Thesis for the Degree of Master of Engineering

A Study on Job Shop Scheduling Problem Using a Genetic Algorithm in Production planning

by

Fuladi Shubhendu

Department of Information Systems

The Graduate School

Pukyong National University

February 2021

A Study on Job Shop Scheduling Problem Using a Genetic Algorithm in Production planning

생산 계획에서의 유전자 알고리즘을 적용한 Job Shop Scheduling Problem에 관한 연구

Advisor: Prof. Chang Soo Kim

by

Fuladi Shubhendu

**A thesis submitted in partial fulfillment of the requirements
for the degree of**

Master of Engineering

**in the Department of Information Systems, the Graduate School,
Pukyong National University**

February 2021

A Study on Job Shop Scheduling Problem

Using a Genetic Algorithm in Production planning

A dissertation

by

Fuladi Shubhendu

Approved by:

(Chairman) ***Ha-Kyun Kim***

(Member) ***Young-Bong Kim*** (Member) ***Chang-Soo Kim***

February 19th, 2021

Contents

Contents

List of Figure

List of Table

생산 계획에서의 유전자 알고리즘을 적용한 Job Shop Scheduling Problem에 관한 연구

FULADI SHUBHENDU

부경대학교 대학원 정보시스템협동과정

요약

Job shop scheduling problem (JSSP) is a critical factor in smart manufacturing, which can provide a practical approach to arrange materials, resources properly under the constraints and requirements in real-life. In production planning, scheduling is one of the most important steps. Scheduling is a technique to determines when an operation is to be performed or when work is to be completed. The minimization of the make-span is the main goal of the job shop scheduling. To schedule the jobs there are some constraints for job processing in the job shop scheduling problem.

The genetic algorithm purpose to job shop scheduling for schedule jobs and minimize the make-span. As a result, it will show the best optimal sequence, optimal value, elapsed time, and Gantt chart for jobs and machines. A genetic algorithm for job shop scheduling gets an accurate result. multi-objective scheduling problem.

The non-dominated sorting algorithm being used for multi-objective scheduling problems. The architecture of NSGA-II is similar to GA. The only major difference steps are Combine parent and offspring population, Non-dominated sorting, Calculate crowding-distance, etc There are two goals, first is to minimize the total completion time (make-span) and the second is the total weighted early time and delay time (total weighted earliness and tardiness, TWET).

Keyword-*:* Job shop scheduling, genetic algorithm, Nondominated sorting genetic algorithm, Make-span,

Chapter 1.

Introduction

1.1 Background

Production planning is concerned with the efficient use of resources. It is the planning of production and manufacturing module in a company or industry it utilizes the resource allocation of activities of employees materials and production capacity to serve different customers, different types of production methods such as single manufacturing, batch production, mass production, etc. production planning is the predetermined process which includes the use of raw materials, resources, machines, etc. It helps to make the right decision at the right time and the right place to achieve maximum efficiency. production planning is used in several different industries including the agriculture industry, textile industry, etc. Production planning is a plan for future production, in which the facilities needed are determined and arranged. To maintain the flow is the goal of production planning. Scheduling and choosing the actual work to be started in the manufacturing facility. Setting up and delivering production orders to production facilities. Use of company resources with minimum downtime being put into the process. There are two types of production planning one is static planning and the other is dynamic planning. Static planning defines all steps in the process will not change. In dynamic planning, the environment being customized and steps in the process will be changed. Planning, routing, scheduling, and loading are the functions involves in production planning

Production planning
Planning
Routing
scheduling
Loading

Figure 1. Steps of production planning

1.2 Study Objective

In production planning, scheduling is one of the most important steps. Scheduling is a technique to determines when an operation is to be performed or when work is to be completed. The process of Arranging, controlling, and optimizing work and workloads in the production process is the main process in scheduling. To achieve the required rate of output with a minimum of delay. Determining the starting and completion timing of each operation and that of the manufacturing process so that the machines can be utilized to the maximum. The minimization of the make-span is the main goal of the job shop scheduling.

The genetic algorithm purpose to job shop scheduling for schedule jobs and minimize the make-span. As a result, it will show the best optimal sequence, optimal value, elapsed time, and Gantt chart for jobs and machines. A genetic algorithm for job shop scheduling gets an accurate result. The future goal is the development of a neural network scheduler for scheduling job-shop with using a genetic algorithm to generate an optimal schedule.

For multiobjective scheduling problems, a nondominated sorting genetic algorithm gives the best solution. The objectives are to minimize the total completion time that means minimize the make-span and total weighted early time and delay time (total weighted earliness and tardiness). These two objectives are conflicting objectives.

Chapter 2

Scheduling

To schedule the job there are mainly three types of scheduling; open shop scheduling, flow shop scheduling, and job shop scheduling. In open shop scheduling, there are no ordering constraints on the operation. If the processing sequence of each workpiece on the machine is the same then it is flow shop scheduling. And in job shop scheduling operation of the job totally in an ordered manner.

2.1 Job Shop Scheduling

One common scheduling problem is job shop scheduling. The job shop scheduling problem is one of the most occurring problems in the general production scheduling problem, each job consists of a sequence of tasks, which must be performed in a given order, and each task must be processed on a specific machine.

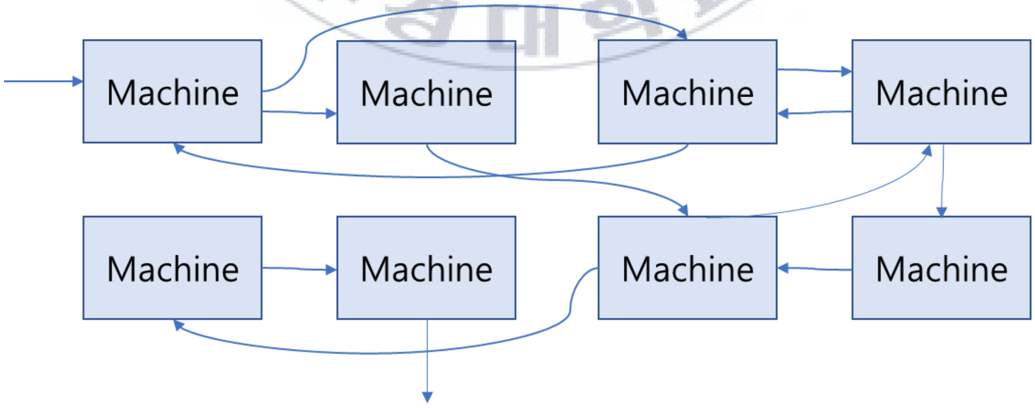


Figure 2. JSS

Completion of jobs on the machine with the shortest computational time is the main aim of scheduling the jobs. J jobs have to be processed on M different machines. the objective of the problem is to minimize the make-span. The job shop problem is widely known as the NP-hard problem (numeric problems). In job shop scheduling there is numbers of jobs i.e. J = {J1,J2,J3,…,JJ} need to be processed using M = {m1,m2,m3,…,mM} machines, each job consist of M different sub-task and processed by different machine.

Multi-job multi-machine JSSP: Given J jobs and M machines, each machine M_m has the particular function for different jobs, and each job J_j contains a series of phases $\{J_{j1}, J_{j2}, \dots, J_{jm}, \dots, J_{jM}\}$. Each phase J_{jm} must be processed by corresponding machine M_m

once with time T_{jm}. The whole process will end until all jobs are completed.

Make-span: The time of M machines to process J jobs, which is the maximum end time among all jobs as symbolized C_{max}. The objective of JSSP is to find minimum C_{max} as follows:

$$Objective = Min(C_{max})$$

2.2 Constraint of Job Shop Scheduling

Some constraints for the job shop scheduling problem are as follows

(1) Two operations of the job cannot be processed at the same time

(2) A job must not visit the same machine more than once. Each sub-task must be processed by machine m once.

(3) Operations can’t be interrupted. The process ends until the machine deal with all of the jobs

(4) Each machine can process only one job at a time

2.3 Job Shop Scheduling Solver

To solve the NP-hard m * n JSSP problem there are some methods (Where m is the number of machines and n is the number of jobs)

(1) To solve Job shop scheduling problem, there are some traditional algorithms:

These methods are simplest but not accurate. Some examples are as follows.

FIFO algorithm (First-in, First-out), Stochastic processing time (SHPT), Shortest processing time (SPT) method.

(2) Some population-based methods are as follows:

These methods give accurate results but they are time consumption methods.

Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Fruit Fly Optimization Algorithm (FFOA), Cat Swarm Optimization (CSO), Bat Algorithm (BA), Grey Wolf Optimization (GWO).

(3) Learning-based methods:

Shallow learning: Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP). Shallow learning methods give not accurate results.

Deep learning: Convolution Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Brief Networks (DBN). These methods are hard to construct.

(4) Gene-based:

The gene-based Algorithm gives accurate results.

The genetic algorithm: Based on genes to solve the optimization problem it gives accurate results.

Nondominated sorting Genetic Algorithm: It is similar to GA but is specifically used to solve problems with multiple goals.

2.3 Advantages of Job Shop Scheduling

• Easy to set up - Small businesses often implement jobs shops because they are simple to set up and the initial investment is minimal. You can begin with one or two machines and add them as needed.

• High flexibility - It is easy to add, change, or remove stages in the process. If a part needs to be re-machined, it is simply sent to the corresponding machine.

• Easy to increase capacity - Because it is easy to add a new machine to a job shop, you can increase capacity incrementally by adding a new machine.

2.3 Disadvantages of Job Shop Scheduling

- Hard to automate - Traditionally, it has been very difficult to automate job shops because automation has required consistency. This is starting to change, as we discuss below.
- Hard to schedule - Scheduling in job shops is notoriously hard. Most of the time, it is done on-the-fly. This causes a lack of standardization, meaning that the same product can take wildly different times to pass through the process.
- Hard to measure and improve - The inconsistent nature of the process makes it difficult to measure the different stages of the process, which means that the improvement of job shops is also a challenge.

Chapter 3

Genetic algorithm

The genetic algorithm is introduced by john Holland based on the concept of the theory of evolution. The mutation, crossover, selection are the main steps in GA. Genetic algorithms are specially used in the optimization problem toward better solutions. parameter for solving the problem is represented by a chromosome. The chromosome is converted into a string or numerical value by encoding, and each numerical value or The word string represents the gene in the chromosome, which represents a certain part of the solution, and then through mutation and mating, the next generation is generated, that is, different potential solutions, and finally the good solutions are survived by the fittest and the uncomfortable are eliminated. Make reservations for the next round of mating mutations to produce better solutions, until the set stopping conditions are reached, and hope to escape the local solution to find the global best solution in the future.

3.1 flow chart of genetic algorithm

The main concept of the genetic algorithm depends on chromosomes. The chromosome is a group of work processing sequences and composes of genes. The genetic algorithm can be used to solve most optimization problems, and this topic mainly focuses on the combination and application of GA and scheduling problems. Therefore, the following will introduce the concept of GA and implement it in the implementation unit. It explains how GA is applied to scheduling issues.

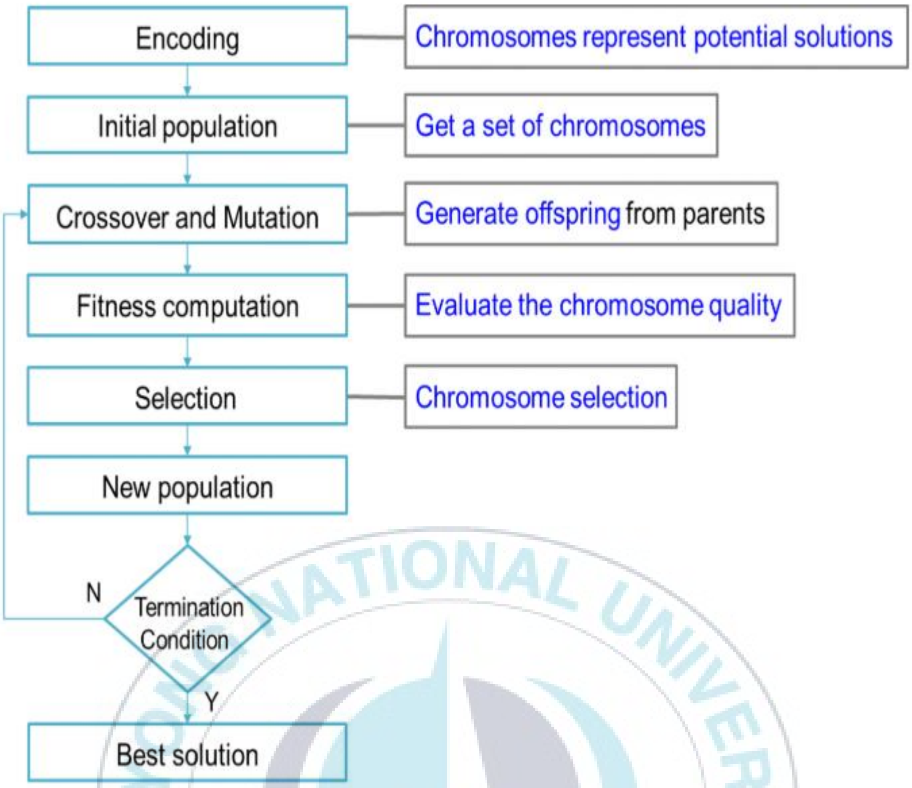


Figure 3. Genetic algorithm flow chart

3.1 Some initial parameters

Chromosome- Every individual in the population is known as a chromosome.

Gene- every unit in chromosomes is a gene.

Population- a collection of all chromosomes is known as population.

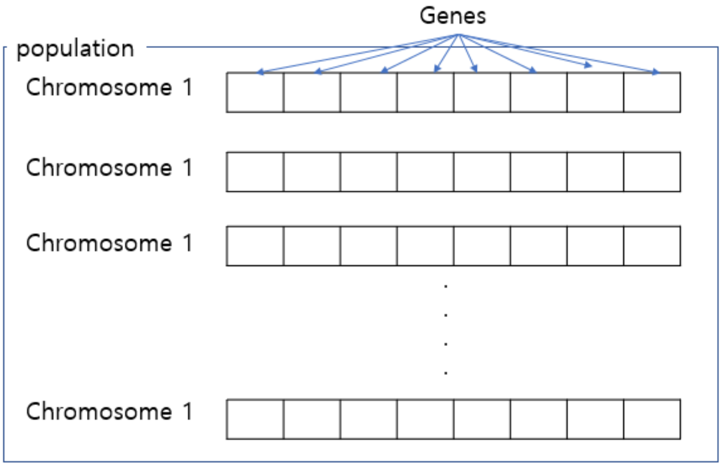


Figure 4. initial parameters

3.2 Operations of genetic algorithm

There are some steps in GA to solve the problem are as follows.

3.2.1 Encoding and Decoding

During GA entire execution, will the so-called code space (Coding space) and the solution space (Solution space) alternately runs inside, but mainly performing genetic manipulation encoding space, like mutation and mating operation, the solution space In the middle, perform evaluation and selection, as shown in the following figure:

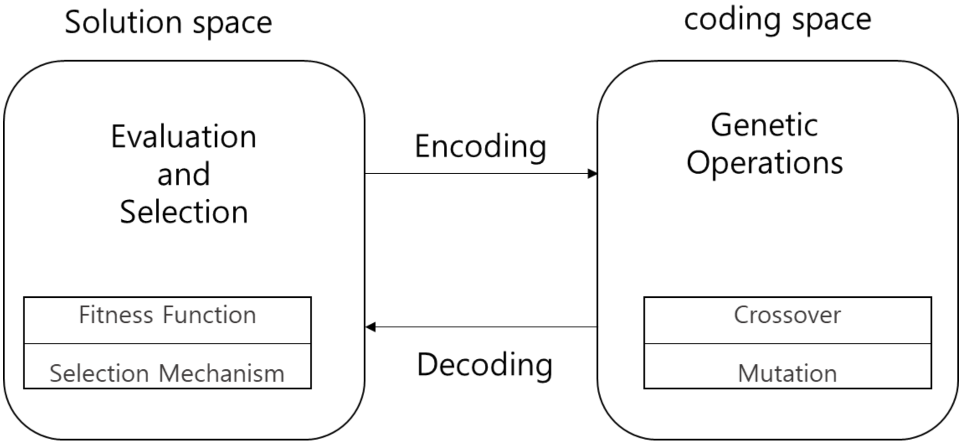


Figure 5. Encoding and Decoding

In the coding space, a solution is represented in the form of coding**.** The most common encoding method is Binary encoding, which is to convert the solution into a string of 1 and 0. This method is also most commonly used when your solution is in numerical form.

3.2.2 Initial population

In the beginning, we must first generate a group of chromosomes as the initial population, which is the so-called initial solution. The initial parents (Parents) of an organism are a bit like the first generation of ancestors of an organism, and then through the following steps of crossover and mutation to produce offspring (Offspring), to breed more and better offspring, so here The steps must first determine the population size.

In genetic operations-mating and mutations are used to produce offspring, that is, new potential solutions and it is expected to achieve exploration (exploration) The effect is to increase the diversity of solutions, hoping to escape the local solution and find more and better solutions.

3.2.3 Crossover

Some common methods for crossover are uniform crossover, single-point crossover, and multi-point crossover.

Uniform crossover is to randomly generate a binary code with the same length as the parental chromosome, called Crossover Mask. When the value in Mask is 1, the genes corresponding to the parental chromosome and Mask must be exchanged with each other, and the rest No exchange is necessary, and new offspring are produced.

Randomly select a certain gene position as the matching point, and cut the parental chromosome into two segments based on the gene position, then fix one segment and exchange the other segment to produce two new offspring.

The concept of multi-point mating is similar to single-point mating, except that multiple gene positions are selected at one time as the mating point, and then certain segments are fixed according to personal settings, and the rest are exchanged to produce new offspring. According to the mutation rate operation being performed

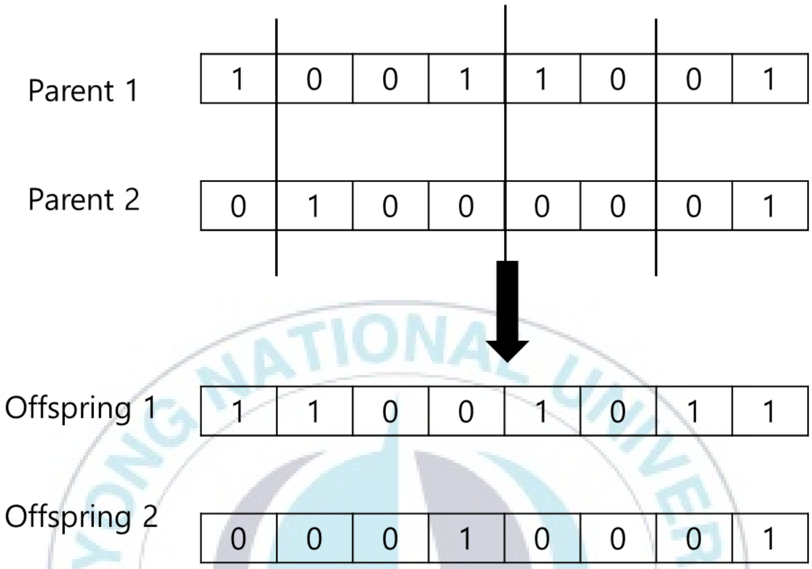


Figure 6. Two-point crossover

3.2.4 Mutation

To increase the diversity of solutions and avoid falling into local solutions, for each chromosome, it will be determined whether a certain chromosome should be mutated according to the set mutation rate. The genes in a single chromosome are changed randomly. The method is to randomly select several genes in the chromosome to exchange for a single chromosome, as shown in the following figure

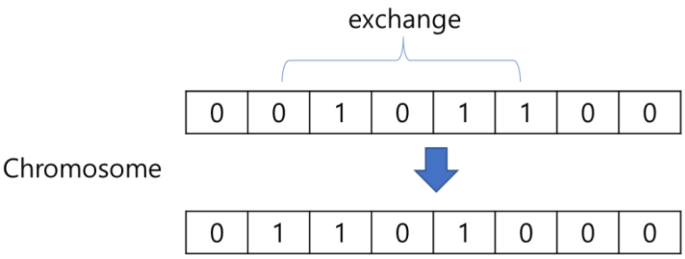


Figure 7. Mutation

2.3.5 Fitness function

When using GA to solve a problem, it is necessary to formulate the fitness function that belongs to the problem. The fitness function is a mechanism to evaluate the quality of chromosomes**.** The fitness value is converted to determine the fitness of the chromosome. The better the fitness value, the greater the probability that the chromosome will be retained and continue to multiply when the chromosome is selected in the next step. On the contrary, the worse the fitness value, the more likely it is to be eliminated.

2.3.6 Selection

To retain better chromosomes for evolution, this step is mainly based on the chromosomes produced in the above steps, through some selection mechanisms, to select, leaving the better-quality chromosomes to form a new group. For the next round of evolution, the following selection mechanisms will be introduced:

Roulette wheel-

The concept of the roulette method can be imagined as a game of darts shooting. First, we divide a rotatable roulette wheel into many fan-shaped areas of different sizes. Each chromosome has a corresponding exclusive fan-shaped area, and then we take out one Darts, shoot at this roulette at will when I shoot into which sector, the chromosomes belonging to this sector will be selected, so it is conceivable that with a larger area of chromosomes, there will be a greater chance of being selected, Of course, the fan-shaped area to which the chromosomes belong is not divided randomly, but derived from their fitness values.

2.3.7 Termination condition

Usually, a stopping mechanism is set as the termination condition. Once the set stopping condition has not been reached, the new population generated in the last step will be returned to the mutation and mating step, and then other steps will be executed in sequence. Continue to loop until the set stop condition is reached, and finally, the best solution obtained in all iterations can be obtained. Generally speaking, the common stopping mechanism is as follows:

- Number of iterations
- The number of times the solution remains unchanged for several consecutive times
- Stop when the solution difference is less than the set number for several consecutive times.

Chapter 4

Propose GA to JSSP

4.1 Problem description

The goal of this example is to minimize the total completion time (Make-span), which means to minimize the execution time of the entire schedule. This example is a 10x10 Job shop problem. There are a total of 10 workpieces and 10 machines. The processing order of each workpiece is different on each machine. Each workpiece will go through 10 processing operations. The table below records the processing machine and processing time of each workpiece in each processing operation.

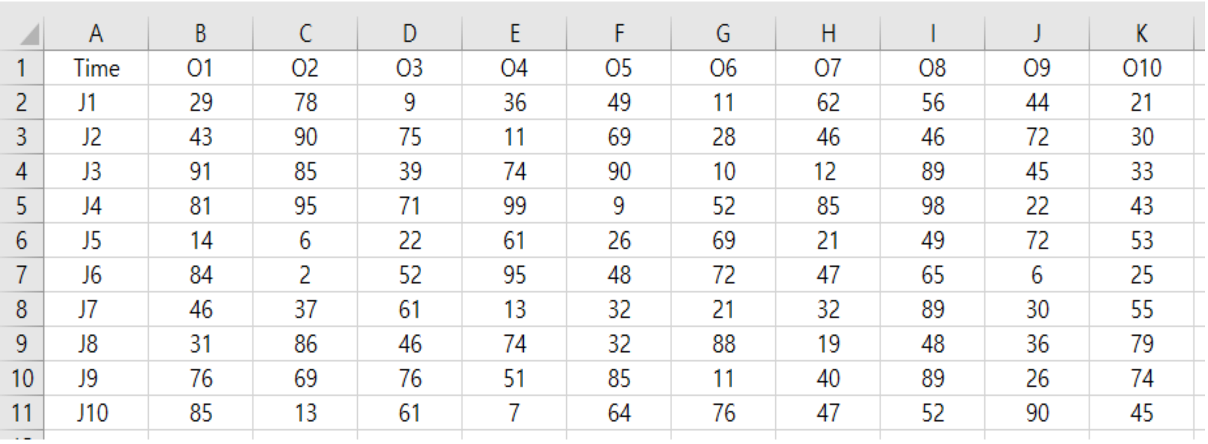
4.2 Data Description

Two tables are available here for scheduling one has records of processing time and the other has a record of processing machine sequence.

4.2.1 Processing time

The data for processing time is as follows.

Table 1. Processing time data set



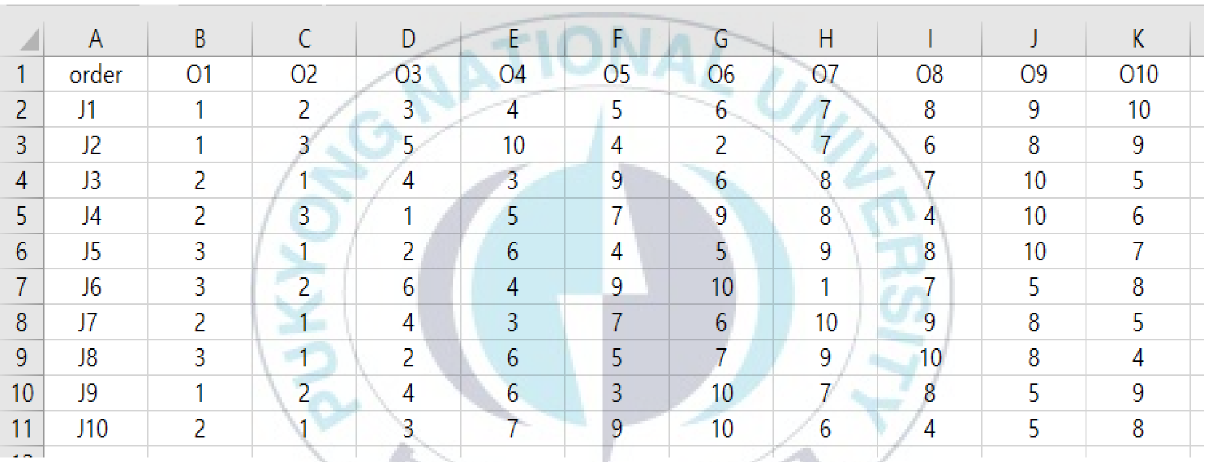
	A	B	C	D	E	F	G	H	I	J	K
1	Time	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
2	J1	29	78	9	36	49	11	62	56	44	21
3	J2	43	90	75	11	69	28	46	46	72	30
4	J3	91	85	39	74	90	10	12	89	45	33
5	J4	81	95	71	99	9	52	85	98	22	43
6	J5	14	6	22	61	26	69	21	49	72	53
7	J6	84	2	52	95	48	72	47	65	6	25
8	J7	46	37	61	13	32	21	32	89	30	55
9	J8	31	86	46	74	32	88	19	48	36	79
10	J9	76	69	76	51	85	11	40	89	26	74
11	J10	85	13	61	7	64	76	47	52	90	45

There are 10 jobs (J1, J2, J3, J4, J5, J6, J7, J8, J9, J10) and 10 operations (O1, O2, O3, O4, O5, O6, O7, O8, O9, O10). Every operation has different processing times with respective jobs.

4.2.2 Machine Sequence

In this table, there is an order of 10 machines. Every operation (O1, O2, O3, O4, O5, O6, O7, O8, O9, O10) goes through every job with the given sequence.

Table 2. Machine sequence dataset



	A	B	C	D	E	F	G	H	I	J	K
1	order	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
2	J1	1	2	3	4	5	6	7	8	9	10
3	J2	1	3	5	10	4	2	7	6	8	9
4	J3	2	1	4	3	9	6	8	7	10	5
5	J4	2	3	1	5	7	9	8	4	10	6
6	J5	3	1	2	6	4	5	9	8	10	7
7	J6	3	2	6	4	9	10	1	7	5	8
8	J7	2	1	4	3	7	6	10	9	8	5
9	J8	3	1	2	6	5	7	9	10	8	4
10	J9	1	2	4	6	3	10	7	8	5	9
11	J10	2	1	3	7	9	10	6	4	5	8

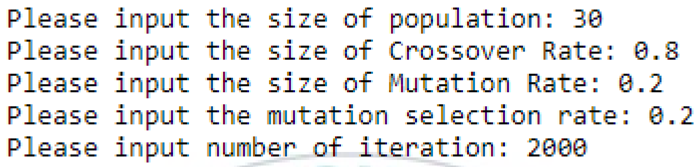
4.2 Methodology

The concept of this method is to represent the chromosome as a group of workpiece processing sequence, a gene represents a workpiece processing operation, according to the number of times the workpiece appears on the chromosome, to know the current processing operations of each workpiece, and then to correspond to the processing of each workpiece Machine and processing time to schedule.

Suppose there is now a Job shop scheduling problem with N workpieces and M machines. That chromosome will be composed of N x M genes, because each workpiece will only be processed once on each machine It is processed by M machines, so each workpiece will appear M times in the chromosome. Here is an example of the above 10 x 10 Job shop problems.

Each chromosome has a total of 10 x 10 = 100 genes

The raw inputs are needed to carry on other steps like crossover, mutation. Some inputs are population size, crossover rate, size of mutation rate, mutation-selection rate, and the number of iterations.



```
Please input the size of population: 30
Please input the size of Crossover Rate: 0.8
Please input the size of Mutation Rate: 0.2
Please input the mutation selection rate: 0.2
Please input number of iteration: 2000
```

Figure 8. Inputs for GA

In this experiment the raw inputs are:

Population size = 30 (integer value)

Crossover rate = 0.8 (float value)

Mutation rate = 0.2 (float value)

Mutation selection rate = 0.2 (float value)

Number of iterations = 2000 (integer value)

The number of mutation jobs is equal to the multiplication of the number of genes and mutation-selection rates.

Number of mutational jobs = num gene * mutation-selection rate

Two-point crossover- At the beginning, a set of random sequences used to select the parental chromosomes will be generated first, and then two or two will be picked out from the sequence. According to the mating rate, it is determined whether to perform mating. The two-point mating method is used to generate Two offspring, and replace the original mother chromosome

Repair

This example is a 10 x 10 Job shop problem, so the number of occurrences of each artifact on the chromosome is 10, but due to the mating action above, it will cause the occurrence of artifacts in some chromosomes to be less than 10 or greater than 10. An infeasible scheduling solution is formed, so here it is necessary to perform repair actions on the infeasible chromosomes to make it a feasible scheduling

Mutation-

the mutation is carried out by gene displacement. The mutation method is as follows:

According to the mutation-selection rate, determine what percentage of the genes in the chromosome are to be mutated. Assuming that there are six genes in each chromosome and the mutation-selection rate is 0.5, there are 3 genes to be mutated.

Randomly select the genes to be shifted, assuming that 5, 2, 6 are selected (in this case, the gene at this position is to be mutated)

Adaptation value calculation-

Calculate the completion time of the scheduling result formed by each chromosome and record it so that it can be compared during subsequent selection. It should be noted here that because this is a problem of minimization, the fitness value calculated for each chromosome, that is, the completion time, must be recorded in a reciprocal manner (chrom_fitness) so that the roulette method can be used later Select the chromosome with the smaller completion time, but there is still another record of the original completion time (chrom_fit) of each chromosome so that when the last round of the best solution is selected, it can be directly compared

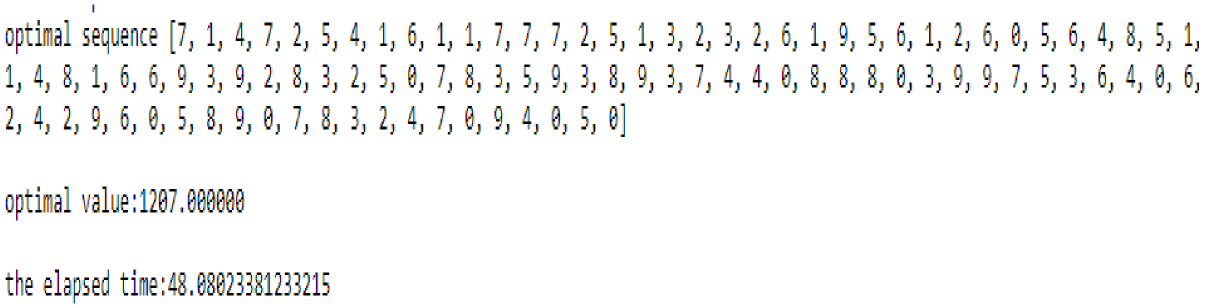
For selection, the Roulette wheel mechanism was adopted here.

Compare

First compare the completion time of each chromosome (chrom_fit), select the best solution found in this round (Tbest_now), and then compare it with the best solution found so far (Tbest), once the solution in this round is better than the one found so far The solution is better, just replace Tbest and record the scheduling result of the solution

4.3 Results

After the iteration ends, output the best-scheduled result (sequence_best) found in all iterations, the completion time of the result, and the program execution time



```
optimal sequence [7, 1, 4, 7, 2, 5, 4, 1, 6, 1, 1, 7, 7, 7, 2, 5, 1, 3, 2, 3, 2, 6, 1, 9, 5, 6, 1, 2, 6, 0, 5, 6, 4, 8, 5, 1,
1, 4, 8, 1, 6, 6, 9, 3, 9, 2, 8, 3, 2, 5, 0, 7, 8, 3, 5, 9, 3, 8, 9, 3, 7, 4, 4, 0, 8, 8, 8, 0, 3, 9, 9, 7, 5, 3, 6, 4, 0, 6,
2, 4, 2, 9, 6, 0, 5, 8, 9, 0, 7, 8, 3, 2, 4, 7, 0, 9, 4, 0, 5, 0]

optimal value:1207.000000

the elapsed time:48.08023381233215
```

Figure 9. The best solution for JSSP using GA

The relation between make-span and generation are as follows:

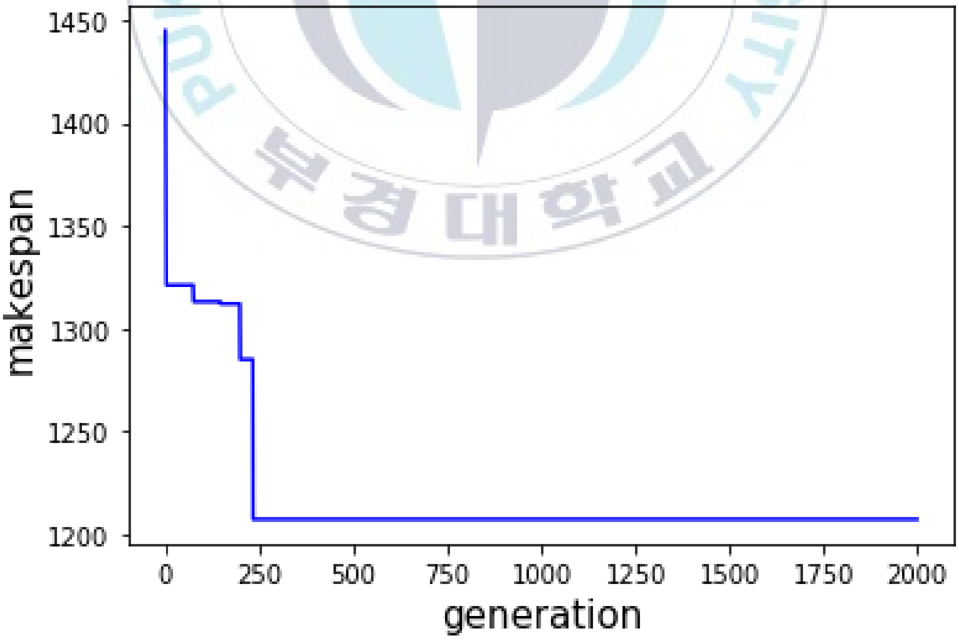


Figure 10. The relation between make-span and generation

Gantt chart

The Gantt chart shows the visualization of the best sequence for jobs.

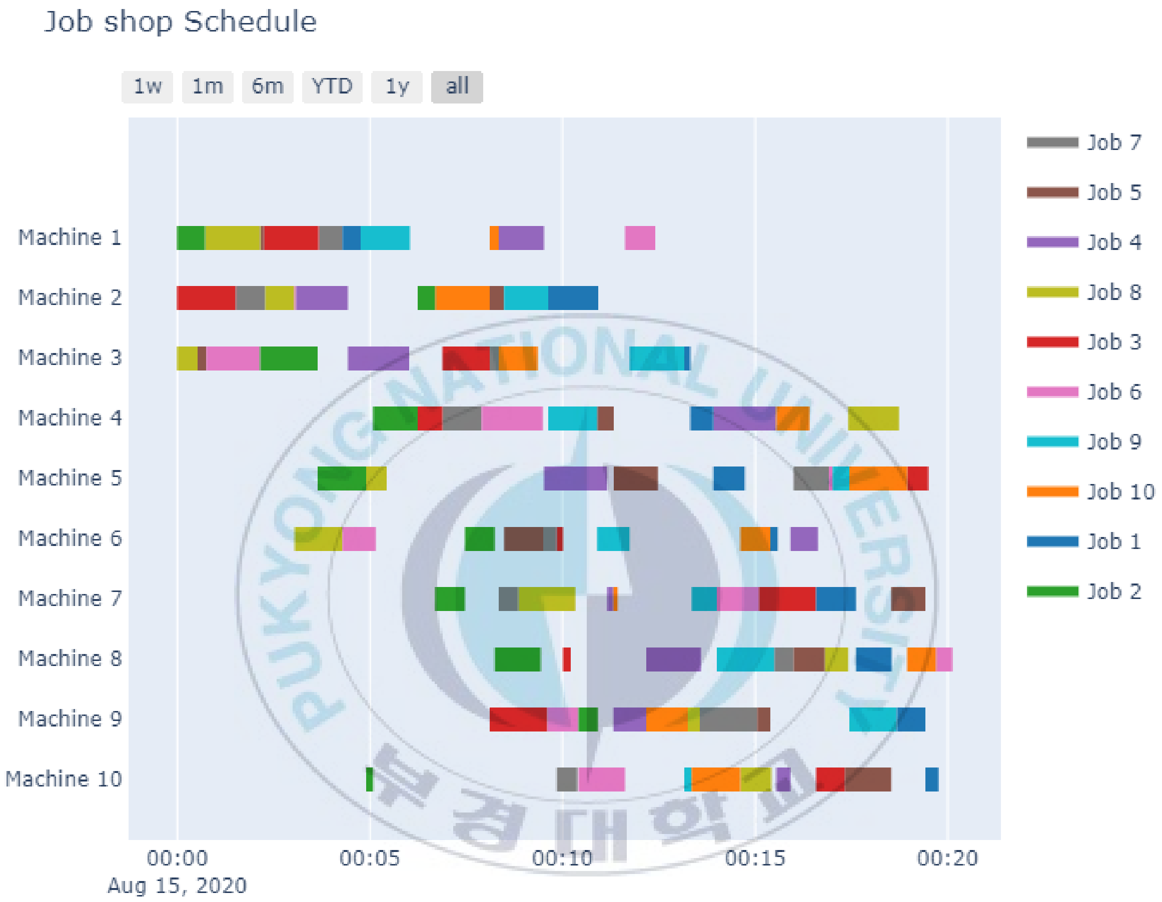


Figure 11. Gantt chart for JSSP using GA

chapter 5

Non-dominated Sorting Genetic Algorithm-II

The non-dominated sorting algorithm is use for multi-objective scheduling problems. There are two goals, first is to minimize the total completion time (make-span) and the second is the total weighted early time and delay time (total weighted earliness and tardiness, TWET). Each workpiece will go through 10 processing operations.

5.1 Architecture of NSGA-II

The architecture of NSGA-II is as shown in the figure below. Its architecture is similar to GA. The only major difference steps are Combine parent and offspring population, Non-dominated sorting, Calculate crowding-distance, etc.

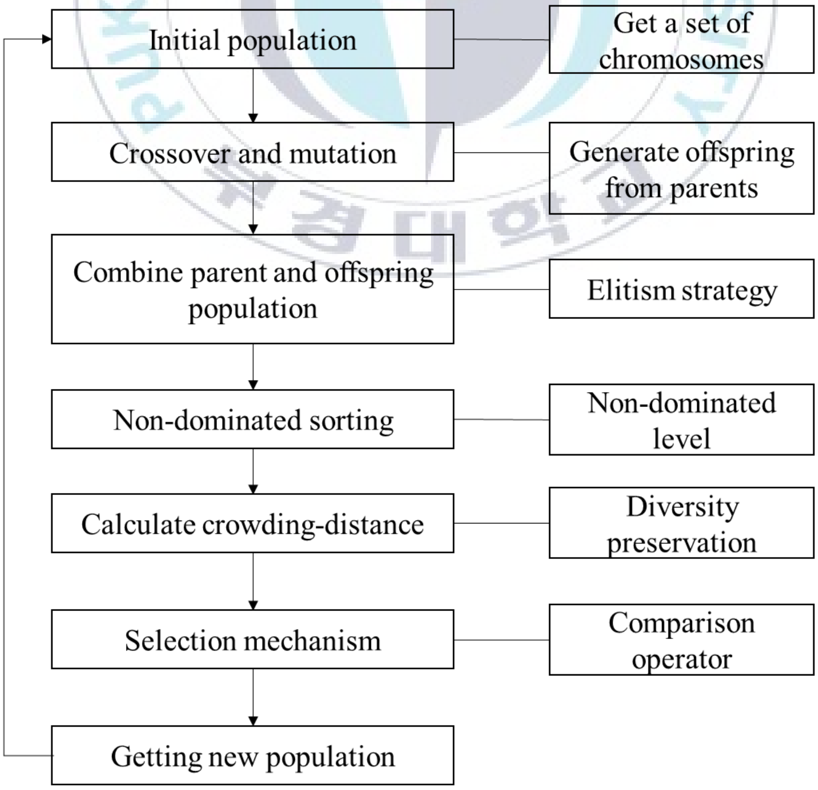


Figure 12. flow chart of NSGA-II

5.2 Nondominated Sorting Approach

NSGA-II proposes a faster non-overlap sorting method and has less time complexity.

Pseudo Algorithm

```
 Fast-non-dominated-sort(P)
For each $p \in P$
    $S_p = \emptyset$
    $n_p = 0$
    for each $q \in P$
        if $(p < q)$ then                        If p dominates q
            $S_p = S_p \cup \{q\}$               Add q to the set of solutions dominated by p
        else if $(q < p)$ then
            $n_p = n_{p+1}$                      Increment the domination counter of p
      if $n_p = n_p + 1$                          p belongs to the first front
        $p_{rank} = 1$
        $F_1 = F_1 \cup \{p\}$
$i = 1$                                          Initialize the front counter
while $Fi \neq \emptyset$
  $Q = \emptyset$                                Used to store the members of the next front
  For each $p \in F_i$
      for each $q \in S_p$
          $n_q = n_q - 1$
          if $n_q = 0$ then                      q belongs to next front
              $q_{rank} = i + 1$
              $Q = Q \cup \{q\}$
  $i = i + 1$
  $F_i = Q$
```

Figure 13. pseudo algorithm for Fast-non-dominated-sort

Step 1. Calculating two entities for each solution**:** n_p, S_p

p is the code name of the calculated solution, n_p represents the number of solutions p (which can be imagined as how many solutions p is bullied), and Sp is the solution set of solutions p (that is, who is bullied by p Ling), taking the example above as an example, the following table can be obtained:

It can be seen from the figure on the right that solution A exceeds all solutions, So S_A={B, C, D}, and n_A =0; B is only given to Ling Yue by A, and Ling Yue understands D, so n_B=1, S_B={D}, and so on...

Step 2. Finding the members of the first nondominated front: $n_p = 0$

Through the previous step, we can get the table of transcending relations between each solution and other solutions, and then we will rank these solutions so that they can be used as indicators for the final selection of chromosomes (solutions). The concept is shown in the figure below. We will use The relationship table divides these solutions into different levels. solutions of the first level have the highest level (that is, the Platonic frontier solution), and the second level has the second high level, and so on, the higher the level has The higher priority is selected as the new population (population).

Therefore, in the beginning, we must first find the first-level priority solution, that is, the solution with n_p=0 in the table formed in the previous step. In this case, it is solution A and the solution on the blue line, and give these solutions The sort level of is 1.

Step 3. For each solution with $n_p = 0$, visit each solution (q) in the set of these solutions S_p, and subtract one from the number of transitions n_p in the set.

(For each solution with $n_p = 0$, we visit each member (q) of its set S_p and reduce its domination count by one.)

Step 4. In the process of visiting each solution in the previous step, if any solution np becomes 0, the solution belongs to the second non-transiting front edge, so it is assigned a rank of 2

(If for any member the domination count becomes zero, it belongs to the second nondominated front.)

From Step 2, we know that the solution with n_p=0 is only A, and the solutions overtaken by A are B, C, D (from Sp), so we visit these solutions one by one and reduce np by one, The updated table is as follows, and during the visit, it is found that the np of solution B and solution C becomes 0, so they are the second non-transitory solutions, so they are assigned a rank of 2, that is The second priority is selected as the population solution

Step 5. Repeat the above steps until all the leading edges are identified

(The above procedures are continued until all fronts are identified.)

5.3 Elitism strategy

To ensure that the remaining chromosomes are excellent and feasible, the elite strategy is adopted before fitness evaluation. In simple terms, this strategy is to combine the parental generation before mating and mutation with mating and mutation The later offspring are kept together for selection to prevent the chromosomes from getting worse and worse, and to avoid losing the high-quality solutions found.

5.4 Crowding distance

To maintain the diversity of solutions and make choices when different solutions are at the same non-overstepping level, the crowding distance method is proposed here to evaluate the density relationship between each solution in the group and its surrounding solutions. The concept is as follows As shown, when calculating the crowding distance of a specific solution, we will follow the non-transit front edge where the solution is located, and find the two adjacent solutions that are closest to the specific solution along with each target in the front edge. , To calculate the average distance of these two solutions, and finally add up the distance calculated for each target to get the crowded distance of the specific solution. For the two-objective example in the following figure, the crowded distance of the ith solution at its leading edge is the average side length of the rectangle enclosed by the two closest solutions to the solution i.

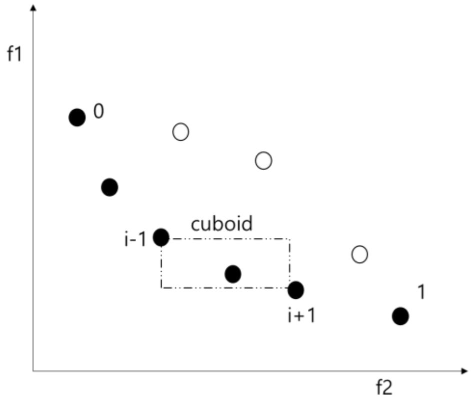


Figure 14. crowding distance

As mentioned in the previous paragraph, calculating the crowding distance helps to maintain the diversity of the solutions. This means that when choosing solutions from a group of solutions that are located on the same non-transit front edge, they tend to choose the one with a larger crowding distance. The solution, because the larger the crowding distance, the greater the difference between the solution and other solutions, which helps to avoid falling into the local solution in the subsequent iteration of the algorithm, and achieve the effect of exploration. In the hope of finding more and better solutions.

```
Crowding-distance-assignment (I)
$l = | I |$
For each i, set $I[i]_{distance} = 0$
For each objective m
    $I = sort(I, m)$
    $I[1]_{distance} = I[l]_{distance} = \infty$
    For i = 2 to ( l - 1 )
        $I[i]_{distance} = I[i]_{distance} + ( I[i + 1].m – I[i - 1].m)/ f_m^{max} - f_m^{min}$
```

Figure 15. pseudo-code for crowding-distance

5.5 selection mechanism

After the above process, each chromosome (solution) in the final population has two attributes:

- Nondomination rank
- Crowding distance

Finally, when selecting new population members, they will be selected according to the following rules:

1. First, compare the non-overlapped levels of each solution, the higher the level of the solution (smaller number), the higher the priority to be selected
2. If the non-overlapped levels of the two solutions are the same, compare the crowded distance, the larger the crowded distance, the higher the priority

5.6 NSGA procedure

Finally, through the diagram below, the entire process of NSGA-II will be integrated. For each iteration, the following actions will be performed until the set conditions are reached:

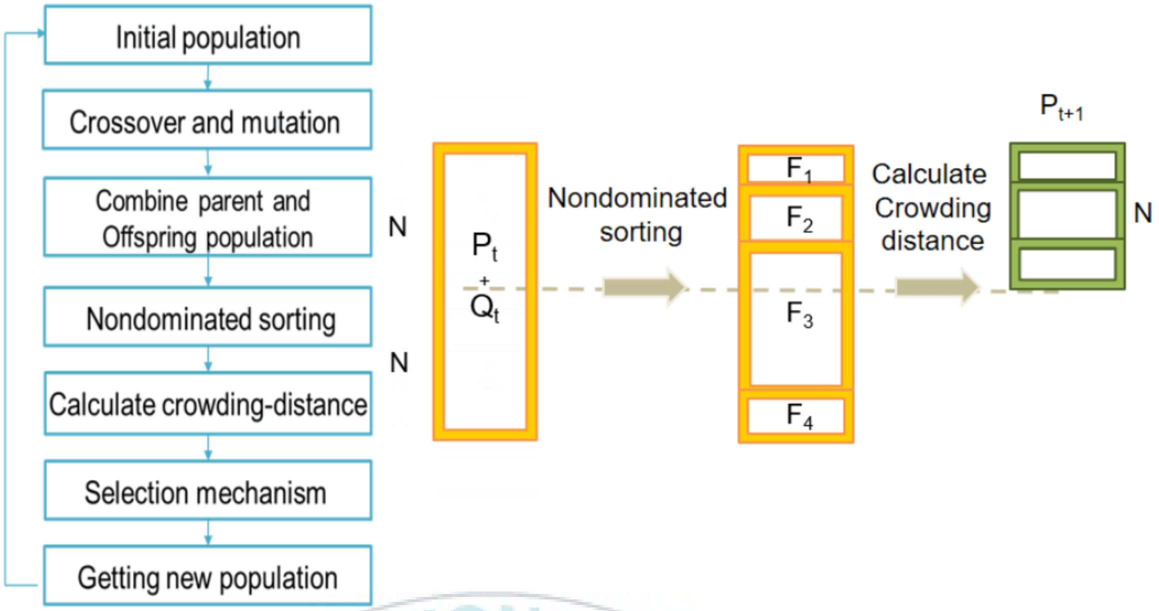


Figure 16. NSGA procedure

1. First, there is an initial population (parental generation) P $_t$ which contains N chromosomes, which will produce offspring Q $_t$ after mutation and mating.
2. Due to the elitism strategy, the parents and offspring are kept together for selection.
3. Then perform non-dominated sorting to obtain the non-dominated level of each solution (F $_1$ <level 1>, F $_2$ <level 2>.....).
4. Finally, select the new N chromosomes as the population for the next iteration. First, select according to the level of non-overlapped. The number of remaining chromosomes to be selected into the new population is less than the next non-overlapped population. The number of chromosomes in the higher level is selected based on the crowded distance, and the larger crowded distance is selected to enter the new population.
5. Finally, a new population P $_{t+1\ is}$ generated, enter the next iteration, and repeat the above process.

Chapter 6

Propose NSGA to JSSP

6.1 Problem description

Suppose there is a job shop scheduling problem with N workpieces and M machines. That chromosome will be composed of N x M genes because each workpiece will only be processed once on each machine. It is processed by M machines, so each workpiece will appear M times in the chromosome. This example is a 10x10 Job shop problem with 10 workpieces and 10 machines. This problem is a multi-objective scheduling problem. There are two goals in total: Minimizing the total completion time (Make-span) and the total weighted early time and delay Time (Total weighted earliness and tardiness, TWET), the workpiece information is shown in the table below. The workpiece information is presented in the processing procedure of the workpiece. Each workpiece will go through 10 processing operations.

Scheduling goal

This example is a multi-objective scheduling problem. There are two objectives: Minimize the total completion time (Make-span) and Total weighted earliness and tardiness (TWET). These two objectives are conflicting objectives, make-span It is expected that the sooner the completion is better, but TWET hopes that the completion time is more in line with the delivery date, and the penalty value will be given for completion too early or too late. Therefore, a trade-off must be made between these two solutions.

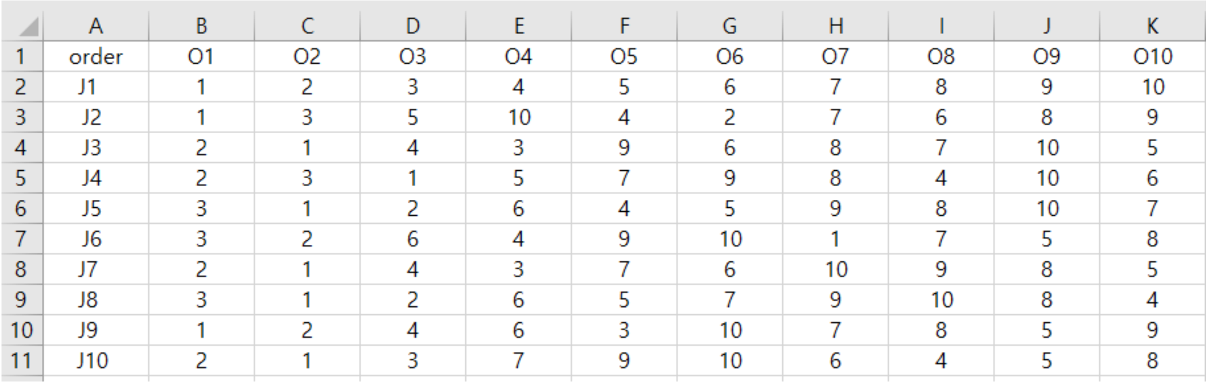
6.2 Data Description

The concept of this method is to express the chromosome as a set of work processing procedures for the workpiece. A gene represents the processing work of one workpiece. According to the number of times the workpiece appears on the chromosome, the current processing work of each workpiece is known, and then the processing of each workpiece is corresponding. The machine and processing time can be used for scheduling. Three data tables are available here for scheduling one has records of processing time, the second has a record of processing machine sequence and the third is a priority and due date.

6.2.1 Processing time

There are 10 jobs (J1, J2, J3, J4, J5, J6, J7, J8, J9, J10) and 10 operations (O1, O2, O3, O4, O5, O6, O7, O8, O9, O10). Every operation has different processing times with respective jobs.

Table 3. processing time dataset

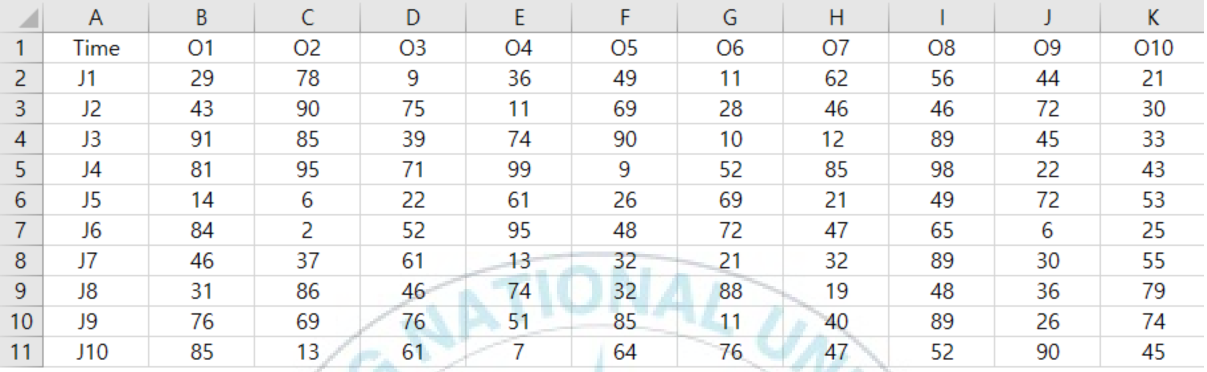


	A	B	C	D	E	F	G	H	I	J	K
1	order	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
2	J1	1	2	3	4	5	6	7	8	9	10
3	J2	1	3	5	10	4	2	7	6	8	9
4	J3	2	1	4	3	9	6	8	7	10	5
5	J4	2	3	1	5	7	9	8	4	10	6
6	J5	3	1	2	6	4	5	9	8	10	7
7	J6	3	2	6	4	9	10	1	7	5	8
8	J7	2	1	4	3	7	6	10	9	8	5
9	J8	3	1	2	6	5	7	9	10	8	4
10	J9	1	2	4	6	3	10	7	8	5	9
11	J10	2	1	3	7	9	10	6	4	5	8

6.2.2 Machine Sequence

In this table, there is an order of 10 machines. Every operation (O1, O2, O3, O4, O5, O6, O7, O8, O9, O10) goes through every job with the given sequence

Table 4. Machine Sequence dataset

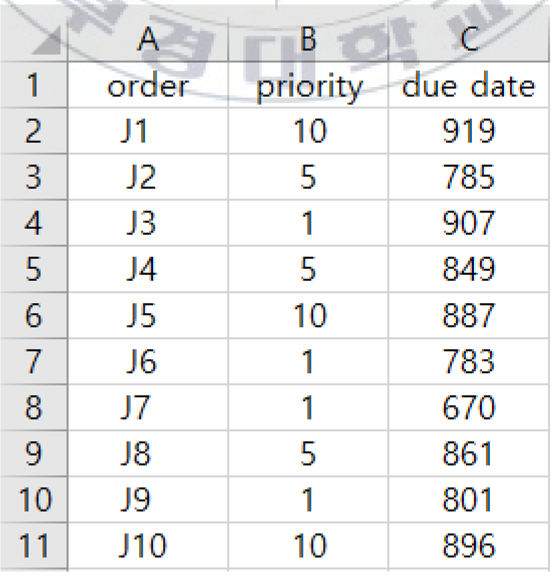


	A	B	C	D	E	F	G	H	I	J	K
1	Time	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
2	J1	29	78	9	36	49	11	62	56	44	21
3	J2	43	90	75	11	69	28	46	46	72	30
4	J3	91	85	39	74	90	10	12	89	45	33
5	J4	81	95	71	99	9	52	85	98	22	43
6	J5	14	6	22	61	26	69	21	49	72	53
7	J6	84	2	52	95	48	72	47	65	6	25
8	J7	46	37	61	13	32	21	32	89	30	55
9	J8	31	86	46	74	32	88	19	48	36	79
10	J9	76	69	76	51	85	11	40	89	26	74
11	J10	85	13	61	7	64	76	47	52	90	45

6.2.3 Priority and Due date

In this data set, every job (J1, J2, J3, J4, J5, J6, J7, J8, J9, J10) has its priority and due date.

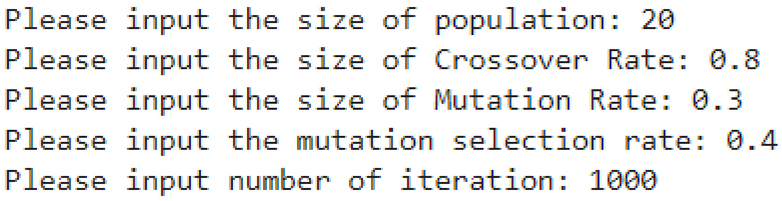
Table 5. Priority and Due date dataset



	A	B	C
1	order	priority	due date
2	J1	10	919
3	J2	5	785
4	J3	1	907
5	J4	5	849
6	J5	10	887
7	J6	1	783
8	J7	1	670
9	J8	5	861
10	J9	1	801
11	J10	10	896

6.3 Methodology

The raw inputs are needed to carry on other steps like crossover, mutation. Some inputs are population size, crossover rate, size of mutation rate, mutation-selection rate, and the number of iterations.



```
Please input the size of population: 20
Please input the size of Crossover Rate: 0.8
Please input the size of Mutation Rate: 0.3
Please input the mutation selection rate: 0.4
Please input number of iteration: 1000
```

Figure 17. Input for JSSP using NSGA

In this experiment the raw inputs are:

Population size = 20 (integer value)

Crossover rate = 0.8 (float value)

Mutation rate = 0.3 (float value)

Mutation selection rate = 0.4 (float value)

Number of iterations = 1000 (integer value)

Generate initial solution

According to the population size set above, the initial population is generated randomly, each chromosome has 10 x 10 = 100 genes, and each chromosome is stored by a list

mating

The double-point mating method is used here. In the beginning, a set of random sequences used to select parental chromosomes will be generated, and then two by two will be picked out from the sequence, and the mating rate will be used to determine whether to mate or not. If so, then Mating produces two offspring and replaces the original parental chromosomes

mutation

The mutation-selection rate determines the percentage of genes in the chromosome to be mutated. Assuming that each chromosome has six genes and the mutation-selection rate is 0.5, then there are 3 genes to be mutated.

Randomly select the gene to be shifted, assuming that 5, 2, and 6 are selected (here means that the gene under this position needs to be mutated)

fix

This example is a 10 x 10 job shop problem. Therefore, the number of occurrences of each artifact on the chromosome is 10 times. However, due to the above mating action, the number of occurrences of artifacts in some chromosomes will be less than 10 or more than 10. Form an infeasible scheduling solution, so here must repair the infeasible chromosomes to make it a feasible schedule

Non-dominated sorting and calculate crowded distance

parents and offspring are kept together for perform the the non-dominated sorting to obtain non-dominated level (F1, F2, F3, F4) . Sort the solutions of each target from smallest to largest The number of chromosomes in the higher level is selected based on the crowded distance, and the larger crowded distance is selected to enter the new population.

Input: the index of the chromosome contained in the leading edge to be calculated, the fitness value of all chromosomes at present (you can use the index entered by the former to grasp the fitness value of the chromosome to be calculated)

Output: the crowded distance of the calculated chromosome

Select function

This function calls the function to calculate the crowding distance (calculate_crowding_distance) because, in the process of selecting chromosomes to form a new population, when the number of chromosomes remaining to be selected is less than the number of chromosomes in the current crossing front, it is necessary to Judge which chromosome I want to choose based on the crowding distance.

Input: the population size, the index of the chromosomes contained in each leading edge obtained by the non-linger function, the fitness value of all chromosomes to be selected, and the scheduling result of each chromosome list

Output: the new ethnic group list and the index in the original ethnic group list within the ethnic group

Fitness calculation

Calculate two target values for each chromosome make-span and TWET

Here, the parent (parent_list) and offspring (offspring_list) will be merged into a big list (total_chromosome), and subsequent selections will be made from this big list to generate a new ethnic group. After that performed Non-transit sorting calculation and selection.

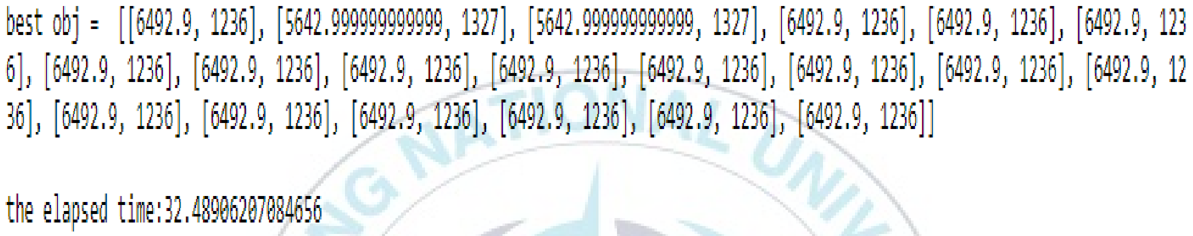
Then compare the best solutions found in this round with the best solutions found in the iteration so far

Result

Calculated two target values for each chromosome-makespan and TWET

In the end, it will output the best solution found in all iterations. Since this is a multi-objective problem, there may be multiple sets of solutions. The setting here is to output the same number of solutions as the population size

output



```
best obj =  [[6492.9, 1236], [5642.999999999999, 1327], [5642.999999999999, 1327], [6492.9, 1236], [6492.9, 1236], [6492.9, 123
6], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 12
36], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236], [6492.9, 1236]]

the elapsed time:32.48906207084656
```

Figure 18. Best solution JSSP using NSGA

Gantt chart

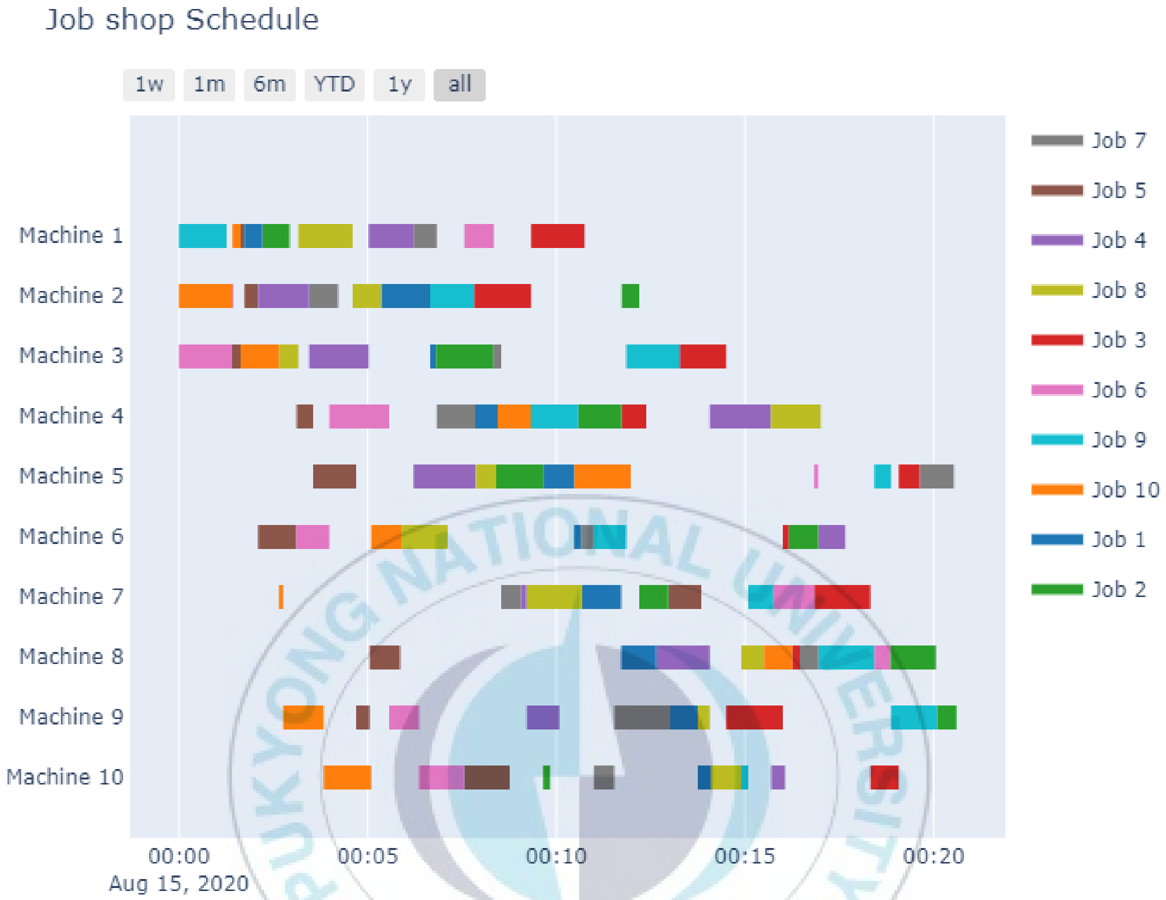


Figure 19. Gantt chart for JSSP using NSGA

Chapter 7.

Future Studies and conclusion

This thesis focused on how to schedule jobs in production planning. The job shop scheduling is one of the most occurring problems in production planning. The traditional methods for job shop scheduling are FIFO, stochastic processing time, Shortest processing time these methods are simple but not give accurate results.

and some population-based methods like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Fruit Fly Optimization Algorithm (FFOA), Cat Swarm Optimization (CSO), Bat Algorithm (BA), Grey Wolf Optimization (GWO). These methods are time consumption methods.

The genetic algorithm purpose to job shop scheduling for schedule jobs and minimize the make-span. As a result, it will show the best optimal sequence, optimal value, elapsed time, and Gantt chart for jobs and machines. A genetic algorithm for job shop scheduling gets an accurate result. the Genetic Algorithm produced better optimality solutions compared to the AIS algorithm

For multi-objective scheduling problems, a nondominated sorting genetic algorithm gives the best solution. The objectives are to minimize the total completion time that means minimize the make-span and total weighted early time and delay time (total weighted earliness and tardiness). These two objectives are conflicting objectives. The future goal is the development of a neural network scheduler for scheduling job-shop with using a genetic algorithm to generate an optimal schedule.

References

[1] G. Da Col and E. C. Teppan, “Google vs IBM: A constraint solving challenge on the job-shop scheduling problem,” *Electron. Proc. Theor. Comput. Sci. EPTCS*, vol. 306, pp. 259–265, 2019, doi: 10.4204/EPTCS.306.30.

[2] H. Zhu, M. Chen, Z. Zhang, and D. Tang, “An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing constraint,” *IEEE Access*, vol. 7, pp. 125113–125121, 2019, doi: 10.1109/ACCESS.2019.2938548.

[3] B. Khan, S. Hanoun, M. Johnstone, C. P. Lim, D. Creighton, and S. Nahavandi, “Multi-objective job shop scheduling using i-NSGA-III,” *12th Annu. IEEE Int. Syst. Conf. SysCon 2018 - Proc.*, pp. 1–5, 2018, doi: 10.1109/SYSCON.2018.8369558.

[4] O. V. Shylo and H. Shams, “Boosting Binary Optimization via Binary Classification: A Case Study of Job Shop Scheduling,” pp. 1–23, 2018, [Online]. Available: http://arxiv.org/abs/1808.10813.

[5] T. Yang, J. Li, and H. Chen, “Multi-objective flexible job-shop scheduling problems with limited resource constraints using nondominated sorting genetic algorithm II,” *2018 IEEE Int. Conf. Inf. Autom. ICIA 2018*, no. August, pp. 1127–1131, 2018, doi: 10.1109/ICInfA.2018.8812599.

[6] G. Janes, M. Perinic, and Z. Jurkovic, “Applying Improved Genetic Algorithm for Solving Job Shop Scheduling Problems,” vol. 3651, pp. 1–5, 2017.

[7] L. Xu, Z. Y. Xia, and H. Ming, “Study on improving multi-objective flexible job shop scheduling based on Memetic algorithm in the NSGA-II framework,” *Proc. 2016 2nd Int. Conf. Cloud Comput. Internet Things, CCIOT 2016*, pp. 1–7, 2017, doi: 10.1109/CCIOT.2016.7868293.

[8] M. Zandieh, A. R. Khatami, and S. H. A. Rahmati, “Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm,” *Appl. Soft Comput. J.*, vol. 58, pp. 449–464, 2017, doi: 10.1016/j.asoc.2017.04.060.

[9] Y. Liu, D. Ouyang, W. Gu, and L. Wang, “A discrete artificial bee colony algorithm for permutation flow shop scheduling,” *Proc. - 2016 9th Int. Symp. Comput. Intell. Des. Isc. 2016*, vol. 2, pp. 161–164, 2016, doi: 10.1109/ISCID.2016.2046.

[10] A. Ansari and A. A. Bakar, “A Comparative Study of Three Artificial Intelligence Techniques: Genetic Algorithm, Neural Network, and Fuzzy Logic, on Scheduling Problem,” *Proc. - 2014 4th Int. Conf. Artif. Intell. with Appl. Eng. Technol. ICAIET 2014*, pp. 31–36, 2015, doi: 10.1109/ICAIET.2014.15.

[11] A. S. Muhamad, S. Deris, and Z. Zakaria, “Minimize the makespan for job shop scheduling problem using artificial immune system approach,” *J. Theor. Appl. Inf. Technol.*, vol. 80, no. 1, pp. 141–146, 2015.

[12] S. Noor, M. Ikramullah Lali, and M. Saqib Nawaz, “Solving Job Shop Scheduling Problem With Genetic Algorithm,” *Int . (Lahore)*, vol. 27, no. 4, pp. 3367–3371, 2015.

[13] N. Al-Hinai and T. Y. Elmekkawy, “Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm,” *Int. J. Prod. Econ.*, vol. 132, no. 2, pp. 279–291, 2011, doi: 10.1016/j.ijpe.2011.04.020.

[14] Z. Jia, X. Lu, J. Yang, and D. Jia, “Research on job-shop scheduling problem based on genetic algorithm,” *Int. J. Prod. Res.*, vol. 49, no. 12, pp. 3585–3604, 2011, doi: 10.1080/00207543.2010.481293.

[15] Y. Mati, S. Dauzere-Pérès, and C. Lahlou, “A general approach for optimizing regular criteria in the job-shop scheduling problem,” *Eur. J. Oper. Res.*, vol. 212, no. 1, pp. 33–42, 2011, doi: 10.1016/j.ejor.2011.01.046.

[16] J. Liu, C. Zhang, L. Gao, and X. Wang, “Research on flexible job-shop scheduling problem under uncertainty based on genetic algorithm,” *Proc. - 2010 6th Int. Conf. Nat. Comput. ICNC 2010*, vol. 5, no. Icnc, pp. 2462–2467, 2010, doi: 10.1109/ICNC.2010.5583493.

[17] H. W. Ge, L. Sun, Y. C. Liang, and F. Qian, “An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans*, vol. 38, no. 2, pp. 358–368, 2008, doi: 10.1109/TSMCA.2007.914753.

[18] M. Chandrasekaran, P. Asokan, S. Kumanan, T. Balamurugan, and S. Nickolas, “Solving job shop scheduling problems using artificial immune system,” *Int. J. Adv. Manuf. Technol.*, vol. 31, no. 5–6, pp. 580–593, 2006, doi: 10.1007/s00170-005-0226-3.

[19] A. Yahyaoui and F. Fnaiech, “Recent trends in intelligent job shop scheduling,” *2006 1st IEEE Int. Conf. E-Learning Ind. Electron. ICELIE*, pp. 191–195, 2006, doi: 10.1109/ICELIE.2006.347186.

[20] F. Zhao, Y. Hong, D. Yu, X. Chen, and Y. Yang, "Integration of artificial neural networks and genetic algorithm for job-shop scheduling problem," *Lect. Notes Comput. Sci.*, vol. 3496, no. I, pp. 770–775, 2005, doi: 10.1007/11427391_123.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002, doi: 10.1109/4235.996017.

[22] C. Y. Lee, S. Piramuthu, and Y. K. Tsai, "Job shop scheduling with a genetic algorithm and machine learning," *Int. J. Prod. Res.*, vol. 35, no. 4, pp. 1171–1191, 1997, doi: 10.1080/002075497195605.

[23] Y. Wu and B. Li, "Job-shop scheduling using genetic algorithm," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 3, pp. 1994–1999, 1996, doi: 10.1109/icsmc.1996.565434.

Acknowledgments

From the bottom of heart, I would like to say a big thanks of gratitude to my advisor professor Chang-Soo Kim. Your calmness and smiling face always make a positive environment for working. Being your student is my fortune. Can’t forget your help when my father was in critical condition.

Thanks to each professor who taught me in two years of study. I would like to thank my lab mate’s Shao and Sehun, thanks for fun and support, and thanks for my all dearest friend who support me. I found myself very grateful for having such kind of friends.

Finally, I want to thank my family. Especially to my late father professor Kshitij H. Fuladi, who was powerful role models, He is no more with us but his motivational words and blessing always with me to achieve success and to my mom Darshana K. Fuladi and sister Sharvari who taught me love and kindness.

I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas well. I would like to get suggestions on the thesis report, it will beneficial for future improvement.