Performance Analysis of Opposition Based Particle Swarm Optimization with Cauchy Distribution in Minimizing Makespan Time in Job Shop Scheduling

Anil Kumar K. R., Edwin Raja Dhas

Abstract*: In the contemporary circumstances, manual solving of job shop scheduling problem (JSSP) is quite time consuming and inaccurate. The main intention of this paper is to analyze the performance of various optimization techniques in JSSP in order to minimize makespan time. This paper aims to analyze four optimization techniques viz, particle swarm optimization (PSO), genetic algorithm (GA), opposition based genetic algorithm(OGA) and opposition based particle swarm optimization with Cauchy distribution (OPSO CD) in addition to the existing optimization techniques applied in various research papers on combinatorial optimization problems viz., JSSP. A comparative study of these optimization techniques were conducted and analyzed to find out the most effective optimization technique on solving JSSP. Results show that OPSO CD is found to possess minimum makespan time in comparison with other algorithms for JSSP.

Keywords: Job shop scheduling, Opposition based particle swarm optimization with Cauchy distribution, makespan time, combinatorial optimization

I. INTRODUCTION

JSSP refers to the scheduling of various machines to handle a set of jobs. Each job needs to be processed by a given machine for a particular uninterrupted period of time. Makespan time refers to the duration in which all operations for all jobs are completed by a set of machines on the shop floor. The optimum scheduling of machines used in the production area heavily influences the actual cost of product and overall profitability of organization. JSSP deals with optimizing makespan time. Manual scheduling is time consuming and cumbersome. The solution is to select an appropriate optimization algorithm and apply to JSSP. The process of finding the suitable possible schedule from a numerous schedules is the main difficulty faced in scheduling problems [1]. JSSP is classified as an NP hard problem [2, 3, 4, 5] as the possible population size and the computational time for algorithms to find a suitable possible schedule increases exponentially with job machine size.

For JSSP, many heuristic and meta heuristic algorithms have been proposed such as fast simulated annealing hybridized with quenching [6] of Kashif Akram et al., TLBO algorithm of H.S.Keesari et al [7]., and Adil Baykasoglu et al[8]., Hybrid Genetic algorithm with a knowledge based operator of Hamid Piroozfard et al [9]., particle swarm optimization of Tsung- Lieh- Lin et al [1] ant colony optimization, fire fly algorithm, bee colony optimizations, cuckoo search optimization of K.Kanaka Vardhini et al[11].,simulated annealing approach of Baykosoglu[12] ,Van Laarhoven et al[13], Steinhoel et al[14], Suresh et al[15], Tabu search method of Nowicki et al[16], genetic algorithm of Gonclaves et al[17], Park et al[18], Wang and Zheng[19],Watanabe et al[20]. Jain and Meeran [21], Fan and Zhang [22], Vaessens et al [23] provided a comprehensive survey of various methods on JSSP. In this paper, OPSO CD has been found to achieve better solution in combinatorial optimization problems like JSSP.

PSO is developed by Kennedy and Eberhart [24] in 1995. This transformational algorithm was developed by the movement of a flock of birds seeking for food. This algorithm starts with an initial population of scattered solution called particles that are characterized by a position and a velocity. In PSO, a best fit solution is denoted by a particle that modifies its position and velocity so that the global best particle is obtained [25]. Once fallen in to local optima, the standard PSO fails to recognize the best existing local optima if any. This is a serious draw back with standard PSO. For achieving a better solution, some local search schemes should be incorporated with PSO as PSO’s local search ability is sluggish than global searching ability [10]. In this context, OPSO CD is applied to solve JSSP.

The remainder of this paper is structured as follows: Section 2 introduces OPSO CD algorithm and its implementation, whereas Section 3 illustrates the analysis of the proposed algorithm on JSSP with its results and discussions. Finally, Section 4 concludes the finding of this study with a few remarks.

II. OPPOSITION BASED PSO WITH CAUCHY DISTRIBUTION ALGORITHM AND ITS IMPLEMENTATION

Hamid R Tizhoosh introduced opposition based learning and proved its effectiveness by applying the same in optimization problems.

Revised Manuscript Received on February 01, 2020.
***** Correspondence Author
Anil Kumar K. R.*, Research Scholar, Noorul Islam Centre for Higher Educarion, Kanyakumari, India. Email: akrcem@yahoo.co.in
Dr. Edwin Raja Dhas, Department of Automobile Engineering, Noorul Islam Centre for Higher Educarion, Kanyakumari, India. Email: edwinrajadhas@rediffmail.com

The significant intention of this learning approach is to generate an opposition based solution for initially generated random solution.
The function to calculate opposition solution from initial solution is as follows;

$$O_i = x + y - I_i \quad (1)$$

$I_i \in I_1, I_2, I_3 \ldots$ NP, are initial solution randomly generated from which aforementioned equation (1) implies to generate opposition based solution where x and y are minimum and maximum values. For updating the initially generated solution and opposition-based solution, the following equations, (2) and (3) are utilized.

$$V_i^{(k+1)} = h * V_i^{(k)} + c_1 * r_1() * (p_{best_i} - X_i^{(k)}) + c_2 * r_2() * (g_{best} - X_i^{(k)})$$

(2)

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)} \quad (3)$$

Where inertia factor is represented as *h; r_1* and r_2 are the parameters where two random numbers are assigned in between zero and one. A couple of controlling parameters involved in this updating process is c_1 and c_2 and it is assigned as two. Position and velocity of *i* is said to be X_i and V_i and then i^{th} particles of previous best is considered to be p_{besti} and g_{best} whereas g_{best} (global best) is considered to be the best particle so far attained. The recent theoretical analysis verdicts that the particle on PSO fluctuates between previous and global best before it converge. To assist the entire particle in a better position, it is essential to include global best neighbors in consecutive generating solutions. This will be proficient by encompassing Cauchy distribution function to be a part of global best in every generation. The one-dimensional Cauchy density function centered at the origin is represented by:

$$f(x) = \frac{1}{\pi}\frac{t}{k^2 + x^2}, \quad -\infty < x < \infty \quad (4)$$

where, k > *0* is a scale parameter.

The Cauchy distributed function is:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{k}\right) \quad (5)$$

The Cauchy distribution operator utilized as a part of OPSO is as follows:

$$W(i) = \left(\sum_{j=1}^{pop\,size} V[j][i]\right) \bigg/ pop\,size \quad (6)$$

where *Pop Size* is the population size, *V[j][i]* is the i^{th} velocity vector of the j^{th} particle in the population and *w(i)* is a weight vector within [$-W_{max}$, W_{max}].

$$g_{best'(i)} = g_{best(i)} + w(i) * C_{df}(X_{\min}, X_{\max}) \quad (7)$$

where, C_{df} is a Cauchy distributed function with the scale parameter $k = 1$, and $C_{df}(X_{min}, X_{max})$ is a random number within [X_{min}, X_{max}], which is a characterized domain of a test function.

A. Pseudo-code for OPSO-Cauchy distribution

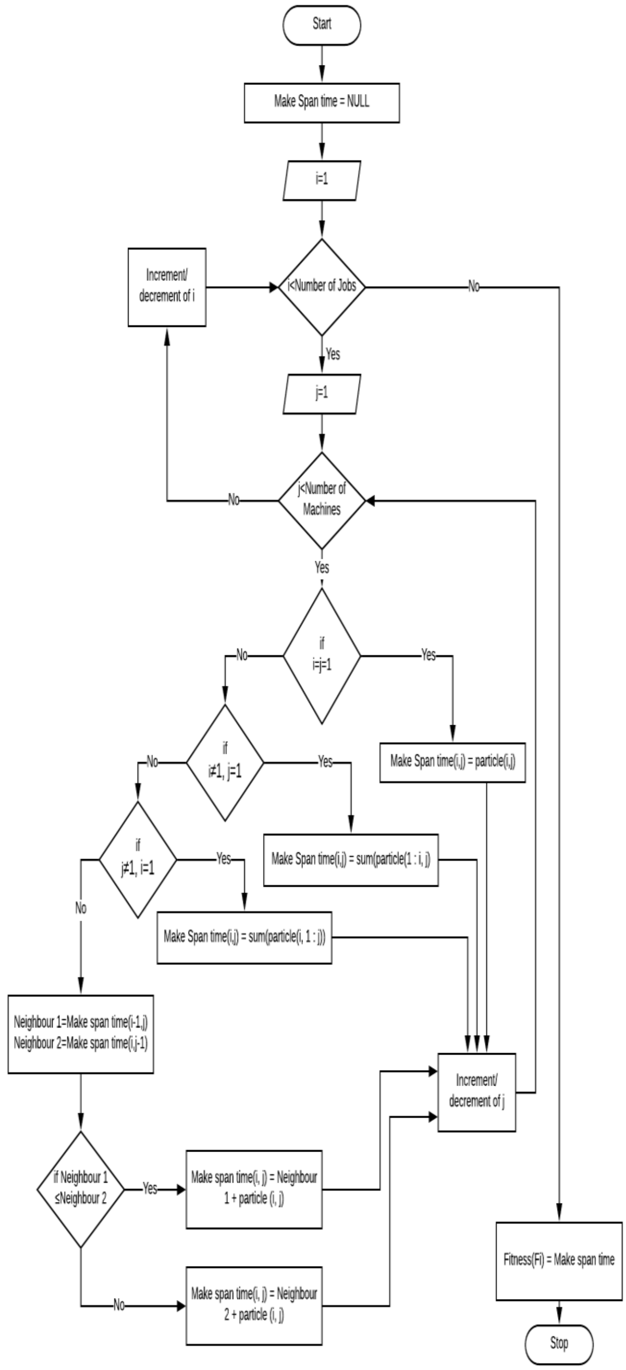
```
Generate initial solution [Ii]
Generate opposite solution [Oi]
Fitness computation [Fi]
Sorting [Pbest → Gbest]
Particle updation
  for i=1:NP
    Calculate particle velocity (2)
      Update particle (3)
    end
  for i=1:NP
    Calculate w(i) according (6)
    If w(i) > Wmax
      w(i) = Wmax
    else if w(i) < -Wmax
        w(i) = -Wmax
```

B. Influence of OPSO-Cauchy distribution in JSSP

OPSO-Cauchy distribution is a tool which is utilized to solve JSSP to attain minimum make span time. Here, machines are rearranged in a random sequential order for job allocation. This sequential rearranging considered as initial random solution generation in OPSO-Cauchy distribution for JSSP. On the other hand, opposition based generation is also generated for better utilization of initially generated solution.

Based on fitness value, the solutions sorted as P_{best} the optimal in that series is assumed to be G_{best} for the first time and those solutions are further updated by equation (2) and (3). Then, further G_{best} is updated by equation (7). The algorithm and flow chart of OPSO-Cauchy distribution in JSSP are shown below:

```
Make span time =Null matrix (size of problem)
  for i=1 : Number of jobs
    for j=1 : Number of machine
      if i=j=1
        make span time (i,j) = particle (i,j)
      else if i≠1 and j=1
        make span time (i,j) = sum (particle (1:i,j))
      else if i=1 and j≠1
        make span time (i,j) = sum (particle (i:1,j))
      else
        Neighbour 1 = make span time (i-1,j)
        Neighbour 2 = make span time (i,j-1)
      if Neighbour 1 ≤ Neighbour 2
        make span time (i,j) = Neighbour 2 + particle (i,j)
      else
        make span time (i,j) = Neighbour 1 + particle (i,j)
      end
    end
  end
Fitness (Fi) = make span time (end).
```



II. ANALYSIS OF ALGORITHM, RESULTS AND DISCUSSION

This section is devoted to the analysis of OPSO CD algorithm compared to genetic algorithm, PSO and opposition based genetic algorithm in addition to the existing eleven optimization techniques applied in research papers on combinatorial optimization problems viz., JSSP.

A. Analysis of Algorithm

Four optimization techniques like, GA, PSO, OPSO CD, OGA were applied to existing data which included eleven optimization techniques applied on 20 different benchmark problems [8]. The optimization algorithms studied in this platform are:

1. Teaching Learning Based Optimisation (TLBO)
2. Hybrid Genetic Algorithm (HGA 1)
3. Hybrid Genetic Algorithm (HGA 2)
4. Memetic Algorithm (MA)
5. Memetic Algorithm (Gap Reduction and Restricted Swapping) (MA GRRS)
6. Filler and Fan Algorithm (F& F)
7. Hybrid Genetic Algorithm (HGA)
8. Parallel Genetic Algorithm (PGA)
9. Greedy Randomised Adaptive Search Procedure (GRASP)
10. Hybrid Intelligent Algorithm (HIA)
11. Beam Search Algorithm (BSA)
12. Particle Swarm Optimisation (PSO)
13. Genetic Algorithm (GA)
14. Opposition Based Genetic Algorithm (OGA)
15. Opposition Based Particle Swarm Optimisation with Cauchy Distribution (OPSO CD)

Analysis was done on MATLAB with i3 processor having 6 GB RAM with a CPU speed of 2.20 GHz.

Table-I (a) Influence of contemporary heuristic techniques in benchmark problems [8]

Problems	BM [19]	TLBO [19]	HGA1 [19]	HGA2 [19]	MA [19]	MA(GR-RS) [19]	F&F [19]	HGA [19]
FT06	55	55	55	55	55	55	55	55
FT10	930	938	930	938	930	930	930	938
FT20	1165	1165	1165	1165	1165	1165	1165	1165
LA01	666	666	666	666	666	666	666	666
LA02	655	655	655	655	655	655	655	655
LA03	597	597	597	597	597	597	597	597
LA04	590	607	590	590	590	590	590	590
LA05	593	593	593	593	593	593	593	593
LA06	926	926	926	926	926	926	926	926
LA07	890	890	890	890	890	890	890	890
LA08	863	864	863	863	863	863	863	863
LA09	951	951	951	951	951	951	951	951
LA10	958	958	958	958	958	958	958	958
LA11	1222	1222	1222	1222	1222	1222	1222	1222
LA16	945	946	945	945	945	945	947	945
LA21	1046	1091	1046	1046	1055	1079	1052	1046
LA27	1235	1256	1236	1256	1261	1286	1242	1256
LA31	1784	1784	1784	1784	1784	1784	1784	1784
LA36	1268	1332	1287	1287	1281	1307	1381	1279
LA40	1222	1241	1229	1241	1233	1252	1228	1241
Average	**928.05**	**936.85**	**929.4**	**931.4**	**931**	**935.7**	**934.75**	**931**

Table-I (b) Influence of contemporary heuristic techniques in benchmark problems

Problems	BM [19]	PGA [19]	GRASP [19]	Beam search [19]	HIA [19]	PSO	GA	OGA	OPSO
FT06	55	55	55	55	55	55	55	55	55
FT10	930	936	938	1016	930	930	934	932	930
FT20	1165	1177	1169	none	1165	1165	1165	1165	1165
LA01	666	666	666	666	666	666	666	666	666
LA02	655	666	655	704	655	668	660	666	655
LA03	597	597	604	650	597	606	606	605	597
LA04	590	590	590	620	590	603	609	607	590
LA05	593	593	593	593	593	593	595	593	593
LA06	926	926	926	926	926	926	926	926	926
LA07	890	890	890	890	890	890	890	890	890
LA08	863	863	863	863	863	863	863	863	863
LA09	951	951	951	951	951	951	951	951	951
LA10	958	958	958	958	958	958	958	958	958
LA11	1222	1222	1222	1222	1222	1222	1222	1222	1222
LA16	945	977	946	988	945	951	959	957	946
LA21	1046	1047	1091	1152	1256	1050	1056	1052	1046
LA27	1235	1260	1320	1316	1784	1290	1269	1290	1240
LA31	1784	1784	1784	1784	1784	1784	1784	1784	1784
LA36	1268	1305	1334	1401	1281	1282	1310	1282	1277
LA40	1222	1252	1259	1347	1240	1239	1244	1237	1225
Average	928.05	935.75	940.7	952.73	967.55	934.6	936.1	935.05	928.95

From the above Table-1(a) and 1(b), it is quite convincing that optimization technique could do much better that conventional formulation. From Table 1 (b) it is evident that the average makes span time of OPSO CD is 99.1% closer to the average standard bench mark value. It is observed from the analysis that OPSO CD is found to be superior to the other 14 optimization techniques with respect to make span time.

B. Results and Discussions

1000 iterations were done in MATLAB and convergence graphs of the four optimization techniques applied viz., PSO, GA, OGA, and OPSO CD were obtained with makespan time versus iteration. It is evident from the convergence graph that OPSO CD reveals better convergence over other optimization techniques.

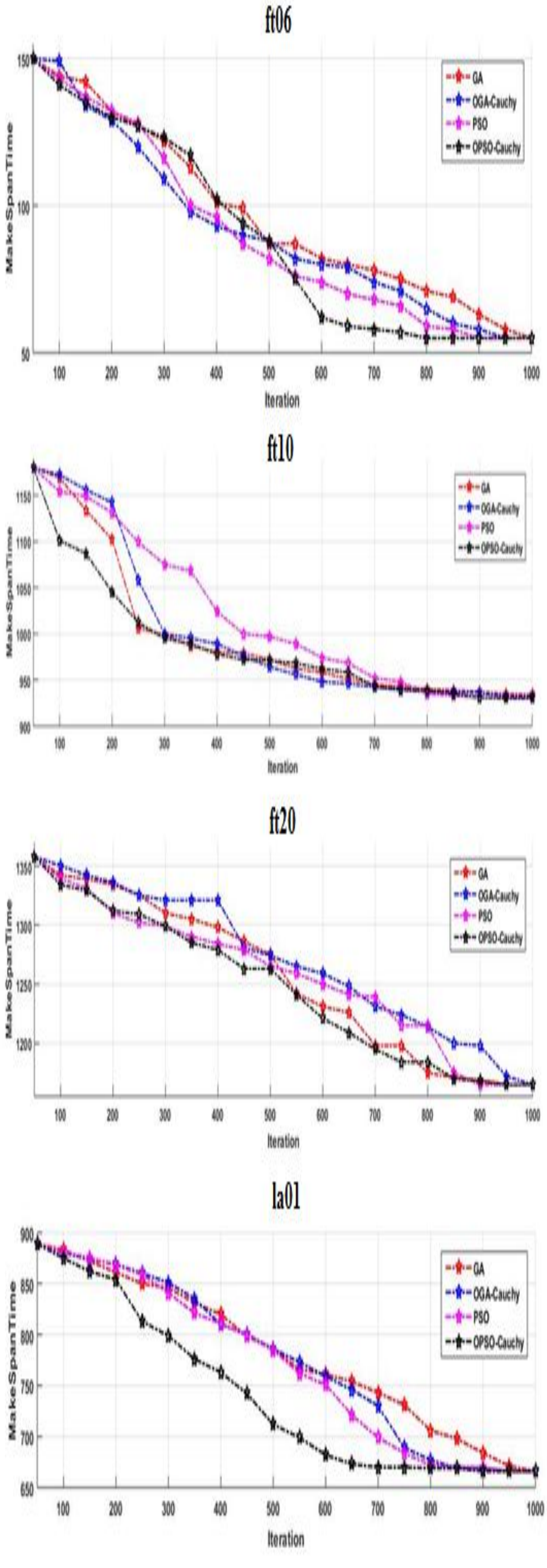


Fig. 1. Convergence graph for FT06, FT10, FT20, LA01

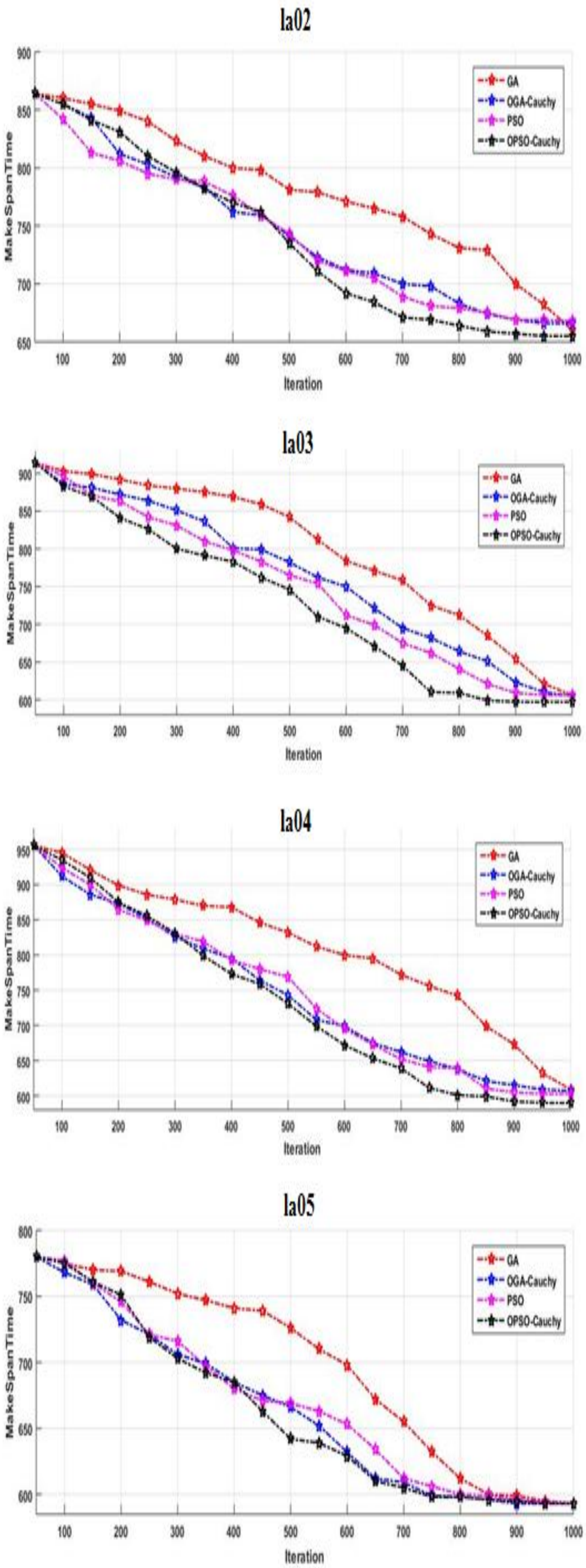


Fig. 2. Convergence graph for LA02, LA03, LA04, LA05

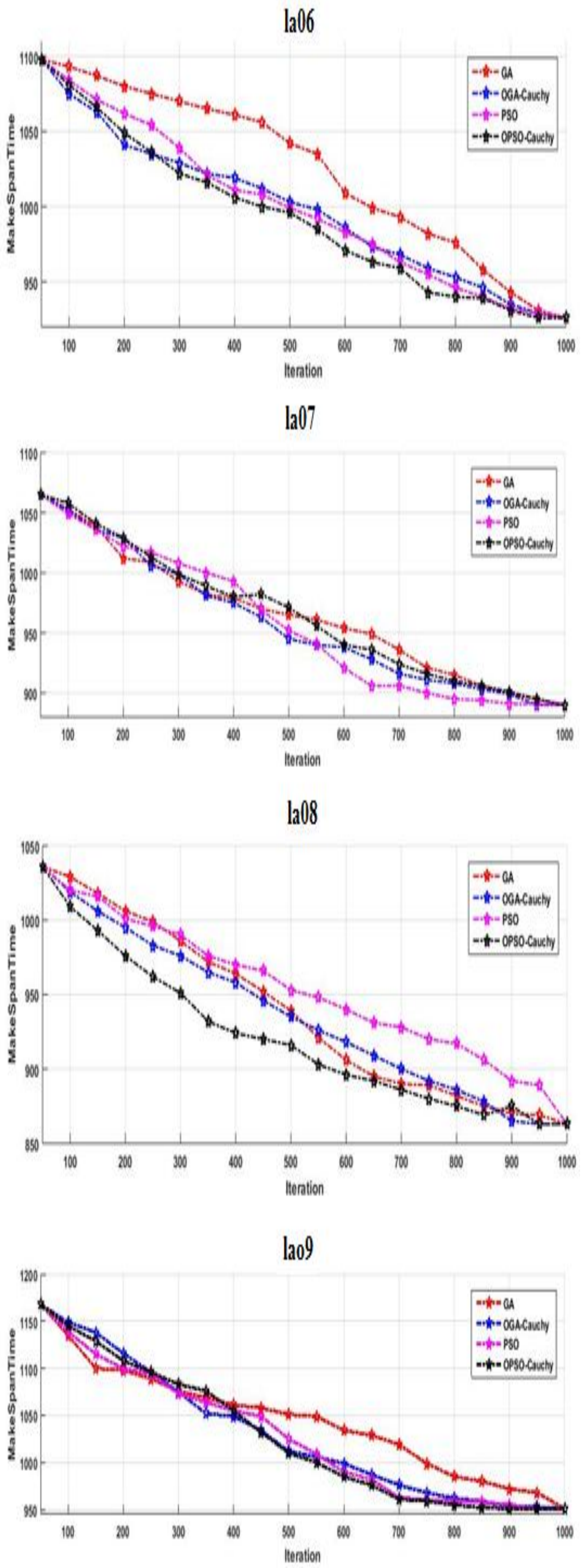


Fig. 3. Convergence graph for LA06, LA07, LA08, LA09

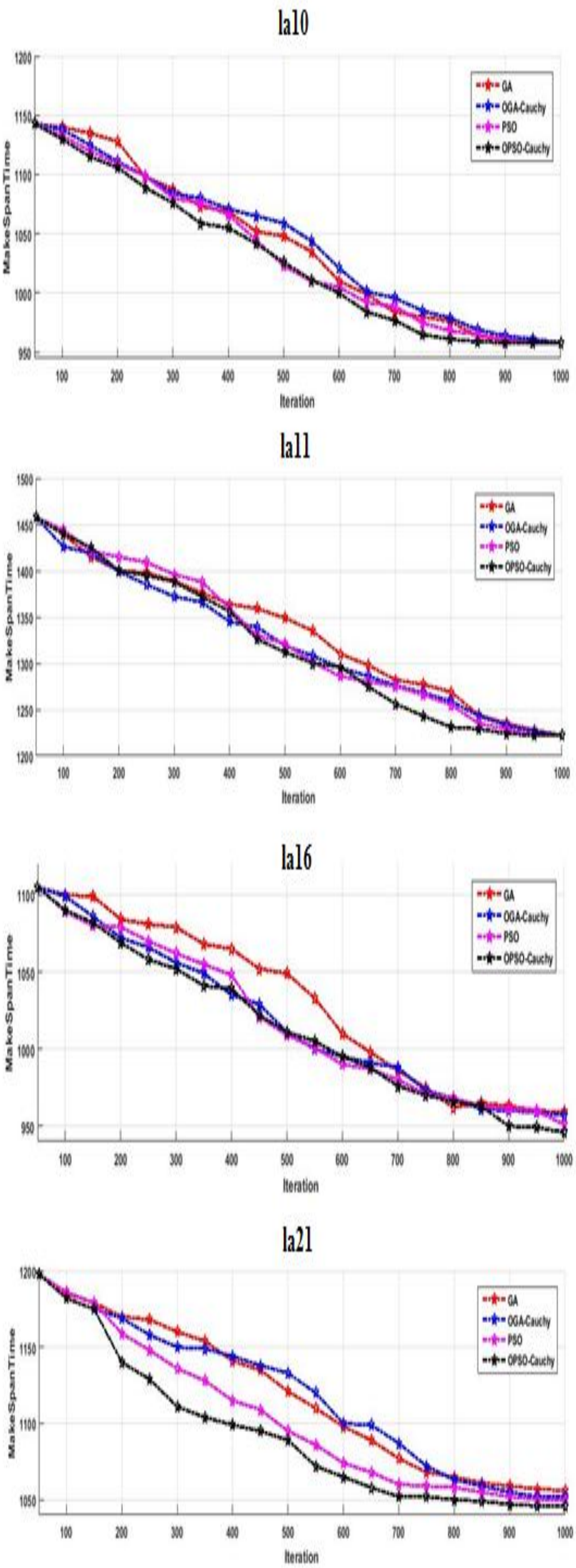


Fig. 4. Convergence graph for LA10, LA11, LA16, LA21

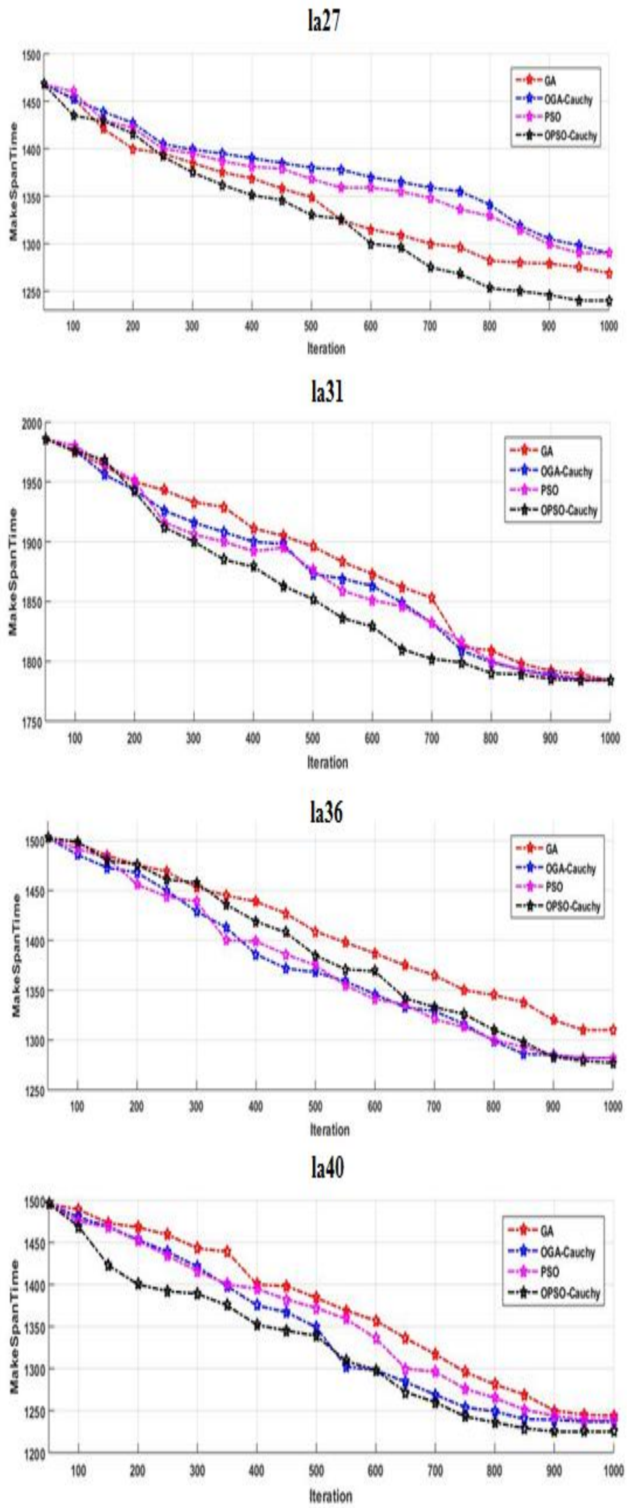


Fig. 5. Convergence graph for LA27, LA31, LA36, LA40

III. CONCLUSION

This study concludes that OPSO CD in JSSP is superior to other contemporary optimization methods. The results reveal that the benchmark values obtained through OPSO CD in JSSP is 99.1% closer to that of standard benchmark values. The minimized makespan time

Will certainly enhance the overall profitability of an organization. The OPSO CD in JSSP has a greater potential in providing better solution for makespan time minimization. It is a well known fact that JSSP is amongst the highly intricated combinatory optimization problems in literature. The OPSO algorithm on JSSP can give an idea about its possible performance for solving other combinatory optimization problems. However, there is further scope for application of newer optimization techniques on solving JSSP to further minimize makespan time.

REFERENCES

1. C. Anandaraman, An improved sheep flock heredity algorithm for job shop scheduling and flow shop scheduling problems, Int. J. Indust. Eng. Comput. 2(2011) 749–764.
2. B. Giffler, G.L. Thompson, Algorithms for solving production scheduling problems, Oper. Res. 8 (1960) 487–503
3. K.R. Baker, Introduction to Sequencing and Scheduling, Wiley, New York, 1974
4. M. Garey, D. Johnson, R. Sethi, The complexity of flowshop and jobshop scheduling, Math. Oper. Res. 1 (1976) 117–129
5. S. French, Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop, E. Horwood, Chichester, West Sussex, 1982
6. Kashif Akram, Khurram Kamal and Alam Zeb,"Fast simulated annealing hybridized with quenching for solving jobshop scheduling problem",Applied Soft Computing, Vol.49, pp.510-523, 2016.
7. H.S.Keesari and R. V. Rao,"Optimization of job shop scheduling problems using teaching-learning-based optimization algorithm",Opsearch, Vol.51, No.4, pp.545-561, 2014.
8. Adil Baykasoglu, Alper Hamzadayi and Simge Yelkenci Kose,"Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases", Information Sciences, Vol.276, pp.204-218, 2014.
9. Hamid Piroozfard, Kuan Yew Wong and Adnan Hassan,"A Hybrid Genetic Algorithm with a Knowledge-Based Operator for Solving the Job Shop Scheduling Problems",Journal of Optimization, Vol.2016, pp.1-13, 2016.
10. Tsung-Lieh Lin, Shi-Jinn Horng, Tzong-Wann Kao, Yuan-Hsin Chen, Ray-Shine Run, Rong-Jian Chen, Jui-Lin Lai and I-Hong Kuo,"An efficient job-shop scheduling algorithm based on particle swarm optimization" ,Expert Systems with Applications, Vol.37, pp.2629-2636, 2010.
11. K. KanakaVardhini and T. Sitamahalakshmi,"A Review on Nature-based Swarm Intelligence Optimization Techniques and its Current Research Directions",Indian Journal of Science and Technology, Vol.9, No.10, pp.1-13, 2016.
12. A. Baykasog˘lu, Linguistic based meta-heuristic optimisation model for flexible job shop scheduling, Int. J. Prod. Res. 40 (17) (2002) 4523–4543.
13. Van Laarhoven, P. J. M., Aarts, E. H. L., & Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. Operaons Research, 40(1), 113–125
14. Steinhöel, K., Albrecht, A., & Wong, C. K. (1999). Two simulated annealing-based heuristics for the job shop scheduling problem. European Journal of Operational Research, 118, 524–548
15. Suresh, R. K., & Mohanasundaram, K. M. (2005). Pareto archived simulated annealing for job shop scheduling with multiple objectives. The International Journal of Advanced Manufacturing Technology, 29, 184–196
16. Nowicki, E., & Smutnicki, C. (2005). An advanced tabu search algorithm for the job shop problem. Journal of Scheduling, 8, 145–159.
17. Goncalves, J. F., Mendes, J. J. D. M., & Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research, 167, 77–95.
18. Park, B. J., Choi, H. R., & Kim, H. S. (2003). A hybrid genetic algorithm for the job shop scheduling problems. Computers and Industrial Engineering, 45, 597–613.
19. Wang, L., & Zheng, D. Z. (2001). An effective hybrid optimization strategy for job shop scheduling problems. Computers and Operations Research, 28, 585–596.
20. Watanabe, M., Ida, K., & Gen, M. (2005). A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. Computers and Industrial Engineering, 48, 743–752.
21. A. Jain, S. Meeran, Deterministic job-shop scheduling: past, present and future, Eur. J. Oper. Res. 113 (1999) 390–434.
22. K. Fan, R-q. Zhang, An analysis of research in job shop scheduling problem (2000–2009), in: Y. Li, W. Desheng (Eds.), Proceedings 2010 IEEE International Conference on Advanced Management Science (ICAMS), Chengdu, 2010, pp. 282–288.
23. R. Vaessens, E. Aarts, J. Lenstra, Job shop scheduling by local search, INFORMS J. Comput. 8 (1996) 302–317.
24. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of IEEE international conference on neural networks (Vol. 4, pp. 1942–1948).
25. X.. Hu, Y. Shi and R. Eberhart, Recentt Advenes in Particle Swarm, Congress on Evolutionary Computation Portland, Oregon, June 19-23,90-97, 2004.

AUTHORS PROFILE

Anil Kumar K. R. graduated with B. Tech. degree in Mechanical Engineering from M. G. University Kottayam in the year 1994 and the M. Tech. degree focusing on Production Engineering from Calicut University in year 1999. He has nineteen years of teaching experience in Govt. Engineering College Munnar Kerala and currently serves as the Head of Mechanical Engineering Department of the college. He is a member in ISTE and is the President of ASME chapter of College of Engineering Munnar. His research interests are Industrial Engineering and, Production and Operations Management.

Edwin Raja Dhas received the B. E. Degree in Mechanical Engineering in the year 2000, M. E. in CAD/CAM in the year 2002 and the Ph.D. degree in Intelligent Manufacturing from National Institute of Technology Tiruchirappalli Tamil Nadu India. He has 16 years working experience in academic field as Project Associate, Lecturer and Professor. His research focuses are on Intelligent Manufacturing Systems, Computer aided Design and Drafting, Finite Element Analysis, Optimization in Design and Scheduling and, Intelligent Automation. He is Associate Member of Institution of Engineers, Member – Society of Automotive Engineers and Life Member of Indian Welding Society.