ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

An efficient Genetic Algorithm Approach for Minimising the Makespan of Job Shop Scheduling Problems

[1]Pardeep Kumar, [2]Pawan Kumar, [3]Ram Bhool , [4]Virender Upneja

[1,2,3,4] Assistant Professor, Panipat Institute of Engineering and Technology, Samalkha, Panipat, Haryana, India

Abstract— The Job-Shop Scheduling Problem (JSSP) is a schedule planning for low volume systems with many variations in requirements. In Job-Shop Scheduling Problem (JSSP), there are 'k' operations and 'n' jobs to be processed on 'm' machines with a certain objective function to be minimized i.e. makespan.

In this dissertation, a promising genetic algorithm for the job-shop scheduling problems is proposed with new operating parameters i.e., a random population generation with a population size of 50, operation based chromosome structure, tournament selection as selection scheme, 2-point random crossover with probability 80% (Pc = 0.8), 2-point mutation with probability 20% (Pm = 0.2), elitism, repairing of chromosomes and no. of iteration is 1000. An algorithm is programmed for JSSP using MATLAB 2009 a 7.8 and seven different bench mark problems are tested with their operating parameters for makespan which also validates the computer codes developed.

The Genetic Algorithm (GA) with new operating parameters is then applied to the same problems. The results show that it gives better value of make span for the given problems. The results also show that the Genetic Algorithm is the best optimization technique for solving the scheduling problems due to its implications to more practical and integrated problems.

Index Terms— **Genetic Algorithm (GA) MATLAB 2009 a 7.8 and Job-Shop Scheduling Problem (JSSP)**

I. INTRODUCTION

Manufacturing industries are the backbone in the economic structure of a nation, as they contribute to both increasing GDP/GNP and providing employment. Productivity, which directly affects the growth of GDP, and benefits from a manufacturing system, can be maximized if the available resources are utilized in an optimized manner. Optimized utilization of resources can only be possible if there is proper scheduling system in place. This makes scheduling a highly important aspect of a manufacturing system. This chapter presents a review of scheduling in general and Job-Shop Scheduling in particular. Finally, a brief review of the scheduling and Job-Shop Scheduling is given below:

Scheduling can be defined as, "the allocation of resources over a period of time to perform a collection of tasks" [Noor, (2007)]. According to [Hitomi, (1996)] "it is a function to determine an actual (optimal or feasible) implementation plan as to the time schedule for all jobs to be executed; that is, when, with what machine, and who does what operation". Scheduling has its applications everywhere, for example; flights scheduling, train scheduling and production scheduling. According to [Wiers, (1997)] manufacturing scheduling is the performance of operations on a set of jobs, with the help of already allocated set of machines, within a specified time.

Shortest Processing Time (SPT)

According to shortest processing time the priority for job sequencing on a machine is the length of processing time. Shortest the processing time of a job, the earliest the job is to be processed. The objectives associated with this technique are reduction in the average work in process, average job completion and average job lateness [Smith, (1989)], [Fogarty et al. (1991)].

Genetic Algorithms

Genetic algorithms (GAs) are the research technique based on principles of natural selection and genetics [Fraser, (1957), Holland, (1975)]. Nowadays, search algorithms based on branch and bound methods and several approximation algorithms have been developed. However, results from the branch and bound method sometimes are really unpredictable and requires a lot of time. It depends on the size of the problem. Thus, schedulers are usually satisfied with an acceptance result which is not far from optima. One of the widely used techniques in industries is the local search. One of the search techniques that have been used is genetic algorithm (GA). Genetic algorithms solve a problem using the principal of evolution. In the search process, it will generate a new solution using genetic operator such as selection, crossover and mutation.

Genetic Algorithms (GAs) are widely known Evolutionary Algorithms that are used in practice. GAs and other EAs are population based search techniques that explore the solution space in a discrete manner. When GAs/EAs are hybridized with local search, they are known as Mimetic Algorithms (MAs). In solving JSSPs, the priority rules can be used as local search in conjunction with genetic algorithms, which is basically a mimetic algorithm. These approaches have been proved to be efficient for solving job-shop scheduling problems. In scheduling, genetic algorithms represent schedules as individuals or a population's members. Each individual has its own fitness value which is measured by the objective function. Each generation consists of individuals who survive from the previous generations. Usually, the population size remains constant from one generation to the next generation.

2 Types of Scheduling

2.1 Project Scheduling:

It is the scheduling of activities involved in carrying out a project. A project can be construction of a factory, a bridge, a high way or maintenance and repair of a factory or a plant etc. A number of software based approaches are available to handle such type of scheduling. Some well known techniques involve; Graphical Evaluation and Review Technique (GERT), Critical Path Method (CPM), Project Evaluation and Review Technique (PERT).

2.2 Operations Scheduling:

Operation scheduling can be defined as, "the processing of a set of jobs, in a given amount of time, on the already allocated corresponding set of machines, in a workshop consisting of several machines or production facilities including operative workers" [Hitomi, (1996)]. [Jain, (1998)] classified the available operations scheduling models as job sequencing, flow-shop scheduling, mixed-shop scheduling, Job-Shop scheduling and open-shop scheduling.

2.3 Job-Shop Scheduling:

Job-Shop Scheduling Problem (JSSP) is one of the well known hardest combinatorial optimization problems [Gary and Johnson, (1979)]. There is still a lot of room for improvement in the existing techniques because of its large solution space and JSSP is considered to be comparatively one of the hardest problems to solve. If there are 'n' jobs and 'm' machines, the number of theoretically possible solutions is equal to (n!) m [Noor, (2007)]. Among these solutions, an optimal solution, for a certain measure of performance, can be found after checking all the possible alternatives. But the checking of all the possible alternatives can only be possible in small size problems.

JSSP is one of the most well-known problems in both fields of production management and combinatorial optimization. JSSP problem is a method of resource allocation for simple-objective or multi-objective optimization scheduling with constraints. Scheduling derives its importance from two different considerations.

Ineffective scheduling results in deprived utilization of available resources. A noticeable symptom is the idleness of facilities, human resources and apparatus waiting for orders to be processed. Thus, cost of production increases.

Poor scheduling normally create delays in the flow of some orders through the systems. Thus, it calls for special measures that further increase cost of production can be thought of as production process or production scheduling problems.

3 Literature Review

This dissertation focuses on JSSP optimization using shortest processing time (SPT), transportation time (TT) with Genetic Algorithm approach. In this area, there is an immense body of literature. Hence, this chapter reviews the prior work that is most related to this dissertation.

Several researchers have addressed the problem of Job Shop Scheduling. The important contributions using GA are discussed below:

Dirk et al. (2004) considered JSSP with release and due-dates as well as various tardiness objectives. GA can be applied almost directly, but come along with apparent weaknesses. Two ways of reducing a search space are investigated by considering short-term decisions made at the machine level and by long-term decisions made at the shop floor level [1].

Linyan Sun et al. (2006) worked on open JSSP in which operations can be performed in any order. They developed a heuristic for the open JSSP using genetic algorithm to minimize make span. Genetic algorithm operators were suitably modified to maintain feasibility. The results were statistically compared and found to be significantly better than the earlier reported results [2].

Mati et al. (2008) considered the JSSP with release dates and due dates, with the objective of minimizing total weighted tardiness. In this research the efficiency of GA does no longer depend on the schedule builder when an iterated local search was used. Computational experiments carried out on instances of the literature show the efficiency of the proposed algorithm. GA was combined with an iterated local search that used a longest path approach on a disjunctive graph model. A design of experiments approach was employed to calibrate the parameters and operators of the algorithm. Previous study on GA for the JSSP point out that these algorithms were highly depended on the way the chromosomes are decoded [3].

Wang et al. (2008) proposed a two-stage GA which attempts to firstly find the fittest control parameters, namely, number of population, probability of crossover, and probability of mutation, for a given job shop problem with a fraction of time using the optimal computing budget allocation method, and then the fittest parameters were used in the GA for a further searching operation to find the optimal solution. In the past few decades, GAs had demonstrated considerable success in providing efficient solutions to many Non-polynomial-hard (NP-hard) optimization problems. But there was no literature available considering the optimal parameters when designing GAs. Unsuitable parameters may generate an inadequate solution for a specific scheduling problem [4].

Rao et al. (2008) applied an effective hybrid GA for the JSSP. Three novel features for this algorithm to solve the JSSP. Firstly, a new full active schedule (FAS) procedure based on the operation-based representation was presented to construct a schedule. After a schedule was obtained, a local search heuristic was applied to improve the solution. Secondly, a new crossover operator, called the Precedence Operation Crossover (POX), was proposed for the operation-based representation, which can preserve the meaningful characteristics of the previous generation. Thirdly, in order to reduce the disruptive effects of genetic operators, the approach of an improved generation alteration model is introduced. The proposed approaches were tested on some standard instances and compared with other approaches. The superior results validate the effectiveness of the proposed algorithm [5].

Yih-Long et al. (2009) observed that the problem was even more complex because businesses often judge solution goodness according to multiple competing criteria. They used heuristics such as shortest processing time (SPT) and earliest due date (EDD) can be used to calculate a feasible schedule quickly, but usually do not produce schedules that were close to optimal in these job shop environments. Real world job shops have to contend with jobs due on different days, material ready times that vary, reentrant workflows and sequence-dependent setup times. Genetic algorithms (GA) can be used to produce solutions in times comparable to common heuristics but closer to optimal. Therefore, a GA can be easily applied and modified for a variety of production optimization criteria in a job shop environment that includes sequence-dependent setup times [6].

Jinwei et al. (2009) proposed Novel Parallel Quantum Genetic Algorithm (NPQGA) for the stochastic JSSP Problem with the objective of minimizing the expected value of make span, where the processing times are subjected to independent normal distributions. Based on the parallel evolutionary idea and some concepts of quantum theory, they simulate a model of parallel quantum computation. There were some demes (sub-populations) and some universes (groups of populations), which were structured in super star-shaped topologies. A new migration scheme based on penetration theory was developed to control migration rate and direction adaptively between demes and a novel quantum crossover strategy is devised among universes [7].

Andrea Rossi et al.(2009) presented an advanced software system for solving the flexible manufacturing systems (FMS) scheduling in a job-shop environment with routing flexibility, where the assignment of operations to identical parallel machines has to be managed, in addition to the traditional sequencing problem. Two of the most promising heuristics from nature for a wide class of combinatorial optimization problems, genetic algorithms (GA) and ant colony optimization (ACO), share data structures and co-evolve in parallel in order to improve the performance of the constituent algorithms. A modular approach is also adopted in order to obtain an easy scalable parallel evolutionary-ant colony framework. The performance of the proposed framework on properly designed benchmark problems is compared with effective GA and ACO approaches taken as algorithm components [8].

Zandieh et al. (2009) investigated JSSP with sequence-dependent setup times and preventive maintenance policies. The optimization criterion is to minimize make span. The performances of the proposed algorithms were evaluated by comparing their solutions through two benchmarks based on Tail lard's instances. Although many researchers have proposed different techniques to integrate production scheduling and preventive maintenance, these techniques have some drawbacks. For example, some of them are so intricate that one cannot easily implement them, or some strongly exploit specific features of the original studied problem that one cannot apply them to other problems. Propose two techniques that were easy to understand and code, yet simplistically adaptable to any other machine-scheduling problems [9].

Deming Lei et al. (2010) suggested an efficient Random Key Genetic Algorithm (RKGA) to maximize the minimum agreement index and to minimize the maximum fuzzy completion time. In RKGA, a random key representation and a new decoding strategy were proposed and two-point crossover (TPX) and discrete crossover (DX) are considered. RKGA is applied to some fuzzy scheduling instances and performance analyses on random key representation, and the comparison between RKGA and other algorithms were done. Computation results validate the effectiveness of random key representation and the promising advantage of RKGA on fuzzy scheduling [10].

Defers et al. (2010) presented a mathematical model for a job-shop scheduling problem incorporating sequence-dependent setup time, attached or detached setup time, machine release dates, and time lag requirements. In order to efficiently solve the developed model a parallel GA that runs on a parallel computing platform. Numerical examples show that parallel computing can greatly improve the computational performance of the algorithm. JSSP is an extension of the classical job-shop scheduling problem by allowing an operation to be assigned to one of a set of eligible machines during scheduling. Thus, the problem is to simultaneously assign each operation to a machine (routing problem), prioritize the operations on the machines (sequencing problem), and determine their starting times. The minimization of the maximal completion time of all operations was a widely used objective function in solving this problem [11].

Guohui Zhang et al. (2011) proposed an effective genetic algorithm for solving the flexible job-shop scheduling problem (FJSP) to minimize make span time. In the proposed algorithm, Global Selection (GS) and Local Selection (LS) were designed to generate high-quality initial population in the initialization stage. An improved chromosome representation was used to conveniently represent a solution of the FJSP, and different strategies for crossover and mutation operator are adopted. Various benchmark data taken from literature were tested. Computational results prove the proposed GA effective and efficient for solving flexible job-shop scheduling problem [12].

Rubiyah,(2011) employed a new hybrid parallel GA (PGA) based on a combination of asynchronous colony GA (ACGA) and autonomous immigration GA (AIGA) to solve benchmark JSSP and effort of searching an optimal solution for scheduling problems. An autonomous function of sharing the best solution across the system was enabled through the implementation of a migration operator and a "global mailbox". The solution was able to minimize the make span of the scheduling problem, as well as reduce the computation time. To further improve the computation time, micro GA which works on small population was used in this approach. The result shows that the algorithm was able to decrease the make span considerably as compared to the conventional GA [13].

Zejko et al. (2011) described a methodology of automatic genetic algorithm parameters adjustment dedicated to a JSSP with a no-wait constraint with a make span criterion. The numerical results show that in a given problem, the efficiency of an algorithm with auto-tuning was placed at the level of an algorithm steered in a classical way with the best-fit steering parameters [14].

Oddi et al. (2011) presented a heuristic algorithm for solving a JSSP with sequence dependent setup times and min/max separation constraints among the activities (SDST-JSSP/max). The algorithm relied on a core constraint based search procedure, which generates consistent orderings of activities that require the same resource by incrementally imposing precedence constraints on a temporally feasible solution. Key to the effectiveness of the search procedure was a conflict sampling method biased toward selection of most critical conflicts and coupled with a non-deterministic choice heuristic to guide the base conflict resolution process. This constraint-based search was then embedded within a larger iterative-sampling search framework to broaden search space coverage and promote solution optimization. The efficiency of the overall heuristic algorithm is demonstrated empirically both on a set of previously studied job-shop scheduling benchmark problems with sequence dependent setup times and by introducing a new benchmark with setups and generalized precedence constraints[15].

Tammany et al. (2012) proposed a modified version of the genetic algorithm for flexible job-shop scheduling problems (FJSP). The GA class of stochastic search algorithms is very effective at finding optimal solutions to a wide variety of problems. The proposed modified GA consists of:

1) An effective selection method called "fuzzy roulette wheel selection".

2) A new crossover operator that uses a hierarchical clustering concept to cluster the population in each generation.

3) A new mutation operator that helps in maintaining population diversity and overcoming premature convergence.

The objective of this research is to find a schedule that minimizes the make span of the FJSP. The experimental results on 10 well-known benchmark instances show that the proposed algorithm is quite efficient in solving flexible job-shop scheduling problems [16].

Manier et al. (2012) proposed a model for Flexible JSSP (FJSSP) with transportation constraints and bounded processing times. This was a NP hard problem with a objectives to minimize the make span and the storage of solutions. A genetic algorithm with tabu search procedure was proposed to solve both assignment of resources and sequencing problems on each resource. In order to evaluate the proposed algorithm's efficiency, five types of instances were tested. Three of them consider sequencing problems with or without assignment of processing or/and transport resources. The fourth and fifth ones introduce bounded processing times which mainly characterize Surface Treatment Facilities (STFs). Computational results show that our model and method were efficient for solving both assignment and scheduling problems in various kinds of system [17].

Ren et al. (2012) proposed a new genetic algorithm (GA), in order to increase the diversity of the population, a mixed selection operator based on the fitness value and the concentration value was given. To make full use of the characteristics of the problem itself, new crossover operator based on the machine and mutation operator based on the critical path were specifically designed. To find the critical path, a new algorithm to find the critical path from schedule was presented. Furthermore, a local search operator was designed, which can improve the local search ability of GA greatly. Based on all these, a hybrid genetic algorithm was proposed and its convergence was proved. The computer simulations were made on a set of benchmark problems and the results demonstrated the effectiveness of the proposed algorithm [18].

Zhou et al. (2012) proposed JSSP with due dates and deadlines in the presence of tardiness and earliness penalties. Due dates were desired completion dates of jobs given by the customer, while deadlines were determined by the manufacturer based on customer due dates. Due dates can be violated at the cost of tardiness, whereas deadlines must be met and cannot be violated. The aforementioned scheduling problem, which is NP-hard, can be formulated with the objective function of minimizing the sum of weighted earliness and weighted tardiness of jobs subject to due dates and deadlines. In order to solve the problem, an enhanced genetic algorithm (EGA) was introduced. EGA utilizes an operation-based scheme to represent schedules as chromosomes. After the initial population of chromosomes was randomly generated, each chromosome was processed through a three-stage decoder, which first reduces tardiness based on due dates, second ensures deadlines were not violated, and finally reduces earliness based on due dates [19].

Sun et al. (2013) proposed a hybrid algorithm combining particle swarm optimization (PSO) and tabu search (TS) was to solve the JSSP with fuzzy processing time. The objective was to minimize the maximum fuzzy completion time, i.e., the fuzzy make span. In the proposed algorithm, PSO performs the global search, i.e., the exploration phase, while TS conducts the local search, i.e., the exploitation process. The global best particle was used to direct other particles to optimal search space. Therefore, in the proposed algorithm, TS-based local search approach was applied to the global best particle to conduct find-grained exploitation. In order to share information among particles, one-point crossover operator is embedded in the hybrid algorithm. The proposed algorithm was tested on sets of the well-known benchmark instances [20].

Wang et al. (2013) proposed a GA with very special chromosome encoding to handle flexible JSSP that can adapt to disruption to reflect more closely the real-world manufacturing environment. They hope that by using just-in-time machine assignment and adapting scheduling rules, they can achieve the robustness and flexibility we desire. Either partial flexible job shop or total flexible job shop were studied and discussed in large amount. However, it was still far from a real-world manufacturing environment, in which disruptions such as machine failure must be taken into account. After detailed algorithm design and description, experiments were carried out. A right-shifting reschedule repairs schedules by delaying affected operations until the disruption is over. A prescheduled works on each disruption scenario separately, treating disruptions like prescheduled downtime. Experiments showed that approach was able to adapt to disruptions in a manner that minimized lost time than compared benchmark algorithms [21].

Rau et al. (2013) used Lamarckian evolution for improve the quality of the schedules for large instances, and modify the OG&T algorithm to further reduce the idle time of the machines and operators, in this case at the risk of leaving all optimal schedules out of the search space. Conducted a large experimental study showing that these improvements allow the genetic algorithm to reach high quality solutions in very short time, and so it was quite competitive with the current state-of-the-art methods. The JSSP with operators is a very interesting problem that generalizes the classic job shop problem in such a way that an operation must be algorithm to solve this problem considering make span minimization. The GA uses permutations with repetition to encode chromosomes and a schedule generation scheme, termed OG&T, as decoding algorithm. This combination guaranties that at least one of the chromosomes represented and optimal schedule and, at the sachet machines and operators were idle while an operation is available to be processed [22].

María et al. (2013) proposed a new schedule generation scheme for this problem; termed OG&T; prove that it can generate optimal schedules for any instance. This scheme can be used in different settings such as heuristic search, to define a branching strategy, or evolutionary algorithms, to define a decoder. In order to evaluate OG&T, they consider the first option and exploit it to devise at any time exact algorithm. This algorithm is enhanced with two heuristic estimations, designed from two problem relaxations, and with a pruning method which relies on dominance properties among states of the search. The algorithm was evaluated across several benchmarks and compared with other approaches. The results of the experimental study show that our approach outperforms the state-of-the-art methods [23].

3.1 Solution Techniques to Handle JSSP

A number of solution techniques to handle the JSSP have been developed over the years. Initially, the techniques are divided into two classes as approximation and optimization techniques. A complete classification is shown in Figure 3.1.

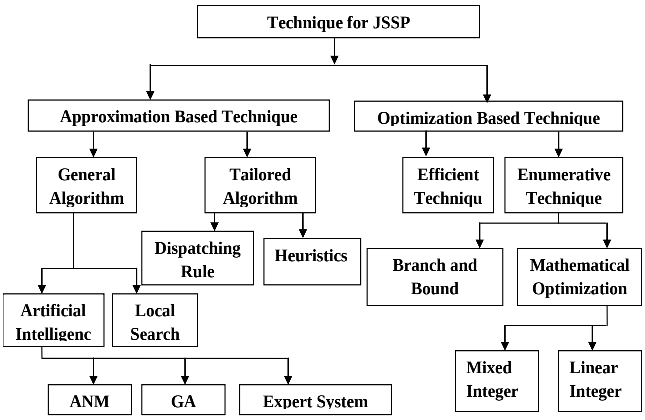

Fig. 3.1 Solution Approaches to JSSP [Jain (1998)]

3.3.1 Optimization Based Technique

Optimization techniques typically encountered in engineering optimization applications. The techniques were classified as either local or global algorithms and considered both constrained and unconstrained optimization problems. The designer should be aware that no single optimization algorithm exists that will solve all optimization problems. Some knowledge of the different algorithms available will help to select the appropriate algorithm for the problem at hand. A few guidelines for algorithm selection are:

1. Local algorithms are well suited for optimization problems with many design variables (more than 50), where the analyses are computationally expensive when numerical noise is not a severe problem, gradients are readily available, and local minima is not an issue. Local optimization techniques are further classified into following types:
    1. Gradient- Based Optimization.
    2. Newton's Method.
    3. Unconstrained Optimization.
    4. Constrained Optimization.
    5. Non-gradient based Methods.
2. Global algorithms are well suited for optimization problems with fewer design variables (less than 50), where, the analysis is computationally inexpensive for discrete and combinatorial optimization problems, when numerical noise is severe, in cases, where the gradient does not exists, when the objective and/or constraint functions are discontinuous, and when a global optimum is required. Global optimization algorithms may be classified:

Evolutionary Algorithms
Deterministic Algorithms

3. 3.2 Approximation Based Technique

Approximation based approaches offer a good alternatives for solving the JSSP in terms of the quality of solution and computational time. Though, these techniques do not guarantee optimality, but still solutions obtained are feasible and near to optimum always. Another main advantage of these techniques is the ease with which they can be implemented in practice.
Approximation approaches are further classified as follows.

1. Tailored Algorithm.
2. General Algorithms / Artificial Intelligence.
3. Artificial Neural Networks (ANN).
4. Genetic Algorithms (GA).

4 Problem Formulations

4.1 Research Gap

Literature review reveals that many researches have been done into the field of evolutionary computations or Meta heuristic techniques and Genetic Algorithm (GA). No attempts so far have been made with modified Genetic Algorithm, such as tournament selection as selection scheme, 2-point random crossover with probabilities of (0.8), 2-point random mutation with probabilities of (0.2) in addition with transportation time and Short Processing Time as dispatching rule for Job Shop Scheduling Problem (JSSP).

4.2Mathematical Formulation of Problem

In the present work, there are a set of 'n' no. of jobs (J) may be given as: J = {J1, J2, Jn} and each job composed of more than one operation (O) that must be processed on a set of 'm' no. of machines (M), given as M = {M1, M2, . . ., Mm}. Each operation occupies one of the machines for a fixed duration. In order to simplify the problem, the following notations are used.
j= job (j=1, 2,….,n)
i= machine (i=1, 2,…..,m)
tt= transportation time
P= processing time
In this type of environment, where, the products are made to order, the job-shop scheduling problem (JSSP) can be described as follow:
1) Job sets
J = {J1, J2,….., Jj} | j = 1,2,3,……,n
2) Machine sets
M = {M1, M2,……, Mi}| i = 1,2,3,……,m
3) Operations
O = {O1, O2,……,Oo} | o = 1,2,3,…….,k
4) Processing time for each operation

Pij = {P11,P12,………,Pij},| i = 1,2,3,……,n ; j =1,2,3,……,m

5) Transportation time

ttij = {tt11,tt12,……….,ttij} | i = 1,2,3,……,n ; j =1,2,3,……,m

4.3 Problems Discussion

Problem No. 1: Moth & Thompson (1963), No of machine (m) = 6, No of job (n) = 6, Makes span = 49

Moth et al. (1963) described a GA approach which produced reasonable good results very quickly on standard bench mark JSSP and compare to existing conventional search method. The representation used in variant work moderately well known for travelling salesman problem.

The make span with operating parameters i.e. number of iterations 300, random selection, crossover probability (0.6), Mutation probability (0.2), came out to be 55 as shown in Fig. 4.1 below:

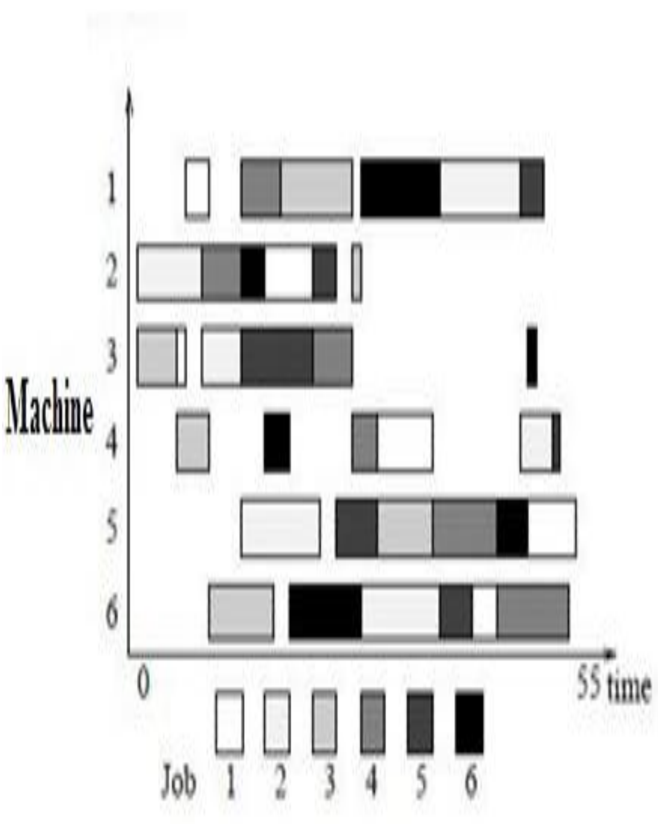

Fig.4.1 Optimal schedule of JSSP (6×6)

In the present work, problem no. 1 is solved by using new operating parameters [Refer to Table no5.1]. The make span for problem no. 1 is reduced to 49 which can be shown in Fig. 4.2 and 4.3.

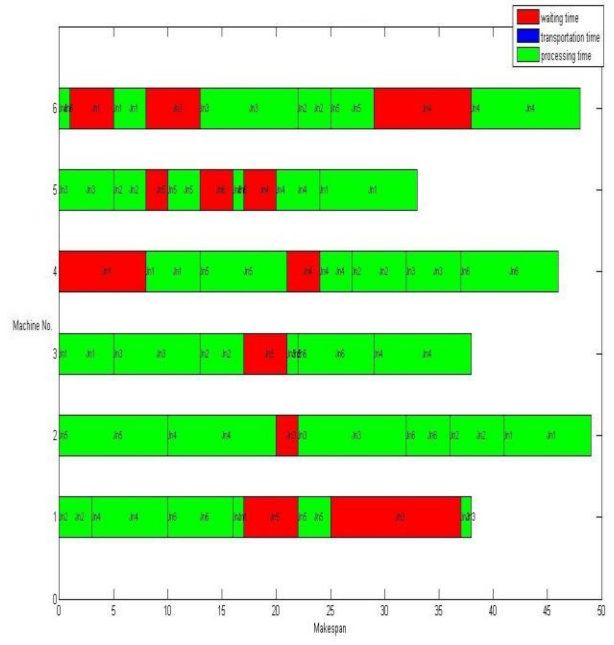

Fig. 4.2 Gantt chart of Experiment No. 1

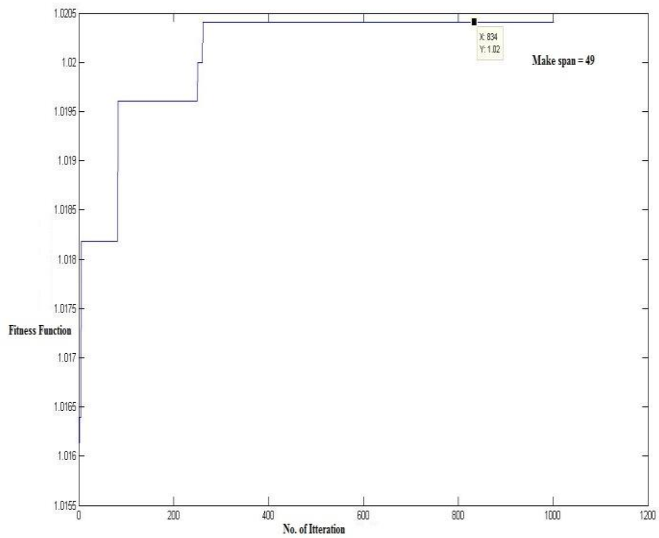

Fig. 4.3 Convergence curve of Fitness Function of Experiment No. 1

Problem No. 2: Rao et al. (2012), No of machine (m) = 5, No of job (n) = 10, makes span = 152

In this research, a new meta-heuristic solution approach for Multi-objective Job Shop Scheduling Problems (JSSP) was presented. The proposed algorithm made a use of Mehrabian

& Lucas's heuristic 'Invasive Weed Optimization' (IWO) in generating optimal schedules. For performance evaluation of solutions in a Multi-objective scenario, a concept called' Fuzzy dominance' has been employed. The results obtained show that the proposed algorithm can be used as a new alternative solution technique for finding good solutions to the complex Multi-objective Job Shop Scheduling problems. In this work, the algorithm has been programmed for JSSP using MATLAB with a maximum allowable population size ie.20×no of operations. The makes span for job shop problem came out was 165.

In the present work, problem no. 2 is solved by using new operating parameters [Refer to Table no 5.1]. The make span for problem no. 2 is reduced to 152 which can be shown in Fig. 4.4 and 4.5.

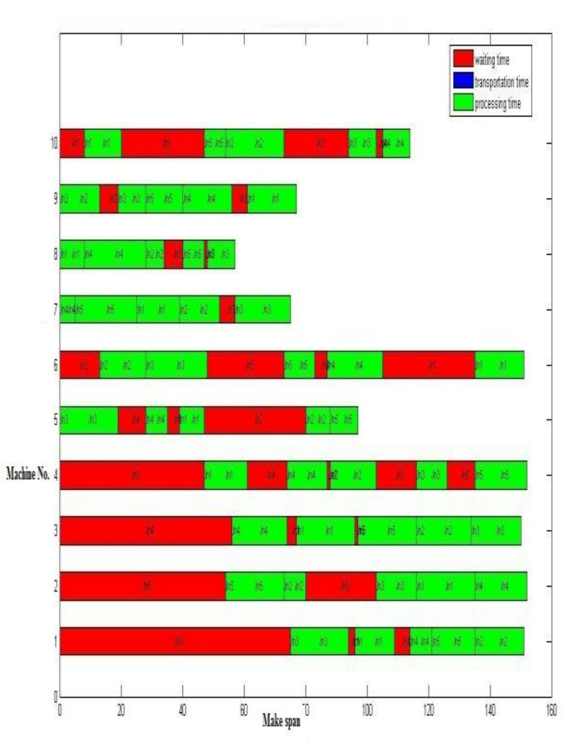

Fig .4.4 Gantt Chart of Experiment No.2

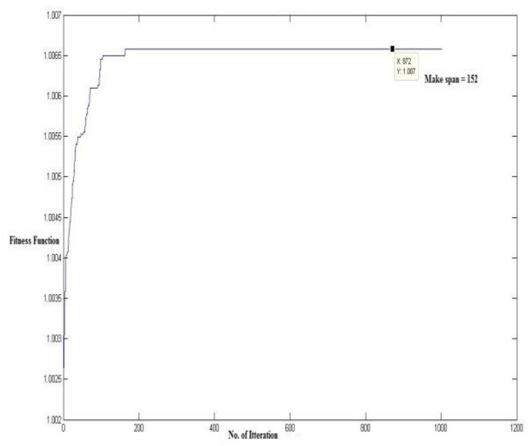

Fig.4.5 Convergence curve of Fitness Function of Experiment No. 2

Results and Discussion

Two different problems based on Job Shop Scheduling (JSS) are taken with different operating parameters [refer to table 3.1]. A modified GA is then proposed to optimize the make span by taking new operating parameters. In order to verify the validity of proposed algorithm, a computer programme using MATLAB is developed and run for seven different problems with their operating parameters which gives the same value of make span. The new operating parameters to solve the JSSP are given as:

Table 5.1 Operating Parameters used in present work

Parameters	**Values**
Population Size	50
Selection Scheme	Tournament selection
Cross over	2-Point Random ($P_c = 0.8$)
Mutation	2-Point Random ($P_m = 0.2$)
Termination Criteria or No. of Iteration	1000

The application of Genetic Algorithm (GA) with new operating parameters to two different problems gives an improvement in the make span as illustrated in Table 5. 2

Table 5.2 Make Span of Seven different problems

Problem No.	**Author Name & Year**	**Old Makespan**	**Proposed Approach**	**Percentage Improvement**
1	Fang, (1993)	55	49	10.9 %
2	Madivada, (2012)	169	152	10.05 %

Conclusions & Scope for Future Work

In this research, a promising genetic algorithm for job shop scheduling problems is proposed. The traditional genetic algorithm is modified by a random population generation with new operating parameters i.e. Population-size 50, operation based chromosome structure, tournament selection as selection scheme, 2-point random crossover with a probabilities 80 % (Pc = 0.8), 2-point random mutation with a probabilities 20 % (Pm=0.2), elitism, repairing of chromosomes and no. of iteration is 1000. The algorithm has been programmed for Job Shop Scheduling Problems (JSSP's) using MATLAB 2009 a7.8 and bench mark problems have been tested for make span and validation of computer codes developed.

The following conclusions can be drawn from this work:

1) A modified Genetic Algorithm (GA) with new operating parameters is proposed for optimizing the Job Shop Scheduling Problems (JSSP's) in terms of makespan.

2) Results have shown that the proposed Genetic Algorithm (GA) is able to obtain the optimum or a useful near optimum value of make span.

3) The proposed Genetic Algorithm (GA) clearly outperforms the best known exact and approximate approach.

Future work can also be done by considering the following points:

Maintenance time and setup time is assumed constant, further work can be done with varying setup time and maintenance time.

Actual breakdown can be done by taking into account for solving the JSSP's.

The jobs are considered to be atomic. Further work can be done by creating schedules with various parts manufactured on different machines.

The proposed algorithm can also be applied to more practical and integrated problems.

References

[1] Dirk et al.(2004) "An efficient genetic algorithm for job shop scheduling with tardiness objectives" European Journal of Operational Research 155, 616–630.

[2] Linyan Sun et al. (2006)"Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm" J Intell Manuf 17,493–507.

[3] Mati et al. (2008) "A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem" Computers & Operations Research 35, 2599 – 2616.

[4] Wang et al, (2008) "A two-stage genetic algorithm for large size job shop scheduling problems" Int J Adv Manuf Technol 39:813–820.

[5] Rao et al.(2008) "An effective hybrid genetic algorithm for the job shop scheduling problem" Int J Adv Manuf Technol 39:965–974.

[6] Yih-Long et al.(2009)"Multi-criteria sequence-dependent job shop scheduling using genetic algorithms" Computers & Industrial Engineering 56,179–185.

[7] Jinwei et al. (2009)"*A novel parallel quantum genetic algorithm for stochastic job shop scheduling*" Journal of Mathematical Analysis and Applications J. Math. Anal. Appl. 355, 63–81**.**

[8] Andrea Rossi (2009) "A hybrid heuristic to solve the parallel machines job-shop scheduling problem" Advances in Engineering Software 40,118–127.

[9] Zandieh et al.(2009)"Scheduling sequence-dependent setup time job shops with preventive maintenance" Int J Adv Manuf Technol 43:170–181.

[10] Deming Lei (2010) "Solving fuzzy job shop scheduling problems using random key genetic algorithm" Int J Adv Manuf Technol 49:253–262.

[11] Defersha et al.(2010) "A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups" Int J Adv Manuf Technol 49:263–279.

[12] Guohui Zhang (2011)"An effective genetic algorithm for the flexible job-shop scheduling problem" Expert Systems with Applications 38, 3563–3573.

[13] Rubiyah Yusof (2011)"Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm" Applied Soft Computing 11,5782–5792.

[14] Zejko et al (2011) "Solving the no-wait job-shop problem by using genetic algorithm with automatic adjustment" Int J Adv Manuf Technol 57:735–752.

[15] Oddi et al. (2011) "Solving job shop scheduling with setup times through constraint-based iterative sampling: an experimental analysis" Ann Math Artif Intell 62:371–402.

[16] Teekeng et al.(2012) "Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems" Procedia Computer Science 12,122 – 128.

[17] Manier et al.(2012) "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times" Computers & Operations Research 39,1713–1723.

[18] Ren et al.(2012)"*A new hybrid genetic algorithm for job shop scheduling problem*" Computers & Operations Research 39, 2291–2299.

[19] Zhou et al.(2012)"An effective artificial bee colony algorithm for the flexible job shop scheduling problem" Int J Adv Manuf Technol 60:303–315.

[20] Sun et al.(2012)"Job shop scheduling based on earliness and tardiness penalties with due dates and deadlines: an enhanced genetic algorithm" Int J Adv Manuf Technol 61:657–666.

[21] Li et al. (2013) "A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem" Int J Adv Manuf Technol 66:583–596.

[22] Rui Zhang (2010) "A hybrid approach to large-scale job shop scheduling" Appl Intell 32: 47–59.

[23] Wang et al(2013) "A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions" Int J Adv Manuf Technol DOI 10.1007/s00170-013-4923-z.

[24] Rau et al (2013)"A genetic algorithm for job-shop scheduling with operators enhanced by weak Lamarckian evolution and search space narrowing" Springer Science+Business Media Dordrech.

[25] María et al (2013)"New schedule generation schemes for the job-shop problem with operators" J Intell Manuf DOI 10.1007/s10845-013-0810-6.

Pardeep Kumar received his B. Tech. degree in Mechanical Engineering from Panipat Institute of Engineering & Technology, Samalkha, Panipat, Haryana, India, under Kurukshetra University, in 2011, and M. Tech. degree in Mechanical Engg. From U.I.E.T. kurukshetra university. Pardeep Kumar is an Assistant Professor,,with Department of Mechanical Engg. in Panipat Institute of Engineering and Technology, Samalkha Near Panipat (Haryana).). He has more than 3 years teaching experience in Mechanical Engineering.

Pawan Kumar received his B.Tech degree in Mechanical Engineering from N.C. College of Engg. Israna Panipat, Kurukshetra University, in 2008, the M.Tech degree in Manufacturing Engineering from MMU, Mullana-Ambala, Haryana in 2012. He is Assistant Professor, with Department of Mechanical Engg. In Panipat Institute of Engineering & Techanology, Samalkha Near Panipat (Haryana). He has more than 7 years teaching experience in Mechanical Engineering.

Ram Bhool received his B. Tech. Degree in Mechanical Engineering from Shree Ram Mulkh Institute of Engineering and Technology, Naraingarh (Ambala) under Kurukshetra University, in 2011, and M. Tech.Degree in Mechanical Engg. From Deenbandhu Chhotu Ram University of Science and Technology, Sonipat, (Haryana) in 2013. Ram Bhool is an Assistant Professor, with Department of Mechanical Engg. in Panipat Institute of Engineering and Technology, Samalkha Near Panipat (Haryana)

Virender upneja received his B. Tech. Degree in Mechanical Engineering from JMIT, radaur under Kurukshetra University, in 2006, and M. Tech.Degree in Mechanical Engg. From Thapar university Punjab in 2012. Virender upneja is an Assistant Professor, with Department of Mechanical Engg. in Panipat Institute of Engineering and Technology, Samalkha Near Panipat (Haryana). He has more than 9 years teaching experience in Mechanical Engineering.