SOLVING OPEN SHOP SCHEDULING BY HYBRID GENETIC ALGORITHM WITH SIMULATED ANNEALING

A Thesis Presented to the
Department of Computer Sciences and Software Engineering
Faculty of Basic & Applied Sciences

In Partial Fulfillment of the requirement of
Master of Sciences (Computer Sciences)
By

Adeeba Zulfiquar
(674-FBAS/MSCS/S12)
International Islamic University Islamabad (2014)
Faculty of Basic & Applied Sciences
Department of Computer Science

SOLVING OPEN SHOP SCHEDULING BY HYBRID GENETIC ALGORITHM WITH SIMULATED ANNEALING

Researcher

Name: Adeeba Zulfiquar

Reg # 674/MSCS/S12

Supervisor

Name: Fareeha Anwar

Designation: Lecturer

Department of Computer Sciences and Software Engineering
International Islamic University Islamabad

FINAL APPROVAL

It is certificate that we have read the thesis submitted by Ms Adeeba Zulfiquar and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the M.S Degree in Computer Sciences.

COMMITTEE

External Examiner

Prof. Dr. Sikander Hayat Khiyal

Professor, Faculty of computer science

Preston University Islamabad

Internal Examiner

Sadia Arshad

Lecturer, Faculty of computer science

International Islamic University Islamabad

Supervisor

Fareeha Anwar (Lecturer)

Lecturer, Faculty of computer science

International Islamic University Islamabad

A thesis submitted to Department of Computer Sciences and Software Engineering,
International Islamic University, Islamabad as a partial

Fulfillment of requirement for the award of the

Degree of MS computer Science

DECLARATION

I hereby declare that the work present in the following thesis is my own effort, except where otherwise acknowledged and that the thesis is my own composition. No part of the thesis has been previously presented for any other degree.

Date ________

Adeeba Zulfiquar

____

CONTENTS

1. Introduction..3
1.1. Scheduling Problems...3
1.2. Importance of scheduling problems in manufacturing industry........................4
1.3. Scheduling problem example..5
1.3.1. Shop scheduling problem..5
1.3.1.1. Job shop problem 6
1.3.1.2. Flow shop problem...6
1.3.1.3. Open shop problem 6
1.3.1.4. Flexible job shop problem..6
1.4. Different criteria for measuring efficiency..6
1.4.1. Makespan minimization...6
1.4.2. Total job tardiness minimization...6
1.4.3. Job lateness minimization..7
1.4.4. Job flow time minimization...7
1.4.5. Mean flow time minimization..7
1.4.6. Importance of make span criteria...7
1.5. Open shop problem and its importance..8
1.5.1. Problem formulation...8
1.6. Methods for solving open shop scheduling problem...................................9
1.6.1. Exact methods..10
1.6.2. Approximate Algorithm...11
1.7. Genetic Algorithm...11
1.8. Simulated Annealing..13
1.9. Hybridizing GA and SA for open shop scheduling......................................13
1.10. Problem Statement...14
1.11. Research Questions..14
1.12. Organization of thesis...14
2. Literature Review...16

2.1. Exact procedures……17
2.1.1. Branch and bound method……17
2.1.2. Constraint propagation method……19
2.1.3. Linear Programming method……19
2.2. Approximate Algorithm……20
2.2.1. Heuristics……22
2.2.2. Meta Heuristics……24
3. Research Design……31
3.1. Proposed Hybrid Genetic Algorithm……31
3.1.1. Representation……32
3.1.2. Selection method……33
3.1.3. Cross over method……34
3.1.4. Mutation method……35
3.2. Local optimization heuristics……35
3.2.1. Temperature……35
3.2.2. Metro polis criteria……35
4. Case Studies……36
4.1. Case study 1……38
4.2. Case study 2……40
5. Experimentation and conclusion……44
5.1. Experimental results and discussion……45
5.2. Conclusion……52
References……53

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENT

"To Him belongs the dominion of the Heaven and the Earth, it is He who gives life and death and He has power over all things"
(Al-Quran)

In the name of **ALLAH**, the most gracious, most merciful and most beneficial who endowed us with the will and thoughts to complete this project with the praise of his beloved **Holy Prophet (PBUH).**We have heartfelt appreciation for our respectable instructor **Ms Fareeha Anwar** who kept our morale high by his suggestions and appreciation. Without his precious guidance and help we could never be able to complete this project. We would like to acknowledge parents for their support and courage. We would like to admit that we owe all our achievements to our truly, sincere and most loving parents, who mean the most to us, and whose prayers are a source of determination for us .Thanks as extended to the commuters who helped us during our project and their co-operation in data collection. In the end we would like to say thanks to all of those who helped us in any way during our project and made it easier for us to reach our aim.

LIST OF ABBREVIATION

Word	ABBREVIATION
Genetic Algorithm	GA
Simulated Annealing	SA
Open shop scheduling problem	OSSP
Open shop scheduling	OSP
Flow shop problem	FSP
Job shop problem	JSP
Flexible job shop problem	FJSP
Cmax	Make span minimization
Branch and bound method	BB
Linear Programming	LP
Tabu search	TS
Partially mapped crossover	PMX
LISA	Library of scheduling algorithm

LIST OF FIGURES

Figure NO. Caption.................................. Page Number

Figure 1.1. Solution Approach for open shop scheduling problem..............................10

Figure 1.2. Simple Genetic Algorithm..12

Figure 5.1. Snap shot of LISA..45

Figure 5.2. Snap shot for generating processing time...45

Figure 5.3. Snap shot for selecting number of jobs..46

Figure 5.4. Snap shot for selecting number of machines.......................................47

Figure 5.5. Snap shot for initial parameters...47

Figure 5.6. Best makespan value...48

LIST OF TABLES

Table no. Caption.................................. Page Number

Table 3.1. Representation of job machine combination..................................33

Table 4.1. 6x2 open shop scheduling problem..38

Table 4.2. Operation times...39

Table 4.3. Machine Allocation..39

Table 4.4. Allocation of jobs...40

Table 4.5. Colors Assigned to jobs...40

Table 4.6. BSCS course Detail..41

Table 5.1. Problem Size..44

Table 5.2. Instances.. 51

ABSTRACT

This research solves the problem of open shop scheduling problem by minimizing makespan. Hybrid approach is adopted in this research by combining genetic algorithm with simulated annealing. Firstly two case studies were used to evaluate the results. Then instances generated by LISA were used to perform comparison with previous results. Optimal solution is found in most cases including improvement in the previous best results. Computational results shows that this technique competes the previous algorithm in many cases and very competitive when compared to other Meta heuristics available in the literature.

CHAPTER 1

INTRODUCTION

1-INTRODUCTION

In shop scheduling environment open shop scheduling is one of the most difficult combinatorial problems. Several techniques have been used to solve this tricky phenomenon. In this research work open shop scheduling problem has been solved by using genetic algorithm with simulated annealing.

1.1: Scheduling problems

Scheduling problem arises in every field of life therefore research in scheduling has been a vast area for many decades. There is a lot of research dealing with scheduling problems arising in many situations. They are known as NP-hard and need complex algorithms because they are difficult to solve. In scheduling algorithm we have to choose from different options available therefore selecting appropriate solution play a vital role in today competitive world. It has also become important to produce efficient scheduling algorithms that guide us in our goals.

Scheduling is a method in which process are assigned to resources to find an optimal schedule satisfying most of the constraints. There is a need to specify the objective we want to optimize otherwise research will be undirectional. Different criteria are used to measure the performance of scheduling algorithms such as meeting due dates, customer satisfaction, quality of product, inventory cost, etc. the efficiency of all these criteria is totally dependent on scheduling algorithm.

Scheduling decisions has 4 primary stages called analysis, synthesis, formulations and evaluation [5]. Mathematical models are used to build the scheduling algorithms which are related to the mathematical functions and algorithms. The development of the mathematical algorithm is the interface between theory and interface.

Feasible schedule of scheduling problems contains 2 types of limitations namely availability of assets and restrictions on the order of tasks. Different techniques are available for solving scheduling problems which include Constraint Programming method, Local Search method, Mathematical Programming, heuristics, Meta heuristics etc. Most of these techniques are for specific problems and environment, and they need a lot of expansion to adapt them to different scheduling problems and real life problems as well. Some techniques can be used for general (e.g., Constraint Programming) purpose, but they fail to work on larger number of instances. Clever heuristics are the solution of this problem.

In the process of scheduling there is a race between different jobs for resources. As in the process of scheduling the number of jobs and resources increased complexity of the problem is also increased.

Basically scheduling problem is just like a puzzle solving activity in which lot of options are available but there is only one optimal solution. In scheduling problem we have problem constraints and objective criteria as well. The need is just to figure out how to best utilize the capacity over time.

Scheduling problems are known as optimization problems. Optimization problems can be defined in a more general way. The problem is extensive to how to choose the best one when several substitute options are available. The main concept of scheduling is concerned with how to efficiently schedule these problems to obtain the best outcome.

Different methods have been developed in the past to solve different types of scheduling problems in different shop environment for different objectives depending upon the situation. They include conventional methods such as branch and bound, mathematical programming and priority rules to metaheuristics and artificial intelligence (AI) based methods as well.

Scheduling problem has to work on these important three points

1. Satisfy all restrictions on the execution of any problem

2. Satisfy all restrictions on the use of resources

3. Find optimal solution keeping in view the objective

1.2: Importance of scheduling problems in manufacturing industry

Scheduling problems have a vital role in manufacturing industry now a day due to the growing consumers demand for diversity; reduce product life cycles, changing market with global completion and rapid availability of new technologies. Actually scheduling is a decision making process because of contradictory goals, limited number of resources and the main problem is how to tackle these real word problems.

In an industrial job assignment problem, scheduling activities are known as operations, and resources are known as machines. Problems in industrial environments are mostly combinatorial. Scheduling is a difficult task. So it is particularly difficult to solve scheduling problems. The scheduling problem is tremendously difficult to solve in which resources must be assigned to the required job keeping in view the objectives. We have to maximize machine utilization and minimize the time required to carry out the whole process. [3].

Conventional optimization problems are failed to solve manufacturing industry problems as these problems are quiet complex and huge in nature. The complexity of these problems has a direct dependence on restraints and shop scheduling environments upon which these problems are scheduled. In the field of scheduling problems open shop scheduling is the hardest one combinatorial problem because it has a large solution space as compared to other shop scheduling problems. The purpose of scheduling in manufacturing industry is to improve machine utilization and reduce product cycle times.

1.3: Scheduling Problem Example

It is not possible to review all scheduling problem in this a short span of time therefore some well known shop scheduling problems will be discussed here

1.3.1: Shop scheduling problems

In shop scheduling problems we have n jobs and m machines. In which each job consists of number of operations, job represents activities and machine represents resources. One job can be processed by one machine at a time. Objectives of shop scheduling contains

1. Consumption of time should be less than or equal to some constant T
2. Minimize the time required to complete the jobs

These problems has been studied for a long instance, and solved with a large variety of methods. It also raises a large number of variations discussed below

1.3.1.1: Job Shop Problem (JSP)

In a job shop scheduling operations of a job are totally ordered.

1.3.1.2: Open Shop Problem (OSP)

In open shop scheduling operations of a job can be executed in any combination. Or it is a special type of job shop scheduling in which operation can be carried out in any sequence. Its mean that in open shop scheduling problem there is no ordering restrictions on operations.

1.3.1.3: Flow Shop Problem (FSP)

It is a special case of job shop scheduling problem in which all jobs have to follow the same defined order.

1.3.1.4: Flexible job shop problem (FJSP)

In this problem, the requirements of machines are changed. Each job may use any resource among a given set. Different degrees of suppleness exist for example there may be alternative different machines available for doing the same work. Many others also exist which are actually the modified versions of the discussed problem need not to be discussed here.

1.4: Different Criteria for measuring efficiency

Different criteria's are used to measuring efficiency described below

1.4.1: Makespan Minimization

This measure is known to be a controlling measure. In makespan minimization the assignment is to assign jobs to machines in such a way so that the total time taken to complete the whole process, also called the makespan is minimized. It can also be said that we want to minimize the maximum total processing time of any machine. It can also called measure of efficiency.

1.4.2: Total job tardiness Minimization

How long after the due date a job was completed, measures due date performance. Before starting any project due date is guessed or sometimes given by customer. So this measure tries to minimize the due date performance means this measure tries to complete the job within the time period provided by customer.

1.4.3: Job lateness Minimization

This measures tries to compute the time on which job is completer of on or behind the schedule. It can be calculated by using following formula

Average lateness = total late days/number of jobs

The aim is to minimize the jab lateness.

1.4.4: Job flow time minimization

This measure tries to compute the total processing time and waiting time or in other words time a job is completed minus the time the job was first available for processing. It can be calculated by using the following formulas

Average Mean flow time = (Total processing time + total waiting time)/Number of jobs

1.4.5: Mean Flow time minimization

Mean flow time tries to minimize sum of completion time for all jobs. It can be calculated by using following formula.

Mean Flow Time= (sum job flow times)/ Number of jobs

These are some important criteria which has been used mostly in the literature so far for measuring efficiency although there are others measures too such as Machine utilization, inventory cost, , customer satisfaction, quality of products etc.

1.4.6: Importance of makespan criteria

Total time taken by any job to complete all operations on all available resources is called makespan. It lays between the starting times of the first operation of any job to the last operation of any job. The objective of makespan minimization is to find out a schedule that gives up the minimum makespan. It is also possible that we are going to have multiple solutions that have the minimum makespan time, but the objective is to find out any one of best schedule in minimum amount of time.

Several measures of performance has been taken by several researchers as discussed above, there is still a lot of scope for the central measure, known as make span minimization, because there is very little work considering open shop problem for make span minimization. Further, this particular measure is considered to be a controlling measure because the completion time of jobs (make span) has direct posture on shop floor planning.

1.5. Open shop problem and its importance

There is a lot of difference between open shop job shop and flow shop yet they seem same. In open shop operations of a job are carried out in any order. There are many situations in which open shop scheduling problem applies such as testing and repair. For example in an automobile plant configuration there is lot of large quality center with specialized testing stations. Each new device requires engine, transmission system and test of electronics. These all tests can be performed in any sequence. Sequence does not matter in this case. Other real life examples include time table scheduling problem, class assignment problem, and exam scheduling problem, auto mobile repair, quality control center, satellite communication, automobile repair and some operations of manufacturing heat exchange. [25].

There is lot of literature available for multi criteria and multi objective job shop and flexible job shop scheduling problem however there is very little work on open shop scheduling with multi objectives. So this specific area also needs a lot of attention from the researchers.

1.5.1: Problem formulation (Open shop)

Formulation of open shop scheduling problem is given below

There is a set of n jobs {j1,j2,.......jn}has to processed on m machines {m1.m2,m3.......mn}. each job consist of m operations{m1,m2,m3....mn}. operations of a job are carried out in any order. Each job can be process by only one machine at a time simultaneously each process can be executed only on one machine at one time. All jobs are available at the start of process. Preemption is not allowed.

The objective of open shop scheduling problem is to find a minimum completion time called makespan minimization.

1.6: Methods for solving open shop problem

It is the need of hour to develop efficient method for solving OSSP especially for large instances. Moreover, OSSP is acknowledged as one of the most challenging NP-hard problems and there is no any algorithm can be engaged to solve OSSP consistently even when the numbers of instances are small. So it has taken the attention of researches.

Two approaches have been developed to solve scheduling problems. First method is to find the exact solution of any problem called exact method. While second approach is to find a optimal solution called approximate algorithm. Figure 1 elaborates different methods for solving open shop scheduling in detail.

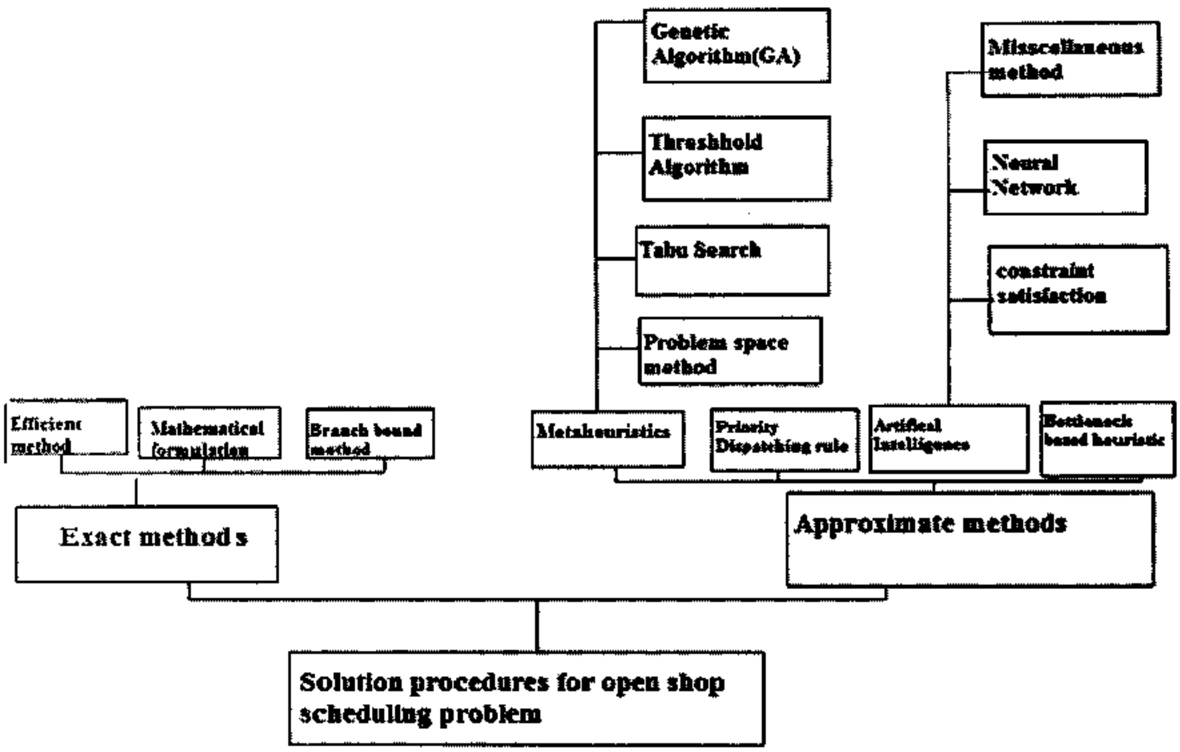

Figure 1.1: **Solution approach for open shop scheduling**

In figure 1 two main methods have been defined for solving open shop scheduling problem called exact and approximate methods. Exact methods include mathematical formulation, branch and bound etc. while approximate methods include metaheuristics, priority dispatching rule, artificial intelligence and bottle neck based heuristics.

1.6.1: Exact methods

Exact methods will always find the global optimum and recognize it too. That being said, they don't scale (not even beyond toy problems) and are therefore mostly useless. The most general exact solutions method mostly uses mathematical programming formulations, followed by branch and bound method and elimination method to obtain optimal schedule.

Exact methods, shown in figure 1.1 guarantee to find global convergence and have been successfully applied in solving small problems. When the number of instances increase or problem size becomes larger then these methods works to fail. So a lot of researcher tries to solve these problems by Meta heuristics and intelligent hybridization scheme for open shop scheduling problem.

1.6.2: Approximate Algorithm

Approximate algorithms include both heuristics and metaheuristics. Theses algorithms do not guarantee optimal solution but near optimal is found. Heuristic uses domain knowledge to explore the search space. These methods are actually based on estimation. While metaheuristics is the combination of heuristic and some algorithm

Due to the complexity of OSSP, exact techniques, such as branch and bound method [4, 6], and dynamic programming [13] are only applicable to modest scale problems. Most of them fail to obtain good solutions solving large scale problems because of the huge memory and lengthy computational time required. On the other hand, heuristic methods include shifting bottleneck, Lagrangian relaxation and dispatching priority rules approach, are alternatives for large scale instance. With the materialization of new techniques from the field of artificial intelligence (AI), much attention has been taken by researchers to meta-heuristics. One main group of meta-heuristics is the construction and improvement heuristic, such as simulated annealing [17] and tabu search [15-17]. Population based heuristic is the other class of metaheuristics. Successful examples of population based algorithms include genetic algorithm (GA) [16, 17], particle swarm optimization (PSO) [27], Ant colony [18] and so on.

Optimization problems have been solved by metaheuristics techniques successfully. There are some cases in which common search algorithms could not perform well especially in the case when the searching area is large. In such circumstances we have to develop special metaheuristics depending upon the nature of the problem or hybridization with problem specific search technique which improves the results [2]. By hybridization two goals of local optimal solution and global optimal solution can be achieved.

Therefore by combining genetic algorithm which have global convergence ability and simulated annealing which have local convergence ability a new algorithm is proposed for solving open shop scheduling problem.

1.7: Genetic Algorithm (GA)

Genetic algorithms are inspired by biological evaluation. Genetic algorithm is the most popular one among population based algorithms. Among the entire evolutionary algorithms Genetic algorithm is most suitable for combinatorial optimization problem because it contains more simulation to living system than other evolutionary algorithm. Genetic algorithm starts its search concurrently from many initial points [26].

There are many cases in which simple genetic algorithms are not capable of finding optimal or near optimal solutions within a reasonable time frame. It is responsible for evaluating chromosomes. In genetic algorithms solutions mostly are encoded as bit string. To improve the quality of genetic algorithm hybridization can be done.

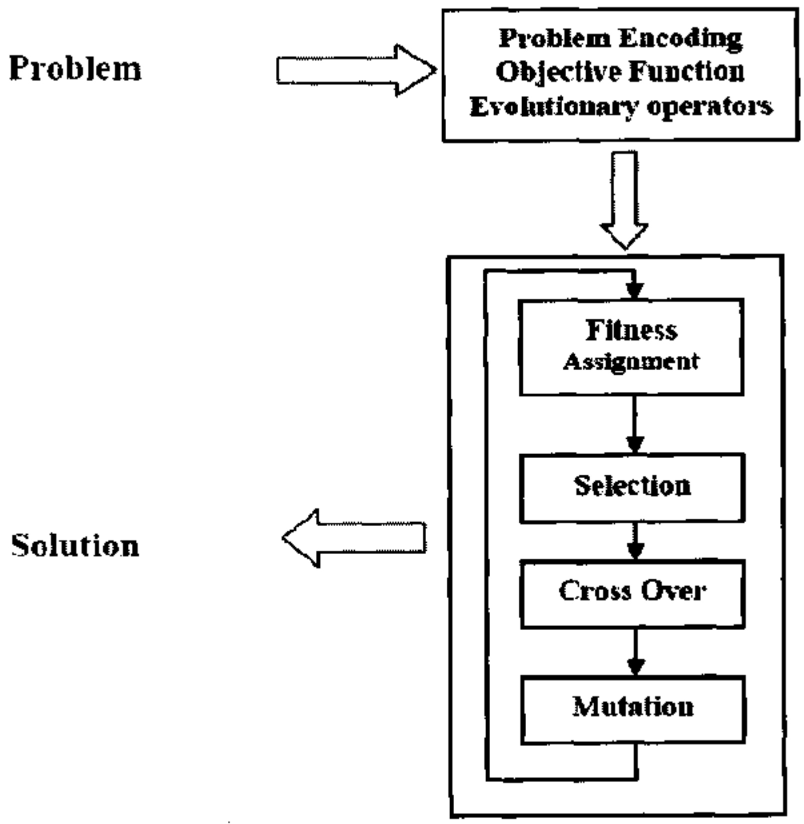

Figure 1.2: **Simple Genetic Algorithm [17]**

To successfully apply a GA to solve a problem one needs to determine the following [17]

1. Encoding technique (gene, chromosome)
2. Initialization procedure (creation)
3. Evaluation function (environment)
4. Selection of parents (reproduction)

1.8: Simulated Annealing (SA)

This method is inspired by annealing in metal process [17]. To get a pure structure metal is heated up very powerfully and then cooled slowly. So that number of irregularities becomes minimal. High temperatures pick up the pace of the movement of particles. During the cooling process optimal solution is obtained. Annealing refers to a process in which metal is raised to high energy level and then cooled gradually to obtain a solid state. Purpose of this process is to arrive at the lowest energy state. Physical substance moves from higher state to lower state if the cooling process is adequately slow.

Simulated annealing contains Monte Carlo method and this method is specially suited for simulated annealing. Metropolis Monte Carlo is the simulation of simulated annealing that starts at a high temperature. To become the search smaller or feasible temperature is reduced slowly and in the low temperature condition the system is in feasible state [29].

A basic simulated annealing algorithm starts its process by generating initial solutions taken as the current starting solution. Then a neighbor is generated and difference is found between these two solutions. The new solution is accepted if it has a better solution value. If generated objective function does not decrease with the pace of iteration then the new solution can also accepted by using a probability called temperature. The temperature is slowly reduced to enter into a feasible or optimal solution.

1.9: Hybridizing Genetic algorithm and simulated annealing for open shop scheduling

In this research simulated annealing is hybridized with genetic algorithm for open shop scheduling. Genetic algorithm can find global optimistic result in reasonable time while simulated annealing can find local optimistic result. And it can also avoid the problem of local minimum. Combining Genetic algorithm and simulated annealing, learning from other's strong points to counterbalance one's weaknesses each other, this is the main idea of proposed solution.

1.10: Problem statement

Based on all above discussion the research problem in this thesis is to solve open shop scheduling problem for minimizing makespan by using hybrid genetic algorithm with simulated annealing.

1.11: Research Objectives

The objective of the research described here is

1. To determine the best job schedule by minimizing makespan
2. To efficiently solve the open shop scheduling described above

1.12: Organization of Thesis

This thesis is organized as follows: Chapter 2 gives a detailed literature review. Chapter 3 presents the problem description and the methodology followed to develop the algorithm called hybrid genetic algorithm with simulated annealing. Chapter 4 explains the problems tested and experimentation to evaluate the effectiveness of the hybrid genetic algorithm model. Chapter 5 provides the results for the experimentations followed by the discussion of trends observed. Finally, chapter 6 provides the conclusions and future research directions

CHAPTER 2

LITERATURE REVIEW

2-LITERATURE REVIEW

Many techniques have been developed in the literature for solving open shop scheduling problems. As compared to other shop scheduling problem open shop problem receives less attention from researchers because it is most complex combinatorial optimization problem in shop scheduling problems. In open shop environment operations of a job can be executed in any order. Therefore number of possible solution in the search space increased which results in increasing the complexity of problem. So it is said about open shop that it belongs to the class of difficult tractable problems

Mainly there are two techniques which have been designed for solving open shop scheduling problem. One method is to find the exact solution for the problem called exact approach another method is to find near optimal solution called approximate approach. Generally most exact approaches follow mathematical programming formulation, elimination method and branch and bound method to obtain exact solution for the problem [4]. However exact approaches only solve small instances to optimality while for medium to large size problem use of these approaches is inadequate because lot of time is needed to solve such problems.

Approximate algorithms include both heuristics and metaheuristics. Experience-based techniques for problem solving method are known as heuristics while the brute force search is not possible. Mental shortcuts were used to find the optimal solution knows as heuristics. Having vast ides of search space is required to apply heuristics method. Only experienced person can perform this process. It could be good guess, rule of thumb, a judgment, or commonsense

Metaheuristics are designed to handle complex combinatorial problems. In past special heuristics were used to handle complex problems. So every time a new approach was needed to solve a specific problem. With the appearance of new algorithms that are for general purpose researchers paid their attention to these algorithms for solving complex combinatorial problems. Genetic algorithm, tabu search, simulated annealing etc can be applied generally to any problem. Only we have to represent the problem according to algorithm nature. This requires a less effort and produces significantly better results. Also near optimal solution can be obtained in reasonable amount of time.

2.1: Exact Procedures

Exact methods will always find the global optimum and recognize it too. That being said, they don't scale (not even beyond toy problems) and are therefore mostly useless. The most general exact solutions mostly use of mathematical programming formulations, followed by branch and bound method and elimination method to obtain optimal schedule.

Exact methods, such as linear programming, Lagrangian relaxation and branch and bound assure global convergence and have been successfully applied for solving smaller instances. Computation time increase with the large problem size and there comes a situation in which these methods could not find exact solution.

The most common exact method is branch and bound method described below

2.1.1: Branch and Bound method

Branch and bound method is a general way to find an optimal solution without an exhaustive search. This method consists of branching. Branch and bound method follows two techniques 1st method is to generate branches when searching the search space 2nd way is to generate a bound so that the branches can be terminated. The Problem is divided into set of active schedule and each active schedule contains an optimal solution. In solution method all active schedules are generated and best schedule is selected. For improvement generation scheme is used in branch and bound method.

It will work efficiently in the average case because many branches can be terminated at early stage. But in the worst case complexity a very large tree can be generated.

Brucker et al. [4] have developed a branch and bound algorithm based on disjunctive graph formulation of the problem. Computational results showed that it was quiet effective for small instances. This technique is still unbeatable in all branch and bound method still proposed but some problem from size 7 remains unsolved. As this technique used depth first search tree which uses chronological backtracking algorithm to explore the search space. Such an exploration leads to useless branch trees. It is impossible to solve NP-hard problems with branch and bound methods because they are exact algorithms and can works only for small problems. Therefore heuristics and metaheuristics have been widely applied to solve these problems.

Another branch and bound algorithm was proposed by Gueret and Prins [6] .Concerning its practical value new lower bound was computed very quickly despite the NP-hardness. An obvious solution will be a branch and bound algorithm for the general job shop problem. In these context researchers has generated open shop problem generator designed to create instances with large gaps between the new bound and classical lower bound. Generated instances are very hard since the best branch and bound method developed by Brucker et al [4] fails to solve most instances of size 7×7 and larger. To speed up the computations and discards the unnecessary schedules simple heuristic is applied to get an upper bound. This heuristic is fast and of greedy version. The major drawback of the proposed algorithm is that researchers considers only square instances (m=n) in which number of jobs (n) is equal to number of machines (m). While instances in which m>n and m<n are also very important.

Gueret et al. [12] gave an idea to improve branch and bound method. This method was based on intelligent backtracking system applied on branch and bound method which is an application of open shop problem. Researcher's presents an improved technique based on branch and bound method applied to Brucker algorithm [4] for open shop problems.

This method was very efficient to avoid useless exploration of the search which does not contain optimal values which favors the algorithm in reducing the time. This method would be also applicable to other branch and bound method which have already been applied to solve job shop scheduling problem and flow shop scheduling problem for which the branching scheme consist in fixing disjunction. The spatial complexity of approach was strongly related to the depth of search tree. As the depth of tree increases the computational time also increases. Benchmarks which were used in this study to evaluate the result were composed of 10 square instances of size 4, 5, 7 and 10. Among the problem of size 10 four are still unsolved. This technique does not work well for small problems. The other problem with this method is that it was applied on only 40 instances so it is difficult to generalize the problem with only 40 instances among 4 were unsolved.

Best available method in literature could not solve the problem of 7x7 [4]. The optimal schedule for open shop scheduling problems can be easily found for m=1. For m=2 there are some polynomial time algorithms. As the open shop is NP-hard for m>=3 [10].

2.1.2: Constraint propagation method

To reduce the search space of combinatorial problems and optimization problems constraint propagation is the primary method. The basic concept used behind constraint propagation method is to get all solutions for the problem and then discard all those solutions that conflict in all adjacent. Optimal solution is that which satisfy all the constraints. Main advantage of using this technique is that it is declarative in nature and cooperative problem solving as well. Weaknesses include unpredictable behavior of the algorithm, non incremental, weak solver collaborator.

An exact algorithm proposed by Dorndorf et al. [21] based on constraint propagation method for reducing the search space. Benchmark problems were solved first time by using this method. In particular problems given by Taillard [7] all instances with 10 jobs and 10 machines all but one instance with 15 jobs and 15 machines and 7 instances with 20 jobs and 20 machines have been solved. Some exact algorithms have been proposed in the literature that guarantees to be an optimal solution to the open shop scheduling problem described below. Recently an exact algorithm has been proposed by Laborie [22] and Tamura et al. [23].

2.1.3: Linear programming method

Linear programming is based on linear equations of mathematical programming. This method requires that all function in mathematical programming should be linear. This method is useful to find the best available value such as highest profit, lowest cost etc. Linear programming is a special case of mathematical programming (mathematical optimization). This method assures the best possible use of resources and to highlight the bottleneck problems. The problem with this method is that it can be only applied to linear functions while in real life constraints and objectives are not linear.

A mixed integer linear formulation for the open shop earliness-tardiness scheduling problem was proposed by Hashemi [30]. Problem of minimizing total tardiness penalty in an open shop scheduling environment with non identical parallel machines was discussed in this paper. This problem was formulated as a mixed integer linear programming model using time based decision variables. Computational results proved that the proposed mathematical models were efficient in solving open shop problems with 5 stages and 3 jobs..

Exact algorithms are not capable of finding optimal solution for NP-hard problems because they have a slow convergence rate and they can solve only small size problems. But real time environment problems are of large size so adopting exact methods for large size problems seems inadequate which requires lot of time and effort. Meta heuristics are trapped by these local optima or poorly guided because of the generally small solution gaps. The tiny gaps also affect the branch and bound methods. Therefore heuristic and metaheuristics have been developed described below.

2.2: Approximate Algorithms

The high computational time and the complexity involved with the exact procedures led several researchers' attention to the development of approximate methods for solving large sized problems. Approximate procedures are adapted to large size problems to obtain near optimal solution within reasonable computational time. The approximate procedures include the local search methods, meta-heuristics, priority dispatching rules, artificial intelligence and bottleneck based heuristics.

2.2.1: Heuristics

Steady evolution has been done in the last few years in the development of heuristic providing optimal solutions in reasonable time. Systemize decision making process done by hand was the first proposed heuristics. We can test huge amount of combinations in a short amount of time with the help of computer. Solution generated which turned to be of much better quality when compared to what an expert in the field could produce by hand. Obtaining a feasible solution was the main aim in early heuristics and possibly applying to it a post optimization progress. Most combinatorial optimization problems are NP-hard. It means that partial enumeration based exact algorithms have a slow convergence rate, and only small instances can be solved by using these algorithms.[21]. Due to complexity and NP-hard nature of the open shop scheduling problem some heuristics and metaheuristics were proposed in the literature.

Fang et al. [14] have developed a promising hybrid GA/heuristic approach for open-shop scheduling problems. The simplicity of the approach, its apparent success, and the evident potential for much further improvement and extension seem to render it a promising method warranting further research. A hybridized Genetic Algorithm/heuristics method thus tends to seem the better practical choice, offering a better tradeoff in term of speed vs. quality on most of the problems.

The weakness of this technique is that there is no guarantee that the approach will generalize successfully to real problems and as well on different and large bench marks. It is clearly a promising enough basis for continued research along these lines. Other weakness includes extra time cost. Ultimately of course comparisons with other Artificial Intelligence techniques will be instructive. Also the approach as presented fails to meet some possible needs which schedule managers may have in machine shop environment. Genetic Algorithm configuration used in this technique is not optimal.

Liaw [8] have proposed an iterative improvement approach for the non preemptive open shop scheduling problem. An iterative improvement approach based on Benders decomposition to a mixed integer programming formulation was presented.

Robust heuristic open shop scheduling problem was presented in this research. Initial feasible solutions were generated by heuristic dispatching rule. The code which was used for developing a proposed method was experimental in nature computational speed can be increased by production version of this algorithm.

Often it is the possibility that using heuristics feasible solutions cannot be obtained in reasonably amount of time. Stopping criteria is based on sequence improvement procedure. When the sequence improvement procedure can no longer find a better sequence then this procedure stops.

So this whole procedure is heuristic. Solving a longest path problem is the most time consuming part of the proposed methodology. Thus solving the longest path problem efficiently is the most challenging task in this procedure and may need attention from the researchers. Longest remaining processing jobs (LRPT) are preferred for selection so shortest processing jobs will face starvation. Researchers were not able to solve the instances optimally. Best solution found by the algorithm is compared with the well known algorithm in all problem instances. It can also be concluded from the results that algorithm does not performed extremely well for problems for which number of jobs are not equal to the number of machines. By using minor changes this proposed algorithm can also be used to solve non preemptive open shop scheduling problems with other performance measures such as tardiness, flow time.

Gueret and Prins [10] have presented classical and new heuristics for the open shop scheduling. Two new heuristics were presented in this paper. Results were very adorable randomly generated instances were solved up to 90 % while classical methods can solve these problems up to only 20%. It makes open shop instances to solve easily so generating hard instances for open shop was the problem.

Algorithm H1 requires a lot of computation. The worst case complexity was O (N2.M2). The algorithm H1 was stopped after the 50 main steps because no further improvement was seen. New heuristics presented are not as efficient as for the randomly generated problems, but they still work much better than first come first serve, shortest processing time remaining and longest processing time remaining. The average make span is still quite good: less than 4% from the optimum.H1 algorithm is less efficient than H2 algorithm.

Brasel et al. [14] have suggested constructive heuristics algorithms for the open shop scheduling problem to minimize mean flow time. Heuristic algorithms were developed by researchers for strongly NP-hard problems. Among constructive algorithms they considered matching algorithms, priority dispatching rules, insertion and appending procedures combined with beam search.

Up to Problems with 50 jobs with 50 machines were presented for computational results respectively. Lower bound was used to check the quality of solutions. The quality of the solutions was evaluated by a lower bound for the corresponding preemptive open shop problem and by an alternative estimate of mean flow time. Among the matching algorithms, the best results were found for all square problems with n = m.

There are many papers available in literature on multi criteria scheduling problems where the objective was the minimization of mean flow time. Although this measure is more important therefore this research was carried out.

Senthikumar and Shahabudeen [19] consider GA based heuristics for the open job shop scheduling problem. To minimize the make span heuristics for open shop scheduling problem were generated by using genetic algorithm. To maintain feasibility genetic algorithms operators were modified. Separate routine is developed to compute make span.

Very little work is reported in literature on the heuristics for open shop by using heuristics. The problem with this approach is that it produces better results only for large size. Although it can be inferred that GA based heuristics for the open shop scheduling yields minimum make span when compared to the existing heuristics for the moderated job shop scheduling problem.

Low and yeh [16] have presented a genetic algorithm based heuristics with some restriction such as independent and dependent removal time. The objective of this study was to minimize total job tardiness. First the problem was described in a 0-1 format and then solved. Optimum solution was not found in the 50 running hour by using this technique therefore genetic based heuristics have been proposed in this research. In order to check the performance of algorithm some comparison were made and reported as well. These heuristics can perform well for larger instances in which job size was greater than 10.

Among these heuristics genetic algorithm performs well than simulated annealing and tabu search. Because genetic algorithm provides more opportunities to solve population to population search while simulated annealing and tabu search provides point to point search. Although these algorithms require more computations but are still acceptable because they provides good results when compared with classical metaheuristics.

The performance of the double genetic algorithm (DGA) was slightly better than the simulated annealing Genetic algorithm (SAGA) and the Tabu search Genetic Algorithm (TSGA) heuristics and more robust as well. Computation time for all three hybrid genetic heuristics was less than 0.4s.

Classical heuristics are faster than the hybrid genetic based heuristics from practical point of view. Tabu search algorithm was faster than the two other genetic based heuristics presented in the research which require more computation. By comparing all the algorithms it could be concluded that Tabu search Genetic algorithm is the fastest, double genetic algorithm was the faster, simulated genetic algorithm was the worst . Finally it is concluded that in spite of the good performance of hybrid genetic based heuristics they still require more computations; so adjustable parameter setting such as cross over rate, mutation rate, population size etc can be introduced to achieve the goal of reduced computation time.

2.2.2: **Metaheuristics**

Meta heuristics is an iterative generation process which guides a subordinate heuristics by combining the concept of exploitation and exploring. They are usually non deterministic. Metaheuristics is the combination of heuristic and randomization. It actually did not know what the problem is it solving therefore it can be applied to different problems.

While several methods are being addressed in literature, Tabu Search, Simulated Annealing (Laarhoven and Aarts, 1987) and Genetic algorithms (Davis, 1985) are the most successful meta-heuristic methods. The success of these methods is defined by their capability in producing good solutions (near optimal) in less computational time.

Meta heuristics mainly depends on two principles local optimality and global optimality on which there have been a lot of work in last 15 years. In local search technique exploration of search space is achieved by moving in the neighborhood. Simulated annealing [17, 1], tabu search [16] and variable neighborhood search are the most famous local search methods. Population search consists of maintaining a pool of good solutions and combining them in order to produce hopefully better solutions. Classical examples are genetic algorithms [17, 1, 2] and adaptive memory procedures [21].

It is not possible to generalize the metaheuristics method for general use. It totally depends on the nature of the problem. Although Hertz and Widmer [11] tried to give some rules that could help in designing of successful alteration.

Liaw [24] have developed a algorithm which hybridized tabu search as a local operator and genetic algorithm as a global operator. The HGA significantly outperforms other methods in term of solution quality. Hybrid genetic algorithm performs extremely well in computational results. Optimum solution was found for all instances. Those instances which were not solved by using this algorithm solution gap were extremely small. There are some instances as well which were solved to optimality for the first time in literature.

The quality of this technique was that local improvement method was applied to each newly generated off spring so that they will not stuck into local optimum before moving into the next population. In this manner, Genetic algorithm was used to perform global search while local search procedures are used to perform local exploitation. One of the major advantages of using this technique is that active schedule is used in the procedure which means that no operation can delay another operation. It will give us minimized make span. HGA competes previous algorithms not in solution quality but in time as well.

For computational reasons tabu search was run only for small numbers of time each time a local improvement setup is executed. If the numbers of iterations for local improvement procedure are increased the results could be better. The algorithm could not perform well for 3 instances taken from benchmark.

Andresen et al. [1] conducted a study on simulated annealing and genetic algorithm for minimizing mean flow time in an open shop. They used two iterative algorithms simulated annealing and genetic algorithm for this strongly NP-hard problem known as open shop.

Algorithm was restricted to 30,000 evaluation solutions settled in initial state. Up to 50 jobs and 50 machines were evaluated by this method. The algorithm performs better when fixing the number of solutions to 30000.

There is no paper available in literature on multi criteria scheduling problems with the objective of minimization of mean flow time. Although this measure is more important therefore this research was carried out.

So this paper contributes a lot in the literature. The most exhaustive thing is that researchers considers all pairs (n, m), n=m, n<m, n>m with GA-R, GA-C, SA(S), SA(API). Genetic algorithms performs well than simulated algorithm in most of the cases in which n was less than or equal to m. while for many instances with n less than m the preemptive lower bound is reached.

Tamilarasi and Anantha [17] have suggested an enhanced genetic algorithm with simulated annealing for job shop scheduling in which genetic algorithm is hybridized with simulated annealing. Minimizing the makespan was the objective of the study. 21 instances taken from the OR-library were used as a bench mark to test the proposed algorithm. Algorithm gives the optimal solution in considerable amount of time.

This algorithm provides a very good combination of local and global search hybridization for job shop scheduling which motivate us to perform this study for open shop scheduling. It can also escape from local minimum. Problem representation was easy and simple. Also this algorithm reduces the computational complexity. Although other hybridization heuristics methods can be used to solve the job shop scheduling problem and produces better results.

Ahmadziar and Farahani [2] hybridized genetic algorithm with local optimization heuristics for open shop scheduling. Minimizing the make span time was the purpose of the study. This algorithm works efficiently as compared to other algorithms available in the literature.

It can be seen in the results section that although the algorithm outperforms in term of efficiency but could not compete the compared algorithms in term of time. The other major weakness of this algorithm is that it favors longest remaining processing (LRPT) time jobs therefore shortest process remaining time (SRPT) jobs will face starvation.

Panahi and Tavakkoli [18] hybridize ant colony and simulated annealing for multi objectives called makespan minimization and total job tardiness. There was no previous study available in literature in order to validate the results therefore comparisons were made with a multi objective genetic algorithm called NSGA II. Comparisons were also made for single objective case as well.

Lot of literature is available for multi objective and multi criteria for job shop and flow shop scheduling problem. No work has been done in past which can dealt open shop scheduling with two or more objectives. Hence the researchers were motivated to perform this study

As it can be noticed from this technique the number of Pareto solutions obtained by hybrid ant colony optimization (HACO) is more than NAGSII.

There is no such a measure for measuring the weakness of this algorithm as there is no previous study related to open shop scheduling with multi objectives. This is a new area of search which needs a lot of attention from researchers.

Apart from all these papers Liaw [15] proposed a hybrid genetic algorithm for open shop scheduling. Prins [25] proposed a competitive genetic algorithm for open shop scheduling. Blum [26] presented a Beam ACO in which hybridization of ant colony was done with beam search for open shop scheduling. Hsu and Shah [27] proposed a new particle swarm optimization for open shop scheduling.

Optimization problems have been solved by metaheuristics techniques successfully. There are some cases in which common search algorithms could not perform well especially in the case when the searching area is large. In such circumstances we have to develop special metaheuristics depending upon the nature of the problem or hybridization with problem specific search technique which improves the results [2]. By hybridization two goals of local optimal solution and global optimal solution can be achieved.

Mostly optimal solutions are dispersed in a large solution space. Therefore Meta heuristics are stuck by local optimal problem because of poorly guided techniques. Branch and bound methods are also affected by this problem. So local optimum solution is found very quickly but these Meta heuristics take very long time to find global optimum because they fall into local optimum. Most of the above discussed method use disjunctive graph model which becomes less powerful when precedence constraints are applied.

Therefore, to efficiently solve the open shop scheduling problem with the objective of minimizing the make span Genetic algorithm is hybrid with simulated annealing for solving open shop scheduling problem.

CHAPTER 3

RESEARCH DESIGN

3-RESEARCH DESIGN

This chapter briefly describes the research methods and the proposed approach. Ant colony algorithm is hybrid with simulated annealing to solve open shop scheduling problem in the literature [18]. However Ant colony algorithm has slower convergence rate than Genetic algorithm and Genetic algorithm is the most popular one in evolutionary algorithm and Genetic algorithm has the higher probability of reaching global optimum in a defined time interval. On the other side simulated annealing and genetic algorithm were hybridized for minimizing mean flow time in an open shop environment and produces significantly better results[1]. Moreover there is no work considering on hybridizing Genetic algorithm and Simulated annealing for minimizing make span criteria. While it produces optimal solution for job shop scheduling. Therefore hybrid Genetic Algorithm with simulated annealing is proposed to solve open shop scheduling for minimizing makespan.

The main objective of proposed was to minimize the makespan (total time taken by all operations).

3.1: Proposed Hybrid Genetic Algorithm

There are three well known metaheuristics techniques [16] simulated annealing, tabu search and evolutionary algorithm. It is indicated from the past research that in simulated annealing optimal solution can be obtained in all situations. Tabu search totally works on heuristics techniques. They often stuck into local optima problem although they have faster convergence ratios as compared to other randomization optimization techniques. Evolutionary algorithms are those algorithms which include living systems phenomena for example Genetic algorithm, evolutionary strategy, evolutionary programming and ant colony optimization.

Genetic algorithm is the most popular one among these algorithms. Among the entire evolutionary algorithms Genetic algorithm is most suitable for combinatorial optimization problem because it contains more simulation to living system than other evolutionary algorithm. Genetic algorithm starts its search concurrently from many initial points while simulated annealing and tabu search start from only one initial point and search consecutively. Therefore, Genetic algorithm has the highest ranking in reaching global optimum in defined time when compared with other optimization techniques mentioned above solution effectiveness and efficiency can also be achieved [16].

Solving optimization problem by using genetic algorithm is a versatile and effective approach. However in many cases simple genetic algorithm could not perform effectively therefore different hybridization techniques have been proposed. In this research genetic algorithm is hybridized with simulated annealing to form hybrid algorithm. With the proposed hybrid genetic-based algorithm, global exploration among a population was performed using a genetic algorithm, while the local optimization heuristics simulated annealing has been used in the population to move it to a local optimum before moving it to the next generation. The general pseudo code of the hybrid genetic algorithm is described below.

Step 1. Initialization (Generate a random solution as the initial population)

Step 2. Evaluation (for each individual calculate fitness value)

Step 3 Crossover (Apply order 1 cross over)

Step 4 Mutation (Apply swap mutation)

Step 5 Local optimization method (apply simulated annealing to the population)

Step 6 Survivor selections (use roulette wheel selection method to generate a new population with higher fitness value)

Step 7 Termination (Repeat steps from 2 to 6 until the termination criterion is met)

3.1.1: Representation

Suitable representation of chromosomes is the main key point of genetic algorithm to solve real problems. Before applying genetic algorithm careful study is necessary to assure suitable representation. There are two main issues which we have to face in open shop scheduling problems:

1. Formation of the routing of each job

2. Finding the sequence for each job according to machine

These two things in chromosomes are represented by using permutation representation. In this representation each gene stands for operation. Operations of a job are numbered in a list in which they have to schedule. A permutation which starts from 1 to total number of operations represents a chromosome.

Consider a simple example with 3 jobs and 2 machines. Each job consists of 2 operations. Thus a possible representation could be

Jobs	1		2		3	
Operation	10	6	2	3	4	2
Machine	1	2	1	2	1	2

Table 3.1: **Representation of job machine combination**

Each job must visit every machine once. Orders of Operations can be processed in any order. A feasible schedule is given below

Machine no 1. Job1→Job2→job3

Machine no 2. Job1→Job2→Job3

3.1.2: Selection method

Selection method replicates the solution having higher fitness value according to their rate proportional to their quality. There are different selection methods available depending on the nature of problem. A roulette wheel approach is found to be more adequate as the selection procedure for the proposed algorithm. It is a fitness proportional selection method in which an individual having lower fitness value has the chance to drop out from population. It is also possible that an individual having higher fitness value will be choose multiple times The roulette wheel selection operator used in the proposed approach is as follows

Step1. For each individual calculate fitness (makespan)

Step2. Choose individual proportional to their fitness value

3.1.3: **Crossover method**

Cross over method is the critical and important phase of genetic algorithm. It decomposes two distinct solutions and then randomly mixes their part to form a new solution which contains the properties of their parents. Good genes are passed to the next generation by using this method. So designing a cross over operator that can store genes having higher probability is the important aspect. In open shop scheduling it is not possible to generate a new chromosome which can increase the fitness value so partially mapped crossover (PMX) is used to preserve the position and contents of the genes. . In open shop scheduling problem, the performance quality is dependent on order of jobs instead of permutation of operations so order 1 crossover is used because idea is to preserve the relative order of jobs. The crossover operator works as follows

1. Firstly Choose an random part from the first parent
2. Copy this arbitrary part to the first child
3. Remaining numbers will be copied from second parent
4. The remaining part will be copied into the second child
5. Copy the number from the first parent that are not in the second child

The following example will illustrate the procedure. Suppose that the following two individual have been selected for crossover

Parent1. {5, 3, **2, 1**, 4, 6}

Parent2. {1, 2, **3, 4, 5, 6**}

The offspring will be changed as follows

Offspring1. {3, 4, 2, 1, 5, 6}

Offspring2. {2, 1, 5, 3, 4, 6}

The procedure will have been repeated for all other machines.

3.1.4: Mutation method

Mutation operator is used to randomly perturb the contents of solution. It causes local or global movement of the search space. It is also used to keep the diversity of population. Swap mutation is used in this proposed algorithm to keep the contents saved. In swap mutation two locations are selected randomly and then their contents are changed according to the random positions selected

Pseudo code of swap mutation is given below

Step1. Select a random position *(m1)* in the chromosome *A*

Step2. Select another random position *(m2)* in the chromosome *A*

Step3. Swap the contents of the position *m1* and *m2* of the chromosome *A*

Following example will elaborates the process

A: 5 3 2 1 4 6

Suppose m1=3 and m2=4 after mutation

A: 5 3 1 2 4 6

3.2: Local optimization heuristic (SA)

3.2.1: Temperature

Optimum solution in simulated annealing depends on initializing the high and low effective temperature. In the proposed algorithm high and low temperature for a given problem is dependent on the acceptance probability depending on the condition.

High temperature is denoted by t_h and low temperature is denoted by t_l.

Step1. $T=t_h$

Step2. When the temperature is higher then t_l

Step3. Search for the new solution x' (perturb)

Step4. Metropolis criteria (Defined below)

Step5. Decrease the temperature

3.2.2: **Metropolis criteria**

The metropolis process which is used in this proposed algorithm described below

Step1. Probability $p_{accept} = [X-X'] / T$

Step2. Let N= Random (0, 1)

Unconditionally accepted if

Step3. $X'<X$ (the new solution is better)

Probably accepted if

Step4. $X'>=X$ the new solution is worse

Accept only if $N< p_{accept}$

CHAPTER 4

CASE STUDIE

4. CASE STUDIES

This chapter presents the details of the experimentations conducted on the Hybrid genetic algorithm with simulated annealing. Experimentation was conducted on two case studies. Later on some instances generated from Library of scheduling algorithm were evaluated to identify the efficiency of proposed algorithm. In the first case study, 6 jobs with 2 machines (6x2) were evaluated. In the second case study exam scheduling was done for BSCS students of international Islamic university (IIU). Both case studies are briefly described below.

4.1: Case study 1

In the first case study 6x2 problem was evaluated. Table 2 shows a 6x2 problem in detail. Operation time and machine ordering is given in following tables.

Jobs	**Operation**	
J1	O1	O2
J2	O3	O4
J3	O5	O6
J4	O7	O8
J5	O9	O10
J6	O11	O12

Table 4.1: **6x2 open shop scheduling problem**

Table 3 shows operation times of different jobs. There are two operations in each job and there are 2 machines available as resources. The purpose is to organize these jobs in such a way that the makespan criteria should be minimized.

Jobs	Operation	
J1 (O1,O2)	10	6
J2 (O3,O4)	7	9
J3 (O5,O6)	3	8
J4 (O7,O8)	10	2
J5 (O9,O10)	12	7
J6 (O11,O12)	6	6

Table 4.2: **Operation times**

Table 4 shows a machine allocation.

Machine Allocation	
M1	**M2**
O1	O2
O4	O3
O6	O5
O8	O7
O9	O10
O12	O11

Table 4.3: **Machine Allocation**

The feasible schedule for the above open shop scheduling problem is 1,2,4,3,6,5,8,7,9,10,12,11 gives the makespan of 45 which is feasible schedule for given open shop scheduling problem. The Gantt chart for the following feasible schedule is drawn below. In the proposed methodology schedules have been takes as chromosomes.

	5	10	15	20	25	30	35	40	45
M1									
M2									

Table 4.4: **Allocation of jobs**

Jobs	J1	J2	J3	J4	J5	J6
Color						

Table 4.5: **Colors Assigned to jobs**

This case study was totally dependent on the scenario. As there were only two operation for each job and 2 machines were available. So a mixed type of favoring short and long job scenario was used. If in the first session long job was favored then in the next interval priority will be given to short job. In this way the whole case study was implemented and the results were outstanding. This type of algorithm can also be applied in case studies in which number of jobs is equal to the number of machines and there will be no starvation.

4.2: Case study 2

In the 2nd case study exam scheduling was done for BSCS students of International Islamic University Islamabad which is an application of open shop scheduling. Sessions of BSCS F10, F11, F12, and F13 were taken as input. The objective of hybrid genetic algorithm is to find a schedule the exams by minimizing the number of days which is called makespan criteria. It is assumed that each class is attending four courses because in open shop environment numbers of operations in all jobs are equal. Each course has a unique ID which is called the course code. Number of room denotes the machines. Each session of BSCS denotes a job which consists of courses (operation). The main purpose is to schedule exams by minimizing makespan. The detail is of courses is given below

Class	Course code	Number of students
BSCS_ F10	CS_451	48
BSCS_ F10	CS_341	100
BSCS_ F10	CS_371	51
BSCS_ F10	SE_431	86
BSCS_ F11	GC_101	50
BSCS_ F11	CS_314	39
BSCS_ F11	MATH_121	39
BSCS_ F11	SE_341	57
BSCS_ F12	GC_151	50
BSCS_ F12	CS_701	30
BSCS_ F12	CS_223	31
BSCS_ F12	CS_331	50
BSCS_ F13	CS_111	118
BSCS_ F13	GC_102	275
BSCS_ F13	GC_191	80
BSCS_ F13	110	105

Table 4.6: **BSCS course detail**

It is assumed that there are only three rooms available for BSCS. Each course is assigned a specific room according to its capacity. It is supposed that each room can accommodate 100 students. The Objective or fitness function takes the input as the number of students, number of Operations, session and course code. The fitness function assigns weights to different jobs according to their make span values. Jobs with higher make span gets less weight and jobs with less makespan got more weight. The output will be a feasible schedule of exams for students in which there is no conflict and minimum completion time.

CHAPTER 5

EXPERIMENTATION &CONCLUSION

6. EXPERIMENTATION

This chapter deals with the experimentation conducted on Hybrid genetic algorithm with simulated annealing for open shop scheduling. LISA (Library of scheduling algorithm) was used to generate the instances. Results were compared with the 3 well known algorithms found in literature described below. Hybrid genetic algorithm produces significantly better results as compared to previous results. Details of experimentation are defined below.

6.1: Experimental Results and Discussion

In order to conduct the experiment proposed algorithm was implemented in Matlab 7 and run on a pc with 2.0 GHZ and 2.0GB ram. Large instances generated from LISAwere used to evaluate the proposed algorithm .The best solution were selected from each problem. The performance of algorithm is compared with other famous algorithm present in literature.

Designed hybrid genetic algorithm with simulated annealing was tested using LISA. Instances were generated by using special software called Library of scheduling algorithm (LISA). It is a software package for solving deterministic scheduling problems and for generating instances for different shop scheduling problem. Following problem sizes have been taken into consideration

n(Jobs)	1	2	3	4	5	6	7
M(machines)	2-9	2-9	2-9	2-9	2-9	2-9	2-9

Table 5.1: **Problem size**

n represents jobs in table 5.1 and m represent machines. As for example (3,3) mean 3 jobs and 3 machines. 7 test problems are executed for each scenario. Machine environment was selected as open shop in LISA. Objective function was selected as cmax (makespan minimization) from menu. Number of machines and jobs were given in the text box to generate test type.

Processing time for each job was created using the time seed and machine seed given in LISA as seen in figure 5.2. Well known algorithms available in literature were used to compare the results. For genetic algorithm crossover probability was taken as 0.35 and mutation probability was taken as 0.25. While in simulated annealing high temperature depends on the situation. The proposed gives us the best optimal values in most cases. The maximum numbers of iterations were initially selected as 50.

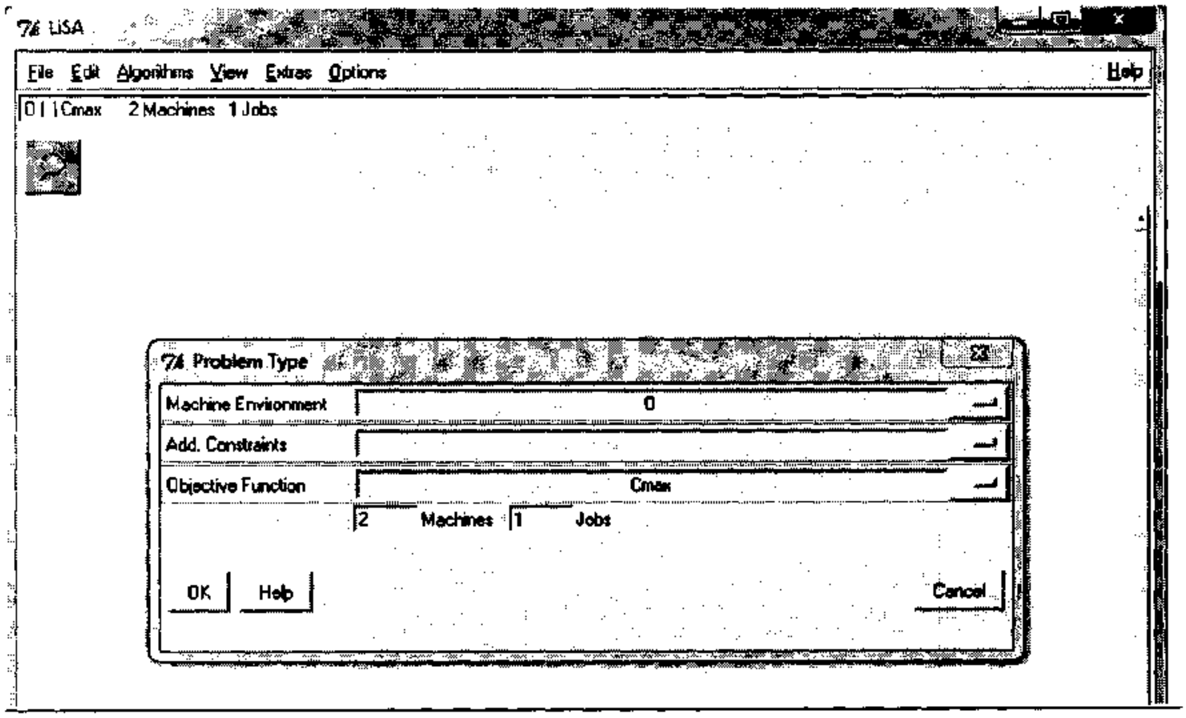

Figure 5.1: **Snap short of LISA**

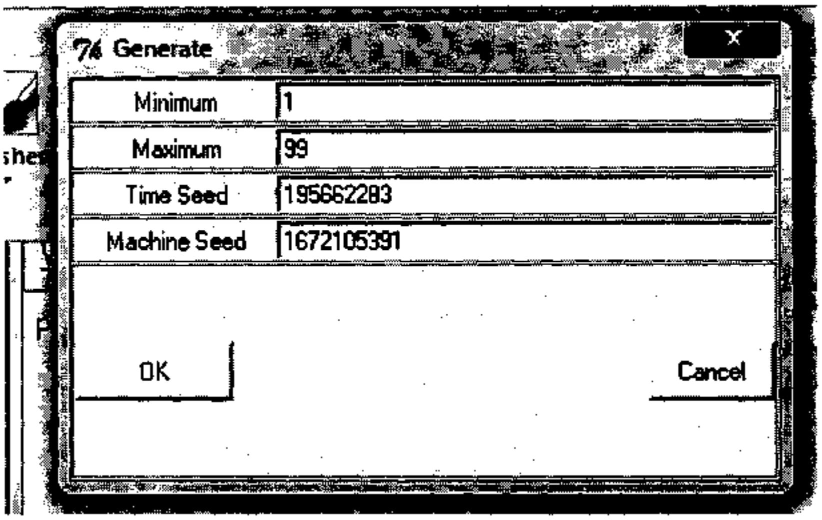

Figure 5.2: **Snap shot for generating processing time**

Results of hybrid genetic algorithm were compared with the well known algorithms beam ant colony [26], iterative improvement approach [8] and genetic algorithm [13].

Two files were used for implementing hybrid genetic algorithm with simulated annealing. Test_data contain the information about jobs and number of machines and SA_GA contain the algorithm. First number of jobs is selected from the following menu.

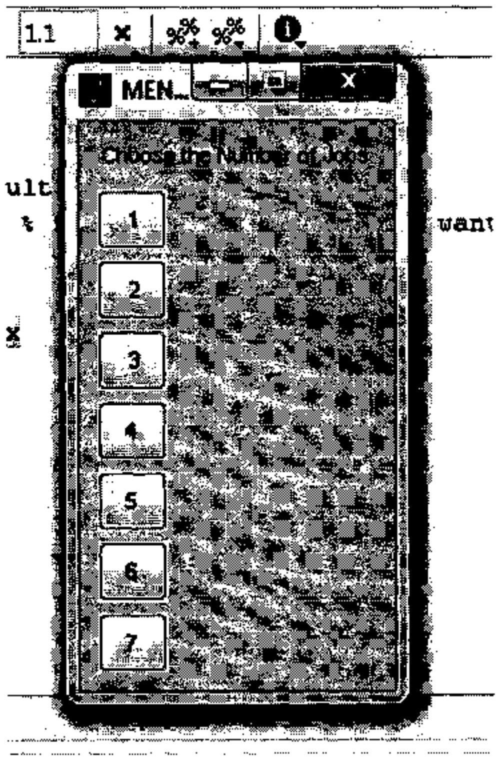

Figure 5.3: **Snap shot for selecting number of jobs**

After selecting number of jobs number of machines will be selected from the following menu

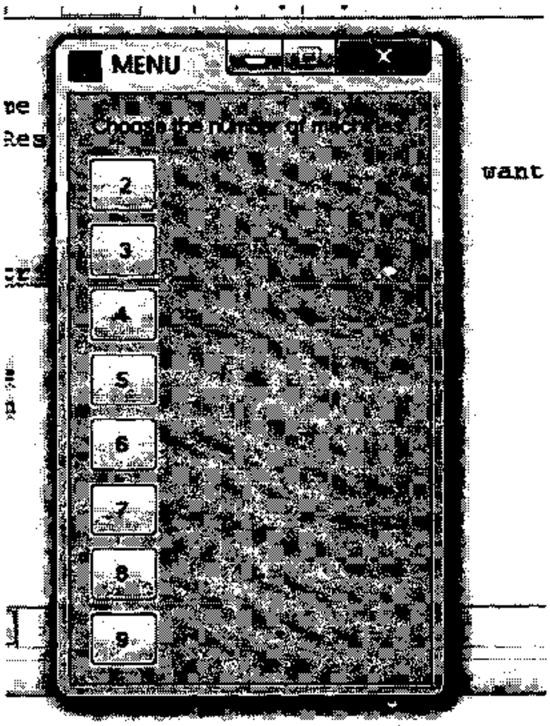

Figure 5.4: **Snap shot for selecting number of machines**

Selection of number of jobs and number of machines will lead us to menu which will help us in selecting the crossover probability and mutation rate. Number of generations and population size will also be selected from this menu.

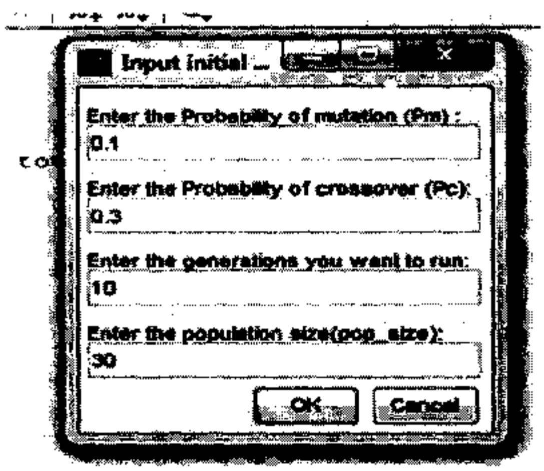

Figure 5.5: **Snap shot for initial parameters**

After pressing the ok button minimum makespan will be calculated and given to the user as given in the example

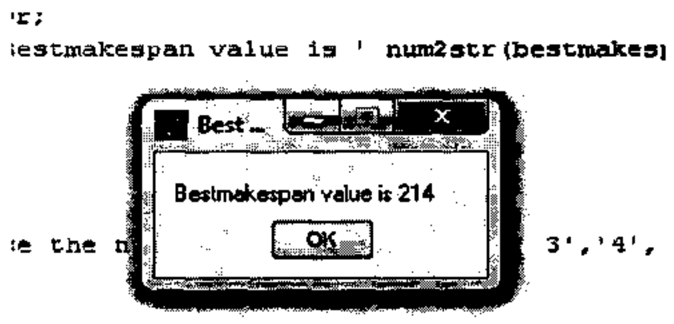

Figure 5.6: **Best make span value**

The results of the problem with n=m are given in the following table 7

(n,m)	HGA	GA	Beam ant colony	Iterative approach
1x2	32	32	32	32
1x3	172	172	172	172
1x4	214	214	214	214
1x5	202	202	202	202
1x6	245	245	245	245
1x7	180	180	180	180
1x8	379	379	379	379
1x9	460	460	460	460
2x2	86	86	86	86
2x3	111	111	111	111

2x4	214	214	214	214
2x5	245	245	245	245
2x6	294	294	294	294
2x7	439	439	439	439
2x8	430	430	430	430
2x9	415	415	415	415
3x2	128	128	128	128
3x3	**244**	**253**	**253**	**253**
3x4	**193**	**195**	**195**	**195**
3x5	**276**	**275**	**275**	**275**
3x6	341	341	341	341
3x7	477	477	477	477
3x8	392	392	392	392
3x9	589	589	589	589
4x2	**199**	**203**	**203**	**203**
4x3	**241**	**253**	**253**	**253**
4x4	**304**	**307**	**307**	**307**
4x5	290	290	290	290
4x6	313	313	313	313
4x7	457	457	457	457
4x8	553	553	553	553

4x9	629	629	629	629
5x2	**326**	**330**	**330**	**330**
5x3	**327**	**330**	**330**	**337**
5x4	**294**	**296**	**296**	**296**
5x5	**367**	**386**	**386**	**386**
5x6	417	417	417	417
5x7	414	414	414	414
5x8	422	422	422	422
5x9	513	513	513	513
6x2	**369**	**370**	**370**	**370**
6x3	**250**	**253**	**253**	**254**
6x4	**369**	**376**	**376**	**376**
6x5	**420**	**426**	**426**	**426**
6x6	**465**	**470**	**470**	**470**
6x7	482	482	482	482
6x8	488	488	488	488
6x9	668	668	668	668
7x2	**392**	**395**	**395**	**395**
7x3	**437**	**440**	**440**	**440**
7x4	**423**	**425**	**425**	**425**
7x5	**483**	**487**	**487**	**489**

7x6	**464**	**465**	**465**	**465**
7x7	**460**	**465**	**465**	**465**
7x8	504	504	504	504
7x9	571	571	571	571

Table 5.2: **Instances**

The above table list size of problem (number of jobs x number of machines) the best known makespan and the best known makespan obtained by each of the compared algorithm.

Best makespan value obtained by all algorithms can be viewed in table 2. We obtained the optimal solution for many problems which were showed in table by highlighted text. The algorithm was not tested on large instances because complexity of algorithm increases with the increase in jobs and machines. It can also be noted that algorithm outperforms on small instances but when the number of jobs and machines increased the algorithm efficiency decreased.

The results obtained by simple Hybrid genetic algorithm with simulated annealing are better than those results of algorithm that did not use local search operator. So this shows that the proposed algorithm local search procedure competes the previous results and quality of solution is improved due to this operator.

6.2: Conclusion

In this research genetic algorithm is hybridized with simulated annealing to solve the open shop scheduling problem for minimizing makespan criteria. By Combining the Genetic algorithm and simulated annealing field of search can be narrowed. Exploitation can also be achieved by using these algorithms.

In the proposed problem the representation of individual is easy and fitness function is very simple when compared with other heuristics method. Firstly two case studies were used to evaluate the algorithm. Instances were generated by special software called LISA.

In order to evaluate the performance of HGA several instances were generated by using LISA. It can be noted by viewing the results that HGA outperforms in all cases except 7 jobs and 7 machines. This could be due to the less number of iterations. If number of iterations will increase then algorithm may outperforms in all cases but time complexity increases with increasing in number of iterations.

Even though the good performance of proposed algorithm certain mechanism can be introduced into genetic algorithm such as parameter setting in the searching process i.e, mutation rate, crossover rate and population size for reduced computation or best optimal solution

References:

[1]. Michael. A, Heidemarie. B, Marc. M, Jan. T, Frank. W and Per. W, *"Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop"*, Mathematical and Computer Modeling, 48(7), 1279-1293, 2008.

[2]. Fardin. A, Mehdi. H. F, *"A novel hybrid genetic algorithm for the open shop scheduling problem"*, The International Journal of Advanced Manufacturing Technology, 62(5-8), 775-787, 2012.

[3]. Khaled. M, Slim. H and Pierre. B, *"Evolutionary algorithms for job-shop scheduling"* International Journal of Applied Mathematics and Computer Science, 14(1), 91-104, 2004.

[4]. Peter. B, Johann. H, Bernd. J and Birgit. W *"A branch & bound algorithm for the open-shop problem"*, Discrete Applied Mathematics, 76(1), 43-59, 1997.

[5]. Brucker, P., Jurisch, B., & Sievers, B. (1994). A branch and bound algorithm for the job-shop scheduling problem. Discrete Applied Mathematics, 49(1), 107-127.

[6]. Christelle. G and Christian. P, *"A new lower bound for the open-shop problem"*, Annals of Operations Research, 92, 165-183, 1999.

[7]. Taillard, E. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278-285.

[8]. Ching. F. L, *"An iterative improvement approach for the nonpreemptive open shop scheduling problem"*, European Journal of Operational Research, 111(3), 509-517, 1998.

[9]. Teofilo. G, Sartaj. S, (1976). Open shop scheduling to minimize finish time. Journal of the ACM (JACM), 23(4), 665-679.

[10]. Christelle. G and Christian. P, *"Classical and new heuristics for the open-shop problem"*, A computational evaluation, European Journal of Operational Research, 107(2), 306-314, 1998.

[11]. Alain. H and Marino. W, *"Guidelines for the use of meta-heuristics in combinatorial optimization"*, European Journal of Operational Research, 151(2), 247-252, 2003.

[12]. Christelle. G, Jussien. N, Christian. P, *"Using intelligent backtracking to improve branch-and-bound methods"* An application to open-shop problems, European Journal of Operational Research, 127(2), 344-354, 2000.

[13]. Hsiao. F, Ross. P and Corne. D, *"A promising hybrid GA/heuristic approach for open-shop scheduling problems"* In Proc. 11 the European Conference on Artificial Intelligence (pp. 590-594). 1994.

[14]. Heidemarie. B, Andre. H, Marc. M, Thomas. T, Jan. T, and Frank. W, *"Heuristic constructive algorithms for open shop scheduling to minimize mean flow time"*, European Journal of Operational Research, 189(3), 856-870, 2006.

[15]. Ching. F. L, *"A hybrid genetic algorithm for the open shop scheduling problem"*, European Journal of Operational Research, 124(1), 28-42, 2000.

[16]. Chinyao. L and Yuling. Y, *"Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated"*, Robotics and Computer-Integrated Manufacturing, 25(2), 314-322, 2009.

[17]. Tamilarasi. A and Anantha. T, *"An enhanced genetic algorithm with simulated annealing for job-shop scheduling"*, International Journal of Engineering, Science and Technology, 2(1), 144-151, 2010.

[18]. Hadi. P, Reza. T, *"Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization"*, Expert Systems with Applications, 38(3), 2817-2822, 2011.

[19]. P. Senthikumar and P. Shahabudeen. (2006). *"GA based heuristic for the open job shop scheduling problem"* The international journal of advanced manufacturing technology, 30(3-4), 297-301 2006.

[20]. Alain. H, Marino. W, "Guidelines for the use of meta-heuristics in combinatorial optimization", European Journal of Operational Research, 151(2), 247-252, 2003.

[21]. Dorndorf. U, Pesch. E, Phan-Huy.T, "Solving the open shop scheduling problem, Journal of Scheduling" 4 (2001) 157–174.

[22]. Laborie. P, "Complete MCS-based search, application to resource constrained project scheduling, in", Proceedings of International Joint Conference on Artificial Intelligence, vol. 19, 2005, pp. 181–186.

[23]. Tamura. N, Taga. A, Kitagawa. S, Banbara. M, "Compiling finite linear CSP into SAT, in: Proceeding of the 12th International Conference on Principles and Practice of Constraint Programming, CP'06", in: Lecture Notes in Computer Science, vol. 4204, Springer, 2006, pp. 590–603.

[24]. Liaw, Ching-Fang. "A hybrid genetic algorithm for the open shop scheduling problem." *European Journal of Operational Research* 124.1 (2000): 28-42.

[25]. Prins, Christian. "Competitive genetic algorithms for the open-shop scheduling problem." *Mathematical methods of operations research* 52.3 (2000): 389-411.

[26]. Blum, Christian. "Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling." *Computers & Operations Research* 32.6 (2005): 1565-1591.

[27]. Sha, D. Y., and Cheng-Yu Hsu. "A new particle swarm optimization for the open shop scheduling problem." *Computers & Operations Research* 35.10 (2008): 3243-3261.

[28]. Sha, Lui, et al. "Real time scheduling theory: A historical perspective." Real-time systems 28.2-3 (2004): 101-155.

[29]. Innani, Alok. Applying Data Mining to Job-Shop Scheduling using Regression Analysis. Diss. Ohio University, 2004.

[30]. Monette, Jean-Noël. "Solving scheduling problems from high-level models." *4OR* 9.3 (2011): 317-320.

[31]. Liu, C. Y., and R. L. Bulfin. "Scheduling ordered open shops." *Computers & operations research* 14.3 (1987): 257-264.