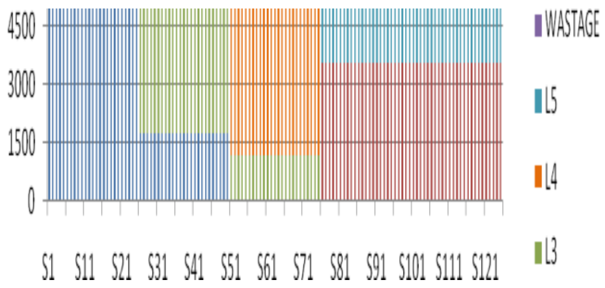

Fig.3. Outcome of the BFOA Based Procedure (For Leaf Thickness 13.50 mm)

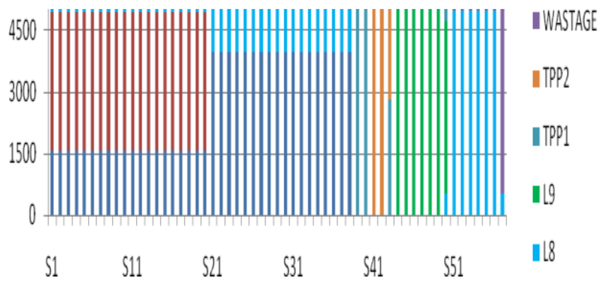

Fig.4. Outcome of the BFOA Based Procedure (For Leaf Thickness 12.70 mm)

From the charts, the total trim loss for the manufacturing of 100 leaf spring assembly can be easily conceptualized. For the 11.11 mm thickness leaves, only 600 mm of the raw material is wasted out of 60500 mm (i.e. 11 full length stocks). For the 13.50 mm thickness leaves, only 16,800 mm of the raw material is wasted out of 6,87,500 mm (i.e. 125 full length stocks). For the 12.70 mm thickness leaves, only 9,208 mm of the raw material is wasted out of 3,13,500 mm (i.e. 57 full length stocks). From the above charts, the effectiveness of BFOA procedure can also be easily evaluated.

Table 3. Results of Mathematical Model

Sl. No.	**Methodology Adopted**	**Objective Function Value**
1	Bacterial Foraging Optimization Algorithm based procedure	0.025349
2	Tabu Search based procedure	0.048350

From the Table 3., it is found that the results of the experiments based on the BFOA procedure produces better solutions then the TSM based procedure.

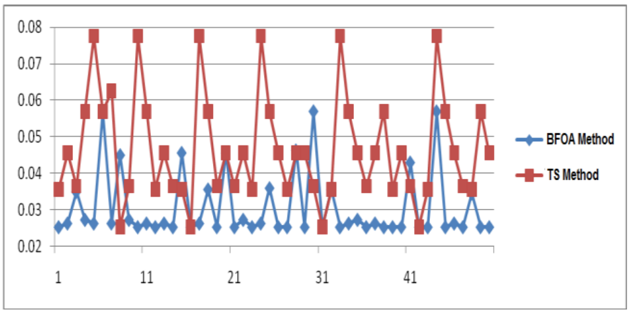

Fig.5. Comparison of BFOA and TSM Procedures (For 50 Trials)

For the defined problem instance, two different approaches BFOA and TSM were applied and the results of the trials conducted using both procedures are explained in the Fig.5. The BFOA based procedure is found to yield better solutions during the exploration of the search space. The BFOA based procedure is also found to maintain consistency in producing better solutions and the deviations of other poor solutions are found to be very small from the better solutions.

VIII. CONCLUSION

The classical optimization techniques are not found to be sufficient to solve the trim loss problems within the time constraints. In this paper, optimization procedures based on the Bacterial Foraging Optimization Algorithm and Tabu Search Method were developed to solve the marker problem existing in the manufacturing of leaf spring assembly. The results out of both the procedures were compared and the BFOA based procedure was found to yield better solutions consistently for the given manufacturing environment. In future, BFOA based procedure can be combines with other meta heuristics to make a better hybrid optimization procedure.

REFERENCES

[1] H.A.Dyckhoff, Typology of cutting and packing problems, European journal of Operations research 44, pp. 145-159, 1990.

[2] M.Gradisar, M.Kljajic and G.Resinovic, A hybrid approach for optimization of one-dimensional cutting, European journal of Operations research 199, pp. 165-174, 1999.

[3] Salim Ali E. and S.M. Abad-Elazin. 2010. Optimal PID Tuning for Load Frequency Control Using Bacteria Foraging Optimization Algorithm. Conference Proceedings (MEPCON'10), Cairo University, Egypt. pp. 19-21.

[4] F.Glover, Tabu Search, Part 2, ORSA Journal on Computing 2, pp.4-32, 1990.

[5] Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham, Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications, Foundations of Computational Intelligence, DOI: 10.1007/978-3-642-01085-9_2, Volume 3, pp.23-55

[6] Senthil Kumar, M. and Renuga,P. Application of Bacterial Foraging Algorithm for Optimal Location of FACTS Devices with Multi-Objective Functions, International Review of Electrical Engineering (IREE), Vol. 6, No. 4, August 2011, pp1905-1915, 2011

[7] Riya Mary Thomas, Survey of Bacterial Foraging Optimization Algorithm, International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-1, Issue-4, March 2013

[8] M. Gowrisankar, and A. Nirmal Kumar, Soft computing based controllers for electric drives - a comparative approach, ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 1, January 2015, ISSN 1819-6608

[9] Vipul Sharma S.S. Pattnaik Tanuj Garg E&C Engg Dept, Gurukul NITTTR, Chandigarh E&C Engg Dept, Gurukul Kangri University,Haridwar, A Review of Bacterial Foraging Optimization and Its Applications, National Conference on Future Aspects of Artificial intelligence in Industrial Automation (NCFAAIIA 2012) Proceedings published by International Journal of Computer Applications® (IJCA)

[10] Kevin M.Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag., Vol. 22, No. 3, pp. 52–67, Jun. 2002.

[11] Fred Glover and Manuel Laguna, Tabu Search-Modern Heuristic Techniques for Combinatorial Problems, pp.70-150, Blackwell Scientific Publications, Oxford, 1993.

[12] Chun-Wei Tsai, Shih-Pang Tseng, Ming-Chao Chiang, Chu-Sing Yang, A Time-Efficient Method for Metaheuristics: Using Tabu Search and Tabu GA as a Case,Hybrid Intelligent Systems, International Conference on, vol. 2, pp. 24-29, 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009.

[13] A. S. Rudnev, Probabilistic tabu search algorithm for the packing circles and rectangles into the strip, Diskretn. Anal. Issled. Oper., 16:4, pp.61–86, Novosibirsk State University, Novosibirsk, Russia, 2009.

[14] Nadjib Aitsaadi, Nadjib Achir, Khaled Boussetta and Guy Pujolle, A Tabu Search WSN Deployment Method for Monitoring Geographically Irregular Distributed Events, Wireless Sensor Technologies and Applications, ISSN 1424-8220, Sensors 2009, 9(3), pp.1625-1643,2009.