International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 3, No. 2, April 2014

Research Paper

OPTIMIZING OF MAKESPAN IN JOB SHOP SCHEDULING PROBLEM: A COMBINED NEW APPROACH

T Varun Kumar[1]* and B Ganesh Babu[1]

*Corresponding Author: T Varun Kumar, ✉ ercrazyvarun @ gmail.com

Scheduling is the most important issue to be solved in the real time environment. One such emerging problem in the scheduling is the job shop scheduling problem, applied in various fields of engineering. The Job Shop Scheduling Problem (JSP) is one of the hardest combinatorial optimization problems. The performance of schedules released to a shop floor may greatly be affected by unexpected disruptions. The main objective of the JSP is to find a schedule of operations that can minimize the maximum completion time (called makespan) that is the completed time of carrying total operations out in the schedule of n jobs and m machines. Recently many works have been reported to reduce the makespan time in JSP. No Scheduling technique has guaranteed optimality. This paper aims at providing a well optimized scheduling technique; minimize the makespan, process time and the number of iterations. This paper proposes a Genetic algorithm with Unordered Subsequence Exchange cross-over (USXX) and Hybrid approach called a PSO-GA. This algorithm is a stochastic procedure that uses a population of solution, called particles, which move in search space. Using the special cross over technique USXX the most of the benchmark results are compared and obtain the results near to optimal value of the benchmark problems. The hybrid approach produced the better computational time compare to the GA. This approach is also applied to maximize net present value. Multiple runs of both algorithms are performed and the results are averaged in order to achieve meaningful comparisons. These finding are very promising and demonstrate the applicability of this hybrid approach for this existing problem.

Keywords: Job Shop Scheduling Problem (JSP), Makespan, Genetic algorithm, Particle swarm optimization

INTRODUCTON

The JOB-SHOP scheduling problem (JSSP) is a very important practical problem in both fields of production management and combinatorial optimization. Very important to bring out the efficient methods for solving the

[1] Department of Mechanical Engg., Roever College of Engg &Technology, Perambalur, India.

JSSP have significant effects on profitability and product quality. The JSSP has drawn the attention of researchers for last three decades, but because scheduling problem varies widely according to specific production tasks and most of them are strongly nondeterministic polynomial time hard problems, some test problems of moderate size are still unsolved. In fact, only small-size instances of the problems can be solved within a reasonable computational time by exact optimization algorithms such as branch and bound (Heilmann, 2003; and Akkan and Karabati, 2004) and dynamic programming (Potts and Van Wassonhove, 1987; and Lorigeon, 2002), including the benchmark instance 10 x 10 of Fisher and Thompson, which was proposed in 1963 and only solved 20 years later. Problems of dimension 15 x 15 are still considered to be beyond the reach of today's exact methods. These methods mainly include dispatching priority rules (Jensen *et al.*, 1995; Klein, 2000; and Canbolat and Gundogar, 2004), shifting bottleneck approach (Pezzella and Merelli, 2000; and Huang and Yin, 2004), Lagrangian relaxation (Kaskavelis and Caramanis, 1998; and Chen and Luh, 2003), and tabu search (Ponnambalam *et al.*, 2000; Watson *et al.*, 2003; and Geyik and Cedimoglu, 2004) and have made considerable achievement. In recently much attention has been devoted to metaheuristics with the emergence of new techniques from the field of artificial intelligence such as Genetic Algorithm (GA) (Hajri *et al.*, 2000; Gang and Wu, 2004; Amirthagadeswaran and Arunachalam, 2006; and Liu *et al.*, 2006), simulated annealing (Low *et al.*, 2004), ant colony optimization, Particle Swarm Optimization (PSO), artificial neural network, Artificial Immune System (AIS). These metaheuristics can be regarded as problem independent approaches and are well suited to solve complex problems that may be difficult to solve by traditional technique. Moreover, they are capable of producing high-quality solutions with reasonable computational effort. Among the metaheuristics algorithms, GA has been used with increasing frequency to address scheduling problems and may not remain to have much room for improvement. However, the use of PSO and GA for the solution of scheduling problems has been scrace, specifically the PSO. In addition, many research results of the JSSP show that it is difficult to obtain a good enough solution only by single search scheme. Motivated by these perspectives, we propose a novel hybrid intelligent algorithm for the JSSP based on PSO and GA in this paper. Both PSO and GA are evolutionary computation techniques based on swarm intelligence. They exhibit implicit parallelism and contain certain redundancy and historical information of past solutions. Therefore, the proposed hybrid algorithm also effectively exploits the capabilities of distributed and parallel computing of swarm intelligence approaches.

OVERVIEW OF JSSP

Job shop Scheduling Problem is a well known constraints satisfaction problem in the field. In a JSP we have a finite set of N jobs, where $N = \{1, \ldots, n\}$, that have to process in a set M of machines, where $M = \{1, \ldots, m\}$. Each and every job should divide into a series of m operations O_{ik}, where subscript k indicates the machine M_k on which the operation has to be processed. The technological order of machines, i.e., process routing for job is

predefined. We consider the static job shop scheduling case every job is a chain of operations and every operation has to be processed on a given machine for a given time. The task is to find the completion time of the very last operation is minimal.

Table 1: The Notations of Conceptual Model

Notations	Description
M	Number of machines
N	Number of jobs
$O_{i,j}$	j^{th} operation of job i
$T_{i,j}$	Processing time of $O_{i,j}$
$F_{i,j}$	Finish time of $O_{i,j}$
C_i	Completion time of job i
N_i	Number of operation of job i
C_{max}	Maximum completion time of all job
O_{ik}	The start time of j on machine k

Minimization of makespan:

$$f(x) = C_{\max}(O_{jk} + T_{ij}) \quad ...(1)$$

w.r.t

$$F_{ij} \le T_{ij+1}\ i = 1, \ldots, n_i \quad ...(2)$$

$$Y_{mij} \le a_{mij}\ m = 1, \ldots, M \quad ...(3)$$

Constraint (1) specifies the makespan completion time and (2) indicates the Precedence Constraints among the operation is executed. Constraint (3) indicates that can be assigned to just one machine from among the given machines.

Example Problem

This problem contains 3 machines (m), 3 jobs (n), Process routing, processing time also given. If the operation sequence is given it will work based On the Precedence constraints and capacity constraints and satisfies the constraints of the problem also. Let we see how it works.

Permutation Representation

$J_1 = \{M_3, M_1, M_2\}$

$J_2 = \{M_2, M_3, M_1\}$

$J_3 = \{M_2, M_1, M_3\}$

Operation Sequence

$M_1 = (J_3, J_2, J_1)$

$M_2 = (J_3, J_2, J_1)$

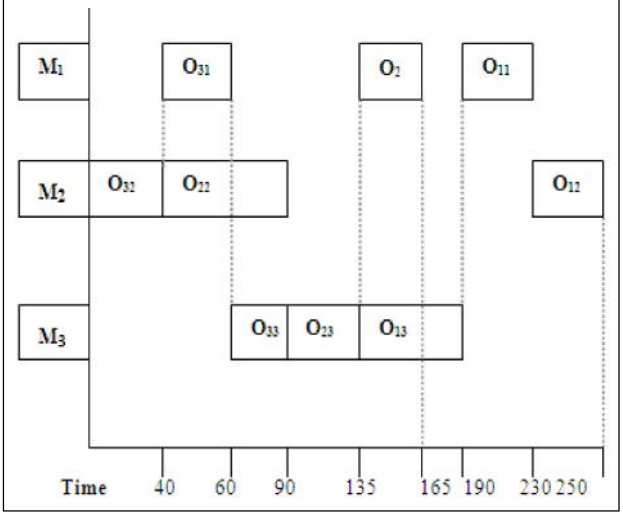
$M_3 = (J_3, J_2, J_1)$

Table 2: Opeartion Processing Times

Jobs	Machines		
	M_1	M_2	M_3
J_1	40	20	55
J_2	30	50	45
J_3	20	40	30

The above inputs schedule information and makespan time is calculated using the Gantt chart.For the above operation sequence or schedule and machine allocation or

Figure 1: Scheduled Visual Information Using Gantt Chart

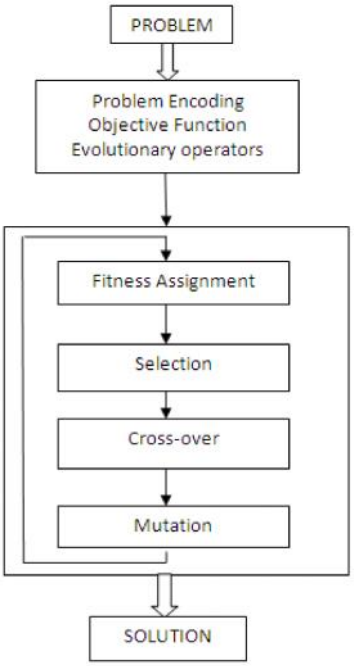


permutation, with operation time, the makespan time is 250 calculated through Gantt chart. The scheduled information in Figure 1. Clearly showed satisfies the precedence and capacity constraints.

GENETIC ALGORITHM FOR JSSP

The Genetic Algorithm is a Meta heuristic technique, which may be used to solve maximization optimization problem. The genetic algorithm works based on natural populations evolve according to the principle of natural selection that is survival of the fittest, first clearly stated by Charles Darwin in the Origin of Species. It starts with initial solution called Populations and it is filled with chromosome. Each element in chromosome is called gene. Job is represented by each gene in chromosome and the job sequence in a schedule based on the position of the gene.

Figure 2: Simple Genetic Algorithm

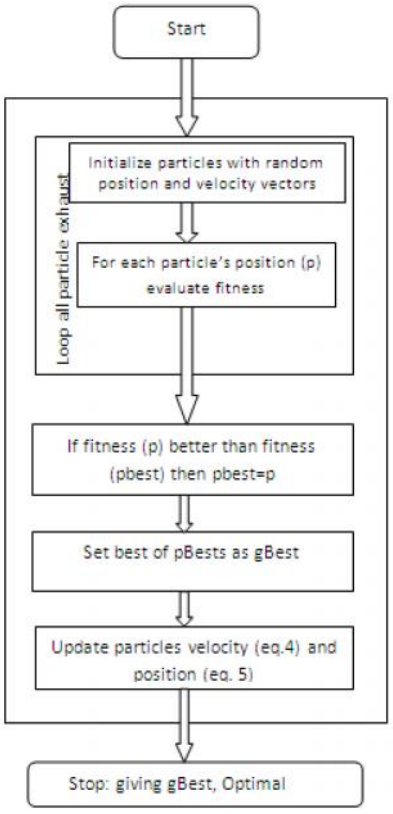


GA uses Crossover and Mutation operation to generate a new Population. A typical Genetic algorithm illustrated in Figure 2.

PARTICLE SWAM OPTIMIZATION

Particle Swarm Optimization (PSO) is a population based search algorithm inspired by bird flocking and fish schooling originally designed and introduced by kennedy and Eberhart in 1995. In contrast to evolutionary computation paradigms such as genetic algorithm, a swarm is similar to a population while a particle is similar to an individual. The fly through a multidimensional search space in which position of each particle is adjusted to its own experience and the experience of its neighbors. PSO System combines local

Figure 3: PSO Flow Diagram



search methods through self experience with global search methods through neighboring experience, attempting to balance exploration and exploitation. The general flow diagram for PSO illustrated in Figure 3.

The mathametical model for calculating velocity vector and updating position as follows,

$$V_{td} = W * V_{td} + C_1 * r_1(P_{td} - X_{td}) + C_2 * r_2(P_{td} - X_{td}) \quad ...(4)$$

$$X_{td} = x_{td} + V_{td} \quad ...(5)$$

where W is the inertial weight. It is used to control the amount of the previous velocity between the global exploration abilities of the swarm. C_1 and C_2 are two positive constants, and they represent the weight of the acceleration terms that pull each particle toward P_{td} and P_{gd} positions, r_1, r_2 are two random functions in the range [0, 1].

For Equation (4), in traditional PSO, Update of the velocity consists of the following three parts. $W*V_{td}$ is referred to as "thrust" part which represents the influence of the last velocity towards the current velocity. $C_1*r_1(P_{td} - X_{td})$ is the cognitive part,which represents the self thinking by itself. $C_2*r_1(P_{td} - X_{td})$ is the social part, which represents the cooperation among particles. The traditional PSO resolves the simple optimization concept that individuals are evolved by cooperation and competition among the individuals to accomplish a common goal. In large search process, each particle of the swarm shares the known information globally and benefits from discoveries and previous experiences of all other social group.

The value stream map for the present state is constructed as shown in Figure 3. A first design of VSM is realized according to the original data and the layout, identifying the key times of each assembly station. This design represents the starting point of improvement. Next, the map of the parts flow is shown to verify the materials movement between the stations, calculating the productive and unproductive times, stocks and metrics that will help to characterize the process and marking some targets of progress. Metrics used are DtD and LR (Equations 1 and 2).

The present state is based on the housing of the core components and the assembly of the engine is a push system. The majority of the raw materials is sourced from local industries and is stored in the engine storey department and during the assembly process the raw materials are brought from the stores and are washed, cleaned and is separated among the assembly line. Since the number of raw materials needed for the assembly process is unknown there is either an over stock.

PROPOSED SCHEME

Genetic Approach

The proposed scheme objective is to solve a job shop scheduling problem to minimize the makespan time. In order to solve a JSSP artificial intelligence technique Genetic Algorithm (GA) is used. The genetic algorithm is a probabilistic Meta heuristic technique, which is used to solve optimization problems. They are based on the genetic process of chromosome. Over many generations, natural population evolves according to the principles of natural selection that is survival of the fittest. It starts with the initial solution called population and it is filled with chromosome.

Each element in chromosome is called gene. Job is represented by each gene in chromosome and the job sequence in a schedule based on the position of the gene. In our proposed algorithm unordered subsequence exchange crossover (USXX) and shift change Mutation is used.

Objective Function

The main objective of the JSSP is to find a schedule of operations that can minimize the maximum completion time called makespan that is the completed time of carrying total operations out in the schedule for n jobs and m machines. The objective function takes the input as the number of jobs, number of operations, chromosome, operation time sequence and machine sequence of the corresponding operation sequence.

$$C_{\max} = \max_{1 \le i \le n} C_i \rightarrow \min \quad ...(6)$$

Fitness Function

The objective is to minimize the makespan, so the following fitness function is applied,

$$f(x) = \frac{1}{makespan(x)} \quad ...(7)$$

Chromosome Representation and Encoding Scheme

The searching space is created in $n \times m$ dimensions space for n jobs on m machines job shop scheduling problem. The problem solution is represented as a chromosome in Genetic Algorithm. The position of a chromosome consists of n×m dimensions and is represented with n×m real numbers. Chromosome is initialized based on jobs and machines. Then assign the rank with respect to smaller value in the chromosome and vice versa. Decoding by using the formula,

$$(R_i \bmod n) + 1 \quad ...(8)$$

where,

R_i is integer number by ranking

n, is number of jobs

From the Equation (8) proposed scheme will obtain the operation sequence. The following Table 2 shows representation of chromosome and generating the operation sequence.

Table 3: Chromosome and Operation Sequence

Chromosome	4.5	1.3	2.3	8.2	4.1	3.5
Give the Rank (R_i)	5	1	2	6	4	3
($R_i mod\ n$) + 1	3	2	3	1	2	1
Operation Sequence	O_{31}	O_{21}	O_{32}	O_{11}	O_{22}	O_{11}

From the Table 2 operation sequence defines 3rd job 1st operation, 2nd job 1st operation, 3rd job 2nd operation, 1st job 1st operation, 2nd job 2nd operation, 1st job 2nd operation. This operation Processed based on Process Routing that is Precedence constraints based. That means allocated jobs are proceed as per planned without changing the job sequence.

Algorithm Steps

Step 1: Initialize the number of chromosomes by generating *n*m* real numbers for each chromosome.

Step 2: Assign the operation time sequence and Machine sequence for selected chromosomes.

Step 3: Find the makespan value for each and every chromosome using the objective function and also find the minimum makespan value among different values.

Step 4: Select *N*/2 chromosomes using the Roulette-Wheel selection from different chromosomes.

Step 5: Applying the Unordered subsequence exchange crossover and shift change Mutation to generate the new chromosomes.

Step 6: Find the makespan values for newly generated chromosomes using the objective function.

Step 7: Choose the best chromosomes which have the minimum makespan values from newly generated and also from old chromosomes.

Step 8: Find the minimum makespan value among different chromosomes.

Step 9: Terminates if the maximum number of iteration is reached or optimal value is obtained.

Proposed Hybrid PSO-GA Algorithm

The idea of the proposed hybrid algorithm, called PSO-GA. The PSO possesses high search efficiency by combining local search and global search. By reasonably hybridizing these two methodologies, a hybrid PSO-GA algorithm model could be proposed to omit the concrete velocity-displacement updating method in traditional PSO for the JSS problem. The Proposed hybrid algorithm includes two phases: (1) the initial solutions are randomly generated as per Table 2. But assume like chromosome are now particles. (2) the PSO algorithm combined with GA algorithm is run. The General outline of the hybrid algorithm is summarized as follows:

Step 1: Generate initial population. If any of the generated schedules is infeasible, reconstruct them until all the initial solutions are feasible.

Step 2: Evaluate each particles objective value in the swarm, and compare them, then set the pbest position with a copy of particle itself and gbest position the particle with the lowest fitness in the swarm.

Step 3: If the termination criterion which is usually a sufficiently good fitness or a specified number of generations are not met, then go to step 4; otherwise go to step 7.

Step 4: Set iteration = iteration + 1

Step 5: Apply Unordered Subsequence Exchange cross-over and Shift change Mutation.

Step 6: Convert current particles to a JSSP scheduling solution; then go to step 2.

Step 7: Computational results.

Unordered Subsequence Exchange Cross-Over

Unordered subsequence exchange crossover creates a new children's even the subsequence of parent 1 is not in the same order in parent 2. The algorithm for USXX is as follows.

Step 1: Select the two parents from the different chromosomes initialized with the sequence of all operation. Say P1 and P2.

Step 2: Generate two children from P1 and P2 name it as C1 and C2.

Step 3: Select the gene from P1 and same gene selected in P2 but it should unordered position in P2.

Step 4: Crossover started from P1 that is P2 unselected genes are move to the P1 in

unselected position. So C1 is generated. Likewise to generate C2 crossover from P2 to P1.

Shift Change Mutation

Shift change mutation is implemented by shift the same job index in every place into the same direction so that new sequence is generated.

EXPERIMENTAL RESULTS

The Performance of the proposed Genetic algorithm and Hybrid PSO-GA for the JSSP is examined by using some 12 test problems taken from the OR-Library. Numerical experiments are performed in Java Language on a Workstation with Intel (R) Core (TM) i5 2.50 GHz Processor and 2 GB memory. Table 4 summarizes the results of the experiments. The content of the table include the name of the test problem (instance). The size of the problem, the Best Known Value (BKV). The value of the best solution found by using Genetic Algorithm and Hybrid PSO-GA. Table 3 summarizes the parameters of the GA and PSO-GA.

From the above tabulated results it is shown that for benchmarks LA01, LA05, LA13 the Hybrid PSO-GA yields better results than GA with unordered subsequence exchange cross-over by reducing makespan.

Table 4: Parameters of Proposed Approach

Parameter Name	Value
Swarm Size	20
number of Particles	100
C1 and C2	2.1
Crossover Probability	0.7
Mutation Probability	0.2
Number of Iterations	15000

Table 5: Comparison of Results Between the GA and PSO-GA

Instance	Size	Best Known Value	GA	Proposed PSO
FT06	6 x 6	55	55	55
LA01	10 x 5	666	**739**	**666**
LA05	10 x 5	593	**600**	**593**
LA06	15 x 5	926	926	926
LA07	15 x 5	890	890	890
LA08	15 x 5	863	863	863
LA09	15 x 5	951	951	951
LA10	15 x 5	958	958	958
LA11	20 x 5	1222	1222	1222
LA12	20 x 5	1039	1039	1039
LA13	20 x 5	1150	**1160**	**1150**
LA14	20 x 5	1292	1292	1292

CONCLUSION

in this paper, according to characteristics of static JSSP, a GA with Unordered subsequence exchange cross over and a Hybrid approach called PSO-GA is proposed to solve the JSSP. Two proposed algorithm are compared and evaluated. The best result is obtained by using Hybrid approach. In future the proposed algorithm can solve a Job Shop Scheduling problem in dynamic environment with multi-criteria objective. Other hybridization heuristic methods are to be used to solve the Job-Shop Scheduling Problem.

REFERENCES

1. Akkan C and Karabati S (2004), "The Two-Machine Flowshop Total Completion Time Problem: Improved Lower Bounds and a Branch-and-Bound Algorithm", *Eur. J. Oper. Res.*, Vol. 159, No. 2, pp. 420-429.
2. Amirthagadeswaran K S and Arunachalam V P (2006), "Improved

Solutions for Job Shop Scheduling Problems Through Genetic Algorithm with a Different Method of Schedule Deduction", *Int. J. Adv. Manuf. Technol.*, Vol. 28, Nos. 5/6, pp. 532-540.

3. Canbolat Y B and Gundogar E (2004), "Fuzzy Priority Rule for Job Shop Scheduling", *J. Intell. Manuf.*, Vol. 15, No. 4, pp. 527-533.
4. Chen H X and Luh P B (2003), "An Alternative Framework to Lagrangian Relaxation Approach for Job Shop Scheduling", *Eur. J. Oper. Res.*, Vol. 149, No. 3, pp. 499-512.
5. Gang X and Wu Z M (2004), "Deadlock-Free Scheduling Strategy for Automated Production Cell", *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, Vol. 34, No. 1, pp. 113-122.
6. Geyik F and Cedimoglu I H (2004), "The Strategies and Parameters of Tabu Search for Job-Shop Scheduling", *J. Intell. Manuf.*, Vol. 15, No. 4, pp. 439-448.
7. Hajri S, Liouane N, Hammadi S and Borne P (2000), "A Controlled Genetic Algorithm by Fuzzy Logic and Belief Functions for Job-Shop Scheduling", *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, Vol. 30, No. 5, pp. 812-818.
8. Heilmann R (2003), "A Branch-and-Bound Procedure for the Multi-Moderesource-Constrained Project Scheduling Problem with Minimum and Maximum Time Lags", *Eur. J. Oper. Res.*, Vol. 144, No. 2, pp. 348-365.
9. Huang W Q and Yin A H (2004), "An Improved Shifting Bottleneck Procedure for the Job Shop Scheduling Problem", *Comput. Oper. Res.*, Vol. 31, No. 12, pp. 2093-2110.
10. Jensen J B, Philipoom P R and Malhotra M K (1995), "Evaluation of Scheduling Rules with Commensurate Customer Priorities in Job Shops", *J. Oper. Manage.*, Vol. 13, No. 3, pp. 213-228.
11. Kaskavelis C A and Caramanis M C (1998), "Efficient Lagrangian Relaxation Algorithms for Industry Size Job-Shop Scheduling Problems", *IIE Trans.*, Vol. 30, No. 11, pp. 1085-1097.
12. Klein R (2000), "Bidirectional Planning: Improving Priority Rule-Based Heuristics for Scheduling Resource-Constrained Projects", *Eur. J. Oper. Res.*, Vol. 127, No. 3, pp. 619-638.
13. Liu T K, Tsai J T and Chou J H (2006), "Improved Genetic Algorithm for the Job-Shop Scheduling Problem", *Int. J. Adv. Manuf. Technol.*, Vol. 27, Nos. 9/10, pp. 1021-1029.
14. Lorigeon T (2002), "A Dynamic Programming Algorithm for Scheduling Jobs in a Two-Machine Open Shop with an Availability Constraint", *J. Oper. Res. Soc.*, Vol. 53, No. 11, pp. 1239-1246.
15. Low C, Yeh J Y and Huang K I (2004), "A Robust Simulated Annealing Heuristic for Flow Shop Scheduling Problems", *Int. J. Adv. Manuf. Technol.*, Vol. 23, Nos. 9/10, pp. 762-767.
16. Pezzella F and Merelli E (2000), "A Tabu Search Method Guided by Shifting Bottleneck for the Job Shop Scheduling Problem", *Eur. J. Oper. Res.*, Vol. 120, No. 2, pp. 297-310.

17. Ponnambalam S G, Aravindan P and Rajesh S V (2000), "A Tabu Search Algorithm for Job Shop Scheduling", *Int. J. Adv. Manuf. Technol.*, Vol. 16, No. 10, pp. 765-771.
18. Potts C N and Van Wassonhove L N (1987), "Dynamic Programming and Decomposition Approaches for the Single Machine Total Tardiness Problem", *Eur. J. Oper. Res.*, Vol. 32, No. 3, pp. 405-414.
19. Watson J P, Beck J C and Howe A E (2003), "Problem Difficulty for Tabu Search in Job-Shop Scheduling", *Artif. Intell.*, Vol. 143, No. 2, pp. 189-217.