International Journal of Computational Intelligence Systems, Vol. 5, No. 6 (November, 2012), 1184-1199

Comparison of Firefly algorithm and Artificial Immune System algorithm for lot streaming in *m*-machine flow shop scheduling

G.Vijay Chakaravarthy*
Department of Mechanical Engineering, Fatima Michael College of Engineering & Technology, Madurai - 625 020, India.

S. Marimuthu
Department of Mechanical Engineering, Latha Mathavan Engineering College, Madurai - 625 301, India.

A. Naveen Sait
Department of Mechanical Engineering, Chendhuran College of Engineering & Technology, Pudukkottai- 622 507, India.

Received 11 March 2012
Accepted 7 October 2012

Abstract
Lot streaming is a technique used to split the processing of lots into several sublots (transfer batches) to allow the overlapping of operations in a multistage manufacturing systems thereby shortening the production time (makespan). The objective of this paper is to minimize the makespan and total flow time of *n*-job, *m*-machine lot streaming problem in a flow shop with equal and variable size sublots and also to determine the optimal sublot size. In recent times researchers are concentrating and applying intelligent heuristics to solve flow shop problems with lot streaming. In this research, Firefly Algorithm (FA) and Artificial Immune System (AIS) algorithms are used to solve the problem. The results obtained by the proposed algorithms are also compared with the performance of other worked out traditional heuristics. The computational results shows that the identified algorithms are more efficient, effective and better than the algorithms already tested for this problem.

Keywords: Flow shop; Lot streaming; Scheduling; Firefly Algorithm; Artificial Immune System Algorithm

Nomenclature

FA	Firefly Algorithm
AIS	Artificial Immune System algorithm
F_1	completion time for first job
F_2	completion time for second job
i	machine
j	job
m	number of machines
MP	makespan
n	number of jobs
n_j	number of sublots of job *j*
P_{ij}	processing time for job *j* on machine *i*
T_{max}	total flow time for generated sequence
P_{ij}	processing time of job *j* on machine *i*
C_{ij}	completion time of job *j* on machine *i*
T_{ij}	total flow time of job *j* on machine *i*
S	initial sequence
S''	generated sequence
S_{ij}	setup time for job *j* on machine *i*
TFT	total flow time
Δ_i	idle time on the machine *i*
$C_{max(s)}$	makespan for the sequence *s*
$C_{max(s')}$	makespan for the sequence *s'*
L_j	number of sublot

* Author for correspondence
Email: vijay_vanitha123@yahoo.com

Published by Atlantis Press

1. Introduction

This paper concentrates on solving the lot streaming problem in flow shop scheduling systems. The flow shop problem can be briefly described as follows: A set of jobs and a set of machines are given. Each job consists of a sequence of operations, which need to be processed during an uninterrupted time period of a given length on a given machine. A schedule is an allocation of the operations to time intervals on the machines. In a traditional flow shop, each job must be processed on every machine and all jobs must follow the same machine sequence (route). One of the common restrictions made in most research studies is that a job cannot be transferred to the next machine before its processing is finished. This need not be the case in many practical situations because a job may be split in to a number of smaller sub lots. When a sub lot of the job is completed, it can be immediately moved to the next machine. By splitting jobs, the idle time on successive machines can be reduced. The process of splitting jobs into sub lots is usually called "Lot Streaming" which was first introduced by Reiter[1]. So, Lot streaming represents the concept of dividing a lot into multiple smaller sublots, so that they can be transferred to the next stage immediately upon their completion. For the application of Firefly algorithm and Artificial Immune System algorithm, as a result of operation overlapping, idling time of machines, makespan and total flow time can be substantially reduced. Comparison of experimental results with other meta heuristics have clearly shown the competence of the FA and AIS algorithms in solving flow shop scheduling problems and the improvement in optimal solutions.

The remainder of the paper is organized as follows: In the subsequent section, describes the literature review and section 3 explains the problem statement. Section 4 addresses the determination of schedules applying Firefly algorithm. Section 5 addresses the determination of schedules applying Artificial Immune System algorithm. Section 6 provides the numerical illustration of the proposed algorithms. Section 7 provides a detailed analysis of computational results. Brief conclusions are summarized in section 8.

2. Literature Review

Many researchers have attempted on different directions to solve lot streaming problems using various techniques. But the usage of intelligent algorithms proved to be an effective tool for solving these types of problems. The details of some of the major research work carried out to solve lot streaming problems are discussed and presented in detail in this section. Some studies showed that lot streaming can significantly improve the schedule performance with respect to the makespan as reported by Baker and Pyke[2]. Previous studies considering two-stage or special cases of three-stage flow shop lot streaming can be found in Potts and Baker[3], Vickson and Alfredsson[4], Glass *et al.*[5], Chen and Steiner[6], Sriskandarajah and Wagneur[7].Chao- Tang Tseng and Ching- Jong Liao[8]attempted to solve flow shop scheduling problems based on weighted earliness and tardiness by proposing a new algorithm called "Discrete Particle Swarm Optimization algorithm (DPSO)". Rahime Sancar Edis and M. Arslan Ornek[9] considered consistent sublot types and discrete sublot sizes and presented the combination of simulation and tabu search with the objective of minimizing makespan. This work proved that heuristic algorithms provide efficient results compared to the deterministic models. Quan- Ke Pan *et al.*[10] presented Shuffled Frog Leaping Algorithm (SFLA) for solving a lot-streaming flow shop scheduling problem with equal-size sublots, where a criterion is to minimize makespan under both an idling and no-idling production cases. Serdar Birogul *et al.*[11] examined how the lot streaming affects both the Gantt scheme and the genetic algorithm, and how to adapt the Hybrid Genetic Algorithms (HGA) to job shop scheduling problems. An ant based algorithm for solving multi-level lot sizing problems based on the concept of MAX-MIN ant system was proposed and evaluated by Rapeepan Pitakaso *et al.*[12].Marimuthu *et al.*[13] proposed two meta heuristics, namely Simulated Annealing algorithm (SA) and Tabu Search algorithm (TS), to evolve the optimal sequence for makespan and total flow time criteria in an *m*-machine flow shop with lot streaming. Marimuthu *et al.*[14] addressed two more evolutionary algorithms namely, Genetic Algorithm (GA) and Hybrid Evolution Algorithm (HEA) to evolve best sequence for makespan/total flow time criterion for *m*-machine flow shop involved with lot streaming and set-up time. Marimuthu *et al.*[15] introduced Ant Colony Optimization algorithm (ACO) and Threshold Accepting algorithm (TA) to evolve best sequence for makespan/total flow time criterion for *m*-machine flow shop involved with lot streaming and setup time. Firefly Algorithms for Multi model Optimization was introduced by Xin -she Yang[16] who was inspired by firefly behaviours. Mohammad Kazem Sayadi *et al.*[17] proposed a new discrete firefly meta-heuristic to minimize the makespan for the permutation flow shop scheduling problem and compared with existing ant colony optimization technique. The results indicated that the new proposed technique performs better than the existing method.

Liu, S.C.[18] addressed a heuristic method for discrete lot streaming with variable sublots to determine a continuous solution (sublots with real values) for variable lot streaming and deriving a discrete solution by rounding up the continuous solution. Shu-Chu Liu *et al.*[19] introduced the multi-product variable lot streaming (MPVLS) in a flow shop is to determine product sequence and to determine lot streaming for each machine, in order to minimize makespan. Fantahun M. Defersha and Mingyuan Chen[20] developed a mathematical programming model and a hybrid genetic algorithm for *n*-job *m*-machine lot streaming problems with variable sublots considering setup times. Biskup. D., Feldmann.M[21] presented a mixed integer programming formulation to split a given lot into sublots so as to allow their overlapping production in a flow shop environment. Computational results confirmed that the exploitation of variable sublots were advantageous and may lead to a significant increase in productivity. Fantahun M. Defersha, Mingyuan Chen[22] developed a hybrid genetic algorithm for one-job *m*-machine lot streaming problems with variable sublots and setup. Computational results showed that the performance of the proposed genetic algorithm was encouraging.

Ranga V. Ramasesh *et al.*[23] presented an economic production lot size model with lot streaming to minimize the total relevant cost. Subhash C. Sarin *et al.*[24] presented a polynomial-time procedure for determining the number of sublots of a single-lot, multiple-machine flow shop lot-streaming problem in order to minimize makespan, mean flow time, work-in-process where sub lot-attached setup and transfer times. Suk-Hun Yoon, Jose A. Ventura[25] presented linear programming formulations for minimizing the mean weighted absolute deviation from due dates for lot streaming under flow shop environment.

Jiang Chen, George Steiner[26] presented two quickly obtainable approximations of very good quality for the discrete lot streaming problem in flow shops.

The literature review reveals that *m*- machine flow shop under lot sizing is one of the active areas of research. From the extensive literature survey carried out, it is identified that FA and AIS algorithms are not used to solve the lot streaming problems. It is observed that only few papers addressed that lot streaming in flow shops with variable sublots. This research gap is bridged by applying the FA and AIS algorithms for the flow shop lot streaming problem with equal and variable size sublots with the objective of minimizing the makespan and total flow time and to determine the optimal sublot size.

3. Problem Statement

The sequencing and scheduling problem considered in this paper is *n* job – *m* machine flow shop scheduling problem with equal and variable lot streaming. Statement of the problem, Illustrative example and mathematical formulation of the problem are described in this section.

3.1 Statement of the Problem (equal sublots)

Figure 1 shows an example schedule of 2 jobs, each with 3 equal sublots in the sequence 1-2 being processed through three machines. The first job completes its process at time F_1 and the second job at time F_2 in the schedule. The sum of F_1 and F_2 is the total flow time value of the schedule and the maximum of F_1 or F_2 thus becomes the makespan (*MP)* of the schedule.

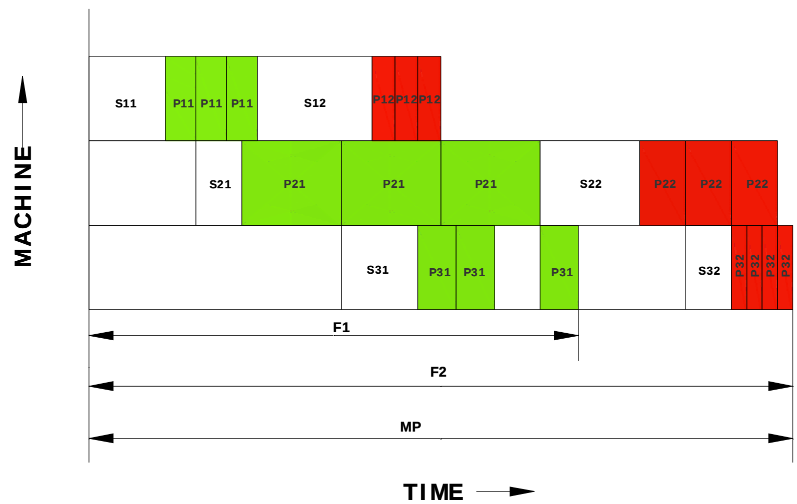

Fig 1-Scheduling 2 jobs through 3 machines with 3 equal size sublots in the sequence 1-2

The problem of *m*-machine flow shop with equal lot streaming and setup time can be stated as:

Let P_{ij} denote the processing time of the job *j* on machine *i* $(1 \leq i \leq m,\ 1 \leq j \leq n)$. Also let X_{jk} $(1 \leq j \leq n,\ 1 \leq k \leq n_j)$ denote size of sublot k on machine *i*. Thus, the processing time of this sub lot is $P_{ij}X_{jk}$. The sub lots are consistent if $X_{jk} = x_{j,k+1}$ *for* $1 \leq j \leq n,\ 1 \leq k \leq n_j$, X_{jk} and $X_{j,k+1}$ contain the same items; otherwise the sublots are variable. Let Δ_i be the idle time on the machine *i* and n_j be the number of sublots of job *j*. Determination of optimal makespan time and total flow time of a sequence of *n*- jobs available at time zero in a *m*- machine flow shop in which first sub lot of j^{th} job is set on machine *i* on its arrival with the setup time of S_{ij} and all the number of equal size sub lots n_j of job *j* are processed continuously on machine *i.*

Makespan (*MP*) for 3 machine 2 job problem (figure.1) is written as follows:

$$MP = \sum_{j=1}^{2} S_{3j} + \sum_{j=1}^{2} P_{3j}\ n_j\ + \Delta_3 \text{ ....(1)}$$

3.2 Statement of the Problem (variable sublots)

The problem considered in this section can be described as follows: there are *n* jobs and *m* machines in a flow shop. Each job *j* □ *J* = {1,2,…*n*} will be sequentially processed on *m* machines and the job sequence is the same on each machine *i* □ *M* = {1,2,…*m*}. In order to reduce the lead time and to accelerate the production, each job *j* can be split into number of sublots with variable size. Once the processing of a sublot on a machine is completed, it can be transferred to the downstream machine immediately.

Similarly, all the sublots of job j should be processed continuously. At any time, each machine can process at most one sublot and each sublot can be processed on at most one machine. Let the processing time of each sublot of job j on machine i be P_{ij}, and the setting of first sublot of each job j on machine i consumes a time of S_{ij} on its arrival to that machine. Given that the release time of all jobs is zero, and sublot transportation time is included in the processing time, then the objective is to find a sequence with the optimal sublot to minimize the makespan and total flow time. Let C_{ij} be the completion time of job j in machine i , T_{ij} be the total flow time of job j in machine i and $A_{j(i+1)}$ be the arrival time of j^{th} job on $(i+1)$ machine. Let Z be the makespan objective function and Y be the total flow time objective function, the model is

Minimize,

$$Z \geq C_{ij}, \forall\, i, j \quad (2)$$

$$Y \geq T_{ij}, \forall\, i, j \quad (3)$$

Subject to

$$A_{ij} + S_{ij} + P_{ij} \geq Z, \quad \forall\, i, j \quad (4)$$

$$\Sigma\left(A_{ij} + S_{ij} + P_{ij}\right) \geq Y, \quad \forall\, i, j = m \quad (5)$$

$$A_{ij} + S_{ij} + P_{ij} \geq A_{j(i+1)}, \quad \forall\, i, j \quad (6)$$

$$S_{ij} \geq 0 \quad \forall\, i, j \quad (7)$$

$$P_{ij} \geq 0 \quad \forall\, i, j \quad (8)$$

$$C_{ij} \geq 0 \quad \forall\, i, j \quad (9)$$

$$T_{ij} \geq 0 \quad \forall\, i, j \quad (10)$$

Equation (2) shows that makespan is greater than or equal to completion time of job j in machine i. Equation (3) shows that total flow time is greater than or equal to completion time of job j in machine i. Equation (4) calculates a schedule of jobs that minimizes makespan. Equation (5) determines a schedule of jobs that minimizes total flow time. Equation (6) computes arrival time of current job sequence to be processed is greater than completion time of previous job sequence. Equation (7) shows setup time of all job j on all machine i is greater than or equal to zero. Equation (8) shows processing time of all job j on all machine i is greater than or equal to zero. Equation (9) shows completion time of job j on machine i is greater than or equal to zero. Equation (10) shows total flow time of job j on machine i is greater than or equal to zero.

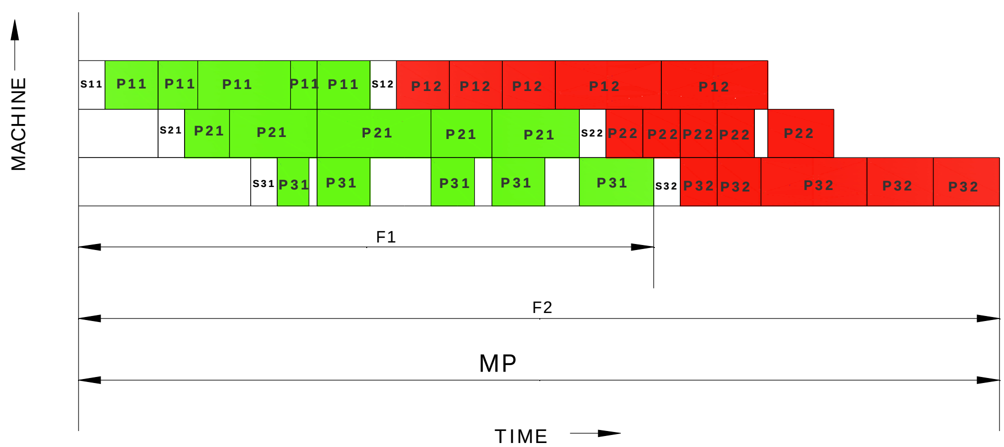

Fig.2 -An example showing 5 sublots, two Jobs being processed through three machines

4. Firefly Algorithm

4.1Introduction

Firefly algorithm (FA) is an intelligent metaheuristic algorithm, inspired by the flashing behavior of fireflies. The Firefly Algorithm (FA) is a population-based technique to find the global optimal solution based on swarm intelligence, investigating the foraging behavior of fireflies. The flashing signal by fireflies is to attract mating partners and preys and share food with others [27]. The swarm of fireflies will move to brighter and more attractive locations by the flashing light intensity that associated with the objective function of problem considered in order to obtain efficient optimal solutions. The development of firefly-inspired algorithm was based on three idealized rules[28]: i) artificial fireflies are unisex so that sex is not an issue for attraction; ii) attractiveness is proportional to their flashing brightness which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. Since the most attractive firefly is the brightest one, to which it convinces neighbours moving toward. In case of no brighter one, it freely moves any direction; and iii) the brightness of the flashing light can be considered as objective function to be optimized. In this work, the evaluation on the goodness of schedules is measured by the makespan, which can be calculated using equation (11), where C_k is completed time of job *k*.

Minimize: $C_{max} = max\ (C_1, C_2 \ldots C_k)$ ……… (11)

The distance between any two fireflies *i* and *j* at x_i and x_j, respectively, can be defined as a Cartesian distance (r_{ij}) using equation (12), where $x_{i,k}$ is the k^{th} component of the spatial coordinate x_i of the i^{th} firefly and *d* is the number of dimensions[16].

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \quad (12)$$

The calculation of attractiveness function of a firefly is shown in equation (13),

$$\beta_{(r)} = \beta_0 \times exp\ (-\gamma r^m), \text{ with } m \geq 1 \quad (13)$$

The movement of a firefly *i* which is attracted by a more attractive (i.e., brighter) firefly *j* is given by the following equation (14),

$$x_i = x_i + \beta_0 \times exp\ (-\gamma r_{ij}^2) \times (x_j - x_i) + \alpha\ (rand - \tfrac{1}{2}) \quad (14)$$

The settings of FA parameters: Light absorption coefficient (γ) = 1.0, Randomization parameter (α) = 0.3, Attractiveness value (β_0) = 1.0 and rand= 0.2.

4.2 General Schema of FA

Objective function makespan, z = (C_1, C_2...C_n)
Generate machines (i = 1, 2... m)
Generate job sequences (j = 1, 2... n)
Evaluate makespan (C_1, C_2, C_3….C_n) for all population
While Gen. < Max Gen.
    For each job sequence j = (1, 2, 3…, n)
        For each lot sizes (X_{ij})
            Move firefly in d-dimensional space
            Determine the attractiveness based on distance r_{ij}
            Evaluate makespan
        End For
    End For
    Assess light intensity
    Select job sequences and lot sizes for Gen. +1
End While
Rank and choose the best job sequences and sublot sizes

5. Artificial Immune System (AIS) algorithm

5.1 Introduction

Immune Algorithm (IA) or Artificial Immune System (AIS) is the recently developed evolutionary technique inspired by the theory or immunology or immune system. According to Castro and Zuben[29] AIS is defined as ‘‘An abstract or metamorphic computational system using the ideas gleaned from the theory and components of immunology’’. AIS emulate the immune system in general and Clonal selection in particular. The artificial Immune system is built around the two principles of immune system. They are a) Clonal selection principle, b) Affinity maturation principle.

In Clonal selection principle, each sequence (antibody) has a makespan (overall completion time) value which refers to the affinity of the antibody. Affinity value of each sequence is calculated from affinity function given as:

$$Affinity\ (p) = \frac{1}{makespan}$$

So a lower makespan value gives higher affinity value. Further cloning in antibodies is done directly proportional to their affinity function values. So antibodies with lower makespan values will generate more clones. An affinity function is defined based on the makespan value of the sequence.

The two methods employed in Affinity Maturation Principle are mutation and receptor editing. A two phased mutation procedure were used for the generated clones.

a) Inverse mutation b) Pair wise interchange mutation

5.2 General Schema of AIS

```
Objective function makespan, z = (C1, C2...Cn)
Generate machines (i = 1, 2... m)
Generate job sequences (j = 1, 2... n)
Evaluate makespan (C1, C2, C3….Cn) for all population
While Gen. ≤ Max Gen.
    For each job sequence j = (1, 2, 3…, n)
        For each lot sizes (Xij)
            If random number ≤ 0.8
                Recombine and inverse mutate the offspring
            End If
            If random number ≤ 0.4
                Perform pair wise mutation the offspring
            End If
                Evaluate makespan
        End For
    End For
    Assess affinity function
    Select job sequences and lot sizes for Gen. +1
End While
Rank and choose the best job sequences and sublot sizes
```

6. Numerical Illustration

The results of the algorithms for an example problem (5job–2machine) are given in this section. Table 1 & 2 provides the data considered for equal and variable sublots in the example problem.

Table 1 - Data of 5 job - 2 machine (equal sublots)

Problem Data	*Jobs*				
	1	*2*	*3*	*4*	*5*
n_j	2	2	3	2	2
S_{1j}	2	3	4	2	3
P_{1j}	2	3	3	2	2
S_{2j}	2	3	2	3	4
P_{2j}	3	2	3	1	2

Table 2 - Data of 5 Jobs – 2 Machines (variable sublots)

Problem data	*Jobs*				
	1	*2*	*3*	*4*	*5*
Number of jobs(n_j)	10	8	12	9	6
Number of sublot (L_j)	2	4	3	4	3
Sublot size	{64}	{2132}	{453}	{2133}	{231}
S_{1j}	2	3	2	3	2
P_{1j}	1	2	1	2	1
S_{2j}	2	3	2	2	3
P_{2j}	2	1	1	2	2

6.1FA for makespan criterion

Table 3 illustrates FA for equal sublots, how the reproduced sequences (new population) are evolved from the seed sequence (old population) and also the determination of makespan.

Table 3 - FA Illustration (equal sublots)

Seed Sequence	*Distance (r_{ij})*	*Attractiveness (β_r)*	*Movement (x_i)*	*Sequence for next generation*	*makespan*
2 3 5 1 4	0.6	0.55	0.9986	2 3 5 1 4	48
3 2 5 1 4	0.7	0.497	1.6388	3 2 5 1 4	48
2 4 3 5 1	0.4	0.6703	0.4692	2 4 3 5 1	45
1 5 2 3 4	1.2	0.3012	2.2943	**1 2 5 3 4**	**45***

The final job sequence is 1- 2 -5 -3- 4, the corresponding makespan time is 45 and it is shown in Figure 3.

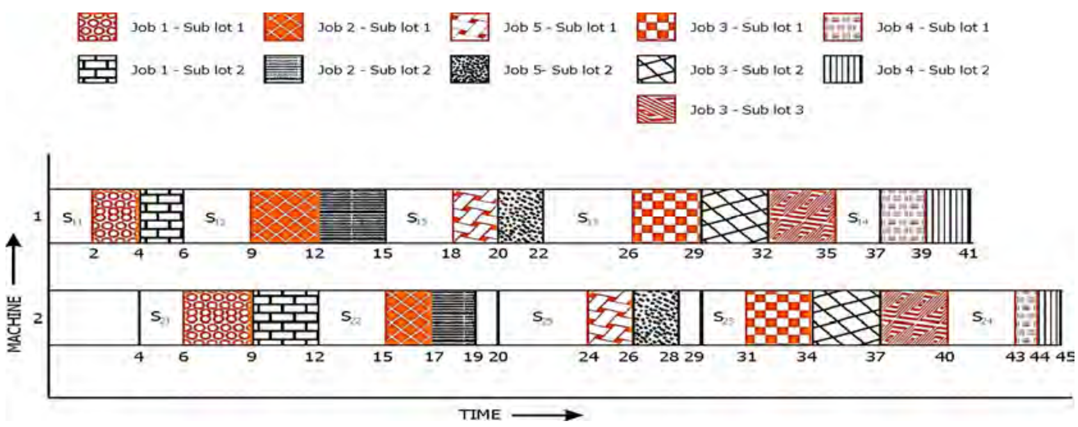

Fig.3 - Schedule for the sequence 1-2-5-3-4

Table 4 illustrates FA for variable size sublots, how the reproduced sequences (new population) are evolved from the seed sequence (old population) and also the determination of makespan with sublot size.

Table 4 - FA Illustration (variable sublots)

Seed Sequence	*Distance (r_{ij})*	*Attractiveness (β_r)*	*Movement (x_i)*	*Sequence for next generation*	*makespan*	*Sublot size*
5 4 2 3 1	1.2	0.3012	2.2943	**5 3 1 2 4**	**85***	**{132}{543}{46}{3212}{1233}**
3 1 4 5 2	0.4	0.6703	1.2692	3 1 4 5 2	88	{453}{64}{2133}{231}{2132}
4 5 1 3 2	0.2	0.8187	1.2178	4 5 1 3 2	89	{2133}{231}{64}{453}{2132}
3 4 1 2 5	0.3	0.7408	0.8358	3 4 1 2 5	89	{453}{2133}{64}{2132}{231}

The final result is 5-3-1-2-4, the corresponding makespan time is 85, Sublot Size is {132}{543}{46}{3212}{1233} and it is shown in figure 4.

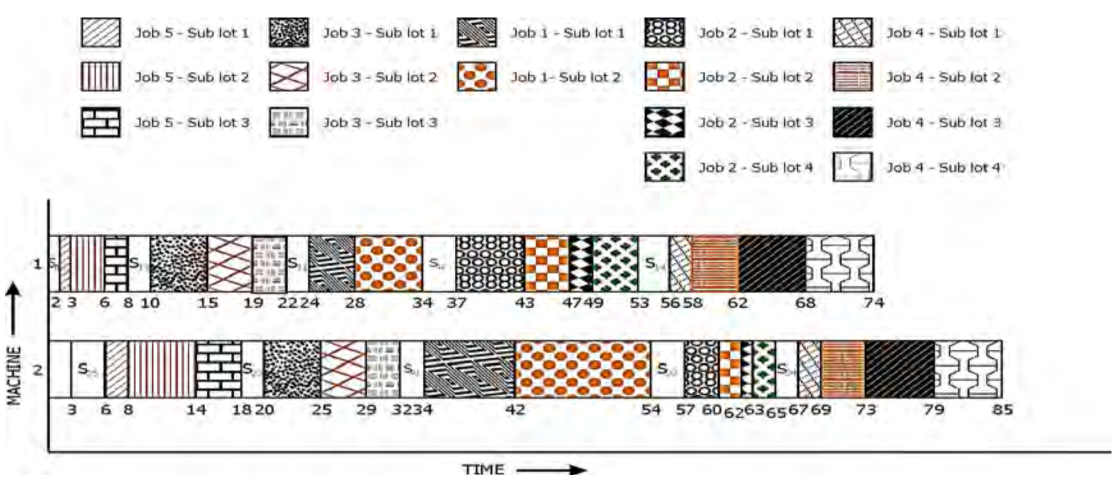

Fig.4 - Schedule for the sequence 5-3-1-2-4

6.2 AIS for makespan criterion

Table 5 illustrates AIS algorithms for equal sublots, how the reproduced sequences (new population) are evolved from the seed sequence (old population) and also the determination of makespan.

Table 5 - AIS Illustration (equal sublots)

Seed sequence	*makespan*	*Affinity (1/z)*	*mutation* Inverse mutation probability ≤ 0.8 RN	Y/N	*Inverse mutation*	*Pair wise mutation probability ≤ 0.4* RN	Y/N	*Pair wise mutation*	*makespan*
3 2 1 4 5	48	0.0208	0.78	Y	2 3 1 5 4	0.3	Y	5 4 1 2 3	**46***
2 1 3 4 5	47	0.0213	0.9	N	-	0.56	N	2 1 3 4 5	47
3 2 5 1 4	48	0.0208	0.6	Y	3 5 2 1 4	0.28	Y	2 1 3 5 4	46
2 3 5 1 4	48	0.0208	0.58	Y	2 5 3 1 4	0.35	Y	3 1 2 5 4	46

The final job sequence is 5- 4 -1 -2- 3, the corresponding makespan time is 46 and it is shown in Figure 5.

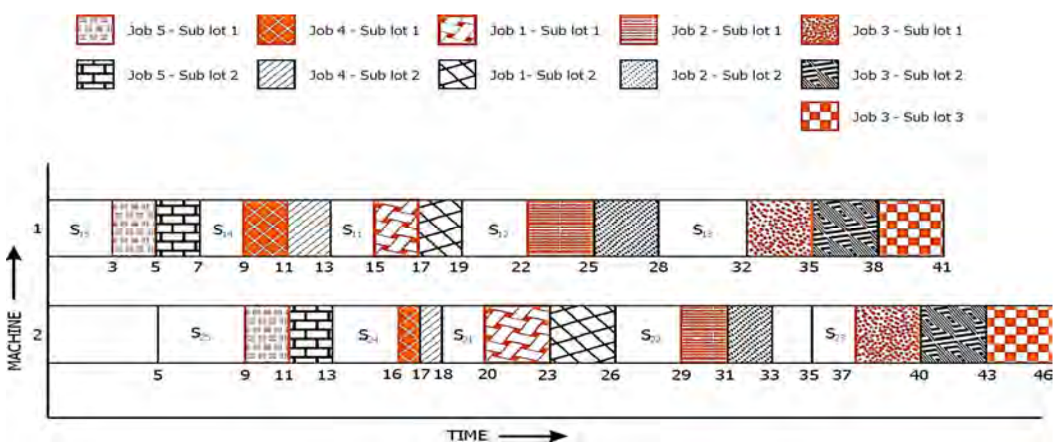

Fig.5 - Schedule for the sequence 5-4-1-2-3

Table 6 illustrates AIS algorithms for variable size sublots, how the reproduced sequences (new population) are evolved from the seed sequence (old population) and also the determination of makespan with sublot size.

Table 6 - AIS Illustration (variable sublots)

Seed sequence	*makespan*	*Affinity (1/z)*	*mutation* Inverse mutation probability ≤ 0.8 RN	Y/N	*Inverse mutation*	*Pairwise mutation probability ≤ 0.4* RN	Y/N	*Pair wise mutation*	*makespan*	*Sublot size*
2 1 4 5 3	90	0.01111	0.6	Y	1 2 4 3 5	0.3	Y	3 5 1 2 4	**87***	**{354}{213}{46}{3221}{1323}**
2 5 4 3 1	90	0.01111	0.48	Y	5 2 3 4 1	0.28	Y	3 4 5 2 1	88	{453}{2133}{231}{2132}{64}
5 1 2 3 4	90	0.01111	0.5	Y	1 5 2 4 3	0.25	Y	1 4 3 5 2	88	{46}{1233}{345}{123}{1223}
3 4 5 2 1	89	0.01123	0.9	N	-	0.5	N	3 4 5 2 1	89	{453}{2133}{231}{2132}{64}

The final result is 3-5-1-2-4, the corresponding makespan time is 87, Sublot Size is {354}{213}{46}{3221}{1323} and it is shown in figure 6.

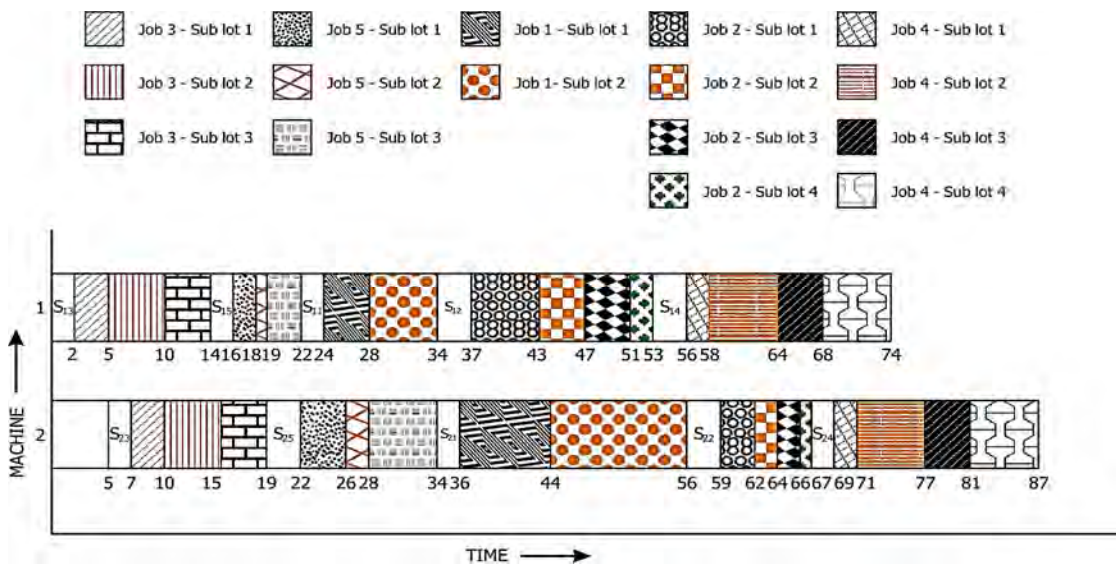

Fig.6 - Schedule for the sequence 3-5-1-2-4

7. Performance Analysis

7.1 Test problem instances generation

Different test problems of several job size and machine are generated with the prescribed bounds for equal and variable size sublots are presented in the Table 7. The results for FA and AIS are simulated with the help of a C++ program on a Core i3 Processor system of 3.10GHz and 4 GB RAM. The processing time is much less than a minute.

Table 7 – Upper and Lower bound for problem data generation

Problem data	*Lower bound*	*Upper bound*
n_j	1	30
L_j	1	9
S_{jk}	1	9
P_{jk}	1	9

7.2 Discussions and Analysis

AIS based on a genetic algorithm extended by a search technique to further improve individual's fitness that may keep high population, diversity and reduce the likelihood premature convergence. Our objective is to determine the performance of FA in comparison with AIS; our experimental result shows that FA is superior to AIS. The second comparison was made upon the search space. From the extensive experiments, it was found that AIS can be seemed to provide almost minimum makespan and total Flow time is achieved in shorter time. Results indicate that FA is extremely powerful technique and the most efficient algorithm for the *n*-job, *m*-machine lot streaming problem in a flow shop with equal and variable size sublots.

The convergence speed of FA is greater than the AIS. AIS converge faster while FA searches the solution space with better accuracy. However both AIS and FA provide better solution than the already tested algorithms. AIS is very easy to implement and it requires very little parameter adjustments. The minimal makespan value is obtained with increase in number of generations for AIS. The result shows that FA produces clear and consistent superior results. With AIS, there is a good tradeoff between speed and avoidance of premature convergence. The solution is obtained with specific number of iteration in the proposed algorithms. AIS provide better solution with increase in the number of generations evaluated when compared with FA. The best optimal makespan value and total flow time value could be obtained by fine tuning the parameters.

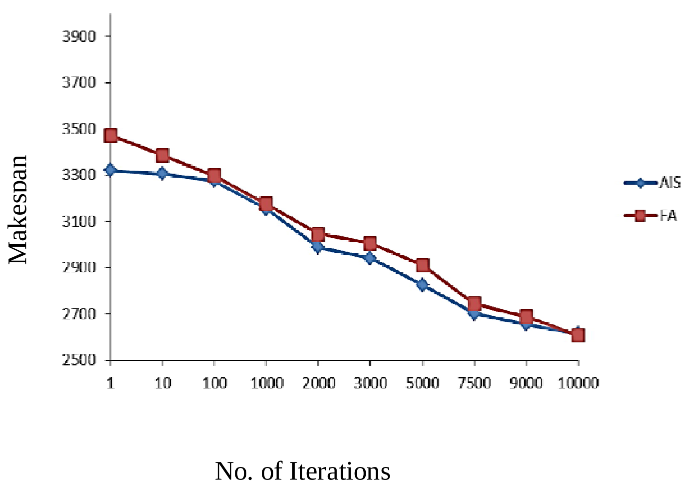

Fig.7–Convergence graph for 30×10 variable sublot sizes problem (makespan performance results)

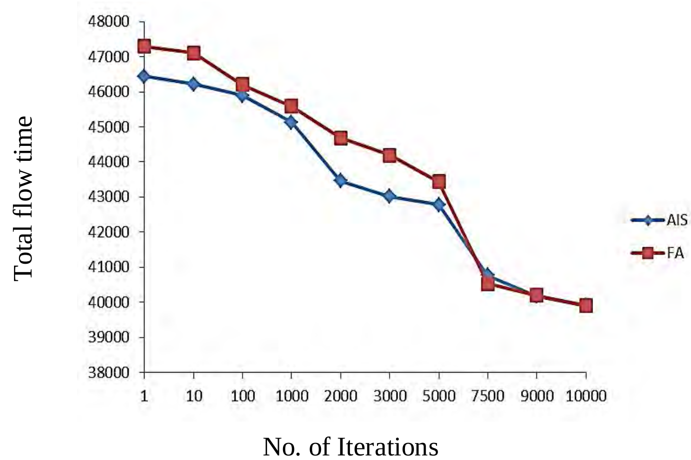

Fig.8–Convergence graph for 30×10 variable sublot sizes problem (total flow time performance results)

7.3 Performance evaluation of FA and AIS

In this section, the computational results of proposed FA and AIS for equal size sublots (table 8- makespan criterion, table 9- total flow time criterion) are compared with the results of Particle Swarm Optimization (PSO) and Differential Evolution Algorithm (DEA) (Vijay Chakaravarthy *et al.*[30]). The computational results of proposed FA and AIS for variable size sublots (table 10- makespan criterion, table 11- total flow time criterion) are compared with the results of Genetic Algorithm (GA) and Memetic Algorithm (MA) (Marimuthu *et al.*[31]). It is clear from the tables, that the proposed FA and AIS outperforms the other compared algorithms at the same computational time. We also find that FA provides higher quality results for makespan and total flow time, compared with AIS in given number of iterations.

Table 8 – Comparison between FA, AIS, DEA and PSO for equal size sublots (makespan)[30]

m × n	*Problem Set*	*makespan*							
		PSO	*DEA*	*FA*	*% Deviation from PSO*	*% Deviation from DEA*	*AIS*	*% Deviation from PSO*	*% Deviation from DEA*
3×10	1	796	796	796*	0	0	796*	0	0
	2	691	691	691*	0	0	691*	0	0
	3	625	625	625*	0	0	625*	0	0
	4	623	623	623*	0	0	623*	0	0
5×20	1	854	851	843*	1.30	0.95	845	1.07	0.71
	2	749	748	745*	0.54	0.40	745*	0.54	0.40
	3	634	638	625*	1.44	2.08	625*	1.44	2.08
	4	677	677	677*	0	0	677*	0	0
7×30	1	896	888	872*	2.75	1.83	881	1.70	0.79
	2	778	766*	765	1.70	0.13	769	1.17	0.39
	3	709	702	694*	2.16	1.15	698	1.58	0.57
	4	697	690	685*	1.75	0.73	685*	1.75	0.73

*refers to minimum makespan

Table 9 – Comparison between FA, AIS, DEA and PSO for equal size sublots (total flow time)[30]

m × n	*Problem Set*	*Total flow time*							
		PSO	*DEA*	*FA*	*% Deviation from PSO*	*% Deviation from DEA*	*AIS*	*% Deviation from PSO*	*% Deviation from DEA*
3×10	1	11135	11205	11283	1.33	0.70	10978*	1.43	2.07
	2	8994	8794	8603*	4.54	2.22	8785	2.38	0.10
	3	8644	8743	8520*	1.46	2.62	8552	1.08	2.23
	4	8555	8466	8374	2.16	1.10	8161*	4.83	3.74
5×20	1	12871	12847	12628*	1.92	1.73	12831	0.31	0.12
	2	10527	10532	10232	2.88	2.93	10117*	4.05	4.10
	3	9759	9598*	9647	1.16	0.51	9636	1.28	0.40
	4	9836	9874	9718*	1.21	1.61	9770	0.68	1.06
7×30	1	13767	13696	13530*	1.75	1.23	13538	1.69	1.17
	2	11253	11391	11300	0.42	0.81	10945*	2.81	4.07
	3	10757	10838	10748*	0.08	0.84	10801	0.41	0.34
	4	10682	10748	10412	2.59	3.23	10335*	3.36	3.99

*refers to minimum total flow time

Table 10 – Comparison between FA, AIS, GA and MA for variable size sublots (makespan)[31]

m × n	*Problem Set*	*makespan*							
		GA	*MA*	*FA*	*% Deviation from GA*	*% Deviation from MA*	*AIS*	*% Deviation from GA*	*% Deviation from MA*
3×10	1	786	779	776*	1.29	0.39	776*	1.29	0.39
	2	3496	3480	3472*	0.69	0.23	3475	0.60	0.14
	3	3511	3510	3498*	0.37	0.34	3498*	0.37	0.34
	4	3706	3706	3706*	0	0	3706*	0	0
5×20	1	1399	1368	1357*	3.10	0.81	1357*	3.10	0.81
	2	7324	7288*	7322	0.03	0.47	7320	0.06	0.44
	3	7268	7075	7062*	2.92	0.18	7062*	2.92	0.18
	4	7359	7354	7350*	0.12	0.05	7355	0.05	0.01
7×30	1	2028	2001	1998*	1.50	0.15	1998*	1.50	0.15
	2	10517	10194	10180	3.31	0.14	10178*	3.33	0.16
	3	10794	10702	10698*	0.90	0.04	10704	0.84	0.02
	4	10890	10764	10760*	1.21	0.04	10762	1.19	0.02

*refers to minimum makespan

Table 11 – Comparison between FA, AIS, GA and MA for variable size sublots (total flow time)[31]

m × n	*Problem Set*	*Total flow time*							
		GA	*MA*	*FA*	*% Deviation from GA*	*% Deviation from MA*	*AIS*	*% Deviation from GA*	*% Deviation from MA*
3×10	1	4644	4036	4032*	15.18	0.10	4032*	15.18	0.10
	2	16318	16315	16315*	0.02	0	16315*	0.03	0
	3	17750	17750	17750*	0	0	17750*	0	0
	4	16960	16932	16930*	0.18	0.01	16930*	0.18	0.01
5×20	1	13391	13045	13032*	2.75	0.10	13038	2.71	0.05
	2	74724	70585	70512*	5.97	0.10	70530	5.95	0.08
	3	74057	73935	73825*	0.31	0.15	73825*	0.31	0.15
	4	75384	74664	74600*	1.05	0.09	74650	0.98	0.02
7×30	1	28625	28625	28625*	0	0	28625*	0	0
	2	165518	165508*	165708	0.11	0.12	165708	0.11	3.63
	3	168938	165983	165950*	1.80	0.02	166285	1.78	0.18
	4	168662	168662	168662*	0	0	168662*	0	0

*refers to minimum total flow time

8. Conclusion

This paper has addressed the *n*-job, *m*-machine lot streaming problem in a flow shop with equal and variable size sub lots, where the objective is to minimize the makespan and total flow time. The two proposed heuristics FA and AIS, both suitable for providing solutions to any scheduling criterion. In order to verify the feasibility and the performance of the proposed algorithms, four different problem sets were tested. The success rate is defined by the ratio between the number of problems for which a particular method was the best solution and the total number of problems solved. Therefore, when two methods get the best solution for the same problem, their percentages of success are the same. Computational results summarized in tables clearly shows that the proposed FA and AIS outperform the other algorithms reported in the literature. The proposed algorithm sufficiently describes the processing dynamics of individual lots and enables the simultaneous determination of schedules on machines with equal and variable sublot sizes. FA and AIS optimize the makespan, as well as total flow time of jobs with the test problem instances generated using a random generation method. The comparison between them reveals that FA performs better than AIS in providing quality solutions with small increase in generations. This work can be extended by implementing other local search techniques and testing the features to solve combinatorial optimization problems, dynamic problems in real variables and other stochastic problems. The parameters of FA and AIS are tested with limited number of problem sets. It could be fine-tuned and related to problem size with more rigorous analysis so that computational effort could be minimized considerably. However, the closeness to optimality and consistency need to be established with further research.

References

1. Stanley Reiter, A system for managing job shop production, *The Journal of Business*, 39(3) (1966), 371-394.
2. Kenneth R. Baker, David F. Pyke, Solution procedures for the lot-streaming problem, *Decision sciences*, 21(3) (1990), 475-491.
3. C.N. Potts, K.R. Baker, Flow shop scheduling with lot streaming. *Operations Research Letters*, 8(6) (1989), 297–303.
4. Vickson, R. G., Alfredsson, B. E., Two and three machines flow shop scheduling problems with equal sized transfer batches. *International Journal of Production Research*, 30(7) (1992), 1551–1574.

5. C.A. Glass, J.N.D. Gupta, C.N. Potts, Lot streaming in three-stage process, *European Journal of Operational Research,* 75(2) (1994), 378–394.
6. Jiang Chen, George Steiner, Lot streaming with detached setups in three-machine flow shops. *European Journal of Operational Research*, 96(3) (1996), 591–611.
7. C. Sriskandarajah, E. Wagneur, Lot streaming and scheduling multiple products in two machine no-wait flow shops, *IIE Transactions,* 31(8) (1999), 695–707.
8. Chao-Tang Tsang and Ching- Jong Liao, A discrete Particle swarm optimization for Lot-streaming flow shop scheduling problem, *European Journal of Operational Research,* 191(2) (2008), 360-373.
9. Rahime Sancar Edis, M. Arslan Ornek, A Tabu Search – based Heuristic for single – product lot streaming problems in flow shops, *International Journal of Advanced Manufacturing Technology,* 43(11-12) (2009), 1202-1213.
10. Quan-Ke Pan, Ling Wang, Liang Gao and Junqing Li, An effective shuffled frog-leaping algorithm for lot streaming flow shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 52(5-8) (2011), 699 – 713.
11. Serdar Birogul, Cetin Elmas, Tuncay Yigit , Lot streaming based job- shop scheduling problem using hybrid genetic algorithm, *Scientific Research and Essay* , 6 (14) (2011), 2873-2887.
12. Rapeepan Pitakaso, Christian Almeder, Karl F. Doerner, Richard F. Hartl, A MAX- MIN ant system for unconstrained multi- level lot- sizing problems, *Computers and Operations Research,* 34(9) (2007), 2533-2552.
13. S. Marimuthu, S.G. Ponnambalam, N. Jawahar, Tabu search and simulated annealing algorithms for scheduling in flow shops with lot streaming, *Journal of Engineering Manufacture,* 221(2) (2007) 317-331.
14. S. Marimuthu, S.G. Ponnambalam, N. Jawahar, Evolutionary algorithms for scheduling *m*-machine flow shop with lot streaming, *Robotics and Computer Integrated Manufacturing,* 24(1) (2008) 125–139.
15. S. Marimuthu, S.G. Ponnambalam, N. Jawahar, Threshold accepting and ant-colony optimization algorithm for scheduling m-machine flow shop with lot streaming, *Journal of Materials Processing Technology,* 209 (2) (2009), 1026–1041.
16. Xin-She Yang, Firefly Algorithms for multimodal optimization, *Lecture Notes in Computer Science*, 5792 (2009), 169-178.
17. Mohammad Kazem Sayadi, Reza Ramezanian, Nader Ghaffari-Nasab, A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems, *International Journal of Industrial Engineering Computations*, 1(2010), 1-10.
18. S.C. Liu**,** A heuristic method for discrete lot streaming with variable sublots in a flow shop, *The International Journal of Advanced Manufacturing Technology*, Volume 22(9-10) (2003), 662-668.
19. Shu-Chu Liu, En-Chang Chen , H. T. Liu, A Heuristic method for multi- product variable lot streaming in a flow shop, *Journal of the Chinese Institute of Industrial Engineers*, 23(1) (2006), 65-79.
20. Fantahun M. Defersha, Mingyuan Chen, A hybrid genetic algorithm for flow shop lot streaming with setups and variable sublots, *International Journal of Production Research*, 48(6) (2010), 1705-1726.
21. D. Biskup, M. Feldmann, Lot streaming with variable sublots: an integer programming formulation, *Journal of the Operational Research Society*, 57(3) (2006), 296- 303.
22. Fantahun M. Defersha, Mingyuan Chen, A genetic algorithm for one-job m-machine flow shop lot streaming with variable sublots, *International Journal of Operational Research*, 10(4) (2011), 458 – 468.
23. Ranga V. Ramasesh, Haizhen Fu, Duncan K.H. Fong, Jack C. Hayya, Lot streaming in multistage production systems. *International Journal of Production Economics*, 66(3) (2000), 199-211.
24. Subhash C. Sarin, Adar A. Kalir, Ming Chen, A single-lot, unified cost-based flow shop lot-streaming problem, *International Journal of Production Economics*, 113(1) (2008), 413-424.
25. Suk-Hun Yoon, Jose A. Ventura, Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling, *Computers and Operations Research*, 29(10) (2002), 1301-1315.
26. Jiang Chen, George Steiner, Approximation methods for discrete lot streaming in flow shops, *Operations Research Letters*, 21(3) (1997), 139-145.

27. J. Senthilnath, S.N. Omkar, V. Mani, Clustering using firefly algorithm: Performance study, *Swarm and Evolutionary Computation*, 1(3) (2011), 164-171.
28. Xin-she Yang, Firefly algorithms for multimodal optimization. Stochastic Algorithms: *Foundations and Applications*, (2009), 169-178.
29. Leandro N. de Castro, Fernando J. Von Zuben, Learning and optimization using the Clonal selection principle. *IEEE Transactions on Evolutionary Computation* 6(3) *2002*, 239–351.
30. G.Vijay Chakaravarthy, S.Marimuthu, A.Naveen Sait, Performance Evaluation of proposed Differential Evolution and Particle Swarm Optimization algorithms for scheduling *m*-machine flow shops with lot streaming, *Journal of Intelligent Manufacturing,* (2011) DOI: 10.1007/s10845-011-0552-2.
31. S.Marimuthu, S.G. Ponnambalam, N. Jawahar, "Memetic Algorithm and Genetic Algorithm for lot streaming in m-machine, n-job flow shop scheduling with variable size sublot", proceedings of *International conference on Modeling and Simulation (MS2006),*April 3-5, (2006), University of Malaya, Kualalumbur, Malaysia.