An Interactive Method for Solving Multiple Objectives Flexible Job Shop Problems (MOFJSP)

Leoneed Kirilov[1], Vassil Guliashki[1], Krasimira Genova[1], Pasko Paskov[2]

[1]Inst. of Information and Communication Technologies - BAS, Sofia, Bulgaria.
lkirilov@iinf.bas.bg; vggul@yahoo.com; kgenova@yahoo.com
[2]Step-Soft Alliance Ltd, Sofia, Bulgaria
pasko@stepsoft.bg

Abstract: An interactive method for solving multiple objective flexible job shop is proposed in this paper. It allows the DM to search for a best compromise schedule in the set of Pareto optimal schedules. The method works with an arbitrary number of objective functions defined in advance. The dialog is in terms of objectives' weights or aspiration levels in the objective space. A genetic algorithm is used as a tool for generating the Pareto optimal schedules at each iteration.

Keywords: flexible job shop, schedules, genetic algorithms, multiple objectives, interactive methods

1 Introduction

The job shop scheduling problem (JSP) is well-known from operations research and computer science and is of high practical value with applications in many real-life situations [1, 17]. While first approaches in this area consider optimality of schedules for a single objective function, multiple objectives formulations of the problem have become gradually of increasing importance [4]. A theory of multiple objectives scheduling is presented in [20].

The most used in practice is the JSP. 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, 12, 13].

In this paper an interactive method for MOFJSPs with arbitrary number of objectives is presented.

2 Available approaches

FJSP is an extension of the JSP. 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. Among the first researchers suggesting this model are Bruker and Schlie – [2].

As noted in [3] 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.

Two variants of FJSP are known: 1) when each operation of each job can be executed on any, no matter which, machine (total FJSP – T-FJSP), 2) even not each (but at least one) operation can be performed on any machine (partial FJSP – P-FJSP). Recently, an extensions of flexible job shop model are considered in [9, 10].

There are a large amount of methods for solving FJSP, including exact search methods, dispatching rules, improving heuristics and metaheuristics – see for example [5,15]. One survey of multiple-objective job-shop and flow-shop scheduling techniques is presented in [16]. A survey of GA for JSP is presented in [11]. Important details in this respect are given in [19]. Most of the publications consider the case with two or three clearly defined objectives. But there does not exist a method considering the general case – MOFJSP or MOJSP with an arbitrary number of objectives with concrete definition of objectives depending by the concrete problem to be solved and/or the preferences of the Decision Maker (DM).

3 The proposed approach

Our goal is to construct a method for solving FJSP or EFJSP with several objective functions. As it is known, there is no a unique solution in the objective space but a set of equally good solutions, namely Pareto optimal solutions for multiple objective problems – [14]. They form the so called Pareto or Efficient frontier.

One approach is to formulate an exact model (for example linear integer or mixed integer) taking into account the objectives. And this model could be solved by a suitable known possibly modified method. Unfortunately the objectives are not known a priori. They may depend and they depend on the preferences of the DM and/or by the specifics of the task to be solved. Also, the model depends very strong from the included objective(s). Therefore another very promising approach is to solve the problem by using heuristic method(s).

The proposed method belongs to the class of interactive methods – [14]. It means that the solution process is controlled by Decision Maker. In the phase of dialog the DM gives his/her preferences in terms of weights for each objective function or in terms of aspiration levels in the objective space. The DM also evaluates the generated solution(s) in this phase. If he/she is satisfied by one of the solutions then the process stops. Otherwise a number of solutions are generated in the phase of computations and the process continues with the phase of dialog and so on.

As we said the method works with an arbitrary number of objectives. We give the DM the possibility to choose how many and which one objectives to optimize on the base of preliminary given set of objectives. Of course he/she can use another objective functions. We propose a preliminary set of 24 objective functions such as make-span, total machine workload, tardiness etc. The DM may choose a subset of the objectives according to the optimization goals.

We choose genetic algorithm for generating a set of schedules in the phase of computations. Genetic Algorithms (GA) are inspired by Darwin's theory of natural selection and survival of the fittest. They were invented by Holland [8] and first proposed for scheduling problems by Goldberg [7]. The main advantage of GA is that this is a global search technique, performing robust on a large class of problems. They use also population of individuals (solutions), which is very useful for generating representative set of solutions on the Pareto frontier in the multiple objective optimization.

A specific schedule is defined when for each operation from each job it is known the machine to be performed on and the starting time (or the ending time). One way of schedule representation is the Gant' diagram. But for our purposes we need more effective way of schedule representation in order to perform different operations with it, namely the coding. This way of schedule representation is very effective when genetic algorithms are used. We will use the two-vector coding - see [21]. A schedule is presented in the forms of vector A and vector B, see fig. 1 and fig. 2.The two strings have equal length and it is just the sum of all operations for all jobs. For example if we have a number of four jobs J1(O1, O2, O3), J2(O1, O2, O3), J3(O1, O2, O3, O4), J4(O1, O2) to be performed on a set of 5 machines - M1, M2, M3, M4, M5; then in this case the strings' length is 12. The number in each box of string A denotes on which machine the corresponding operation to be performed - see fig. 1.

The operations and precedence relations between job's operations are given in string B - see fig. 2. If we take for example job J1(O1, O2, O3) we see three boxes with value "1". The first one found from left to right means O1,1, the second one found from left to right means O1,2 and etc.

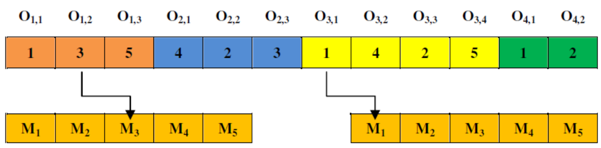

Fig.1 Representation of A-string

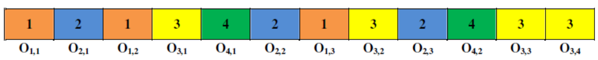

Fig.2 Representation of B-string

Fig.3 Representation of C-string

Sometimes we will use in the description of a schedule the starting times of the operations explicitly. Therefore we introduce array C with the same length $J = J_1 + ... + J_n$ and the same construction as an array A - see fig. 3. But each cell of an array C contains the starting time for the corresponding machine from the same cell in the array A.

As a fitness function we choose the so called augmented Tchebycheff Problem (TchP) - [18]:

$$\min_{\Omega} s(z, z^{I}, \Omega)$$

where

$$s(z, z^{I}, \Omega) = \max_{k \in K} (\omega_k (z_k^{I} - f_k(x) / (z_k^{U} - f_k(x)) +$$
$$\rho \sum_{k \in K} \omega_k (z_k^{I} - f_k(x)) / (z_k^{U} - f_k(x)),$$

K - number of objectives;

$z = f(x) = (f_1(x), \dots, f_K(x));$

z^{I} – pseudo-ideal vector;

z^{U} – utopia vector ($z^{U} >> z^{I}$ – sufficiently large number for each objective);

ω_k – weighting coeffcients;

Ω – the feasible set of schedules;

ρ – real positive number in the interval $[0.001, 0.1]$.

The above function guarantees finding of only Pareto optimal but not weak Pareto optimal schedules. When solve the above scalarizing problem by means of genetic algorithm the received solution is an approximate Pareto optimal solution.

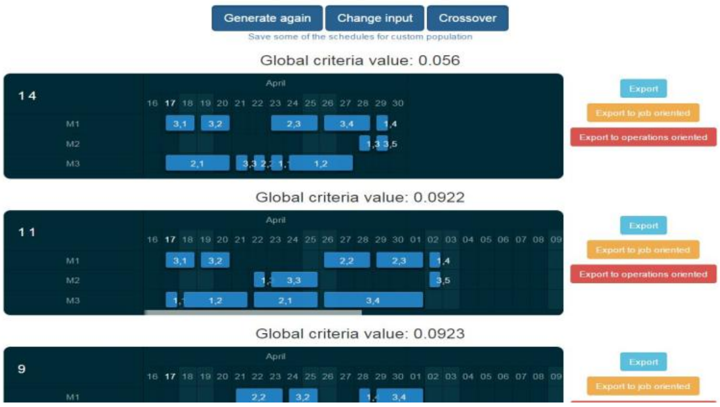

Fig. 4. A dialog screen of the Interactive method for MOFJSP.

4 An algorithmic scheme

The proposed interactive multiple objectives method can be presented in a generalized scheme as follows.

Step 1. Initialization

Input parameters of the problem. The DM selects the set of objectives to be optimized.

Step 2. Generating the initial population.

A heuristic procedure for random generating a set of schedules is proposed. The heuristic is based on a rule for uniform loading of the machines.

Multiple objective Iteration counter h = 0

Step 3. Generating the current population.

Multiple objective iteration counter h = h+1

Start of the genetic algorithm

The augmented Tchebycheff Problem (TchP) is used as a fitness function.

Step 4. Compute pseudo pay-off table and pseudo ideal vector, i.e. only for the computed population of solutions.

Step 5. Rank the solutions according to the fitness function and show them to the DM.

Step 6. Dialog: The DM evaluates the solutions according to the selected objectives. If he/she is satisfied by some solution – then stop the procedure. Otherwise he/she set the weights or aspiration levels.

Step 7. Go to *Step 3*.

The proposed algorithm is programmed as a software module to be included in an ERP (Enterprise Resource Planning) system designed for small and medium enterprises. One of the dialog screens is shown on fig.4.

5 Conclusion

The proposed method for solving FJSP has the following characteristics:
- It is an interactive and iterative method;
- The dialog is user-friendly in terms of weighting coefficients or aspiration levels in the objective space;
- It works with an arbitrary number of objective functions;
- A genetic algorithm is proposed as a sub-procedure for generating schedules;
- The generated solutions are approximated Pareto optimal solutions;
- The method can generate each Pareto optimal schedule from the Pareto frontier;
- The method could solve both FJSP and EFJSP.

Acknowledgments:
This study is partially supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions) and 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] 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.
[2] Bruker, P., R. Schlie (1990) Job shop with multi-purpose machine, Computing, vol. 45, 1990, pp. 369-375.
[3] 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.
[4] Daniels, R., Incorporating preference information into multi-objective scheduling. *European Journal of Operational Research*, 77:272-286, 1994.
[5] Kacem I, Hammadi S, Borne P (2002a) Approach by localization and multi-objective evolutionary optimization for flexible job shop scheduling problems. IEEE Transaction on Systems, Man and Cybernetics, PART C 32(l):l-l3.
[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.,1989. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley. Reading. MA.
[8] Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
[9] Birgin, E., P. Feofiloff, C. Fernandes, E. de Melo, M. Oshiro and D. Ronconi (2013) A MILP model for an extended version of the Flexible Job Shop Problem, Optimization Letters, vol. 8, No 4, pp. 1417-1431.
[10] Kirilov, L., V. Guliashki (2014) An extension of flexible job shop problem (FJSP) and method for solving, *Proceedings of CompSysTech*'14 - "15th International Conference on Computer Systems and Technologies", ACM New York, NY, USA ©2014, ACM International Conference Proceeding Series, Vol. 883 , pp. 210-217.
[11] Kumar A. and A. K. Dhingra, 2012, Optimization of Scheduling Problems: A genetic algorithm survey, *International Journal of Applied Sciences and Engineering Research*, Vol. 1, No. 2, pp. 11-25.
[12] 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.
[13] Lenstra, J. K., A. R. Kan, P. Brucker, (1977), "Complexity of Machine Scheduling Problems", *Annals of Discrete Mathematics*, vol. 1, 1977, pp. 343-362.
[14] Miettinen, K., *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999.
[15] 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.
[16] 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.
[17] Pinedo, M. *Planning and Scheduling in Manufacturing and Services.* Springer Verlag, Berlin, Heidelberg, New York, 2005.
[18] Steuer, R. E., E. U. Choo, (1983), An interactive weighted Tchebycheff procedure for multiple objective programming, *Mathematical Programming* 26(3): 326-344.
[19] Thörnblad K., 2013, *Mathematical Optimization in Flexible Job Shop Scheduling. Modelling, Analysis and Case Studies*., Ph.D. Thesis, Department of Mathematical Sciences, Division of Mathematics, Chalmers University of Technology and University of Gothenburg, Göteborg, Sweden 2013.
[20] T'kindt, V., J.-C. Billaut.*Multicriteria Scheduling: Theory, Models and Algorithms.* Springer Verlag, Berlin, Heidelberg, New York, 2002, 2006.
[21] Zhang, G., Shao, X., Li, P., Gao, L, (2008) An Effective Hybrid Particle Swarm Optimization Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problem, Computers & Industrial Engineering, Vol. 56 (4), 1309-1318.