

Flow shop scheduling decisions through Techniques for Order Preference by Similarity to an Ideal Solution (TOPSIS)

Arun Gupta^{a*} and Shailendra Kumar^b

^a Mechanical Engineering Department, College of Engineering, Teerthanker Mahaveer University, Moradabad (UP) India.

* engg.arun12@gmail.com

^b Mechanical Engineering Department, Meerut Institute of Engineering and Technology, Meerut (UP) India.

tyagi_sk@yahoo.com

Abstract: The flow-shop scheduling problem (FSP) has been widely studied in the literature and having a very active research area. Over the last few decades, a number of heuristic/meta-heuristic solution techniques have been developed. Some of these techniques offer excellent effectiveness and efficiency at the expense of substantial implementation efforts and being extremely complicated. This paper brings out the application of a Multi-Criteria Decision Making (MCDM) method known as techniques for order preference by similarity to an ideal solution (TOPSIS) using different weighting schemes in flow-shop environment. The objective function is identification of a job sequence which in turn would have minimum makespan (total job completion time). The application of the proposed method to flow shop scheduling is presented and explained with a numerical example. The results of the proposed TOPSIS based technique of FSP are also compared on the basis of some benchmark problems and found compatible with the results obtained from other standard procedures.

Key words: Scheduling, Multi-Criteria Decision Making, TOPSIS, Flow-shop.

1. Introduction

In modern world, where things are changing with a pace never before and every aspect of human life is affected from this change. Every industry is looking for strategies and technologies not only to retain their profit margins and market-share but to improve them, manufacturing industry is not an exception. Concept of cellular manufacturing seems to be a solution in this situation. An enormous and growing body of literature is available on it. In simplest words cellular manufacturing is a manufacturing process in which every manufacturing cell acts as an independent manufacturing unit. Cell formation, cell layout and scheduling are the three basic steps in design of any cellular manufacturing system (Kumar and Sharma, 2015). Cell formation involves identification and grouping of machine cells and part families (Albadawi *et al.*, 2005; Kumar and Sharma, 2014, 2015). Cell layout emphasizes placing of machines within a manufacturing cell. Scheduling is simply sequencing of tasks on different machines in a manufacturing cell. Researchers are

working tirelessly on all three stages of cellular manufacturing, considering them independently as well as concurrently. Out of these three stages, scheduling is one of the important and critical issue in the production planning and operation of a manufacturing system. Scheduling is the allocation of a set of limited resources to a number of jobs over time, with the objective of optimizing one or more performance criterion (French, 1982). In the set of machine, scheduling finds a sequence of jobs with constraints to optimize one or more objectives such as makespan, tardiness, work-in-process inventory, number of tardy jobs, idle time, etc. (Nakhaeinejad and Nahavandi, 2012). Two common issues that frequently appear in the scheduling literature are flow shop scheduling (Dannenbring, 1977; King and Spachis, 1980; Taillard, 1990; Fink and Vob, 2003) and job shop scheduling (Guinet and Legrand, 1998; Cheung and Zhou, 2001; Wang and Zheng, 2001; Schuster and Framinan, 2003). In flow shop scheduling, it is generally assumed that the jobs must be processed on the machines in the same technological or machine order. In job shop scheduling, however, jobs are commonly processed

following different machine orders (Laha and Chakraborty, 2009).

Flow shop scheduling has become one of the most popular scheduling problems due to its amazingly increasing practical usage in manufacturing industries (Nakhaeinejad and Nahavandi, 2012). The flow shop scheduling performs a set of jobs on a set of dedicated machines, where each job follows the same processing operation order. Each machine processes one job at a time and each job is processed on one machine at a time without pre-emption. The criterion that is most commonly studied in flow-shop scheduling literature is the minimization of total completion time, better to say makespan (C_{max}), of the production sequence. Researchers have optimised makespan in flow-shop problems by using various techniques like statistical tools, artificial neural network, fuzzy logic, genetic algorithm, simulated annealing, tabu search, etc. and combination of these /such techniques (Ruiz *et al.*, 2006; Zobolas *et al.*, 2009). Multi-Criteria Decision Making (MCDM) methods found their application in scheduling problems too. Techniques for Order Preference by Similarity to the Ideal Solution (TOPSIS) is such a technique which can be effectively used in scheduling decisions, however a comparatively less literature is observed on application of it in scheduling decision.

In this paper, efforts to bring out the application of TOPSIS in FSP with different weight factors in scheduling problems are made. The organisation of rest of the paper is as follows: review of literature relevant to flow shop scheduling is carried out in section 2. Procedure for computation of makespan, and TOPSIS are explained in section 3, while the implementation of the same on an illustrative problem is elaborated in section 4. Results are given and compared for same problem with different weight factors also, for different problems with different methods in section 5. Conclusions are drawn in section 6.

2. Literature Review

Literature pertaining to flow shop scheduling problem is studied in the following categories on the basis of problem solving techniques:

- I. Exact/Statistical methods
- II. Heuristic techniques
- III. Meta-heuristic techniques
- IV. Hybrid Heuristic/Meta-heuristic techniques.

The flow shop scheduling has been widely studied by the researchers because of its NP-completeness in the sense when number of machines is greater than three (Garey *et al.*, 1976; Gonzalez and Sahani, 1978, may be referred). Due to the complexity of the problem, exact methods for the general flow shop scheduling (FSP) failed to achieve high quality solutions for problems of large size in reasonable computational time. Therefore, academic research focused on heuristic approaches rather than exact methods to solve scheduling problems involving a large number of jobs (Gupta and Chauhan, 2015).

The flow-shop problem (FSP) was first studied by Johnson (1954) for two machines. Many researchers have generalized the Johnson's rule to 'm' machine flow shop problems (Ruiz and Stutzle, 2007). It is proved that m-machine flow shop problem with the makespan objective is NP-hard (Garey *et al.*, 1976; French, 1982). The first heuristic for makespan minimization for the flow shop scheduling problem was introduced by Palmer (1965). Subsequent work includes the one on the CDS heuristic (Campbell *et al.*, 1970) and Rapid access (RA) heuristic (Dannenbring, 1977). The NEH heuristic by Nawaz *et al.* (1983) was regarded as the best performing heuristic method (Turner and Booth, 1987; Ruiz and Maroto, 2005). More advanced methods are published by Koulamas (1998), Davoud Pour (2001), and Laha and Chakraborty (2009). Various heuristics with makespan as decision criterion have been reviewed by Hejazi and Saghafian (2005), and Ruiz and Maroto (2005).

In the pursuit for solutions closer to the optimum, it has become inevitable that new solution approaches should be followed by some difficult problem instances and academic interest switched to artificial intelligence based optimization methods, and meta-heuristics (Zobolas *et al.*, 2009). Simulated annealing based algorithms proposed by Osman and Potts (1989), and Ogbu and Smith (1990) are the first proposed meta-heuristics for the FSP. Widmer and Hertz (1989), Taillard (1990), Reeves (1993) and Nowicki and Smutnicki (1996) demonstrated different Tabu search approaches. Genetic algorithms for solving the FSP have also appeared in Chen *et al.* (1995), Reeves (1995), Wang and Zheng (2003), and Aldowaisan and Allahvedi (2003). Other algorithms are the path-based method of Werner (1993), the iterated local search of Stützle (1998), two very effective ant-colony optimization algorithms by Rajendran and Ziegler (2004), and a fast Tabu search approach of Grabowski and Wodecki (2004).

The results generated by meta-heuristics are found near to the best one/optimum, which provide the basis for researchers to develop advanced hybrid approaches, by combining different concepts or components of more than one meta-heuristic (Blum and Roli, 2003). Hybridization, when properly applied, may further enhance the effectiveness of the solution space search, and may overcome any inherent limitations of single meta-heuristic algorithms. Therefore, new opportunities emerge, which may lead to even more powerful and flexible solution methods for combinatorial optimization problems.

Even though, by the use of these methods the quality of solution is improved, but these methods are complex and iterative in nature, need special programming skills and require more computations to arrive a solution of flow shop problem. On the other hand, TOPSIS is a simple multi-criteria decision making techniques that provides the solution in few steps. The philosophy and procedure of TOPSIS could be understood easily. Further it does not require any special computation technique and advance mathematics for its implementation in a flow shop scheduling problem.

3. Methodology

The proposed work is an effort of implementation of pioneer work of Yoon and Hwang (1980) (called TOPSIS) in FSP. The methodology adopted in this paper is described in two subsections. In first subsection details of TOPSIS and its implementation steps in FSP for generation of job sequence whilst in second subsection the computation of makespan (i.e. total completion time) for selected job sequence is explained

3.1. Generation of job sequence by Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS developed by Yoon and Hwang (1980), is a simple and one of the most commonly utilized multi-criteria/ multi-attribute decision making (MCDM/ MADM) procedure for wide range of real world problems.

It helps decision makers to carryout comparisons, rankings and analysis among various available options in order to select a best one. (Behzadian *et al.*,

2012; Shih *et al.*, 2007; Vega *et al.*, 2014). It attempts to choose alternatives that simultaneously have the shortest distance from the positive ideal solution and the farthest distance from the negative-ideal solution. The positive ideal solution maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution maximizes the cost criteria and minimizes the benefit criteria (Behzadian *et al.*, 2012; Sarraf *et al.*, 2013). Thus, the best alternative, is the alternative having shortest relative distance from positive ideal solution and farthest distance from negative ideal solution. Researchers have developed large number of variants of TOPSIS by combining it with different distribution technique, for those (Behzadian *et al.*, 2012) may be referred. The step-wise implementation of Standard TOPSIS in flow shop scheduling problem is explained through a self-explanatory flow chart shown in Figure 1.

Recently, a heuristic based on the reduced weight-age scheme of machines to generate different combination of sequences for a FSP has been developed by Gupta and Chauhan, (2015). Similarly, here the weight factors used in TOPSIS are selected as per different weighting schemes to obtain a combination of job sequences in order to minimize the makespan.

3.2. Computation of makespan

In a flow-shop scheduling problem, a set of n jobs (1, ..., n) are processed on a set of m machines (1, ..., m) in the same technological order, i.e. first in machine 1 then on machine 2 and so on until machine m. The objective is to find a sequence for the processing of the jobs in the machines so that the total completion time or makespan of the schedule (Cmax) is minimized. Let t_{ij} denote the processing time of the job in position i (i = 1, 2, ..., n) on machine j (j = 1, 2, ..., m). Let C_{ij} represents the completion time of the job in position i on machine j. Therefore,

$$C_{i,1} = t_{i,1} \tag{1}$$

$$C_{i,1} = C_{i-1,1} + t_{i,1} \text{ for } i = 2, \dots, n \tag{2}$$

$$C_{i,j} = C_{i,j-1} + t_{i,j} \text{ for } j = 2, \dots, m \tag{3}$$

$$C_{i,j} = \max(C_{i,j-1}, C_{i-1,j}) + t_i > \text{ for } i = 2, \dots, n \text{ j} = 2, \dots, m \tag{4}$$

$$\text{Total Completion Time (Cmax)} = C_{n,m}$$

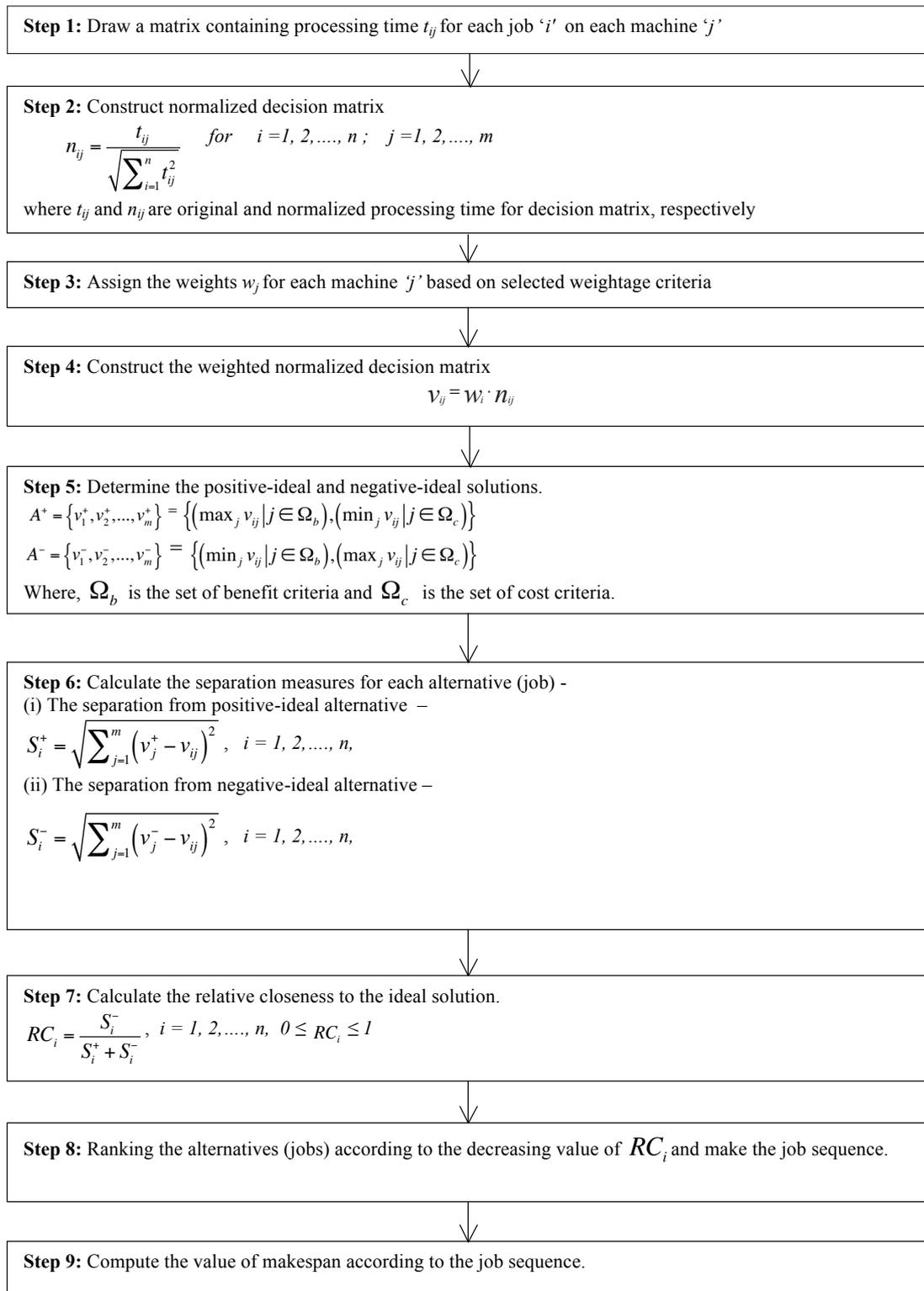


Figure 1. Steps for implementation of standard TOPSIS in flow shop scheduling.

4. Implementation with illustration

The proposed flow shop scheduling procedure is implemented on an arbitrarily designed flow shop scheduling problem illustrated below in subsection 4.1.

4.1. Illustrative problem

For illustration purpose, a flow shop scheduling problem of five jobs and five machines with random data has been developed and given in Table 1 as processing time matrix. Jobs and machines are represented in rows and columns of problem matrix respectively. Order of machines are pre-fixed i.e. each job has to be processed in a pre-defined sequence of machines. Each cell in the matrix represents the processing time of respective job on machine concerned. In order to minimize makespan (total job completion time), job sequence is to be determined.

Table 1. Processing time matrix for flow shop scheduling problem.

Jobs/Machines	M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	8	12	9	6	4
J ₂	14	10	11	2	15
J ₃	10	7	8	11	2
J ₄	7	8	14	9	2
J ₅	3	9	5	13	8

4.2. Implementation of standard TOPSIS in the above flow shop problem

Step 1: First of all, a problem matrix is introduced that consists of the processing time of each job on each machine (refer Table 1).

Step 2: The normalized decision matrix is constructed as per the procedure given in step 2 of Figure 1. It transforms the various attribute dimensions into non-dimensional attributes for the purpose of comparison across the attributes. The normalized matrix for the illustrative problem is given in Table 2.

Table 2. Normalized decision matrix.

Jobs/Machines	M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	0.39	0.57	0.41	0.30	0.23
J ₂	0.68	0.48	0.50	0.10	0.85
J ₃	0.49	0.33	0.36	0.54	0.11
J ₄	0.34	0.38	0.63	0.44	0.11
J ₅	0.15	0.43	0.23	0.64	0.45

Step 3: Weight factors ‘w_j’ are assigned for each machine ‘j’ based on selected weightage scheme. Here, the weight factors are selected based on the decreasing rank order of machine.

Step 4: Now, the weighted normalized decision matrix is constructed by multiplying the elements of normalized decision matrix by the relevant weight factor. Thus, obtained weighted normalized decision matrix for the illustration is given in Table 3.

Table 3. Weighted normalized decision matrix

Jobs/Machines	M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	1.96	2.29	1.22	0.59	0.23
J ₂	3.42	1.91	1.50	0.20	0.85
J ₃	2.45	1.34	1.09	1.09	0.11
J ₄	1.71	1.53	1.90	0.89	0.11
J ₅	0.73	1.72	0.68	1.28	0.45

Step 5: At this stage, the positive-ideal and negative-ideal solutions are determined as per the details given in Figure 1. Since the matrix (Table 3) is based on the processing times of jobs on respective machines, therefore, in order to minimize the total completion time of jobs (i.e. makespan), the minimum and maximum value in each column of the weighted normalized decision matrix is selected as positive-ideal (highlighted by bold letters in Table 4) and negative ideal (highlighted by bold letters in Table 5) solution respectively.

Table 4. Positive-ideal solution shown in weighted normalized decision matrix.

Jobs/Machines	M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	1.96	2.29	1.22	0.59	0.23
J ₂	3.42	1.91	1.50	0.20	0.85
J ₃	2.45	1.34	1.09	1.09	0.11
J ₄	1.71	1.53	1.90	0.89	0.11
J ₅	0.73	1.72	0.68	1.28	0.45

Positive ideal solution

$$A^+ = (0.73, 1.34, 0.68, 0.20, 0.11)$$

Table 5. Negative -ideal solution shown in weighted normalized decision matrix.

Jobs/Machines	M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	1.96	2.29	1.22	0.59	0.23
J ₂	3.42	1.91	1.50	0.20	0.85
J ₃	2.45	1.34	1.09	1.09	0.11
J ₄	1.71	1.53	1.90	0.89	0.11
J ₅	0.73	1.72	0.68	1.28	0.45

Negative ideal solution

$$A^- = (3.42, 2.29, 1.90, 1.28, 0.85)$$

Step 6: This stage requires the calculation of the separation measures from positive-ideal and negative-ideal solutions for each alternative (job) according to step 6 of Figure 1. The separation of each alternative (job) from positive ideal and negative ideal solution is tabulated in Table 6 and 7 respectively.

Table 6. Separation from positive-ideal solution.

Jobs	Separation from positive-ideal solution
J ₁	1.70
J ₂	2.97
J ₃	1.97
J ₄	1.72
J ₅	1.20

Table 7. Separation from negative-ideal solution.

Jobs	Separation from negative-ideal solution
J ₁	1.86
J ₂	1.22
J ₃	1.76
J ₄	2.05
J ₅	3.03

Step 7: Now, relative closeness to the ideal solution is calculated as per the details elaborated in step 7 of Figure 1. The same is presented in Table 8.

Table 8. Relative closeness to ideal solution for each alternative (job).

Jobs	Relative closeness to ideal solution (RC _i)
J ₁	0.52
J ₂	0.29
J ₃	0.47
J ₄	0.54
J ₅	0.72

Step 8: Jobs (alternatives) are ranked according to the decreasing value of relative closeness RC_i and job sequence are made to be processed on the pre-defined order of machines. From the above table, the job sequence J5-J4-J1-J3-J2 has been obtained.

Step 9: Finally, makespan value is computed based on the above job sequence using the equation 1-4.

Table 9. Comparison of makespan with different weightage scheme.

S. No.	Weightage for machines		Job sequence obtained through TOPSIS	Makespan (units)
	Weightage Scheme	Weight Factors		
1.	Decreasing order	5, 4, 3, 2, 1	J ₅ -J ₄ -J ₁ -J ₃ -J ₂	80
2.	Increasing order	1, 2, 3, 4, 5	J ₁ -J ₃ -J ₄ -J ₅ -J ₂	96
3.	Higher Machine utilization	3, 4, 5, 2, 1	J ₅ -J ₃ -J ₁ -J ₄ -J ₂	84
4.	Lower Machine utilization	3, 2, 1, 4, 5	J ₁ -J ₄ -J ₃ -J ₅ -J ₂	99
5.	Equal weightage	1, 1, 1, 1, 1	J ₁ -J ₃ -J ₄ -J ₅ -J ₂	96

The makespan value of 80 units is obtained for the illustrative problem, according to the job sequence generated in step 8.

5. Result comparison

Results are compared in two phases, as explained in section 5.1 and 5.2.

5.1. Comparison on the basis of different weighting scheme:

In this phase the TOPSIS procedure is repeated for the same illustrative problem for different weightage schemes (explained below) to obtain different makespan value.

I. Decreasing order – Under this criteria, machine having the first rank (or position), is given highest weightage which decreases with the number of rank increases.

II. Increasing order – Under this criteria, machine having the first rank (or position), is given lowest weightage which decreases with the number of rank increases.

III. Higher machine utilization – A machine with the highest load (or utilization), is given highest weightage, which decreases with decreasing the load.

IV. Lower machine utilization – A machine with the higher load (or utilization), is given lowest weightage which increases with decreasing the load.

V. Equal weightage – All the machines is assigned with the equal weights to process jobs.

Job sequences are generated according to above different weighting schemes, by using standard TOPSIS procedure. Then makespan is computed for each job sequence. The same is summarized in Table 9.

Selection of weighting scheme is one of key consideration in any TOPSIS application (Olson, 2004). In this problem, a minimum value of makespan is observed with decreasing order weight scheme. Makespan value for job sequence based on weighting scheme namely higher machine utilisation is close to minimum makespan value.

5.2. Result comparison with some standard methods

First, the above problem is solved by five well-known heuristic algorithms namely Palmer (Palmer, 1965), Gupta (Gupta, 1971), Campbell, Dudek and Smith (CDS) (Campbell *et al.*, 1970), Rapid Access (RA) (Dannenbring, 1977) and Nawaz, Ensore and Ham (NEH) (Nawaz *et al.*, 1983) makespan values of 85, 82, 82, 79 and 79 respectively are obtained. For results from TOPSIS based proposed method, all five weighting schemes discussed above are implemented on a particular problem, and the minimum value of those, presented here, is 80. Results are found to be compatible.

Secondly, for better comparison, results of some benchmark problems are compared with other well-known methods. These benchmark problems prescribed by J. Carlier (1978) and C.R. Reeves (1995) are taken from the OR-Library (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/flowshop1.txt>). 11 problem instances are taken for comparison purpose, 8 instances of Carlier and 3 instances of Reeves (1995). These problems are designed for the purpose of comparing the heuristic algorithms with an objective of makespan minimization. Details

of problems taken for comparison are shown in Table 10.

For above tabulated problems, minimum makespan time for the completion of the process is calculated based on the schedule derived by the respective algorithms. The results are presented in Table 11. The minimum figures across the algorithms are presented by bold letters in the table.

Table 10. Benchmark problems taken for result comparison.

S. No.	Size of the problems		
	Instances	No. of jobs	No. of machines
1.	Carlier 01	11	5
2.	Carlier 02	13	4
3.	Carlier 03	12	5
4.	Carlier 04	14	4
5.	Carlier 05	10	6
6.	Carlier 06	8	9
7.	Carlier 07	7	7
8.	Carlier 08	8	8
9.	ReC 01	20	5
10.	ReC 03	20	5
11.	ReC 05	20	5

The results indicate that introduction of TOPSIS method in the field of scheduling provides compatible results to some extent. For benchmark problems of 10x6 and 7x7, result of proposed method is closest to the least makespan values obtained by NEH and CDS respectively and also for other problems, makespan value is near about the least ones. Most of the minimum results obtained from proposed methods are obtained by using the decreasing order weighting scheme.

Table 11. Comparison of makespan time obtained by TOPSIS based proposed method with other standard heuristics.

S. No.	Problem Instances	Makespan (time measurable unit)					
		Proposed Method	Palmer	Gupta	CDS	RA	NEH
1.	Carlier01-11x5	7332	7472	7348	7202	7817	7038
2.	Carlier02-13x4	8123	7940	7534	7410	7509	7940
3.	Carlier03-12x5	8567	7725	7399	7399	7399	7503
4.	Carlier04-14x4	9170	8423	8423	8423	8357	8003
5.	Carlier05-10x6	8309	8520	8773	8627	8940	8190
6.	Carlier06-8x9	9647	9487	9441	9553	9514	9159
7.	Carlier07-7x7	7563	7639	7639	6819	6923	7668
8.	Carlier08-8x8	9345	9023	9224	8903	9062	9032
9.	ReC01-20x5	1595	1391	1434	1399	1399	1334
10.	ReC03-20x5	1289	1223	1380	1273	1159	1136
11.	ReC05-20x5	1479	1290	1429	1338	1434	1294

6. Conclusion

This paper successfully presented a new approach based on a MCDM method called 'TOPSIS' for minimization of makespan criterion in flow shop scheduling. The technique demonstrated is simple, easy to understand, and implement. It has been illustrated by using five different weighting schemes to generate the job sequences. The makespan

value obtained from these sequences shows the compatibility of this technique in flow shop environment. The results can further be improved by incorporating a weighting scheme which could consider machine utilization and other production related parameters. The proposed procedure could also be implemented for other scheduling problems such as job shop, flexible job shops and flow shops.

References

- Albadawi, Z., Bashir, H.A., Chen, M. (2005). A mathematical approach for the formation of manufacturing cells. *Computers & Industrial Engineering*, 48(1): 3–21. <http://dx.doi.org/10.1016/j.cie.2004.06.008>
- Aldowaisan, T., Allahvedi, A. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*, 30(8): 1219–31. [http://dx.doi.org/10.1016/S0305-0548\(02\)00068-0](http://dx.doi.org/10.1016/S0305-0548(02)00068-0)
- Behzadian, M., Otaghsara, S., Yazdani, M., Ignatius, J. (2012). A state-of-the-art survey of TOPSIS applications. *Expert Systems with Applications*, 39(17), 13051–13069. <http://dx.doi.org/10.1016/j.eswa.2012.05.056>
- Blum, C., Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308. <http://dx.doi.org/10.1145/937503.937505>
- Campbell, H.G., Dudek, R.A., Smith, M.L. (1970). A heuristic algorithm for the n -job, m -machine problem. *Management Science*, 16(10), B630–7. <http://dx.doi.org/10.1287/mnsc.16.10.B630>
- Carlier, J. (1978). Ordonnements a contraintes disjonctives, R.A.I.R.O. *Recherche operationelle/Operations Research*, 12(4), 333–351.
- Chen, C-L., Vempati, V.S., Aljaber, N. (1995). An application of genetic algorithms for flow shop problems. *European Journal of Operational Research*, 80(2), 389–396. [http://dx.doi.org/10.1016/0377-2217\(93\)E0228-P](http://dx.doi.org/10.1016/0377-2217(93)E0228-P)
- Cheung, W.M., Zhou, H. (2001). Using genetic algorithms and heuristics for job shop scheduling with sequence-dependent setup times. *Annals of Operations Research*, 107(1), 65–81. <http://dx.doi.org/10.1023/A:1014990729837>
- Dannenbring, D.G. (1977). An evaluation of flow-shop sequencing heuristic. *Management Science*, 23(11), 1174–1182. <http://dx.doi.org/10.1287/mnsc.23.11.1174>
- Davoud Pour, H. (2001). A new heuristic for the n -job, m -machine flow-shop problem. *Production Planning and Control*, 12(7), 648–53. <http://dx.doi.org/10.1080/09537280152582995>
- Fink, A., Vob, S. (2003). Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, 151(2), 400–414. [http://dx.doi.org/10.1016/S0377-2217\(02\)00834-2](http://dx.doi.org/10.1016/S0377-2217(02)00834-2)
- French, S. (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Ellis Horwood Limited.
- Garey, M.R.D., Johnson, D.S., Sethi, R. (1976). The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*, 1, 117–29. <http://dx.doi.org/10.1287/moor.1.2.117>
- Gonzalez, T., Sahni, S. (1978). Flow shop and job shop schedules. *Operations Research*, 26, 36–52. <http://dx.doi.org/10.1287/opre.26.1.36>
- Grabowski, J., Wodecki, M. (2004). A very fast tabu search algorithm for the permutation flowshop problem with makespan criterion. *Computers and Operations Research*, 31(11), 1891–909. [http://dx.doi.org/10.1016/S0305-0548\(03\)00145-X](http://dx.doi.org/10.1016/S0305-0548(03)00145-X)
- Guinet, A., Legrand, M. (1998). Reduction of job-shop problems to flow-shop problems with precedence constraints. *European Journal of Operational Research*, 109(1), 96–110. [http://dx.doi.org/10.1016/S0377-2217\(97\)00129-X](http://dx.doi.org/10.1016/S0377-2217(97)00129-X)
- Gupta, A., Chauhan, S. (2015). A heuristic algorithm for scheduling in a flow shop environment to minimize makespan. *International Journal of Industrial Engineering Computations*, 6(2), 173–184. <http://dx.doi.org/10.5267/j.ijiec.2014.12.002>
- Hejazi, S.R., Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research*, 43(14), 2895–2929. <http://dx.doi.org/10.1080/0020754050056417>
- Johnson, S.M. (1954). Optimal two and three-stage production schedules with set-up times included. *Naval Research Logistics Quarterly*, 1(1), 61–68. <http://dx.doi.org/10.1002/nav.3800010110>
- King, J.R., Spachis, A.S. (1980). Heuristics for flowshop scheduling. *International Journal of Production Research*, 18(3), 343–357. <http://dx.doi.org/10.1080/00207548008919673>
- Koulamas, C. (1998). A new constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research*, 105(1), 66–71. [http://dx.doi.org/10.1016/S0377-2217\(97\)00027-1](http://dx.doi.org/10.1016/S0377-2217(97)00027-1)
- Kumar, S., Sharma, R.K. (2014). Cell formation heuristic procedure considering production data. *International Journal of Production Management and Engineering*, 2(2), 75–84. <http://dx.doi.org/10.4995/ijpme.2014.2078>

- Kumar, S., Sharma, R.K. (2015). Development of a cell formation heuristic by considering realistic data using principal component analysis and Taguchi's method. *Journal of Industrial Engineering International*, 11(1), 87-100. <http://dx.doi.org/10.1007/s40092-014-0093-3>
- Laha, D., Chakraborty, U.K. (2009). An efficient hybrid heuristic for makespan minimization in permutation flow shop scheduling. *International Journal of Advanced Manufacturing Technology*, 44(5), 559-569. <http://dx.doi.org/10.1007/s00170-008-1845-2>
- Nakhaeinejad, M., Nahavandi, N. (2012). An interactive algorithm for multi-objective flow shop scheduling with fuzzy processing time through resolution method and TOPSIS. *International Journal of Advanced Manufacturing Technology*, 66(5), 1047-1064. <http://dx.doi.org/10.1007/s00170-012-4388-5>
- Nawaz, M., Enscore, Jr. E., Ham, I. (1983). A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem. *Omega, International Journal of Management Science*, 11, 91-95.
- Nowicki, E., Smutnicki, C. (1996). A fast tabu search algorithm for the permutation flow-shop problem. *European Journal of Operational Research*, 91(1), 160-75. [http://dx.doi.org/10.1016/0377-2217\(95\)00037-2](http://dx.doi.org/10.1016/0377-2217(95)00037-2)
- Ogbu, F.A., Smith, D.K. (1990). The application of the simulated annealing algorithms to the solution of the $n/m/C_{max}$ flowshop problem. *Computers & Operations Research*, 17(3), 243-53. [http://dx.doi.org/10.1016/0305-0548\(90\)90001-N](http://dx.doi.org/10.1016/0305-0548(90)90001-N)
- Olson, D.L. (2004). Comparison of weights in TOPSIS models. *Mathematical and Computer Modeling*, 40(7-8), 721-727. <http://dx.doi.org/10.1016/j.mcm.2004.10.003>
- Osman, I.H., Potts, C.N. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega, The International Journal of Management Science*, 17(6), 551-7.
- Palmer, D.S. (1965). Sequencing jobs through a multistage process in the minimum total time: a quick method of obtaining a near optimum. *Operations Research Quarterly*, 16(1), 101-7. <http://dx.doi.org/10.1057/jors.1965.8>
- Rajendran, C., Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2), 426-38. [http://dx.doi.org/10.1016/S0377-2217\(02\)00908-6](http://dx.doi.org/10.1016/S0377-2217(02)00908-6)
- Reeves, C.R. (1993). Improving the efficiency of tabu search for machine scheduling problems. *Journal of the Operational Research Society*, 44(4), 375-82. <http://dx.doi.org/10.2307/2584415>
- Reeves, C.R., (1995). A genetic algorithm for flowshop sequencing. *Computers and Operations Research* 22(1), 5-13. [http://dx.doi.org/10.1016/0305-0548\(93\)E0014-K](http://dx.doi.org/10.1016/0305-0548(93)E0014-K)
- Ruiz, R., Maroto, C., Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. *Omega, International Journal of Management Science*, 34, 461-76. <http://dx.doi.org/10.1016/j.omega.2004.12.006>
- Ruiz, R., Stutzle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3), 2033-2049. <http://dx.doi.org/10.1016/j.ejor.2005.12.009>
- Ruiz, R., Maroto, C. (2005). A comprehensive review and evaluation of permutation flow shop heuristics. *European Journal of Operational Research*, 165(2), 479-494. <http://dx.doi.org/10.1016/j.ejor.2004.04.017>
- Sarraf, A.Z., Mohaghar, A., Bazargani, H. (2013). Developing TOPSIS method using statistical normalization for selecting Knowledge management strategies. *Journal of Industrial Engineering and Management*, 6(4), 860-875. <http://dx.doi.org/10.3926/jiem.573>
- Schuster, C.J., Framinan, J.M. (2003). Approximate procedures for no wait job shop scheduling. *Operations Research Letters*, 31(4), 308-318. [http://dx.doi.org/10.1016/S0167-6377\(03\)00005-1](http://dx.doi.org/10.1016/S0167-6377(03)00005-1)
- Shih, H.S., Syur, H.J., Lee, E.S. (2007). An extension of TOPSIS for group decision making. *Mathematical and Computer Modeling*, 45(7-8), 801-813. <http://dx.doi.org/10.1016/j.mcm.2006.03.023>
- Stützle, T. (1998). *Applying iterated local search to the permutation flowshop problem*. Technical Report, AIDA-98-04, TU Darmstadt, FG Intellektik;
- Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47(1), 65-74. [http://dx.doi.org/10.1016/0377-2217\(90\)90090-X](http://dx.doi.org/10.1016/0377-2217(90)90090-X)
- Turner, S., Booth, D., (1987). Comparison of heuristics for flow shop sequencing. *OMEGA, The International Journal of Management Science* 15(1), 75-78. [http://dx.doi.org/10.1016/0305-0483\(87\)90054-5](http://dx.doi.org/10.1016/0305-0483(87)90054-5)
- Vega, A., Aguaron, J., García-Alcaraz, J., Moreno-Jiménez, J.M. (2014). Notes on Dependent Attributes in TOPSIS. *Procedia Computer Science*, 31, 308 - 317. <http://dx.doi.org/10.1016/j.procs.2014.05.273>
- Wang, L., Zheng, D. (2001). An effective hybrid optimization strategy for job-shop scheduling problems. *Computers & Operations Research*, 28(6), 585-596. [http://dx.doi.org/10.1016/S0305-0548\(99\)00137-9](http://dx.doi.org/10.1016/S0305-0548(99)00137-9)
- Wang, L., Zheng, D.Z. (2003). An effective hybrid heuristic for flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 21(1), 38-44. <http://dx.doi.org/10.1007/s001700300005>
- Werner, F. (1993). On the heuristic solution of the permutation flow shop problem by path algorithms. *Computers and Operations Research* 20(7), 707-722. [http://dx.doi.org/10.1016/0305-0548\(93\)90058-Q](http://dx.doi.org/10.1016/0305-0548(93)90058-Q)
- Widmer M, Hertz A. (1989). A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*, 41(2), 186-93. [http://dx.doi.org/10.1016/0377-2217\(89\)90383-4](http://dx.doi.org/10.1016/0377-2217(89)90383-4)

Gupta, A. and Kumar, S.

Yoon, K., Hwang, C.L. (1980). *Multiple Attribute Decision Making Methods and Applications. A State of the Art Survey*. Berlin: Springer Verlag.

Zobolas, G.I., Tarantilis, C.D., Ioannou, G. (2009). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research* 36(4), 1249-1267. <http://dx.doi.org/10.1016/j.cor.2008.01.007>