A Promethee-based Approach to Multi-Criteria Flexible Job Shop Scheduling Problem

Vassil Guliashki [1] and Leoneed Kirilov[2]

Abstract **– In this paper an approach combining a genetic evolutionary algorithm with Promethee-based evaluation of generated solutions is introduced. Pair-wise comparisons of criteria values are used for calculation of fitness measure for all solutions in the current population. The proposed new Promethee-based approach gives a possibility for better quality evaluation of generated solutions and achieving better parent-selection mechanism in the genetic algorithm. It also reduces very much the efforts of Decision Maker (DM) in the choice of final compromise solution.**

Keywords – **Multi-Criteria optimization, Job Shop Scheduling, Multi-Criteria Decision Analysis, Promethee method.**

I. INTRODUCTION

The job shop scheduling problem is well-known from operations research and computer science and is of high practical value with applications in many real-life situations [2, 13]. While first approaches in this area consider optimality of schedules for a single objective function, multi-objective formulations of the problem have become gradually of increasing importance [5]. A theory of multi-criteria scheduling is presented in [14]. A survey of methods for job shop scheduling using multi-criteria decision making is presented in [12]. In this paper the multi-criteria flexible job shop scheduling problem as an extension of the popular multi-criteria job shop scheduling problem is considered. During the last decades many researchers have devoted considerable efforts to developing evolutionary multi-criteria algorithms.

The problem of scheduling arises when planning and controlling the decision-making process of manufacturing and service industries. It can be schematized as follows: There is a number of N jobs to be executed. Each job consists of a given sequence of operations which needs to be performed using a number of M machines. All operations for each job must be performed in the order given by the sequence. Each operation demands the use of a particular machine for a given time. Each machine can process only one operation at a time. The goal is to find a schedule optimizing the above problem according to the given objective function (cost function, make-span, tardiness, maximal workload etc.). Scheduling consists of assigning each operation of each job a start time and a completion time on a time scale of the machine with the preference relations.

The most used in practice is the job shop scheduling problem. It is a difficult computational problem. Optimal solutions for job shop scheduling can be found in polynomial time if the number of jobs is 2, or if the number of machines is 2 and all jobs have 1 or 2 operations, or if the number of machines is 2 and all operations have duration 1. In all cases the problem obtained by incrementing the number of machines, jobs, operations or durations by 1, is NP-hard [6, 9]. Below are presented the basic formulations of the classical job shop problem (JSP) and of the flexible job shop problem (FJSP):

Job shop problem (JSP)

The JSP is formulated as follows: There is given a set of n jobs: $J_1, \ldots, J_n$, which have to be performed on m machines $M_1, \ldots, M_m$.

For each job there is given the operative consequence of the jobs composing this job. Namely:

$J_i = (O_{i,1}, \ldots, O_{i,j(i)})$, $j(i)$ is the number of operations for the corresponding job, $i=1,\ldots,n$.

It is well-known which operation on which machine should be executed. Therefore another formulation of this model is:

$$J_i = (M_{i,1}, \ldots, M_{i,j(i)}),$$

The processing times for each possible operation on each machine are known: $p_{i,k}$, $i=1,\ldots,n$; $k = 1, \ldots, j(i)$.

The optimal schedule according preliminary given criterion (criteria) has to be found. For example one criterion could be the minimization of make-span (time window) – C_{max}.

This is the most often used and chronologically the earliest developed model – see for example [6, 8].

Flexible job shop problem (FJSP)

This model represents an extension of the above job shop problem. Here each operation can be executed not only on one machine, but on a given subset of machines. This subset is naturally different for each operation. In other words, it is not a priori known which operation on which machine should be performed.

This model is closer to real life production situations and could be applied, when some or all machines are multi-functional (multitasking) – i.e. they could perform more than one operation (not at the same time) with corresponding different processing times. Among the first researchers suggesting this model are Bruker and Schlie – [3].

As noted in [4] the FJSP is a problem of high complexity and practical value, and it has been widely investigated for the last two decades. Researches on its multiobjective version started about ten years ago, but most studies focused on searching for the single optimal solution with respect to a certain aggregated objective. Research works aiming at obtaining the set of Pareto optimal solutions appeared during the recent three years.

There are two variants of FJSP – [11]:

[1]Vassil Guliashki is with the Institute of Information and Communication Technologies – BAS, "Acad. G. Bonchev" Str. Bl. 2, 1113 Sofia, Bulgaria, E-mail: vggul@yahoo.com

[2]Leoneed Kirilov is with the Institute of Information and Communication Technologies – BAS, "Acad. G. Bonchev" Str. Bl. 2, 1113 Sofia, Bulgaria, E-mail: l_kirilov_8@abv.bg

First, when each operation of each job can be executed on any, no matter which, machine. This case is relevant to the total / global flexibility (total flexible job shop problem – T-FJSP).

Second, even not each (but at least one) operation can be performed on any machine. This case refers to the partial flexibility (partial flexible job shop problem – P-FJSP).

Below is considered the multi-criteria FJSSP job problem, where n jobs J (J_i, $i \in \{1,2,\ldots,n\}$) should be processed on m existing machines M(M_k, $k \in \{1,2,\ldots,m\}$). The job J_i consists of n_i operations. For each of these operations (O_{ij}) a predetermined set of capable machines is considered (M_{ij}). One of the capable machines should be selected to perform the operation. The processing time and the start time of operation j (O_{ij}) of job J_i on machine k are denoted by p_{ijk} and t_{ijk} respectively. The assignment decision variables are denoted by X_{ijk}, and the completion time on machine k is denoted by C_k. The problem includes three criteria, which have to be minimized: the makespan (C_{max}), the critical machine workload (CWL) and the total work load of machines (TWL).

$$C_{max} = \max \{C_K \mid k = 1,\ldots,n\} \quad (1)$$

$$CWL = \max\{\sum_{i=1}^{n}\sum_{j=1}^{n_i} P_{ijk} X_{ijk}\} \text{ k: } 1,2,3,\ldots,m \quad (2)$$

$$TWL = \sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{m} P_{ijk} X_{ijk} \quad (3)$$

For this multi-criteria FJSSP we make the following assumptions:

- There is a predetermined and fixed order for the operations of each job.
- There isn't assumed priority restriction among the operations of different jobs, as well as among the jobs.
- At the beginning (at time 0), jobs are released and machines are available.
- Move times between operations and setup times of machines are ignorable.
- Only one job can be processed on each machine at each specific moment and during the process, operations can't be broken off.

The paper is organized further as follows: In section 2 we give some information about the Promethee I method, which is used to estimate and reorder different alternatives. In section 3 we present the Promethee-based approach for multi-criteria FJSP optimization. An illustrative example is presented in section 4. In section 5 we consider some challenges connected with the use of this approach and draw some conclusions.

II. The Promethee-based estimation of alternatives

In this paper the PROMETHEE I method [10, 15] is used to estimate and reorder the different alternatives. It consists in pairwise comparisons. Let $\boldsymbol{T}$ be a finite set of possible alternatives. There are three possible relations between the alternatives **a** and **b**, where $a \in T$ and $b \in T$. These relations are: preference, indifference and incomparability. They will be denoted by *Pr*, *Ind* and *Inc* respectively. Let be considered the particular criterion f(.), which has to be maximized. The effective choice between the alternatives is made interactively by the DM or by an analyst, according to their feeling of the intensities of preference between alternatives. In this connection the following parameters have to be fixed:

q - a threshold defining an indifference area,
v - a threshold, defining a strict preference area,
s - a parameter, which value lies between q and v.

Let d = f(a) – f(b). The preference function Pref(a,b) will be considered. It gives the intensity of preference of a over b in function of the deviation d.

A generalized criterion H(d) is associated to each criterion. Six different types of H(d) are defined in the PROMETHEE I method. In this paper the generalized criterion H(d) of type 5 is used, known as V-shape with indifference area. It is formulated as follows:

$$H(d) = \begin{cases} 0 & |d| \le q \\ (|d|-q)/(v-q) & q < |d| \le v \\ 1 & |d| > v. \end{cases} \quad (4)$$

This criterion has been often used. For this type of H(d) the intensity of preference increases linearly between q and v.

Let the preference index π(a,b) of **a** over **b** over all the criteria be defined in the form:

$$\pi(a,b) = \sum_{l=1}^{k} w_l H_l(a,b), \quad (5)$$

where w_l, l=1,...,k; are weights associated to each criterion and $\sum_{l=1}^{k} w_l = 1$.

Here it is assumed that the weights w_l, l=1,...,k; are equal. In this case π(a,b) is simply the arithmetic mean of all the intensities of preference $Pref_j$(a,b), j = 1,…,k;. For each pair (a,b) the values π(a,b) are calculated. Then for each alternative $a \in T$ the positive outranking flow $\boldsymbol{\Phi}^+(a)$ is calculated as follows:

$$\boldsymbol{\Phi}^+(a) = \sum_{x \in T} \pi(a,x). \quad (6)$$

In our problem the preference index $\pi(b,a) = -\pi(a,b)$. The corresponding negative outranking flow in the problem (1)-(2) is symmetric to $\boldsymbol{\Phi}^+(a)$ and has the opposite direction:

$$\boldsymbol{\Phi}^-(a) = -\boldsymbol{\Phi}^+(a). \quad (7)$$

Hence the positive outranking flow is enough to expresses how each alternative is outranking all the others. The higher $\boldsymbol{\Phi}^+(a)$, the better is the alternative. $\boldsymbol{\Phi}^+(a)$ represents the power of a, it gives its outranking character.

To arrange all explored alternatives in an order according their preference over all the criteria (all the objectives) in the problem (1)-(2) the positive outranking flow $\boldsymbol{\Phi}^+(a)$ is calculated for each explored alternative $a \in T$. Then the alternatives are reordered in a non-increasing order of their $\boldsymbol{\Phi}^+$ values.

III. The Promethee-based approach to multi-criteria FJSSP

For problems with complex, non-smooth and multimodal objective function, where the information of objective

function derivatives is not accessible, have been developed global heuristic algorithms, as well as evolutionary (population based) algorithms, such like Genetic algorithms (see [7]), Scatter search, Ant systems and Particle swarm optimization algorithms. Their main common features are as follows:

- They memorize solutions (or characteristics of solutions) in a population of individuals. Each individual is associated with a feasible solution of the problem at hand.
- They include a generating solutions search procedure, which uses the information (implicitly) stored in the population.

Here is proposed an approach combining a genetic-type evolutionary algorithm with PROMETHEE I – estimation and ranking the generated solutions in the population (considering them as different alternatives), according the calculated Φ^+–values. The corresponding algorithm is called PBGALG.

To start the calculations the shifting bottleneck procedure (see [1]) is used to minimize the makespan C_{max} and the obtained best solution is used as initial solution in PBGALG.

To generate new feasible solutions (individuals) for the current population we use a *selection* operator and a *modified crossover* operator. For *selection* there aren't used two parent solutions, but only the best current solution is taken to be used for offspring generation. The best solution is chosen according the calculated Φ^+–values by means of Promethee - estimation.

The *modified crossover* is based on the representing the corresponding schedule (feasible solution) as a string of consecutive operations (O-string), executed on the given machines. In case more than one operation have one and the same start time t_{ijk} then the operation with lower job index and lower operation index precedes the operations with greater indices in the operation string. Then a number of l operations in this string are chosen randomly to generate l new feasible solutions. Here l depends on the population length. A check is performed if the corresponding operation can be executed on another machine, different from that one used in the current schedule. If there are other possible machines this operation is fixed to be performed on the machine, requiring the minimal processing time for the given operation. After that the operations after this operation in the operation string are reordered by means of the following heuristic to generate a new feasible schedule-solution.

Heuristic procedure:

Let we denote one operation by O_{ikj}, where first index shows to which job corresponds the operation, the second index gives the place of the operation in the operative consequence for the corresponding job and third index denotes the O-string position. Let v be the O-string position of operation, which should be fixed currently, and let s_{total} is the total number of all operations, which should be executed.

Basic cycle

For $j = v+1, s_{total}$

Among the operations still unassigned to machines find the minimal index: $k = \min\{k_1, \ldots, k_i\}$

Machine-Selection:

Let the possible machines for operation $O_{i,k,j}$ are m(1, i(k)), ..., m(r(i(k)), i(k)) with the corresponding summary processing times up to the moment P(1, i(k)), ..., P(r(i(k)), i(k)).

Select the machine m(u, i(k)) such that

$P(u, i(k))+p_{i,k,j} = \min\{ P(1, i(k))+p_{i,k,j}, \ldots, P(r(i(k)), i(k))+p_{i,k,j} \}$.

Select the machine for execution, which has the minimal processing time from the set of possible machines. Assign (fix) the operation $O_{i,k,j}$ to be executed on this machine.

end Machine-Selection

Actualization: $P(u, i(k)) = P(u, i(k)) + p_{i,k,j}$.

end *j*

In this way a new feasible solution (new schedule) is generated, which strictly differs from the parent solution.

The pseudo-code of PBGALG can be written in the following form:

Begin
Initialize the population P
Evaluate the individuals in P (by means of Promethee I – procedure)
Sort P according to the fitness value.
While *no stopping criterion is met, do*
Repeat: Perform parent selection.
Apply the modified crossover operator to generate offspring individuals.
Select individual s to survive.
If *s is better than one individual in P*
Replace the old individual in P by x_j.
The check is performed starting by the worst individual.
EndIf
EndWhile
End

IV. AN ILLUSTRATIVE EXAMPLE

We consider the following test example.

3 machines - M_1, M_2, M_3

5 jobs - $J_1(O_1, O_2, O_3)$, $J_2(O_2, O_3, O_4)$, $J_3(O_1, O_2, O_3, O_4)$, $J_4(O_2, O_4)$, $J_5(O_1, O_2, O_4)$,

Note that some operations are identical.

The processing times are given in the (O, M) matrix:

	M_1	M_2	M_3
O_1	10	12	7
O_2	8	6	X
O_3	X	5	11
O_4	3	4	X

The symbol "X" denotes that the operation cannot be processed at the corresponding machine and vice versa.

According to the proposed algorithm we start the solution process generating an initial solution by the shifting bottleneck heuristic [1]. The obtained schedule is the following:

M1 (O31), (O32), (O12), (O24), (O34), (O54)
time: 10 18 26 29 32 35
M2 (O22), (O42), (O44), (O51), (O52),
time: 6 12 16 28 34

M3 (O11), (O23), (O33), (O13)
time: 7 18 29 40

The number of operations is s_{total} = 15. The obtained makespan-value is C_{max} = 40.

The corresponding operation string is:
O22,O11,O31,O42,O23,O32,O44,O51,O12,O33,O24,O52,O13,O34,O54;

To create an initial population with l = 4, there are chosen randomly 3 operations: O23, O44 and O24 (correspondingly the 5-th, 7-th and 11-th in the string).

For each of them, there is available one possible other machine for execution (M2, M1 and M2).

Applying the above heuristic procedure we obtain the following new solutions:

1) M1 (O31), (O32), (O52), (O24), (O44), (O54)
time: 10 18 26 29 32 35
M2 (O22), (O42), (O12), (O23), (O33), (O34)
time: 6 12 18 23 28 32
M3 (O11), (O51), (O13)
time: 7 14 18-29

2) M1 (O31), (O32), (O44), (O24), (O34), (O54)
time: 10 18 21 24 28-31 34-37
M2 (O22), (O42), (O12), (O13), (O33), (O52)
time: 6 12 18 23 28 34
M3 (O11), (O23), (O51)
time: 7 18 25

3) M1 (O31), (O32), (O12), (O52), (O54)
time: 10 18 26 28-36 39
M2 (O22), (O42), (O44), (O51), (O24) (O13), (O34)
time: 6 12 16 28 32 37 41
M3 (O11), (O23), (O33)
time: 7 18 29

To evaluate the obtained solutions by the Promethee I - procedure, we calculate the following table of alternatives:

Alternative	***TWL***	***CWL***	C_{max}
1 (shift. bottl.)	109	40	40
2	92	35	35
3	92	34	37
4	107	41	41

This table is used by PBGALG to calculate the corresponding Φ^+–values through the Promethee – estimation. The following result is obtained:

$\Phi^+(1)$= 0,078; $\Phi^+(2)$=1,833; $\Phi^+(3)$=1,74; $\Phi^+(4)$=0,04

Hence the obtained solutions are ranked in the following order: Alternative 2, alternative 3, alternative 1 and alternative4. This result is presented to Decision Maker, who decides if the search process should be terminated or should continue. In case the Decision Maker is satisfied with at least one of the obtained solutions he/she can stop the calculations.

V. Conclusion

The presented Promethee-based approach is useful for solving multi-criteria FJSSP. It can facilitate very much the Decision Maker in the choice of best final solution.

One direction for further research is the possibility for determined choice of operations in the *modified crossover*. One such rule, for example, could be: among the operations having minimal processing time on a machine different from the currently used that one with maximal difference between the current and the minimal processing time to be chosen first.

Acknowledgement

This study is partially supported by the project № BG161PO003-1.1.06-0083 to the EU operative program „Development of Bulgarian economy competitiveness" entitled: "Scientific research for the purposes of development of software tool for generating efficient schedules by an innovative method for multiple objective optimization in discrete manufacturing within the scope of small and medium enterprises".

References

[1] Adams J., E. Balas and D. Zawack, (1988), "The Shifting Bottleneck Procedure for Job Shop Scheduling", *Management Science*, Vol. 34, No. 3, pp. 391-401.

[2] Blazewicz J., K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling Computer and Manufacturing Processes.* Springer Verlag, Berlin, Heidelberg, New York, 2. edition, 2001.

[3] Bruker, P., R. Schlie (1990) Job shop with multi-purpose machine, Computing, vol. 45, 1990, pp. 369-375.

[4] Chiang, Tsung-Che, Lin, Hsiao-Jou (2013) A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling, *Int. J. Production Economics*, 141, 87-98.

[5] Daniels, R., Incorporating preference information into multi-objective scheduling. *European Journal of Operational Research*, 77:272-286, 1994.

[6] Garey, M., D. Johnson, and R. Sethi. (1976) The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, 1(2):117–129, 1976.

[7] Goldberg D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, Mass, 1989.

[8] Lawler, E., J. Lenstra, A. Rinnooy Kan, and D. Shmoys. (1993) Sequencing and scheduling: algorithms and complexity. In A.H.G. Rinnooy Kan S.C Graves and P.H. Zipkin, editors, *Logistics of Production and Inventory*, volume 4, Chapter 9, pages 445–522. Elsevier, 1993.

[9] Lenstra, J. K., A. R. Kan, P. Brucker, (1977), "Complexity of Machine Scheduling Problems", *Annals of Discrete Mathematics*, vol. 1, 1977, pp. 343-362.

[10] Moca M. and G. Fedak "Using Promethee Methods for Multi-Criteria Pull-based Scheduling on DCIs", In *Proc. of the 8th IEEE Int. Conf. on eScience*, Chicago, USA, Oct. 2012, pp. 1-8.

[11] Motaghedi-Iarijani, A., Sabri-Iaghaie K., Heydari, M. (2010) Solving job shop scheduling with multi objective approach, *Int. J. of Industrial Engineering and Production Research*, vol. 21, No 4, pp. 197-209.

[12] Parveen S. and H. Ullah, (2010), "Reviev on Job-Shop and Flow-Shop Scheduling Using Multi Criteria Decision Making", *Journal of Mechanical Engineering*, Vol. ME 41, No. 2, December 2010, Transaction of the Mech. Eng. Div., The Institution of Engineers, Bangladesh, pp. 130-146.

[13] Pinedo, M. *Planning and Scheduling in Manufacturing and Services.* Springer Verlag, Berlin, Heidelberg, New York, 2005.

[14] T'kindt, V., J.-C. Billaut. *Multicriteria Scheduling: Theory, Models and Algorithms.* Springer Verlag, Berlin, Heidelberg, New York, 2002, 2006.

[15] Vincke, P., *Multiple Criteria Decision Aid*, John Wiley& Sons, New York (1992)