Shuffled frog leaping algorithm approach to employee timetabling and job shop scheduling

Article *in* International Journal of Internet Manufacturing and Services · January 2014

DOI: 10.1504/IJIMS.2014.062443

2 authors:

Ramya G
REC

Dr.M. Chandrasekaran
Vels institute of Science Technology and Advanced Studies

Shuffled frog leaping algorithm approach to employee timetabling and job shop scheduling

G. Ramya*

Department of Mechanical Engineering,
Sathyabama University,
Jeppiaar Nagar, Old Mamallapuram Road,
Chennai, Tamilnadu, Pin Code: 600119, India
E-mail: ramyacadcam@gmail.com
*Corresponding author

M. Chandrasekaran

Department of Mechanical Engineering,
VELS University,
Velan Nagar, P.V. Vaithiyalingam Road,
Pallavaram, Chennai, Tamilnadu, Pin Code: 600117, India
E-mail: ch_sekaran@yahoo.com

Abstract: In manufacturing systems, generally employee timetabling and job shop scheduling is considered to be too hard and complex. To achieve an integrated manufacturing environment, the integration of employee timetabling and job shop scheduling is essential. It is a process of assigning employees into work slots in a pattern according to the hierarchy level of the employee, job allotment, individual preferences and knowledge skills. This paper proposes a framework model for integration between the two production functions with an objective of minimising labour costs based on employee availability constraint. At first, it provides a common time representation for employee timetabling and scheduling. Then, it provides the application of shuffled frog leaping algorithm used to find an optimum production schedule. An industrial case study also has been analysed for generation of employee timetabling and master production schedule. The result shows that the proposed system is unique for its features such as automatically constructing job sequence and employee work load.

Keywords: job shop scheduling; employee timetabling; integration; heuristics algorithm; shuffled frog leaping algorithm; SFLA.

Reference to this paper should be made as follows: Ramya, G. and Chandrasekaran, M. (2014) 'Shuffled frog leaping algorithm approach to employee timetabling and job shop scheduling', *Int. J. Internet Manufacturing and Services*, Vol. 3, No. 3, pp.178–203.

Biographical notes: G. Ramya is a Research Scholar in Sathyabama University, Chennai. She has completed her UG BE in Mechanical Engineering from Periyar University in 2003. Then, she completed her PG ME in CAD/CAM from Anna University in 2007 and also she has been pursuing her PhD research work in the field of computer integrated manufacturing at Sathyabama University, Chennai, Tamilnadu, India. She has nine years of teaching experience. Now, she is working as an Assistant Professor in the

Department of Mechanical Engineering, Rajalakshmi Engineering College. She has published many papers in international conferences and journals.

M. Chandrasekaran is working as a Professor and Director in Vels University. He has 18 years of teaching experience in reputed colleges. He completed his Master's and Doctorate degree in Production Engineering from the National Institute of Technology, Trichy, India. His research areas include computer integrated manufacturing, composites and scheduling. He published many papers in international journals and conferences. He is an active member in ASME, ISTE, NIQR and Institution of Engineers Professional Societies.

This paper is a revised and expanded version of a paper entitled 'An integrated approach to employee timetabling and job shop scheduling' presented at 4th International & 25th All India Manufacturing Technology, Design and Research Conference [AIMTDR 2012], Production Engineering Department, Jadavpur University, Kolkata, 14–16 December 2012.

1 Introduction

A schedule is an allocation of tasks to the time intervals on the machines. In the job shop scheduling problem *n* jobs have to be processed on different *m* machines. Each job consists of a sequence of tasks that have to be processed during an uninterrupted time period of a fixed length on a given machine. Today's manufacturing planning and scheduling requirements often minimises the overall completion time, rapid response to maximising customer's satisfactions and also minimising the product process cost. In manufacturing systems, the decisions related to employee timetabling and the decisions related to scheduling jobs on the machines are often made in a sequential process. The objective of job scheduling is to find the optimum schedule to minimise the costs whereas the objective of employee timetabling is to maximise employee satisfaction and to minimise labour costs. In many manufacturing industries employee timetabling is first prepared and then the scheduling of jobs prepared based on the resources and employee availability or first the scheduling of jobs is done and the employees timetabling established based on the machine loads. An integrated system is used to improve the production costs, minimising labour costs, maximising completion time and increase the employee satisfaction. Normally, compare to all practical situations these job shop scheduling problems has been considered to be complex problem (Artigues et al., 2006a).

The employee timetabling problem is to assign a single activity to each employee at each time period including idle period. Assigning the load based on such activities to the employee is called timetabling. There are many restrictions to be considered for preparing the possible timetable such as load constraints and employee profiles. To maximise the employee's satisfaction, consideration of the employee timetabling problem is the main objective and also acting as major role. The employee time tabling philosophy is still employed by the majority of manufacturing enterprises for job shop scheduling, process shift planning and production planning (Cummings et al., 1998). Employee time tabling has been found to be an effective way explicitly to consider relationships between the end items and the various processes and labours (Meisels and Schaerf, 2003).

Time tabling systems determine the quantity of each labour that will be used in the production of a prescribed volume of final work, and the times at which each of them must be utilised to meet prescribed due dates for the final products. Employee timetabling systems are highly detailed and an excellent means for assigning loads and tracking resource requirements. As a means for production scheduling, time tabling systems leave a good deal to be desired and provides the means to make broad scheduling decisions (Gecevska et al., 2006). It does not encompass short term scheduling decisions like machine loading and operations sequencing. Once work load has set due dates for each stage, it becomes the responsibility of the shop floor scheduling system to meet such deadlines. This is a critical activity because the load on work centres changes over time. There can be such unexpected events as machine breakdowns, raw material shortage, scrap and rework, all causing the actual lead time to differ from the planned one. Moreover, computation does not take into consideration capacity constraints at the shop level, thus often causing overload: in this situation, the choice between which job to process, and which one to delay, becomes crucial. Production volumes and due dates must be adjusted manually to achieve feasible schedules. However, the main difference between tabling and finite scheduling simply tries to schedule all activities required to meet a given master schedule while holding down work-in-progress inventory. If infeasibility occurs, production management must produce a new master timetable and production schedule to generate another plan or find alternative sources of production capacity. Finite scheduling is an optimisation technique that tries to generate a sequence of operations over a given set of machines with the sole purpose of minimising some type of shop performance measure like makespan, mean flow time, etc.

Job shop scheduler is used to generate job sequences randomly and allocate the tasks to time intervals on the machines and the aim is to find a schedule that minimises the overall completion time, which is called the makespan. For processing all the jobs in machine shop the maximum of completion times needed is based on the constraints that each job has a specified and stipulated processing order through the machines and that each machine can perform its process at most one job at a time (Pezzella and Merelli, 2000). Gantt chart is used to find the initial sequence and makespan in job shop simulator. Bruker (1995) and Garey et al. (1976) show that the job shop scheduling is an NP-hard problem. Scheduling problem is NP-hard because it is little consolation for the algorithm designer who needs to solve the problem. To find an optimal solution the characteristics of NP-hard job-shop scheduling are usually treated as very hard and in the form of mathematical sense are not always necessary in practices (Erschler et al., 1976). Recent research focus has been analysed for to find the nearest optimal solutions by using all the kind heuristic algorithms. Many valid approaches and its advances are compared and shared between competitors in rapid form. Several techniques have been proposed and different heuristics have been designed and developed for solving the minimum makespan problem, the minimum total tardiness problem and so on. Each and every approach has its own valid solution exclusively (Ghoniem, 2002). French predicted that no efficient algorithms will ever be developed for the majority of scheduling problems (Shmoys et al., 1994). As a result, the focus of optimisation research has turned to be enumerative approaches. Sometimes the research result seems that near optimal solutions usually meet few conditions requirements of practical situation problems (French, 1982). An effective SFLA was used for minimising maximum completion time (i.e., makespan) (Yang et al., 2001).

In this work, a model of integrated employee timetabling and job shop scheduling is proposed. This process of employee timetabling and job shop scheduling takes an account of replacing an employee with another employee without notable remarks or interruption of the processed task. In this process, an employee may perform several tasks simultaneously during a shift. A set of activities has to be performed at each shift. Each activity requires a specific number of labours. Integration between the two problems is made by associating each job on machine and a set of activities performed by the employees. The system has been designed to store the data needed for the above mentioned scheme and meets all the required computations. Specifically, this topic covers that the required job profile is not known in advance but if the job-profile is determined by the job schedule, the employee profile is determined by the selected employee schedules. The application developed in this paper aims to realise an integrated system which has rapid response to changing customers' requirements and capability to integrate heterogeneous manufacturing facilities.

To develop a window-based application which helps the organisation to attain the best procurement practices and supports the operation of procurement activity at the optimum total cost in the correct quality at the correct time and location for express gain by signing a contract.

The best approach to solve the resulting problem would consider the integrated problem often decomposed into load assignment part and job scheduling part. In this paper, author proposed to solve integration of employee timetabling and job shop scheduling alternatively in two different kinds of levels. For the first level, a mathematical model has been developed for integration of employee timetabling and job shop scheduling. In which the shuffled frog leaping algorithm (SFLA) is implemented for computing optimal sequence of jobs on each machine and its results are compared with instance of six jobs, four machines, four activities, 15 employees and three shifts per day for a week (Eusuff et al., 2006). Second level, an industrial case study has been analysed for generation of employee timetabling and master production schedule. The case study analysis is successfully completed in Excel Evolver with macrons. These results investigate the interest of proposed integrated method compared with sequence of mathematical programming methods.

1.1 Literature review

Cordeau et al. (2001) developed the integration of employee timetabling and crew task scheduling in complex transportation systems. Time tabling systems determine the quantity of each labour that will be used in the production of a prescribed volume of final work, and the times at which each of them must be utilised to meet prescribed due dates for the final products. Employee timetabling systems are highly detailed and an excellent means for assigning loads and tracking resource requirements. Meisels and Schaerf (2003) developed the modelling and solving employee timetabling problems by using shift planning. The employee time tabling philosophy is still employed by the majority of manufacturing enterprises for job shop scheduling, process shift planning and production planning. Employee time tabling has been found to be an effective way explicitly to consider relationships between the end items and the various processes and labours. Ernst et al. (2004) introduced a review of applications of staff scheduling and roistering that takes account of replacing employee with another employee with no notable remarks or interruption of the processed task, an employee may perform several tasks simultaneously during a shift, at each shift a set of activities has to be performed and each activity requires a specific number of labours. Integration of human resources and project scheduling with consideration of time dependent activities requirements also has been developed.

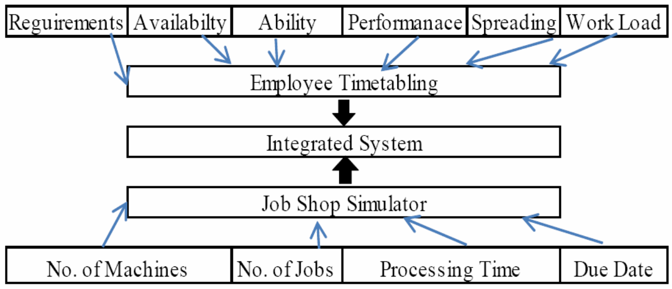
Chen (2004) developed a model of integration between employee timetabling and scheduling has been implies employee satisfactions and production cost improvement. As a means for production scheduling, time tabling systems leave a good deal to be desired and provides the means to make broad scheduling decisions. It does not encompass short term scheduling decisions like machine loading and operations sequencing. Silva et al. (2004) delivered a notes on multi objective meta heuristics for solving integration of scheduling and timetabling with multiple objective functions. It considered a fixed scheduling in a multiple machine environment with the problem of finding the minimum number of employees. Huq et al. (2004) proposed employee scheduling and makespan minimisation in flow shop with multi-processor workshop also used a heuristic approach to minimise the labour cost and total project cost by simulated minimisation of labour cost linked with total number of employees and late delivery costs in assembly shop with multiple workstation.

Hooler (2005) use hybrid method mixing linear programming and constraint programming to solve the integrated problem by alternatively solving it at two different levels based on sequence of jobs on each machine. The employee timetabling problem is to assign a single activity to each employee at each time period including idle period. Assigning the load based on such activities to the employee is called timetabling. There are many restrictions to be considered for preparing the possible timetable such as load constraints and employee profiles. Schaerf et al. (2006) solved problems with foundation of some text instances for standard school timetabling and standard universal benchmark instances. Finite scheduling is an optimisation technique that tries to generate a sequence of operations over a given set of machines with the sole purpose of minimising some type of shop performance measure like makespan, mean flow time, etc. Artigues et al. (2006b) proposed exact hybrid methods for an integrated employee timetabling and job shop scheduling problem based on integer programming and constraint programming. Integration of individual shift scheduling and employee timetabling is a very complex problem, when considering an objective of assigning single activity to each employee at each period of time. The computation experiments are based on representation of randomly generated instances confirms the superiority. Artigues et al. (2006a) proposed a flexible model and hybrid exact approach method such as integer linear programming and constraint programming formulations for integrated employee timetabling and production scheduling problems. This paper shows the flexibility of constraint programming modelling used to represent complex relationships between schedules and activity demands. Guyon et al. (2009) developed the integration of classical employee timetabling problem and feasible production scheduling problem to get near optimal and optimal solutions. To integrate the two stages, a specific decomposition and a cut generation method have been used. To maximise the employee's satisfaction, consideration of the employee timetabling problem is the main objective and also acting as major role. A cut generation method for an integrated employee timetabling and production scheduling problem has also been developed with real interest of an exact method based on a specific decomposition and cut generation process which is several orders of magnitude fast than one of the current MIP server (Guyon et al., 2010). Valls et al. (1996) considered a fixed scheduling in a multiple machine environment with the problem of finding the minimum number of employees. Bailey et al. (1995) proposed an integrated model for project task and manpower scheduling and also used a heuristic approach to minimise the labour cost and total project cost. Faaland et al. (1993) simulated minimisation of labour cost linked with total number of employees and late delivery costs in assembly shop with multiple workstations.

2 Overall system architecture

The opportunities identified that solving level of the proposed system based on measure of operational feasibility and development of the project involved for planning, executing, initiating, monitoring and controlling as shown in Figure 1 (Freling et al., 2003). Because this system will be certainly supported, it produces good result, it satisfies the requirements identified in the requirements analysis phase of system development. The level of acceptance by the user socially depends on the methods that are employed to educate the user about the system and to make him familiar with it.

Figure 1 Overall system architecture (see online version for colours)



Source: Freling et al. (2003)

This project will certainly be beneficially since there will be a reduction in manual work, and increase in the speed of work and does not need any high cost equipment. In the existing system suppliers are predetermined manually by the procuring company but in the developed system it has to be done by the system that is the suppliers are ranked by the system based on payment details.

2.1 Integrated system

The traditional term of constraints in the context of integrated system includes scope, nature of the work, schedule baseline and cost baseline. Here the quality of the timetable is inherent part of scope baseline.

2.2 Timetable life cycle

Timetable life cycle enhance the collection of logically related work activities usually culminating in the completion of a major deliverable. Collectively the time table's phases are known as timetable life cycle. In job shop the production schedule creation is based on grouping of ideas in natural, by taking right decisions in time and undergoing proper actions into product phases in all operations (Chien et al., 1999). Employee timetable framing is based on the interpersonal skills, preferences, understanding the product environment, general knowledge skills, negotiation, leadership, mentoring and knowledge on application areas.

2.3 Purpose of initiating processes

a Based on resources, priority, focus of seniority data to commit (assigning the work load) the labours to a project.

b To define top level work load objectives and set the overall solution directions.

c To secure the necessary approvals and resources, also to identify their expectations and validate alignment

d The process of identifying and documenting relationships among activities. Schedules activities are sequenced with logic relationship.

2.4 Input, output and feasibility modules

The input modules are activity list, activity attributes, milestone list, work load scope statement, preference list, consolidated list with all detailed profiles and organisational process assets. The output modules are production schedule network and project document updates. Forward pass is starting at the beginning of integrated system develop early start and early finish dates for each task, processing to end of the network. Early start date is earliest possible point in time an activity can start based on the integrated network logic and any schedule constraints. Early finish date is earliest possible time the activity can be finished.

3 Job shop model

Typical scheduling problems involve minimising the maximum $g_j(t)$ value (the maximum cost problem) or minimising the sum of $g_j(t)$ values (the total cost problem). Scheduling is defined as the art of assigning resources to tasks in order to insure the termination of these tasks in a reasonable amount of time (Zaccaro et al., 2001). The term ‘scheduling’ in manufacturing systems is used to the determination of the sequence of operations in which parts are to be processed over the production stages. To meet an optimal objective solution or set of objectives these approaches are used for determination of the starting time and finishing time of processing of each part. Some other cases scheduling problem is addressed after the orders are released into the shop floor, along with their process plans and machine routings.

The simulated job shop model development is using the object-oriented paradigm method. The various components of job shop such as job seeds, machines allocation seeds, solutions seeds and generic seeds are developed as generic objects in the simulator and they perform acting role as building blocks for the job shop model.

3.1 Representation of solution seed (sequences) in job shop simulator

Consider the three-job three-machine problem as shown in Table 1 and Table 2. Suppose a seed is given as [3 2 1], where 1 stands for job j1, 2 for job j2, and 3 for job j3. This sequence has to be operated 3 times in the same order because each job has three operations. So that the initial seed as the following format [3 2 1 3 2 1 3 2 1]. There are three 2s in the seed, which stands for the three operations of job j2. The first 2 corresponds to the first operation of job j2 which will be processed on machine 1, the second 2 corresponds to the second operation of job j2 which will be processed on machine 3, and the third 2 corresponds to the third operation of job j2 which will be processed on machine 2. We can see that all operations for job j2 are given the same symbol 2 and then interpreted according to their orders of occurrence in the sequence of this seed. This concept is used to find the makespan for the sequences of the problems where the generated seed (job sequence) is operated equal to the number of machines represented in the particular problem.

Table 1 Machine sequence and processing time

Job	*Machine sequence*			*Processing time*		
	1	*2*	*3*	*1*	*2*	*3*
J_1	M^1	M^2	M^3	3	3	2
J_2	M^1	M^3	M^2	1	5	3
J_3	M^2	M^1	M^3	3	2	3

4 Communication between employee timetabling and job shop scheduling

4.1 Formulation of objective function

Consider the common time representation for employee timetabling and job shop scheduling (Artigues et al., 2006b). We consider the following employee timetabling and job shop scheduling problem with single level jobs. Let T denote a time horizon with a set of elementary time periods $t = 0$, $T = 1$. E denote employees in organisation comprising a set of employees $E = \{1,\dots,E\}$ and M denotes set of machines $M = \{1,\dots,m\}$. Consider a non-pre-emptive job shop with m machines ($M_i = i,\dots m$) and n jobs ($N_i = i,\dots n$). When j_i is the set of job to be processed on machine M_i. The operation sequence of the job j is denoted by O_{ij} (where i^{th} operation on j^{th} machines M_j). Objective functions depend on due date which are associated with the jobs. A job consists of number of operations (O_{i1}, O_{i2},…O_{in}). There is set of activities $A = \{1,\dots,A\}$ where each activity may be required by a job j and has to be performed by one or several employees. The organisation has to process a set of n jobs $J = \{1, n\}$ during the time horizon (T). Each job j has a release date r_j and a due date d_j. We assume that there is a production cost Wjt if job j starts at time t and an employee satisfaction cost $Ceat$ if employee e is assigned to activity a at time t and A contains non-working activities representing employee inactivity (break, lunch, etc.) gathered in set P.

Equation (1) shows the objective of the problem is to minimise the labour cost subject to the following constraints (Artigues et al., 2006a). Equation (2) to equation (6) represent exactly that once the each job has to be started, all the started jobs have to be finished within its time zone, each job can be processed by a machine at each time period with satisfaction of precedence constraint, each employee has to be assigned with certain activity (at least one) at each time period and specific constraints of each employee has to be taken in account for the work to be done within a fixed time. For instance, if no employee can work more than two consecutive shifts, the constraints of the form can be defined for each time period $t = 1$, $T = 2$ for each employee.

$$\min(f) = \sum_{j=1}^{n} \sum_{t=0}^{T-1} WjiXji + \sum_{e=1}^{E} \sum_{a=1}^{A} \sum_{t=0}^{T-1} CeatYeat \tag{1}$$

Subject to

$$\sum_{t=0}^{T=1} Xji; \quad \forall, Xjt = \{1, 0\} \tag{2}$$

$$\sum_{j=1}^{T=n1} Xij; \quad \forall, Xjt = 0; t = \{rj, \ldots dj - pj\} \tag{3}$$

$$\sum_{j=1}^{n} \sum_{t=0}^{t-1} pjt\, djt\, kjt; \quad \forall, djt = \{1 < k, k < p\} \tag{4}$$

$$\sum_{t=0}^{t=T} Yeat = 1; \quad \forall, t = \{0, \ldots, T-1\} \tag{5}$$

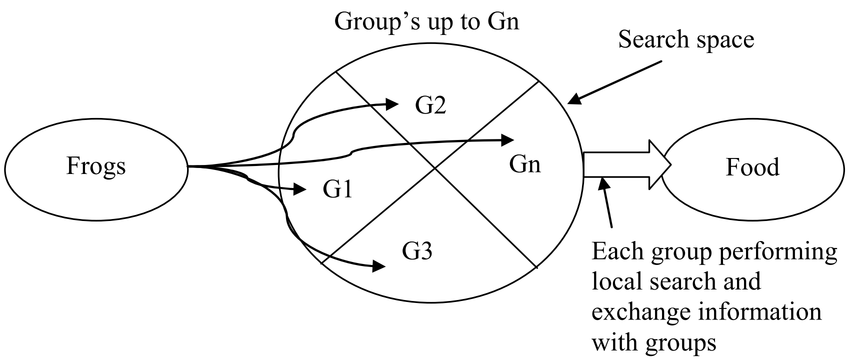
$$\sum_{e=1}^{E} \geq \sum_{j=1}^{n} \sum_{a=1}^{A} \sum_{t=0}^{T-1} Weatj\left(Ceatj - Peatj - Reatj\right); \quad \forall, t = \{0, \ldots, T-2\}, Wj > Pj = \{0, 1\} \,\&\, Rj < n, t = 0, T-2 \tag{6}$$

For a given schedule (*S*), *Ceatj* is the cost at which job *j* finished processing on machine *i* and *Wj* is the weighted time of job *j* spends in the queue before the first machine *i*. All ready times, processing times and due dates are assumed to be integer. In the above function nth job is performed in i^{th} machine with j^{th} operation with unit time consideration for time *Peatj* and cost j_{cost}. If the i^{th} machine is assigned with j^{th} operation for the first job is *Xjt* is 1, 0. If the i^{th} machine is assigned with j^{th} operation for the k^{th} job is $P_{ij}^{(k)}$ is 1, 0. Further, to solve the above objective functions and like to find an optimum solution, the heuristics method named shuffled frog leaping algorithm has to be implemented and validated.

4.2 Shuffled frog leaping algorithm

Farahani et al. (2010) proposed a new meta-heuristic algorithm called SFLA for solving scheduling problems with discrete decision variables. SFLA is a population-based cooperative search metaphor combining the benefits of the genetic-based memetic algorithm and the social behaviour-based particle swarm optimisation inspired by natural memetics (Eusuff and Lansey, 2003). Through the algorithm observation, the parameters such as copying behaviours and modelling the behaviour of frogs searching for food are placed in different stones which are haphazardly positioned in a pond. SFLA has been tested on a large number of combinatorial problems and found to be efficient in finding global solutions (Pan et al., 2011). The SFLA is a population-based cooperative search metaphor inspired by natural memetics and consists of a frog leaping rule for local search and a memetic shuffling rule for global information exchange as shown in Figure 2.

Figure 2 Frogs searching food



The SFLA comprises a set of interacting virtual population of frogs partitioned into different group's population memeplexes which are referred to searching for food. The algorithm functions are simultaneously independent in local search of each memeplex. In terms of processing time and makespan the SFLA compares the results rapid favourably with the sheep flock heredity model algorithm, artificial intelligence system, genetic algorithm, and particle swarm optimisation.

Farahani et al. (2010) identified a new hybrid algorithm called hybrid shuffled leaping frog algorithm based on the identification of the weaknesses of the basic SFLA. At first, the SFLA is initially applied to different functions and to identify the fundamental weaknesses of this method as per the elimination of the effective frogs from memeplexes by solving procedure in consequence order. This method is similar to the SFLA, partitions particles into different groups called memeplexes and identified the best particle in each memeplex thereafter determines its movement through the search space in each iteration of the algorithm toward the global best particle and the worst particle in each memeplex keeps track of its coordinates in the solution space by moving toward the local best particle in the same memeplex.

Fang and Wang (2012) proposed encode concept for the virtual frog as the extended activity list and decode it by the SFLA-specific serial schedule generation scheme. The initial populations are identified and generated by the mutual-based shuffling method and the priority rules. The large group of populated virtual frogs are separated into many set of memeplexes is the next stage and by applying the effective resource-based planning each memeplex are evolves the crossover. Combining the permutation-based local search and forward-backward improvement is to enhance the improved exploitation ability. Virtual frogs are periodically shuffled and rearranged into new set off memeplexes are maintained by diversity of each memeplex. Elbeltagi et al. proposed modified SFLA and its application to project management. In that different memetic vector cells were considered as different nature of frogs, each memetic frogs were performing local search within each memeplexes. All the virtual ideas are processed in each frog group after completion of every memetic evolution process (Elbeltagi et al., 2007). HSLFA also has a distinct advantage over the SFLA in that it reduces the probability of the particles being trapped in the local minima by directing the best local particle toward the global best particle.

4.3 Implementation of SFLA

In this section, an SFLA for solving the JSS problem with minimising total holding cost and makespan criterion are proposed by population initialisation, partitioning scheme, memetic evolution process, shuffling process, and a local search. SFLA is combination of memetic algorithm and particle swarm optimisation. It has been performed from memetic evolution of a group of frogs when seeking for food. The initial population of frogs was partitioned into groups or subsets called ‘memeplexes’ and the number of frogs in each subset was equal.

The SFLA is follows two search techniques

a local search

b global information exchange.

Based on local search to reach the makespan, the frogs in each subset improve their positions to have more foods. After local search, obtained information based on global information exchange between each subset was compared to other to produce best sequence way of schedule. Each operation is decided by meeting pre-specified due dates and minimising objective function. Initial population of sequence generated randomly by increasing order and selected sequence divided into number of meme lexes.

4.3.1 Local search procedure

The division is done with the high level frog (column sequence) arranged in first memeplex, second one arranged in second memeplex, the last frog to the last memeplex and repeated frog back to the next order memeplex. Fitness function evaluated within the limits that the memeplex are infeasible.

4.3.2 Global information exchange

The best frog memeplex values were identified each subset was compared to each other to produce best sequence way of schedule. For each iteration, the frogs with the best fitness and worst fitness were identified and also the frog with the makespan schedule was identified. Finally, if the convergence criteria are not satisfied the position of the worst frog for the memeplex is adjusted and new subsets of memeplex will be created for the next iteration.

4.3.3 SFLA heuristics algorithm procedure

Start;

Step 1:	*Randomly generate the population size of frogs P in Feasible situation & Initialize the population size equal to no. of memeplexes;*
Step 2:	*For each individual population P, calculate the fitness size (i) & Calculate size of each memeplex subsets;*
Step 3:	*Rearrange the population size randomly;*
Step 4:	*Evaluate P based on the hierarchy order of their fitness & Divide P into m memeplexes with i = 1 to no. of generations;*
Step 5:	*Perform Local search to Improve frog position to have best food;*
Step 6:	*For each memeplex; determine the best and worst frogs; improve the worst frog position by removing worst frogs in frame;*
Step 7:	*Shuffle each improved memeplexes and Combine the evolved memeplexes;*
Step 8:	*Sort the population P in descending order of their fitness;*
Step 9:	*If Convergence criteria satisfied (Make pan) move to end or else move to step 1*

End;

4.4 Benchmark problems

Consider a problem comprising six jobs × four machines, four activities × 15 employees and three shifts per day for two days (Artigues et al., 2006a). The total schedule horizon is considered as 48 hours for six shifts. The number of machines and its job duration is represented in Table 2. The duration (p_{ij}) has been generated randomly between 1 and 10. Based on logic formulation the job shop specific constraint must be taken as $C_{\max}$ greater than C_{im}, where i as $1...n$. The employee data with activities and assignment costs are shown in Table 3.

Table 2 Job data on machines with durations

J_i	m_{ij}				p_{ij}			
1	3	2	0	1	3	7	2	9
2	3	0	1	2	8	10	3	1
3	2	1	3	0	3	4	8	6
4	1	3	0	2	10	3	3	9
5	2	0	3	1	10	8	4	7
6	2	1	0	3	2	3	10	4

4.4.1 Schedule builder of SFLA

Step 1 Initiations

Initial population of job seed sequences are generated randomly by increasing order and selected sequence divided into number of memeplexes.

Initialisation: Initial population (job sequence) is selected randomly

(1-5-2-4-3-6) / (4-1-5-6-3-2) / (1-4-5-2-3-6) / (4-3-2-5-1-6) / (3-6-1-5-4-2)

- Labour cost (employee): 31
- Makespan: 47.

Table 3 Employee data, activities and cost

Employee (E)	*Activities (A)*	*Employee cost (Ceas)*							
1	2	5	4	2	1	4	1	5	5
	4	3	4	4	4	2	4	4	4
2	1	4	2	2	1	2	4	3	1
	3	5	4	4	4	1	1	2	2
3	1	2	3	1	1	5	3	2	1
	4	4	5	1	2	4	3	5	1
4	2	5	2	4	1	5	3	1	4
	3	1	2	5	4	3	3	4	3
5	2	4	4	4	2	4	5	2	5
	3	5	5	2	1	3	3	2	2
6	1	2	4	4	5	4	2	5	5
	4	5	2	1	5	4	3	1	4
7	3	5	2	1	3	1	4	3	1
	4	2	3	5	3	5	4	5	3
8	1	5	5	1	1	5	4	3	2
	2	1	4	1	5	3	1	5	1
9	1	3	3	3	3	3	2	2	1
	2	1	5	5	2	5	3	3	5
10	3	3	5	1	3	4	4	4	1
	4	5	5	1	5	4	1	3	1
11	2	5	4	2	4	1	3	2	1
	3	5	1	5	2	5	2	3	1
12	1	3	2	3	4	2	1	2	4
	4	4	5	5	1	3	1	2	5
13	1	5	5	4	2	3	5	4	5
	4	2	3	2	3	1	3	4	2
14	2	5	3	3	5	1	2	3	5
	3	5	4	2	5	2	5	5	3
15	1	4	1	2	2	2	1	5	4
	4	1	2	5	3	5	4	1	4

Step 2 Population creation

For each individual population $P(i)$, calculate the fitness function $f(i)$. Based on the fitness function, calculate the size of each memeplex subsets and also randomly generate the population of the job sequence.

Step 3 Mutation

In the mutation operation, a memeplex subsets are generated using the mutation strategy to find the population P in descending order based on their fitness. Then evaluate and divide the population sequence P into m memeplexes with consideration of populations which is selected randomly.

Step 4 Crossover

Mutation operation generators are used to generate a trial function vector. In this operation, a random population sequence is generated in between 0 to 1 and if the random number is less than the crossover constant value copy the target value otherwise the mutant operation sequence value will be changed as 0 or 1 for $i = 1$ to number of generations.

Step 5 Local search

The division is done with the high level frog (column sequence) arranged in first memeplex, second one arranged in second memeplex, the last frog to the last memeplex and repeated frog back to the next order memeplex. Fitness function evaluated within the limits that the memeplex are infeasible. Then perform the local search to improve the frog position to have best food.

Step 6 Global information exchange

The best frog memeplex values were identified with each subset was compared to each other to produce best sequence way of schedule. For each memeplex, determine the best and worst frogs and improve the worst frog position by removing worst frogs in the operation sequence frame.

Step 7 Shuffling

The trial sequence obtained by the crossover operation generation is compared with the target sequence to determine the jobs and machine schedule that participates in the next generation and the fittest is passed on to the next generation. Finally, shuffle each improved memeplexes and combine the evolve memeplexes and also sort the population P in descending order of their best fitness value.

Step 8 Iterations

For each iteration, the frogs with the best fitness and worst fitness were identified and also the frog with the makespan schedule was identified.

- Iteration-1

  Labour cost (employee): 523

  Makespan: 50

- Iteration-2

  Labour cost (employee): 521

  Makespan: 48

- Iteration-3

  Labour cost (employee): 516

  Makespan: 47

- Iteration: final

  Labour cost (employee): 510

  Makespan: 46.

Step 9 Control parameters

Finally, if the convergence criteria are not satisfied the position of the worst frog for the memeplex is adjusted and new subsets of memeplex will be created for the next iteration. This procedure is repeated for desired number of iterations to reach optimal result.

4.5 Test setup and validation

The proposed integrated model has been tested for different instances of six jobs and four machines (Artigues et al., 2006a), eight jobs and eight machines, ten jobs and ten machines, 15 jobs and 15 machines. The experiments are conducted in ten mutations and 50 shuffling generations. Each problem has been tested for several times with standard mutation rates.

If the objective value $f(i)$ is lower than required processing value $P(i)$, then random value replaces the best compared value, otherwise, consider best fitness. The experiment results show that the proposed approach gives results almost equivalent to the previous methods in all kind of objectives. In Table 4, the results obtained in proposed method are compared with other procedure.

Table 4 Results obtained by proposed method

Problem	*J*	*M*	*Proposed approach*			*Previous approach (Artigues et al., 2006a)*		
			Labour cost	*Mutation*	*CPU time in sec*	*Labour cost*	*Mutation*	*CPU time in sec*
EJS-1	6	4	508	0.2	28	510	0.2	96
EJS-2	6	4	542	0.3	31	544	0.3	93
EJS1-1	8	8	576	0.4	23	576	0.4	28
EJS1-2	8	8	610	0.5	29	610	0.5	35
EJS2-1	10	10	845	0.6	34	839	0.6	98
EJS2-2	10	10	856	0.6	21	856	0.6	32
EJS3-1	15	15	712	0.6	19	713	0.6	37
EJS3-1	15	15	746	0.6	19	746	0.6	37

5 Case example

Orders received from the customer as per employee timetabling plan then those loads are allotted to corresponding labours in workstations for completing production tasks. Job shop scheduler will generate optimal schedule. The proposed SFL algorithm can be successfully implemented in industries handling wide variety of products in small volumes and the industries working with general purpose machines which can handle different operations. The job processing and waiting times can be conveniently cut down, employee workloads and machine loads can be balanced and also the user has a choice of choosing a solution from the set of alternative solutions as per his desired objective criteria. To validate the working of integrated system, a firm which is manufacturing finned tubes is considered. The customer order for manufacturing finned tubes and details of needed quantity in terms of week are as shown in Table 5.

Table 5 Customer order

Job/ part no.	*Part description*	*Lead time (week)*	*Stock on hand*	*Needed quantity*
j_1	Stainless steel tube	Three weeks	3 m in 200 nos 2.5 m in 50 nos	1,500
j_2	Copper tube aluminium	Four weeks	3 m in 100 nos 6 m in 150 nos	1,400
j_3	Copper tube aluminium	Five weeks	3 m in 50 nos 6 m in 100 nos	2,000
j_4	Aluminium	Two weeks	6 m in 50 nos	1,400
j_5	Carbon steel tube	Three weeks	6m in 350nos	3,000

Table 6 Finning plan and machine allocation for the month April 2011

Machine	*Fin OD/ Fpi (mm)*	*Total meters to be produced*	*Apr 1 and 2*	*Apr 4 to 9*	*Apr 11 to 16*	*Apr 18 to 23*	*Apr 25 to 30*	*Produced meters/ cumulative*
M/c a	57/10 and 9	6,733	1,600 m	2,400 m	2,400 m	333	286 m	6,733/6,733
M/c b	54/9	20,089	889m	2,400 m	2,400 m	2,400 m	2,400 m	10,489/20,089
M/c c	64/10 and 10	6,733	1,600 m	2,400 m	2,400 m	333	286 m	6,733/6,733
M/c d	55/10	2,093	-	-	-	2,093	-	2,093/2,093
M/c e	60/10 and 11	40,685	7,800 m	7,800 m	7,800 m	7,800 m	2,082 m	31,200/31,200
M/c f	56/11	20,089	2,400 m	2,400 m	2,400 m	2,400 m	1,200 m	9,600/9,600
M/c g	57/10 and 12	6,733	1,600 m	2,400 m	2,400 m	333	286 m	6,733/6,733
M/c h	57/12	40,685	7,800 m	7,800 m	7,800 m	7,800 m	2,082 m	31,200/31,200
M/c i	57/10 and 15	20,089	889m	2,400 m	2,400 m	2,400 m	2,400 m	10,489/20,089
M/c j	63/15	40,685	-	1,685	-	-	7,800 m	9,485/40,685

Note: Date: April 3, 10, 17 and 24 falls on Sunday.

Table 7 Daily work process report

Daily report		*Shift: I*	*Shift: II*	*Shift: II*
1	No. of finned tubes produced	46 nos.	41 nos.	12 nos.
2	No. of finned tubes accepted	46 nos.	41 nos.	13 nos.
3	No. of finned tubes rejected (with reason for rejection)	Nill	10 nos 41 nos.	2 nos. Nill
4	Qty of lubricant topped up	20 litres	Nill	Nill
5	No. of fin strip coil issued/wt	11 coils	8 coils	3 coils
6	No. of base tube issued	46 nos.	43 nos.	33 nos.
7	Inward material list	Nill	Nill	Nill
8	Any other tools consumed	Circlip – 92 nos.	Circlip – 82 nos.	Circlip – 54 nos.
9	Tools changed	No	Setting changed	Setting changed
10	Time of interruption of machine total hours of interruption	Nill	Due to setting changed from 11 fpi to 8fpi.	Due to setting changed from 11 fpi to 8fpi.
11	Reason for interruption of machine	Nill	Nill	Nill
12	How it was rectified	Nill	Nill	Nill

Based on the customer order, the process plan for the production of fin tubes and allotment of machines are scheduled in Table 6. The daily work process reports for the different shifts are shown in Table 7.

5.1 Progress on production objectives

The commercial production for 54 mm fin OD is completed in fourth machine. The production of finned tubes in the fourth high speed machine is continuous and the quality is satisfactory. Trials are taken for reduction of aluminium consumption of seven FPI finned tubes. If any requirements not stated in the enquiry but determined as necessary for the specified use, are identified and discussed with customer during preparing the quotation. A statutory and regulatory requirement related to the product, if applicable, is determined during preparing the quotation. Changes to the product requirements or any requirements such as quantity, delivery time or any other issue by the customer are recorded and modified production plan with rapid response. Once the customer receives the products the feedback regarding product quality, delivery, service are received and analysis are transferred to the master customer file. The competence reviewed for personnel perform work affecting product quality is determined. The enhanced competence of employees is measured by work output, planned output interpersonal skills, team sprit and communication. Production-based training should be conducted to create awareness, to update knowledge of production skills, to induce motivation for increasing productivity and to meet the organisational and functional changes from time to time.

Table 8 Processing time for employees

Example 1: processing hours details														
PLANT – I														
		Working hours – weak I							*Total Hrs*	*Rate/Hr*	*Gross amount*	*Others (less)*	*Net amount*
Sl. no	*Name*	*Day 1*	*Day 2*	*Day 3*	*Day 4*	*Day 5*	*Day 6*	*Day 7*					
1	Gokul NP	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
2	Karthi P	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
3	Muruga K	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
4	Kumar T	HD	12	12	8	12	Sun	12	44	16.67	733	180	553
5	Prakash G	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
6	Kavin P	HD	12	12	8	12	Sun	12	44	16.67	733	180	553
7	Prasanth A	HD	12	12	0	12	Sun	12	36	16.67	600	180	420
8	Prasanth B	HD	12	12	8	12	Sun	12	44	16.67	733	180	553
9	Murugan S	HD	12	12	8	12	Sun	12	44	16.67	733	180	553
10	Pradeep A	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
11	Ganapathy M	HD	14	12	8	12	Sun	12	46	16.67	767	180	587
12	Tamilselvan K	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
13	Vaikka D	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
14	Tamileniyan K	HD	12	12	12	12	Sun	12	48	16.67	800	180	620
15	Santhosh	HD	12	12	8	12	Sun	12	44	16.67	733	180	553
Daily total hrs		0	182	180	144	180		0	686	-	11,436		8,736

Table 8 Processing time for employees (continued)

Example 1: processing hours details													
PLANT – I													
		Working hours – weak II							*Total Hrs*	*Rate/Hr*	*Gross amount*	*Others (less)*	*Net amount*
Sl. no	*Name*	*Day 1*	*Day 2*	*Day 3*	*Day 4*	*Day 5*	*Day 6*	*Day 7*					
1	Muruga K	0	5	4	6	3	3	0	21	16.67	350	180	170
2	Kumar T	0	3	5	3	4	4	2	21	16.67	350	180	170
3	Prakash G	0	5	0	5	5	3	5	23	16.67	383	180	203
4	Kavin P	0	7	5	4	0	2	6	24	16.67	400	180	220
5	Prasanth A	0	4	6	3	2	5	4	24	16.67	400	180	220
6	Prasanth B	0	3	2	0	6	5	3	19	16.67	317	180	137
7	Murugan S	0	1	3	2	5	5	6	22	16.67	367	180	187
8	Pradeep A	0	4	0	5	7	0	4	20	16.67	333	180	153
9	Ganapathy M	0	0	6	4	5	4	2	21	16.67	350	180	170
10	Tamilselvan K	0	6	5	7	5	0	0	23	16.67	383	180	203
11	Vaikka D	0	0	6	3	3	3	6	21	16.67	350	180	170
12	Tamileniyan K	0	5	3	5	5	5	4	27	16.67	450	180	270
13	Santhosh	0	0	2	4	3	4	8	21	16.67	350	180	170
Daily total Hrs		0	43	47	51	53	43	50	287		4,784		2,444
Total amount:													11,180

Table 9 Production monitoring system module (see online version for colours)

April	*2012*		*Attendance*	*Half day work*	*Holiday work*	*Night shift*	*Sunday*	*Absent*	*Day and night (continuous working)*	*Leave*	*Day and OT*	*Present*	*Off*	*Holiday*	*Unplanned leave*	*Planned Leave*	
				0.5	*W*	*N*	*S*	*A*	*PP*	*L*	*PO*	*P*	*O*	*H*	*SL*	*CL*	
S. no.	*Employee name*		*Skill*	*SUN*		*MON*		*TUE*		*WED*		*THU*		*FRI*		*SAT*	
				D	*OT*	*D*	*OT*	*D*	*OT*	*D*	*OT*	*D*	*OT*	*D*	*OT*	*D*	*OT*
1	Gokul NP	PM	Production maintenance	W	6	P		P		P		P		P		P	
2	Karthi P	P	Production	W	4	L		P		H		P		P		P	
3	Muruga K	P	Production	S		P		P		P		P		P		P	
4	Kumar T	P	Production	S		P		P		P		P		P		P	
5	Prakash G	RM	Record maintenance	S		P		P		P		P		P		P	
6	Kavin P	P	Production	S		P		P		P		O		P		P	
7	Prasanth A	Q	Quality	S		P		P		P		P		P		P	
8	Prasanth B	P	Production	S		P		P		P		P		P		0.5	
9	Murugan S	RM	Record maintenance	S		P		P		P		P		P		P	
10	Pradeep A	P/A	Purchase/administration	S		P		P		P		P		P		P	
11	Ganapathy M	P	Production	S		P		P		P		P		0.5		P	
12	Tamilselvan K	FMO	Finning machine operation	W	6	P		P		P		P		P		P	
13	Vaikka D	FMO	Finning machine operation	W	6	P		P		P		P		P		P	
14	Tamileniyan K	FMO	Finning machine operation	W	6	P		P		P		P		P		P	
15	Santhosh	T	Turning and tool making	W	6	P		P		P		P		P		P	

Table 9 Production monitoring system module (continued) (see online version for colours)

Total present	0	0	14	0	15	0	14	0	14	0	14	0	14	0
Total off	0		0		0		0		1		0		0	
Total absent	0		0		0		0		0		0		0	
Total on leave	0		1		0		0		0		0		0	
Total night shift	0		0		0		0		0		0		0	
Total half day	0		0		0		0		0		1		1	
Total work on Sundays	6		0		0		0		0		0		0	
Total OT hours		34		0		0		0		0		0		0
Total count	6	0	15	0	15	0	14	0	15	0	15	0	15	0
Department wise present list														
Total M/C (day shift)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total night shift	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total QC (day shift)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total QC night shift	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total M/C present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total assy. present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total plating present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total A/C present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total system present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total purchase	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total method	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total maint. present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total admin. present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total store present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total QC present	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total present	0	0	0	0	0	0	0	0	0	0	0	0	0	0

To impart training to enable employees to assume higher responsibilities and to ensure, monitor growth of personnel in the organisation. Training needs are identified for the period April to March every year for each employee by the section in charge with assistance from the concerned supervisory personnel. If no need is identified for a particular employee, it shall be clearly mentioned against his/her name. Manpower requirements are identified from time to time depending on the volume of orders received from customers. A new employee who is joining the organisation in the middle of the year in any category needs training which is identified by the section in charges. Awareness to quality policy and objectives are imparted to every new employee by the section in charge on joining the organisation. Based on all the requirements and constraints, employee timetabling model can be developed with multiple stages. The first stage development of processing hours for each employee based on job production is shown in Table 8. The next stage recording the work process and load for each employee, load on machine, total numbers of persons, hours are developed in production monitoring system module is shown in Table 9.

Table 10 Work load allotment

		Sl. no.	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
		Name	*Gokul NP*	*Karthi P*	*Muruga K*	*Kumar T*	*Prakash G*	*Kavin P*	*Prasanth A*	*Prasanth B*	*Murugan S*	*Pradeep A*	*Ganapathy M*	*Tamilselvan K*	*Vaikka D*	*Tamileniyan K*	*Santhosh*
Work load allotment	Monday	Shift 1	g-4-(2)	a-1-(1)		e-5-(1)		f-1-(2)		b-2-(1)			d-4-(1)		c-3-(1)		j-5-(2)
		Shift 2			a-6-(1)		e-10-(1)		h-3-(2)		b-7-(1)	i-2-(2)		d-9-(1)		c-8-(1)	
		Shift 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Tuesday	Shift 1			f-1-(3)						g-4-(3)	j-5-(3)			i-2-(3)		h-3-(3)
		Shift 2		b-7-(2)			e-10-(2)		a-8-(2)				c-6-(2)	d-9-(2)			
		Shift 3	b-7-(3)			e-10-(3)		a-6-(3)		d-9-(3)			c-8-(3)				

Locations: – Machines a, b, c, d, e, f, g, h, i and j –
Job sequence 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 – Jobs (1), (2), (3), (4) and (5)

Table 10 Work load allotment (continued)

Sl. no.	*Name*	Work load allotment – Wednesday Shift 1	Wednesday Shift 2	Wednesday Shift 3	Thursday Shift 1	Thursday Shift 2	Thursday Shift 3	Friday Shift 1	Friday Shift 2	Friday Shift 3
1	*Gokul NP*	a-3-(4)		-	d-4-(4)		-	b-1-(5)		
2	*Karthi P*		a-6-(3)	-		g-5-(5)	-			d-10-(4)
3	*Muruga K*		g-5-(3)	-			-	e-5-(5)		
4	*Kumar T*	e-5-(4)		-	c-3-(4)	i-4-(5)	-		i-2-(5)	
5	*Prakash G*		e-10-(3)	-			-			c-8-(4)
6	*Kavin P*		i-4-(3)	-		h-1-(5)	-	d-4-(5)		
7	*Prasanth A*	b-2-(4)		-	a-2-(4)		-		j-6-(5)	
8	*Prasanth B*		b-7-(3)	-			-			a-6-(4)
9	*Murugan S*		h-1-(3)	-			-		f-1-(5)	
10	*Pradeep A*		j-2-(3)	-	e-5-(4)		-			e-9-(4)
11	*Ganapathy M*		c-8-(3)	-		f-3-(5)	-	c-3-(5)		
12	*Tamilselvan K*	d-4-(4)		-			-		b-7-(4)	
13	*Vaikka D*		d-9-(3)	-	b-1-(4)		-			f-7-(3)
14	*Tamileniyan K*	c-1-(4)		-			-	a-2-(5)		
15	*Santhosh*		f-3-(3)	-		j-2-(5)	-		g-4-(5)	

Locations: – Machines a, b, c, d, e, f, g, h, i and j –
Job sequence 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 –Jobs (1), (2), (3), (4) and (5)

5.2 Master production schedule according to the case study

In this case study examine a company consisting of 15 employees, ten machines and ten jobs. Consider a work plan of a week except Sunday and includes three shifts per day. The constraint indicating that each shift must be assigned to a single employee and also each machine can be loaded with single operation of each job (keeping no machine in idle position). Table 10 explains at first the employee job sequence (1 to 10), at second number of machine (*a* to *j*) and at last number of jobs (1 to 10) work allotment.

5.3 Integrated system: implementation and evaluations

The Excel solver with macrons inter coding is used as platform for development of integrated application. The complete integrated system analysis and development design has been successfully tested. The experiment shows that the integrated system can rapidly respond to customers order. On the other hand, the employee timetabling system can effectively collaborate with job shop to meet requirement of the customers.

Implementation of integrated system is the important stage where theoretical design is turned to a working system. Two major factors for the implementation are testing the system and training the user. Since the existing system involves manual operations the new system is implemented in parallel with the existing. This was done to build users confidence about the system, and also to check the efficiency of the developed system. The existing system in schedule planning is Visual Basic with MS Access. It is not user friendly. Every time it requires series of commands for performing specific operations. It is time consuming system. Updating of data globally is difficult. File maintenance creates confusion for large purchase requisition. The main drawbacks of the existing system did not provide to the need of the management fully, programming is too complex, comparatively very slow and proper coding is not used.

In order to avoid these problems, integration of employee timetabling and job shop scheduling is effective and user friendly. After establishing employee timetabling, the system needs to send the daily planned order to the job shop. Once the customers add or modify the orders, this monitoring system in web server will invoke the employee timetable to update the data automatically. Then the updated work plan will be send to Job shop simulator which will then generate optimised job sequence based in the data base. When order is completed, the client residing in the job shop will communicate with server residing in timetable system to return finished jobs to timetable coordinator. Then the server residing in timetable system sends relevant data to database (Kragelund, 1997).

If the customer changes the requirements/orders employee timetable system will automatically update the workloads, master time table as it send purchase order to the supplier and process the randomised schedule order. In turn, the supplier will send the material in time and scheduler will generate the optimal schedule.

9 Conclusions

In this paper, at the first level an integration of employee timetabling and job shop scheduling has been developed by using mathematical programming. Further, SFL algorithm has been used for minimisation of labour cost and makespan. The results show that the proposed mathematical model is best compared to other integrated models. At the second level, an industrial case study has been taken into consideration is based on web solver by using excel macrons. The case study analysis describes employee timetabling system effectively collaborates with job shop to meet requirement of production. The integrated system application shows according to the testing results that the proposed system has good performance in terms of response to customer and customer satisfaction. In this paper, it has been proved that by both the methods mathematical programming with heuristic algorithm and web-based integrated system can be used for integration of employee timetabling and job shop scheduling. The proposed system is unique for its features such as automatically constructing job sequence, machine sequence, machine workload and employee workload.

References

Artigues, C. et al. (2006a) ‘A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling’, *Practice and Theory of Automated Timetabling VI*, Brno Czech Republic, DOI : 10.1007/978-3-540-77345-0_5.

Artigues, C. et al. (2006b) ‘Solving an integrated employee timetabling and job shop scheduling problem via hybrid branch and bound’, *PATAT 2006*, pp.64–81, ISBN 80-210-3726-1.

Bailey, J. et al. (1995) ‘Optimization and heuristic models to integrate project task and manpower scheduling’, *International Journal of Computers and Industrial Engg.*, Vol. 29, No. 1, pp.473–476.

Bruker, P. (1995) ‘*Scheduling Algorithms*, 2nd ed., Springer-Verlag, Berlin.

Chen, Z. (2004) ‘Simultaneous job scheduling and resource allocation on parallel machines’, *Annals of Operations Research*, Vol. 129, Nos. 1–4, pp.135–153.

Chien, S., Rabideau, G., Willis, J. and Mann, T. (1999) ‘Automating planning and scheduling of shuttle payload operations’, *Artificial Intelligence Journal*, Vol. 114, Nos. 1–2, pp.239–255.

Cordeau, J. et al. (2001) ‘Benders decomposition for simultaneous aircraft routing and crew scheduling’, *Transportation Science*, Vol. 35, No. 4, pp.375–388.

Cummings, M.D.H. et al. (1998) ‘Minimizing production flow time in a process and assembly job shop’, *International Journal of Production Research*, Vol. 36, No. 8, pp.2315–2332.

Elbeltagi, E., Hegazyz, T. and Griersonmad, D. (2007) ‘A modified shuffled frog-leaping optimization algorithm applications to project management’, *Structure and Infrastructure Engineering*, March, Vol. 3, No. 1, pp.53–60.

Ernst, A.T., Jiang, H., Krishnamoorthy, M. and Sier, D. (2004) ‘Staff scheduling and rostering: a review of applications, methods and models’, *European Journal of Operational Research*, Vol. 153, No. 1, pp.3–27.

Erschler, J.F., Roubellat, J.P. and Vernhes, R. (1976) ‘Finding some essential characteristics of the feasible solutions for a scheduling problem’, *Operations Research*, Vol. 24, No. 4, pp.774–783.

Eusuff, M. and Lansey, K.E. (2003) ‘Optimization of water distribution network design using the shuffled frog leaping algorithm’, *Journal of Water Resource Planning and Management*, Vol. 129, No. 3, pp.210–225.

Eusuff, M., Lansey, K. and Pasha, F. (2006) ‘Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization’, *Engineering Optima*, Vol. 38, No. 2, pp.129–154.

Faaland, B. et al. (1993) ‘Cost based scheduling of workers and equipment in fabrication and assembly shop’, *Operational Research*, Vol. 41, No. 2, pp.253–268.

Fang, C. and Wang, L. (2012) ‘An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem’, *Journal Computers and Operations Research*, May, Vol. 39, No. 5, pp.890–901.

Farahani, M., Movahhed, S.B. and Ghaderi1, S.F. (2010) ‘A hybrid meta-heuristic optimization algorithm based on SFLA’, *2nd International Conference on Engineering Optimization*, 6–9 September, Lisbon, Portugal.

Freling, R. et al. (2003) ‘Models and algorithms for integration of vehicle and crew scheduling’, *Journal of Scheduling*, Vol. 6, No. 1, pp.63–85.

French, S. (1982) *Sequencing and Scheduling – An Introduction to the Mathematics of the Job-Shop*, Ellis Horwood, John-Wiley & Sons, New York.

Garey, M. et al. (1976) 'The complexity of flow shop and job shop scheduling', *Mathematics of Operations Research*, Vol. 1, No. 2, pp.117–129.

Gecevska, V. et al. (2006) 'Modeling of manufacturing activities by process planning knowledge representation', *International Journal of Simulation Model*, Vol. 5, No. 2, pp.69–81, ISSN: 1726-4529.

Ghoniem, A.S. (2002) *Static and Dynamic Job-shop Scheduling Using Rolling Horizon Approaches and the Shifting Bottleneck Procedure*, Thesis, Master of Science in Industrial and Systems Engineering.

Guyon, O. et al. (2010) 'Cut generation for an integrated employee timetabling and production scheduling problem', *European Journal of Operational Research*, Vol. 201, No. 2, pp.557–567.

Guyon, O., Lemaire, P., Pnson, E. and Rivreau, D. (2009) 'Near optimal and optimal solutions for an integrated employee timetabling and production scheduling problem', *13th IFAC Symposium on Information Control Problems in Manufacturing*, Moscou, Russian Federation.

Hooler, J.N. (2005) 'A hybrid method for planning and scheduling', *Constraints*, Vol. 10, No. 4, pp.385–401.

Huq, F. et al. (2004) 'Employee scheduling and makespan minimization in flow shop with multi-processor workshop: a case study', *Omega*, Vol. 32, No. 2, pp.121–129.

Kragelund, L. (1997) 'Solving a timetabling problem using hybrid genetic algorithms', *Software | Practice and Experience*, Vol. 27, No. 10, pp.1121–1134.

Meisels, A. and Schaerf, A. (2003) 'Modelling and solving employee timetabling problems', *Annals of Mathematics and Artificial Intelligence*, Vol. 39, Nos. 1–2, pp.41–59, Kluwer Academic Publishers, The Netherlands.

Pan, Q-K., Wang, L., Gao, L. and Li, J. (2011) 'An effective shuffled frog-leaping algorithm for lot-streaming flow shop scheduling problem', *Int. J. Adv. Manufacturing Technology*, February, Vol. 52, Nos. 5–8, pp.699–713.

Pezzella, F. and Merelli, E. (2000) 'A tabu search method guided by shifting bottleneck for the job shop scheduling problem', *European Journal of Operational Research*, Vol. 120, No. 2, pp.297–310.

Schaerf, A. et al. (2006) 'Measurability and reproducibility in timetabling research: state-of-the-art and discussion', *International Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*, pp.53–62.

Shmoys, D.B., Stein, C. and Wein, J. (1994) 'Improved approximation algorithms for job shop scheduling problems', *SIAM Journal of Computer Science*, June, Vol. 23, No. 3, pp.617–632.

Silva, J.D.L. et al. (2004) 'An introduction to multiobjective metaheuristics for scheduling and timetabling', in Gandibleux, X., Sevaux, M., Sorensen, K. and T'Kindt, V. (Eds.): *Metaheuristics for Multiobjective Optimisation*, Springer Lecture Notes in Economics and Mathematical Systems, Vol. 535, pp.91–129.

Valls, V. et al. (1996) 'A graph colouring model for assigning a heterogeneous workforce to a given schedule', *European Journal of Operational Research*, Vol. 90, No. 2, pp.285–302.

Yang, S. et al. (2001) *A New Adaptive Neural Network and Heuristics Hybrid Approach for Job-Shop Scheduling*, National Nature Science Foundation (No. 69684005) and National High-Tech Program of P.R. China (No. 863-511-9609-003), Vol. 28, No. 10, pp.955–971.

Zaccaro, S.J. et al. (2001) 'Team leadership', *The Leadership Quarterly*, Vol. 12, No. 4, pp.451–483.