

Hybrid Sorting Immune Simulated Annealing Algorithm For Flexible Job Shop Scheduling

N. Shivasankaran*

*Department of Mechanical Engineering
K. S. R. College of Engineering, Tiruchengode, Tamil Nadu – 637215, India*

P. Senthil Kumar

*Department of Mechanical Engineering
K. S. R. College of Engineering, Tiruchengode, Tamil Nadu – 637215, India
E-mail: cryosenthil@yahoo.co.in*

K. Venkatesh Raja

*Department of Automobile Engineering
K. S. R. College of Engineering, Tiruchengode, Tamil Nadu – 637215, India
E-mail: kvenkateshrraja@hotmail.com*

Received 20 June 2013

Accepted 7 January 2015

Abstract

Hybrid sorting immune simulated annealing technique (HSISAT), a Meta - heuristic is proposed for solving the multi objective flexible job-shop scheduling problem (FJSP). The major objectives are distributing the time of machines among the set of operations and scheduling them to minimize the criterion (makespan, total workload and maximum workload). The processing time is sorted for isolating the critical machines and immune simulated annealing (ISA) is applied to increase the convergence speed. Several case studies have been taken from the literature to demonstrate the convergence speed of the proposed algorithm. The computational results have proved that the proposed hybrid algorithm is an effective approach to solve the multi-objective FJSP.

Keywords: Scheduling; Sorting; Immune Simulated Annealing Technique; Flexible Job Shop; Multi-objective Optimization; Make span.

1. Introduction

The aim of scheduling problems is to allocate resources for execution of operations by satisfying single or a set of objectives and constraints ^[1]. These scheduling problems arise in varieties of production, project, service organizations and supply chain networks where constraints such as definite sequences and limited resources exist ^[2-5]. JSP is a branch of production scheduling which consists of y jobs to be processed through m machines in definite sequences ^[5]. Flexible job shop is an extension of job shop scheduling which permits an operation to be executed by one machine out of a set of machines. The machine set is defined for each operation and the corresponding

processing time depends on the selected machine. The most common goals of these scheduling problems are assigning the machines for a set of operations and defining the operations sequence that minimizes the make-span (completion time required for all jobs). The poor performance of exact methods on these scheduling problems such as Branch & bound method and dynamic programming leads to the development of different metaheuristic techniques. The converging processes of metaheuristic and local search optimization strategies such as genetic algorithm (GA), tabu search (TS), scatter search (SS), particle swarm optimization (PSO) and ant colony optimization (ACO) algorithms tend to be too slow for solving practical problems ^[6].

Simulated annealing (SA) algorithm can be used to solve these scheduling problems with less iterative process [7]. Few researches have been carried out by integrating the SA with several other procedures such as shifting bottleneck heuristic algorithm and controlled search simulated annealing algorithm [8], integrated simulated annealing and exchange heuristic algorithm [9], hybrid SA with random insertion perturbation scheme [10], hybrid adaptive memory programming and simulated annealing (AMPSA) [11] in job shop scheduling (JSP) for improving the makespan. Moreover, hybrid SA has been applied in multi objective flexible job shop scheduling problems (FJSP) by Yazdani, M et al [12], Fattahi, P [13], Dalfard, VM and Mohammadi, G [14] and Shamsavari-Pour, N and Ghasemishabankareh, B [15]. The hybrid SA is also used in scheduling problems with overlapping operations which has been applied in chemical, petrochemical and glass industries has been developed by Loukil T et al [16] and Khalife, MA et al [17]. In solving multi objective problems using SA significantly requires a hybrid algorithm for improving the computational time for reaching optimal solution [18].

In this paper, we propose an integrated approach based on hybridizing sorting simulated annealing [19, 20] algorithm and immune algorithm to solve flexible JSP. Sorting confines the initial search space by enhancing the convergence speed and immune simulated annealing algorithm provides the solution to converge quickly.

The remainder of this paper is organized as follows: In Section 2, the assumptions and formulation of the job shop scheduling problem have been dealt in

detail. Section 3 describes the standard SA algorithm and its application to JSP. The proposed HSISA algorithm is shown in Section 4. The experimental results and comparison with other algorithms in the literature is presented in Section 5 to demonstrate the superiority of the HSISA performance. Finally, the last section presents concluding remarks of our work.

2. Problem formulation

The flexible job shop scheduling is required to perform n jobs by m machines with multiple constraints. The scheduling task is to find the feasible schedule with minimum completion time. If n jobs and m machines are considered in general case, there will be $(n!)^m$ feasible assignments [21]. So if n jobs have p operations then, there will be $(np!)^m$ possible assignments. It can be proved from the Pareto optimal results obtained. The scheduling problem described above is having the following hypothesis:

- (i) m machines are to be scheduled.
- (ii) Each machine has to start the assigned task only after the completion of previous operation.
- (iii) The workload of the machines should be balanced.
- (iv) Each machine can perform at most one operation at any time (resource constraint).
- (v) Each job is associated with a predetermined machining order (technological constraint).
- (vi) No parallel operations are to be executed at a time in the same job.
- (vii) Job tardiness is not allowed.

Indices

i, h	job
j, g	operation
k	machine

Notations:

m	Number of machines
Y	Largest number
n	Number of jobs
r_i	Number of operations for job i
C_{ijk}	completion time of the operation j in job i by machine k
a_i	arrival time of job i
O_{ij}	operation j of job i
ST_{ijk}	starting time of operation j of job i by machine k
FT_{ijk}	finishing time of operation j of job i by machine k
P_{ijk}	processing time of operation O_{ij} on machine k ($P_{ijk} \geq 0$)

$$X_{ijk} = \begin{cases} 1, & \text{if operation } j \text{ of job } i \text{ is assigned to machine } k \\ 0, & \text{Otherwise} \end{cases}$$

$$\text{Min } f_1 = \text{Max}_{1 \leq i \leq n} \{ C_{ijk} \} \quad (1)$$

$$\text{Min } f_2 = \left\{ \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^{r_i} X_{ijk} P_{ijk} \right\} \quad (2)$$

$$\text{Min } f_3 = \text{Max}_{1 \leq k \leq m} \left\{ \sum_{i=1}^n \sum_{j=1}^m X_{ijk} P_{ijk} \right\} \quad (3)$$

Subject to

$$ST_{ijk} \geq a_i \quad (4)$$

$$ST_{i(j+1)k} \geq ST_{ij} + X_{ijk} P_{ijk} \quad (5)$$

$$FT_{ijk} = ST_{ijk} + X_{ijk} P_{ijk} \quad (6)$$

$$\sum_{k \in m_{ij}} X_{ijk} = 1 \quad (7)$$

$$ST_{ijk} + X_{ijk} P_{ijk} - Y(1 - Z_{ijhkg}) \leq ST_{hkg}, \forall k \in m_{ij} \cap m_{hg}, \forall O_{ij} \neq O_{hg} \quad (8)$$

$$ST_{hkg} + X_{ijk} P_{ijk} - Y(1 - Z_{hgijs}) \leq ST_{ijs}, \forall k \in m_{ij} \cap m_{hg}, \forall O_{hg} \neq O_{ij} \quad (9)$$

Three objectives used in this model are to minimize

- (1) f_1 , Maximum completion time of all jobs. This objective improves the reliability of setting delivery due date.
- (2) f_2 , Total Workload of all machines.
- (3) f_3 , Maximum Workload of all machines. This objective is used to balance the work load of all machines.

3. Problem formulation

SA has been developed from modified Monte Carlo integration method^[22] of computing equilibrium states for substances containing interacting molecules. Annealing refers to the process, when physical substances are raised to a higher energy level and gradually cooled until some solid state is reached. To achieve the lowest energy state, the cooling process must be sufficiently slow. However, there is a probability at each stage of cooling process which creates a transition from higher energy state to lower energy state. But this transition decreases when the temperature reduces. SA is a generic probabilistic metaheuristic used in the optimization problems for locating a good global approximation of a function in a large search space.

The probability of accepting a worse move is a function of both the change in the cost function and

temperature of the system^[23]. At temperature, t , the probability that the rate of change in the magnitude of energy, δE , is given by

$$P(\delta E) = \exp(-\delta E / kt) \quad (10)$$

Where k is a constant known as Boltzmann's constant.

If the energy decreases, then the system moves to the new energy state that can be calculated by using the simulation in the Metropolis algorithm. At the same time if the energy increases, this type of new energy state is accepted only by using above formula (10). Temperature can be decreased after certain number of iterations is carried out at each temperature. This process is repeated until the system freezes to a steady state. This equation is directly used in simulated annealing algorithm. Generally Boltzmann constant is introduced in an equation only to cope different materials that can be dropped out here. Therefore, the

probability of accepting an inferior state is given by the equation

$$P = \exp(-c/t) > r \quad (11)$$

Where c = change in evaluation or objective function value
 t = the current temperature
 r = random no. between 0 and 1

4. Hybrid Sorting Immune Simulated Annealing algorithm

SA algorithm usually starts with a random solution and annealing process improves on that solution. However, it may be better to start this process with a heuristic solution. For example, when trying to solve a JSP problem, solution should satisfy all the constraints from the first iteration onwards.

4.1. Sorting algorithm

The application of sorting algorithm^[24] in multi objective optimization problems reduces computational complexity by eliminating the worst solution spaces and achieving initial feasible assignment in minimum time. In this sorting procedure, all non-dominated machines are ranked 1 and are temporarily removed from the population. The next set of non-dominated machines in the population is then defined and ranked 2. The procedure is continued until all the solutions are ranked. So this algorithm is used in this paper to sort the machines and restricting the selection of machines which delays production. The sorting algorithm also helps during vaccination in the immune simulated annealing algorithm by providing a technique for prioritizing the machine selection.

4.2. Immune simulated annealing algorithm

During the last decade, artificial immune systems (AIS) have been successfully applied to several theoretical problems and practical applications^[25]. In real situations, immune-inspired algorithms provide a new approach to use specific features of the problems for improving the solution quality. The biological immune system is an adaptive, complex system that helps our body to defend against infection and disease. The bacteria or virus introduced into the body that causes disease is referred to as antigen. In response to a specific antigen, the immune system produces a protein substance called antibody which destroys or weakens the invading antigen. Vaccines are used to help the body to generate certain antibodies. Vaccine consists of only a portion of the pathogen's structure and it is injected

into the body to induce the production of correct antibodies against the specific disease (e.g. smallpox). Based on the above knowledge, optimization problem (antigen) is solved by analyzing their features (vaccination) and optimal solution (antibody) is obtained.

Based on self-immunity concept, an integrated immune genetic algorithm (IGA) concept^[26] has been used to entrench heuristics for solving travelling salesman problem. For minimizing the expected damage of own force assets in weapon–target assignment, Immunity based ant colony optimization (IACO) algorithm^[27] was applied. Immune simulated annealing (ISA) Algorithm^[9] proposed for job shop scheduling is focused more on computational effort to optimize bottleneck (critical) jobs. The encoding in ISA is based on machine priority list of m machines. The order of priority of machines for an operation reduces when the processing time increases. If two or more machines have the same processing time, then the random selection of machines were done.

4.3. Hybrid Sorting Immune Simulated Annealing Algorithm

Hybrid sorting immune simulated annealing algorithm shown in Fig. 2 represents the evaluation of global optimal solution in FJSP. Let us consider an example as shown in Fig. 1(a) having 4 jobs, 4 operations and 5 machines in which the HSISA algorithm is implemented. Main steps of heuristic procedure are detailed as follows:

- (i) Sort the k capable machines based on the increasing order of processing time P for each operation of a job as shown in Fig. 1(a).
- (ii) The generation of initial machine assignment is based on selecting the less processing time for an operation. If two or more machines have identical processing time, then the machine is selected in random.
- (iii) The vaccination procedure is performed on the selection of machine with least processing time in random and sequence generation to generate initial feasible solution. Firstly, the vaccine tries to balance the workload of the machines using machine workload constraint. Secondly, the vaccine tries to swap the initial randomly generated sequence based on machine waiting time and job waiting time to the more suitable position, so that the final sequence obtained by

swapping is optimal. If there is no possibility for swapping due to waiting time principle, then the final solution is displayed for that iteration.

First column 2,1,3 of the antibody generation shown in Fig. 1(b) indicates Job 2, operation 1 to be processed in machine 3 generated randomly after satisfying machine and total workload constraints and initial schedule generation is shown in Fig. 1(c). Here 4 operations are assigned to Machine 2, since the processing time, total workload and makespan increases when these operations are assigned to other machines. The immunization process is done by vaccinating the antibody using the job availability and idle condition of the machine as shown in Fig. 2 (a). Here the job 4 is available at time 0. So, the position of the 5th operation of the antibody is swapped with the 7th operation and then the optimal solution is generated. Here, the machine availability is tightened to improve the makespan and job completion time is also checked for

reducing the idleness of the job. This process is continued till the machine idle time and job waiting time reduces to minimum or zero. Then the iteration is stopped and the result is displayed.

(iv) If ISA generates new state O' from O and the difference of objective values is $\Delta O = O'(t) - O(s)$, Where $O(\cdot)$ denotes the objective value of a solution (solution represents Makespan, Maximum workload and Total workload), then go to step v, to update the current solution.

(v) If $\Delta O \leq 0$, accept the new solution O' . Else if $\Delta O > 0$, accept the new solution O' with the probability determined by the acceptance function $A = \min \{1, \exp(-\Delta O / T)\}$, Where T is the current temperature. A uniformly distributed random number $R \sim u[0, 1]$ is generated and tested with acceptance function for accepting the solution under this probability. If $R \leq A$, Then the solution O' is accepted.

Job, Operation	Processing Time					Ranking				
	M1	M2	M3	M4	M5	1	2	3	4	5
1,1	2	5	4	1	2	M ₄	M ₁	M ₅	M ₃	M ₂
1,2	9	8	2	7	9	M ₃	M ₄	M ₂	M ₁	M ₅
1,3	6	2	7	5	9	M ₂	M ₄	M ₁	M ₃	M ₅
2,1	2	5	1	7	8	M ₃	M ₁	M ₂	M ₄	M ₅
2,2	5	2	9	8	6	M ₂	M ₁	M ₅	M ₄	M ₃
2,3	7	4	6	7	4	M ₂	M ₅	M ₃	M ₁	M ₄
2,4	2	5	4	2	1	M ₅	M ₁	M ₄	M ₃	M ₂
3,1	4	2	5	6	7	M ₂	M ₁	M ₃	M ₄	M ₅
3,2	1	5	2	4	12	M ₁	M ₃	M ₄	M ₂	M ₅
3,3	5	3	8	4	7	M ₂	M ₄	M ₁	M ₅	M ₃
4,1	1	5	3	4	6	M ₁	M ₃	M ₄	M ₂	M ₅
4,2	10	5	12	11	10	M ₂	M ₁	M ₅	M ₄	M ₃

a) Application of sorting to an FJSP Instance

2,1,3	3,1,2	1,1,4	2,2,2	3,2,1	1,2,3	4,1,1	1,3,2	3,3,4	2,3,5	2,4,5	4,2,2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

b) Representation of antibody using random generation process

	1	2	3	4	5	6	7	8	9	10	11	12
M1			3		4							
M2	3		2		1			4				
M3	2	1										
M4	1			3								
M5					2		2					

c) Gantt Chart showing the initial Solution

Fig. 1 Problem and Initial Solution generation

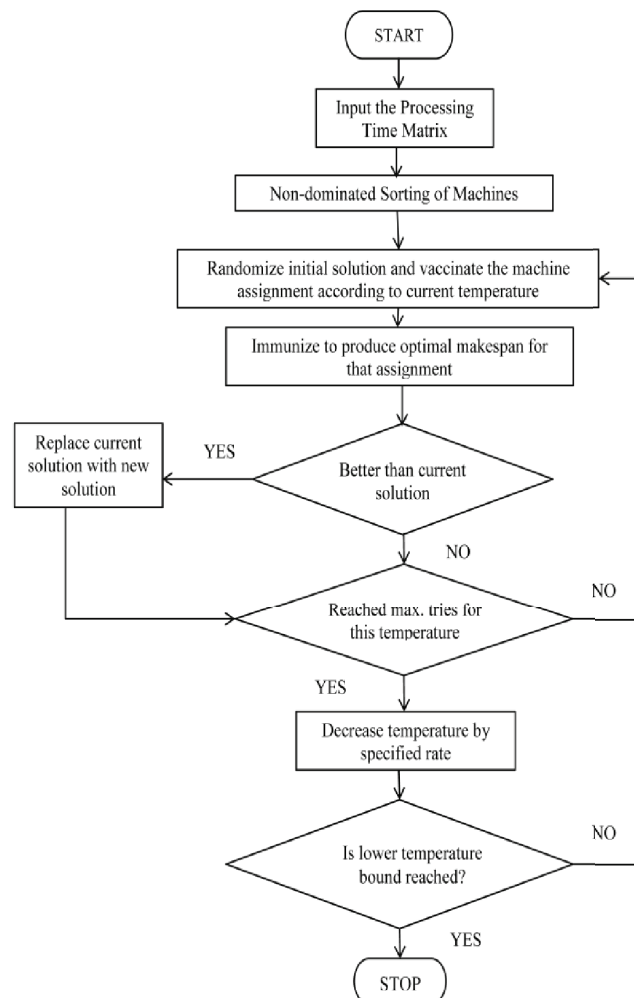
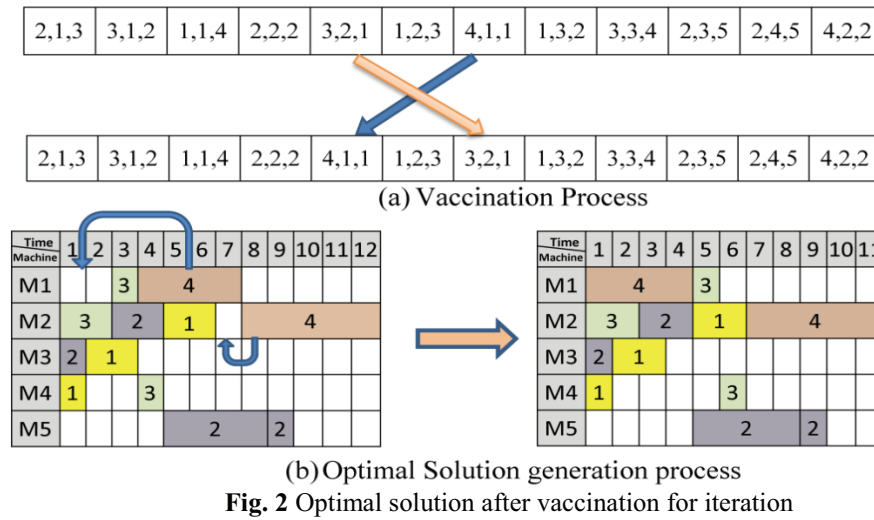


Fig. 3 Flow chart showing HSISA algorithm

5. Experimental Results & Discussion

The hybrid algorithm proposed is coded in Turbo C++ on Core2Duo desktop computer having a 2.27 GHz processor with an initial temperature parameter setting of 1000, reduced by a cooling factor of 0.99 with a termination temperature of 600. These parameters are set as presented by Parthasarathy S. et al. [28] for allowing the algorithm for 30 temperature ranges.

Six bench mark instances based on practical data have been selected. Each instance can be characterized by the following parameters: number of jobs (n), number of machines (m), and each operation O_{ij} of job i . Five instances (4 X 5, 10 X 7, 10 X 10, 15 X 10 and 8 X 8) with partial and total flexibilities are taken from Kacem et al. [29, 30]. Sixth instance with total flexibility and varying date of availability of the job for processing is taken from Mesghouni et al. [31, 32].

Fig. 4 - 9 shows the obtained optimal solution by our proposed approach in the form of a Gantt chart for the Kacem instances. These charts present the assignments which have not been shown by other algorithms. Some instances show multiple Pareto optimal solutions as given in Table 01. Fig. 9 shows the convergence curves for the total workload of the algorithm for Kacem instances. Here the algorithm produces initial random solution using workload assignment constraint and then vaccination procedure described in the Fig. 3 helps the initial solution to become feasible since the waiting time of both machine as well as job is checked to optimize the variables. So the curve does not flip either upwards or downwards. This proves the computational efficiency of the algorithm.

Fig. 10 shows the optimal solutions by the HSISA algorithm in the form of Gantt chart for the Mesghouni instance which is similar to Kacem instances having an additional constraint of date of availability of the job. Here the algorithm drastically improves all the objective functions and are shown in Table 02. The makespan is reduced by 1 unit and total completion time reduces by 1 unit which proves the effectiveness of the algorithm. The minimum number of iterations to obtain optimal solutions drastically reduces and gives optimal results from the initial iteration onwards with minimum time.

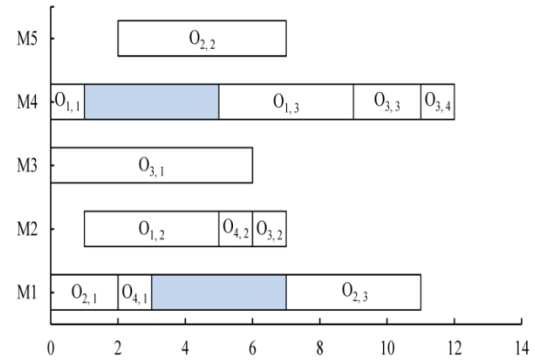


Fig. 4 Optimal Solution for 4x5 instance ($f_1 = 12, f_2 = 8, f_3 = 32$).

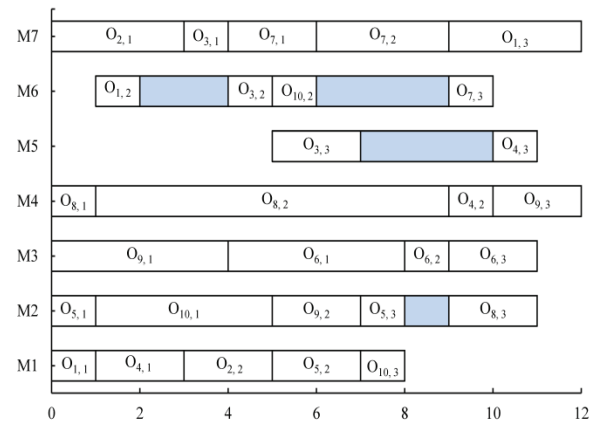


Fig. 5 Optimal Solution for 10x7 instance ($f_1 = 12, f_2 = 12, f_3 = 60$).

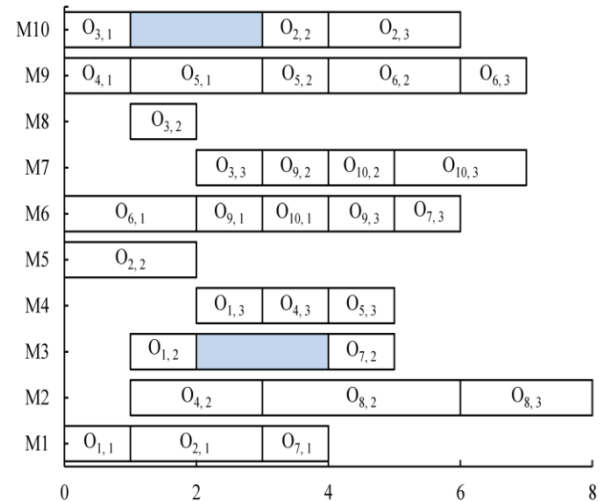


Fig. 6 Optimal Solution for 10X10 instance ($f_1 = 8, f_2 = 7, f_3 = 41$).

Table 01 Comparison of results on the five Kacem instances.

Problem Size ($n \times m$)	Attributes		Comparison of results on the five Kacem instances.										Proposed HSISA						
			Kacem et al [29, 30] AL + CGA		Shao et al [33]		Xing et al [34] Simulation		Rajkumar et al [35]		Li et al [36] HTSA		Mahnam et al [37]	Xia et al [38]					
	S1	S2	S1	S2	PSO+SA	S2	PSO+TS	S1	S2	S1	S2	GRASP	S1	S2	hGA	S1	S2	MOPSO+LS	PSO+SA
4X5	Make-span	f_1	18	18	11	11	-	12	-	16	-	16	-	11	12	16	-	12	11
	Maximum workload	f_2	8	7	10	-	-	8	-	9	-	9	-	10	8	8	-	8	8
	Total load	f_3	32	33	32	-	-	32	-	32	-	32	-	32	32	32	-	32	32
	Number of Optimal Solutions		3		1		1		1		1		1		2	2	-	2	2
	Av. comp. time (sec) for 20 runs		-		0.34		2.58		-		0.15		-		0.08		-	-	0.08
10X7	Make-span	f_1	16	15	-	-	-	11	11	-	-	11	11	16	-	16	-	16	16
	Maximum workload	f_2	12	11	-	-	-	11	10	-	-	10	11	12	-	12	-	12	12
	Total load	f_3	60	61	-	-	-	61	62	-	-	62	61	60	-	60	-	60	60
	Number of Optimal Solutions		5		-		2		2		2		3		8	8	-	8	8
	Av. comp. time (sec) for 20 runs		-		-		109.99		-		2.58		-		0.502		-	-	0.502
10X10	Make-span	f_1	8	7	7	6	8	8	8	9	8	7	8	8	7	8	7	8	8
	Maximum workload	f_2	7	5	6	6	5	6	6	7	7	5	5	7	6	7	6	7	7
	Total load	f_3	41	45	43	44	42	42	42	41	42	43	42	41	44	41	44	41	41
	Number of Optimal Solutions		3		1	1	3		3		1	1	3	2	1	1	1	1	1
	Av. comp. time (sec) for 20 runs		-		2.05		39.74		-		3.12		-		0.02		-	-	0.02
15X10	Make-span	f_1	23	24	12	11	11	11	11	12	16	11	11	23	12	13	12	13	13
	Maximum workload	f_2	11	11	11	11	11	11	11	12	13	11	10	11	11	12	11	12	12
	Total load	f_3	95	91	91	93	91	92	92	96	94	91	93	91	91	91	91	91	91
	Number of Optimal Solutions		2		2		4		4		2		2	1	1	1	1	20	20
	Av. comp. time (sec) for 20 runs		-		10.88		865.18		-		25.13		-		9.23		-	-	9.23
8X8	Make-span	f_1	15	16	14	15	14	15	15	16	15	14	15	15	15	16	15	16	16
	Maximum workload	f_2	-	-	12	12	12	12	12	13	17	12	12	12	12	13	12	13	13
	Total load	f_3	79	75	77	75	77	76	76	73	75	77	75	75	75	73	75	73	73
	Number of Optimal Solutions		-		2		2		2		2		2	1	2	2	2	3	3
	Av. comp. time (sec) for 20 runs		-		1.67		39.37		-		3.08		-		0.032		-	-	0.032

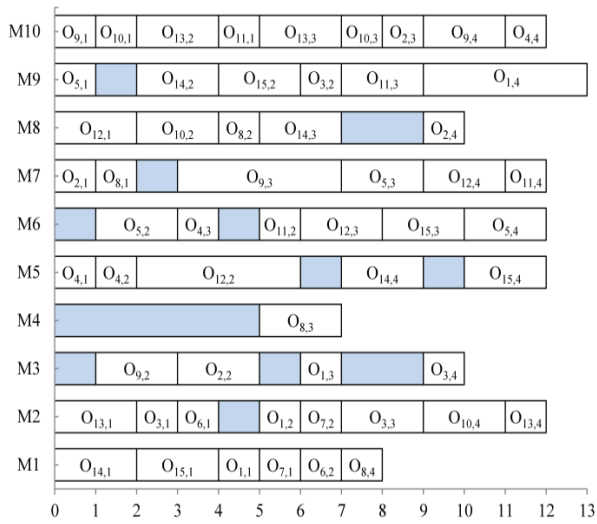


Fig. 7 Optimal Solution for 15X10 instance ($f_1 = 13, f_2 = 12, f_3 = 91$)

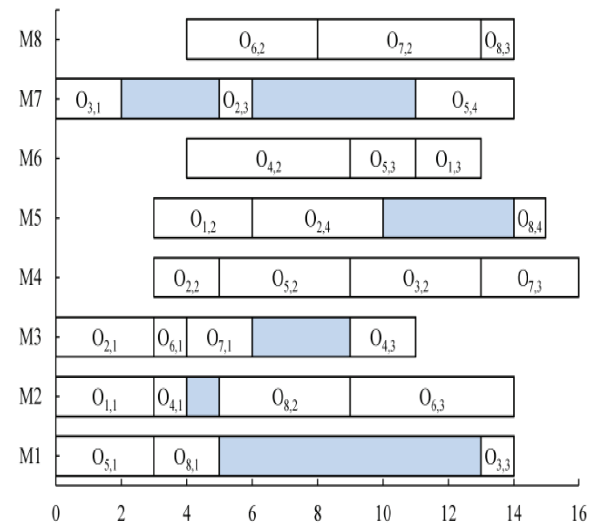


Fig. 8 Optimal Solution for 8X8 instance ($f_1 = 16, f_2 = 13, f_3 = 73$)

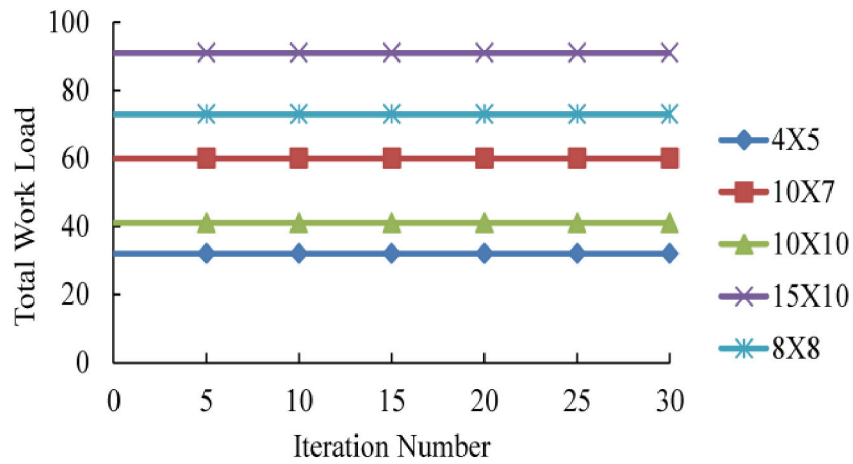


Fig. 9 Convergence curve for Kacem instances

Table 02 Results of Mesghouni instance.

Attributes	K. Mesghouni et al. [31]	K. Mesghouni et al. [32]	N. Shivasankaran et al. [19]	Proposed HSISA
Maximum Make span / job	16	7	7	7
Optimal Make span of all jobs	19	16	10	9
Optimal Completion time of all jobs	73	46	41	41
Number of Optimal Solutions	1	1	10	7
Minimum number of Generations for optimality	500	1853	1	1
Computational Time (sec)	-	-	0.0549	0.089

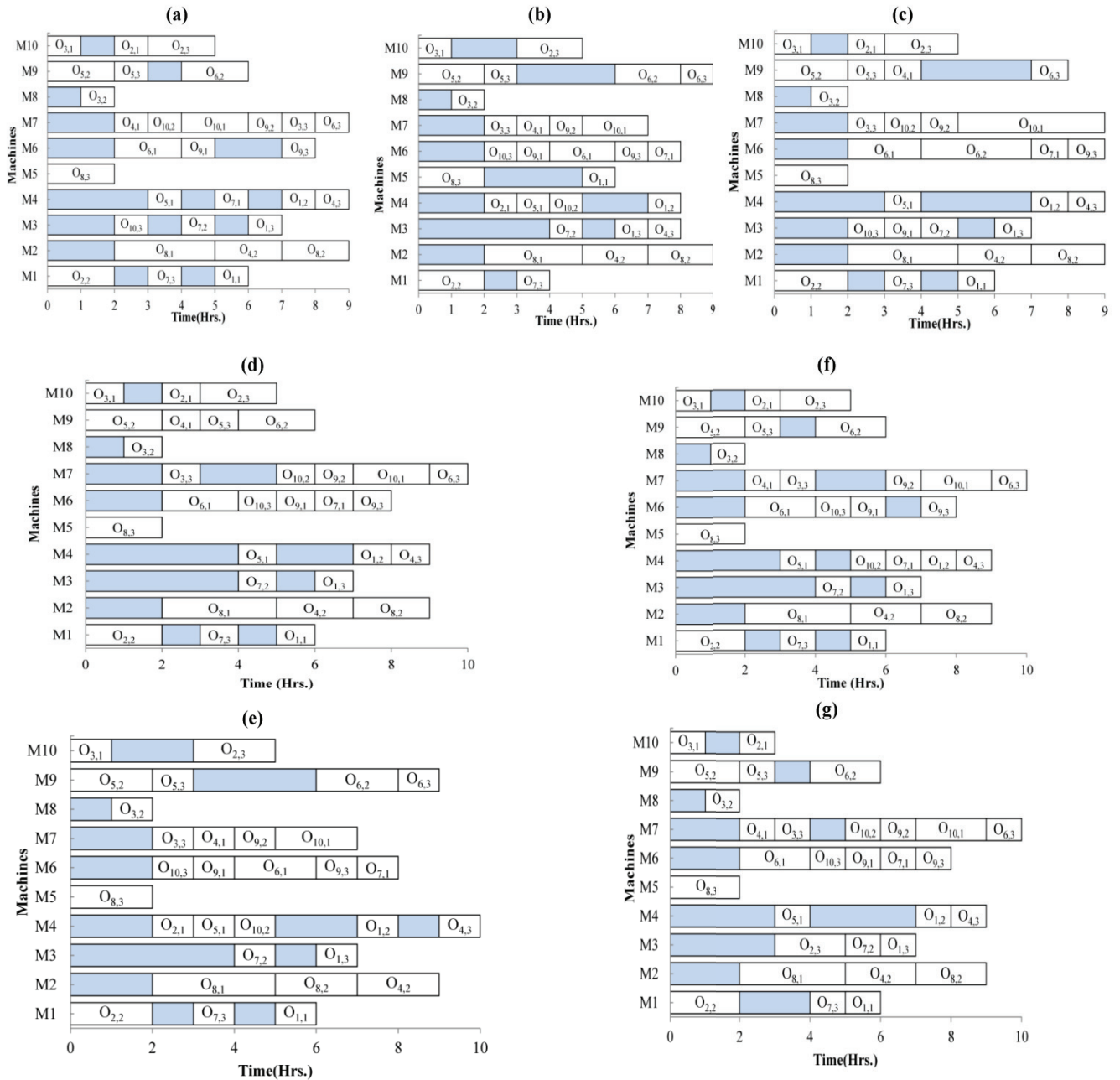


Fig. 10 Gantt charts for Mesghouni instance

6. Conclusion

In this paper, we propose hybrid sorting immune simulated annealing algorithm for multi objective Job shop scheduling problem. The sorting and immune algorithms confine the search spaces and in turn enhance the local search ability of SA by focusing on elimination of critical machines that consumes more

processing time for an operation. The computational time for obtaining the optimal solution is minimum than other algorithms and can solve relatively large size problems in minimum time. In a Scheduling system, an efficient heuristic search is required to explore the large solution space of valid resource allocation under various interacting constraints. The proposed algorithm can be treated as quite effective, and can find an optimal

solution in a faster rate (the number of iterations for convergence and running time of the algorithm are drastically reduced). The result proves that the algorithm could be treated as quite effective in solving the NP-Complete problems like resource allocation and scheduling. Our future work is to investigate the other meta-heuristics for the multi-objective job-shop scheduling problems and generalize the application of the sorting immune algorithm to solve other combinatorial optimization problems.

References

- [1] K. R. Baker, *Introduction to Sequence and Scheduling*, (Wiley, New York, 1974).
- [2] I. M. J. Orji, S. Wei (2013), Project Scheduling Under Resource Constraints: a Recent Survey, *Int. J. Eng. Res. Technol.*, 2(2), pp. 1 – 20.
- [3] A. Kaddoussi, N. Zoghlami, S. Hammadi and H. Zgaya (2013), An Agent-Based Distributed scheduling for crisis management Supply Chain, *Int. J. Comput. Int. Sys.*, 6(1), pp. 156 – 173.
- [4] Z. Zheng, L. Shumin, G. Ze and Z. Yueni (2013), Resource-constraint multi-project scheduling with priorities and uncertain activity durations, *Int. J. Comput. Int. Sys.*, 6(3), pp. 530 – 547.
- [5] Z. Zheng, Z. Guo, Y. Zhu and X. Zhang (2014), A critical chains based distributed multi-project scheduling approach, *Neurocomputing*, 143, pp. 282 – 293.
- [6] L. Wang, D. Z. Zheng (2001), An effective hybrid optimization strategy for job-shop scheduling problems, *Comput. Oper. Res.*, 28(6), pp. 585 – 596.
- [7] Z. He, T. Yang, A. Tiger (1996), An exchange heuristics imbedded with Simulated Annealing for due-dates job-shop scheduling, *Eur. J. Oper. Res.*, 91, pp. 99 – 117.
- [8] P. J. M. Van Laarhoven, E. H. L. Aarts, J. K. Lenstra (1992), Job shop scheduling by Simulated Annealing, *Oper. Res.*, 40, pp. 113 – 125.
- [9] R. Zhang, C. Wu (2010), A hybrid immune simulated annealing algorithm for the job shop scheduling problem, *Appl. Soft. Comput.*, 10, pp. 79 – 89.
- [10] S. G. Ponnambalam, N. Jawahar, P. Aravindan (1999), A simulated annealing algorithm for job shop scheduling, *Prod. Plan. Control*, 10(8), pp. 767 – 777.
- [11] A. El-Bouri, N. Azizi, and S. Zolfaghari (2007), A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: The case of job shop scheduling, *Eur. J. Oper. Res.*, 177, pp. 1894 – 1910.
- [12] M. Yazdani, M. Gholami, M. Zandieh and M. Mousakhani (2009), ‘A simulated annealing algorithm for flexible job-shop scheduling problem’, *J. Appl. Sci.*, 9(4), pp. 662 – 670.
- [13] Fattahi, P (2009), ‘A hybrid multi objective algorithm for flexible job shop scheduling’, *Int. J. Comput. Math. Sci.*, 3, pp. 215 – 220.
- [14] VM. Dalfard , and G. Mohammadi 2012, ‘Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints’, *Comput. Math. Appl.*, 64, pp. 2111 – 2117.
- [15] N. Shahsavari-Pour and B. Ghasemishabankareh, 2013, ‘A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling’, *J. Manuf. Syst.*, 32, pp. 771 – 780.
- [16] T. Loukil, J. Teghem and P. Fortemps 2007, A multi-objective production scheduling case study solved by simulated annealing, *Eur. J. Oper. Res.*, 179(3), pp. 709 – 722.
- [17] MA. Khalife, B. Abbasi and AHKD. Abadi 2010, ‘A Simulated Annealing Algorithm for Multi Objective Flexible Job Shop Scheduling with Overlapping in Operations’, *J. Ind. Eng.*, 5, pp. 17 – 28.
- [18] N. Shivasankaran , P. Senthil Kumar , G. Nallakumarasamy and K. Venkatesh Raja (2013) Repair Shop Job Scheduling with Parallel Operators and Multiple Constraints using Simulated Annealing, *Int. J. Comput. Int. Sys.*, 6(2), pp. 223 – 233.
- [19] N. Shivasankaran, P. Senthil Kumar, G. Nallakumarasamy and K. Venkatesh Raja 2012, A Hybrid Bubble Sorting Simulated Annealing algorithm for job shop scheduling, *Third International Conference on Computing Communication & Networking Technologies (ICCCNT)*, IEEE, pp. 1 – 5.
- [20] N. Shivasankaran, P. Senthilkumar and K. Venkatesh Raja (2014), Hybrid Non-dominated Sorting Simulated Annealing Algorithm for

- Flexible Job Shop Scheduling Problems, *Adv. Intel. Sys. Comp.*, 248 (1), pp. 101 – 107.
- [21] P. Mellor (1966), A Review of Job Shop Scheduling, *Oper. Res. Quart.*, 17(2), pp. 161 – 171.
- [22] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller (1953), Equation of State Calculations by Fast Computing Machines, *J. Chem. Phys.*, 21(6), pp. 1087 – 1092.
- [23] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi (1983), Optimization by Simulated Annealing, *Sci.*, 220 (4598), pp. 671 – 680.
- [24] H. Ghiasi, D. Pasini & L. Lessard (2011), A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems, *Eng. Optimiz.*, 43 (1), pp. 39 – 59.
- [25] E. Hart, J. Timmis (2008), Application areas of AIS: the past, the present and the future, *Appl. Soft. Comput.*, 8 (1), pp. 191 – 201.
- [26] L. Jiao, L. Wang (2000), A novel genetic algorithm based on immunity, *IEEE T. Syst. Man. Cyb. A*, 30 (5), pp. 552 – 561.
- [27] Z. J. Lee, S. F. Su, C. Y. Lee (2002), A Genetic Algorithm with Domain Knowledge for Weapon-Target Assignment Problems, *J. Chin. Inst. Eng.*, 25(3), pp. 287 – 295.
- [28] S. Parthasarathy and C. Rajendran (1997), A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs-a case study, *Prod. Plan. Control*, 8 (5), pp. 475 – 483.
- [29] I. Kacem, S. Hammadi, P. Borne (2002a), Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Math. Comput. Simulat.*, 60, pp. 245 – 276.
- [30] I. Kacem, S. Hammadi, P. Borne (2002b), Approach by localization and multi objective evolutionary optimization for flexible job-shop scheduling problems, *IEEE T. Syst. Man. Cyb.*, pp. 1 – 13.
- [31] K. Mesghouni, S. Hammadi, P. Borne (2004), Evolutionary Algorithms for Job-Shop Scheduling, *Int. J. Appl. Math. Comp. Sci.*, 14 (1), pp. 91 – 103.
- [32] K. Mesghouni, P. Pesin, D. Trentesaux, S. Hammadi, C. Tahon and P. Borne (1999), Hybrid approach to decision-making for job-shop scheduling, *Prod. Plan. Control*, 10 (7), pp. 690 – 706.
- [33] G. Zhang, X. Shao, P. Li, L. Gao (2009), An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Comput. Ind. Eng.*, 56, pp. 1309 – 1318.
- [34] L. N. Xing, Y. W. Chen, K. W. Yang (2009), Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling, *Appl. Soft Comp.*, 9, pp. 362 – 376.
- [35] M. Rajkumar, P. Asokan, V. Vamsikrishna (2010), A GRASP algorithm for flexible job-shop scheduling with maintenance constraints, *Int. J. Prod. Res.*, 48 (22), pp. 6821 – 6836.
- [36] J. Li, Q. Pan, Y. Liang (2010), An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems, *Comput. Ind. Eng.*, 59, pp. 647 – 662.
- [37] G. Moslehi, M. Mahnam (2011), A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *Int. J. Prod. Econ.*, 129, pp. 14 – 22.
- [38] W. Xia, Z. Wu (2005), An effective hybrid optimization approach for multi-objective Flexible Job-Shop Scheduling Problems, *Comput. Ind. Eng.*, 48, pp. 409 – 425.