RESEARCH ARTICLE OPEN ACCESS

A Review On Job Shop Scheduling Using Non-Conventional Optimization Algorithm

K.Mallikarjuna*, Venkatesh.G**, Somanath.B***
*(Ass..Prof, Dept of M E,Ballari Institute of Tech and Management, Bellary, Karnataka, India,)
** (Dept of M E,Ballari Institute of Tech and Management, Bellary, Karnataka, India)
*** (Dept of M E,Ballari Institute of Tech and Management, Bellary, Karnataka, India)

ABSTRACT
A great deal of research has been focused on solving job shop scheduling problem (∫J), over the last four decades, resulting in a wide variety of approaches. Recently much effort has been concentrated on hybrid methods to solve ∫J, as a single technique cannot solve this stubborn problem. As a result much effort has recently been concentrated on techniques that lead to combinatorial optimization methods and a meta-strategy which guides the search out of local optima. In this paper, authors seek to assess the work done in the job-shop domain by providing a review of many of the techniques used. It is established that Non- conventional optimization methods should be considered complementary rather than competitive. In addition, this work suggests guide-lines on features that should incorporated to create a good ∫J system. Finally, the possible direction for future work is highlighted so that current barriers within ∫J may be surmounted as researchers approach in the 21st century.
Keywords - Exact algorithm, job shop, non conventional algorithms, scheduling, review

I. Introduction

Problems encountered in fields like scheduling, assignment, vehicle routing are mostly NP hard. These problems need efficient solution procedures. If confronted with an NP-hard problem, one may have three ways to go: one chooses to apply an enumerative method that yields an optimum solution, or apply an approximation algorithm that runs in polynomial time, or one resorts to some type of heuristic technique without any a priori guarantee for quality of solution and time of computing (Aarts & Lenstra, 2003). Research in scheduling theory has evolved over the past four decades and has been the subject of much significant literature with techniques ranging from unrefined dispatching rules to highly sophisticated parallel branch and bound algorithms and bottleneck based heuristics. Not surprisingly, approaches have been formulated from a diverse spectrum of researchers ranging from management scientists to production workers. However with the advent of new methodologies, such as neural networks and evolutionary computation, researchers from fields such as biology, genetics and neurophysiology have also become regular contributors to scheduling theory emphasising the multidisciplinary nature of this field.

One of the most popular models in scheduling theory is that of the job-shop, as it is considered to be a good representation of the general domain and has earned a reputation for being notoriously difficult to solve. It is probably the most studied and well developed model in deterministic scheduling theory, serving as a comparative test-bed for different solution techniques,old and new and as it is also strongly motivated by practical requirements it is clearly worth understanding.

The evolution of optimization techniques has been mainly attributed to the increase in complexity of problems encountered two branches of heuristics exist: constructive and improvement (Onwubolu and Mutingi 1999). Constructive methods are usually problem dependent (Cambell et al. 1970, Nawaz et al. 1983). Improvement methods are those involving population-based heuristics which usually follow a naturally occurring paradigm. Many approximate methods have been developed to overcome the limitations of exact enumeration techniques. These approximate approaches include genetic algorithms (GA), tabu search (TS), differential evolution algorithm (DE) neural networks (NN), simulated annealing (SA) and particle swamp optimization (PSO).

Meta-heuristic techniques are the most recent development in approximate search methods for solving complex optimisation problems (Osman and Kelly 1996a). ∫J meta-heuristics are based on the neighbourhood strategies developed by Grabowski et al. (1986, 1988), Matsuo et al. (1988), Van Laarhooven et al. (1992) and Nowicki and Smutnicki (1996). Vaessens et al. (1995) present a template that captures most of the schemes proposed and they suggest that multi-level local search methods merit more investigation. Pirlot (1996) indicates that few serious comparative studies have been performed with regard to meta-solvers such as Simulated Annealing (SA), Tabu Search (TS) and Genetic Algorithms (GAs) and from his analysis GAs appear to be the weakest of these three methods both empirically and analytically. In a recent work Mattfeld et al. (1998) analyse the structure of the fitness landscape of ʃJ with respect to how it appears for an adaptive search heuristic. They indicate that adaptive search heuristics are suitable search techniques for ʃJ, all that is required is an effective navigation tool.

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:

- Minimize the make Span for different feasibility of job sequence.
- Minimize the waiting time of job

The objectives considered under the minimizing the makespan are,
(a) Minimize machine idle time
(b) Minimize the in process inventory costs
(c) Finish each job as soon as possible
The objectives considered under the minimizing the waiting time are,
(a) Minimize the cost due to not meeting the due dates
(b) Minimize the total tardiness
(c) Minimize the number of late jobs

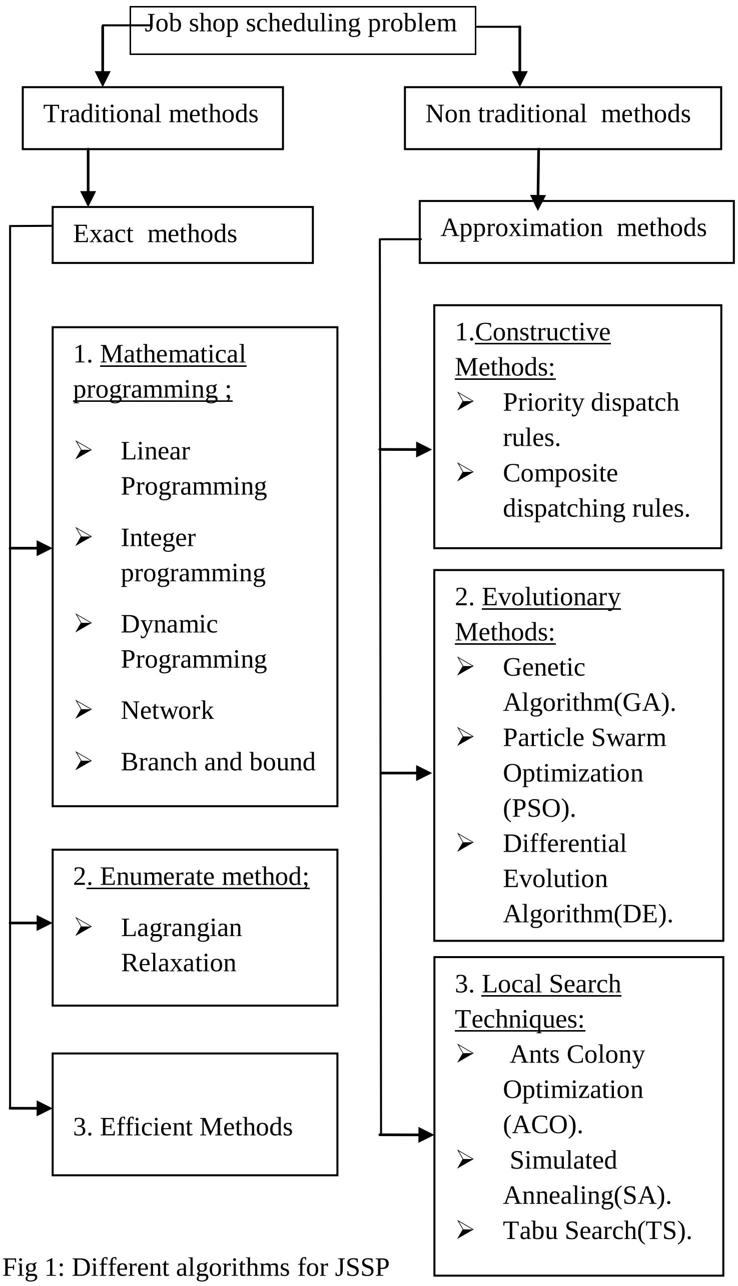

Fig 1: Different algorithms for JSSP

III. Literature review on JSSP scheduling

Many researchers have been focusing on scheduling during the last few decades. A number of approaches have been developed and employed for solving various problems of Job Shop Scheduling considering various objectives. The following table discuss the Review on Job Shop Scheduling using non traditional optimization techniques.

Table.1: Review on Job Shop Scheduling using Non Traditional Optimization Techniques

SI.NO	METHOD	AUTHOR 1	AUTHOR 2
1.	Tabu search algorithm	Fred Glover (1977, 1986) E.Nowicki (2005) Dipak Laha (2008) Sumanta Basu (2008) Wassim Jaziri	Rafael Martí (2004,2006) C.Smutnicki (2005) Uday Kumar C (2008) Diptesh Ghosh (2008)
2.	Differential evolution algorithm	Warisa Wisittipanich (2011) Donald Davendra Vanita G.Tonge (2012) Zuzana Cickova (2010)	Voratas Kachitvichyanukul(2011) Godfrey Onwubolu Prof.P.S.Kulkarni (2012) Stanislav Stevo (2010)
3.	Genetic algorithm	Goldberg D.E (1989) Hameshbabu Nanvala Dirk C. Mattfeld (2004) Jason Chao-Hsien Pan (2009)	Christian Bierwirth (2004) Han-Chiang Huang (2009)
4.	Simulated Annealing	Reeves C.R (1993) T.Yamada (1995) Aarts, B. J. M (1996) Kolonko M (1998) Peter J.M	R.Nakano (1996) Emile H.L
5.	Particle swarm optimization	Tsung-Lieh Lin D.Y.Sha (2006) Deming Lei (2008) Hsing-Hung Lin (2009) Guohui zhang (2009)	Zhiming Wu(2005) Weijun Xia(2005) Xingsheng Gu(2008)
6.	Ant colony optimization	Colorni et al (1995,1996) S.Goss, S. Aron J.-L. Colorni, M. Dorigo et Betul Yagmahan	Deneubourg et J.-M. Pasteels V.Maniezzo (1991)
7.	Artificial immune system	U.Aickelin Bagheri Mahdi Mobini	E Burke Zandieh Zahra Mobini
8.	Sheep Flock Heredity Algorithm	S.Gobinath Koichi Nara	Prof.C.Arumugam Hyunchul Kim

IV. Scheduling techniques

There are number of optimization and approximation techniques are used for scheduling of job shop scheduling problem. The techniques are generally,

- Conventional techniques Conventional techniques are also called as optimization techniques. These techniques are slow and guarantee of global convergence as long as problems are small. Mathematical programming (Linear Programming, Integer programming, Goal Programming, Dynamic Programming, Transportation, Network, Branch-and-Bound, Cutting Plane / Column Generation Method, Mixed Integer Linear programming, Surrogate Duality), Enumerate Procedure Decomposition (Lagrangian Relaxation) and Efficient Methods.
- Non conventional techniques Non conventional techniques are also called as approximation methods. These methods are very fast but they do not guarantee for optimal solutions. Constructive Methods(priority dispatch rules, composite dispatching rules), Insertion Algorithms (Bottleneck based heuristics, Shifting Bottleneck Procedure(SBP)), Evolutionary Programs(Genetic Algorithm, Particle Swarm Optimization), Local Search Techniques(Ants Colony Optimization, Simulated Annealing, adaptive Search, Tabu Search, problem Space Methods like Problem & Heuristic Space and GRASP), Iterative Methods((Artificial Intelligence Techniques, Expert Systems, Artificial Neural Network), Heuristics Procedure, Beam-Search, and Hybrid Techniques.

V. Meta-heuristic procedures

It is possible to classify meta-heuristics in many ways. Different view points differentiate the classifications. Blum and Roli (2003) classified meta-heuristics based on their diverse aspects: nature-inspired (e.g. GA, ACO) vs. non-nature inspired (e.g. TS); population-based (e.g. GA) vs. single point search (also called trajectory methods, e.g. TS); dynamic (i.e. guided local search) vs. static objective function; one vs. various neighborhood functions (i.e. variable neighborhood search); memory usage vs. memory-less methods. A classification of meta-heuristics is given in

the Table 5.1 in which “A” represents the adaptive memory property, “M” represents the memory-less property, “N” represents employing a special neighborhood, “S” represents random sampling, “1” represents iterating-based approach, and “P” represents a population-based approach. Population based approaches, also referred to as evolutionary methods, manipulate a set of solutions rather than one solution at a stage.

Meta-heuristic	Classification
Tabu-Search	A/N/1-P
Simulated Annealing	M/S-N/1
GA	M/S-N/P
ACO	M/S-N/P
GRASP	M/S-N/1
PSO	M/S-N/P

Table 5.1 - Classification of Meta-heuristics (modified from Glover, 1997)

Almost all meta-heuristic procedures require a representation of solutions, a cost function, a neighborhood function, an efficient method of exploring a neighborhood, all of which can be obtained easily for most problems (Aarts & Lenstra, 2003). It is important to mention that a successful implementation of a meta-heuristic procedure depends on how well it is modified for the problem instance at hand.

5.1 Tabu Search (TS)

TS can be considered as a generalization of iterative improvements like SA. It is regarded as an adaptive procedure having the ability to use many methods, such as linear programming algorithms and specialized heuristics, which it guides to overcome the limitations of local optimality (Glover, 1989).

TS applies restrictions to guide the search to diverse regions. These restrictions are in relation to memory structures that can be thought of as intelligent qualifications. Intelligence needs adaptive memory and responsive exploration (Glover & Laguna, 1997). For example, while climbing a mountain one remembers (adaptive memory) attributes of paths s/he has traveled and makes strategic choices (responsive exploration) on the way to peak or descent. TS also uses responsive exploration because a bad strategic decision may give more information than a good random one to come up with quality solutions. TS has memory property that distinguishes it from other search designs. It has adaptive memory that is also different from rigid memory used by branch and bound strategies. Memory in TS has four dimensions: quality, recency, frequency, and influence. A basic tabu search algorithm for a maximization problem is illustrated in Figure 5.1

```
algorithm Tabu search
begin
    T:= [ ];
    s:=initial solution;
    s*:=s
    repeat
        find the best admissible s' є N(s);
        if f(s') > f(s*) then s*:=s'
        s:=s';
        update tabu list T;
    until stopping criterion:
end;
```

Figure 5.1 – A basic tabu search algorithm

where T is a tabu list and *N(s)* is the set of neighbourhood solutions. A generic flowchart of TS algorithm can be given as follows in Figure 5.2:

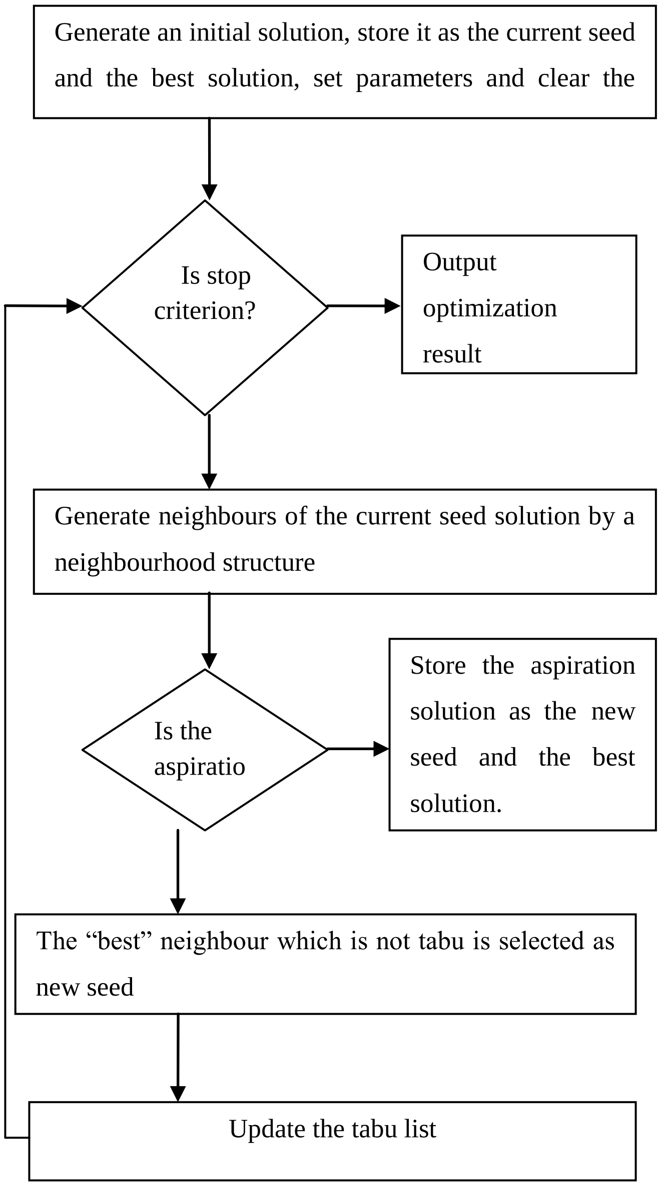

Figure 5.2 - Generic flowchart of TS algorithm

(Zhang et al. 2007)

5.2 Simulated Annealing (SA)

SA is a randomized algorithm that tries to avoid being trapped in local optimum solution by assigning probabilities to deteriorating moves. In SA a threshold value is chosen. The increase in cost of two moves is compared with that threshold value. If the difference is less than the threshold value, then the new solution is chosen. A high threshold value may be chosen to explore various parts of solution space while a low threshold value may be chosen to guide the search towards good solution values. The threshold value is redefined in each iteration to enable both diversification and intensification. Starting with high threshold values and then decreasing the value may result in finding good solutions. SA uses threshold as a random variable. In other words SA uses expected value of threshold. In a maximization problem acceptance probability of a solution is defined as follows:

$$IP\{s'\} = \begin{cases} 1 & f(s') \geq f(s) \\ \exp\left(\dfrac{f(s') - f(s)}{Ck}\right) & f(s') < f(s) \end{cases}$$

where *ck* is the temperature that gives the expected value of the threshold. A generic SA algorithm for a maximization problem is given in Figure 5.3 below:

algorithm Simulated annealing

begin

s:= initial solution

k:=1;

repeat

generate an s’ ϵ N(s);

if f(s’) ≥ f(s) **then** s:=s’

else

if $\exp\left(\dfrac{f(s')-f(s)}{Ck}\right) >$ random[0,1) **then** s:=s’;

k:=k+1;

until stop criterion:

end;

Figure 5.3 – A simulated annealing algorithm

The cooling schedule is important in SA. Temperature values (Ck) are specified according to the cooling schedule. In general, the cooling schedule’s temperature is kept constant for a number of iterations before it is decreased.

5.3 Genetic Algorithms (GAs)

GAs are used to create new generation of solutions among trial solutions in a population. In a GA, a “fitness function” is utilized and hence a quantitative study is performed. The fitness function evaluates candidate solutions, determines their weaknesses and deletes them if they are not expected ones. After this step, the reproduction among the candidates occurs and new solutions are obtained and compared using the fitness function again. The same process keeps repeating for number of generations.

With the above description in mind, Figure 5.4 shows a general scheme of using GA for minimization problems. The initial step is to determine P_0, the first population of solutions. Using the fitness function, improvements are made to the initial population of solutions. Afterwards, the algorithm enters into a loop in which crossover and mutation operations are performed until a stopping criterion is met. A typical stopping criterion is to perform all the steps for a fixed number of generations.

Begin

P_0 := set of N solutions;

/*Mutation*/

replace each s ϵ P_0 by Iterative_Improvement(s);

t :=1;

repeat

Select Pt ⊆ Pt-1;

/* Recombination */

extend Pt by adding offspring;

/* Mutation */

replace each s ϵ P_t by Iterative_Improvement(s) ;

t :=t+1;

until stop criterion;

end;

Figure 5.4 - A genetic local search algorithm for a minimization problem (Michiels et.al.,2003)

GAs have many application areas in Aerospace Engineering, Systems Engineering, Materials Engineering, Routing, Scheduling, Robotics, Biology, Chemistry, etc.

5.4 Ant Colony Optimization (ACO)

ACO is another branch of meta-heuristics that is used to solve complex problems in a reasonable amount of time. In Figure 5.5, a general type of ant colony optimization is given.

```
procedure ACO_Meta-heuristic
    while (not_termination)
    generate Solutions ()
    pheromone Update ()
    daemon Actions ()
    end while
end procedure
```

Figure 5.5 - A general ant colony optimization procedure

As seen from the general algorithm, a set of initial solutions should be generated in each turn of the while loop, then the pheromone levels should be updated and actions should be taken. When the termination criterion is reached, the procedure ends. This algorithm can be modified to fit the needs of the specific problem.

5.5 Greedy Randomized Adaptive Search Procedure (GRASP)

GRASP is another meta-heuristic method used for solving combinatorial optimization problems. Figure 5.6 demonrates how GRASP works for a minimization problem.

```
procedure GRASP
    while (termination condition not met) do
        S ← Construct Greedy Randomized Solution
        ˆS ← Local Search(S)
        If f (ˆS) < f (Sbest) then
            Sbest ← ˆS
        end-if
    end-while
    return Sbest
end-procedure
```

Figure 5.6 - High level pseudo-code for GRASP

This algorithm is composed of two main phases: a construction phase and a local search phase. In the construction phase, there is a greedy function which maintains the rankings of partial solutions. This step is very important because it affects the time efficiency of the algorithm. After ranking the partial solutions, some of the best ones are stored in a restricted candidate list (RCL). In the local search phase, as shown in Figure 5.6, a comparison is done to differentiate the quality of solutions. The algorithm terminates after a fixed number of iterations.

Fogel & Michalewicz (2000) provide a GRASP application to solve a TSP with 70 cities. They randomly select a city to begin the tour and then add the other 69 cities one at a time to the tour. After constructing an initial solution, they run the algorithm and evaluate 2415 different solutions. In such big TSP problems, GRASP seems to find good solutions in reasonable amounts of time.

5.6 Particle Swarm Optimization (PSO)

PSO is inspired from the collective behaviors of animals. In this section, we will present a sample PSO algorithm to demonstrate how it works and talk about the kinds of problems it is applied to.

There are two key definitions in using PSO algorithms that have been defined in Section 4 earlier: position and velocity. The position and velocity of particle i at time t are represented by x_i (t) and v_i (t) respectively. The position and velocity of a particle changes based on the following equations:

$$\mathbf{x}_i (t) = \mathbf{x}_i (t-1) + \mathbf{v}_i (t-1) \qquad (1)$$

equivalently, $\mathbf{x}_i$ (t) can be represented as a function of the previous position, previous velocity, $\mathbf{p}_i$, and $\mathbf{p}_g$ where, $\mathbf{p}_i$ is the local best position of particle i, and $\mathbf{p}_g$ is the neighborhood best position.

$$\mathbf{x}_i (t) = f(\mathbf{x}_i (t-1), \mathbf{v}_i (t-1), \mathbf{p}_i, \mathbf{p}_g) \qquad (2)$$

$$\mathbf{v}_i (t) = \mathbf{v}_i (t-1) + \Phi_1 (\mathbf{p}_i - \mathbf{x}_i (t-1)) + \Phi_2 (\mathbf{p}_g - \mathbf{x}_i (t-1)) \qquad (3)$$

Equation (8) shows the velocity of particle i.
Where, Φ1 and Φ2 are randomly chosen parameters.
Φ1 represents the individual experience and Φ2 represents the social communication. In figure 5.7 the PSO algorithm is given for n particles:

```
    For i = 1 to n :
        If F(x_i) > F(p_i) then :
            For d = 1, . . . , D :
                p_id = k_id // p_id is thus the best
                found individual
            end d
        end if
        g = i
        For j =index of the neighbours :
            If F(p_j) > F(p_g) then:
                g = j // g is the best individual
                in the neighbour hood
            end if
        end j
        For d = 1, . . . , D :
                v_id(t) = v_id(t − 1) + Φ1 (p_id − x_id
                    (t − 1)) + Φ2 (p_gd −x_id (t − 1))
                v_id є (−V_max_ + V_max)
                v_id(t) = x_id(t − 1) + v_id(t)
        end d
    end i
end
```

Figure 5.7 - The PSO algorithm for n particles (Dréo et al., 2006)

As seen in Figure 5.7, this algorithm can be used in multiple dimensions. This PSO algorithm can applied to many problems in the real life such as the TSP, the vehicle routing problem, the flow shop scheduling problem, etc. However, it is more commonly used in training of artificial neural networks.

VI. CONCLUSIONS

Since job shop scheduling problems fall into the class of NP-complete problems, they are among the most difficult to formulate and solve. Some optimization problems (including various combinatorial optimization problems) are sufficiently complex that it may not be possible to solve for an optimal solution with the kinds of exact algorithms. In such cases, heuristic methods are commonly used to search for a good (but not necessarily optimal) feasible solution. Several metaheuristics are available that provide a general structure and strategy guidelines for designing a specific heuristic method to fit a particular problem. A key feature of these metaheuristics procedures is their ability to escape from local optima and perform a robust search of a feasible region

This paper introduces the most prominent types of non-conventional type algorithms or meteheuristics.Tabu search moves from current trial solution to the best neighboring trial solution at each iteration, much like a local improvement procedure, except that it allows a non improving move when an improving move is not available. It then incorporates short-term memory of the past search to encourage moiving toward new parts of the feasible region rather than cycling back to previously considered solutions. In addition, it may employ intensification and diversification strategies based on long-term memory to focus the search on promising continuious.

The following are the advantages of non-traditional techniques over the traditional techniques:

- The non-traditional techniques yield a global optimal solution.
- The techniques use a population of points during search.
- Initial populations are generated randomly which enable to explore the search space.
- The techniques efficiently explore the new combinations with available knowledge to find a new generation.
- The objective functions are used rather than their derivatives.

REFERENCES

Journal Papers:

[1] E.Nowicki and C.Smutnicki, "An Advanced Tabu Search Algorithm for the Jobshop Problem" *Journal of Scheduling **8:** 145–159, 2005. Springer Science + Business Media, Inc. Printed in the Netherlands.*

[2] Dipak Laha and Uday Kumar Chakraborty, **"**An efficient hybrid heuristic for makespan minimization in permutation flow shop scheduling**"** *Int J Adv Manuf Technol (2009) 44:559–569 DOI 10.1007/s00170-008 1845-2.*

[3] Vanita G.Tonge & Prof.P.S.Kulkarni, "Permutation flowshop scheduling problem using DE" *International Journal of Societal Applications of Computer Science Vol 1 Issue 1 November 2012,* Computer Technology, RCERT, chandrpur.

[4] Dirk C. Mattfeld, Christian Bierwirth "An efficient genetic algorithm for job shop scheduling with tardiness objectives*" European Journal of Operational Research, Volume 155, Issue 3, 16 June 2004, Pages 616-630*

[5] Kolonko, M. (1998) Some New Results on "Simulated Annealing Applied to the Job Shop Scheduling Problem", to appear in the *European Journal of Operational Research*.

Chapters in Books:

[6] Wassim Jaziri(Ed), The book "*Local Search Techniques: Focus on Tabu Search*", Published by In-Teh branch of I-Tech Education And Publishing KG, Vienna, Austria.

Theses:

[7] Fred Glover, "*Tabu search*" Leeds School of Business, University of Colorado, Campus Box 419, Boulder, CO 80309;

[8] Rafael Martí, "*Tabu search*" Dpto. De Estadística e Investigación Operativa, Universidad De Valencia, Dr. Moliner 50, 46100 Burjassot (Valencia) Spain.

[9] A. Hassam and Z. Sari, "*Meta-heuristics for real time routing selection in Flexible Manufacturing Systems (FMS)*" Tlemcen Automatic Laboratory, Abou bekr Belkaïd University of Tlemcen, PoBox 230, Tlemcen 13000, Algeria.

[10] Takeshi Yamada, "*Metaheuristics for Jobshop and Flowshop Scheduling Problems*" Kyoto university kyoto, japan November, 2003

[11] Warisa Wisittipanich and Voratas Kachitvichyanukul, "*Differential Evolution Algorithm for Job Shop Scheduling Problem*" IEMS Vol. 10, No. 3, pp. 203-208, September 2011, Industrial and Manufacturing Engineering School of Engineering and Technology,

[12] Asian Institute of Technology P.O. Box 4, Klong Luang, Pathumthani 12120, Thailand.

[13] Zuzana Cickova and Stanislav Stevo, "*Flow Shop Scheduling using Differential Evolution*" Management Information Systems Vol. 5 (2010), No. 2, pp. 008-013.

[14] Donald Davendra, "*Flow shop scheduling using enhanced differential evolution algorithm*" Faculty of Applied Informatics Tomas Bata University Zlin, Czech Republic.

[15] Godfrey Onwubolu, "*Flow shop scheduling using enhanced differential evolution algorithm*" School of Engineering and Physics University of the South Pacific Suva, Fiji Islands.

[16] Goldberg, D., E. 1989. "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addison Wesley Publishing Company, Inc.

[17] Jason Chao-Hsien Pan, Han-Chiang Huang "*A hybrid genetic algorithm for no-wait job shop scheduling problems*" Expert Systems with Applications, Volume 36, Issue 3, Part 2, April 2009, Pages 5800-5806

[18] Aarts, B. J. M. (1996) "*A Parallel Local Search Algorithm for the Job Shop Scheduling Problem",* Masters Thesis, Department of Mathematics & Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, April.

[19] Reeves C.R., 1993. "*Modern heuristic techniques for combinatorial problems"*. Orient Longman, Oxford.

[20] Guohui Zhang, "*An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem*" Computers & Industrial Engineering, Volume 56, Issue 4, May 2009, Pages 1309-1318

[21] D.Y. Sha, "A hybrid particle swarm optimization for job shop scheduling problem", Computers & Industrial Engineering, Volume 51, Issue 4, December 2006, Pages 791-808

[22] Colorni, M. Dorigo et V. Maniezzo, "*Distributed Optimization by Ant Colonie"s*, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[23] S. Goss, S. Aron, J.-L. Deneubourg et J.-M. Pasteels, "*The self-organized exploratory pattern of the Argentine ant*", Naturwissenschaften, volume 76, pages 579-581, 1989

Proceedings Papers:

[24] C.S.P. Rao, "*A review on non traditional algorithms for job shop Scheduling", Professor*, Department of Mechanical Engineering, National Institute of Technology –Warangal, Warangal – 506004, India.

[25] Hymavathi Madivada, "*A review on non traditional algorithms for job shop Scheduling*", Research Scholar, Department of Mechanical Engineering, National Institute of Technology – Warangal, Warangal – 506004, India.

[26] Anant Singh Jain & Sheik Meeran, "*A state-of-the-art review of job-shop scheduling techniques*",
Department of Applied Physics, Electronic and Mechanical Engineering University of Dundee, Dundee, Scotland, UK, DD1 4HN