Vol. 1, Issue 2, pp 69-75 14,Aug, 2015

eISSN : 2454-1435 © 2015

pISSN : 2454-1443 © 2015

http://www.ijrmmae.in

IJRMMAE
Our Breath is Research ..

International Journal of Research in Mechanical Mechatronics and Automobile Engineering

Innovative Criterions for Job Shop Scheduling Problems
A Predominant Approach

Henok Mekonnen

Department of Mechanical Engineering, Jimma, Jimma Institute of Technology, Ethiopia

Received 10, Aug 2015 | Accepted 21, Aug 2015

1. ABSTRACT

Generally, Job Shop Scheduling Problem (JSSP) is an operational sequencing problem to process *n* jobs on *m* machines in a given sequence so as to optimally utilize the resources by complete processing of all jobs in a minimum possible time. JSSP has belongs to the category of NP hard problems where the search space of the problem is $(n!)^m$. Several naturally inspired evolutionary techniques / methods have recently been developed to address these problems to get near optimal solutions in a reasonable time period thus several unsolved / difficulty to solve JSSPs became target for many researchers.

Some Traditional Optimization Techniques includes Priority Dispatch Rules, Efficient Methods solvable in polynomial times. Enumerative methods and Mathematical formulation like Linear Programming, Mixed Linear Programming, Lagrangian Relaxation, Branch and Bound and Disjunctive Graph techniques etc. the research has been accelerated with applications of Nontraditional Techniques. Application of Artificial Intelligence, insertion Algorithms, Bottleneck Heuristics, Neural Networks, Expert Systems and Local Search algorithms. Application of Evolutionary Algorithms for scheduling found from mid 80s to till date. Genetic Algorithm, Variation in GAs, PSO, ACO, BCO, Memetic Algorithm, Immune Algorithm and several Hybrid Algorithms. The BFO algorithms, HS algorithms, IWO have been found applied in JSSP in the recent years.

More focus on the application of Non Traditional methods for JSSP is increasing compared to traditional techniques. For the same cost and time non traditional methods yield better solutions compared to traditional methods.

Hence, in this paper classical evolutionary algorithms namely Invasive weed optimization (IWO), Bacterial Foraging Optimization (BFO) and Music Based Harmony Search principles and fine-tuned the mechanisms to model and solve JSSP. Several Bench Mark instances available in OR library were thoroughly tested to prove the efficiency of the proposed methods by selective and random generation of populations.

Two different types of population generations were considered

i) Random Population (RP): The initial population is randomly generated and applied to the algorithm procedures and let us call such methods as PSO with RP, HSPO with RP, AIA with RP, HAIA with RP, MBHS with RP,
IMBHS with RP, BFO and IWO with RP.

ii) Selective Population: The initial Populations are generated using priority dispatching rules. A priority dispatching rule is a simple mathematical formula that, based on some processing parameters, specifies the priority of operations to be executed. 10 initial schedules i.e., populations are generated using 10 commonly used priority dispatching rules given in the Table 1 let us call these methods as PSO with SP, HSPO with SP, AIA with SP, HAIA with SP, MBHS with SP, IMBHS with SP, BFO with SP and IWO with SP.

EXPRESSION	DESCRIPTION
Shortest Processing Time (SPT)	The job with shortest time on machines selected. pi≤ pi+1≤ pi+2≤…….≤ pn
Longest Processing Time(LPT)	The job with longest time on machines selected. pi≥ pi+1≥ pi+2≥…….≥ pn
Minimum Slack Time Per Operation(MINSOP)	Time remaining until the due date – Processing time Remaining
Minimum Due Date(MINDD)	The job with earliest due date is processed first Di≤ Di+1≤ Di+2≤…….≤ Dn
Critical Ratio(CR)	Remaining due date/Remaining processing time
Most work remaining (MWKR)	Select the operation associated with the job of the most work remaining to be processed
Least work remaining(LWKR)	Select the operation associated with the job of the least work remaining to be processed
Shortest remaining Minimum Processing Time(SRMPT)	Min(processing tine remaining- minimum processing time)
Longest remaining Maximum Processing Time(LRMPT)	Max(processing tine remaining- maximum processing time)
RANDOM(random selection)	Select the next job to be processed randomly.

Table.1: List of common Priority Dispatching rules used to generate the initial population.

The coding of these algorithms is done in MATLAB, optimized by speed, and run on Intel Core2Duo T6400 @ 2.00GHz and each algorithm was made to run 30 times on each problem of 250 Bench Mark Problem Instances. In this chapter the results obtained by executing the above algorithm on 250 Bench Marking Problems are reported only.

2. BENCHMARKPROBLEMS

To find the comparative merits of the various techniques and algorithms, one needs to test on the bench mark problems. These benchmark problems are formulated by various authors, **Fisher and Thompson**, 1963 **(FT)** 3 problems of 3 different sizes: 6×6, 10×10, 20×5; **Lawrence**, 1984 **(LA)** 40 problems of 8 different sizes: 10×5, 15×5, 20×5, 10×10, 15×10, 20×10, 30×10 and 15×15; **Adams Balas & Zawak**, 1988 **(ABZ)** 5 problems of 2 different sizes: 10×10, 20×15; **Applegate and Cook**, 1991 **(ORB)** 10 problems of 10×10 size; **Storer, Vaccari & Wu**, 1992 **(SWV)** 20 problems of 3 different sizes: 20×10 , 20×15, 50×10; **Yamada and Nakano**, 1992 **(YN)** 4 problems of 20×20 size; **Taillard**, 1993 **(TA)** 80 problems of 8 different sizes: 15×15, 20×15, 20×20, 30×15, 30×20, 50×15, 50×20; **Demirkol, Mehta & Uzsoy**, 1998 **(DMU)** 80 problems of 8 different sizes: 20×15, 20×20, 30×15, 30×20, 50×15, 50×20, 100×20; **Jacques Carlier** , **(CAR)** 8 problems of 8 different sizes: 11×5, 13×4, 12×5, 14×4, 10×6, 8×9, 7×7, 8×8; and are freely available. In this paper all the known 250 benchmark problems of JSSP have been taken into consideration for testing with the algorithms being developed.

3. ALGORITHMS DEVELOPED

3.1. Particle Swarm optimization (PSO) Algorithm

One of the best evolutionary techniques for unconstrained continuous optimization is particle swarm optimization (PSO) proposed by Kennedy and Eberhart (1995), inspired by social behavior of bird flocking or fish schooling. PSO has been successfully used in different fields due to its ease of implementation and computational efficiency. Particles move toward the ***pbest*** position and ***gbest*** position with each iteration. The ***pbest*** position is the best position found by each particle so far. The ***gbest*** position is the best position found by the swarm so far. The particle moves itself according to its velocity. For each particle k and dimension j, the velocity and position of particles can be updated by the following equations:

$$v_{kj} = w \times v_{kj} + c_1 \times rand_1 \times (pbest_{kj} - x_{kj}) + c_2 \times rand_2 \times (gbest_j - x_{kj}) \quad \rightarrow (3.1)$$

$$x_{kj} = x_{kj} + v_{kj} \quad \rightarrow (3.2)$$

In Equations (3.1) and (3.2), is the velocity of the particle k on dimension j, and is the position of particle *k* on dimension *j*. The is the ***pbest*** position of particle *k* on dimension *j*, and is the ***gbest*** position of the swarm on dimension *j*. The inertia weight *w* is used to control exploration and exploitation. The particles maintain high velocities with a larger *w*, and low velocities with a smaller *w*. The constants and are used to decide whether particles prefer moving toward a ***pbest*** position or ***gbest*** position. The and are random variables between 0 and 1. The process of working of PSO and its flow chart is shown in Figure.1.

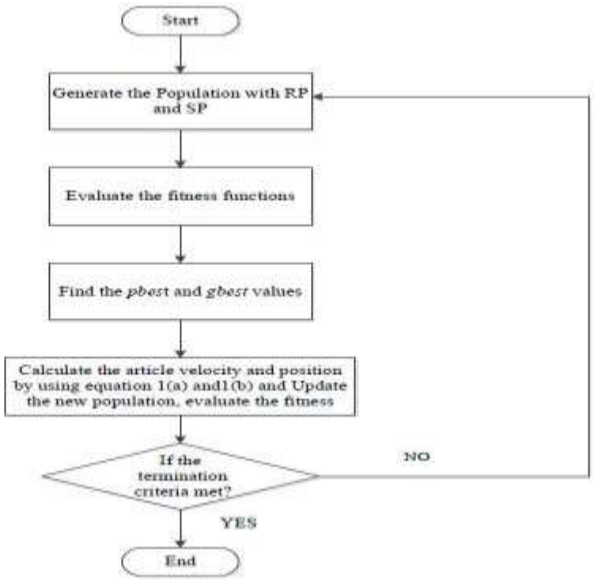

Figure.1 Flowchart for Particle Swarm Optimization

3.2. Hybrid Particle Swarm Optimization (HPSO) Algorithm

PSO is a stochastic search algorithm; it is prone to inadequate global search-ability at the end of a run. PSO may fail to find the required optima in cases when the problem to be solved is too complicated and complex. The original PSO was designed for a continuous solution space. Simulated Annealing (SA) has certain probability to avoid becoming trapped in a local optimum and the search process can be controlled by the cooling schedule. By reasonably combining these two different search algorithms, we develop a general, fast and easily implemented hybrid optimization algorithm called HPSO, it can be seen that PSO provides initial solution for SA during the hybrid search process. Moreover, such HPSO can be applied to many combinatorial optimization problems by simple modification. The psedudo code of HPSO is given on Figure.2.

3.3. Artificial Immune Algorithm (AIA)

Artificial Immune Systems are adaptive systems applied to problem solving which have been derived from the principles, models and functions of the human immune system. The field of AIS was initially developed in 1986 by J. D. Farmer et al. in their paper called 'The Immune System, Adaptation and Machine Learning' and was followed by another paper by G.W. Hoffman called 'A Neural Network Model Based on the Analogy with the Immune System'. Artificial Immune Model is, a basic model with understanding of the Functioning of the human immune system is essential. The human immune system is characterized by its adaptive and robust nature.

This can be observed by considering a simple example of an infection attacking the body. The infection (or antigen) attacking the body is countered by the defence mechanism called the antibody. The artificial immune system was built on the following two principles of the immune system.

i. Clonal selection principle

ii. Affinity maturation principle

i. Clonal selection principle

Each schedule (antibody) has a makespan value that refers to the affinity value of that antibody. Affinity value of each schedule is calculated from the affinity function. The affinity function is defined as

$$\text{Affinity (p)} = 1/\text{makespan}$$

From this relation, a lower makespan value gives a higher affinity value. Further the cloning of antibodies is done directly proportional to their affinity function values. Therefore, there will be more clones of antibodies that have lower makespan values than those with higher makespan values in the new generated clone population. An affinity function is defined based on makespan values of the schedules. Also they have given a function to calculate the number of clones that would be proliferated.

ii. Affinity maturation principle

The affinity maturation principle consists of a method namely mutation.

Mutation: A two phased mutation procedure were used for the generated clones.

a. Inverse mutation
b. Pair wise interchange mutation

a. Inverse mutation: For a sequence s, let i and j be randomly selected two positions in the sequences. A neighbour of s is obtained by inversing the sequence of jobs between i and j positions. If the makespan value of the mutated sequence (after inverse mutation) is smaller than that of the original sequence (a generated clone from an antibody), then the mutated one is stored in the place of the original one. Otherwise, the sequence will be mutated again with random pair wise interchange mutation.

b. Pair wise interchange mutation: Given a sequence s, let i and j be randomly selected two positions in the sequence s. A neighbour of s is obtained by interchanging the jobs in positions i and j. If the makespan value of the mutated sequence (after pair wise interchange mutation) is smaller than that of the original sequence, then store the mutated one in the place of the original one. In the case where the algorithm could not find a better sequence after the two-mutation procedure, then it stores the original sequence (generated clone).

Figure.3 is the flowchart and gives the pseudo code of AIS algorithm for solving the job shop scheduling problem. The possible schedules are represented by integer-valued sequences of length n (jobs). The *n* elements of the strings are the jobs which will be sequenced. Therefore, the strings are composed of permutations of *n* (jobs) elements. Those strings are accepted as antibodies of the AIS. The algorithm goes up to solution by the evolution of these antibodies.

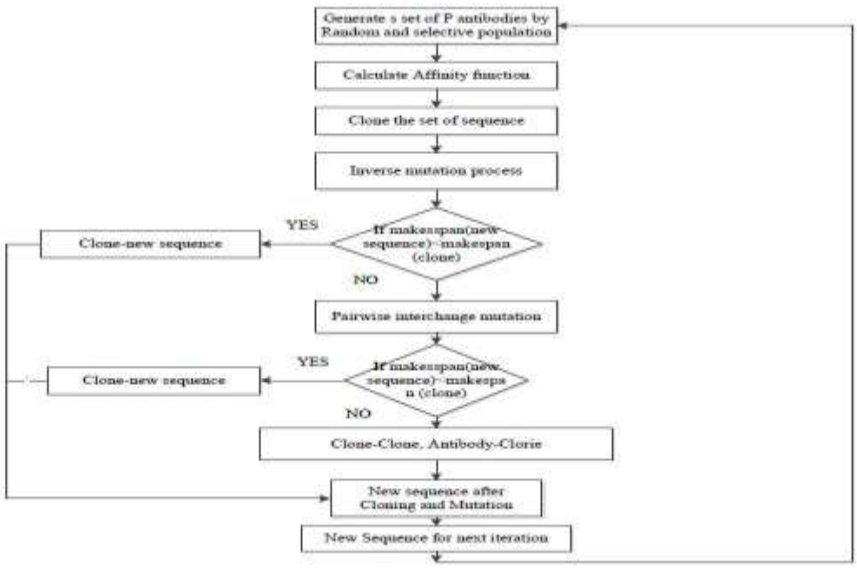

Figure. 3. Flow chart of artificial immune system algorithm

3.4. Hybrid Artificial Immune Algorithm (HAIA)

The hybrid algorithm based on AIS theory and PSO is described for the JSSP. Hybrid Artificial Immune Algorithm adopts the antigens for finding optimum solutions efficiently. In this case antigen is a potential solution and the algorithm helps the antigens to evolve and generate better population thus giving rise to fitter antigens which represent competitive schedules.

To implement a basic artificial immune system, four decisions have to be made: encoding, similarity measure, selection and mutation. Once an encoding has been fixed and a suitable similarity measure is chosen, the algorithm will then perform selection and mutation, both based on the similarity measure, until stopping criteria are met. The pseudo code showing the main procedures of the algorithm is shown in Figure.4 To accelerate the convergence speed of the search algorithm, a neighbourhood search mechanism is formulated especially for this problem

3.5. Invasive Weed Optimization (IWO) Algorithm

Invasive weed optimization (IWO), first designed and developed by Mehrabian and Lucas (2006), is a relatively novel numerical stochastic optimization algorithm inspired from colonization of invasive weeds. The algorithm is simple but has shown to be effective in converging to optimal solution by employing basic properties, e.g. seeding, growth and competition, in a weed colony. A weed is any plant growing where it is not wanted; any tree, vine, shrub or herb may qualify as a weed, in any specified geographical area, depending on the situation. Weeds have shown a very robust and adaptive nature that renders them undesirable plants in agriculture.

In D-dimensional search space, a weed which represents a potential solution of the objective function is represented by W = (w1, w2,⋯ , wm) . Firstly, M weeds, called a population of plants, are initialized with random growth position, and then each weed produces seeds depending on its fitness and the colony's lowest fitness and highest fitness to simulate the natural survival of the fittest process. The number of seeds each plant produce increases linearly from minimum possible seed production to its maximum. The generated seeds are being distribution randomly in the search area by normal distribution with mean equal to zero and a variance parameter decreasing over the number of iterations. By setting the mean equal to zero, the seeds are distributed randomly such that they locate near to the parent plant and by decreasing the variance over time, the fitter plants are grouped together and inappropriate plants are eliminated over times.

3.6. Bacterial Foraging Optimization (BFO) Algorithm

Bacterial Foraging Optimization (BFO) was first introduced by Passino inspired from the Swarm Intelligence. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space. During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counter clockwise direction helps the bacterium to swim at a very fast rate. In the above- mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment.

Let us define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination-dispersal event.

Also let $P(j, k, l) \{ (j, k, l) \mid i$ 1,2,..., $S\}$ i = q = represent the position of each member in the population of the S bacteria at the j-th chemotactic step, k-th reproduction step, and l-th elimination-dispersal event. Here, let $J(i, j, k, l)$ denote the cost at the location of the i-th bacterium i. Note that we will interchangeably refer to J as being a "cost" (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, S can be very large (e.g., S =109), but p = 3. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows $p > 3$ so that we can apply the method to higher dimensional optimization problems. The whole process is depicted in Figures.6.

REFERENCES

1. Bagheri, M. Zandieh, I. Mahdavi, M. Yazdani, An artificial immune algorithm for the flexible job - shop scheduling problem Future Generation Computer Systems, In Press, Accepted Manuscript, Available online 15 October 2009.
2. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.
3. El-Bouri, N. Azizi, S. Zolfaghari, A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: The case of job shop scheduling European Journal of Operational Research, Volume 177, Issue 3, 16 March 2007, Pages 1894-1910.
4. R. Mallahzadeh,H. Oraizi and Z. Davoodi-Rad "Application of Invasive Weed Optimization technique for antenna configuration" Progress in Electro-magnetic Research, PIER 79, 2008Pages 137-150
5. A.S Jain, S.Meeran Mascis, Dario Pacciarelli "DeterministicJob-shop scheduling : Past , Present and Future" European Journal of Operational Research, Volume 113, 1999, Pages 390-434
6. Adil Baykasoğlu, Lale Özbakır, Türkay Dereli "Multiple Dispatching Rule based heuristic for Multi-objective Scheduling of Job Shops using Tabu Search" 5th International Conference on Managing Innovations in Manufacturing, September 2002
7. Anniruddha Basak, Siddharth Pal, Swagatam Das "A Modified Invasive Weed Optimization for Time Modulated Linear antenna Array Synthesis " istance- and Connectivity-based Node Localization in Wireless Sensor Networks", Engineering Applications of Artificial Intelligence, in press, June 2012.
8. Blazewickz J (1996) ,"The job shop scheduling problem: Conventional and new solutions techniques", Eur J Oper Res pp 931–30.
9. Blazewickz J (1996) ,"The job shop scheduling problem: Conventional and new solutions techniques", Eur J Oper Res pp 931–30.
10. Bou Shaala, A., Shouman, M.A., and Esheem.S, "Some Heuristic Rules for Job Shop Scheduling Problem" Industrial and manufacturing systems engineering, Garyounis University, Libya.
11. Carlier J, Pison E (1989) ,"An algorithm for solving the job shop problem", Manage Sci 35:164–176
12. Castro L. de and Timmis J., 2003. "Artificial Immune Systems as a Novel Soft Computing Paradigm". Soft
    Computing Journal, vol. 7, Issue 7.
13. D.Y. Sha, Cheng-Yu Hsu, A hybrid particle swarm optimization for job shop scheduling problem Computers Industrial Engineering, Volume 51, Issue 4, December 2006, Pages 791-808.
14. Dam Scheduling: Geem, Z. W. "Optimal Scheduling of Multiple Dam System Using Harmony Search Algorithm", Lecture Notes in Computer Science, 2007.
15. Dasgupta, D., (1999), 'An Overview of Artificial Immune Systems and Their Applications', In Artificial Immune Systems and Their Applications, D. Dasgupta (ed.), Springer-Verlag, pp. 3 – 21.
16. Ferdinando Pezzella, Emanuela Merelli "A tabu search method guided by shifting bottleneck for the job shop scheduling problem" European Journal of Operational Research, Volume 120, Issue 2, 16 January 2000, Pages 297-310.
17. Heat exchanger design: Fesanghary, M., Damangir, E. and Soleimani, I. "Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search", Applied Thermal Engineering, In press.
18. Helena Ramalhinho Lourenço "Job-shop scheduling: Computational study of local search and large-step optimization methods" European Journal of Operational Research, Volume 83, Issue 2, 8 June 1995, Pages 347-364.
19. K. Lee and Z. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", Computer Methods in Applied Mechanics and Engineering 194, pp.3902-3933,2005.
20. K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," IEEE Control Systems Magazine, vol. 22, no. 3, pp. 52–67, 2002.
21. K. Steinhöfel, A. Albrecht, C. K. Wong "Two simulated annealing-based heuristics for the job shop scheduling problem"European Journal of Operational Research, Volume 118, Issue 3, 1 November 1999, Pages 524-548.
22. M. Chandrasekaran . P. Asokan . S. Kumanan . T. Balamurugan . S. Nickolas "Solving job shop scheduling problems using artificial immune system" 8 February 2005 / Accepted: 30 July 2005 / Published online: 3 January 2006 # Springer-Verlag London Limited 2006.
23. M. Dorigo et L.M. Gambardella, Ant Colony System : A Cooperative Learning Approach to the

Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, volume 1, numéro 1, pages 53-66, 1997.

24. M.Mahdavi, M.Fesanghary, E.Damangir, "An improved harmony search algorithm for solving optimization problems", Applied Mathematics and Computation188 ,pp.1567–1579,2007.
25. M.Ramezani Ghalenoei, H.Hajmirsadeghi, C.Lucas “Discrete Invasive Weed Optimization and its application to Co-operative multi-task assignment of UAVs” Comin prac. 48th IEEE conference on Decision and Control, Dec 2009,in Press
26. Ritwik Giri ,Aritra Chowdhury, Arnob Ghosh “A Modified Invasive Weed Optimization for training of feed forward neural networks”
27. Runwei Cheng, Mitsuo Gen, Yasuhiro Tsujimura “A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies” Computers & Industrial Engineering, Volume 36, Issue 2, April 1999, Pages 343-364.
28. S. D. Müller, J. Marchetto, S. Airaghi, and P. Koumoutsakos, “Optimization based on bacterial chemotaxis,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, pp. 16–29, 2002.
29. Satellite Heat Pipe Design: Geem, Z. W. and Hwangbo, H. “Application of Harmony Search to Multi-Objective Optimization for Satellite Heat Pipe Design”, Proceedings of US-Korea Conference on
    Science, Technology, & Entrepreneurship (UKC 2006), CD-ROM, Teaneck, NJ, USA, Aug. 10-13 2006.
30. Siddharth Pal, Anniruddha Basak and Swagatam Das “A Invasive Weed Optimization method based Multi-user detection for MC-CDMA Interference suppression over multiple-path fading channel ” 978-422-6588-0/10/&25.00, IEEE,2010
31. Xueni Qiu · Henry Y. K. Lau, “An AIS-based hybrid algorithm for static job shop scheduling problem” Journal of Intelligent Manufacturing, 2019.