ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20200208192653
Original scientific paper

An Integrated Solution Approach for Flow Shop Scheduling

Ilknur KARACAN*, Ismet KARACAN, Ozlem SENVAR, Serol BULKAN

Abstract: This study seeks to integrate Random Key Genetic Algorithm (RKGA) and Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) to compute makespan and solve the Flow Shop Scheduling Problem (FSSP). FSSP is considered as a Multi Criteria Decision Making Problem (MCDM) by setting machines as criteria and jobs as alternatives. RKGA is employed to determine the best weights for the criteria that directly affect the robustness of the solution. The proposed methodology is presented with illustrative example and applied to benchmark problems. The solutions are compared to well-known construction heuristics. The proposed methodology provides the best or reasonable solutions in acceptable computational times.

Keywords: flow shop scheduling; multicriteria decision making; random key genetic algorithm; technique for order preference by similarity to an ideal solution

1 INTRODUCTION

Scheduling is a systematic approach for planning execution of activities using assets under defined constraints to satisfy defined objectives. It is adopted to solve many types of production and service problems and has many real-world applications [1, 2]. Sequencing tasks over a CPU, crane operations, railway operations etc. are common examples. Scheduling has also a wide range of application in manufacturing from supply chain activities to production operations [3, 4]. From manufacturing point of view, scheduling seeks for an optimum execution sequence of given jobs on a set of machines to optimize defined objective(s) such as makespan, flow time, tardiness, idle time, etc. Scheduling function deals with the determination of time-sequence of jobs, orders, tasks, and operations as well as the allocation of the required resources to accomplish the related set of jobs, orders, operations, and tasks. The Flow Shop Scheduling Problem (FSSP) seeks an order of execution for defined n *j*obs at m machines with an unchanging path on machines by satisfying defined objective function(s). A single job can be processed concurrently only on one machine and likewise a single machine can process only one job at a time. Most common and widely studied objective function for FSSP is known as makespan [5, 6].

FSSP is a complex problem in its nature. A FFSP with makespan objective on two machines ($F_2||C_{max}$) can be solved in polynomial time [7]. However, the problem turns into a minimal NP-Hard problem if the number of machines is increased by only one ($F_3||C_{max}$) [8]. The problem is NP-Complete where number of machines is greater than three [9]. At these circumstances, no algorithm is known to solve the problem optimally in polynomial time. Numerous algorithms are proposed to provide reasonable solutions to this hard problem. Approximation algorithms in literature prove the complexity to achieve favorable solutions [10]. Hence, many hybrid or integrated algorithms are suggested to obtain better solutions in less runtime. The combinations involve hybridization or integration of metaheuristics, mathematical programming models and metaheuristics, construction heuristics and metaheuristics, construction heuristics and mathematical programming models, hyper heuristics, etc. In this context, construction heuristics assumes an important role by providing initial solutions and parameters for further steps.

In the literature, many construction heuristics have been proposed for providing favorable initial solutions for metaheuristic algorithms [11]. However, combination of Multi Criteria Decision Making Problem (MCDM) methods and metaheuristic algorithms rarely appear to solve FSSP. In this study, a MCDM process is proposed as a construction heuristic. The proposed methodology involves integration of Random Key Genetic Algorithm (RKGA) and Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method to solve the FSSP and provide a robust initial solution. TOPSIS is used to calculate the makespan and RKGA is deployed to hint weights required by TOPSIS. To the best of our knowledge, integration of RKGA and TOPSIS for computation of makespan and solution of FSSP has not been handled in the literature yet. There are studies that apply MCDM methods to solve FSSP problems in the literature. However, lack of optimized criteria weights avoids MCDM algorithms to achieve favorable solutions. Apart from studies in the literature, the proposed algorithm in this study calculates the best criteria weights automatically and provides favorable solutions.The rest of the study is organized as follows: In section 2, literature review is summarized including TOPSIS and RKGA. In section 3, methodology is explained. In section 4, application and computational results are presented. In the final section, conclusion and recommendations are given.

2 LITERATURE REVIEW

According to Ruiz and Stützle [12], a flow shop consists of n jobs and m machines, where all jobs are processed in the same order. More clearly, jobs follow the same processing order from Machine 1 to Machine m. The aim is to find processing sequence of jobs that optimizes the objective function(s).

FSSP is a well-defined real-life problem and has many application areas. However, the problem still preserves its complexity. Many algorithms are proposed to solve the problem for providing an initial solution or an exact solution. Rather than dealing with complex algorithms, actual interlocutors of the problem tend to handle the problem solution methods that contain easy implementation and user-friendly interface. Thus, many construction heuristics are introduced to literature that promise comprehensive algorithms and acceptable solutions. NEH (Nawaz, Enscore and Ham) [13] algorithm is known as the best polynomial time algorithm to provide convenient solutions in practice [14]. In addition to NEH algorithm, algorithms like Palmer [15], Gupta [16], Campbell, Dudek and Smith (CDS) [17] and Rapid Access (RA) [18] are prominent methods that appear in many studies and real-life applications. This study proposes an analogous construction heuristic by integrating the popular MCDM method TOPSIS with RKGA.

2.1 MCDM

Concluding to a decision for real life problems may require evaluation from many aspects. In many cases, decision makers are forced to peruse many possible choices, reveal their strengths and weaknesses as per defined measures and choose the best candidate. With a scientific focus, such a complex process is called as MCDM. To solve a MCDM problem, decision makers should compare available alternatives according to specified criteria and conclude with the best decision. Many MCDM solution algorithms are proposed to simplify the decision process. However, minority of them are broadly acknowledged and practiced. Weighted Sum Model (WSM) and its extension Weighted Product Sum are among foremost algorithms and have great reputation [19]. Famous Analytical Hierarchical Process (AHP) by Saaty [20] allowed revealing mutual pros and cons of alternatives according to defined scales by pairwise comparisons with group decision making capability [21]. A search for popular MCDM methods returns algorithms like TOPSIS by Hwang and Yoon [22], ELECTRE by Benayoun [23] and its novel variants, Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) by Brans [24] and its family, Data Envelopment Analysis by Charnes [25], Goal Programming by Charnes [26], etc.

2.2 TOPSIS

Hwang and Yoon [22] introduced TOPSIS in 1981 as an alternative MCDM method. According to this method, alternatives are evaluated by their distances to Positive Ideal Solution (PIS) and Negative Ideal Solution (NIS). Thus, alternatives are aligned according to their distances to PIS and NIS. The most preferable alternative would be the one that is closest to PIS and naturally farthest from the NIS, where PIS contains the best achievable values for all criteria and NIS contains the worst values. Rather than examining k-dimensional criteria vs. alternatives, TOPSIS enables to figure relative sequence of alternatives with 2-dimensional distances. The steps and notation of TOPSIS algorithm are described in methodology section.

TOPSIS has many real-world applications since it can be simply customized and adapted to fit many decision problems. Chamodrakas et al. [27] evaluated customer selection process where suppliers produce make-to-order [MTO] rather than make-to-stock [MTS] with an agile manufacturing policy to compete with their opponents and satisfy part of their customers considering their capacity. Goyal et al. [28] recommended a tuning model for configuring reconfigurable manufacturing system. Karatas et al. [29] assessed energy management performance of Turkey by an integrated model. Gungor and Kocamis [30] evaluated financial performance of UK soccer clubs. The samples may be diversified by surveying the studies in the literature. This study utilizes TOPSIS conducive to assigning process order of jobs to minimize makespan.

2.3 RKGA

Bean [31] introduced Random Key Genetic Algorithm (RKGA) as a variant of Genetic Algorithm. The algorithm is aimed to solve a wide sort of sequencing and general optimization problems. The method proposes a novel technique for representation to handle infeasibility issues of Genetic Algorithm that occurs during transition from parent to offspring. This novel technique, namely random keys, has an inherent coding system by demonstrating the solution via random numbers. Sorting the vector of these random numbers composes a natural encode/decode system. As advantage of the algorithm, searching the space by random numbers provides a robust exploration which cannot be achieved by binary numbers. Another advantage is the feasibility of crossover operation. Crossover is carried out over random numbers, not on sequential numbers. Hence, no additional logic is required to keep the feasibility of emergent chromosome.

RKGA still preserves its value today and many researchers benefit from its robust structure in various studies. Leonhart et al. [32] implied biased version to solve protein-ligand docking problem where molecular docking is a valuable tool for drug discovery. Ruiz et al. [33] implemented biased model to overcome the complexity of open vehicle routing problem with capacity and distance constraints. Faria et al. [34] studied electric distribution network reconfiguration problem that is widely studied in power system analysis. Famous Travelling Salesman Problem (TSP) is extensively handled with RKGA. Samanlioglu et al. [35] suggested a memetic RKGA to solve symmetric multi-objective TSP. Andrade et al. [36] utilized a biased version to solve *k*-Interconnected Multi-Depot Multi TSP that is a novel problem where k-terminal vertices are selected from base graph and connected with a cycle. A detailed survey and notation for the algorithm has been reviewed by Gonçalves and Resende [37].

RKGA is also applied to scheduling problems. Gonçalves et al. [38] deployed a hybrid genetic algorithm to solve Job Shop Scheduling Problem in which chromosomes are represented by random keys. Lei [39] addressed an approach for solving Job Shop Scheduling Problem with fuzzy processing times. Dauzère-Pérès and Mönch [40] proposed two mixed integer linear programming models for the weighted and unweighted number of tardy jobs on a single batch processing machine with incompatible job family problems and deployed RKGA methodology to solve these problems.

In this paper, a variant of RKGA is applied to supply best weights for TOPSIS algorithm rather than solving whole FSSP problem.

2.4 Solving FSSP with MCDM

As stated in literature review section, MCDM algorithms may be customized to solve many types of real-life problems. An example for this customization is the study by Subramaniam [41]. Rather than focusing on objective function(s), the study deals with dynamical selection of the best dispatching rule by Analytic Hierarchy Process AHP Method so that objection function(s) is/are ideally satisfied. A similar study is proposed by Singh [42] for multi-criteria dynamic job shop scheduling problems by integration of TOPSIS and goal programming. Leu and Yang [43] adopted an integration of Genetic Algorithm and TOPSIS, in which Genetic Algorithm processes overall calculations to solve the problem and TOPSIS strives to balance multiple cost and time consumption objectives. Kiran Kumar [44] integrated AHP and TOPSIS methods, where AHP calculates weights of the criteria (objective functions) according to the assessments of decision makers and supplies the weights to TOPSIS method, and TOPSIS sorts priority dispatching rules according to their ranks. In their study, Jadhav and Bajaj [45] had a different approach, so that TOPSIS is determining the processing orders jobs. Their featured assumption was efficiency of machines. As criteria of TOPSIS method, machines have been weighted according to their efficiency. Nakhaeinejad and Nahavandi [46] integrated interactive resolution and TOPSIS method to solve multi objective FSSP in fuzzy environment. The purpose of integrating TOPSIS was to handle multiple objectives more efficiently so that they are ensured to be satisfied in balance. Lin [47] integrated AHP and Genetic Algorithm to solve FSSP with re-entrant jobs to satisfy multi criteria of the problem and expedite convergence to a near optimal solution by better chromosome selection. Rohaninejad [48] presented a nonlinear model for capacitated flexible job shop scheduling problem with multiple objectives and tried to solve the problem with integration of Genetic Algorithm and ELECTRE. Braglia and Grassi [49] announced a novel algorithm, namely MOGI algorithm, by integrating NEH Heuristic and TOPSIS to optimize FSSP with makespan and maximum tardiness objective functions.

Gupta and Kumar [50] tried to solve the problem solely with TOPSIS. In their study, machines represent criteria and jobs stand for alternatives. As an output, alternatives are sorted according to their scores which indicates the order of production to provide minimum makespan. They also applied several weight schemes to obtain better results. However, the results are not robust enough to outrival known algorithms.

From this standpoint, FSSP is considered as MCDM problem by setting machines as criteria and jobs as alternatives in this study. RKGA is conducted to determine the best weights for the criteria that directly affect the robustness of the solution.

3 METHODOLOGY

Intuitive solution approaches to hard problems provide reasonable solutions in acceptable computational times. Many of these approaches iteratively update initial solutions that are supplied for the corresponding problem. Finding makespan for FSSP with more than two machines are examples of NP-Hard problems. Many construction heuristics are proposed to provide favorable initial solutions for metaheuristic algorithms. In this study, a MCDM process is proposed as a constructive heuristic. The proposed methodology is illustrated in Fig. 1.

This study integrates RKGA and TOPSIS method to solve the single objective FSSP. FSSP is considered as decision making problem in which optimum sequence is determined for processing order of jobs. MCDM algorithms comprise the selection of best alternative(s) from a set of alternatives according to a given set of criteria. A MCDM process has three components, which are alternatives, criteria, and weights.

Alternatives are any collection of selectable items, objects, actions, etc. In this case, jobs to be processed are alternatives in FSSP and ranking order of alternatives in MCDM methods is the sequence, which is required to calculate the makespan.

Criteria are the factors that influence the selection of alternatives. The criteria should affect evaluations that may yield to an objective ranking of alternatives. For the considered FSSP, the criteria are derived from the machines. The characteristic of FSSP has a comprehensive assumption that is valid for most of the problems in the class. To minimize the makespan in FSSP, the best approach is to allocate the jobs with longer process times on earlier machines and shorter process times on the following machines toward the end of the sequence [51]. The idea behind the rule is quite simple. The jobs with shorter process times on earlier machines rapidly clear the machines for the upcoming jobs. Hence, the number of jobs in the queue is reduced to avoid unnecessary delays. The criteria for FSSP are the machines. The machines are grouped into two equal partitions so that the machines, which are visited earlier, are included in the first partition. It is obvious that the process times for machines in first partition should be minimized and the ones in second partition are to be maximized.

Weights are assigned to criteria representing importance of each criterion. Usually, the weights symbolize the priorities of decision-makers. In fact, FSSP is dealing with makespan rather than the weights, several parameters and algorithms are tried to find appropriate weights. The best alternative between these approaches is a modification of RKGA. Typical RKGA uses randomly generated keys to encode and decode solutions. It fits well to problems in which solutions are permutation of "integers" [52]. However, weights of a MCDM problem are fractions rather than sequences.

In this study, random keys are not encoded/decoded. Instead, the random keys are used as solutions without any conversion. Thus, every criterion (machine) has its own weight within interval [0, 1). This interval provides adequate distinction between the values of weights. In addition, all three RKGA operators (elitism, crossover and new generation) described by Bean [31] are used for our proposed methodology.

Positive and negative ideal solutions for all alternatives are calculated as output of TOPSIS algorithm. Determination of closeness to the ideal solution is performed. Then, ranking the alternatives according to relative closeness values in descending order is completed. This order provides the sequence of jobs. The fitness value of a chromosome in RKGA is the makespan calculated using this sequence.

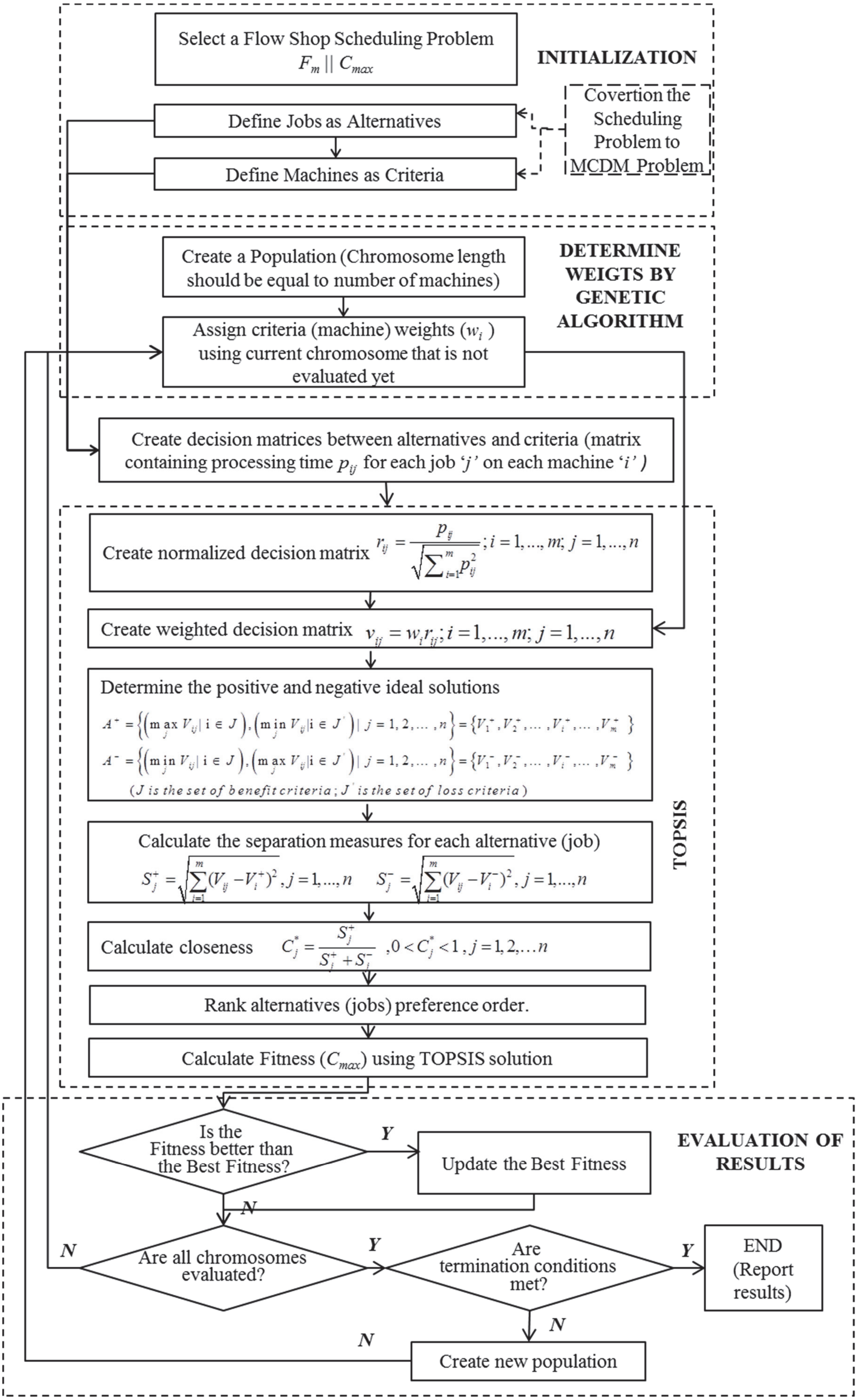

Figure 1 The Flowchart of Proposed Methodology

The proposed methodology requires nxm number of TOPSIS runs and makespan calculation with different weights provided by RKGA, where n is the number of generations, m is the new chromosomes (m excludes the number of elite chromosomes, which are directly relocated to the new generation). Upon termination of RKGA, the sequence with the best (minimum) fitness (makespan) is returned as solution.

The steps of the proposed methodology are clarified in the following items:

Step 1: Define the FSSP with $F_m \| C_{\max}$ form:

$J_j \in J$ represents j^{th} Job to be processed in the Jobs cluster with n Jobs where $j = \{1,..,n\}$.

$M_i \in M$ represents i^{th} Machine available for processing in the Machines cluster with m Machines where $i = \{1,..,m\}$.

p_{ij} represents processing time of Job J_j on Machine M_i.

C_{ij} represents completion time of Job J_j on Machine M_i

$C_{\max} = \overset{\max}{1..n}\{C_{mj}\}$ maximum completion time (makespan)

Step 2: Define Jobs of the FSSP as Alternatives of MCDM Problem.

Step 3: Define Machines of the FSSP as Criteria of MCDM Problem.

Step 4: Create decision matrix by processing times.

$$\begin{pmatrix} p_{11} & \cdot & p_{i1} \\ \cdot & \cdot & \cdot \\ p_{1j} & \cdot & p_{ij} \end{pmatrix} \tag{1}$$

Step 5: Create normalized decision matrix.

$$\begin{pmatrix} r_{11} & \cdot & r_{i1} \\ \cdot & \cdot & \cdot \\ r_{1j} & \cdot & r_{ij} \end{pmatrix} \tag{2}$$

where

$$r_{ij} = \frac{p_{ij}}{\sqrt{\sum_{i=1}^{m} p_{ij}^2}}; i = 1,..,m; j = 1,..,n \tag{3}$$

Step 4: Create chromosomes of the RKGA in the form of decimal vector with m elements.

$$Ch_k = [dec_1, dec_2, .., dec_m] \tag{4}$$

Step 5: Select an unevaluated chromosome Ch_k from population. Set the chromosome as the next weight to be evaluated in MCDM.

$$Ch_k = w_k = [w_{1k}, .., w_{mk}] \tag{5}$$

Step 6: Calculate weighted normalized decision matrix.

$$\begin{pmatrix} v_{11} & \cdot & v_{i1} \\ \cdot & \cdot & \cdot \\ v_{1j} & \cdot & v_{ij} \end{pmatrix} \tag{6}$$

where $v_{ij} = r_{ij} \times w_{ik}$

Step 7: Determine negative ideal and positive ideal solutions.

$$\begin{aligned} A^+ &= \{(\max_j v_{ij} \mid j \in J^+), (\min_j v_{ij} \mid j \in J^-) \mid j = 1,..,n\} \\ A^+ &= (V_1^+, .., V_m^+) \end{aligned} \tag{7}$$

$$\begin{aligned} A^- &= \{(\min_j v_{ij} \mid j \in J^+), (\max_j v_{ij} \mid j \in J^-) \mid j = 1,..,n\} \\ A^- &= (V_1^-, .., V_m^-) \end{aligned} \tag{8}$$

Step 8: Calculate the separation measures for each Job.

$$S_j^+ = \sqrt{\sum_{i=1}^{m} (v_{ij} - v_i^+)^2}, j = 1,..n \tag{9}$$

$$S_j^- = \sqrt{\sum_{i=1}^{m} (v_{ij} - v_i^-)^2}, j = 1,..n \tag{10}$$

Step 9: Calculate closeness to positive ideal solution.

$$C_j^* = \frac{S_j^+}{S_j^+ + S_j^-} \tag{11}$$

Step 10: Order the Jobs according to their closeness value in ascending order.

Step 11: Calculate fitness $(C_{\max})$.

$$C_{\max} = \overset{\max}{1..n}\{C_{mj}\} \tag{12}$$

Step 12: The evaluation steps of the algorithm can be seen in Fig. 1.

4 APPLICATION AND COMPUTATIONAL RESULTS

The proposed algorithm was developed as well as implemented and compared to popular construction heuristic algorithms via C#. The weights of TOPSIS method is normally provided manually by the decision maker. Unlike the studies in the literature, the weights are determined by using a genetic algorithm with random key coding in this study. The purpose of this approach is to evolve the weights to converge to the best sequence of the jobs so that minimum makespan value can be achieved. RKGA depends on several parameters in parallel to many other stochastic algorithms in which these parameters affect the evolution, thereby the result of the algorithm. The preferred evolution provides smart evaluation of promising regions of the solution space and ensures a smooth convergence that prevents rapid sticking to local optima. Parameters suggested by Bean [31] for RKGA have been slightly changed in this methodology. Since this study uses directly random keys, generation of new decimals gains more importance. Hence, generation percentage of new chromosomes is updated as 10% instead of generating 1% at each iteration. Percentage of Elite Operator that delegates elite chromosomes to new population keeps its percentage as 20%. Percentage of Crossover Operator that creates new offspring from existing population is decreased to 70% of new offspring except existing chromosomes. Another challenging parameter is gene inheritance probability for crossover operator. Instead of a fair selection, a biased inheritance of genes selected from a more fit chromosome to a new offspring is preferred to form better populations. This probability value is set to 70% for better chromosomes.

Tab. 1 assures convenient solutions that lead to converged makespan values for this study.

Termination criteria and number of chromosomes in population are not fixed but changing according to the size of the problem.

Table 1 Parameters for RKGA

Termination Criteria	100-5000 new generations
# of Chromosomes	100
Elite operator	20%
Crossover operator	70%
New Chromosome Operator	10%
Gene Inheritance Probability	70%
Crossover	Random
# of offspring from a crossover operation	Single

4.1 Illustrative Example

This part includes a small-sized illustrative example with 5 jobs and 5 machines. For benchmark purposes, the considered problem is taken from the paper of Gupta and Kumar [50]. The processing times of jobs on machines are shown in Tab. 2. Since the problem is a permutation FSSP, no further problem data is required.

Step-1: In this step, the RKGA is initialized by predetermined parameters and initial population is created. A typical chromosome of the population consists of 5 real valued numbers from interval [0, 1). An illustrative chromosome that will be used as weights (w_k) within example is [0.841, 0.556, 0.021, 0.516, 0.536].

Table 2 Processing Time Matrix

J/M	M_1	M_2	M_3	M_4	M_5
J_1	8	12	6	9	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

Step-2: Each chromosome should be evaluated with the fitness function. The procedure of evaluation in this methodology contains finding a job processing order using TOPSIS and calculating makespan.

Step-2.a: The processing times should be normalized as the first step of TOPSIS algorithm. Normalization assists scaling each dimensional attribute vectors into new vectors that are in the scale. Normalized matrix is demonstrated in Tab. 3.

Table 3 Normalized Matrix

J/M	M_1	M_2	M_3	M_4	M_5
J_1	0,391	0,573	0,408	0,296	0,226
J_2	0,685	0,478	0,498	0,099	0,848
J_3	0,489	0,334	0,363	0,543	0,113
J_4	0,342	0,382	0,634	0,444	0,113
J_5	0,147	0,430	0,227	0,641	0,452

Step-2.b: The weighted normalized matrix is obtained by multiplying each M_j vector with corresponding weight factor (w_j) as shown in Tab. 4. Thus, the new matrix includes the importance of each machine. In this example, the weight factors are obtained from chromosome that is generated in Step 1.

Table 4 Weighted Normalized Matrix

J/M	M_1	M_2	M_3	M_4	M_5
J_1	0,329	0,319	0,009	0,153	0,121
J_2	0,576	0,266	0,010	0,051	0,454
J_3	0,411	0,186	0,008	0,280	0,061
J_4	0,288	0,213	0,013	0,229	0,061
J_5	0,123	0,239	0,005	0,331	0,242

Step-2.c: The positive and negative ideal-solution vectors are presented in Tab. 5. For each machine, Gupta and Kumar [50] considered positive ideal solutions (PIS) as the minimum weighted normalized values and negative ideal solutions (NIS) as the maximum weighted normalized values. However, due to property of the good sequences of FSSP [51], we have divided the machines into two groups in which PIS in the first group are the minimum weighted normalized values and NIS in the second group are the minimum weighted normalized values. Notably, the machines in the first group should be visited by each job before the machines in the second group.

Table 5 Objectives and Ideal Solutions

	M_1	M_2	M_3	M_4	M_5
Objective	min	min	min	max	max
Pos.-Ideal	0,123	0,186	0,005	0,331	0,454
Neg.-Ideal	0,576	0,319	0,013	0,051	0,061

Step-2.d: This step requires the calculation of separation of alternatives (Jobs) from PIS and NIS.

Table 6 Separation of Jobs from Ideal Solutions

Job/Separation	Separation From Positive-Ideal Solution	Separation From Negative-Ideal Solution
J_1	0,45	0,27
J_2	0,54	0,40
J_3	0,49	0,31
J_4	0,44	0,35
J_5	0,22	0,57

Step-2.e: The final step of TOPSIS algorithm contains calculation of relative closeness of each job to PIS and finds the sequence of jobs. The ascending order of relative closeness values returns the sequence of processing.

Step-2.f: The result in Tab. 7 grants the sequence of processing as J_5, J_4, J_2, J_3 and J_1. The matrix in the Tab. 8 demonstrates the completion times of jobs on each machine. The makespan is calculated as 74. This final sub-step of Step-2 provides evaluation of the selected chromosome.

Table 7 Relative Closeness to Positive Ideal Solution and Sequence of Processing

	Relative Closeness to Positive-Ideal Solution	Order of Processing
J_1	0,62	5
J_2	0,58	3
J_3	0,61	4
J_4	0,55	2
J_5	0,28	1

Step-3: If the best makespan is higher than 74, it is updated as 74 and current weight factors are kept in memory as the best weight factors.

Step-4: If all chromosomes are not evaluated and termination criteria are not met, go to step 2

Table 8 Job Completion Times

Job / Machine	M_1	M_2	M_3	M_4	M_5
J_5	3	12	17	30	38
J_4	10	20	34	43	45
J_3	24	34	45	47	62
J_2	34	41	53	64	66
J_1	42	54	63	70	74

Table 9 Benchmark for Carlier and Reeves datasets

Instance	Size	NEH	Palmer	Gupta	CDS	RA	G & K	Proposed Methodology	Computation Time / s
CAR01	11x5	**7038**	7472	7348	7202	7817	7332	**7038**	0,1
CAR02	13x4	7940	7940	7534	7410	7509	8123	**7376**	0,1
CAR03	12x5	7503	7725	**7399**	**7399**	**7399**	8567	**7399**	0,2
CAR04	14x4	**8003**	8423	8423	8423	8357	9170	**8003**	1,9
CAR05	10x6	8190	8520	8773	8627	8940	8309	**7821**	0,1
CAR06	8x9	9159	9487	9441	9553	9514	9647	**8505**	0,8
CAR07	7x7	7668	7639	7639	**6819**	6923	7563	6926	0,1
CAR08	8x8	9032	9023	9224	8903	9062	9345	**8366**	0,3
REC01	20x5	1334	1391	1434	1399	1399	1595	**1317**	0,4
REC03	20x5	1136	1223	1380	1273	1159	1289	**1120**	0,4
REC05	20x5	1294	1290	1429	1338	1434	1479	**1266**	0,5
REC07	20x10	1637	1715	1678	1697	1722	1776	**1584**	1,9
REC09	20x10	1692	1915	1792	1639	1714	1805	**1604**	1,8
REC11	20x10	1635	1685	1765	1597	1636	1687	**1491**	1,8
REC13	20x15	2030	2213	2457	2107	2359	2191	**1996**	27,6
REC15	20x15	2037	2145	2199	2095	2152	2311	**1995**	27,2
REC17	20x15	2117	2216	2567	2094	2080	2312	**2027**	26,9
REC19	30x10	**2189**	2438	2529	2382	2401	2467	2223	27,5
REC21	30x10	2157	2323	2543	2358	2314	2367	**2121**	26,1
REC23	30x10	2233	2371	2426	2299	2360	2465	**2163**	25,8

*Bold solutions highlight the best solutions produced by compared algorithms.

Table 10 Benchmark Values for Taillard's Dataset

Instance	Size	NEH	MOD	CDS	Gupta	Palmer	G & K	Proposed Methodology	Computation Time / s
TAI001	20x5	1299	1322	1436	1400	1384	1377	**1297**	0,2
TAI002	20x5	**1365**	1433	1424	1380	1439	1479	1366	0,4
TAI003	20x5	1132	1136	1255	1247	1162	1342	**1114**	0,6
TAI004	20x5	1329	1475	1485	1554	1420	1444	**1324**	1
TAI005	20x5	1305	1355	1367	1370	1360	1387	**1250**	1,9
TAI006	20x5	1251	1299	1387	1333	1344	1344	**1224**	0,6
TAI007	20x5	**1251**	1366	1403	1390	1400	1481	1291	0,6
TAI008	20x5	**1215**	1312	1395	1410	1290	1390	1228	2
TAI009	20x5	1284	1371	1360	1444	1426	1350	**1271**	1,6
TAI010	20x5	**1126**	1235	1196	1194	1229	1330	1144	1,2
TAI011	20x10	1681	1789	1833	2027	1790	1898	**1667**	1,8
TAI012	20x10	1766	1802	2021	1960	1948	2051	**1735**	14,8
TAI013	20x10	**1562**	1621	1819	1780	1729	1752	1565	14,4
TAI014	20x10	**1416**	1575	1700	1730	1585	1537	1445	22,5
TAI015	20x10	1502	1714	1781	1878	1648	1735	**1501**	22,2
TAI016	20x10	1456	1607	1875	1650	1527	1667	**1426**	22,1
TAI017	20x10	1531	1650	1826	1761	1735	1663	**1528**	22,3
TAI018	20x10	1626	1799	2056	2084	1763	1849	**1619**	22,7
TAI019	20x10	**1639**	1731	1831	1837	1836	1922	1650	96,7
TAI020	20x10	**1656**	1917	2010	2137	1898	1861	1661	21,6
TAI021	20x20	2443	2787	2808	2821	2818	2700	**2417**	14,7
TAI022	20x20	**2134**	2331	2564	2586	2331	2571	2165	62,3
TAI023	20x20	2414	2598	2977	2900	2678	2742	**2400**	62,4
TAI024	20x20	**2257**	2541	2603	2670	2629	2611	2292	41,8
TAI025	20x20	2370	2615	2733	2868	2704	2775	**2352**	41,9
TAI026	20x20	2349	2439	2707	2722	2572	2575	**2306**	41,8
TAI027	20x20	2383	2465	2684	2796	2456	2628	**2361**	62,5
TAI028	20x20	**2249**	2467	2523	2612	2435	2466	2274	7,7
TAI029	20x20	**2306**	2550	2617	2701	2754	2642	2378	26,3
TAI030	20x20	**2257**	2557	2649	2690	2633	2594	2297	26,9

*Bold solutions highlight the best solutions produced by compared algorithms.

Step-5: If all chromosomes are evaluated and termination criterion is not met, then the new population is generated.

Fig. 2 shows crossover operation in RKGA described as in Bean [31].

Rand. Number	0,654	0,812	0,452	0,119	0,913
Chromosome 1	**0.431**	0.653	**0.287**	**0.732**	0.876
Chromosome 2	0.214	**0.346**	0.117	0.912	**0.945**
Offspring	0.431	0.346	0.287	0.732	0.945

Figure 2 Sample Crossover Operation

Two random numbers are generated to determine the parent chromosomes, from which the genes are inherited to offspring. As demonstrated in Tab. 1, Gene Inheritance Probability from a better chromosome used in this study is 70%. In Fig. 2, Chromosome 1 represents a chromosome with better fitness value. A random number is generated for each gene. If the random number is smaller than 0.7, the corresponding gene is inherited from Chromosome 1. Otherwise, Chromosome 2 is used for inheritance.

Step-6: If termination criterion is met, then STOP.

With the best weights provided by RKGA, the proposed approach returns a makespan of 74 for the sample problem. Gupta and Kumar [50] applied TOPSIS with 5 different weighting schemes to obtain weights. The best makespan was 80, which was provided by the scheme of "decreasing order". Decreasing order imposed weights as (5, 4, 3, 2, 1) for machines. However, the weight set (0.841, 0.556, 0.021, 0.516, 0.536) provided the makespan as 74, as shown in the illustrative example. The results revealed the importance of weight assignment for a MCDM problem.

In the literature, several construction heuristics were presented and used extensively. The solutions of the proposed methodology are compared to these well-known heuristics. Initially, the algorithms provided by Palmer, Gupta, CDS, RA and NEH found the makespan as **85, 82, 82, 79** and **79**, respectively for the considered problem.

Benchmarks problems created by Carlier (CARXX), Reeves (RECXX), and Taillard (TAIXXX) are used to compare our proposed method with the above-mentioned construction heuristics.

The solutions for the benchmark datasets of Reeves and Carlier are shown in Tab. 9. and Tab. 10 contains benchmark values for Taillard dataset. Our proposed methodology provides the best solutions for almost every problem. The results support the robustness of MCDM on FSSP and the heuristic approach that stipulates early processing of jobs with shorter process times on earlier machines.

5 CONCLUSION

The study is based on the study of Gupta and Kumar [50] in which they propose simple weighting schemes whereas we try to find out optimized weights using RKGA. In this study, an integrated methodology is proposed combining RKGA and TOPSIS. Our proposed methodology provides better solutions than the compared construction algorithms for most of the benchmark problems. Since the proposed methodology is a MCDM process, the solution is achieved in reasonable times. Hence, the proposed methodology provides the best or reasonable solutions in acceptable computational times.

As a result, the proposed method shows that the impact of MCDM techniques can be improved by determining the parameters (weights) using heuristics rather than assigning subjectively or arbitrarily.

Considering the problem with other objective functions associated with the scheduling problems and devising other solution methods by means of other MCDM approaches is recommended for future extensions of this research.

Acknowledgment

Authors of the research are members of the Marmara University Industrial Engineering Research Group. The studyis supported by AN-EL A.S.

6 REFERENCES

[1] Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling: from theory to applications*. Springer Science & Business Media.

[2] Malega, P., Rudy, V., Kanasz, R., & Gazda, V. (2020). Decentralized optimization of the flexible production lines. *Advances in Production Engineering & Management, 15*(3), 267-276. https://doi.org/10.14743/apem2020.3.364

[3] Dolgui, A., Ivanov, D., Sethi, S. P., & Sokolov, B. (2019). Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International Journal of Production Research, 57*(2), 411-432. https://doi.org/10.1080/00207543.2018.1442948

[4] Istokovic, D., Perinic, M., Dobovicek, S., & Bazina, T. (2019). Simulation framework for determining the order and size of the product batches in the flow shop: A case study. *Advances in Production Engineering & Management, 14*(2), 166-176. https://doi.org/10.14743/apem2019.2.319

[5] Fernandez-Viagas, V., Costa, A., & Framinan, J. M. (2020). Hybrid flow shop with multiple servers: a computational evaluation and efficient divide-and-conquer heuristics. *Expert Systems with Applications*, 113462. https://doi.org/10.1016/j.eswa.2020.113462

[6] Khurshid, B., Maqsood, S., Omair, M., Nawaz, R., & Akhtar, R. (2020). Hybrid evolution strategy approach for robust permutation flowshop scheduling. *Advances in Production Engineering & Management, 15*(2), 204-216. https://doi.org/10.14743/apem2020.2.359

[7] Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly, 1*(1), 61-68. https://doi.org/10.1002/nav.3800010110

[8] Emmons, H. & Vairaktarakis, G. (2012). *Flow shop scheduling: Theoretical Results, Algorithms, and Applications*. Springer Science & Business Media, 182.

[9] Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research, 1*(2), 117-129. https://doi.org/10.1287/moor.1.2.117

[10] Zhang, X. & van de Velde, S. (2012). Approximation algorithms for the parallel flow shop problem. *European Journal of Operational Research, 216*(3), 544-552. https://doi.org/10.1016/j.ejor.2011.08.007

[11] Framinan, J. M., Gupta, J. N., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop

scheduling with makespan objective. *Journal of the Operational Research Society, 55*(12), 1243-1255. https://doi.org/10.1057/palgrave.jors.2601784

[12] Ruiz, R. & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research, 177*(3), 2033-2049. https://doi.org/10.1016/j.ejor.2005.12.009

[13] Nawaz, M., Enscore Jr, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, *n*-job flow-shop sequencing problem. *Omega, 11*(1), 91-95. https://doi.org/10.1016/0305-0483(83)90088-9

[14] Zheng, D. Z. & Wang, L. (2003). An effective hybrid heuristic for flow shop scheduling. *The International Journal of Advanced Manufacturing Technology, 21*(1), 38-44. https://doi.org/10.1007/s001700300005

[15] Palmer, D. S. (1965). Sequencing jobs through a multi-stage process in the minimum total time a quick method of obtaining a near optimum. *Journal of the Operational Research Society, 16*(1), 101-107. https://doi.org/10.1057/jors.1965.8

[16] Gupta, J. N. (1971). A functional heuristic algorithm for the flowshop scheduling problem. *Journal of the Operational Research Society, 22*(1), 39-47. https://doi.org/10.1057/jors.1971.18

[17] Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management Science, 16*(10), B630-B367. https://doi.org/10.1287/mnsc.16.10.B630

[18] Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science, 23*(11), 1174-1182. https://doi.org/10.1287/mnsc.23.11.1174

[19] Triantaphyllou, E. (2000). Multi-criteria decision making methods. In Multi-criteria decision making methods: A comparative study. *Applied Optimization, 44*, 5-21. https://doi.org/10.1007/978-1-4757-3157-6_2

[20] Saaty, T. L. (1980). *The Analytic Hierarchy Process*. McGraw-Hill. New York, 324. https://doi.org/10.21236/ADA214804

[21] Saaty, T. L. (1989). Group decision making and the AHP. *The Analytic Hierarchy Process*, 59-67. https://doi.org/10.1007/978-3-642-50244-6_4

[22] Hwang, C. L. & Yoon, K. (1981). Methods for multiple attribute decision making. *Multiple Attribute Decision Making*, 58-191. https://doi.org/10.1007/978-3-642-48318-9_3

[23] Benayoun, R., Roy, B., & Sussman, B. (1966). Electre: Une méthode pour guider le choixenprésence de points de vue multiples. *Note de travail, 49*.

[24] Brans, J. P. (1982). *L'ingénierie de la décision: l'élaboration d'instruments d'aide a la décision*. Université Laval, Faculté des sciences de l'administration.

[25] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European *Journal of Operational Research, 2*(6), 429-444. https://doi.org/10.1016/0377-2217(78)90138-8

[26] Charnes, A., Cooper, W. W., & Ferguson, R. O. (1955). Optimal estimation of executive compensation by linear programming. *Management Science, 1*(2), 138-151. https://doi.org/10.1287/mnsc.1.2.138

[27] Chamodrakas, I., Alexopoulou, N., & Martakos, D. (2009). Customer evaluation for order acceptance using a novel class of fuzzy methods based on TOPSIS. *Expert Systems with Applications, 36*(4), 7409-7415. https://doi.org/10.1016/j.eswa.2008.09.050

[28] Goyal, K. K., Jain, P. K., & Jain, M. (2012). Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *International Journal of Production Research, 50*(15), 4175-4191. https://doi.org/10.1080/00207543.2011.599345

[29] Karatas, M., Sulukan, E., & Karacan, I. (2018). Assessment of Turkey's energy management performance via a hybrid multi-criteria decision-making methodology. *Energy, 153*, 890-912. https://doi.org/10.1016/j.energy.2018.04.051

[30] Güngör, A. & Uzun Kocamış, T. (2018). Analysis of financial performance in public soccer clubs by topsis method: UK application. *Hitit University Journal of Social Sciences Institute, 11*(3), 1846-1859.

[31] Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing, 6*(2), 154-160. https://doi.org/10.1287/ijoc.6.2.154

[32] Leonhart, P. F., Spieler, E., Ligabue-Braun, R., & Dorn, M. (2019). A biased random key genetic algorithm for the protein-ligand docking problem. *Soft Computing, 23*(12), 4155-4176. https://doi.org/10.1007/s00500-018-3065-5

[33] Ruiz, E., Soto-Mendoza, V., Barbosa, A. E. R., & Reyes, R. (2019). Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. *Computers & Industrial Engineering, 133*, 207-219. https://doi.org/10.1016/j.cie.2019.05.002

[34] Faria, H., Resende, M. G., & Ernst, D. (2017). A biased random key genetic algorithm applied to the electric distribution network reconfiguration problem. *Journal of Heuristics, 23*(6), 533-550. https://doi.org/10.1007/s10732-017-9355-8

[35] Samanlioglu, F., Ferrell Jr, W. G., & Kurz, M. E. (2008). A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem. *Computers & Industrial Engineering, 55*(2), 439-449. https://doi.org/10.1016/j.cie.2008.01.005

[36] Andrade, C. E., Miyazawa, F. K., & Resende, M. G. (2013). Evolutionary algorithm for the k-interconnected multi-depot multi-traveling salesmen problem. *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 463-470. https://doi.org/10.1145/2463372.2463434

[37] Gonçalves, J. F. & Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics, 17*(5), 487-525. https://doi.org/10.1007/s10732-010-9143-1

[38] Gonçalves, J. F., de Magalhães Mendes, J. J., & Resende, M. G. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research, 167*(1), 77-95. https://doi.org/10.1016/j.ejor.2004.03.012

[39] Lei, D. (2010). Solving fuzzy job shop scheduling problems using random key genetic algorithm. *The International Journal of Advanced Manufacturing Technology, 49*(1-4), 253-262. https://doi.org/10.1007/s00170-009-2379-y

[40] Dauzère-Pérès, S. & Mönch, L. (2013). Scheduling jobs on a single batch processing machine with incompatible job families and weighted number of tardy jobs objective. *Computers & Operations Research, 40*(5). https://doi.org/10.1016/j.cor.2012.12.012

[41] Subramaniam, V., Lee, G. K., Hong, G. S., Wong, Y. S., & Ramesh, T. (2000). Dynamic selection of dispatching rules for job shop scheduling. *Production Planning & Control, 11*(1), 73-81. https://doi.org/10.1080/095372800232504

[42] Singh, A. (2015). Multicriteria Dynamic Scheduling by TOPSIS and Goal Programming. *Annals of DAAAM & Proceedings, 26*(1).

[43] Leu, S. S. & Yang, C. H. (1999). GA-based multicriteria optimal model for construction scheduling. *Journal of Construction Engineering and Management, 125*(6), 420-427. https://doi.org/10.1061/(ASCE)0733-9364(1999)125:6(420)

[44] Kiran Kumar, K., Nagaraju, D., Gayathri, S., & Narayanan, S. (2017). Evaluation and Selection of Best Priority Sequencing Rule in Job Shop Scheduling using Hybrid MCDM Technique. *Materials Science and Engineering Conference Series, 197*(1), 012059. https://doi.org/10.1088/1757-899X/197/1/012059

[45] Jadhav, V. S. & Bajaj, V. H. (2011). Application of Fuzzy Topsis Method for Solving Job-Shop Scheduling Problem. *International Journal of Operational Research & Optimization, 2*(2), 333.
[46] 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. https://doi.org/10.1007/s00170-012-4388-5
[47] Lin, D., Lee, C. K. M., & Wu, Z. (2011). Integrated GA and AHP for re-entrant flow shop scheduling problem. *Quality and Reliability (ICQR), 2011 IEEE International Conference, IEEE*, 496-500. https://doi.org/10.1109/ICQR.2011.6031769
[48] Rohaninejad, M., Kheirkhah, A., Fattahi, P., & Vahedi-Nouri, B. (2015). A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 77*(1-4), 51-66. https://doi.org/10.1007/s00170-014-6415-1
[49] Braglia, M. & Grassi, A. (2009). A new heuristic for the flowshop scheduling problem to minimize makespan and maximum tardiness. *International Journal of Production Research, 47*(1), 273-288. https://doi.org/10.1080/00207540701500486
[50] Gupta, A. & Kumar, S. (2016). Flow shop scheduling decisions through Techniques for Order Preference by Similarity to an Ideal Solution (TOPSIS). *International Journal of Production Management and Engineering, 4*(2), 43-52. https://doi.org/10.4995/ijpme.2016.4102
[51] Pinedo, M. (2012). *Scheduling*. New York: Springer. https://doi.org/10.1007/978-1-4614-2361-4
[52] Snyder, L. V. & Daskin, M. S. (2006). A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research, 174*(1), 38-53. https://doi.org/10.1016/j.ejor.2004.09.057

Contact Information:

Ilknur KARACAN, PhD Candidate
(Corresponding author)
1) AN-EL Anahtar ve Elektrikli Ev Aletleri San. A.S., R&D Center
Velibaba Mah. Ankara Cad. No: 188, 34896 Pendik/İstanbul
2) Marmara University, Institute of Pure and Applied Sciences
Fahrettin Kerim Gökay Cad., 34722 Kadıköy/İstanbul
E-mail: ilknurkaracan@marun.edu.tr

Ismet KARACAN, PhD Candidate
1) AN-EL Anahtar ve Elektrikli Ev Aletleri San. A.S., R&D Center,
Velibaba Mah. Ankara Cad. No: 188, 34896 Pendik/İstanbul
2) Marmara University, Institute of Pure and Applied Sciences,
Fahrettin Kerim Gökay Cad., 34722 Kadıköy/İstanbul
E-mail: ismetkaracan@marun.edu.tr

Ozlem SENVAR, Assoc. Prof. Dr.
Marmara University, Industrial Engineering,
Fahrettin Kerim Gökay Cad., 34722 Kadıköy/İstanbul
E-mail: ozlemsenvar@gmail.com

Serol BULKAN, Prof. Dr.
Marmara University, Industrial Engineering
Fahrettin Kerim Gökay Cd., 34722 Kadıköy/İstanbul
E-mail: sbulkan@marmara.edu.tr