International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

RESEARCHERID
THOMSON REUTERS

ISSN: 2277-9655
[Shukla *et al.*, 7(11): November, 2018] Impact Factor: 5.164
IC™ Value: 3.00 CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

SCHEDULING OF PERMUTATION FLOW SHOPS USING A NEW HYBRID ALGORITHM

Brahma Datta Shukla [*1] & Pragya Singh Tomar[2]
[*1&2]Institute of Computer Science, Vikram University, Ujjain

DOI: 10.5281/zenodo.5150017

ABSTRACT

In the current situation, modern engineering and industrial built-up units are encountering a jumble of issues in a variety of areas, including machining time, electricity, manpower, raw materials, and client restraints. One of the most important industrial behaviors, particularly in manufacturing planning, is job-shop scheduling. This study provides a new updated suggested approach of johnson's algorithm as well as the gupta's heuristic algorithm to solve the permutation flow shop sequencing problem with the goal of making the makespan as little as possible. This work is about determining the processing order of n tasks in m machines. Although, because the problem is np-hard for three or more computers, this results in a near-optimal solution to the given issue. The suggested approach is straightforward and easy to comprehend, and it is accompanied with a numerical example.

KEYWORDS: Optimal sequence, flow shop scheduling, makespan , heuristic, Johnson's technique, gupta's method.

1. INTRODUCTION

Scheduling is the process of allocating resources (such as machines) to tasks (such as jobs) in order to guarantee that these activities are completed in an acceptable period of time. Some strategies are used to pick the most suited work. Due to resource constraints, only a limited number of jobs may be run on the processor. The scheduling should be done according to whether the processor is a single processor or a multiprocessor.

The main aim is to determine the best option for running jobs on various computers or processors. The number of processors or tasks may vary from one to the next. As a result, it is a crucial duty to do.

A job shop is a workplace with a range of general-purpose workstations or equipment that may be used to complete a range of tasks.

The challenge of job shop scheduling is sometimes known as the challenge of work shop scheduling. It is a situation in which resources must be optimized. At specific periods, they are assigned to perfect tasks. State n jobs J1, J2,.... Jn, each of which is made up of a chain of processes. Various sizes, planned on m indistinguishable machines (m > 2), with a desire to reduce the makespan. At most, each machine can handle one operation at a time. Preemption of an operation is not permissible in any procedure.

Flow shop scheduling is a type of job shop scheduling in which all activities must be completed in a specific order. Flow shop scheduling issues are those in which the flow control must allow for suitable sequencing for each task and processing on a group of machines or with other resources 1,2,...,m in accordance with provided processing instructions. It is preferable to maintain a continual flow of processing jobs with the least amount of idle time and waiting time possible.

Flowshop Scheduling: Flowshop Scheduling establishes the best order for n jobs to be handled on m machines in the same order, i.e. every job must be handled in the same order on machines 1,2,...,m.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

The flowshop scheduling problem is a production challenge in which a set of n jobs must be completed on m machines using identical flow patterns. The permutation flowshop sequencing production environment exists when the sequence of jobs processing on all machines is the same. The flow-shop problems are investigated with the following assumptions:

The processing times for operations on the machines are known and fixed, and some may be nil if a task is not processed on a machine.Set-up times are included in processing times and are independent of work position in the task sequence.Each task is processed on just one machine at a time, and each machine processes only one work. The machine task activities cannot be pre-empted [4].

At time zero, a flowshop has M machines in sequence and N separate jobs ready for processing. At any one moment, each machine can only handle one work. Each work is processed in the same technological sequence on M machines that are accessible. This distinguishes it from an usual jobshop issue. The quantity of limited resources allotted to these operations determines the processing time, including setup time, of jobs executed on the computer. For I = 1, 2,..., N and j = 1, 2,..., M, the processing time of task I on machine j is indicated by tij [12].

Heuristic algorithms are more efficient and cost-effective in obtaining a realistic solution, albeit they may not always achieve the best results [12]. The flow shop scheduling problem involves processing n jobs on m machines.. The machines are arranged in a specific order. We assume that each machine processes one job at a time, with no pre-emption. Flow shop scheduling is an NP-complete problem. There are (n!) m different job schedules in general [9]. Total processing time, also known as makespan, is the amount of time it takes to complete all of the operations. When it comes to the JSP problem, some assumptions must be made.

There are a set number of operations for each job.
Each operation's processing time has been determined.
Each job has its own machine sequence.
Each job must be completed according to a pre-determined sequence of operations.
Job does not return to the same machine.
The time spent setting up is included in the processing time.
Only one job can be processed at a time by a machine.
No machine can perform more than one task at a time.
Operations must not be disrupted.
There is no mention of a release date or a deadline.

Each work should be processed through the machines in a specific order, or technological constraints, as they are also known. [10].

Johnson's algorithm is a well-known approach for solving the issue of optimally scheduling n jobs on two machines in polynomial time. When there are n tasks on three machines, the problems become NP-complete (i.e., they can't be solved in polynomial time), and the Johnson's technique can only be used in a few specific circumstances that meet certain criteria [11]. J.F. Gonçalves et al. [15] recently proposed a new neighborhood-based local search strategy for the job-shop scheduling issue, which generates schedules by decoding the chromosome given by the genetic algorithm. [16] presents a Scatter Search (SS)-based approach for solving work shop scheduling challenges. It takes into account availability constraints in a hazy job shop scheduling situation. It addresses the possibility of a machine becoming unavailable owing to maintenance, repair, or an unexpected failure. In the case of a hazy work shop scheduling situation, it reduces tardiness and earlyness. A flexible job-shop scheduling issue with no-wait constraint was sought for by S. Sundar et al. [17]. (FJSPNW). It combines the characteristics of two distinct job shop scheduling problems: flexible and no-wait scheduling. R. Zhang et al. [18] desired a hybrid differential evolution (DE) strategy for solving the job shop scheduling issue with unpredictable processing times, with the goal of reducing projected overall tardiness. An intuitionist fuzzy set Job Shop Scheduling Problem was handled by X Zhang et al. [19]. Based on Gupta's heuristics, R Kumari et al. devised a fuzzified job shop scheduling system.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

The remainder of the paper is structured as follows: Section 2: Permutation Flowshop Scheduling The difficulty of job shop scheduling is discussed in section 3. The fourth section introduces Gupta's heuristics, followed by a discussion of the suggested hybrid job shop scheduling method. The performance of the suggested strategy is examined in section 6. Finally, the study is ended in section 7.

2. PERMUTATION FLOWSHOP SCHEDULING

Permutation Flowshop Scheduling is a subset of FSPs in which the same task sequence is followed by all machines, i.e. the job processing order on each machine is the same [1].

The permutation flow shop scheduling problem (PFSP) is a production issue that involves determining the optimal sequence of tasks for machines to complete in order to minimize a specified objective function. This situation occurs in manufacturing plants when jobs (parts) are transported from machine to machine using material handling equipment with no passing permitted. The issue is severely NP-complete, and the total number of alternative schedules (sequences) is for jobs. Total flow time and makespan are critical performance indicators that contribute to quick work turnaround and reduced in-process inventory [3].

The issue of permutation flow shop scheduling is classified as NP-Hard. In the usual meaning, a scheduling issue is NP-hard if Partition (or a related issue) can be reduced to this issue using a polynomial time approach and The scheduling problem may be solved using an algorithm with pseudo polynomial time complexity [6]. It belongs to the NP (nondeterministic polynomial time) category of problems: Any given answer to L may be readily checked (in polynomial time). It's also included in the category of NP-hard problems: Any NP issue may be transformed to L via a polynomial time translation of the inputs. Although every given solution to such a problem may be verified rapidly, there is no known efficient means to find a solution in the first place; fact, the most remarkable feature of NP-complete problems is that there is no known fast solution to them. That is, as the size of the issue rises, the time necessary to solve it using any currently known method climbs rapidly.

Threads, processes, and data flows are provided access to system resources through scheduling (e.g. processor time, communications bandwidth). The necessity for most contemporary systems to do multitasking (running many processes at the same time) and multiplexing necessitates the use of a scheduling mechanism (transmit multiple flows simultaneously).
The scheduler is primarily concerned with:
The total number of processes that finish their execution per time unit is known as throughput.
We aim to keep the CPU active as much as feasible [2].

Latency, specifically
Turnaround time is the amount of time it takes for a procedure to be completed once it has been submitted. Or, to put it another way, turnaround time is the total of the time spent waiting to be remembered [2].
Response time is the time it takes for a request to be processed from the moment it is made to the time it receives its first response.

Fairness / Waiting Time - Each process receives equal CPU time (or, more broadly, suitable periods based on the importance of each task). It's the amount of time a process stays in the ready queue [2].

3. JOB SHOP SCHEDULING PROBLEM

The following is a formula for the flow shop problem. For n > 1, each of n jobs from the job set I = 1, 2,...., n has to be handled in the sequence specified by the machine indexing on m machines 1,2,....,m. Thus, job j, j $ J, is made up of a series of m operations, each of which corresponds to the processing of job j on machine I during a period of time pij 0. It is considered that a machine's zero processing time refers to a work completed in an infinitesimal amount of time. Machine j, j = 1, 2,....,m, can only process one work at a time, and each machine is expected to handle the jobs in the same sequence. The goal is to design a processing sequence for the jobs on the machines that minimizes the overall completion time, or schedule makespan (Cmax). The processing times

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

required for jobs on the machines are indicated as pij, where I = 1,...,n and j = 1,...,m; these durations are constant, predictable, and non-negative.

Several assumptions are made in relation to this problem:

Each work I can only be processed by one machine j at a time.
At any one moment, each machine m can only process one job i.
Preemption is not permitted, which means that the processing of a job I on a machine j cannot be disrupted.
The processing timeframes include the time it takes to set up tasks on computers.
The machines are available at all times.

Inventory that is in the process of being created is permitted. If the next machine in a task's sequence is unavailable, the work might wait and enter the queue at the next available machine [5].

4. THE GUPTA'S HEURISTIC ALGORITHM

Gupta predicted that a heuristic methodology [13] will be used to attain a near-minimal makespan. By comparing the dispensation times of the first and last machines in each work, the Gupta heuristic algorithm divides all jobs into two categories. Calculate the minimal processing time by adding the processing times of any two adjacent tasks in a job for each group, and then schedule the jobs in order of their minimum summed processing times.

Johnson's method is mostly utilized for two machines [14], although the Gupta algorithm's concept may be used to more than two machines [13]. This method specifies a m machines, a set of n tasks, and a chain of actions that must be performed in the same order on each machine. To solve it in polynomial time, Gupta developed the following heuristic approach [13]. Gupta's heuristic is outlined in the Algorithm below.

Input: A set of n jobs, each with m (m > 2) tasks that are executed on one of m machines.
Output: A schedule that keeps the last job's completion time to a bare minimum:

Step 1: Create a U-shaped group of jobs that take less time on the first machine than they do on the last. U = I | t1i tmi, for example.
Step 2: Create a V-shaped group of jobs that take less time on the last machine than they do on the first. V = j | tmj t1j is the first condition.
Step 3: Find the minimum of (tkj + t(k+1)j) for each job Ji in U for k = 1 to m-1;

Set rewritten:
for k=1 to m-1, i=min(tki+t(k+1)i)
Step 4: Find the minimum of (tkj + t(k+1)j) for each job Jj in V for k = 1 to m-1;

Restated set:
$\pi_j=\min(t_{kj}+t_{(k+1)j})$ for k=1 to m-1

Step 5: Sort the jobs in U in ascending order of π_i' s ; if two or additional jobs have the equal value of π_i, sort them in an random order.
Step 6: Sort the jobs in V in descending order of π_j's; if two or more jobs have the same value of π_j, sort them in an arbitrary order.
Step 7: Schedule the jobs on the machines in the sorted order of U, then in the sorted order of V.[8]

5. PROPOSED NEW MODIFIED ALGORITHM FOR PERMUTATION FLOW SHOP JOB SCHEDULING

Gupta's heuristic has been modified to create the anticipated job shop scheduling algorithm. Gupta's heuristic and Johnson's algorithm are combined in this algorithm. It deals with some fuzzy logic rules, which are based on each

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

job's operation time. It calculates a new value of execution for a combination of two machines using the operation time of each job (like, machine 1 and machine 2, machine 2 and machine 3 and so on). This is a paper about

Input: A set of n jobs, each with m (m > 2) tasks that are executed on one of m computers.
Output: A timetable that keeps the last job's completion time to a bare minimum.

Step 1: Maximum processing time on machine1 should be greater than or equal to mimimum processing time on machine m2 m3 m-1.
Step 2: Minimum processing time on machine m should be greater than or equal to maximum processing time on machine m2 m3 m-1.
Step 3: If one of these above condition is met then find out the shortest processing time.
Step 4: Form two hypothetical machines on which we perform the summed up of job's processing time.
Step 5: Form the group of jobs U that take less time on the first machine than on the last such that $U = \{ i \mid t_{1i} < t_{mi} \}$.
Step 6: Form the group of jobs V that take less time on the last machine than on the first such that $V = \{ j \mid tmj \leqq t1j \}$.
Step 7 **:** Merge these two group by taking the first two jobs and schedule them in order to minimize the partial makespan as if there were only these two jobs
Step 8**:** Calculate the make-span time for the sequence obtained in step 7.
Step 9**:** Find the ordered pair of jobs from *'job list'*, which corresponds at a minimum makespan.
Step 10: Set k = 3

Choose the k^{th} job from the sorted list and place it in the best partial sequence's k possible positions. Choose the best partial sequence with the shortest makespan from the k partial sequences.

Set $k = k + 1$

Example: Consider the table 1 problem of a 7-job, 4-machine flow shop with processing time.

Table I: problem

JOBS	M_1	M_2	M_3	M_4
J_1	3	1	4	12
J_2	8	0	5	15
J3	11	3	8	10
J_4	4	7	3	8
J5	5	5	1	10
J_6	10	2	0	13
J7	2	5	6	9

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Makespan calculation using GUPTA's Algorithm:

Group X= {J_1, J_2, J_4, J_5, J_7}
Group Y= {J_3, J_6}
J_1= Min {(3+1, 1+4, 4+12)} = Min (4, 5, 16) = 4
J_2= Min {(8+0, 0+5, 5+15)} = Min (8, 5, 20) = 5
J_3= Min {(11+3, 3+8, 8+10)} = Min(14,11,18) = 11
J_4= Min {(4+7, 7+3, 3+8)} = Min(11, 10, 11) = 10
J_5= Min {(5+5, 5+1, 1+10)} = Min(10, 6, 11)= 6
J_6= Min {(10+2, 2+0, 0+13)} = Min(12, 2, 13)= 2
J_7= Min {(2+5, 5+6, 6+9)} = Min(7, 11, 15)= 7
Ordered collection in Ascending X= {J_1, J_2, J_5, J_7, J_4}
Ordered collection in Descending Y= {J_3, J_6}
The Total Sequence is = {J_1, J_2, J_5, J_7, J_4, J_3, J_6} Makespan for that Sequence is = 85

Minimum Processing time on machine m should be greater than aor equal to the maximum processing time on machine m_2, m_3…m-1 - 8>=8

Condition satisfies then we introduce two hypothetical machines X and Y respectively.

JOBS	X	Y
J_1	8	17
J2	13	20
J_3	22	21
J4	14	18
J_5	11	16
J_6	12	15
J_7	13	20

Summed up the processing time first three machines namely M_1+M_2+M_3= X

Summed up the processing time last three machines namely M_2+M_3+M_4= Y

Ordered Group in Ascending X= {J1, J5, J6, J7, J2, J4} Ordered Group in Descending Y= {J3}

As a result, we must determine the ordered sequence of unscheduled jobs based on their processing time, so that {J_1, J_5, J_6, J_7, J_2, J_4, J_3}

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Table II: Makespan calculation using GUPTA's Algorithm

JOBS	M1	M2	M3	M4
J1	0-3	3-4	4-8	8-20
J2	3-11	11-11	11-16	20-35
J5	11-16	16-21	21-22	35-45
J7	16-18	21-26	26-32	45-54
J4	18-22	26-33	33-36	54-62
J3	22-33	33-36	36-44	62-72
J6	33-44	44-46	46-46	72-85

Table III: Calculation of the Makespan using a novel hybrid permutation flow shop scheduling method

JOBS	M_1	M_2	M_3	M_4
J1	0-3	3-4	4-8	8-20
J_5	3-7	7-14	14-17	20-28
J_6	7-17	17-19	19-19	28-41
J7	17-19	19-24	24-30	41-50
J2	19-27	27-27	30-35	50-65
J4	27-31	31-38	38-41	65-73
J3	31-42	42-45	45-53	73-83

Makespan calculation using a novel hybrid permutation flow shop scheduling.

Maximum Processing time on machine1 should be greater than or equal to maximum processing time on machine m_2, m_3,…m-1 11 > = 0 →

Makespan for this generated sequence is = 83
As a result, the proposed novel hybrid permutation job shop scheduling algorithm outperforms existing algorithms, as evidenced by the solutions.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

6. CONCLUSION

For even more than two machines, this paper proposed a new hybrid job shop scheduling algorithm. This algorithm outperforms Gupta's heuristic by a large margin. It offers both a minimum and a better partial makespan. The proposed algorithm also offers multiple options for achieving the best result. We get a sequence that has a shorter makespan than Gupta's heuristic. As a result, the proposed new modified job shop scheduling algorithm can shorten the time it takes for a job to be completed.

REFERENCES

[1] Nawaz, M., Enscore Jr., E.E. and Ham, I. (1983) 'A heuristic algorithm for the m-machine n-job flowshop sequencing problem', OMEGA International Journal of Management Science, Vol. 11, pp.91–95.
[2] Silberschatz Avi, Galvin P.B., Gagne Greg, "Operating System Concepts", Willey & Sons, 2008, ISBN 978-81-265-0885-3.
[3] Phaphan Wikanda, Wetweerapona Jeerayut, Remsungnen Tawun,"Combined Heuristics methods for total flow time minimization in permutation flow shops scheduling ", The 2nd International Conference on Applied Science (ICAS) The 3 rd International Conference on Science and Technology for Sustainable Development of the Greater Mekong Sub-region (STGMS) Souphanouvong University, Luang Prabang, Lao PDR. 24-25 March, 2011.
[4] Tyagi Neelam, R.G. Varshney, "A model of study Genetic Algorithm for the flowshop scheduling problem",Journal of Information and Operations Management,Bioinfo publications, ISSN:0976-7754 & E-ISSN:0976-7762, Volume 3, Issue 1,2012,pp-38-42
[5] Samuel Robin Kumar, VenKumar P.,"Some Novel Methods For Flowshop Scheduling", Robin Kumar Samuel et al. / International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462 Vol. 3 No.12 December 2011 8403
[6] Schwiegelshohn Uwe," Scheduling Problems and Solutions",CEI University Dortmund Summer Term 2004
[7] O. Odior Andrew, A. Oyawale Festus," A job Scheduling algorithm for 2-machine flow shop problem", International Journal of Engineering & Technology, 1(4)(2012) 305 -314.
[8] Hong Tzung-Pei, Chuang Tzung- Nan ," Fuzzy Gupta Scheduling for Flow Shops with more than two machines".
[9] Laha Dipak, Chakraborty Uday Kumar," 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,Springer-Verlag London Limited 2008.
[10] Kolharkar Shantanu , Zanwar D. R.,"Scheduling in Job Shop Process Industry", IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) ISSN: 2278-1684 Volume 5, Issue 1 (Jan. - Feb. 2013), PP 01-17.
[11] ANCAU Mircea," ON SOLVING FLOWSHOP SCHEDULING PROBLEMS", PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A, Volume 13, Number 1/2012, pp. 71–79.
[12] Luangpaiboon Pongchanun, Boonvirojrit Atipon, "Comparison of Heuristic Methods for the N-Job and M-Machine FlowShop with Resource Constraints" Department of Industrial Engineering, Faculty of Engineering, Thammasat University, Pathumthani 12121 Thailand.
[13] J. N. D. Gupta, Heuristic algorithm for multistage flow shop problem, AIIE Transactions 4, (1972) 11-18.
[14] S. M. Johnson, Optimal Two- and Three-Stage Production Scheduling with Setup Times Included, Naval Research Logistics Quarterly 1 (1954) 61-68.
[15] J. F. Gonçalves and M. G. C. Resende, An extended Akers graphical method with a biased random-key genetic algorithm for job-shop scheduling. International Transactions in Operational Research (2013).
[16] O. Engin, M. K. Yılmaz, M. E. Baysal and A. Sarucan, Solving Fuzzy Job Shop Scheduling Problems with Availability Constraints Using a Scatter Search Method. Journal of multiple-valued logic and soft computing 21.3-4 (2013): 317-334.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[17] S. Sundar, P. N. Suganthan, and T. J. Chua, A Swarm Intelligence Approach to Flexible Job-Shop Scheduling Problem with No-Wait Constraint in Remanufacturing. Artificial Intelligence and Soft Computing. Springer Berlin Heidelberg, 2013.
[18] R. Zhang,S. Song, and C. Wu, A hybrid differential evolution algorithm for job shop scheduling problems with expected total tardiness criterion. Applied Soft Computing 13.3 (2013): 1448-1458.
[19] X Zhang, Y Deng, FTS Chan and P Xu, IFSJSP: A novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets. International Journal of Production Research ahead-of-print (2013): 1-20.
[20] Sandeep Kumar, Rajani Kumari, Vivek Kumar Sharma , Fuzzified Job Shop Scheduling Algorithm, HCTL Open Int. J. of Technology Innovations and Research HCTL Open IJTIR, Volume 7, pages 1-20 January 2014.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[21] M.Vairamuthu, S.Porselvi, Dr.A.N.Balaji, J.Rajesh Babu," Artificial Immune System algorithm for multi objective flow shop scheduling problem", International Journal of Innovative Research in Science, Engineering and Technology, ISSN (Print) : 2347 – 6710, Volume 3, Special Issue 3, March 2014.
[22] Prof. Dr. Akeela M. Al-Atroshi , Dr. Sama T. Azez Baydaa S.Bhnam," An Effective Genetic Algorithm for Job Shop Scheduling with Fuzzy Degree of Satisfaction, IJCSI International Journal of Computer Science Issues, ISSN (Print): 1694-0814 ,Vol. 10, Issue 5, No 1, pp:180-185,September 2013.
[23] Martin Ždánský, Jaroslav Požívil, "Flowshop Optimization in
[24] Matlab - Genetic Algorithms Approach", kontakt na autory : Technická 1905, 166 28 Praha 6, sponsored program No. MSM 223400007, Czech Republic.
[25] R. Ramezanian, M.B. Aryanezhad , M. Heydari, "A Mathematical Programming Model for Flow Shop Scheduling Problems for Considering Just in Time Production", International Journal of Industrial Engineering & Production Research, ISSN: 2008-4889, September , Volume 21, Number 2, pp: 97-104,2010.
[26] Michal Kutil , Premysl Šucha , Roman Capek and Zdenek
[27] Hanzalek," Optimization and Scheduling Toolbox", Michal Kutil, Premysl Sucha, Roman Capek and Zdenek Hanzalek (2010).
[28] Hassan Gholizadeh and Reza Tavakkoli -Moghaddam ," Minimizing the Makespan in a Flow Shop Scheduling Problem with Sequence-Dependent Setup Times and Periodic Maintenance by a Hybrid Algorithm", International Conference on Industrial Engineering and Operations Management Istanbul, Turkey, July 3 – 6, pp: 806-814,2012