ISSN (e): 2250 – 3005 || Vol, 04 || Issue, 5 || May – 2014 ||
International Journal of Computational Engineering Research (IJCER)

A Review on Parallel Scheduling of Machines and AGV'S in an FMS environment

Prof.Mallikarjuna[1],Vivekananda[2], Md.Jaffar[2], Jadesh[2], Hanumantaraya[2]
[1](Assistant Professor, Dept of ME, Ballari Institute Of technology & Management, Bellary, India)
[2](Dept of ME, Ballari Institute Of technology & Management, Bellary, India)

ABSTRACT:
This paper focuses on the problem of parallel scheduling of machines and automated guided vehicles(AGV's) in a flexible manufacturing system (FMS) so as to minimize the make span. Here several algorithms were employed to solve this combinatorial optimization problem. In this paper the authors have attempted to schedule both the machines and AGV's parallelly for which Differential Evolution(DE) and Simulated annealing(SA) techniques are applied. The impact of the major contribution is indicated by applying these techniques to a set of standard benchmark problems. For this particular problem coding has been developed, which gives the optimum sequence with make span value and AGV's schedule for number of job sets. Finally a comparison of DE and SA applied to a highly generalized class of job shop scheduling problems is to be done and results to be concluded

KEYWORDS: *AGV's, Benchmark problem, Combinatorial optimization problem, Differential Evolution, Flexible manufacturing system, Job shop scheduling, Simulated Annealing.*

I. INTRODUCTION

A flexible manufacturing system is that kind of manufacturing system in which there is some amount of flexibility that allows the system to react in the case of changes, whether predicted or unpredicted.
The main purpose of the FMS to combine the job shop and productive of flow line
For this purpose it possesses 2 kinds of flexibility
(1) Machine flexibility (2) Routing flexibility

Machine flexibility covers the system ability to be changed to produce new product types and ability to change the order of operations executed on a part.Routing flexibility consists of ability to use multiple machines to perform the same operation on a part as well as the system ability to absorb large scale changes such as in volume, capacity or capability.

FMS system comprises of 3 main systems those are :
[1] The machines which are often automated to CNC machines
[2] Material Handling system to which the CNC machines are connected in order to optimize the parts flow.
[3] And the Central Control Computer which controls material movement and machine flow.

This paper has mainly considered with an FMS that is required to process various types of job loaded in a least possible time at different processing stations. Each job requires a definite sequence of operations that indicates the order in which the operations are to be performed.The production demands a flexible MHS to move the parts to various processing stations during a production run. Automated Guided Vehicles [AGV's] are used in many material handling situations involving large, heavy loads over flexible routes.AGV's are the most flexible of the floor cart system and follow electromagnetic impulses transmitted from a guided wire embedded in the plant floor.Configuration and operating environment of FMS involving AGV's and CNC Machines is as shown in the below figure

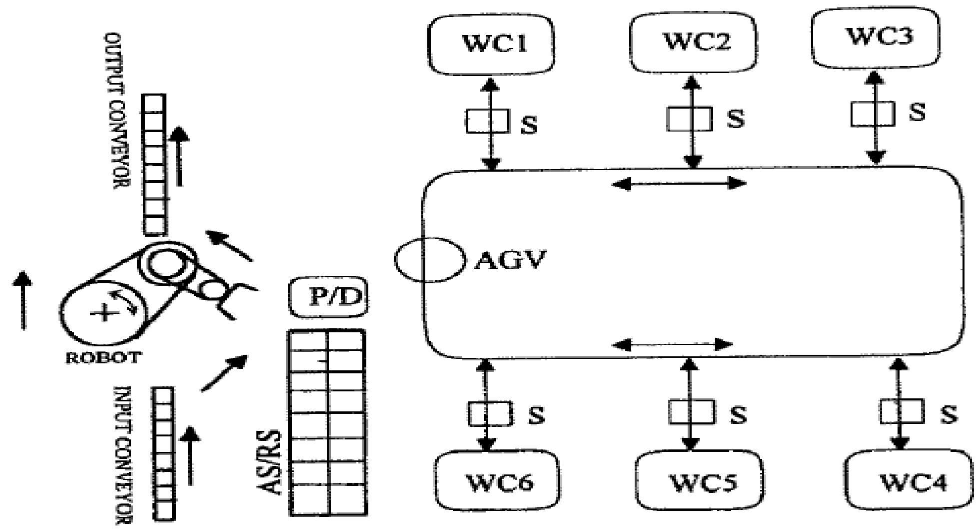


Fig. 2. Configuration of a flexible manufacturing system. WC,work cell; AGV, automated guided vehicle; P/D, pick and deposit; AS/RS, automated storage/retrieval system; S, shuttle.

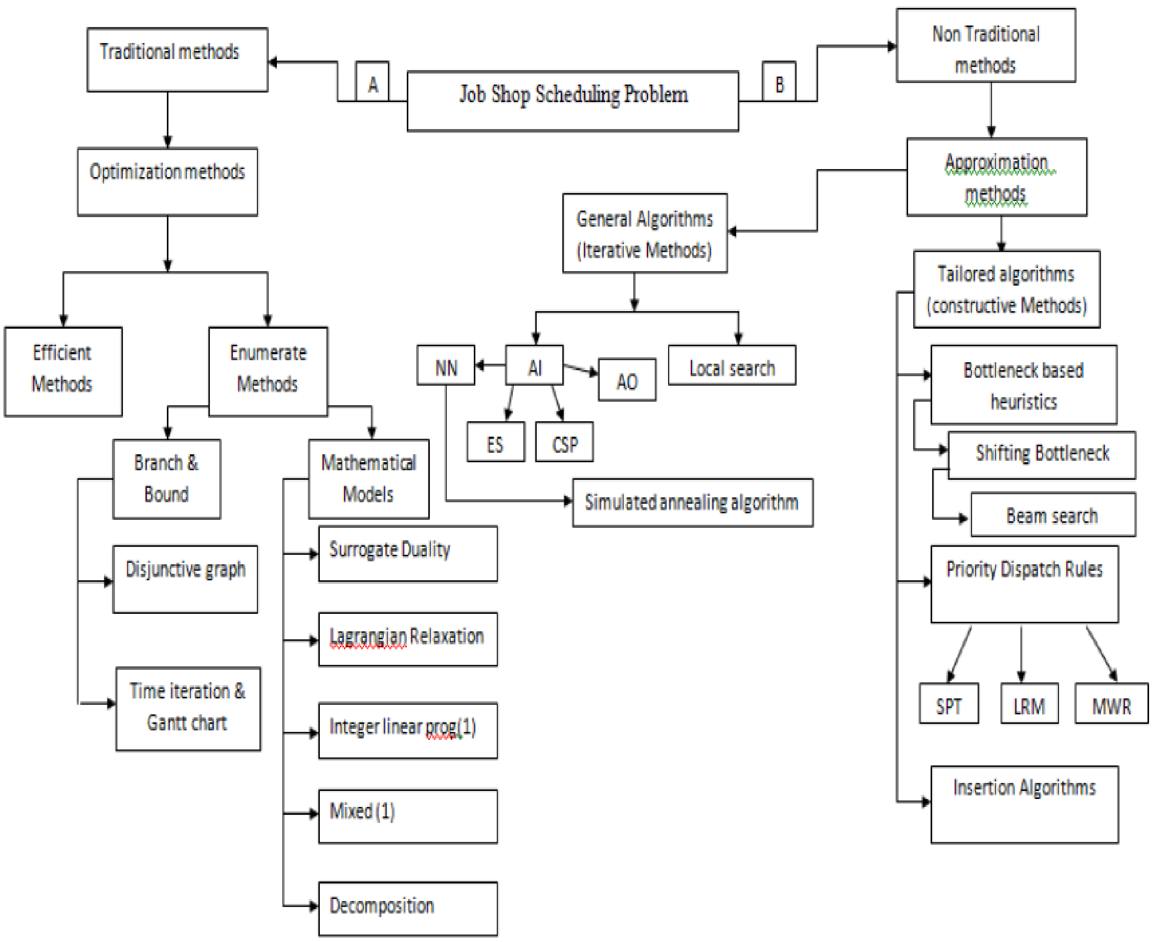
The main advantage of an FMS are its high flexibility in managing manufacturing resources like Time and Effort in order to manufacture a new product.The best application of an FMS is found in the production of small sets of products like those from a mass production.

II. OBJECTIVES OF SCHEDULING :

The scheduling is made to meet specific objectives. The objectives are decided upon the situation, market demands, company demands and the customer's satisfaction. There are two types for the scheduling objectives :

[1] Minimizing the make span
[2] Due date based cost minimization

The objectives considered under the minimizing the make span are,
(a) Minimize machine idle time
(b) Minimize the in process inventory costs
(c) Finish each job as soon as possible
(d) Finish the last job as soon as possible
The objectives considered under the due date based cost minimization are,
(a) Minimize the cost due to not meeting the due dates
(b) Minimize the maximum lateness of any job
(c) Minimize the total tardiness
(d) Minimize the number of late jobs



III. LITERATURE REVIEW:

Sl no.	Authors	Technique used	Remarks
1.	C.S.P Rao, M.V.Satish kumar, G.Rangajanardhan	Hybrid Differential Evolution	In this by the application of differential evolution, the simultaneous scheduling of machines and AGV's has been done. In this, operation based coding system is employed to represent the solution vector, which is further modified to suit the DE application
02.	B.S.P.Reddy, C.S.P. Rao	Hybrid Multi Objective Genetic Algorithm	In this the authors have made an attempt to consider simultaneously the machine and the vehicle scheduling aspects in an FMS and addressed the combined problem for the minimization of make span, mean flow time and mean tardiness objectives.
03.	S. V. Kamble & K. S. Kadam	Particle Swarm Optimization	Minimizing the idle time of the m/c and minimizing the total penalty cost for not meeting the dedline concurrently. To achieve this it is necessary to determine the routing for jobs processing sequence of operation on m/c and the starting time for operation on m/c and starting time for operation in order to balance the workload of machine.

04.	Ghada El Khayat, Andre Langevin, Diane Riopel	Mathematical Programming and Constraint Programming	Machines and MHE are considered as the constraining resources. A methmetical programming model and a constraint programming model presented for the problem and solved optimally on test problems used for modeling, testing and integrating both models in a decision support system. The performance of two methods is comparable when using the data from the literature.
05.	Muhammad Hafidz Fazli bin Md Fauadi and Tomohiro Murata	Binary Particle Swarm Optimization	It exploits a population of particles to search for promising regions of the search space (swarm). While each particle randomly moves within the search space with a specified velocity. It stores data of the best position it ever encountered. This is known as personal best (pbest) position. Upon finishing each iteration, the pbest position obtained by all individuals of the swarm is communicated to all of the particles in the population. The best value of pbest will be selected as the global best position (Gbest) to represent the best position within the population.
06.	K.V.Subbaiah, M.Nageswara Rao and K. Narayana Rao	Sheep Flock Heredity Algorithm	For this particular problem, coding has been developed, which gives optimum sequence with makespan value and AGV'S schedule for ten job sets and four layouts. Most of the time, results of sheep flock algorithm are better than other algorithm and traditional methods.
07.	Paul Pandian,P, S. Saravana Sankar,S.G.Ponnambalam and M. Victor Raj	Jumping Genes Genetic Algorithm	The one of best evolutionary approach i.e., genetic algorithm with jumping genes operation is applied in this study, to optimize AGV flow time and the performance measures of Flexible Job shop manufacturing system. The non dominated sorting approach is used.Genetic algorithm with jumping genes operator is used to evaluate the method.
08.	J.Jerald,P.Asokan, R. Saravanan ,A. Delphin Carolina Rani	Adaptive Genetic Algorithm	Two contradictory objectives are to be achieved simultaneously by scheduling parts and AGVs using the adaptive genetic algorithm. The results are compared to those obtained by conventional genetic algorithm.
09.	M.K.A.Ariffin, M.Badakhshian, S.B.Sulaiman, A.A.Faeiza	Fuzzy Genetic Algorithm	Fuzzy logic controller (FLC) is proposed to control the behaviour of genetic algorithm (GA) to solve the scheduling problem of AGVs. This paper presents an FLC to control the crossover and mutation rate for controlling the GA

IV. SCHEDULING TECHNIQUES :

These techniques are mainly divided into two categories i.e. Traditional and Non Traditional. A brief introduction of these techniques are given below.

(a) Traditional techniques:

❖ These techniques are slow and guarantee of global convergence as long as problems are small. Traditional Techniques are also called as Optimization Techniques. They are

- Mathematical programming like Linear programming, Integer programming, Dynamic programming, Transportation etc.
- Enumerate Procedure Decomposition like Lagrangian Relaxation.

(b) Non traditional techniques:

❖ These methods are very fast but they do not guarantee for optimal solutions. Non Traditional Techniques are also called as Approximation Methods. They involve

- Constructive Methods like Priority dispatch rules, composite dispatching rules.
- Insertion Algorithms like Bottleneck based heuristics, Shifting Bottleneck Procedure.
- Evolutionary Programs like Genetic Algorithm, Particle Swarm Optimization.
- Local Search Techniques like Ants Colony Optimization, Simulated Annealing, adaptive Search, Tabu Search, Problem Space methods
- Iterative Methods like Artificial Intelligence Techniques, Artificial Neural Network, Heuristics Procedure, Beam-Search, and Hybrid Techniques.

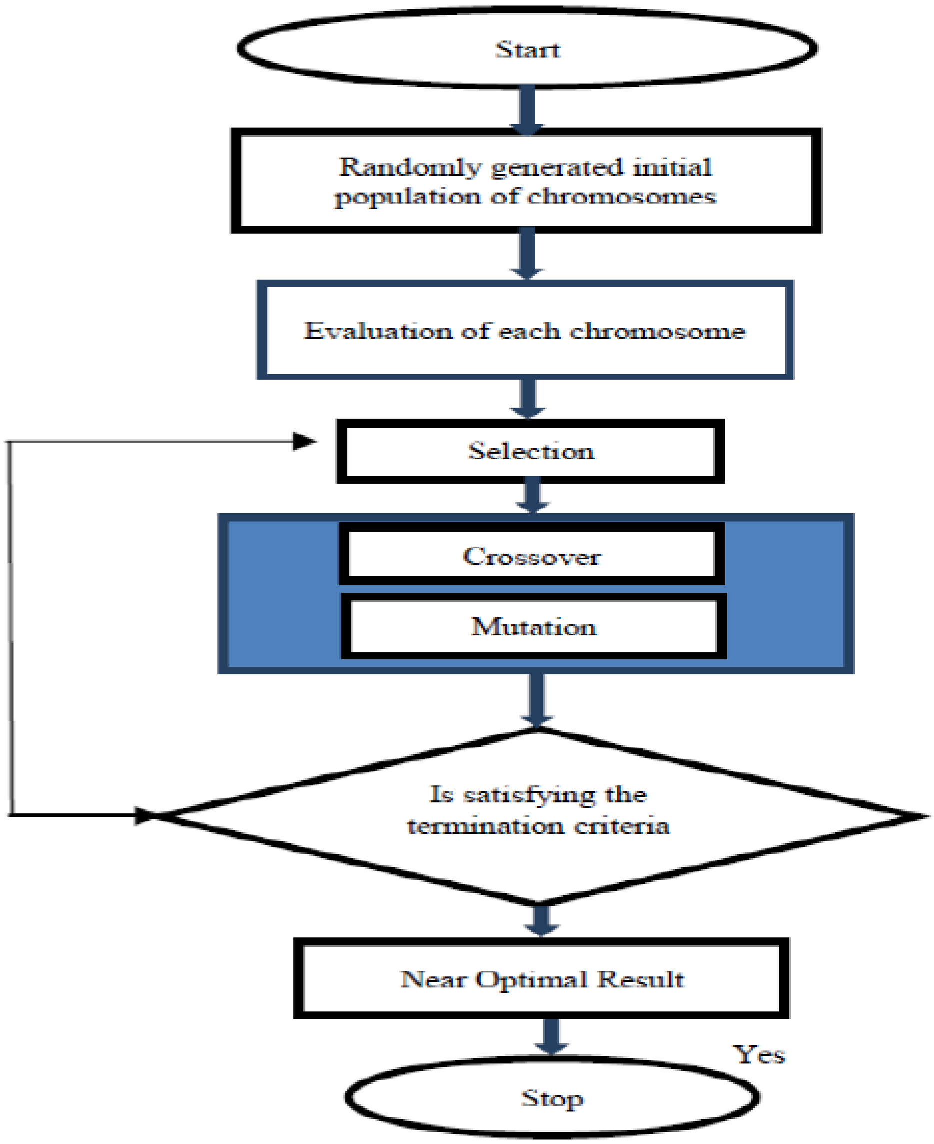
V. SOME NON TRADITIONAL TECHNIQUES :

Genetic algorithm:

A Genetic Algorithm is an `intelligent‟ probabilistic search algorithm that simulates the process of evolution by taking a population of solutions and applying genetic operators in each reproduction. Each solution in the population is evaluated according to some fitness measure. Fitter solutions in the population are used for reproduction. New `off spring‟ solutions are generated and unfit solutions in the population are replaced.

GA coding scheme:
As GA works on coding of parameters, the feasible job sequences (the parameter of the considered problems) are coded in two different ways.
(i) Pheno style coding
(ii) Binary coding

Algorithm:
Step 1: Generate random population of n chromosomes(suitable solution for problem)
Step 2: Evaluate the fitness f(x) of each chromosome x in the population.
Step 3: Create a new population by repeating following steps until the new population is complete
Step 4: Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
Step 5: With a crossover probability crossover the parents to form a new offspring(children). If no crossover was performed, offspring is an exact copy of parents.
Step 6: with a mutation probability mutate new offspring at each locus(position in chromosome)
Step 7: place new offspring in a new population
Step 8: use new generated population for a further run of algorithm.
Step 9: if the end condition is satisfied, stop, and returns the best solution in current population and Go to step 2.



Simulated Annealing:

The simulated annealing algorithm resembles the cooling process of molten metals through annealing. At high temperature, the atoms in the molten metal can move freely with respect to each another. But, as the temperature is reduced, the movement of the atoms gets reduced. The atoms start to get ordered and finally form crystals having the minimum possible energy. However, the formation of the crystal depends on the cooling rate. If the temperature is reduced at a fast rate, the crystalline state may not be achieved at all; instead the system may end up in a polycrystalline state, which may have a higher energy state than the crystalline state. Therefore, to achieve the absolute minimum state, the temperature needs to be reduced at a slow rate. The process of slow cooling is known as annealing in metallurgical parlance. SA simulates this process of slow cooling of molten to achieve the minimum function value in a minimization problem. The cooling phenomenon is simulated by controlling a temperature – like parameter introduced with the concept of the Boltzmann probability distribution.

Algorithm:

Step 1: Choose an initial point X^(o), a termination criterion Î. Set T as a sufficiently high value number of iterations to be performed at a particular temperature n, and set t=0.
Step 2: Calculate the neighbouring point X^(t+1) = N(x^(t)). Usually, a random point in the neighbourhood is created.
Step 3: If DE= E(x^(t+1))-E(x^(t))<0, set t=t+1; Else create a random number (r) in the range (0,1). If r_exp(DE/T), set t=t+1; Else go to Step 2.
Step 4: if (x^(t+1)-x^(t))< Î and T is small, Terminate. Else go to Step 2.

VI. DIFFERENTIAL EVOLUTION:

Evolutionary Algorithms (EA) which simulates evolution process in computer have created lot of interest among the researchers, which led to their application in a variety of fields. Genetic Algorithms, GA (Holland, J.H., 1975) are popular among the EAs and they were used to address scheduling problems by many researchers. Even though GA can be considered as a better searching algorithm and many versions of GA were developed by several researchers, still developed a hybrid GA procedure, which uses operation based coding for scheduling machines. They have also developed a heuristic to solve the vehicle scheduling, because of which they have reduced the length of the chromosome to half that created by Ulusoy. Lacomme et al., (2005) addressed the job input sequencing and vehicle dispatching in a single vehicle automated guided vehicle system. They have coupled the heuristic branch and bound approach with discrete event simulation model. Siva P.Reddy et al., (2006) have attempted the same problem set as that of Ulusoy and Tamer with a modified GA approach. Jerald et al.,(2006) have used an adoptive genetic algorithm for solving the simultaneous scheduling of parts and AGVs problem.

Mutation

Unlike GA, where mutation follows crossover, in DE mutation will be performed first.
Three vectors xr1, xr2 , xr3 which are different from the current vector will be randomly selected and the weighted difference of two vectors in the population is added to a third vector to get the resultant vector known as mutant vector (v i, g+1), as given below

V i, g+1= xr1,g+ F(x r2,g – x r3,g)
Where F > 0 is scaling factor, which controls the magnitude of the differential variation of (xr2,g - x r3,g).

VII. TABU SEARCH:

The basic idea of Tabu search (Glover 1989, 1990) is to explore the search space of all feasible scheduling solutions by a sequence of moves. A move from one schedule to another schedule is made by evaluating all candidates and choosing the best available, just like gradient-based techniques. Some moves are classified as tabu (i.e., they are forbidden) because they either trap the search at a local optimum, or they lead to cycling (repeating part of the search). These moves are put onto something called the Tabu List, which is built up from the history of moves used during the search. These tabu moves force exploration of the search space until the old solution area (e.g., local optimum) is left behind. Another key element is that of freeing the search by a short term memory function that provides "strategic forgetting". Tabu search methods have been evolving to more advanced frameworks that includes longer term memory mechanisms. These advanced frameworks are sometimes referred as Adaptive Memory Programming (AMP, Glover 1996). Tabu search methods have been applied successfully to scheduling problems and as solvers of mixed integer programming problems. Nowicki and Smutnicki (Glover 1996) implemented tabu search methods for job shop and flow shop scheduling problems. Vaessens (Glover 1996) showed that tabu search methods (in specific job shop scheduling cases) are superior over other approaches such as simulated annealing, genetic algorithms, and neural networks.

```
Algorithm :
Begin
        T:=[ ];
        S:=initial solution;
        S*:=s
        Repeat
                Find the best admissible s' cN(s);
                If f(s')>f(s*)then s*:=s'
                S:=s';
                Update tabu list T;
        Until stopping criterion:
End;
```

Particle swarm Optimization :

One of the latest evolutionary techniques for unconstrained continuous optimization is particle swarm optimization (PSO) proposed by Kennedy and Eberhart (1995), inspired by social behaviour of bird flocking or fish schooling. PSO learned from these scenarios are used to solve the optimization problems. In PSO, each single solution is a "bird" in the search space.[37],[38],& [39]. We call it "particle". All of particles have fitness values, which are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles. The particles are "flown" through the problem space by following the current optimum particles.PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (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 population. This best value is a global best and called 'gbest'.

Algorithm :

Step 1: initialize a population on *n* particles randomly.

Step 2: Calculate fitness value for each particle. If the fitness value is better than the best fitness value (pbest) in history. Set current value as the new *pbest*.

Step 3: Choose particle with the best fitness value of all the particles as the *gbest*.

Step 4: for each particle, calculate particle velocity according to the equation

*V[]=c1*rand()*(pbest[]-present[])+c2*rand()*(gbest[]-present[])*

Where present[]=present[]+v[]

V[] is the particle velocity, *present[]* is the current particle(solution),

rand() is random functions in the range [0,1].

c1,c2 are learning factors=0.5.

step 5: particle velocities on each dimension are clamped to a maximum velocity *Vmax*. If the sum of acceleration would cause the velocity on that dimension to exceed *Vmax* (specified by user), the velocity on the dimension is limited to *Vmax*.

Step 6: Terminate if maximum of iterations is reached. Else, goto Step2

The original PSO was designed for a continuous solution space. In original PSO we can't modify the position representation, particle velocity, and particle movement. Then another heuristic is made to modify the above parameters, called Hybrid PSO. So they work better with combinational optimization problems. This Hybrid PSO was designed for a discrete solution space.

VIII. CONCLUSION :

Flexibility is unavoidable in manufacturing sector now a days, which can be achieved by Flexible Manufacturing System (FMS). Many researchers concentrated on scheduling of FMS only but few of them focused on problems of scheduling of AGV's as well as machines. This paper introduces most important non traditional optimization algorithms to integrate both scheduling of AGV'S and Machines simultaneously. While they are difficult to solve, parallel scheduling problems are among the most important because they impact the ability of manufacturers to meet customer demands and make a profit. They also impact the ability of autonomous systems to optimize their operations, the deployment of intelligent systems, and the optimizations of communications systems. For this reason, operations research analysts and engineers will continue this pursuit well in the next coming centuries.

REFERENCES:

[1]. Hymavathi Madivada and C.S.P. Rao. A Review Of Non Traditonal Algorithm for Job Shop Scheduling

[2]. K. V. Subbaiah, M. Nageswara Rao and K. Narayana Rao. Scheduling of AGVs and machines in FMS with makespan criteria using sheep flock heredity algorithm.

[3]. N.Jawahar, P.Aravindan, S.G.Ponnambalam and R.K.Suresh. AGV Schedule Integrated with Production In Flexible Manufacturing Systems.

[4]. M.V. Satish Kumar, KITS, Singapur, A.P, INDIA C.S.P Rao, NIT, Warangal, A.P, INDIA G.Rangajanardhan, JNTU, Kakinada, A.P, INDIA. A Hybrid Differential Evolution Approach for Simultaneous Scheduling Problems in a Flexible Manufacturing Environment.

[5]. Paul Pandian, P. Saravana Sankar, S.G. Ponnambalam and M. Victor Raj. Scheduling of Automated Guided Vehicle and Flexible Jobshop using Jumping Genes Genetic Algorithm.

[6]. Pramot Srinoi, A/Prof. Ebrahim Shayan and Dr. Fatemeh Ghotb. Scheduling of Flexible Manufacturing Systems Using Fuzzy Logic.

[7]. S. V. Kamble & K. S. Kadam. A Particle Swarm Optimization –Based Heuristic for Scheduling in FMS Review.

[8]. M. V. Satish Kumar, Ranga Janardhana & C. S. P. Rao. Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing.

[9]. B. Siva Prasad Reddy and C.S.P. Rao. Simultaneous Scheduling of Machines and Material Handling System in an FMS.

[10]. J. Jerald · P. Asokan · R. Saravanan · A. Delphin Carolina Rani. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm.

[11]. Moacir Godinho Filho, Clarissa Fullin Barco and Roberto Fernandes Tavares Neto. Using Genetic Algorithms to solve scheduling problems on flexible manufacturing systems (FMS): a literature survey, classification and analysis.

[12]. Muhammad Hafidz Fazli, Md Fauadi and Tomohiro Murata. Makespan Minimization of Machines and Automated Guided Vehicles Schedule Using Binary Particle Swarm Optimization.