An Effective Genetic Algorithm for Job Shop Scheduling with Fuzzy Degree of Satisfaction

Prof. Dr. Akeela M.Al-Atroshi [1] Dr. Sama T. Azez [2] Baydaa S.Bhnam [3]

**[1]Department of Computer Science, University of Dohuk,
Dohuk, +964, Iraq
[2]Department of Software Engineering, Mosul University, College of Computer sciences and Mathematics
Mosul, Ninawa +964, Iraq
[3]Department of Software Engineering, Mosul University, College of Computer sciences and Mathematics
Mosul, Ninawa +964, Iraq**

Abstract

The present study suggests a hybrid new fuzzy-genetic algorithm for solving the job shop scheduling problem. Traditional scheduling method does not keep pace with the requirements of the development in the field of manufacturing. Therefore, the current proposed algorithm offers a hybrid intelligent solution between two approaches: genetic algorithm to arrange the jobs randomly, and applied fuzzy logic to build objective function for genetic algorithm. All these are to find optimal degree of satisfaction that achieves optimal chain to schedule a production order using function max(min(satisfaction degree)). The present study includes the modeling of the objective function and adopting a fuzzy logic to solve the issue of scheduling production orders. The matlab is used for Programmable fuzzy Logic, whereas the C++ is used for programming the genetic algorithm with mechanism for linking C++ and matlab. Finally, the algorithm is tested on instances of 10 working procedures (jobs) and 3 machines. The result shows that the hybrid fuzzy-genetic algorithm has been successfully applied to the job shop scheduling problems.

Keywords: Fuzzy Logic, Genetic Algorithm, Job-shop Scheduling, Degree of Satisfaction.

1. Introduction

Scheduling is widely defined as the process of assigning a set of jobs to resources over a period of time . Effective scheduling plays a very important role in today's competitive manufacturing environment. Performance criteria such as machine utilization, manufacturing lead times, inventory costs, meeting due dates, customer satisfaction, and quality of products are all dependent on how efficiently the jobs are scheduled in the system. Hence, it becomes increasingly important to develop effective scheduling approaches that help in achieving the desired objectives[4,7] .

The scheduling and planning a production order have an important role in the manufacturing system. The diversity of products, increased number of orders, the increased number and size of workshops and expansion of factories have made the issue of scheduling production orders more complicated, hence the traditional methods of optimization are unable to solve them [7,10]. Genetic algorithms are stochastic global optimization methods inspired by the biological mechanisms of evolution and heredity, which have been widely used for scheduling problem in recent years [8].

With respect to related studies, one study suggested a genetic algorithm for scheduling a number of jobs on several machines. Each chromosome is divided into two parts: the first is related with machines and the second is for jobs. Fuzzy logic functions are utilized to calculate the degree of completion time and due date [1]. In the same direction [2] proposed two approaches: Jobs Sequencing List Oriented Genetic Algorithm and Operations Machines Coding. The Oriented Genetic Algorithm has been implemented and compared for solving the Job-Shop scheduling problem. Each approach has its own coding, evaluation function, crossovers and mutations applicable in Job-Shop scheduling problem to minimize the workload of the most loaded machine and the total workload of the machines. Jobs Sequencing List Oriented Genetic Algorithm has been found to be the best out of two approaches to minimize the objectives. While [11] improved genetic algorithm to solve the job shop scheduling problems (JSP) through proposing Taguchi-based genetic algorithm. The TBGA combines the powerful global exploration capabilities of conventional genetic algorithm (GA) with the Taguchi method that exploits optimal offspring.

In this research, the job shop problem is considered. The order production contains n jobs J1,..., Jn with given release dates r1,..., rn and due dates d1,...,dn to be scheduled on a set of m machines M1,...,Mm . Each job j=1,…,n consists of a set of operations determined by a process plan that specifies precedence constraints imposed on the operations. Each order contains N products (J1,

J2, ..., Jn) on M of the machines (M1, M2, .. Mm), and there are several probabilities to arrange business Ji and pass it orderly on the machines. Thus, the research domain is very wide and the intervention may be in context of NP-hard problem for the possibility to generate several different sequences of jobs per order.
This research has used genetic algorithm to solve the problem of selecting an optimum arrangement for jobs. But the problem faced by researchers is to build a special objective function for genetic algorithm. On this basis, a fuzzy hybrid degree of satisfaction has been adopted to be used as an objective function of hybrid fuzzy-genetic algorithm which is proposed in this research, taking into consideration that the system handles open number of jobs and machines. The present study aims to: address the problem of compatibility and integration of hybridization for fuzzy logic and genetic algorithm programmatically and philosophically to be as a new and innovative idea to schedule business.
The large-scale projects often face critical circumstances that prevent accomplishment time, so the research depends on max(min (sat. degree)) as an objective function.
The philosophy of the best worst in the design of objective function is adopted for fuzzy-genetic algorithm to achieve the proposed idea and accordingly get optimal scheduling of jobs. The lowest degree of satisfaction in this series is larger than the lowest level of satisfaction of any other chain within the search space .

2. Providing the basic requirements of the proposed algorithm

The proposed system requires providing of several requirements to support the production manager to make decisions about releasing a production order. The requirements are as follows:

1. Preparation of production order file (order.dat):
The structure of order.dat file represents the order details in terms of the product with code, release_date (first r_inf and final r_sub) and the due_date (first d_inf and final d_sub) for each product in the order. The product passes through several machines (operations), so the processing time is required for each product on all the machines in the production line (routing). Even if we assume that the Pi j represents the processing time for the product i on the machine j, where i = 1 ... n and j = 1 ... m and n represents the number of products in the order and m represents the number of machines in the workshop. The three processing times (optimistic p_a most likely p_b and pessimistic p_c) for each m machines determined in order.dat file. So that, the file structure of order.dat to the process of initializing a single product on the two machines is shown in table (1).

Table 1: Structure of order.dat file

Job	*Machine 1*			*Machine ...*			*Machine m*			*Release_Date*		*Due_Date*	
code	pa	pb	pc	pa	pb	pc	Pa	pb	pc	r_inf	r_sub	d_inf	d_sub
a	5	3	2	...			7	4	6	2	4	3	6
...	...			...			...			...		...	

2. Preparations of Proposed Genetic Algorithm Parameters: Table(2) illustrates the parameters of genetic algorithm, as identified by the researchers.

Table 2: Parameters of proposed GA

Population Size (pop_size)	6
No. Of Generations	20
Chromosome Size	No. of Jobs in the Order = N
Selection Method	Binary Select Method
Crossover Method	Partially Mapped Crossover PMX
Mutation Method	Double simple swapping
Crossover Probability	100%
Mutation Probability	100%

3. Preparations of Fuzzy Logic Parameters:
All Parameters are initialized in C++ language, release_date and due_date are read directly from the file order.dat and all these parameters are transferred to MATLAB. The following equations show how the coefficient of the time accomplishment is calculated [1,9,13] .

$$C_{1,1} = R_1 + P_{1,1} \quad (1)$$

$$C_{i,1} = \max(R_i, C_{i-1,1}) + P_{i,1} \quad ; for \quad i=2,\ldots,n \quad (2)$$

$$C_{1,j} = C_{1,j-1} + P_{1,j} \quad ; for \quad j=2,\ldots,m \quad (3)$$

$$C_{i,j} = \max(C_{i-1,j}, C_{i,j-1}) + P_{i,j} \quad ; for \quad i=2,\ldots,n \; ; for \quad j=2,\ldots,m \quad (4)$$

4. The general outline of the proposed hybrid algorithm:
Genetic algorithm was proposed to schedule production orders (which are programmed using C++) and, choose Fuzzy logic, which is programmed using MATLAB) as an fuzzy objective function to find optimal sequence for an order products. Figure (1) illustrates the general outline of the proposed hybrid algorithm.

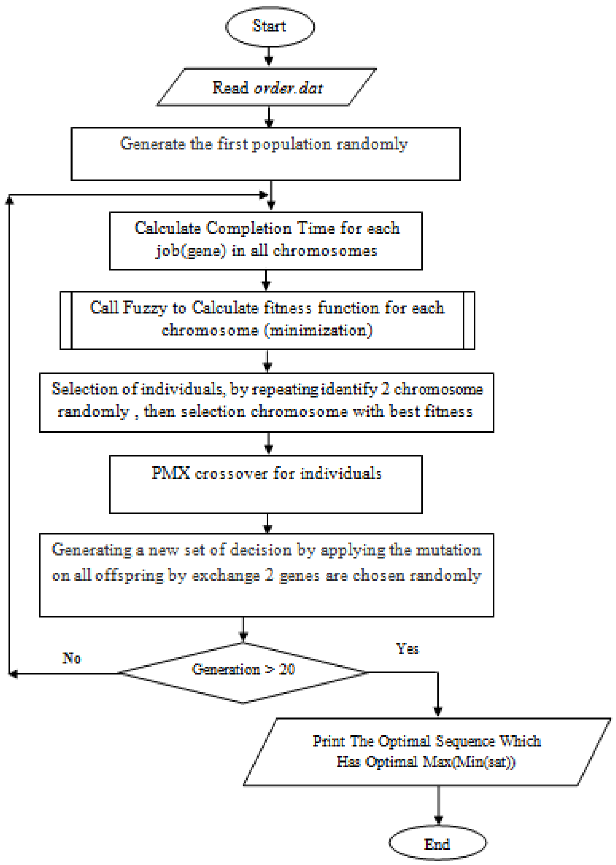

Fig. 1: General flowchart of fuzzy-genetic algorithm

3. Designing a hybrid algorithm and scheduling production orders

Hybrid algorithm was designed using the C++ language , and has been implemented in matlab environment. The implementation strategy is made according to the following sequent steps:

1. Reading the order.dat file: as shown in the table (3), where order of ten jobs is performed on three machines.
No._of_jobs= 10
No._of_machines= 3

Table 3 : Structure and contents of the data file order.dat.

Jobs	Release_date		Process_ time									Due_date	
			M1			M2			M3				
	r-inf	r-sub	Pa	Pc	Pd	Pa	Pc	Pd	Pa	Pc	Pd	d-inf	d-sub
a	2	3	2	3	4	1	2	5	4	5	6	10	15
b	12	16	3	6	7	2	4	5	1	3	4	50	90
c	1	2	5	6	7	3	4	6	7	9	11	11	50
d	6	12	4	6	7	2	5	6	3	6	8	50	80
e	5	9	8	10	11	5	6	7	4	5	8	30	62
f	8	12	9	11	13	7	9	10	2	4	6	41	84
g	4	5	3	6	7	10	12	13	6	7	9	24	78
h	10	13	4	7	9	9	11	13	6	8	9	35	95
i	3	5	8	9	11	10	12	13	5	6	7	18	85
j	3	7	7	9	10	12	13	15	9	11	12	20	70

2. Applying the genetic algorithm: the population size under study is six chromosomes, so the first generation is initialized randomly and ten jobs are ordered randomly. Table (4) shows the first generation as stated in one of implementation cases. Each gene represents the product code in the order file.

Table 4 : The chromosomes of first generation.

1	b	g	d	a	j	f	h	i	c	e
2	j	b	a	i	c	e	h	d	g	f
3	h	e	j	b	d	f	a	i	c	g
4	f	d	a	e	g	b	i	j	c	h
5	a	f	c	j	e	b	g	d	I	h
6	b	a	i	d	e	g	h	j	C	f

3.Callingthe fuzzy algorithm, which is designed to be fuzzy objective function for the proposed genetic algorithm, so, at the same direction, the research used triangular fuzzy release date , due date and comp_time respectively. As shown in figure(2) and (3), fuzzy release date $\tilde{R}$, denoted by a doublet (rinf , rsup) is described by $\mu_{\tilde{R}}$(rinf)=0 , $\mu_{\tilde{R}}$(rsup)=1. Fuzzy due date $\tilde{D}$, denoted by a doublet (dinf , dsup) is described by $\mu_{\tilde{D}}$(dinf)=0 , $\mu_{\tilde{D}}$(dsup)=1 [13],and fuzzy comp_time $\tilde{C}$ denoted by (c1 , c2 , c3) , where $\mu_{\tilde{C}}$(c1)=0,$\mu_{\tilde{C}}$ (c2)=1 and $\mu_{\tilde{C}}$(c3)=0.

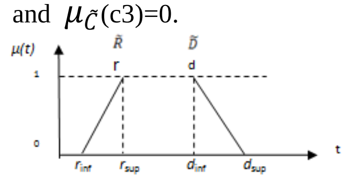

Fig. 2 : Fuzzy release date and fuzzy due date.

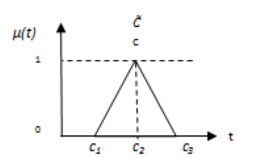

Fig. 3 : Fuzzy completion time.

Three fuzziness criteria are computed and retrieved for each job (gene) in all chromosomes: the objective function represented in a degree of satisfaction SG and the beginning and end time (st , et) respectively is determined as follows:

```
Repeat for each job (gene) in all chromosomes
    if  dinf  = c2 or dinf  = c3 or c2 < dinf < c3
          SG=1
          St = round ( dinf - rsup )/2
          et = dinf
    else
          calculate SG , st and et by :
```

- Sharing point between $\tilde{D}$ & $\tilde{C}$ $(\overline{d \quad d_{sup}} \perp \overline{c_1 \quad c})$ will determine et on x-axis and SG on y-axis.
- The crossing point between $\tilde{R}$ and SG $(\overline{r_{inf} \quad r} \perp SG)$ will determine st on x-axis

```
    Endif
End repeat
```

Table(5) shows the values of fuzziness objective function for the fifth chromosome in the first generation .

Table 5 : Results fuzzy objective function for one chromosome.

jobs	release_date		due_date		Ct			SG	St	et
	r_{inf}	r_{sup}	d_{inf}	d_{sup}	c_1	c_2	c_3			
a	2	3	10	15	9	13	18	0.6667	2.666667	11.666667
f	8	12	41	84	26	36	41	1.00	15.00	41.0
c	1	2	11	50	34	45	52	0.320	1.320	37.52
j	3	7	20	70	50	62	69	0.32258	4.290323	53.870968
e	5	9	30	62	54	67	77	0.1778	5.711111	56.311111
b	12	16	50	90	55	70	81	0.63636	14.545455	64.545456
g	4	5	24	78	64	80	91	0.20	4.200000	67.199997
d	6	11	50	80	67	86	99	0.26531	7.326530	72.040817
i	3	5	18	85	75	96	108	**0.11364**	3.227273	77.38636
h	10	13	35	95	85	109	23	0.11905	10.357142	87.85714

4. Calculating the first part of the objective function min (satisfaction degree),which represents the intermediate results, which are less degree of satisfaction for each chromosome of the current generation. The less degree of satisfaction for each chromosome is stored in crom_mdl_rslt(i), where i= 1,...,6. as shown in table(5). Note that the less degree of satisfaction for the fifth chromosome in the first generation is 0.11364. It is worth mentioning that the SD for second chromosome = 0, so it has less degree of satisfaction among the first generation chromosomes. Therefore, crom_mdl_rslt(2)= 0 and crom_mdl_rslt(5) = 0.11364.
5. Creating the second part of the objective function max(min(satisfaction degree), which represents the final outcome of the optimal degree of satisfaction in the current generation by choosing the chromosome that achieves the highest (best) degree of satisfaction among chromosomes in intermediate results (worst) crom_mdl_rslt. This means selecting the best among the worst max(crom_mdl_rslt) and then storing in the vector gnr_fnl_rslt(i), where i=1,..,20. And, fifth chromosome is the best worst in the first generation and so the gnr_fnl_rslt(1)='afcjebgdih'.
6.Two parents are selected randomly from the current generation, then one of the parents is chosen according to the best degree of satisfaction and this process is repeated until the selection of all parents is done.
7. Crossover process between two consecutive parents to create offspring, on the condition that the gene is not repeated (duplicated) more than once per chromosome by applying the PMX method.
8.Mutation process for all offspring through exchanging two genes chosen randomly.
9.Repeating the previous steps, starting from the third step for the establishment of 20 generations.
10.After the completion of the generating of all 20 populations, finding the best chromosome within vector max(gnr_fnl_rslt(i)), where i= 1,…,20, which is the optimal solution to the scheduling of the order.

4. Implementation of Hybrid Fuzz-genetic algorithm

The hybrid algorithm has been applied to a production order. The details are shown in the table(3). The integrated program for hybrid algorithm (main.cpp) has been implemented and the results as follows:

A: The intermediate results of implementation and the final optimal scheduling result for order.dat are described below:

GENETIC ALG. WITH FUZZY FOR JOB SCHEDULING

No. Of Machines = 3 No. Of jobs = 10

The Minimum Satisfaction Degree Is : 0.098039 for 1st 20 generations

job	Release_Date		Due_Date		Completion_time			Satisfaction_degree	Start	End
a	2	3	10	15	9	13	18	0.666667	2.666667	11.666667
j	3	7	20	70	32	40	44	0.655172	5.620690	37.241379
h	10	13	35	95	38	48	54	0.814286	12.442857	46.142857
c	1	2	11	50	45	57	65	**0.098039**	1.098039	46.176472
b	12	16	50	90	46	60	69	0.814815	15.259259	57.407406
e	5	9	30	62	50	65	77	0.255319	6.021276	53.829788
d	6	11	50	80	53	71	85	0.562500	8.812500	63.125000
i	3	5	18	85	59	78	92	0.302326	3.604651	64.744186
g	4	5	24	78	70	91	104	0.106667	4.106667	72.239998
f	8	12	41	84	73	97	111	0.164179	8.656716	76.940300

(0 Stop / 1 Continue To Generate another 20 Generation). Enter (0 / 1)?: 1

The Minimum Satisfaction Degree Is : 0.113636 for 2nd 20 generations

job	Release_Date		Due_Date		Completion_time			Satisfaction_degree	Start	End
a	2	3	10	15	9	13	18	0.666667	2.666667	11.666667
f	8	12	41	84	26	36	41	**1.000000**	15.000000	41.000000
c	1	2	11	50	34	45	52	0.320000	1.320000	37.520000
j	3	7	20	70	50	62	69	0.322581	4.290323	53.870968
e	5	9	30	62	54	67	77	0.177778	5.711111	56.311111
b	12	16	50	90	55	70	81	0.636364	14.545455	64.545456
g	4	5	24	78	64	80	91	0.200000	4.200000	67.199997
d	6	11	50	80	67	86	99	0.265306	7.326530	72.040817
i	3	5	18	85	75	96	108	**0.113636**	3.227273	**77.386360**
h	10	13	35	95	85	109	123	0.119048	10.357142	87.857140

(0 Stop / 1 Continue To Generate another 20 Generation). Enter (0 / 1)?: 1

The Minimum Satisfaction Degree Is : 0.170455 for 3rd 20 generations

job	Release_Date		Due_Date		Completion_time			Satisfaction_degree	Start	End
a	2	3	10	15	9	13	18	0.666667	2.666667	11.666667
j	3	7	20	70	32	40	44	0.655172	5.620690	37.241379
f	8	12	41	84	34	44	50	0.943396	11.773585	43.433964
c	1	2	11	50	41	53	61	0.176471	1.176471	43.117645
b	12	16	50	90	42	56	65	0.888889	15.555555	54.444443
d	6	11	50	80	45	62	73	0.744681	9.723404	57.659573
e	5	9	30	62	49	67	81	0.260000	6.040000	53.680000
g	4	5	24	78	61	80	91	0.232877	4.232877	65.424660
i	3	5	18	85	70	91	102	**0.170455**	3.340909	73.579544
h	10	13	35	95	80	104	117	0.178571	10.535714	84.285713

(0 Stop / 1 Continue To Generate another 20 Generation). Enter (0 / 1)?: 1

The Minimum Satisfaction Degree Is : 0.203125 for 4th 20 generations

job	Release_Date		Due_Date		Completion_time			Satisfaction_degree	Start	End
a	2	3	10	15	9	13	18	0.666667	2.666667	11.666667
j	3	7	20	70	32	40	44	0.655172	5.620690	37.241379
c	1	2	11	50	39	49	55	0.224490	1.224490	41.244900
h	10	13	35	95	45	57	64	0.694444	12.083333	53.333332
e	5	9	30	62	49	62	72	0.288889	6.155556	52.755554
d	6	11	50	80	52	68	80	0.608696	9.043478	61.739132
g	4	5	24	78	58	75	89	0.281690	4.281690	62.788731
i	3	5	18	85	67	85	97	0.211765	3.423529	70.811768
f	8	12	41	84	71	92	106	**0.203125**	8.812500	75.265625
b	12	16	50	90	72	95	110	0.285714	13.142858	78.571426

(0 Stop / 1 Continue To Generate another 20 Generation). Enter (0 / 1)?: 1

The Minimum Satisfaction Degree Is : **0.265060** for 5th 20 generations **is optimal SD (final result)**

job	Release_Date	Due_Date	Completion_time	Satisfaction_degree	Start	End
a	2 3	10 15	9 13 18	0.666667	2.666667	11.666667
g	4 5	24 78	23 31 36	0.887097	4.887097	30.096775
c	1 2	11 50	30 40 47	0.408163	1.408163	34.081635
j	3 7	20 70	41 52 60	0.475410	4.901639	46.229507
e	5 9	30 62	45 57 68	0.386364	6.545455	49.636364
i	3 5	18 85	52 65 75	0.412500	3.825000	57.362499
d	6 11	50 80	55 71 83	0.543478	8.717391	63.695652
f	8 12	41 84	58 77 90	0.419355	9.677420	65.967743
b	12 16	50 90	59 80 94	0.508197	14.032787	69.672134
h	10 13	35 95	73 96 111	**0.265060**	10.795180	79.096382

(0 Stop / 1 Continue To Generate another 20 Generation). Enter (0 / 1)?: 0

B. Figure (4) and (5) illustrate the Gantt charts for the worst (1st 20 generations) and optimal solution (5th 20 generation) respectively.

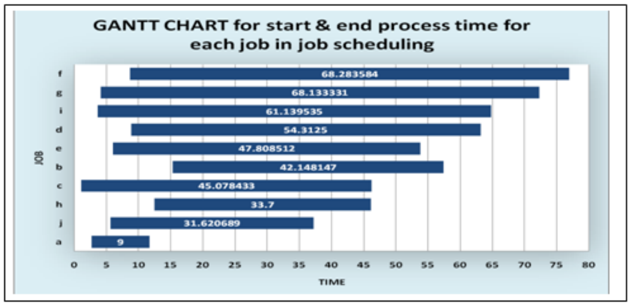

Fig. 4 : Gantt chart for the worst solution of scheduling order.

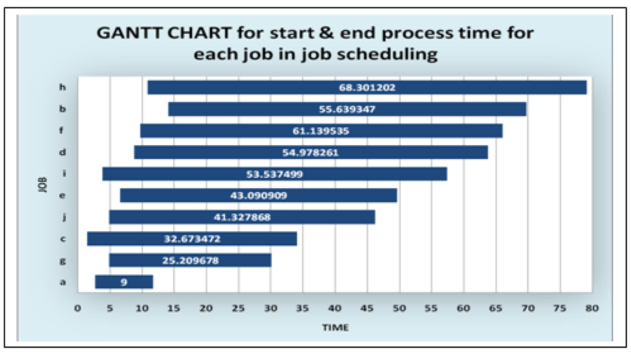

Fig. 5 : Gantt chart for the optimal solution of scheduling order.

C. All the results of implementation begin at the same time start(1'st job)= 2.666667 to all schedules, but the end time of the last job differs from one schedule to another Max(end) as shown in the table(6), which summarizes the result of implementation. Results in the schedule(2) show the highest degree of satisfaction(SD = 1), but the end time of last job in that scheduling is 87.857140 and this situation can be applied to scheduling 3 also, it can be said that it cannot rely Max (SD)as an objective function but its max(end) < max(end) for schedule 2. This undoubtedly supports the hypothesis of research based on a proposal objective function Max (Min (SD)), which is achieved by scheduling5. It is clear that the end time of the last jobs scheduling (time of the completion of the project) is78.571426 and thus has less time to complete the project (for more than 8 hours) compared to scheduling2. Consequently, these results give more flexibility to the technical manager of the project for making a wise decision supported by intelligent techniques on the selection of optimal and best scheduling for the company to complete the project (i.e. the choice between scheduling the first in the case of adopting min(max(end)) and recent in the case of adopting Max (Min (SD)).

Table 6 : A summary of the results of implementation

Running Step	Job Sequence	Min(S.D)	Max(S.D)	Max(end)
1	a j h c b e d i g f	0.098039	0.814815	76.940300
2	a f c j e b g d i h	0.113636	*1.00*	87.857140
3	a j f c b d e g i h	0.170455	*0.943396*	84.285713
4	a j c h e d g i f b	0.203125	0.694444	78.571426
5	**a g c j e i d f b h**	**0.265060**	**0.887097**	**79.096382**

D. The application results have confirmed that the time rate required to complete a single generation is three hundredth of a second (Elapsed time is 0.031 seconds). In other words, the time required to complete the 20 generations and show the best solution is 0.62 or 62 hundredth of a second.

E. The application results have also indicated that the best implementation schedule of the order is obtained through the first ten generations usually. Table(7) shows sequences as various order schedules include their own degree of satisfaction with those obtained during the application. Figure(6) shows optimal solutions chart for twenty generations.

Table 7 : Optimal scheduling for ten generations.

Generation	Optimal Sequence	Min. Sat. Degree
1	a j c h e d g i f b	0.203125
2	a c j e g b i d f h	0.195122
3	a j h c b e d i g f	0.098039
4	a j f c b d e g i h	0.170455
5	g a e i c j f b d h	0.00
6	a c d g e j b i f h	0.216867
7	**a g c j e i d f b h**	**0.265060**
8	a f c j e b g d i h	0.113636
9	a j c b e g d i f h	0.214286
10	a c g j f e i b d h	0.240964

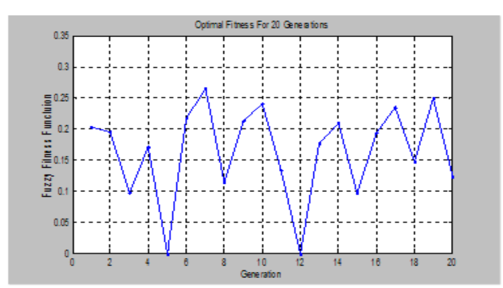

Fig. 6 : optimal solutions chart for twenty generation.

The criteria of fuzziness objective function for optimal solution is (agcjeidfbh) as illustrated in Figure(7).

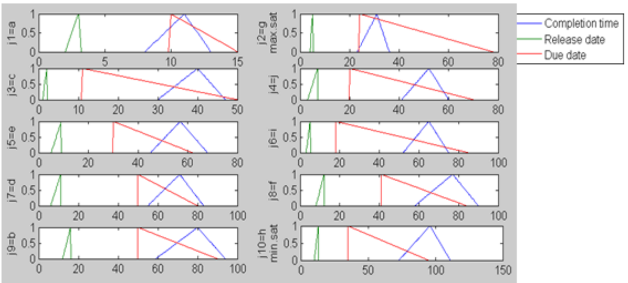

Fig. 7 : The results of the fuzziness objective function for optimal scheduling.

V. Conclusions and Future Works

The designed software has been implemented on an order that contains ten jobs passed on three machines in the job shop to be processed and delivered to the customer. The results were as follows:

1.The proposed system is characterized by specifications of two algorithms: (1)genetic characterized by robustness, reliability and generality [5] as well as (2) fuzzy logic through its ability to handle with incomplete information and sometimes ambiguous[6].

2.The results of the application on several orders confirmed that the best solution can be obtained in the first ten generations of the twentieth generations planned. In other words, best solution can be obtained in the first half of the execution time.

3.There are several traditional ways to resolve the issue of scheduling, but it turns out that the system designer gave special support to resolve the issue of scheduling in a contemporary, technical and intelligent manner, depending on optimization.

4.Hybrid system can be classified as a decision support system, which supports the managers to make their decisions on production scheduling correctly and based on modern and intelligent methods of hybrid.

5.The philosophy of the best worst adopted in the design of objective function genetic algorithm has made the issue of delivery of the order to the customer with the least possible delay in due date.

In the same direction, the current research suggests the concept of scheduling through:

1. Designing a hybrid system through neural networks with genetic algorithm once and with fuzzy logic once again to solve production scheduling problem.
2. Designing a hybrid fuzziness system to integrate the philosophy of the best worst and the philosophy of the highest degree of satisfaction to achieve compatibility between them, and then compared their results with those of the system designer in this current research.
3. Suggesting an objective function min(max(end)) instead of max(min(SD)), especially in the absence of critical conditions for the project.

References

[1] C. Fayad, And S.Petrovic," A Fuzzy Genetic Algorithm for Real-World Job-Shop Scheduling", in M.Ali, F.Esposito, (Eds.), Innovations in Applied Artificial Intelligence,Lecture Notes in Artificial Intelligence 3533, Springer, 2005, pp. 524-533.

[2] A.Kumar1, and R., Singh ," Performance Of Genetic Algorithms For Solving Flexible Job-Shop Scheduling Problem", International Journal of Information Technology and Knowledge Management, vol. 4, No. 1,2011,pp. 105-108.

[3] D.Lei, "Solving fuzzy job shop scheduling problems using random key genetic algorithm", International Journal of Advanced Manufacturing Technology , vol.49,2010,pp.253–262.

[4] H.Metta, "Adaptive, multi-objective job shop scheduling using genetic algorithms", Master's Theses, University of Kentucky,2008.

[5] H.Nanvala," Use of Genetic algorithm based approaches in scheduling of FMS:A Review", International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462,Vol. 3 No. 3,2011,pp. 1936- 1942.

[6] H.Nanvala, "Use of Fuzzy Logic Approaches in Scheduling of FMS: A Review", International Journal on Computer Science and Engineering (IJCSE),ISSN:0975-97,Vol.3,No.4, 2011,pp.1734-1739.

[7] Z.Othman,K.Subari, And N.Morad,"Job Shop Scheduling With Alternative Machines Using Genetic Algorithms",Jurnal Teknologi,41(D) , 2007,pp. 67-78.

[8] M.Omar, A.Baharum, Y.Hasan," A job-shop scheduling problem (jssp) using genetic algorithm (GA)". Proceedings of The 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications. Vol. IV: Computer Sciences & Applications. 13 – 15 June 2006, Penang, Malaysia. pp: 39 – 45.

[9] S.Petrovic, and C.Fayad, "A Genetic Algorithm for Job shop Scheduling with Load Balancing", in AI 2005, Lecture Notes in Artificial Intelligence 3809 edited by S.Zhang and R.Jarvis, pp.339-348.

[10] K.Raja, R. Saravanan, And V.Selladurai, "Mult-Pbjective Optimization Of Parallel Machine Scheduling (PMS) Using Genetic Algorithm And Fuzzy Logic Approach" IE(I)Journal-PR,vol. 87, 2008,pp. 26-31.

[11] J.T. Tsai& others" An improved genetic algorithm for job-shop scheduling problems using Taguchi-based crossover", International Journal of Advanced Manufacturing Technololgy,2008 .

[12] M.Sakawa, and R.Kubota, ,"Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms".European Journal of Operations Research, vol.120,2000, pp. 393-407.

[13] X.Song, and S.Petrovic,"Handling fuzzy constraints in flow shop problem", in B.DeBaets, J.Fodor, D.Radojevic (Eds.), Proceedings of the 9th Meeting of the EURO Working group on Fuzzy Sets–EUROFUSE 2005, Belgrade, Serbia & Montenegro, June 15-18,2005, pp. 235-244.