Hybridization of Soft computing Techniques for Flow Shop Scheduling Problem

Dr.V.Selvi
Assistant Professor,
Department of Computer Science
Mother Teresa Women's University
Kodaikanal.Tamilnadu.India.

***Abstract* --Since the last two decades, metaheuristic algorithms were developed in almost every aspect of various problems, where computational intelligence is used. In the field of computer science and operation research, PSO is used which is an optimization algorithm inspired by the social behaviour of birds flocking and fishes shoaling with its group. The original PSO was used to solve continuous optimization problems. Crossover and mutation of the particle is modified due to the discrete solution spaces of scheduling the optimization problems. Cuckoo search idealized such as breeding behaviour can be applied for optimization problems and it is successfully applied to various paths mostly continuous optimization problems. Swarm intelligence systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. The job scheduling problem is used for assigning the Flexible job shops in a way that will optimize the overall performance of the application, while assuring the correctness of the result. PSO and CS algorithm is proposed in this paper, for solving the Flexible job shop scheduling problem with an intention to decrease the maximum completion time. This paper has modifications to the PSO algorithm, which is based on Genetic Algorithm (GA) of crossover and mutation operators. Such modifications applied to the creation of new candidate solutions to improve the performance of the algorithm. Thus the use of the advantages of particle swarm optimization (PSO) and cuckoo search (CS) algorithm, a hybrid optimization algorithm of PSO and CS is proposed to solve the Flexible job shop scheduling problem. This paper shows that the proposed algorithm exhibits more outstanding performance than PSO-GA.**

Keywords: Particle Swarm Optimization, Artificial Bee Colony, Genetic algorithm, Flow Shop scheduling, cuckoo search

I. INTRODUCTION

Scheduling jobs has been a popular research for many years. There are many different ways to schedule jobs and the threads which make them up. However, only a few mechanisms are used in practice and studied in detail[1][2]. Users submit jobs in a batch wise to a resource management system queue, a centralized scheduler decides how to prioritize and allocate resources for job execution. To minimize the response time, the scheduling system strategy needs to prioritize competing user jobs with varying levels of priorities, importance and allocate resources for them accordingly.

A classical JSSP consists of N and M number of different jobs and machines, respectively. A set of operations and related machines with known processing time are involved in the process of each job. JSSP is a sequencing problem with no machine substitute for each operation while in FJSSP, alternative machines are considered for each operation. Due to the NP-hard class of FJSSP, meta-heuristic approaches have been widely utilized to solve it. Several inhabitants based on an algorithm have been proposed to find near-optimal solutions to the difficult optimization problems like scheduling and routing problems. An inhabitant based algorithm consisting of possible solutions to the problem are modified by applying some operators in the solution depending on the information of its fitness values. Several heuristic traditional algorithms were used for solving the Flexible job shop scheduling problem, based on an algorithm they are classified into Genetic algorithm (GA), Particle Swarm Optimization (PSO) algorithm and A new metaheuristic optimisation algorithm called Cuckoo Cuckoo search (CS) was introduced in 2009, and it has attracted huge attention due to its promising efficiency in solving many optimization problems and real-world applications. In the last few years, many papers have been published based on cuckoo search, and the relevant literature has expanded significantly. The optimal solutions obtained by CS are far better than the best solutions obtained by an efficient particle swarm optimiser.

The optimization methods based on the concept of swarm intelligence have been highlighted in the literature. These methods are based on the behaviour of social intelligence of groups of insects or animals that have characteristics of self-organization and decentralized control, with multiple agents. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA)[5]. The system is initialized with a population of random solutions and searches for optima by updating generations[19][20]. However, unlike GA, PSO doesn't have evolution operators such as crossover and mutation. In PSO, the potential solutions called particles that fly through the problem space by following the current optimum particles. Another reason why

PSO is attractive is that it has only few parameters to adjust. One version, with slight variations, works well with the wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as the specific applications focused on a specific requirement. For the past several years, PSO has been successfully applied in much research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Since the first introduction of Cuckoo Search (CS) by Xin-She Yang and Suash Deb in 2009[6], the literature of this algorithm has exploded. Cuckoo search, which drew its inspiration from the brooding parasitism of cuckoo species in Nature, were firstly proposed as a tool for numerical function optimization and continuous problems. Researchers tested this algorithm on some well-known benchmark functions and compared with PSO and GA, it was found that cuckoo search had better results than the results by PSO and GA[4]. Since then, the original developers of this algorithm and many researchers have also applied this algorithm to engineering optimization, where Cuckoo search also showed best results. Nowadays cuckoo search has been applied in almost every area and domain of function optimization, engineering optimization, image processing, scheduling, planning, feature selection, forecasting, and real-world applications method whose mechanisms are inspired by the swarming or collaborative behaviour of bio-logical populations like flock of birds or shoals of fish. Each single solution is a bird or fish in the search space and it is called a particle[6]. The PSO and its many versions have been popular in several branches of computer engineering including artificial neural network, wireless sensor networks [11] and many others mainly because of its intuitiveness, ease of implementation, and the ability to effectively solve highly nonlinear problems. The drawback of the PSO is its expensive computational cost in some circumstances.

The remainder of this paper is organized as follows: Section 2 briefly reviews the Genetic algorithm. Section 3 presents the basic PSO algorithm and discusses the modified PSO technique. Section 4 presents the basic Cuckoo Search algorithm and discusses the proposed PSO_CS technique, which is detailed in its subsections. Section 5 discusses about the Experimental Results and Discussion. Section 6 concludes the paper.

II. GENETIC ALGORITHM

In the computer science field of artificial intelligence, Genetic Algorithm (GA) is a search of heuristic that mimics the process of natural evolution[10]. This heuristic is routinely used to generate useful solutions to optimization and search problems. Basically, it consists of five components: a random number of generator, a fitness evaluation unit and genetic operators for reproduction; crossover and mutation operations.

Simple generational genetic algorithm procedure

1. *Choose the initial population of individuals*
2. *Evaluate the fitness of everyone in that population.*
3. *Repeat on this generation until termination (time limit, sufficient fitness achieved, etc.):*
   - *i. Select the best-fit individuals for reproduction*
   - *ii. Breed new individuals through crossover and mutation operations to give birth to offspring*
   - *iii. Evaluate the individual fitness of new individuals*
4. *Replace least-fit population with new individuals*

Fig 1**:** The procedure of Genetic algorithm

The initial population required at the start of the algorithm, is a set of food source generated by the random generator. Each position of a food source represents a possible solution of the optimization problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. A fitness value is a measure of goodness of the solution that it represents. Essentially the aim of the genetic operators is to transform this set of food source into sets with superior fitness values. The reproduction operator performs a natural selection function known as seeded selection. The crossover and mutation operator chooses pairs of food sources at random and produces new pairs. The simplest crossover and mutation operation is to cut the original food sources nectar amount at a randomly selected point and exchange. The number of crossover and mutation operations is governed by its rate. The mutation operator randomly mutates or reverses the values of food source. A phase of the algorithm consists of applying the evaluation, reproduction, crossover and mutation operations.

2.1. Genetic algorithm for Flexible Job Shop Scheduling

Step 1. Initialize the population by the input number of processors and number of jobs.
Step2. Start the process

Step2.1 Evaluate the fitness function (makes pan).

Step2. 2. Perform selection of the best individuals from the current population.

Step2. 3. Perform two-point crossover. Choose pairs of chromosomes (task) and a random point exchange machine assignments from that point until the end of the chromosome.

Step 2.4.Mutation: Randomly select a task and reassign it to the new machine.

Step3. The process is repeated until the stopping criterion is met. (Best fitness, minimum completion time) Step4. Stop the process

Fig 2: The procedure of genetic algorithm for flexible job shop scheduling

III. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population based stochastic optimization technique, inspired by social behaviour of bird flocking or fish schooling. Each particle keeps track of its coordinates in the problem spaces which are associated with the best solution (fitness) it has achieved so far[19]. (The fitness value is also stored.) This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbours of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbours, the best value is a global best and is called *gbest*.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) -------- (1)
present[] = present[] + v[] -----(2)

v[] is the particle velocity, present[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. Usually c1 = c2 = 2.

The pseudo code of the procedure is as follows.

```
For each particle
   Initialize particle
END
Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End
   Choose the particle with the best fitness value of all the particles as the gBest
   For each particle
     Calculate particle velocity according to equation---- (1)
      Update particle position according  to equation ----(2)
   End
While maximum iterations or minimum error criteria is not attained
```

Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, that is specified by the user. Then the velocity on that dimension is limited to Vmax.

Advantages and limitations of PSO

1. It's simple; it provides high quality of solution.
2. Fast convergence towards the sub optimal solution
3. PSO is based on the intelligence. It can be applied into both scientific research and engineering use.
4. Fewer parameters settings.
5. Easiness to implement.
6. Have the character of memory.

Limitations

1. Iterative nature of PSO can prohibit its use for high-speed real-time applications.

2. If optimization needs to be carried out frequently and not that convenient.
3. PSO requires large amounts of memory, which may limit its implementation to resource-rich base stations.
4. Premature convergence.
5. Easily drops into regional optimum or local minima.
6. The multiplicity of population is not enough.

3.1 Modified Particle Swarm Optimization for Scheduling

Step1: Initialize swarm size and particle values for each particle (Initialize the number of resources and number of jobs)
Step2: Choose the particle with the best fitness value of all the particles as the gbest does.
Step 3: For each particle: (generate a new fitness value by using the crossover and mutation operations.)

Step3.1 Perform two-point crossover for a Job and Choose a random point exchange machine assignments.
Step 3.2 Mutation: Randomly select a job and reassign it to the new machine.
Step 4: Update particle velocity and calculate it.
Step 5: Update particle position
Step 5: End the process
while maximum iterations or minimum error criteria are not attained (or the process is repeated until the stopping criterion is met. (Best fitness, minimum completion time)

Fig 3**:** The procedure of Modified Particle Swarm Optimization algorithm

IV. CUCKOO SEARCH

Cuckoo search (CS) is an optimization algorithm developed by Xin-she Yang and Suash Deb in 2009[13]. It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). Some host birds can engage direct conflict with intruding cuckoos. For example, if a host bird discovers the eggs which are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Some cuckoo species such as the New World brood-parasitic Tapera have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colors and pattern of the eggs of a few chosen host species[14].

Cuckoo search idealized such breeding behaviour, and thus can be applied for various optimization problems. It seems that it can outperform other metaheuristic algorithms in applications.

Cuckoo search (CS) uses the following representations:

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. In the simplest form, each nest has one egg[20]. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

CS is based on three idealized rules:

1. Every cuckoo lays one egg at a time, and dumps its egg in a arbitrarily chosen nest;
2. The best nests with top quality of eggs can carry over succeeding generation;
3. The number of existing host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa $\in$ [0, 1]. Discovering operates on some set of worst nests, and discovered solutions dumped from additional calculations.

In addition, Yang and Deb discovered that the random-walk style search is better performed by Lévy flights rather than simple random walk.

Based on the above-specified rules, the basic steps of the CS can be summarized as the pseudo code:

Objective function: f(x),x=(x1,x2,x3…,xd);
Generate an initial population of n host nests;
While(t<Max generation) or stop criterion
Get a cuckoo randomly (say ,i) and replace its solution by performing Lévy flights;
Evaluate its quality/fitness Fi
For maximization, $Fi \propto f(xi)$ *;*
Choose a nest among n (say, j) randomly

If ($F_i > F_j$)
Replace j by the new solution
End if;
A fraction (P_a) of the worse nests are abandoned and new ones are built
Keep the best solutions/nests
Rank the solutions/nests and find the current best
Pass the current best solutions to the next generation
End while

A Lévy flight is a random walk in which the step-lengths are distributed according to a heavy-tailed probability distribution. After a large number of steps, the distance from the basis of the random walk tends to a constant distribution. Some of the new solutions should be generated by L´evy walk around the best solution obtained so far, this will speed up the local search. Levy Flights However, a considerable fraction of the new solutions should be generated by far field randomization and whose locations should be far enough from the current best solution; this will make sure the system will not be trapped in a local optimum.

Modified cuckoo search

Step 1: start the process

Step 2: set the parameter and initialize the population

Step 3: calculate the initial fitness value of population

Step 4: set cuckoo search mode for the process.

Step 5: And then the Particle swarm search mode

Step 6: Perform the random elimination mechanism in it.

Step 7: Update the global optimal value and individual optimal value

Step 8 : Repeat the process until it reaches the terminal condition

Step 9: Get the output optimal solution

Step 10. Stop the process

V. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed algorithm is evaluated under three different job datasets and the results are compared against the conventional algorithms such as PSO, GA, CS, PSO_GA and PSO_CS. The results that are obtained in different experiments are given in following Table I.

TABLE 1**:** Job Completion Time for Job scheduling Dataset I by the proposed with (i) 50 iterations, (ii) 100 iterations and (iii) 150 iterations PSO,GA,CS, PSO_GA and PSO-CS algorithm with the number of 10 jobs and resources are 50,100 and 150.

No of jobs	No of Resources	Test Runs	PSO	GA	CS	PSO_GA	PSO_CS
10	50	50	218	176	193	160	154
		100	226	207	184	230	163
		200	210	223	167	220	136
10	100	50	218	235	212	235	210
		100	216	320	220	237	205
		200	194	316	216	186	178
10	150	50	122	192	142	114	101
		100	106	182	114	102	98
		200	95	163	96	80	78

The results that are obtained in different experiments are given in following Figures 4 to 6.

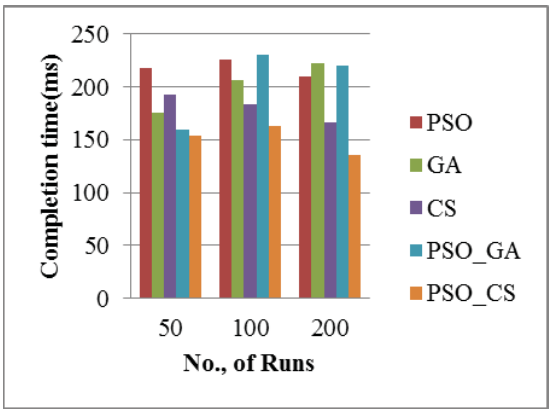

Fig 4: Mean Job Completion Time for individual Job Datasets proposed with 50 iterations, PSO, GA,CS, PSO_GA and PSO_CS

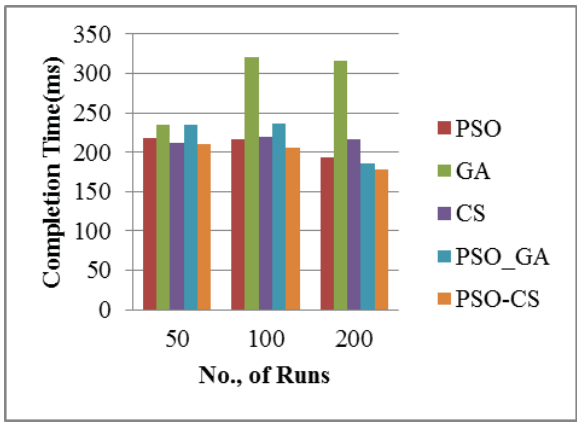

Fig 5: Mean Job Completion Time for individual Job Datasets proposed with 100 iterations PSO, GA,CS, PSO_GA and PSO_CS.

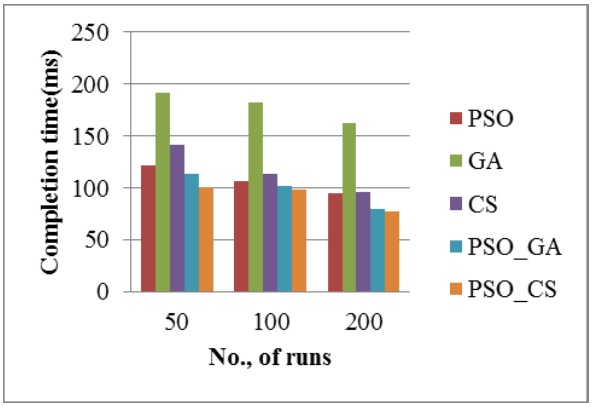

Fig 6: Mean Job Completion Time for individual Job Datasets proposed with 100 iterations PSO, GA,CS, PSO_GA and PSO_CS.

VI. CONCLUSION

In this paper, the proposed method has achieved the minimum completion and makes span time. The drawbacks of existing techniques were solved by considering some efficient factors in the Flexible job shop scheduling process. Thus, the proposed technique has achieved high performance in allocating the available jobs to the precise resources and also attained a high efficiency. The performance of the proposed job scheduling technique

(PSO_CS) was analyzed and compared with various techniques namely PSO, GA, CS, PSO_GA with experimental results that prove that the proposed job scheduling technique as attained high accuracy and efficiency than the other techniques. Hence, the proposed adaptive PSO_CS job scheduling technique is capable of finding the optimal jobs to the resources and also achieving the minimum completion time.

REFERENCES

[1] Dror G. Feitelson, Larry Rudolph and Uwe Schwiegelshohn, "Parallel Job Scheduling -A Status Report", In Proeedings of the Conference on JSSPP, pp.1-16, 2004.

[2] Ivan Rodero, Francesc Guim and Julita Corbalan, "Evaluation of Coordinated Grid Scheduling Strategies", In Proceedings of 11th IEEE International Conference on High Performance Computing and Communications, Seoul, pp. 1-10, 2009.

[3] Hamed Samarghandi et al.,"A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion",The International Journal of Advanced Manufacturing Technology, August 2012, Volume 61, Issue 9, pp 1101-1114.

[4] Ali Allahverdi and Harun Aydilek," Algorithms for no-wait flowshops with total completion time subject to makespan", The International Journal of Advanced Manufacturing Technology,October 2013, Volume 68, Issue 9, pp 2237-2251.

[5] Ala'a Abu et al., "A Hybrid Algorithm Using a Genetic algorithm and Cuckoo Search Algorithm to Solve the Traveling Salesman Problem and its

[6] Application to Multiple Sequence Alignment", International Journal of Advanced Science and Technology Vol.61(2013), pp.29-38,ISSN:2005.

[7] Azarkish et al.,"A new hybrid mutli-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem", Elsevier Expert Systems with Applications (2011).

[8] Grudenic and Bogunovi, "Computer Cluster Scheduling Algorithm Based on Time Bounded Dynamic Programming", In Proceedings of the 34th International Convention on MIPRO, 2011, Opatija, pp. 722-726, 2011.

[9] Abdelrahman Elleithy, Syed S. Rizvi and Khaled M. Elleithy, "Optimization and Job Scheduling in Heterogeneous Networks ", International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, 2008.

[10] Jianchao Tang et al., "Hybrid Genetic Algorithm for Flow Shop Scheduling Problem", Intelligent Computation Technology and Automation (ICICTA), International Conference on (Volume:2), Page(s):449 – 452,E-ISBN :978-1-4244-7280-2, 2010.

[11] Surekha and Sumathi, "Solution to the Job Shop Scheduling Problem using Hybrid Genetic Swarm Optimization Based on (λ, 1)-Interval Fuzzy Processing Time", European Journal of Scientific Research, Vol. 64, No. 2, pp. 168-188, 2011.

[12] Bin Cai, Shilong Wang and Haibo Hu, "Hybrid Artificial Immune System for Job Shop Scheduling Problem", World Academy of Science, Engineering and Technology, Vol. 59, No. 18, pp. 81-86, 2011.

[13] Mohammad Akhshabi, Mostafa Akhshabi and Javad Khalatbari, "Parallel Genetic Algorithm to Solving Job Shop Scheduling Problem", Journal of Applied Sciences Research, Vol. 1, No. 10, pp. 1484-1489, 2011.

[14] Fister Jr., X. S. Yang, D. Fister, I. Fister, Cuckoo search: A brief literature review, in: Cuckoo Search and Firefly Algorithm: Theory and Applications, Studies in Computational Intelligence, vol. 516, pp. 49-62 (2014).

[15] Yang, X.-S., and Deb, S. (2010), "Engineering Optimisation by Cuckoo Search", Int. J. Mathematical Modelling and Numerical Optimisation, Vol. 1, No. 4, 330–343 (2010).

[16] Hadi Mokhtari, "Adapting a Heuristic Oriented Methodology for Achieving Minimum Number of Late Jobs with Identical Processing Machines", Research Journal of Applied Sciences, Engineering and Technology, Vol. 4, No. 3, pp. 245-248, 2012.

[17] Elnaz ZM, Amir MR, Mohammad R, Feizi D (2008). Job Scheduling in Multiprocessor Architecture Using Genetic Algorithm. Proc. IEEE, pp. 248-250.

[18] Tung-Kuan L, Jinn- Tsong T, Jyh-Hong C (2005). "Improved genetic algorithm for the job-shop scheduling problem". International Journal Advanced Manufacture Technology (Spiringer), pp. 1021-1029.

[19] D. Y. Sha and Hsing-Hung Lin ,"A multi-objective PSO for job-shop scheduling problems Expert Systems with Applications", Volume 37, Issue 2, March 2010, Pages 1065–1070.

[20] K. Thanushkodi, K. Deeba, "On Performance Analysis of Hybrid Algorithm (Improved PSO with Simulated Annealing) with GA, PSO for Multiprocessor Job Scheduling", ISSN: 1109-2750 287Issue 9, Volume 10, September 2011.

[21] Kao, Ming-Hsien Chen, and Yi-Ting Huang, "Research Article A Hybrid Algorithm Based on ACO and PSO for Capacitated Vehicle Routing Problems", Mathematical Problems in Engineering,Volume 2012.

[22] Deepak Singh, Ankit Sirmorya, "Solving Real Optimization Problem using Genetic Algorithm with Employed Bee (GAEB)",International Journal of Computer Applications (0975 – 8887) Volume 42– No.11, March 2012.