Journal of Science and Technology
ISSN: 2456-5660 Volume 5, Issue 5, Sep-October 2020

www.jst.org.in *DOI: https://doi.org/10.46243/jst.2020.v5.i5.pp154-161*

Computational Experiments on Workflow Balancing of Parallel Machine Scheduling with Precedence Constraints and Sequence Independent Setup Time

S.Rajakumar
(Department of Mechanical Engineering, Sree Sakthi Engineering College, Karamadai, Coimbatore, Tamilnadu, India)
Corresponding Author:rajakumar6@yahoo.com

To Cite this Article
S.Rajakumar, "Computational Experiments on Workflow Balancing of Parallel Machine Scheduling with Precedence Constraints and Sequence Independent Setup Time ", Journal of Science and Technology, Vol. 05, Issue 05, Sep-October 2020, pp154-161

Article Info
Received: 02-06-2020 Revised: 01-09-2020 Accepted: 04-09-2020 Published: 08-09-2020

Abstract: *The workflow balancing of parallel machines scheduling (PMS) with precedence constraints and sequence independent setup time is considered for study. The setup time consideration produces alternate schedule along with lesser relative percentage of imbalance (RPI) value for PMS problem is demonstrated with an example. The lesser RPI indicates better workflow balancing among machines. The computational experiments are conducted on large instances of randomly generated PMS problems with precedent constraints and setup time. The various combinations of heuristics are used to solve the problems. The results show that genetic algorithm (GA) performs well against the other heuristics with lesser RPI values.*

Keywords*: workflow, workload, balancing, scheduling, genetic algorithm, parallel machine scheduling*

I. Introduction

The parallel machine scheduling (PMS) problems with precedence constraints and sequence independent setup time deals with assigning of *n* jobs on *m* parallel machines. The jobs are assigned on machines satisfying the precedence relationship to break the constraints. The setup time of each job is added with its processing time and it varies with jobs. The objective of scheduling is to achieve workflow balancing among parallel machines. The workflow balancing tends to distribute workloads uniformly on all the machines. The jobs are assigned on machines using list scheduling heuristics namely random (RANDOM), shortest processing time (SPT**)**, longest processing time (LPT**)** and genetic algorithm (GA). Their performance is compared with a measure of relative percentage of imbalance (RPI) proposed by Rajakumar [1]. Firstly, an example is shown to illustrate the difference in schedule and RPI values among without setup time and with setup time consideration. Secondly, the computational experiments are conducted for solving PMS problems with precedence constraints and sequence independent setup time for large instances of varied jobs and machines. The GA shows better performance over other heuristics through lesser RPI values.

II. Literature review

The PMS problems of assigning set of jobs on machines or resources in a shop floor belong to combinatorial optimization problems. They can be solved by heuristic algorithms with reasonable amount of time. Pankaj Sharma et al. [2] discussed that researchers had solved job shop scheduling problems with sequence independent setup time using heuristic and scheduling rules. Previously setup was considered as non value added activity but, now it has to be considered to increase productivity, eliminate waste, improve resource utilization and meet deadlines [3].The LPT rule produces better result for uniform PMS with assumed sequence independent setup time and operations [4]. The proposed GA works better for medium and larger instances of PMS problems with non-renewable resources [5]. The workload balancing is essential for production systems to efficiently use all of their resources [6].Balancing of workload leads to maximize machine tool utilization on each machine and relieving bottlenecks with the intent of maximizing system throughput [7].The workload balancing problem is the decrease of difference among workload of bottleneck machine and workload of the fastest machine [8].Usually, the setup time has been considered as a part of job processing times. This assumption leads to many problems in scheduling process and setup time must be considered for more realistic scheduling [9]. It is evident from the review that disturbance of schedule and workflow balancing have to be addressed due to setup time consideration in PMS problems.

III. Methodology

The PMS problems with precedence constraints and sequence independent setup time have *n* jobs be assigned on a set of *m* machines, so that workflow is balanced. Many researchers claim that setup time of jobs are added to the processing time to solve the sequence independent setup time PMS problems. So, the problems become similar to the PMS problems without setup. But in real life situation it differs and an example is shown to bring out the difference on setup time consideration in PMS problems on job allocation using LPT. The list of randomly generated processing time and setup time for 10 jobs X 3 machines problem is shown in Table 1. The setup time of a job is added to processing time to get the total time of job.

$$T_j = P_j + S_j \qquad 1 \leq j \leq 10 \qquad (1)$$

Table 1. Randomly generated processing time and setup time of jobs

Job Code(J)	Processing time (P)	Setup time(S)	Total time (T)
1	48	20	68
2	46	19	65
3	35	24	59
4	41	10	51
5	30	5	35
6	56	12	68
7	63	42	105
8	85	32	117
9	42	40	82
10	23	50	73

Table 2 shows the list of processing order of jobs with and without setup time according to LPT. The processing order of jobs with setup time differs because setup time of jobs is added with processing time. The jobs are allocated to machines by using workload allocation algorithm. A new job from the ordered list of jobs assigned to the least loaded machine.

Workload allocation algorithm

Step 1 Select a job J from the list of jobs ordered using heuristics.

Step 2 Calculate the workload W on all machines individually i.e. adding processing or total time of jobs assigned to them.

Step 3 Identify the machine M with least workload and assign the job J.

Step 4 Repeat the steps from 1 to 4 until all the jobs are assigned from the list.

The workload imbalances among machines are calculated taking maximum workload W_{max} as index. The performance measure of RPI values for each machines are calculated using the equation.

$$\text{Relative percentage of imbalance (RPI)} = \frac{(W_{max} - W_j)}{W_{max}} X100 \qquad (2)$$

The average RPI values are used to compare the performance of methods.

Table 2. Processing order by LPT

Without setup time		With setup time	
J	P	J	T
8	85	8	117
7	63	7	105
6	56	9	82
1	48	10	73
2	46	6	68
9	42	1	68
4	41	2	65
3	35	3	59
5	30	4	51
10	23	5	35

Table 3. Allocation of jobs without setup time for LPT

Machine 1		Machine 2		Machine 3	
J	P	J	P	J	P
8	85	7	63	6	56
9	42	2	46	1	48
5	30	3	35	4	41
		10	23		
W	157		167		145
RPI	5.99		0		13.17
Average RPI	6.39				

Table 4. Allocation of jobs with setup time for LPT

Machine 1		Machine 2		Machine 3	
J	T	J	T	J	T
8	117	7	105	9	82
1	68	6	68	10	73
4	51	3	59	2	65
				5	35
W	236		232		255
RPI	7.45		9.02		0
Average RPI	5.49				

Table 3 shows allocation of jobs on machines based on processing time alone without setup time. The allocation of jobs on machines based on total time by considering setup time is shown in Table 4.The allocation of jobs on machines differs in both cases. Jobs 8, 9 and 5 are allocated on machine 1 by using processing time without setup time. Jobs 8, 1 and 4 are allocated on machine 1 by using total time with setup time. Similarly, allocation of jobs on other two machines differs with and without setup time considerations. The average RPI values are 6.39% and 5.49% for with setup and without setup respectively. The least RPI values show the balanced workflow among machines. Thus, the processing order of jobs disturbs the allocation of jobs on machines.

Table 5. List of heuristics

Serial Number	Name of the heuristics	Name of the strategy applied to	
		Preceding jobs	Succeeding jobs
1	RR	RANDOM	RANDOM
2	RS	RANDOM	SPT
3	RL	RANDOM	LPT
4	SR	SPT	RANDOM
5	SS	SPT	SPT
6	SL	SPT	LPT
7	LR	LPT	RANDOM
8	LS	LPT	SPT
9	LL	LPT	LPT
10	GA	GA	GA

The PMS problem with precedence constraints and setup time considered for study deals with assigning jobs to obtain workflow balancing among machines. Two lists of jobs namely P representing preceding jobs and S representing succeeding jobs are extracted from the mail list L. These two lists are ordered according to the heuristics RANDOM, SPT, LPT and GA separately. The heuristics combined together get ten different heuristics as shown in table 5. The first letter of heuristics indicates preceding list and second letter succeeding list of jobs. The jobs from the lists are assigned according to the workload allocation algorithm on machines. The GA proposed by Rajakumar [10] is used for computational experiments for varied jobs and machines.

IV. Computational Experimentation

The experiments have been conducted with a program in C++ language on IBM/PC compatible system. The random number generator is used for generating jobs processing and setup times for conducting experiments. They are generated once per instance and all the ten heuristics applied one by one to maintain similarity. Fifty problems are generated for a set of *n* jobs and *m* machines. The number of jobs in each set varied from 10 to 150 and total number of sets considered for the study is 15.Then, total amount of instances are generated is equal to 15 X 50 = 750 for one *m* machines problem. The number of machines varied from 2 to 5 for the computational experiments. Hence, 750 X 5 = 3750 instances are generated to test the performance of heuristics. The output of problems directed to the EXCEL to calculate RPI values for further analysis. The workload imbalances on machines are calculated by considering maximum workload as index. The least average RPI values indicate uniform distribution of workloads. The RPI values for all the ten heuristics are calculated to compare performance of workflow balancing among parallel machines.

V. Results and discussion

The experiments are conducted on all machines from 2 to 6 machines by varying jobs at the interval of 10 from 10 to 150 jobs. The tables and charts for *n x 2* machines, *n x 4* machines and *n x 6* machines problems are shown. The performance of heuristics are shown in plotted lines by using number of jobs and average RPI on X and Y axes respectively.

Table 6. Relative percentage of imbalance for two machines

Number of jobs (n)	RR	RS	RL	SR	SS	SL	LR	LS	LL	GA
10	5.913	7.367	4.277	5.053	6.775	3.436	5.781	6.960	5.032	0.176
20	3.522	4.041	1.956	3.258	3.819	1.468	3.461	4.119	1.729	0.061
30	2.105	2.669	0.823	2.233	2.577	0.618	2.027	2.630	0.970	0.038
40	1.623	2.309	0.484	1.638	2.339	0.409	1.632	2.286	0.519	0.031
50	1.300	1.748	0.289	1.273	1.887	0.253	1.271	1.876	0.379	0.019
60	1.102	1.614	0.192	1.061	1.563	0.206	1.081	1.586	0.207	0.017

70	0.903	1.311	0.160	0.914	1.291	0.179	0.905	1.281	0.158	0.016
80	0.758	1.196	0.108	0.751	1.249	0.088	0.749	1.214	0.108	0.013
90	0.868	1.118	0.083	0.879	1.072	0.067	0.889	1.114	0.098	0.011
100	0.616	0.980	0.079	0.615	0.988	0.069	0.616	0.997	0.070	0.011
110	0.615	0.863	0.055	0.610	0.865	0.053	0.606	0.860	0.055	0.009
120	0.461	0.830	0.049	0.468	0.827	0.039	0.463	0.824	0.053	0.008
130	0.456	0.763	0.046	0.445	0.775	0.042	0.445	0.781	0.049	0.007
140	0.523	0.713	0.037	0.526	0.726	0.034	0.535	0.710	0.038	0.007
150	0.464	0.663	0.038	0.472	0.664	0.032	0.469	0.678	0.029	0.006

Table 6 and Figure 1 show the performance of heuristics for two machines. The number of jobs is varied between 10 and 150. The RS produces maximum RPI value of 7.37 %. For smaller range of jobs, RPI values produced by the nine heuristics except GA are more. The RPI values decrease with the increase in number of jobs. The RPI values are smaller for RL, SL and LL but so smaller for GA. The RS, SS and LS produced more imbalances of workloads among the machines. The RR, SR and LR produced approximately same RPI values. They produce moderate RPI values. The GA produced the least RPI value of 0.006 % for 150 jobs. The GA produced lesser RPI values for all various jobs size. So, the workloads are distributed uniformly among the machines. The less RPI value indicates the schedule with balanced workflow.

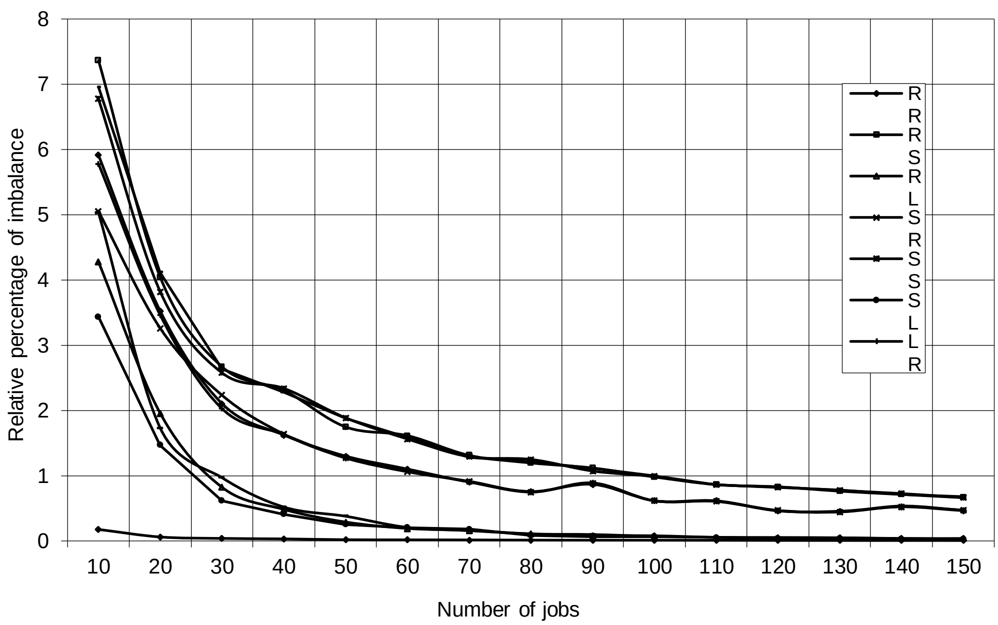

Figure 1. Relative percentage of imbalance for two machines

Table 7. Relative percentage of imbalance for four machines

Number of jobs (n)	RR	RS	RL	SR	SS	SL	LR	LS	LL	GA
10	18.069	19.457	14.509	18.479	19.903	14.713	18.908	20.692	16.913	6.012
20	10.510	12.618	6.146	10.284	12.727	4.580	11.220	12.662	8.049	0.737
30	6.822	8.946	2.976	6.813	8.870	2.454	6.480	8.811	4.014	0.295

40	5.687	6.809	1.629	5.848	6.814	1.548	5.757	6.665	1.846	0.203
50	4.077	5.164	1.184	4.261	5.364	1.079	4.008	5.342	1.288	0.144
60	3.679	4.534	0.959	3.690	4.693	0.776	3.508	4.696	0.999	0.131
70	3.063	4.111	0.876	3.044	4.003	0.629	3.077	4.116	0.828	0.104
80	3.005	3.628	0.535	2.997	3.617	0.450	3.043	3.650	0.584	0.097
90	2.656	3.159	0.364	2.646	3.181	0.403	2.573	3.200	0.471	0.074
100	2.287	2.827	0.316	2.287	2.912	0.300	2.326	2.859	0.311	0.073
110	2.089	2.710	0.265	2.099	2.599	0.218	2.109	2.647	0.272	0.066
120	1.961	2.363	0.251	1.995	2.358	0.209	1.960	2.407	0.248	0.063
130	1.686	2.267	0.175	1.672	2.242	0.159	1.664	2.222	0.197	0.060
140	1.329	2.084	0.198	1.336	2.086	0.158	1.302	2.073	0.217	0.056
150	1.377	1.976	0.153	1.365	1.925	0.121	1.337	1.933	0.136	0.040

The performance of ten heuristics with varied number of jobs is shown with the help of Figure 2 and RPI values for all the ten heuristics are shown in the Table 7 for four machines. The workloads are balanced when the number of jobs is more. The RR, SR and LR show moderate RPI values irrespective of number of jobs. The LS shows the worst performance measure for workflow balancing. The RS, SS and LS show more imbalances in workloads among the machines. Hence, they have more RPI values. The most RPI value is found for LS at 20.69% for 10 jobs. The SL has RPI values lesser than 1% for jobs more than 50. Both RL and LL also show values below 1% for jobs more than 50. But, GA produces RPI values lesser than 1% for jobs more than 10. The GA produces lesser RPI values for all job sizes. This shows that workflow is being balanced by using GA.

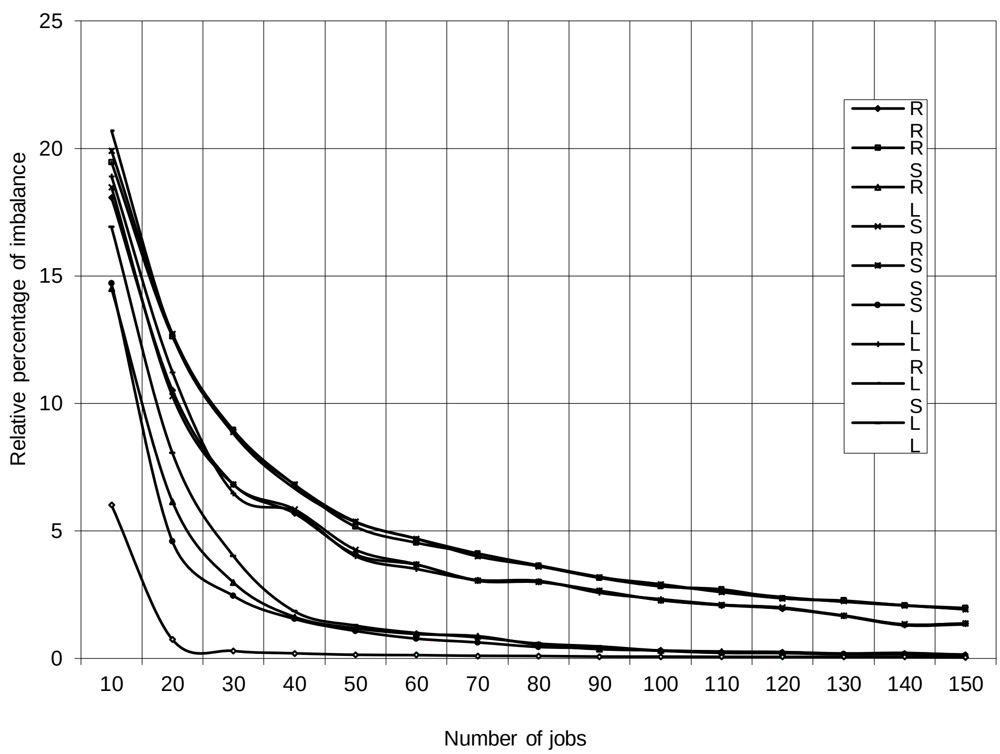

Figure 2. Relative percentage of imbalance for four machines

Table 8. Relative percentage of imbalance for six machines

Number of jobs(n)	RR	RS	RL	SR	SS	SL	LR	LS	LL	GA
10	29.201	30.775	25.758	29.442	31.177	26.819	29.692	30.965	26.379	21.286
20	17.603	19.883	12.180	16.815	19.084	10.261	19.917	21.135	17.942	3.493
30	13.111	15.370	7.456	12.636	15.072	5.536	13.411	14.819	10.586	1.965
40	9.332	11.417	4.108	9.479	11.582	3.819	9.274	11.600	6.261	1.131
50	8.025	9.086	2.442	7.988	9.023	2.042	7.669	9.139	2.971	0.823
60	6.891	7.823	1.737	6.993	7.632	1.557	6.814	7.818	1.948	0.735
70	5.743	6.739	1.379	5.535	6.580	1.014	5.485	6.793	1.645	0.567
80	4.755	5.859	1.124	4.824	5.972	1.039	5.067	5.791	1.283	0.498
90	4.075	5.436	0.846	4.228	5.261	0.687	4.155	5.258	0.843	0.466
100	4.091	4.923	0.655	4.059	4.778	0.657	4.111	4.815	0.756	0.419
110	3.987	4.537	0.571	3.945	4.592	0.500	4.051	4.536	0.696	0.368
120	2.646	3.151	0.391	2.632	3.176	0.326	2.676	3.166	0.454	0.311
130	3.116	3.533	0.469	3.135	3.529	0.405	3.131	3.527	0.500	0.306
140	2.785	3.353	0.384	2.805	3.379	0.330	2.761	3.354	0.386	0.247
150	2.861	3.212	0.323	2.860	3.340	0.302	2.888	3.266	0.361	0.200

The RPI values against the number of jobs illustrated with Table 8 and Figure 3 for six machines. The GA has the least RPI value of 21.29% for 10 jobs and SS has maximum imbalance of 31.18% for 10 jobs. The RS, SS and LS are having imbalances in workloads between 3.21 and 31.18%. They have more RPI values than other heuristics. The RR, SR and LR show moderate RPI values for all ranges of jobs. The RL, SL and LL produce lesser amount of imbalances in workloads among machines as like in the previous cases. They show RPI values lesser than 1% starting from 90 jobs. The GA produces RPI values lesser than 1% from 50 jobs onwards. It is evident that workloads are balanced by using GA in scheduling.

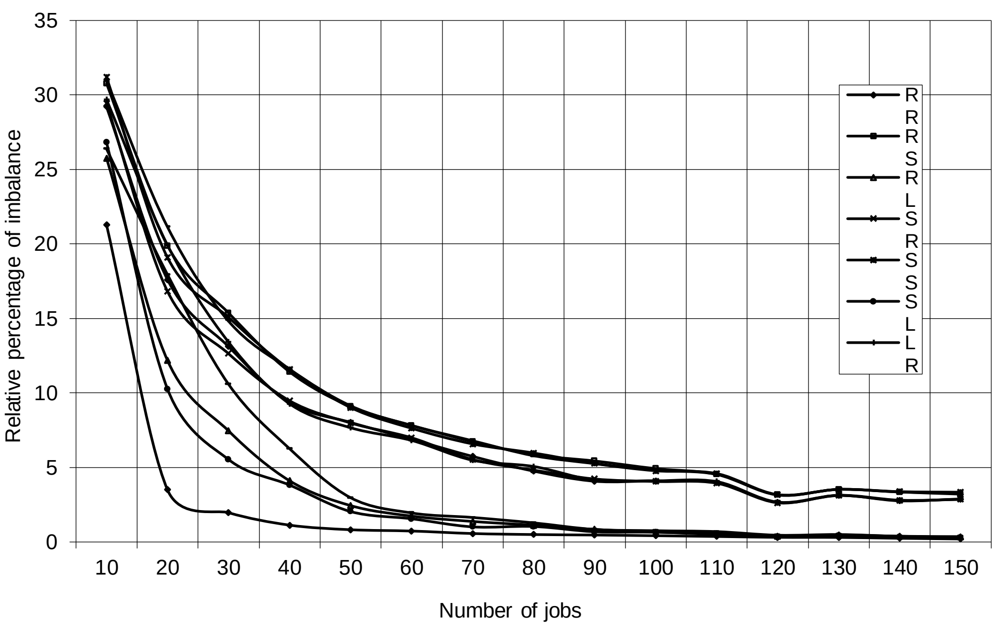

Figure 3. Relative percentage of imbalance for six machines

The above tables and figures show the performance of PMS with precedence constraints and setup time using various scheduling heuristics. The performance measure of RPI in workloads adopted for evaluation. The GA produces satisfied results over other heuristics. They show lesser RPI values for all the cases considered for study. It shows similar performance of producing lesser RPI values for all ranges of jobs and machines. They are producing RPI values lesser than 1% for all jobs sizes in two machines, from 20 in four machines, and 50 in six machines. This shows that the points of 1% RPI value shift rightwards horizontally with the increase in number of jobs. The RR, SR and LR show moderate RPI values in all the cases considered for the study. More RPI values of RS, SS and LS show that workloads are not distributed evenly. It is evident that GA performs better than other heuristics.

VI. Conclusion

The workflow balancing of PMS problem with precedence constraints and sequence independent setup time is studied. The difference in schedules with and without setup for PMS problem is shown with an example. The PMS problem with setup time produced lesser RPI values to indicate better workflow balancing among the parallel machines. The same analogy is used to solve PMS problems with precedence constraints and setup time through computational experiments. The GA produces lesser RPI values for all set of problems against other heuristics. This confirms that workflow balancing among parallel machines can be achieved by considering precedence constraints and setup time using heuristics. The problem may be also extended to solve using other Meta heuristics.

References

[1] Rajakumar S, Workflow balancing in a speed frame assembly shop floor operations through genetic algorithm, International Journal of Advances in Engineering and Management (IJAEM),2020,Volume 2, Issue 4,74-80.

[2] Pankaj Sharma, Ajai Jain, A review on job shop scheduling with setup times, Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture · January 2015(3),517-533.

[3] Andres Muñoz-Villamizara, Javier Santosb, Jairo Montoya-Torresc and Maria Jesus Alvarézb, Improving effectiveness of parallel machine scheduling with earliness and tardiness costs: A case study, International Journal of Industrial Engineering Computations, 2019(10), 375–392.

[4] Jun-Ho Lee, Hoon Jang, Uniform Parallel Machine Scheduling with Dedicated Machines, Job Splitting and Setup Resources, Sustainability, 2019, 11(24), 7137; **https://doi.org/10.3390/su11247137**

[5] F. Belkaid, A. Hadri and M. Bennekrouf, Efficient Approach for Parallel Machine Scheduling Problem, International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Tangier, 2018, 158-163, doi: 10.1109/LOGISTIQUA.2018.8428302.

[6] Anthony Cossari, Johnny C. Ho, Giuseppe Paletta & Alex J. Ruiz-Torres, Minimizing workload balancing criteria on identical parallel machines, Journal of Industrial and Production Engineering, 2013, 30:3, 160-172, DOI: 10.1080/21681015.2013.795504

[7] Maurizio Faccio, Setup time reduction: SMED-balancing integrated model for manufacturing systems with automated transfer, International Journal of Engineering and Technology (IJET), 2013,Vol 5 No 5,4075-4084.

[8] Hussien, Ali Mohamed and Hussein, Mohamed Eid and Hassan, Moatamad Refaat, Improved Workload Balancing of the Scheduling Jobs With the Release Dates in Identical Parallel Machines, 2019, International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol. 9, No.6. SSRN: https://ssrn.com/abstract=3536786 or http://dx.doi.org/10.2139/ssrn.3536786

[9] Abreu, L.R.d. and Prata, B.d.A. A genetic algorithm with neighborhood search procedures for unrelated parallel machine scheduling problem with sequence-dependent setup times, Journal of Modelling in Management,2020, Vol. 15 No. 3, 809-828. https://doi.org/10.1108/JM2-12-2018-0209

[10] Rajakumar S, A genetic algorithm for workflow balancing of parallel machine scheduling with precedence constraints, International Research Journal of Modernization in Engineering Technology and Science Volume:02/Issue:08/August-2020 537-544.