# A Survey on Various Optimization Techniques with Respect to Flexible Job Shop Scheduling

**Vincent Lal [*], C. Anand Deva Durai [**]**

[*] Karunya University, Tamil Nadu, India
[**] Karunya University, Tamil Nadu, India

***Abstract-*** The purpose of this paper is to show the survey study done on the optimization algorithms for the Flexible Job Shop Scheduling Problem. There are many algorithms that have been used for finding the optimal solution of Flexible Job Shop Scheduling Problem. Here the Multi Objective aspects have been also taken into consideration. The FJS problem has already been proved an NP hard problem. Pareto optimality principal enriches the solution with the ease of finding the optimal solution.

***Index Terms*-** Computational Intelligence, Flexible Job Shop Scheduling Problem, Optimization Algorithm, Pareto Optimality.

## I. Introduction

The Job Shop Scheduling refers to concept that there are some jobs which needed to be done in a given time. In order to complete the jobs, jobs are employed to the machines. A job has several operations in it. So to complete a job it is important to complete that operations associated with the job. A set of $n$ jobs must be executed on $m$ machines, whereas every job $j$ composed of $n_i$ operations. When all the operations are completed then the associated job is regarded as complete.

When it comes to the term "Flexible" in Job Shop Scheduling, it tells the idea that all machines can do all operations. That means, every machine is capable of performing all of the operations. This makes the system completely flexible. Since Flexible Job Shop Scheduling problem is an advance version of Job Shop Scheduling problem, and comes from the same family of job scheduling therefore it is also an NP-Hard problem, which has been proved [1]. Since the nature of problem is NP-Hard that is why no procedure have been yet discovered to solve these problems within a given time. A variety of methods have been used to solve these problems such as local search, tabu search, simulated annealing, genetic algorithm etc. They provide solution for the problem of course not the exact solution, but near to the optimal solution.

There are several objectives associated with the problems which when taken with the Flexible Job Shop Scheduling problem then needed to be optimized. Then the problem becomes Multi Objective Flexible Job Shop Scheduling Problem.

These objectives are always in tradeoff with each other. In the case of Multi Objective Flexible Job Shop Scheduling the objectives could be total time for the jobs to finish; makespan, total workload on all the machines; total workload and the maximum workload of the entire working machines. These objectives are together taken into consideration while solving such problems. In the Industry Production environment, the Multi Objective concept has attained a very important place and studied in areas like, Batch Machines [2], Flow Shops [3], Job Shops [4], Parallel Machines [5]

In the past few years, Multi Objective Flexible Job Shop Scheduling has collected a huge attention by researchers. For solving these problems, some contributor to this research has used the approaches in combination with other approach for getting better results. The solution of Multi Objective optimization is also enriched by Pareto Approach. The Pareto approach defines the idea that for about 80% of the problems, 20% of the causes are responsible. In other words, if that 20% causes are removed then most of the problems i.e., about 80% problems will be solved. This approach helped many researchers to find out the Pareto Optimal solution.

A Hybrid tabu search algorithm with some existing dispatching rules to solve the single-objective FJSP was proposed by [6]. [7] used local search algorithm technique and developed two neighborhood functions for the problem.

In this paper, a study on the different techniques and approaches that have been used for solving the Flexible Job Shop Scheduling problem so far in this area of research have been presented. Also some benchmark algorithms for the same problem have been compared.

## II. Problem Formation

The FJSP as described in [8] is defined as:

1. There are $m$ machines in the set $M=\{m_1,m_2,m_3\ldots..m_m\}$.
2. There are $n$ jobs in the set $J=\{j_1,j_2,j_3\ldots..j_n\}$.
3. Each job $j$ ($1\leq j\leq n$) is composed of several operations $i$.
4. Each Operation of each job represented by $O_{ji}$ ($i^{th}$ operation of $j^{th}$ job), is processed on a machine from the set of available machines M.
5. The processing time of an operation performed by machine $m$ is $P_{jim}$, a constant value and is already known. Where $j$ is $j^{th}$ job from set J, $i$ is the $i^{th}$ operation of $j^{th}$ job, $m$ is the machine from set M on which the operation is performed.
6. The completion time is $CT_{ji}$ for $O_{ji}$.

There are some important hypotheses shown below.

1. All the jobs and machines are ready at time zero.
2. All jobs have predefined operations.
3. A machine can perform only one operation at a time.
4. The processing of operation is non-preemptive i.e., if a machine is performing an operation it cannot be stopped.

5. Jobs are not dependent on each other and machines are not dependent on each other.
6. The time required for setting up of machines is not to be considered.
7. The time of moving machines is also negligible.

The most common objective in all the industries or production environments is the time required to finish all the jobs. This objective is known as makespan. For quick response to the market requirements a smaller makespan is desired, which make the production faster. Second objective is this literature is the total workload; this means the total time on all the machines working in the production. Third objective in this literature is maximum workload. The maximum workload is the machine with the highest working time.

These objectives are formulated by [8] are as follows.

1. Makespan denoted by $C_M$ : $f_1 = \max_{j=1\ldots n} C_{jnj}$.
2. Total workload $W_T$: $f_2 = \sum_{k=1\ldots m} \sum_{O_{ji} \in O_k} P_{jik}$
3. Maximum workload $W_M$: $f_3 = \max_{j=1\ldots n} \sum_{O_{ji} \in O_k} P_{jik}$

Descriptions of above terminologies are.
1. Makespan: The makespan is the time required to complete all jobs.
2. Total workload: This is the total processing time of all the machines.
3. Maximum workload: This is the maximum time among all the machines.

## III. Algorithms Used for Solving Multi Objective Flexible Job Shop Scheduling Problem

*Genetic Algorithm*

The Genetic Algorithm is a very efficient algorithm in the context of solving Multi Objective Flexible Job Shop Scheduling Problem. Genetic Algorithms (GAs) are adaptive heuristic algorithm based on the ideas of natural selection and genetic. The basic concept of GA is designed to processes in natural system necessary for evolution. Here a short explanation is given on how the Genetic Algorithm has been used in research work carried out by [8].

Two parameters have been used, the population size ($N_{POP}$) and maximum generation number ($T_{GEN}$).
The detailed steps are:

1. Initialization: Generate the initial population of size $N_{POP}$. Decode the initial solutions, calculate the objectives values, and evaluate individuals. Set the generation number *t=1*.

2. Reproduction: Generate $N_{POP}$ offspring by mating selection and crossover. Among the original individuals and the new offspring, $N_{POP}$ individuals survive to generation *(t+1)* by environmental selection.

3. Balanced exploration and exploitation: Refine duplicate individuals by effective mutation operators.

4. Termination: If $t>T_{GEN}$, stop; otherwise, $t=t+1$, go to Step 2.

*Chromosome Encoding*

The Chromosome Encoding is done to format the population for the routing and sequencing process. This is a three tuple scheme [8] [9] [10]. Each chromosome is a sequence of genes. A Chromosome is a 3-tuple (j, i, k), in which j, i, and k represents the job, operation, and machine, respectively. A tuple (j, i, k) means that the operation $O_{ji}$ will be processed by machine k. The position of each tuple tells its priority. The tuple on the left end has higher priority.

*Chromosome Decoding*

For the decoding process first [8] assigns the machines to operations based on the encoding done. Then the GT algorithm [11] have been used.

*Initial Population*

The routing methods for the initial population used are some listed in [12] [9] and [13]. Apart from these following are the two methods given are:
1. Minimum Processing time: This method assigns an operation to the machine with the minimum processing time.
2. Random Assignment: This method assigns each operation to a machine randomly.

The sequencing methods are:
1. Most work remaining: This method puts the operations into the chromosome according to the remaining processing time.
2. Most number of operations remaining: It puts the operations into the chromosome according to the number of remaining operations in the same job.
3. Random dispatching: It places the operations into the chromosome in a random way.

*Individual Evaluation and selection*

This step requires two sub processes one is choosing individual for producing the offspring and another is choosing individual to survive to the next generation. For evaluating individual NSGA-II [14] have been used. First two individuals are taken randomly and one with higher rank or larger crowding distance is taken as parent. This way parents are selected and two parents produces two offspring through crossover.

*Crossover*

For crossover process two operators are selected. One is Assignment Crossover and another is Preserving order-Based Crossover [15].

*Balanced exploration and exploitation*

The individuals with similar objective vector will not make a good diversity. So these individuals will be removed. But the information encoded on these individuals may be useful. So a

ISSN 2250-3153

balance between the exploration and exploitation should be made. There are five operators given by [8] for this purpose which are as follows:

1. Machine re-assignment for the reduction of maximum workload.
2. Machine re-assignment for the reduction of makespan.
3. Random machine re-assignment.
4. Critical operation swap.
5. Critical operation re-insertion.

*Artificial Immune Algorithm for FJSP*

The natural immune system is a complex system of recognizing the pathogens. It has been used for solving Flexible Job Shop Scheduling Problem. It has been used in [10] for solving Flexible Job Shop Scheduling Problem which is explained here. It exists in the human body and its aim is to defend the body from foreign pathogens. Pathogens are foreign bacteria or viruses that enter the human body and destroy it. This process is called infection.

The immune system is capable of recognizing the pathogens that invade into the human body and proliferate. The pathogens have some proteins (antigens) in them which are recognized by immune cells. The immune system cells then kills the pathogens. These cells are immune cells or antibodies, and are distributed all around the body.

There are two processes that explain that how immune system works. It tells that how the immune system recognizes pathogens and kill them. They are Clonal Selection and Affinity Maturation. The clonal selection process happens in the event of a pathogens invasion in the body. The immune cells that recognize the pathogens get proliferated and some of them become effector cell and some becomes memory cells. The effector cells produces antibodies in large number and memory cells have a larger life. Memory cells stores the pathogens for future use. During the process of cell proliferation the cells with higher affinity to pathogens becomes memory cells. This process is known as affinity maturity.

The proposed Artificial Immune Algorithm by [10] is given below.

*Initialization:* it consists of two steps;

a. Parameter Setting:

This sets the limit of initial population, number of generations, the rate of AssignmentRule1, the rate of AssignmentRule2, the rate of Random rule, the rate of MWR rule, the rate of MOR rule, the number of mutation in each generation, the number of exchangeable antibodies and the implementation probability of each mutation operator.

b. Initial Population generation:

By using the AssingmentRule1 and AssignmentRule2 initial assignments are done. By using the random selection, MOR rule and MWR rule the sequencing is done.

*Objective Function Evaluation:*

This part is responsible for evaluating the fitness function for each antibody.

*Affinity evaluation:*

This calculates the affinity value for each antibody as

$$\text{Affinity} = \frac{1}{\text{makespan}}$$

*Clonal Selection and expansion:*

a. The antibodies with largest affinity are selected.
b. Generate same number of clones or copies from the selected antibodies.

*Producing next population:*

a. Mutation Operator: selecting some antibodies randomly from the clones and apply mutation to produce new antibodies.
b. Add the new antibodies to the current generation.
c. Replace antibodies with the lowest affinity with new ones.
d. Copy the best ones to the next generation.
e. Select antibodies from the current generation by a selection procedure and copy them to the next generation.

*Termination:* If the stopping criterion is met then best antibodies are returned.

*Search Method proposed by [16]*

The steps for their algorithms are as follows:

*Operation Assignment*

This step performs the assignment of operations to machines. First an operation is selected and then search the machine capable of performing that operation in minimum time. Among the machines found randomly a machine is chosen for the operation to be performed. These steps are repeated.

*Operation Sequencing*

In the operation sequencing, operations are sequenced on each machine. The steps for the complete process have been explained in [16] for better understanding it can be referred.

*Feasible Move Search*

The move is for operation from one machine to another machine. And in [16] it is defined as

Move = { $O_{ij}$, $M_s$, $M_t$}

Where $O_{ij}$ is the operation, $M_s$ is the source machine of operation $O_{ij}$, $M_t$ is the destination or target machine. In every move the operation is moved from the source machine to destination machine.

Further there are two move search algorithm defined. First is random handpicked searching algorithm and another is full scaled general searching algorithm. Random handpicked searched algorithm is (1) Select an operation randomly. (2) Move that operation to the machine that is having minimal processing time. The algorithm is known as Full-Scaled general searching algorithm and it is described as (1) Select operations one by one. (2) Move the operation to all the machines except the current machine. The implementation of these algorithms is first the random handpicked searching algorithm is applied first followed by the full scaled general searching algorithm.

*Feasible Move Evaluation*

The performance of one move is evaluated using the total objective value of the resulted schedule.

*Best Move Evaluation*

The best move is the one which have performed better than all the other moves therefore it is regarded as best move and it is obtained after the move evaluation

*Stopping Criteria*

There are two stopping criteria given, (1) Maximum Preset search iterative is completed. (2) The known optimal solution is achieved.

*Genetic Algorithm based on Immune and Entropy principle*

In this research work [17], the genetic algorithm has been used with immune and entropy principle for maintaining the diversity of the individuals.
There are following things done in the work carried out in [17].

*Fitness Assignment Scheme*

The fitness is calculated on the basis of dominance. It is also like the fitness scheme in SPEA2 [18]. But the difference is that the in [17] the diversity strategy have been applied by immune and entropy algorithm instead of the strength niche.

For the non-dominated individuals, the fitness calculation is done as,

$$\text{Fitness} = n_i / (N+1)$$

Where, $N$ is the population, $n_i$ is the individuals that are dominated by individual $i$.
The dominated solutions get their fitness by,

$$\text{Fitness} = \sum_{i \in NDSet, i>j}^{n} \text{fitness}$$

Where, $NDSet$ is the set of non-dominated solutions. $i>j$ represents that, individual $i$ dominates $j$.

*Immune and Entropy principle.*

Under this, the basic definition of immune system has been given. The explanation of antigens, pathogens, immune cells etc have been given a basic explanation. The strategy have been explained is given in a very easy way and can be referred to [17].

*Initialization*

The initialization strategy that have used is easy and simple and described as a combination of two steps. The first step is the generation of operation sequence is randomly done and two machines are selected out of capable machines. The second is, if a random generated number Random is less than 0.8, then choose the one with the shorter processing time on it; else choose one with longer processing time.

*Encoding and decoding*

The representation is done in two parts, first is used to know the processing sequence of the operations. Second is, to assign a suitable machine to each operation. The representation is done by the two parts above explained.

The decoding process is the process of scheduling solutions. For decoding, the steps are performed as. First, choose the machine for each operation based on the machine assignment vector and then find the sequence of operations.

*Selection Operator*

In the selection procedure there are two steps. First, keeping the best individual. Second, tournament selection. The best individuals are kept in a way that 1% best individuals are copied in the parent solution to the children. The tournament selection strategy is actually proposed by [19].

*Crossover Operator*

There are two crossover operators are used. First is the Improved Precedence operation crossover and another is multi point preservative crossover.

For the operation sequencing the Improved Precedence operation crossover and for machine assignment multi point preservative crossover is used.

*Mutation Operator*

For the requirement of improving the ability of local search and maintaining the population the mutation operator are used. For the operation sequencing and machine assignment mutation operator is applied.

*Artificial Bee colony Algorithm*

*Concept of the ABC Algorithm*

The ABC algorithm was initially used for solving continuous problems like multi-variable and multi-modal continuous problems. There are few control parameters in the ABC algorithm, which is the main advantage of the algorithm. Due to its simplicity and ease of implementation, the ABC algorithm has gained more and more attention and has been used to solve many practical engineering problems.

There are two components in the algorithm: the foraging artificial bees and the food source. The position of a food source represents a possible solution and the nectar amount of a food source corresponds to the fitness of the solution.

The artificial bee is divided into three groups; employed bees, onlooker bee, and scouts bees. The employed bees are the one who are performing exploitation on a food source. A bee waiting in the hive for making decision to choose a food source is called an onlooker. The scout bees are the one who performs exploration procedure and random exploitation search to find a new food source.

ABC tries to use natural behavior of real honey bees in food foraging. Honey bees use some techniques such as waggle dance for optimally locating the food sources and to search new ones. Also makes them a good candidate for developing new intelligent search algorithms.

In the research work presented by [20] have presented the artificial bee colony optimization algorithm as a hybrid algorithm named as Pareto based discrete artificial bee colony.

*The hybrid Pareto based ABC*

a. Food Source Representation

The food representation is done by two vectors. The first vector places the assigned machines for each operation on their positions. Another one informs about the scheduling sequence.

b. Local Search approach

There are two operators have been used are; Local Search Operator in Routing Component and Local Search Operator in Operation Scheduling component.

c. Employed Bee Phase

The work of the employed bee is to do local search around the given food source. So exploitation search with local search have been taken. In this way the new food sources are discovered and compared with the old food sources. Finally the better food sources are kept.

d. Crossover Operator

The crossover is done in a random process. The crossover produces new population. This process is an important process in the system. First an employed bee is allowed to select one employed bee randomly and performs crossover. This produces two results. The resulted solutions are compared with the older ones and the better ones are kept.

e. Onlooker bee phase

The onlooker bee selects a food source based on the nectar amount of food source. But this process consumes high computational time, so another approach has been proposed here, tournament selection. In the tournament selection through a random picking process, three food sources are picked up and the one with the highest nectar amount is taken by the onlooker bee. Then each onlooker performs the same local search operator with the employed bee and produces new food sources.

f. Scout bee

The scout bee randomly searches in the Artificial Bee Colony. This increases the diversity in population and helps in avoiding the local minima however it will decreases the search efficacy.

## IV. Performance Comparison

Benchmark Instances

In the study of Job Shop Scheduling there are two most popular benchmark instances available. They are given in [6] and [21]. These two instances are taken as the input in most of the algorithms and also are taken in [8]. The data in [21] set contains 5 instances in the form of n x m (where n is number of jobs and m is number machines) and it ranges from 4 x 5 to 15 x 10. Another data set in [6] contains ten datasets and ranges from 10 x 6 to 20 x 15.

In the literature work done in Flexible Job Shop Scheduling so far there is always performance is presented as per the list of non-dominated solutions found in a certain number of run.

The comparison done in [8] is shown here. The parameter are just two only here which makes this algorithm apart from all algorithms. They are the population size and the generation number.

The Table 1 shows the comparison between the surveyed algorithms. This table has been taken from [8]. The bold values are the non-dominated solutions. This solution is for the 8 x 8 problem from [21].

The survey revealed that the techniques and efforts given in [8] have much more ability to get desired solution. One main thing to be noted here is that the other algorithm has used around 5 to 14 parameters.

Whereas [8] have just used two parameters which makes it less computational complex. While [10] focused on minimizing the makespan only. [16] have aggregated the three objectives by linear weighted sum.

Another comparison of the conceptual level have been introduced in this paper with advantage and disadvantage for the algorithms surveyed. This information can be seen in Table 2.

| Surveyed paper | $C_M$ | $W_T$ | $W_M$ |
|---|---|---|---|
| [8] | 14 | 77 | 12 |
| | 15 | 75 | 12 |
| | 16 | 73 | 13 |
| | 16 | 77 | 11 |
| [10] | 14 | 77 | 12 |
| [16] | 14 | 77 | 12 |
| | 15 | 75 | 12 |
| | 16 | 77 | 11 |
| | 17 | 73 | 13 |
| [17] | 15 | 75 | 12 |
| | 15 | 81 | 11 |
| | 16 | 73 | 13 |
| [20] | 14 | 77 | 12 |
| | 15 | 75 | 12 |
| | 16 | 73 | 13 |

By the two given tables it can be inferred that what the algorithms are capable of doing. The flexible job shop scheduling is an NP-Hard Problem whose exact solution cannot be determined.

But so far such algorithms and techniques have been developed that only tells the near optimal solutions. There are many authors that have addressed this problem. These kind of problems are extremely tough and are hard to solve. Use of pareto optimality principle it becomes easier to find the solution that are optimal for a problem. Pareto approach has been widely used in the literature of Flexible Job Shop Scheduling Problem for solving the problem.

**Table 2: Comparison of Surveyed papers**

| Surveyed paper | Merit | Demerit |
|---|---|---|
| [8] | The algorithm has performed better in | For Brandimarte benchmark |

| | | |
|---|---|---|
| | all cases of objectives. And minimized the number of parameters to just two as compared to other algorithms. | instances it has not performed well. |
| [10] | The algorithm has performed very effectively in minimizing the makespan. | It has confined in minimizing the makespan only which resist it as a Multi Objective Problem. |
| [16] | User can set preferences for the objectives with tags like 'very important', 'important', unimportant'. | It has not performed better than other algorithms in one solution rest it has performed equivalent to all. |
| [17] | For the brandimarte instances the algorithm has performed better than any other algorithm. | High number of parameter usage may lead to complex calculations |
| [20] | A new crossover operator has been introduced. | It uses a search technique that stores non-dominated solutions for next generation which makes causes less diverse population |

## V. Conclusion

This paper is a survey between five algorithms which are summarized in the table 1. Here procedure of these algorithms has been explained in short. For detailed explanation it is recommended to look for the corresponding papers. Multi Objective Flexible Job Shop Scheduling is a challenging problem whose exact solution could not be determined but a near optimal solution is obtained by several research works.

There is also a conceptual comparison done in the table number 2. It consists of the performance and the merits and demerits for the respective algorithms.

## References

[1] Mati, Y., Xie, X, "The complexity of two-job shop problem with multi-purpose unrelated machines." European Journal of Operational Research 2004, 153, 159-169

[2] Reichelt, D., M¨ onch, L, "Multiobjective scheduling of jobs with incompatible families on parallel batch machines." Lecture Notes in Computer Science 2006, 3906, 209–221..

[3] Chang, P.C., Hsieh, J.C., Lin, S.G., "The development of gradual priority weighting approach for the multi-objective flow shop scheduling problem." International Journal of Production Economics 2002, 79 (3), 171–183.

[4] Chiang, T.C., Fu, L.C., "Multiobjective Job Shop Scheduling Using Genetic Algorithm with Cyclic Fitness Assignment." In: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 3266–3273.

[5] Chang, P.C., Chen, S.H., Lin, K.L., "Two-phase sub-population genetic algorithm for parallel machine scheduling problem." Expert Systems with Applications 2005, 29 (3), 705–712..

[6] Brandimarte, P. "Routing and scheduling in a flexible job shop by tabu search". Annals of Operations Research,1993, 41, 157–183..

[7] Mastrolilli, M., & Gambardella, L. M., "Effective neighborhood functions for the flexible job shop problem." Journal of Scheduling,2000, 3(1), 3–20.

[8] Tsung-Che Chiang n, Hsiao-Jou Lin. "A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling." Int. J. Production Economics, 141, 2013, 87–98

[9] Pezzella, F., Morganti, G., Ciaschetti, G., "A genetic algorithm for the flexible job-shop scheduling problem." Computers & Operations Research 2008 35, 3202–3212..

[10] Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M. "An artificial immune algorithm for the flexible job-shop scheduling problem." Future Generation Computer Systems 2010, 26, 533–541.

[11] Giffler, B., Thomspon, G.L., "Algorithms for solving production-scheduling problems." Operations Research 1960, 8, 487–503.

[12] Kacem, I., Hammadi, S., "Borne, P., Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problem." IEEE Transactions on Systems, Man, and Cybernetics 2002a, Part C 32, 1–13.

[13] Zhang, G., Gao, L., Shi, Y., "An effective genetic algorithm for the flexible jobshop scheduling problem." Expert Systems with Applications 2011 38, 3563–3573.

[14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation 2002, 6, 182–197.

[15] Lee, K.M., Yamakawa, T., Lee, K.M., "Genetic Algorithm for General Machine Scheduling Problems.' In: Proceedings of International Conference on Knowledge- based Intelligent Electronic Systems, 1998, pp. 60–66.

[16] Xing, L.N., Chen, Y.W., Yang, K.W., "An efficient search method for multiobjective flexible job shop scheduling problems." Journal of Intelligent Manufacturing 2009b 20, 283–293.

[17] Wang, X., Gao, L., Zhang, C., Shao, X., "A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem." International Journal of Advanced Manufacturing Technology 2010, 51, 757–767.

[18] Zitzler E, "Evolutionary algorithms for multiobjective optimization: Methods and applications." 1999 Dissertation, Swiss Federal Institute of Technology.

[19] Goldberg DE, Deb K (1991) "A comparative analysis of selection schemes used in genetic algorithms." In: Rawlins G (ed) Foundations of genetic algorithms. Morgan Kaufmann, San Mateo, pp 69–93.

[20] Li, J.Q., Pan, Q.K., Gao, K.Z., "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems." International Journal of Advanced Manufacturing Technology 2011a, 55, 1159–1169.

[21] Kacem, I., Hammadi, S., Borne, P., Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and Computers in Simulation 2002b, 60, 245–276.

## Authors

**First Author** – Vincent Lal, Karunya University, Tamil Nadu, India, Vincent Lal received his Bachelor of Technology from Uttar Pradesh University in 2012.

Right now he is pursuing Masters of Technology from Karunya University, Coimbatore. His research interest areas are Computational Intelligence, Evolutionary Computing.

**Second Author** – C. Anand Deva Durai, Karunya University, Tamil Nadu, India, C. Anand Deva Durai received his Master of Science in software Engineering from Madurai Kamraj University in 2004 and Master of Engineering in Computer Science from Anna University, Chennai, in 2007. He completed his doctor of philosophy in computer science and engineering from Manonmaniam Sundaranar University, Tirunelveli and presently working as an Assistant Professor in the school of Computer Science and Technology, Karunya, University, Coimbatore. His primary research interests lie in Pattern recognition, Biometrics, Computer vision, Artificial Intelligence and Intelligent Systems. He is a life time member of the Computer Society of India.