Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job Shop Scheduling Problems

Aakash Bhatia

I. OBJECTIVE

The paper presents two approaches to solve jointly the assignment and job-shop scheduling problems (with total or partial flexibility).

<u>First Approach:</u> Approach by Localization (AL), i.e. by a given set of assignment procedures it makes it possible to solve the problem of resource allocation and build an assignment model (assignment schemata).

<u>Second Approach:</u> The second approach is an evolutionary one which is controlled by the assignment model (generated by the first approach). In this approach, a set of population is chosen randomly from a set that is generated by assignment schemata, onto which the author applies advanced genetic manipulations in order to enhance the quality of the solution.

In both the approaches, the Kacem (2002) aims to minimize the overall completion time (makespan) and the total workload of the machines.

II. PROBLEM FORMULATION

Job shop scheduling problem aims to schedule a certain set of operations that are a part of a particular job to a certain set of machines. This is done on the basis of the number of resources the operations require and the amount of time one machine takes to complete the operation. Due to technological differences, we assume different machines will take different amount of time to complete the same operation.

The problem that the author is focusing on is execution of N jobs on M machines. Each job j represents a number of nj continuous operations. The execution of each operation i of a job j (noted Oi,j) requires one resource or machine selected from a set of available machines.

The author implements the job shop scheduling problem for two cases:

(i) Case of total flexibility: Any operation can be assigned to any machine.

(ii) Case of partial flexibility: Some operations (at least one) cannot be assigned to some machines.

III. APPROACH BY LOCALIZATION

A) <u>TOTAL FLEXIBILITY</u>

Problem data set:

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	3	8	2	1
	O3,1	3	5	4	7
J2	O1,2	4	1	1	4
	O2,2	2	3	9	3
	O3,2	9	1	2	2
	O1,3	8	6	3	5
J3	O2,3	4	5	8	1

Table 1: Table D

First Approach: Approach by Localization

For total flexibility in a job shop scheduling problem the author has two ways in which he solves the above stated problem. One way involves applying the assignment algorithm (Figure 1) to the above tables and the other way brings in more diversification by randomly permuting the rows and randomly choosing the column at which one applies the assignment algorithm (Figure 2).

Inputs:

1. Table D
2. Initialize a new table S with same number of elements as table D
3. Copy elements of D in a new table D1

Figure 1: Assignment Procedure

Inputs:

1. Table D
2. Initialize a new table S with same number of elements as table D
3. Copy elements of D in a new table D1

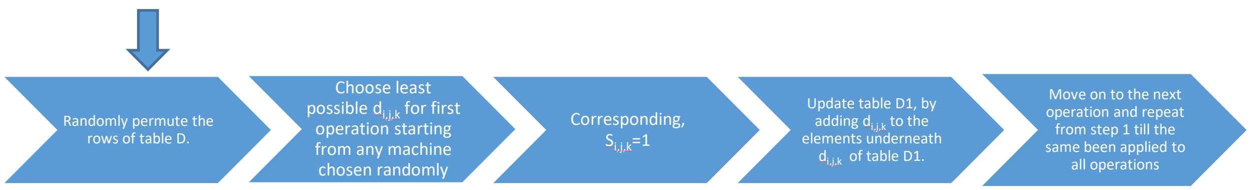


Figure 2: Assignment Procedure (With Diversification)

B) <u>CASE OF PARTIAL FLEXIBILITY</u>

In the case of partial flexibility, some operations are forbidden for some machines. In the input data in Table 2, the symbol X indicates that the assignment is impossible.

The author shows that the assignment procedure stated above can be easily implement for a partial flexibility type of a problem. He states that, as the assignment is designed in such a way that it avoids assignment of an operation to a machine for which the processing time is long. Thus, for each forbidden assignment, the author associates with it a very large fictitious processing time , which automatically makes the assignment procedure reject assignment of that operation to that machine. In this case he has taken that fictitious processing time to be 999.

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	5	3	5	3	3	X	10	9
	2,1	10	X	5	8	3	9	9	6
	3,1	X	10	X	5	6	2	4	5
J2	1,2	5	7	3	9	8	X	9	X
	2,2	X	8	5	2	6	7	10	9
	3,2	X	10	X	5	6	4	1	7
	4,2	10	8	9	6	4	7	X	X
J3	1,3	10	X	X	7	6	5	2	4
	2,3	X	10	6	4	8	9	10	X
	3,3,	1	4	5	6	X	10	X	7
J4	1,4	3	1	6	5	9	7	8	4
	2,4	12	11	7	8	10	5	6	9
	3,4	4	6	2	10	3	9	5	7
J5	1,5	3	6	7	8	9	X	10	X
	2,5	10	X	7	4	9	8	6	X
	3,5	X	9	8	7	4	2	7	X
	4,5	11	9	X	6	7	5	3	6
J6	1,6	6	7	1	4	6	9	X	10
	2,6	11	X	9	9	9	7	6	4
	3,6	10	5	9	10	11	X	10	X
J7	1,7	5	4	2	6	7	X	10	X
	2,7	X	9	X	9	11	9	10	5
	3,7	X	8	9	3	8	6	X	10
J8	1,8	2	8	5	9	X	4	X	10
	2,8	7	4	7	8	9	X	10	X
	3,8	9	9	X	8	5	6	7	1
	4,8	9	X	3	7	1	5	8	X

Table 2: Table P

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	5	3	5	3	3	999	10	9
	2,1	10	999	5	8	3	9	9	6
	3,1	999	10	999	5	6	2	4	5
J2	1,2	5	7	3	9	8	999	9	999
	2,2	999	8	5	2	6	7	10	9
	3,2	999	10	999	5	6	4	1	7
	4,2	10	8	9	6	4	7	999	999
J3	1,3	10	999	999	7	6	5	2	4
	2,3	999	10	6	4	8	9	10	999
	3,3,	1	4	5	6	999	10	999	7
J4	1,4	3	1	6	5	9	7	8	4
	2,4	12	11	7	8	10	5	6	9
	3,4	4	6	2	10	3	9	5	7
J5	1,5	3	6	7	8	9	999	10	999
	2,5	10	999	7	4	9	8	6	999
	3,5	999	9	8	7	4	2	7	999
	4,5	11	9	999	6	7	5	3	6
J6	1,6	6	7	1	4	6	9	999	10
	2,6	11	999	9	9	9	7	6	4
	3,6	10	5	9	10	11	999	10	999
J7	1,7	5	4	2	6	7	999	10	999
	2,7	999	9	999	9	11	9	10	5
	3,7	999	8	9	3	8	6	999	10
J8	1,8	2	8	5	9	999	4	999	10
	2,8	7	4	7	8	9	999	10	999
	3,8	9	9	999	8	5	6	7	1
	4,8	9	999	3	7	1	5	8	999

Table 3: Equivalent of Table P

IV. TEST OF EFFICIENCY OF THE APPROACH BY LOCALIZATION

The author tests the efficiency of the approach of the assignment procedure (i.e. approach by localization) by calculating the value of makespan and workload of machine that each assignment amounted to. And these values are calculated by what the author calls as the ‘scheduling algorithm’ (Figure 3)

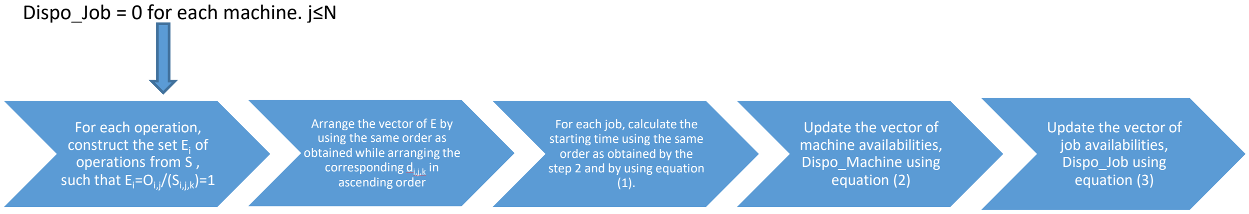
$$t_{I,j}=\text{Max}(\text{Dispo\_Machine}[k], \text{Dispo\_job}[j])\text{.......}(1)$$

$$\text{Dispo\_Machine}[k]=t_{i,j}+d_{i,j,k}\text{.......}(2)$$

$$\text{Dispo\_Job}[j]=t_{i,j}+d_{i,j,k}\text{.......}(3)$$

Inputs:

1. Table D, Table S
2. Initialize the vector of machine availabilities
   Dispo_Machines = 0 for each machine. k≤M
3. Initialize the vector of job availabilities
   Dispo_Job = 0 for each machine. j≤N



V. GENETIC ALGORITHM: NOTION OF ASSIGNMENT SCHEMATA

In this section, the author integrates the like of Approach by Localization to Genetic Algorithm, where GA being a population based heuristic needs an initial population onto which it performs the crossover and mutations to bring about an optimum solution. This initial population is generated by approach by localization using the assignment schemata (Schemata theorem). Stages involved in a typical GA:

<u>Stage I: Genesis - Generation of Initial Population</u>

Schemata theorem:

In the case of binary coding, a schemata is a chromosome model where some genes are fixed and other are free. For example, S=1*011*00 implies that all values except for position 2 and position 6 are fixed. And the value for position 2 and 6 can be either taken by 0 or 1.

The author claims that the schemata theory can make genetic algorithm more efficient and rapid by generating an initial population that is likely to produce a better quality result and with lesser number of iterations, hence reducing computational time.

Assignment Schemata

The assignment schemata uses the concept of the schemata theorem in a way that assignments that are forbidden are taken as zero and the assignments that are needed to obtain a solution closer to the optimum is taken as one and those assignment that are not critical and can take of value of zero or 1 are marked as ‘*’ .

The idea that the author proposes behind the assignment schemata is that it will help him control the GA. The reason being, the assignment schemata is created by observing 100 random assignment that are created from assignment shown in figure 2. And value of the assignment that approaches 1 most of the time (say more than 95 times) is equal to 1 in the assignment schemata and that assignment that does not appear much (say less than 3 times) is equal to zero in the assignment schemata. All other assignments are equal to * and can take any value zero or one. The author terms the quantifying factor (for example 3 and 95 stated previously) as function thresholds α and β.

Inputs:

1. Generate S^z random assignment from algorithm shown in figure 2, where $1 \le z \le cardinal(E)$
2. Initialize another table Sch, which is of the same size as assignment table S
3. Define thresholds α and β

$$S^{ch}_{i,j,k} = S^{ch}_{i,j,k} + \frac{S^{z}_{i,j,k}}{\mathrm{cardinal(E)}} \ldots. (4)$$

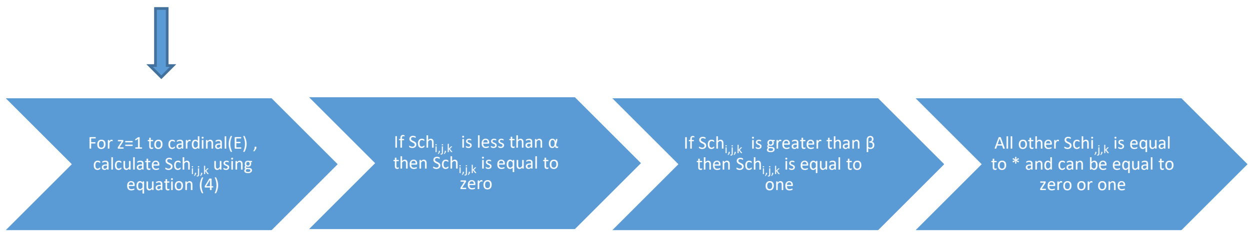


Figure 3: Schemata Generation Algorithm

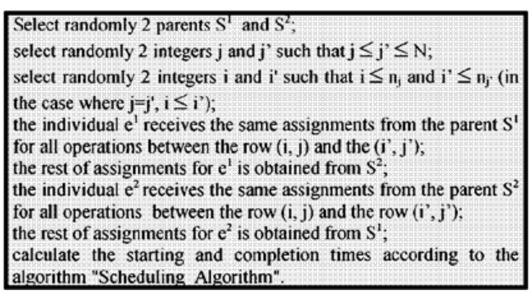
In conclusion, the author claims that this schemata covers the totality of the interesting assignment possibilities and expensive prohibitions in terms of machine workloads.

Stage II-Evaluation

In this stage, each assignment generated in stage II will be evaluated for the value of the makespan and total machine workload it amounts to.

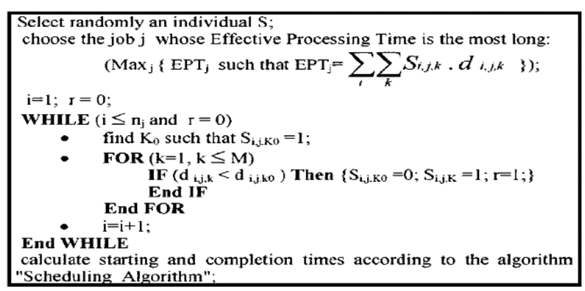
Stage III-Selection and Reproduction:

This stage contributes to the diversification of the search space of the heuristic where the author randomly choose two assignments from the initial population and performs a crossover action amongst them (refer to figure 5 for more details). Following crossover, the author performs mutation of the population using two mutation function, one that contributes to minimize the make span and the other that contributes to minimizing the total workload of machines (refer to figure 4,6 for more details).



Select randomly 2 parents S^1 and S^2;
select randomly 2 integers j and j' such that $j \le j' \le N$;
select randomly 2 integers i and i' such that $i \le n_j$ and $i' \le n_{j'}$ (in the case where j=j', $i \le i'$);
the individual e^1 receives the same assignments from the parent S^1 for all operations between the row (i, j) and the (i', j');
the rest of assignments for e^1 is obtained from S^2;
the individual e^2 receives the same assignments from the parent S^2 for all operations between the row (i, j) and the row (i', j');
the rest of assignments for e^2 is obtained from S^1;
calculate the starting and completion times according to the algorithm "Scheduling Algorithm".

Figure 5: Crossover



Select randomly an individual S;
choose the job j whose Effective Processing Time is the most long:
(Max$_j$ { EPT$_j$ such that EPT$_j$= $\sum_i \sum_k S_{i,j,k} \cdot d_{i,j,k}$ });
i=1; r = 0;
WHILE ($i \le n_j$ and r = 0)
- find K_0 such that $S_{i,j,K0}$ =1;
- **FOR** (k=1, $k \le M$)
  IF ($d_{i,j,k} < d_{i,j,k0}$) **Then** {$S_{i,j,K0}$ =0; $S_{i,j,K}$ =1; r=1;}
  End IF
  End FOR
- i=i+1;

End WHILE
calculate starting and completion times according to the algorithm "Scheduling Algorithm";

Figure 4: Mutation (Minimize make span)

Select randomly an individual S;
find the most loaded machine M_{k1}:

$$W_{k1}=\mathrm{Max}_k\{W_k/\ W_k=\sum_j\sum_i S_{i,j,k}\,.\,d_{i,j,k}\ \};$$

find the less loaded machine M_{k2} ($\mathrm{Min}_k\{\ W_k\ \}$);
choose randomly an operation $O_{i,j}$ such that $S_{i,j,k1}=1$;
assign this operation to the less loaded machine: $S_{i,j,k1}=0$; $S_{i,j,k2}=1$;
calculate the starting and completion times according to the algorithm "Scheduling Algorithm";

Figure 6: Mutation (Balance total workload of machines)

<u>Stage IV-Test:</u> In this stage, the author evaluates the improvement and decides if the solution if efficient. If the objective function reaches a satisfactory value, he takes that solution else if the solution is inefficient , he returns to the second phase and repeat the entire process until he reaches the maximum number of iterations.

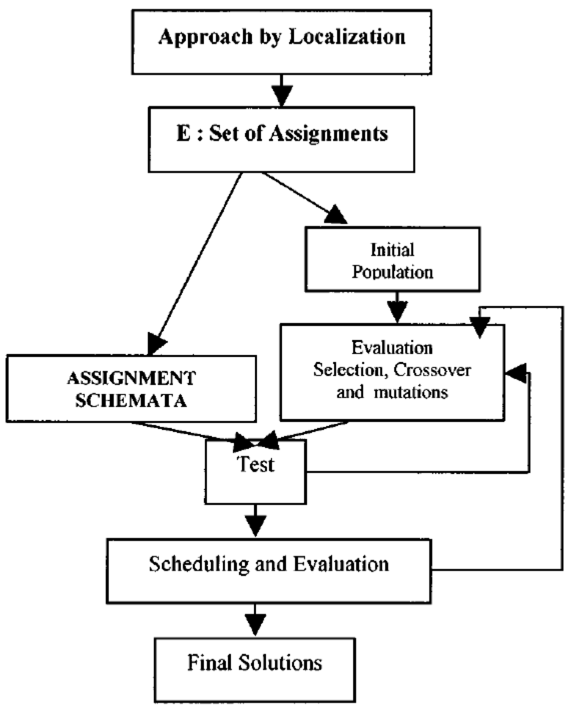


Fig. 11. Controlled genetic algorithm.

Figure 6: Controlled Genetic Algorithm

VI. AUTHORS RESULTS AND CONCLUSION

A. APPROACH BY LOCALIZATION (RESULTS)

(i) Assignment obtained from applying assignment (figure 1) on table D (table 1)

		M1	M2	M3	M4
J 1	O 1 ,1	1	0	0	0
	O 2 ,1	0	0	0	1
	O 3 ,1	1	0	0	0
J 2	O 1 ,2	0	1	0	0
	O 2 ,2	0	1	0	0
	O 3 ,2	0	0	1	0
J 3	O 1 ,3	0	0	1	0
	O 2 ,3	0	0	0	1

Figure 7

(ii) Assignment obtained from applying assignment (figure 2) on table D (table 1) (permuting first and third job)

		M1	M2	M3	M4
J 1	O 1 ,1	0	0	0	1
	O 2 ,1	0	0	0	1
	O 3 ,1	1	0	0	0
J 2	O 1 ,2	0	1	0	0
	O 2 ,2	1	0	0	0
	O 3 ,2	0	1	0	0
J3	O 1 ,3	0	0	1	0
	O 2 ,3	0	0	0	1

Figure 8

Applying scheduling algorithm to the assignment shown in figure 9, the author obtains:

1. Sum of workloads of machine=13
2. Workload of most loaded machine=5
3. Makespan=6

(iii) Assignment obtained from applying assignment algorithm (figure 2) in table P (table 3)

		M1	M2	M3	M4	M5	M6	M7	M8
J 1	O 1 ,1	0	1	0	0	0	0	0	0
	O 2 ,1	0	0	0	0	1	0	0	0
	O 3 ,1	0	0	0	0	0	1	0	0
J 2	O 1 ,2	0	0	1	0	0	0	0	0
	O 2 ,2	0	0	0	1	0	0	0	0
	O 3 ,2	0	0	0	0	0	0	1	0
	O 4 ,2	0	0	0	0	1	0	0	0
J 3	O 1 ,3	0	0	0	0	0	0	1	0
	O 2 ,3	0	0	0	1	0	0	0	0
	O 3 ,3	1	0	0	0	0	0	0	0
J 4	O 1 ,4	0	1	0	0	0	0	0	0
	O 2 ,4	0	0	0	0	0	1	0	0
	O 3 ,4	0	0	1	0	0	0	0	0
J 5	O 1 ,5	1	0	0	0	0	0	0	0
	O 2 ,5	0	0	0	0	0	0	1	0
	O 3 ,5	0	0	0	0	0	1	0	0
	O 4 ,5	0	0	0	0	0	0	1	0
J 6	O 1, 6	0	0	1	0	0	0	0	0
	O 2, 6	0	0	0	0	0	0	0	1
	O 3, 6	0	1	0	0	0	0	0	0
J 7	O 1 ,7	0	0	1	0	0	0	0	0
	O 2 ,7	0	0	0	0	0	0	0	1
	O 3 ,7	0	0	0	1	0	0	0	0
J 8	O 1 ,8	1	0	0	0	0	0	0	0
	O 2 ,8	0	1	0	0	0	0	0	0
	O 3 ,8	0	0	0	0	0	0	0	1
	O 4 ,8	0	0	0	0	1	0	0	0

Figure 9

Applying scheduling algorithm to the assignment shown in figure 10, the author obtains,

1. Sum of workloads of machine=75
2. Workload of most loaded machine=13
3. Makespan=16

The author compares the solution he obtained with the other two methods that were processed in literature.

Method	Total machine workload	Workload of most loaded machine	Make-span
Temporal Decomposition	91	19	19
Classic GA	77	11	16
Approach by Localization	75	13	16

Table 4: Comparison of AL with previous literature solutions

APPROACH BY LOCALIZATION (CONCLUSION)

1. Author concludes that AL can provide interesting solutions as obtained by using the classic GA. He further add that, assignment localized algorithm localizes most of the interesting zones of the search space thereby making scheduling easier.
2. Author also concludes that since the solution obtained from AL is satisfactory, it would be worthwhile to investigate the possible gains by hybridizing the AL with the GA.

B. GENETIC ALGORITHM (RESULTS)

(i) Assignment schemata obtained by applying algorithm shown in figure 3 on Table 3 (with α=0.03 and β=0.95

TABLE XX
ASSIGNMENT SCHEMATA S^{ch}

		M1	M2	M3	M4	M5	M6	M7	M8
J 1	O 1 ,1	*	*	0	*	*	0	0	0
	O 2 ,1	0	0	*	0	*	0	0	*
	O 3 ,1	0	0	0	0	0	*	*	0
J 2	O 1 ,2	*	0	*	0	0	0	0	0
	O 2 ,2	0	0	*	*	*	0	0	0
	O 3 ,2	0	0	0	0	0	0	1	0
	O 4 ,2	0	0	0	0	1	0	0	0
J 3	O 1 ,3	0	0	0	0	0	0	1	0
	O 2 ,3	0	0	*	*	0	0	0	0
	O 3 ,3	1	0	0	0	0	0	0	0
J 4	O 1 ,4	*	*	0	0	0	0	0	0
	O 2 ,4	0	0	*	0	0	*	*	0
	O 3 ,4	*	0	*	0	*	0	*	0
J 5	O 1 ,5	1	0	0	0	0	0	0	0
	O 2 ,5	0	0	*	*	0	0	*	0
	O 3 ,5	0	0	0	0	*	*	0	0
	O 4 ,5	0	0	0	0	0	*	*	*
J 6	O 1, 6	0	0	1	0	0	0	0	0
	O 2, 6	0	0	0	0	0	0	*	*
	O 3, 6	0	1	0	0	0	0	0	0
J 7	O 1 ,7	*	*	*	0	0	0	0	0
	O 2 ,7	0	0	0	0	0	0	0	1
	O 3 ,7	0	0	0	*	0	*	0	0
J 8	O 1 ,8	*	0	0	0	0	*	0	0
	O 2 ,8	0	*	*	0	0	0	0	0
	O 3 ,8	0	0	0	0	*	0	*	*
	O 4 ,8	0	0	*	0	*	*	0	0

Figure 10

(ii) Results obtained by using controlled genetic approach for a mono-criteria evaluation (minimizing makespan) with the following parameters:

- Population Size=Cardinal=100
- Mutation Probability, P_m=0.12
- Crossover Probability, P_c=0.88
- Number of generations=500 (stopping criteria)

TABLE XXXII
CGA SOLUTION: MONO-CRITERION EVALUATION

		M1	M2	M3	M4	M5	M6	M7	M8
J1	O 1,1	0	0	0	0	0, 3	0	0	0
	O 2,1	0	0	0	0	3, 6	0	0	0
	O 3,1	0	0	0	0	0	11, 13	0	0
J2	O 1,2	0	0	1, 4	0	0	0	0	0
	O 2,2	0	0	0	4, 6	0	0	0	0
	O 3,2	0	0	0	0	0	0	9,10	0
	O 4,2	0	0	0	0	10,14	0	0	0
J3	O 1,3	0	0	0	0	0	0	0, 2	0
	O 2,3	0	0	0	6, 10	0	0	0	0
	O 3,3	10,11	0	0	0	0	0	0	0
J4	O 1,4	0	0, 1	0	0	0	0	0	0
	O 2,4	0	0	0	0	0	4, 9	0	0
	O 3,4	0	0	9, 11	0	0	0	0	0
J5	O 1,5	0, 3	0	0	0	0	0	0	0
	O 2,5	0	0	0	0	0	0	3, 9	0
	O 3,5	0	0	0	0	0	9, 11	0	0
	O 4,5	0	0	0	0	0	0	11,14	0
J6	O 1, 6	0	0	0, 1	0	0	0	0	0
	O 2, 6	0	0	0	0	0	0	0	1, 5
	O 3, 6	0	9, 14	0	0	0	0	0	0
J7	O 1,7	0	1, 5	0	0	0	0	0	0
	O 2,7	0	0	0	0	0	0	0	5,10
	O 3,7	0	0	0	10,13	0	0	0	0
J8	O 1,8	0	0	0	0	0	0, 4	0	0
	O 2,8	0	5, 9	0	0	0	0	0	0
	O 3,8	0	0	0	0	0	0	0	10,11
	O 4,8	0	0	11,14	0	0	0	0	0

Figure 11

Applying scheduling algorithm to the assignment shown in figure 12, the author obtains,

Makespan = 14

With the above result, the author concludes that with GA approach, he is able to reduce the makespan from 16 (obtained by AL) to 14 which he obtained after five generations, which seems quick.

Method	Temporal Decomposition	Classic GA	Approach by Localization	AL + GA
Makespan	19	16	16	14

Table 5: Comparison of Techniques with AL+GA

VII. PERSONAL RESULTS AND CONCLUSION

A. <u>APPROACH BY LOCALIZATION (RESULTS)</u>

(i) Assignment obtained from applying assignment (figure 1) on table D

		M1	M2	M3	M4
J1	O1,1	1	0	0	0
	O2,1	0	0	0	1
	O3,1	1	0	0	0
J2	O1,2	0	1	0	0
	O2,2	0	1	0	0
	O3,2	0	0	1	0
	O1,3	0	0	1	0
J3	O2,3	0	0	0	1

Table 6

The result **<u>in line</u>** with authors result

(ii) Assignment obtained from applying assignment (figure 2) on table D (permuting first and third job)

		M1	M2	M3	M4
J1	O1,1	0	0	0	1
	O2,1	0	0	0	1
	O3,1	1	0	0	0
J2	O1,2	0	1	0	0
	O2,2	1	0	0	0
	O3,2	0	1	0	0
	O1,3	0	0	1	0
J3	O2,3	0	0	0	1

Table 7

Applying scheduling algorithm to the assignment shown in Table 7, I obtain:

1. Sum of workloads of machine=13
2. Workload of most loaded machine=5
3. Makespan=6

The above result **<u>in line</u>** with authors result

(i) Assignment obtained from applying assignment algorithm (figure 2) in table P

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	0	1	0	0	0	0	0	0
	2,1	0	0	0	0	1	0	0	0
	3,1	0	0	0	0	0	1	0	0
J2	1,2	0	0	1	0	0	0	0	0
	2,2	0	0	0	1	0	0	0	0
	3,2	0	0	0	0	0	0	1	0
	4,2	0	0	0	0	1	0	0	0
J3	1,3	0	0	0	0	0	0	1	0
	2,3	0	0	0	1	0	0	0	0
	3,3,	1	0	0	0	0	0	0	0
J4	1,4	0	1	0	0	0	0	0	0
	2,4	0	0	0	0	0	1	0	0
	3,4	0	0	1	0	0	0	0	0
J5	1,5	1	0	0	0	0	0	0	0
	2,5	0	0	0	0	0	0	1	0
	3,5	0	0	0	0	0	1	0	0
	4,5	0	0	0	0	0	0	1	0
J6	1,6	0	0	1	0	0	0	0	0
	2,6	0	0	0	0	0	0	0	1
	3,6	0	1	0	0	0	0	0	0
J7	1,7	0	0	1	0	0	0	0	0
	2,7	0	0	0	0	0	0	0	1
	3,7	0	0	0	1	0	0	0	0
J8	1,8	1	0	0	0	0	0	0	0
	2,8	0	1	0	0	0	0	0	0
	3,8	0	0	0	0	0	0	0	1
	4,8	0	0	0	0	1	0	0	0

Table 8

Applying scheduling algorithm to the assignment shown in table 8, I obtain,

1. Sum of workloads of machine=75
2. Workload of most loaded machine=13
3. Makespan=16

The above result **<u>in line</u>** with authors result

B. <u>GENETIC ALGORITHM (RESULTS)</u>

(i) Assignment schemata obtained by applying algorithm shown in figure 3 on Table 3 (with α=0.03 and β=0.95

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	0	*	*	*	*	0	0	0
	2,1	0	0	*	0	*	0	0	*
	3,1	0	0	0	0	0	1	0	0
J2	1,2	*	0	*	0	0	0	0	0
	2,2	0	0	0	1	0	0	0	0
	3,2	0	0	0	0	0	*	*	0
	4,2	0	0	0	0	1	0	0	0
J3	1,3	0	0	0	0	0	0	*	*
	2,3	0	0	*	*	0	0	0	0
	3,3,	*	*	0	0	0	0	0	0
J4	1,4	*	*	0	0	0	0	0	*
	2,4	0	0	0	0	0	*	*	0
	3,4	*	0	*	0	*	0	*	0
J5	1,5	*	*	0	0	0	0	0	0
	2,5	0	0	0	*	0	0	*	0
	3,5	0	0	0	0	*	*	0	0
	4,5	0	0	0	*	0	*	*	*
J6	1,6	*	0	*	*	0	0	0	0
	2,6	0	0	*	0	0	*	*	*
	3,6	1	0	0	0	0	0	0	0
J7	1,7	*	*	*	*	0	0	0	0
	2,7	0	0	0	0	0	0	1	0
	3,7	0	0	0	*	*	*	0	0
J8	1,8	0	0	0	0	0	1	0	0
	2,8	0	0	1	0	0	0	0	0
	3,8	0	0	0	0	*	*	0	*
	4,8	0	0	*	0	*	0	0	0

Table 9

(ii) Results obtained by using controlled genetic approach for a mono-criteria evaluation (minimizing makespan) with the following parameters:

- Population Size=Cardinal=100
- Mutation Probability, P_m=0.12
- Crossover Probability, P_c=0.88
- Number of generations=500 (stopping criteria)

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	0	0	0	1	0	0	0	0
	2,1	0	0	0	0	1	0	0	0
	3,1	0	0	0	0	0	1	0	0
J2	1,2	0	0	1	0	0	0	0	0
	2,2	0	0	0	1	0	0	0	0
	3,2	0	0	0	0	0	0	1	0
	4,2	0	0	0	0	1	0	0	0
J3	1,3	0	0	0	0	0	0	1	0
	2,3	0	0	1	0	0	0	0	0
	3,3,	1	0	0	0	0	0	0	0
J4	1,4	0	1	0	0	0	0	0	0
	2,4	0	0	0	0	0	0	1	0
	3,4	0	0	0	0	1	0	0	0
J5	1,5	1	0	0	0	0	0	0	0
	2,5	0	0	0	1	0	0	0	0
	3,5	0	0	0	0	0	1	0	0
	4,5	0	0	0	0	0	0	1	0
J6	1,6	0	0	1	0	0	0	0	0
	2,6	0	0	0	0	0	0	0	1
	3,6	0	1	0	0	0	0	0	0
J7	1,7	0	1	0	0	0	0	0	0
	2,7	0	0	0	0	0	0	0	1
	3,7	0	0	0	1	0	0	0	0
J8	1,8	1	0	0	0	0	0	0	0
	2,8	0	1	0	0	0	0	0	0
	3,8	0	0	0	0	0	0	0	1
	4,8	0	0	1	0	0	0	0	0

Table 10

Applying scheduling algorithm to the assignment shown in table 10, i obtain a makespan of 15.

The above result is not in line, where the author achieves a minimum makespan of 14.

		M1	M2	M3	M4	M5	M6	M7	M8
J1	1,1	0	0	0	0,3	0	0	0	0
	2,1	0	0	0	0	3,6	0	0	0
	3,1	0	0	0	0	0	6,8	0	0
J2	1,2	0	0	1,4	0	0	0	0	0
	2,2	0	0	0	4,6	0	0	0	0
	3,2	0	0	0	0	0	0	8,9	0
	4,2	0	0	0	0	11,15	0	0	0
J3	1,3	0	0	0	0	0	0	0,2	0
	2,3	0	0	4,10	0	0	0	0	0
	3,3,	10,11	0	0	0	0	0	0	0
J4	1,4	0	0,1	0	0	0	0	0	0
	2,4	0	0	0	0	0	0	2,8	0
	3,4	0	0	0	0	8,11	0	0	0
J5	1,5	2,5	0	0	0	0	0	0	0
	2,5	0	0	0	6,10	0	0	0	0
	3,5	0	0	0	0	0	10,12	0	0
	4,5	0	0	0	0	0	0	12,15	0
J6	1,6	0	0	0,1	0	0	0	0	0
	2,6	0	0	0	0	0	0	0	1,5
	3,6	0	9,14	0	0	0	0	0	0
J7	1,7	0	1,5	0	0	0	0	0	0
	2,7	0	0	0	0	0	0	0	5,10
	3,7	0	0	0	10,13	0	0	0	0
J8	1,8	0,2	0	0	0	0	0	0	0
	2,8	0	5,9	0	0	0	0	0	0
	3,8	0	0	0	0	0	0	0	10,11
	4,8	0	0	11,14	0	0	0	0	0

Table 11

However, in table 12 it can be noticed that with the number of iterations increasing, the average value of the objective function is decreasing and so is the standard deviation of the values from its mean. Thereby concluding the performance of the algorithm, which does improve the quality of my population with each iteration

	ITERATION COUNTER																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	24	24	28	29	29	28	35	35	34	27	23	23	29	21	25	27	24	38	24	24	21	21	25	21
	21	21	17	24	19	24	20	29	25	25	25	21	17	17	21	21	24	24	24	24	26	21	22	22
	22	29	19	20	26	33	22	17	24	27	23	28	27	26	21	26	22	17	21	23	21	25	23	21
	29	29	25	28	28	29	33	22	25	28	24	24	30	30	27	27	24	23	28	28	23	23	23	23
	22	19	29	29	20	26	26	21	21	17	26	19	29	29	36	28	22	21	21	21	19	17	21	21
	27	27	27	27	24	27	31	26	26	26	24	23	27	24	31	26	26	26	30	30	30	30	24	20
	29	31	31	31	22	22	37	37	37	25	29	24	22	25	30	21	21	22	22	17	23	23	23	21
	27	27	16	20	20	21	21	19	16	19	17	21	24	24	31	27	27	27	20	24	21	25	25	22
	37	21	22	21	26	22	20	28	22	26	17	21	24	21	21	26	26	25	29	20	25	28	23	23
	21	21	24	25	29	22	24	24	24	17	28	21	25	25	29	21	21	23	19	21	21	21	25	25
	24	24	23	20	21	21	21	24	24	25	26	26	24	29	29	27	22	22	21	29	24	26	26	26
	18	22	33	33	37	32	29	29	24	29	26	24	28	25	25	21	19	24	23	26	23	26	24	19
	31	21	17	25	25	20	20	20	25	27	30	26	26	26	24	24	26	25	25	25	17	19	22	22
	18	18	17	22	22	20	20	20	25	25	25	27	22	21	20	29	29	29	29	26	26	27	22	26
	25	30	25	33	45	34	26	26	20	25	25	20	31	19	22	29	29	24	21	21	21	21	20	29
	25	23	23	25	27	24	19	22	30	33	24	30	24	26	22	28	24	23	23	21	28	23	22	22
	26	21	21	22	24	24	24	17	17	33	28	28	28	20	21	26	19	18	22	26	21	21	22	22
	28	25	34	34	37	37	22	22	23	22	19	19	21	23	23	23	26	26	19	22	23	23	21	26
	19	24	19	31	25	25	25	24	25	38	29	24	22	33	30	24	24	29	28	28	24	24	24	21
	24	24	29	26	25	24	24	27	28	23	24	24	24	24	29	17	21	23	24	23	21	21	22	23
	22	25	28	27	21	31	31	31	31	31	27	21	28	23	21	30	30	25	23	22	22	21	21	21
	27	22	26	31	27	21	29	21	26	34	23	23	24	21	31	19	28	28	21	17	17	23	21	21
	22	20	21	31	31	24	29	34	29	26	21	21	26	38	21	21	21	21	22	29	29	23	19	24
	19	19	19	30	26	24	38	34	21	27	19	20	21	22	22	19	26	29	29	29	29	23	27	23
	21	25	28	27	25	25	25	20	20	24	24	22	25	28	23	23	23	28	25	25	29	24	25	23
	26	26	28	25	29	23	23	23	21	26	20	25	25	21	26	20	17	17	23	17	17	17	24	19
	24	24	24	21	24	25	17	21	21	33	21	25	18	31	20	23	23	24	26	19	27	27	26	26
	21	37	26	24	21	21	21	36	27	26	26	30	29	21	25	21	23	23	23	22	20	24	24	24
	23	21	31	31	31	21	21	21	27	25	33	26	26	25	25	24	28	24	25	25	21	21	21	23
	20	27	29	29	29	24	23	24	24	24	26	26	26	22	22	25	25	25	28	28	19	19	22	22
	25	28	22	31	26	36	27	27	33	24	24	21	21	23	24	23	23	28	28	25	25	22	21	30
	21	25	25	21	23	27	27	26	39	26	31	31	22	26	28	28	21	23	23	21	23	28	29	20
	21	26	25	30	25	21	22	25	31	36	36	27	26	33	26	17	21	22	27	17	24	19	17	20
	25	21	21	28	22	26	28	23	25	19	22	24	24	26	26	23	23	21	26	26	24	24	24	24
	22	19	20	22	29	35	35	29	26	28	26	26	25	22	22	22	22	25	27	21	17	24	26	21
	24	22	24	19	23	27	22	22	21	28	27	21	21	17	17	26	26	29	32	21	27	23	23	25
	23	25	26	23	22	22	22	23	27	26	31	21	26	26	26	24	29	24	25	25	24	24	19	26
	21	21	24	24	22	21	21	21	21	24	22	20	20	26	24	24	26	21	23	19	20	22	22	22
	21	21	24	22	20	26	26	40	33	24	24	21	24	24	29	17	25	17	17	19	20	21	21	21
	34	29	26	27	20	20	24	24	26	22	29	26	21	25	17	23	30	30	23	21	21	21	23	21
	29	29	30	33	44	44	27	24	22	22	21	31	29	29	25	25	25	25	25	28	19	24	29	31
	31	20	25	34	41	41	22	21	23	20	21	31	38	26	26	21	30	29	21	21	21	29	29	26
	23	27	31	29	35	29	28	28	28	28	28	28	31	21	25	22	22	24	24	21	23	20	26	24
	16	24	22	22	22	28	28	22	21	21	24	30	26	30	26	25	24	26	23	23	21	18	17	17
	21	29	21	37	24	30	30	27	22	24	33	21	26	28	30	27	28	19	23	24	17	22	19	24
	17	17	33	25	30	30	24	24	28	28	20	20	31	38	38	17	23	23	23	23	21	26	19	19
	22	21	21	30	30	29	29	30	20	23	33	25	23	21	17	29	29	29	20	20	22	17	20	19
	27	31	26	21	21	21	26	33	24	21	21	19	21	24	21	23	21	23	21	24	28	28	25	25
	21	27	31	29	29	24	40	27	25	25	23	29	21	20	28	36	30	24	21	24	26	26	28	21
Population	25	19	19	25	25	22	27	31	24	24	22	22	22	35	31	21	25	25	26	27	21	25	26	23
	18	25	24	25	22	19	19	19	21	29	20	24	21	21	23	23	22	24	23	22	22	23	20	17
	28	24	24	29	29	25	25	30	21	21	26	29	20	23	19	31	31	26	24	25	25	23	25	28
	27	24	21	21	21	17	29	22	28	28	33	22	24	21	21	25	19	21	20	19	19	19	19	21
	25	19	21	26	26	36	30	27	22	22	22	27	35	27	38	28	28	29	29	19	19	19	19	26
	27	22	33	31	31	31	26	26	24	24	20	32	26	21	23	24	24	22	26	23	23	23	28	28
	33	33	27	29	27	26	20	23	23	18	31	21	21	24	22	28	29	24	24	21	21	19	21	21
	19	24	31	22	19	22	33	26	26	27	23	24	24	27	27	30	21	21	21	23	28	23	21	21
	24	25	25	23	24	25	25	25	28	21	22	24	24	24	29	22	19	29	31	24	25	25	24	28
	21	31	25	23	19	19	19	22	24	22	19	28	31	28	24	22	24	23	24	24	24	19	19	22
	22	21	20	24	24	21	19	19	19	24	24	21	25	29	26	26	24	31	24	22	22	20	22	24
	29	29	24	29	23	29	29	35	31	30	23	26	24	20	21	21	21	19	28	28	28	20	24	24
	27	24	26	24	27	28	24	31	31	31	31	24	24	24	29	29	29	24	23	23	23	21	23	23
	19	31	27	31	21	27	24	27	20	23	23	28	27	27	22	29	29	21	21	19	19	21	21	21
	26	28	38	23	23	23	22	23	22	31	26	26	26	24	26	28	23	21	21	23	23	25	21	21
	25	26	25	26	24	29	29	29	22	27	31	22	22	27	20	29	29	30	23	27	22	26	27	26
	27	23	18	19	19	20	22	19	17	25	25	21	23	23	23	21	21	23	24	29	23	21	25	24
	26	25	31	19	19	22	22	22	24	26	25	25	25	36	23	22	18	18	22	23	26	23	23	23
	29	18	18	18	24	31	23	22	25	25	33	33	33	24	26	24	22	24	24	18	18	21	21	26
	21	21	20	19	22	32	38	27	27	36	19	23	23	23	23	20	22	34	34	27	23	22	20	17
	22	22	29	19	19	25	23	25	21	24	19	19	19	24	27	36	35	35	31	31	24	26	21	21
	35	31	31	24	24	24	22	22	25	20	29	28	30	26	26	26	26	26	22	19	21	26	24	26
	25	27	27	27	23	25	21	21	34	27	23	29	19	26	21	25	26	21	28	24	25	21	21	21
	25	21	25	26	26	26	26	21	21	23	21	21	21	26	26	22	28	20	19	21	21	23	21	21
	21	24	25	25	25	24	21	30	22	25	25	25	25	29	24	23	26	21	22	22	22	18	21	21
	26	29	22	17	26	22	33	37	33	33	33	37	33	29	26	26	31	26	21	21	21	17	24	24
	19	29	23	22	22	20	24	23	24	21	26	24	24	24	21	21	20	21	23	21	21	21	21	21
	34	20	19	19	23	23	23	23	25	20	19	24	28	25	24	24	24	22	22	25	22	25	27	27
	29	26	28	27	17	21	21	21	30	22	36	19	20	24	25	24	24	24	20	20	26	22	23	22
	22	27	27	25	30	21	21	17	17	26	21	20	23	23	33	37	28	26	21	21	25	25	28	19
	28	28	24	24	24	24	29	20	27	33	22	22	20	20	22	26	21	17	17	27	27	26	23	23
	31	17	21	26	26	30	30	35	30	31	31	25	21	21	21	26	23	29	25	29	22	23	23	21
	25	28	26	24	25	26	17	21	22	22	22	24	29	23	17	38	23	27	35	32	23	23	25	25
	29	24	30	25	25	25	25	25	25	24	29	29	25	27	30	23	23	29	19	22	25	21	26	26
	24	25	25	25	22	24	26	26	27	24	24	22	24	20	19	23	23	21	24	23	23	20	23	22
	31	27	31	21	27	26	26	20	20	22	20	22	17	22	24	26	23	23	23	26	29	29	28	19
	23	21	17	20	26	26	25	21	33	33	25	19	23	20	20	26	24	24	21	23	20	23	29	21
	26	26	25	24	24	21	21	27	30	29	23	23	25	23	23	29	29	23	24	25	22	24	22	23
	26	33	23	25	27	22	23	23	23	29	20	23	19	33	23	21	17	25	20	20	23	25	28	28
	22	25	25	17	31	26	26	26	21	25	29	29	26	19	20	26	29	24	23	21	21	19	21	21
	23	23	26	31	29	29	23	25	23	23	23	30	21	21	21	23	22	20	25	24	24	20	24	29
	30	30	33	25	26	25	25	25	23	25	25	31	28	25	21	21	21	19	23	25	30	30	26	19
	20	34	23	25	22	25	28	36	28	21	25	25	21	22	22	22	27	21	29	27	22	29	23	22
	24	20	31	31	31	29	24	26	24	20	24	20	20	20	27	18	18	24	25	32	32	25	23	19
	21	22	27	28	32	28	27	27	27	21	21	28	24	24	26	24	17	20	20	17	25	23	17	17
	27	27	31	31	34	34	20	22	22	22	22	25	25	22	22	23	28	27	23	23	23	23	26	21
	19	31	26	26	26	21	27	24	24	24	21	34	34	31	22	21	23	25	21	21	21	28	23	25
	26	31	21	31	23	25	31	22	26	22	23	24	22	21	23	23	21	27	28	25	25	32	32	32
	18	19	25	21	21	25	25	25	21	21	17	35	25	28	29	20	28	22	18	18	25	21	23	23
	21	24	24	24	29	33	33	29	24	23	23	21	23	29	28	22	21	21	17	17	18	22	22	22
	37	28	29	21	25	27	21	24	36	28	28	26	23	25	26	23	22	22	17	24	24	24	36	36
Mean	24.53	24.8	25.12	25.5	25.62	25.77	25.36	25.24	25.04	25.39	24.75	24.69	24.65	24.85	24.63	24.46	24.29	24.16	23.68	23.27	22.97	22.97	23.29	22.98
Standard Deviation	4.369	4.135	4.441	4.33	5.069	4.864	4.749	4.885	4.459	4.24	4.293	3.949	3.915	4.187	4.178	4.036	3.617	3.825	3.602	3.518	3.189	3.132	3.182	3.301

Table 12: Evaluation of function objective with each iteration