Job Shop Scheduling Problem Using Genetic Algorithms

Sahar Habbadi
Industrial Engineering
Faculty of Science and Technology
University Sidi Mohammed Ben Abdellah
Fès-Morocco
Sahar.habbadi@usmba.ac.ma

Brahim HERROU
Professor of Industrial Engineering
Faculty of Science and Technology
University Sidi Mohammed Ben Abdellah
Fès-Morocco
brahimherrou@yahoo.fr

Souhail SEKKAT
Professor of Industrial Engineering
Ecole Nationale Supérieure d'Arts et Métiers (ENSAM-MEKNES)
Meknès, Morocco
s.sekkat@ensam.ac.ma

Abstract

The Job Shop scheduling problem is very widespread considering its utility in the industry. Several researchers have worked on this subject with the aim of optimizing work sequences.
This case study provides an overview on genetic algorithms which present a real potential for solving this type of combinatorial problem of job shop scheduling problem.
During this study, the application of this method will be done manually in order to understand the procedure and the process of executing programs based on genetic algorithms.
It is a problem where the decision analysis must figure strongly along the process because of the numerous choices and allocations of jobs to machines at the right time, in a very specific order and over a given duration. This operation is done at the operational level and research must find an intelligent method to find the best and most optimal combination.

Keywords
Optimization, Metaheuristics, Scheduling, Job Shop Scheduling problem, and Algorithm Genetic.

1. Introduction

The Job Shop scheduling problem has always been considered among the most interesting subjects in research and which aims to develop new intelligent methods to optimize the number of jobs, time and use of machines.
Planning and assigning jobs to machines requires determining the right sequence and the most optimal time to complete a set of jobs while guaranteeing the employer a significant gain in terms of time as well as in terms of machine operation.

The most optimal sequence is the best solution in a search space where there is infinity of solutions; it is therefore a combinatorial optimization problem.
The resolution of combinatorial optimization problems is done while using metaheuristic methods given their ability to determine an approximate solution in a reasonable time.
In this article, the use of genetic algorithms will allow the optimization of the make span of Job Shop Scheduling Problem.

The Job Shop Scheduling (JSS) problem dealt with in the rest of this article will consider the following elements: a set of 6 jobs carried out on the 6 machines available to manufacture products in a minimum time while optimizing the sequences of the job. In reality, the scheduling of jobs on machines takes into account several constraints, in particular: machine stoppages due to unexpected breakdowns, maintenance time and other production hazards, that is, the lack of control over work sequences and especially the lack of control of the complete deadlines for carrying out all the work.

2. Literature review

In the article by Ke Shen et al. [1], the two-phase genetic algorithm (CTGA) is applied to the case of a generalized flexible flowshop problem (GFFS) while introducing mixed-integer multi-objective mixed-integer programming (MIP) for the GFFS, then the observation that the CTGA is efficient for the multi-objective optimization due to the flexibility provided by the Genetic algorithm.
the work of Sarahí Báez et al. [2] have the same objective as our article, it aims to study a machine scheduling problem and the criterion used to assess the quality of the planning is the makespan, except that the method used to solve this problem is based on hybrid algorithms which combines GRASP and variable neighborhood search metaheuristics unlike our work which will mainly focus on genetic algorithms.

Chen-Fu Chien and Yu-Bin Lan [3] discuss the case of dynamic planning for semiconductor manufacturing while developing an agent-based approach that integrates deep reinforcement learning and hybrid genetic algorithm for independent parallel machine planning.

Jerry Swan et al. [4] talk in their article about metaheuristic which is based on an iterative process that guides and modifies the operations of subordinate heuristics to efficiently produce high quality solutions. At each iteration, it manipulates either a unique complete (or partial) solution, or a set of such solutions.
The Job Shop scheduling problem is one of the most difficult combinatorial optimization problems where research has not yet been able to develop a performing algorithm to generate a satisfying optimal sequence.
Lawrence M. Wein and Philippe B. Chevalier [5] define a job shop scheduling problem with three dynamic decisions: assign due dates to jobs arriving exogenously, release jobs from a backlog to the job shop, and sequence jobs to each of two shifts workshop work.

Several researchers have worked on this subject while using different methods including:
The use of Lagrangian relaxation to schedule job shops [6]. Lawrence M. Wein and Philippe B. Chevalier [5] offer a general two-step approach to solving this problem: release and sequence work to minimize inventory of work in progress, subject to completion of work at a specified rate, and given the policies of set deadlines dates that attempt to minimize the lead time to the due date, subject to the late work constraint.
Runwei Cheng et al. [7] discuss in their article: A tutorial survey of job-shop scheduling problems using genetic algorithms—I the representation on genetic algorithms and the different representation schemes proposed for JSP in part I of his article and various hybrid approaches of genetic algorithms and conventional heuristics in the second part.

Kumar Ritwik and Sankha Deb [8] in their paper: A genetic algorithm-based approach for optimization of scheduling in job shop environment develops a genetic algorithm based approach to solve the scheduling optimization problem in Job Shop manufacturing environment.

3. Genetic algorithm steps

- Population initialization: Population P can be defined as a set of chromosomes. The initial population, which is the first generation, is usually created randomly [9].
- Fitness function: Calculate the different and possible bad and good solutions, and then eliminate bad solutions from the population. A fitness value consists also in evaluating the solutions.

There are different techniques to implement selection in genetic algorithms. They are:

- Tournament selection
- Roulette wheel selection
- Uniform selection
- Rank selection

- Crossover: The crossover method is used to generate new solutions from the existing parents available in the mating pool after applying selection method. This operator exchanges the gene information between the solutions in the meeting pool

- Mutation: it is an operation which consists in changing certain genes of the population to diversify it.

4. Theoretical algorithms

Initialize population;
Calculate fitness function;
While (fitness value! = termination criteria)
{ Selection; Crossover; Mutation;Calculate fitness function; }

5. Problem statement

Amir Ghasemi et al. [10] say in their article that there are several parameters that impact production, especially products requiring advanced decision-making, such as in the case where there is an increase in production speed or the production load which implies difficulties in the decision-making process. This problem leads to a line of research for researchers interested in the field of artificial intelligence to develop decision support tools. Planning is also part of the decision-making process, this is why it is necessary to conceive an algorithm allowing to seek the optimal solution in order to resolve the original problem.

Concerning Karim Tamssaouet et al. [11] for them, it is not easy to optimize a global scheduling model because of the difficulties which arise at the level of work areas that must be scheduled locally while ensuring coordination between local schedules.

JSS problem analysis provides valuable insight into the solution of scheduling problems encountered in the most complex and realistic systems. Therefore, heuristics and metaheuristics are preferred methods for scheduling the job shop scheduling. The Genetic Algorithm is considered the best-known optimization technique for a class of combinatorial problems.

In this article we consider a matrix of order 6*6 which corresponds to 6 Jobs and 6 machines.

The machines are subjected to the operations below:

It is supposed that the execution time of the operations is the same on each machine.

Table 1. Distribution of operation on machines

	M1	M2	M3	M4	M5	M6
J1	O11	O21			O51	
J2		O22	O32	O42		
J3			O33	O43		O63
J4	O14		O34	O44	O54	
J5	O15	O25				O65
J6			O36			O66

The following symbols are defined:
J = {J1, J2, J3, …Jj, …Jn} represents the job set.
M = {M1, M2, M3, …Mi, …Mm} represents the machine set.
Oij --> O: Operation, i: corresponds to the machine number and j: to the job number.
The first step to resolve a combinatorial problem by genetic algorithm is to establish population initial.
According to T. Loukil et al. [12]: "Population initialization is an important task in evolutionary algorithms because it can affect the convergence speed and also the quality of the final solution. If no information about the solution is available, then random initialization is the most commonly used method to generate candidate solutions".
There is two ways to construct the initial population which are:
- Random Initialization:
Initial population contains only random solutions.
- Heuristic Initialization:
Initial population contains solutions founded by heuristic method.

In this article the random initialization is preferred as a method in order to avoid premature convergence.
- The coding of the genetic algorithm is also part of the first step that is carried out to create our population and it is a crucial step to succeed in finding the optimal solution.
- The coding step consists on representing a solution as a string that conveys the necessary information.
- It consists also in modeling each solution by a chromosome, we consider:
O11, O14, O15, O21, O22, O25, O32, O33, O34, O36, O42, O43, O44, O51, O54, O63, O65, O66. The initial population corresponds to the set of feasible solutions to solve this problem.
The numerous sequences figures on this problem. Bellow table of sequences matrix represents the set of feasible solutions in initial population iteration in order to lead the program to an optimal sequences in the following steps:

Table 2. table of sequences matrix.

	M1	M2	M3	M4	M5	M6
J1	1	2			1	
J2		1	2	2		
J3			1	1		1
J4	2		3	3	2	
J5	3	3				2
J6			4			3

Let start by defining the priority of job execution on machines:
The first case considers the bellow order of sequences:
M5, M1, M2, M3, M4, M6 Prioritize sequence of J1 work on M5 instead of M1.
To evaluate each chromosome, it is necessary to calculate the weight of each chromosome. The value of makespan can use Gantt diagram:
Calculate the value of fitness function by using Gantt diagram will be the second step which is necessary to evaluate the quality of solution:

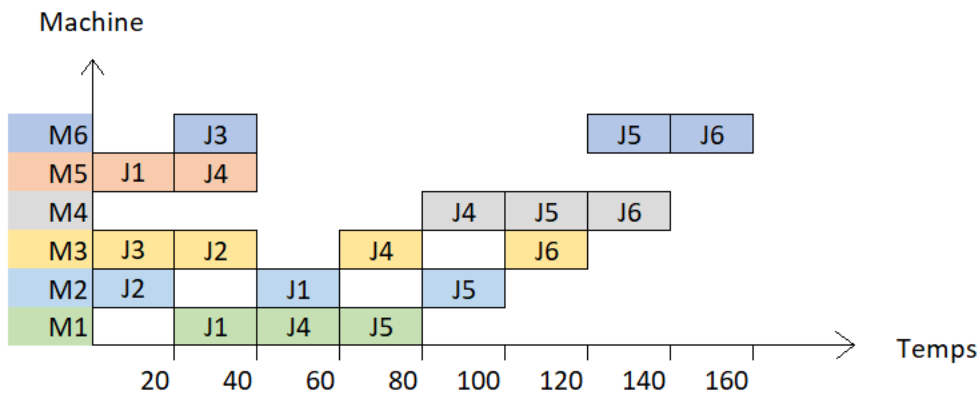

Figure 1. Gantt chart.

T= fitness function = 160 seconds
The second solution to integrate in the population to search the best solution is: M1, M2, M3, M4, M5, and M6.

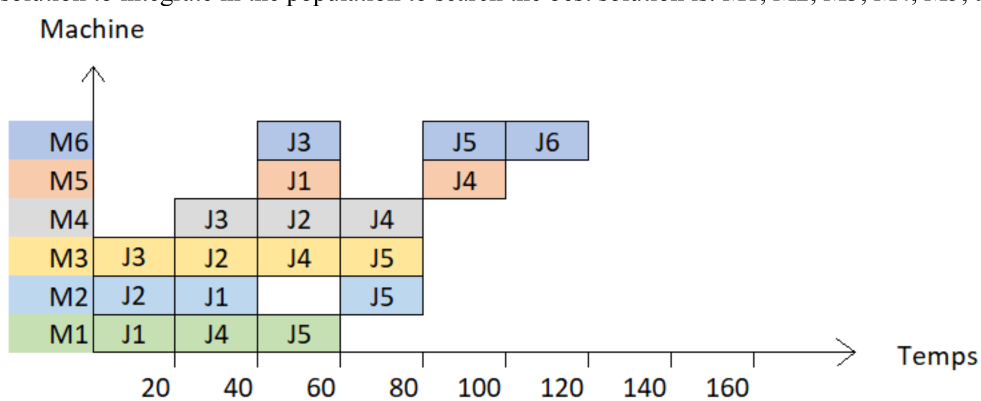

Figure 2. Gantt chart.

T= fitness function = 120 seconds
The 3rd feasible solution possible to integrate into the population to find the best solution: M1, M2, M4, M3, M5and M6.

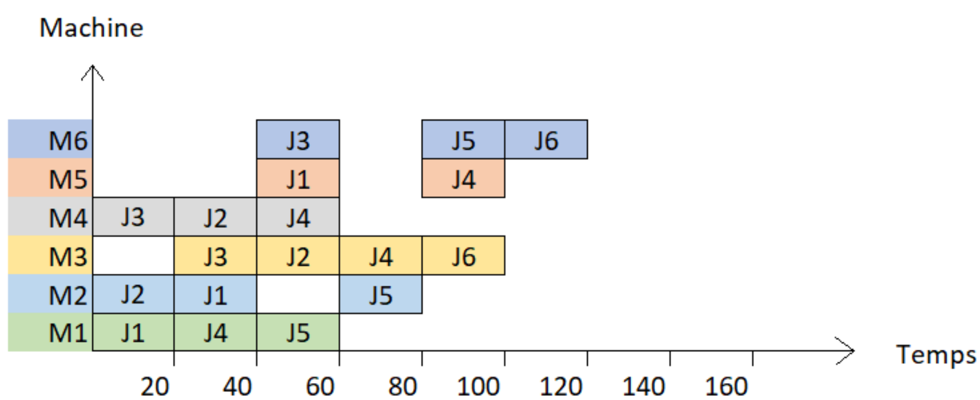

Figure 3. Gantt chart.

T= fitness function = 120 seconds
The 4th feasible solution possible to integrate into the population to find the best solution: M2, M1, M3, M4, M5 and M6.

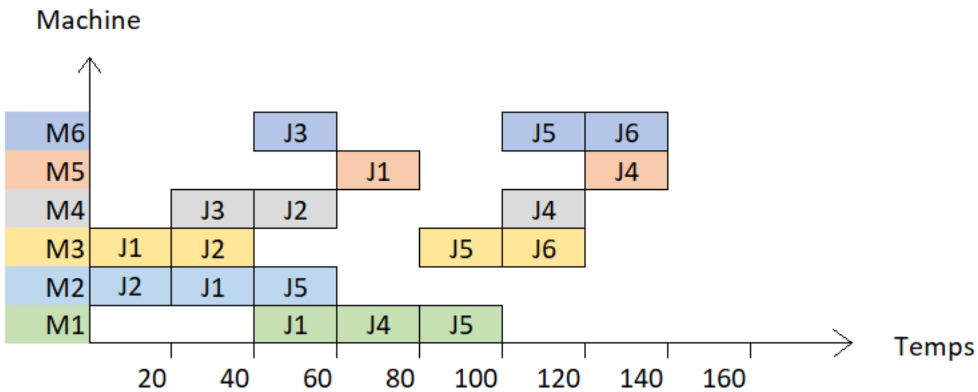

Figure 4. Gantt chart.

T= fitness function = 140 seconds

The third step is to select the parents; there are several methods to select them which are:
Roulette Wheel Selection:

Narayan Behera [13] defined it as a method that is used for selecting all the individuals for the next generation. It consists in associating with each chromosome a segment whose length is proportional to its fitness. These segments are then concatenated on a graduated axis that we normalize between 0 and 1.
We then draw a number random with uniform distribution between 0 and 1, then the selected segment is identified and the corresponding chromosome. With this technique, the good chromosomes will be more often selected.
Tournament selection:
This technique randomly selects two or more individuals from the population and the strongest is selected, i.e. the one with the most interesting fitness.
According to Brad L. Miller and David E. Goldberg [14] Tournament selection is simple to code. Tournament selection can also adjust the selection pressure to adapt to different domains.
Uniform selection:

The selection is done randomly, uniformly and without intervention of the adaptation value. Each individual therefore has a probability 1/P of being selected, where P is the total number of individuals in the population.
In this thesis, we have chosen uniform selection to avoid premature convergence. Frederick M. Cohan [15] says that Uniform selection is generally supposed to cause convergence between populations while drift has the effect of causing divergence.

We select parents from the existing population by randomly choosing two values of the fitness function to define the new parents.
Let's consider the following parents:
P1 = [O51, O54 , O11, O14, O15, O22, O21, O25, O33 , O32, O34,O36, O44, O45, O46, O63, O65,O66]
which correspond to the order of the following machines: M5, M1, M2, M3, M4, M6.
P2 = [O11, O14, O15, O22, O21, O25, O33 , O32, O34,O36, O43, O42 , O44, O51, O54, O63, O65, O66]
Which correspond to the order of the following machines: M1, M2, M3, M4, M5, M6.

The fourth step is crossover; it consists of crossing two parents create two children.
P1 = [O51, O54 , O11, O14, O15, O22, O21, O25, O33 , O32, O34,O36, O44, O45, O46, O63, O65,O66]
P2 = [O11, O14, O15, O22, O21, O25, O33 , O32, O34, O36, O43, O42 , O44, O51, O54, O63, O65, O66]

P1= [O51, O54 , O11, O14, O15, O22, O21, O32, O34 ,O36, O43, O42 , O44, O45, O46, O63, O65, O66]
P2 = [O11, O14, O15, O22, O21, O25, O33 , O43, O42, O32, O34,O36, O44, O51, O54, O63, O65,O66]
The fifth step is the mutation

The mutation will be done at P1 at the following points:

O51 □ O53
O54 □ O56
OS1 = [O_{53}, O_{56}, O_{11}, O_{14}, O_{15}, O_{22}, O_{21}, O_{32}, O_{34} ,O_{36}, O_{43}, O_{42} , O_{44}, O_{45}, O_{46}, O_{63}, O_{65}, O_{66}]
OS2 = [O_{11}, O_{14}, O_{15}, O_{22}, O_{21}, O_{25}, O_{33} , O_{43}, O_{42}, O_{32}, O_{34},O_{36}, O_{44}, O_{51}, O_{54}, O_{63}, O_{65},O_{66}]
The new Gantt chart is:

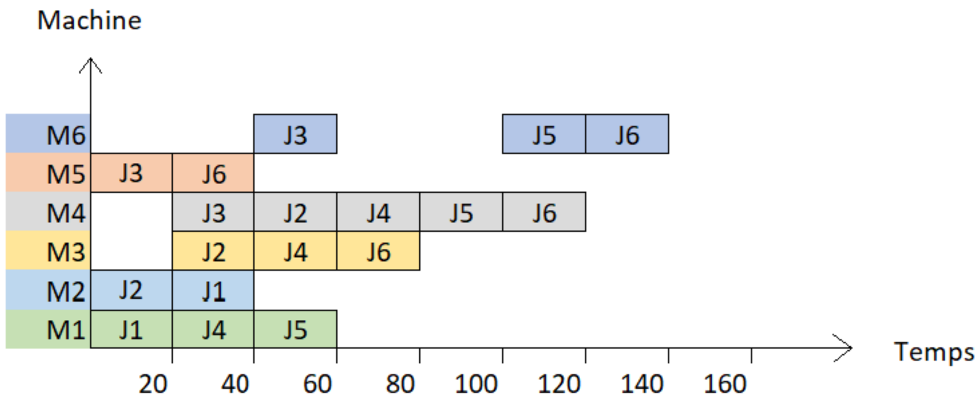

Figure 5. Gantt chart.

T= 140 seconds

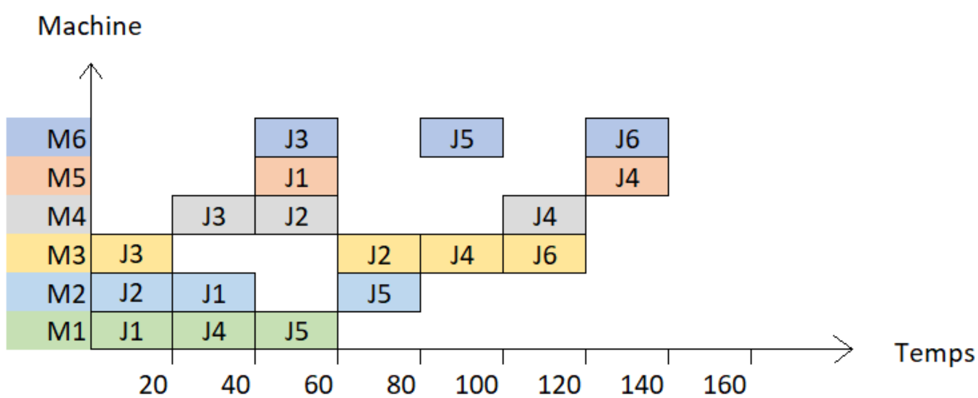

Figure 6. Gantt chart.

T= 140 seconds
The new job sequence matrix with this iteration is as follows:

Table 3. table of sequences matrix.

	M1	M2	M3	M4	M5	M6
J1	1	2				
J2		1	1	2		
J3				1	1	1
J4	2		2	3		
J5	3	3		4		2
J6			3	5	2	3

Based on the job sequence matrix, we move on to the third iteration where we integrate into the new population the sequence obtained after crossing and mutation.
We have 6 scenarios corresponding to the 6 different sequences:

a) M1, M2, M3, M4, M5, M6,
b) M3, M1, M2, M4, M5, M6,
c) M2, M1, M3, M4, M5, M6
d) M1, M2, M6, M4, M5, M3
e) M1, M2, M3, M5, M4, M6
f) M1, M2, M3, M4, M6, M5

A) M1, M2, M3, M4, M5, M6,

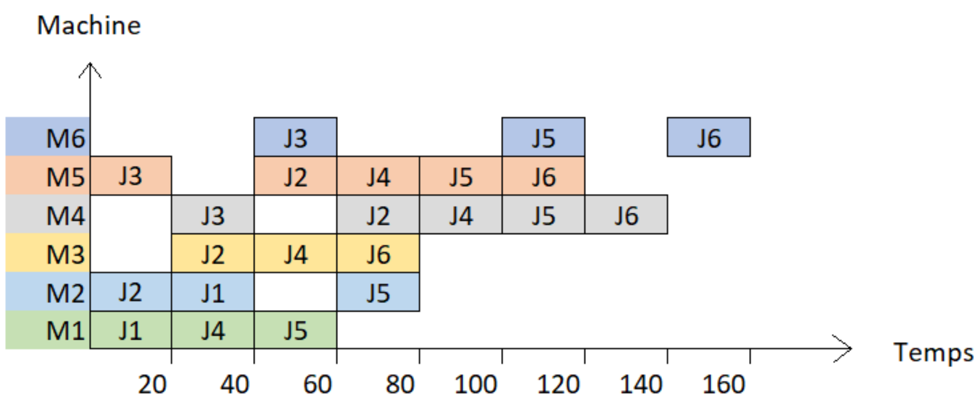

Figure 7. Gantt chart.

T=160 seconds

B) M3, M1, M2, M4, M5, M6,

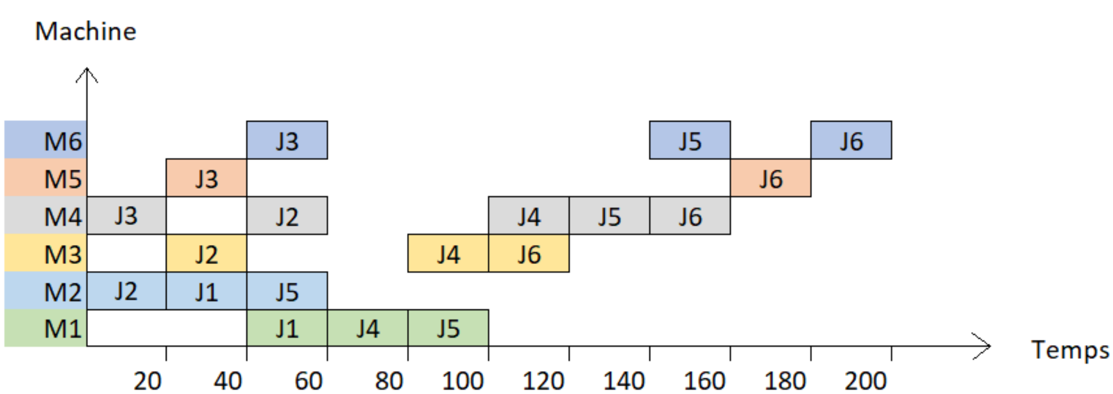

Figure 8. Gantt chart.

T=200 seconds.

C) M2, M1, M3, M4, M5, M6

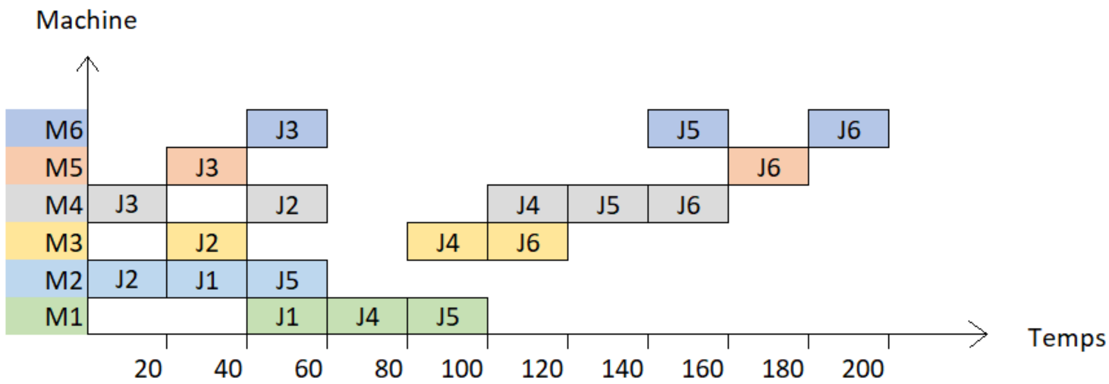

Figure 9. Gantt chart.

T= 200 seconds

D) M1, M2, M6, M4, M5, M3

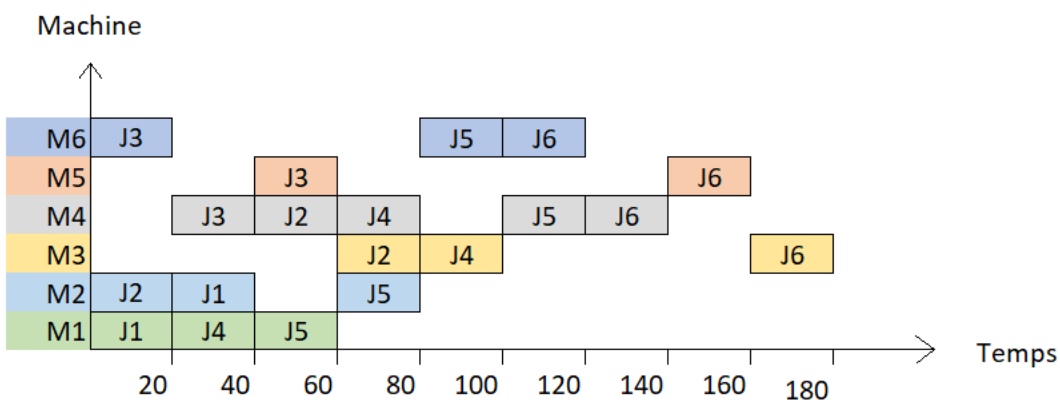

Figure 10. Gantt chart.

T= 180 seconds

E) M1, M2, M3, M5, M4, M6

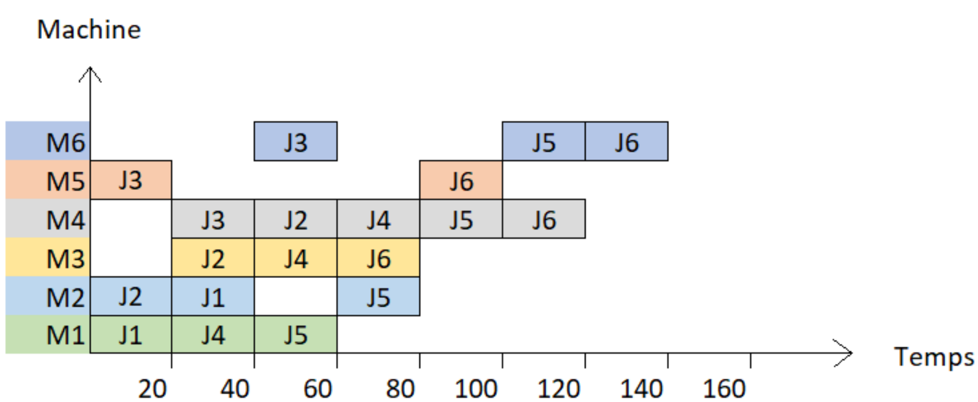

Figure 11. Gantt chart.

T= 140 seconds

F) M1, M2, M3, M4, M6, M5

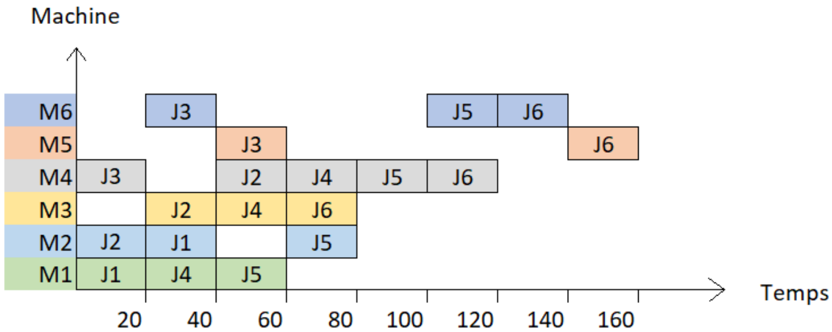

Figure 12. Gantt chart.

T=160 seconds
It is found that the population E which has a minimum fitness value of 140 seconds compared to the other populations.
We select the following parents to carry out our study:
A) M1, M2, M3, M4, M5, M6
P1= O11, O14, O15, O22, O21 O25 O32 O34 O36 O43 O42 O44 O45 O46 O53 O56 O63 O65 O66
E) M1, M2, M3, M5, M4, M6
P2= O11, O14, O15, O22, O21 O25, O32 O34 O36 O53 O56 O43 O42 O44 O45 O46 O63 O65 O66

Table 4. table of sequences matrix.

	M1	M2	M3	M4	M5	M6
J1	1	2				
J2		1	1	2		
J3				1	1	1
J4	2		2	3		
J5	3	3		4		2
J6			3	5	2	3

Crossover:
P1= O11, O14, O15, O22, O21 O25 O32 O34 O36 O43 O42 O44 O45 O46 O53 O56 O63 O65 O66
P2= O11, O14, O15, O22, O21 O25, O32 O34 O36 O53 O56 O43 O42 O44 O45 O46 O63 O65 O66
P1= O11, O14, O15, O22, O21 O25 O32 O34 O36 O53 O42 O44 O45 O46 O53 O56 O63 O65 O66
P2= O11, O14, O15, O22, O21 O25, O32 O34 O36 O43 O56 O43 O42 O44 O45 O46 O63 O65 O66

Mutation:
The operations that undergo a mutation are the following:

Parent P1:
O36□ O43 / O46□ O51/ O53□O52/ O66□ O16
Parent P2:
O56□ O51 / O43□ O63 / O45□ O16 / O63□ O62 / O46□ O53
OS1= O11, O14, O15, O22, O21 O25 O32 O34 O43 O53 O42 O44 O45 O51 O52 O56 O63 O65 O16
OS2= O11, O14, O15, O22, O21 O25, O32 O34 O36 O43 O51 O63 O42 O44 O16 O53 O62 O65 O66

The new Gantt chart:

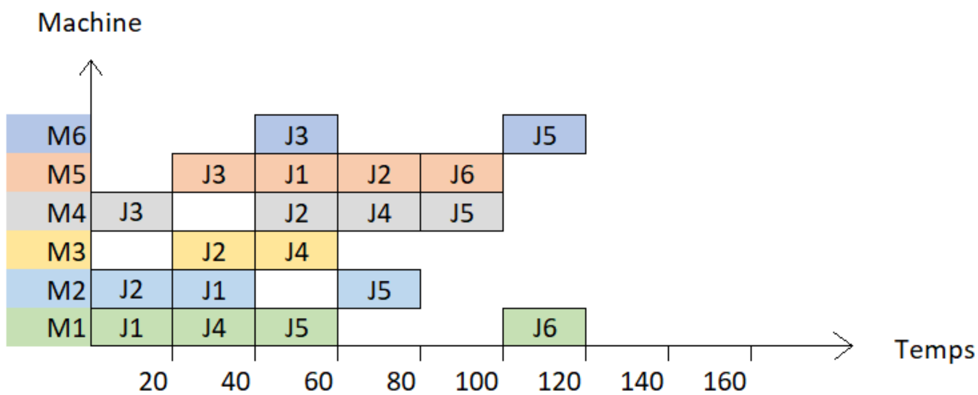

Figure 13. Gantt chart.

T= 120 seconds

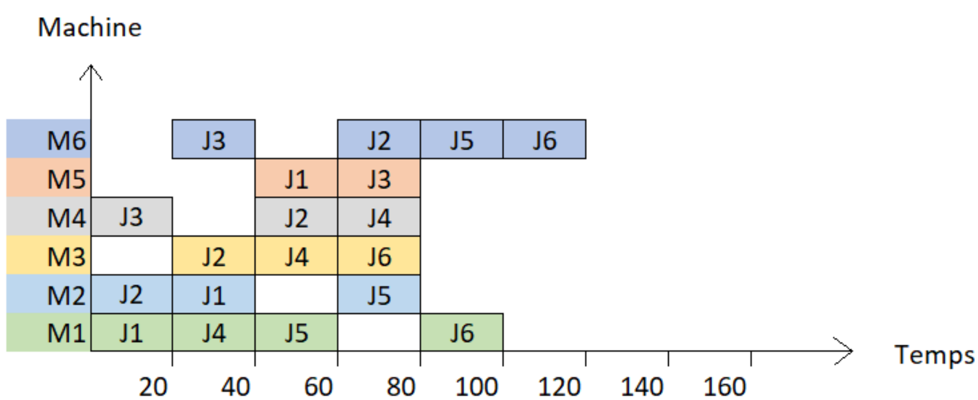

Figure 14. Gantt chart.

T= 120 seconds

6. Results and Discussion

- So the value that corresponds to the fitness function of the new parents is minimal.
- We retain this solution which represents the most optimal sequence for manufacturing a product.
- So the genetic algorithms give good results which are real and which optimize the problems classified as NP difficult.
- Genetic algorithms are mainly based on randomness; the more we choose to start the iteration process by using random methods, the more we will cover the majority of solutions found in the search space and the more we increase the chances of get the global optimal solution.

7. Conclusion

This article details the work done by software and intelligent programs when finding the most optimal sequence in the case of the job shop scheduling problem in a manual way to understand the execution process, this work will allow researchers to move on to more complicated subjects, in particular the requirement of certain constraints in the case of the subject of job shop scheduling problem.
Genetic algorithms allow us to feel closely the usefulness of mathematics and programming in everyday life.

8. References

[1]: Ke Shen, Toon De Pessemier, Luc Martens, Wout Joseph. A parallel genetic algorithm for multi-objective flexible flowshop scheduling in pasta manufacturing. Computers & Industrial Engineering 161 (2021) 107659.

[2]: Sarahí Báez, Francisco Angel-Bello, Ada Alvarez, Belén Melián-Batista. A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times. Computers & Industrial Engineering 131 (2019) 295–305.

[3]: Chen-Fu Chien, Yu-Bin Lan. Agent-based approach integrating deep reinforcement learning and hybrid genetic algorithm for dynamic scheduling for Industry 3.5 smart production. Computers & Industrial Engineering 162 (2021) 107782.

[4]: Jerry Swan, Steven Adriaensen, Alexander E. I. Brownlee, Kevin Hammond, Colin G. Johnson, Ahmed Kheiri, Faustyna Krawiec, J. J. Merelo, Leandro L. Minku, Ender Özcan, Gisele L. Pappa, Pablo García-Sánchez, Kenneth Sörensen, Stefan Voß , Markus Wagner, David R. White. Metaheuristics "In the Large". European Journal of Operational Research 297 (2022) 393–406.

[5]: Lawrence M. Wein, Philippe B. Chevalier. A Broader View of the Job-Shop Scheduling Problem. Published Online:1 Jul 1992/ https://doi.org/10.1287/mnsc.38.7.1018

[6]: D.J. Hoitomt, P.B. Luh, K.R. Pattipati. A practical approach to job-shop scheduling problems Published in: IEEE Transactions on Robotics and Automation (Volume: 9, Issue: 1, Feb 1993)

[7]: Runwei Cheng, Mitsuo Gen, Yasuhiro Tsujimura. Computers & Industrial Engineering Volume 30, Issue 4, September 1996, Pages 983-997.

[8]: Kumar Ritwik, Sankha Deb, A genetic algorithm-based approach for optimization of scheduling in job shop environment,Journal of Advanced Manufacturing Systems 10(2):223-240 ,2011. DOI:10.1142/S0219686711002235

[9]: DataDrivenInvestor, Available: https://medium.datadriveninvestor.com/population-initialization-in-genetic-algorithms-ddb037da6773 on June 21, 2022.

[10]: Amir Ghasemi, Amir Ashoori, CathalHeavey. Evolutionary Learning Based Simulation Optimization for Stochastic Job Shop Scheduling Problems. Applied Soft Computing
Volume 106, July 2021, 107309

[11]: KarimTamssaouet, StéphaneDauzère-Pérès, SebastianKnopp, AbdoulBitar, ClaudeYugma. Multiobjective optimization for complex flexible job-shop scheduling problems. European Journal of Operational Research
Volume 296, Issue 1, 1 January 2022, Pages 87-100.

[12]: T. Loukil, J. Teghem, D. Tuyttens. Solving multi-objective production scheduling problems using metaheuristics. European Journal of Operational Research 161 (2005) 42–61.

[13]: NarayanBehera, Analysis of microarray gene expression data using information theory and stochastic algorithm. Handbook of Statistics Volume 43, 2020, Pages 349-378.

[14]: Brad L. Miller, David E . Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise. Complex Systems 9 (1995) 193- 212.

[15]: Frederick M. Cohan, Can Uniform Selection Retard Random Genetic Divergence Between Isolated Conspecific Populations? Society for the Study of Evolution. Vol. 38, No. 3 (May, 1984), pp. 495-504

World Scientific Publishing Company

8. Biographies

Sahar HABBADI: Industrial engineer graduated from ENSAM-MEKNES promotion 2015 and PhD student, she works on the subject of thesis: optimization of production by metaheuristic methods.

Brahim HERROU: Mechanic engineer graduated from ENSEM-Casablanca, Option Quality control and Maintenance, CQM, promotion 1994 and Ph.D. Professor of Industrial Engineering, mechanics, quality and maintenance in the Department of Industrial Engineering at the Faculty of Science and Technology (FST) of the University Sidi Mohammed Ben Abdellah-Fès-Morocco. He has worked on several subjects in the field of quality, maintenance, logistics as well as artificial intelligence.

Souhail SEKKAT: Mechanic engineer graduated from ENSEM-Casablanca, promotion 1994 and Phd Professor of Industrial Engineering and maintenance in the Department of Industrial Engineering at the Ecole Nationale Supérieure d'Arts et Métiers (ENSAM- Meknes) of the University Moulay Ismail since 1997.
Since 2008, he has been working on a doctoral thesis within the LISER (Computer Science Laboratory for Renewable Energies and Systems.). He developed a formal approach to the design and implementation of a Performance Indicator System (SIP), starting from the industrial system to end up with the machine code.