ISBN : 978-602-97809-0-1

Job Shop Scheduling Problems with Alternative Routings using Variable Neighbourhood Descent to Minimize Makespan

Arif Imran#1, Martino Luis#2, Emsosfi Zaini#3, B. Benny#4

#Production System Laboratory, Industrial Engineering Department Insitut Teknologi Nasional Bandung
Jalan P.H.H. Mustofa No. 23 Bandung 40124, Indonesia

1imran@itenas.ac.id
2luis@itenas.ac.id
3efi@itenas.ac.id
4bny_511@yahoo.com

Abstract - **This paper deals with a generalization of the classical job shop scheduling problem known as the flexible job shop scheduling problem (FJSP). A heuristic algorithm based on variable neighbourhood descent (VND) is put forward to tackle the problem such that the maximum completion time (makespan) of the schedule is minimised. The performance of this VND based approach is assessed using well known data sets from the literature. Some preliminary results are reported and very competitive results are also obtained when compared to the best known results from the literature.**

Keywords – **Job shop, Alternative routings, Heuristics, Makespan**

I. Introduction

In the classical job shop scheduling problem (CJSSP), we are given a set of jobs J_i (i= 1, 2, .., n) that has to be processed on a set of unrelated machines M_k (k = 1, 2, ..., m). Each job J_i is composed of a set of operations O_{ij} (j= 1, 2, .., l) which are processed in a specified order. Every operation O_{ij} follows a technology constraint, called a routing, which is fixed and known in advance. The operation O_{ij} has to be processed without interruption during t_{ij} units of time and a machine is constantly available from time zero. Each machine can process only one operation at a time and operations are performed without preemption. The objective is to find a schedule that minimise makespan (C_{max}), i.e., the maximum completion times of all jobs.

The CJSSP is considered as a hard combinatorial optimisation problem (NP-hard) [1[. The problem has been widely studied and many solution methods, such as exact and heuristic techniques have been put forward. Exact methods are mainly based on the branch and bound algorithms (see [2]-[4]). These techniques guarantee to yield an optimal solution, however, the methods are found to be effective when solving small instances. Moreover, their implementation needs too much computational cost when the size of problems increases. Contrarily, heuristic approaches, which can solve larger problem instances within reasonable computing time, are good alternatives for such problems. These methods consist of priority dispatching rules [5], shifting bottleneck procedures (see [6]-[8]), local search such as tabu search [9], [10], simulated annealing (e.g. [11], [12]), variable neighbourhood search [13], and greedy randomized adaptive search procedures (GRASP) [14], and evolution schemes such as genetic algorithms (i.e. [15], [16]), and ant colony optimizations (for instance, see [17], [18]). For completeness, see [19] and [20] for comprehensive reviews on CJSSP.

This paper concerns with the flexible job shop scheduling problem (FJSP) as a generalization of the CJSSP which the model is a closer approximations to a variance of scheduling problems encounter in real manufacturing systems. The FJSP problem, also known as the job scheduling problem with alternative routings or machines, is an enhancement of the CJSSP by allowing an operation to be performed on any machines out of a set of available machines. The objective is to select for each operation, an eligible machine and a starting time which an operation must be processed so that the maximum completion times C_{max} of all jobs (makespan) is minimized. Therefore, the FJSP is more complex that the classical ones in terms of determining the operation routing sub-problem, that assigns each operation to a machine from a set of eligible machines and the scheduling sub-problem that consists of sequencing the assigned operations on all machines to yield a feasible schedule which minimize a predefined objective function [21]. Alvarez-Valdes *et al.* [22] demonstrate the complexity of the FJSP in a glass factory.

The high complexity of the FJSP makes this problem falls into NP-hard realm. Hence, many previous works in the FJSP concentrates on heuristic/ metaheuristic techniques. Nasr and Elsayed [23] propose a greedy heuristic procedure to deal with job shop scheduling problems with alternative machines. Brandimarte [24] decomposes the FJSP into the routing sub-

ISBN : 978-602-97809-0-1

problem and the scheduling sub-problem. The first sub-problem was solved by some well known dispatching rules and then tabu search algorithm was adopted to focus on the scheduling sub-problem. In the following research, tabu search are also applied by [21], [25], [26] to deal with the FJSP. Baykasoğlu [27] put forward a linguistic based meta-heuristic in solving the FJSP. Jansen *et al.* [28] design a linear time approximation scheme for the FJSP. In recent years, genetic algorithms have been successfully developed to tackle the FJSP as shown in [29]-[31]. In the subsequent research, Gao *et al.* [32] combine genetic algorithms with variable neighborhood descent (VND) to deal with the FJSP. Prassetiyo *et al.* [33] tackle the FJSP by introducing a population based heuristic such as the ant colony system. Very recently, Saleh *et al.* [34] address the FJSP by designing two phase heuristic algorithms based on GRASP heuristic.

In this study, we propose a simple but powerful two phase heuristic schemes to address the FJSP with makespan criterion. In the first phase, priority dispatching rules such as shortest processing time (SPT) is used to deal with the routing sub-problem. The second phase is to schedule assigned operations using a variable neighborhood descent algorithm in order to find the best schedule that minimize makespan.

The remainder of this paper is organized as follows: in Section 2 we present the problem description. Section 3 describes the general framework of the VND heuristic. In Section 4, the computational experiments are discussed and finally, the last section summarizes conclusions and outlines some research avenues that would be worthwhile investigating in the future.

II. Problem Description

The flexible job shop scheduling problem (FJSP) can be defined as follows. We are given a set of independent jobs ($J_i = J_1, J_2, ..., J_n$) and a set of unrelated machines ($M_k = M_1, M_2, \ldots, M_m$). A job J_i has a sequence of operation O_{ij} ($j = 1, 2, ..., l$). Each operation O_{ij} can be performed on any among available machines out of a set of flexible machines. The problem is assumed as a static problem, in a sense that all data are known in advanced and all jobs and machines are available at time zero.

An example of a flexible job shop problem is illustrated in Figure 1. This example is taken from [21], which presents two jobs and three machines in the flexible job shop scheduling problem. Each operation has alternative machines except for O_{22}. Based on Figure 1, operation O_{11} can be executed by machine m_1 for 10 minutes or by machine m_2 for 15 minutes. The problem arises in allocating each operation to an available (a flexible) machine and in sequencing the operations on the machines to minimize one or more predefined objectives.

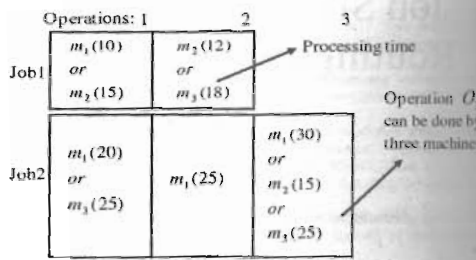

Fig. 1 Example of two jobs and three machines in the FJSP ([21])

III. Solution Framework

Our proposed algorithm is based on the work of Sevkali dan Aydin [13] which investigates the job shop scheduling with fixed routing (CJSSP) using VNS and Nasr and Elsayed [23] that the job shop scheduling problem with alternative routings/ machines using greedy algorithms.

A. A Variable Neighbourhood Descent (VND) for the FJSP

VND is a metaheuristic technique that systematically change neighborhood structures within a possibly randomized local search [35]-[37] to escape from local optima trap. The VND method is obtained if change of neighborhood structures is carried out in a deterministic way. VND has been adopted to find optimal or near optimal solutions for a number of combinatorial optimization problems. One can refer to [37] for more detail on applications of VND.

The idea behind VND is on simple principle that is a local minimum with respect to one neighborhood structure is not necessary so for another. Hence, VND is more applicable than any other metaheuristic approaches as the method does not need any parameters. In this study, we apply VND as a local search in finding the best schedule that minimize makespan.

B. The Proposed Algorithm

This section discusses an algorithm for the FJSP with makespan criterion. The algorithm consists of two phases that find the best allocation of each operation to a set of flexible machines and the best sequence of the all allocated operations that yield a schedule with the best makespan. In the first phase, the greedy algorithm of Nasr and Elsayed [23] is modified where initial solutions to choose alternative machines are generated randomly and then some priority dispatching rules are used to sequence the assigned operations. For simplicity, in this paper, we use shortest processing time (SPT) and other rules may also be applied.

The second phase is a local search that adapts the VNS method presented by Sevkali dan Aydin [13]. The basic idea behind this method is to change neighbourhood structures obtained in initial solutions by alternately applying exchange and insert processes.

For the sake of completeness, we introduce some notations that are used in the proposed algorithms, as follows:

ISBN : 978-602-97809-0-1

n = number of job
m = number of machines
i = index for *job i* (i = 1, 2, ..., n)
j = index for operation of job i
k = index for machine k (k = 1, 2, ..., m)
r_{ij} = ready time of the jth operation of job i
R_k = ready time of machine k
C^b_{ijk} = completion time of the jth operation of job i on machine k for the bth schedule (b = 1, 2, ..., *maxiter*)
t_{ijk} = processing time of the jth operation of job i on machine k
MS_b = makespan for the bth schedule
S_p = sequence of p best schedules, p = 1, 2, .., *maxiter*
L_p = the jth operation of job i in a critical path for a partial schedule S_p
S^h_p = a partial schedule after *insert/exchange* processes at the hth iteration
C^{ph}_{ijk} = completion time of the jth operation of job i on machine k for the pth schedule at the hth iteration
MS = final makespan

The step by step of the proposed algorithm for solving the FJSP to minimize makespan is as follows:

Phase 1 (the Selection of alternative machines)

Step 1

Input data routings and processing times of all jobs. Set *maxiter* = max (2, $\left\lceil \frac{n}{4} \right\rceil$), *Ready time* of the first operation of each job = 0 ([r_{i1} = 0 $\forall$ i = (1, 2,..., n)]). *Ready time* of all machines = 0 ([R_k = 0 $\forall$ k = (1, 2,..., m)]), b = 1, l = 1, and p = max [2, 10%*maksiter*]

Step 2

If $b \leq maxiter$ then choose randomly an alternative machine for every operation j and set $b = b+1$, else go to Step 3

Step 3

Schedule every operation of job i with its chosen alternative machine using shortest processing time (other priority dispatching rules may also be applied)

Step 4

Compute completion time for each job and the *makespan* from the schedules obtained at Step 3 using the following equations:

$$C^b_{ijk} = \max\ [r_{ij}\ ,\ R_k] + t_{ijk};\quad \forall i = (1, 2,...,n),\ \forall b = (1, 2,...,maxiter) \quad (1)$$

$$MS_b = \max\ [\ C^b_{ijk}\ ], \forall b = (1,2,..., maxiter) \quad (2)$$

Step 5

Based on the obtained schedules at Step 3, choose p schedules with the smallest makespan (S_p) and set $S = S_p$ (S = {S_1, ..., $S_{maxiter}$})

Phase 2 (the VND Procedure)

Step 6

For $p = l$, input S_p and L_p and set the last operations of the last job in the critical path into SO_L ($SO_L = \{O_{ijk}\}, O_{ijk} \in L_p$) and $h = 1$

Step 7 (the Insert Procedure)

Change the neighborhood structure by inserting the operations $O_{ijk} \in SO_L$ and schedule all the operations in S^h_p

Step 8

Compute completion time for each job and determine the makespan using the following equations:

$$C^{ph}_{ijk} = \max\ [r_{ij}, R_k] + t_{ijk}\ ;\ \forall\ i = (1,2,...,n) \quad (3)$$

$$MS^{ph} = \max\ [\ C^{ph}_{ijk}\ ] \quad (4)$$

Step 9

Determine L_p in S^h_p and $SO_L = \{O_{ijk}\}$, $O_{ijk} \in L_p$ and set MS_p = min [MS_p, MS^{ph}] and $h = h + 1$

Step 10

If SO_L = {$\varnothing$} then go to Step 11, else return to Step 7

Step 11 (the Exchange Procedure)

Change the neighborhood structure by swapping the operations $O_{ijk} \in SO_L$ and schedule all the operations in S^h_p

Step 12

As Step 8 and remove S_p from S

Step 13

If $MS_p \geq MS^{ph}$, set MS_p = min [MS_p, MS^{ph}] and $h = h + 1$ then go back to Step 7, else set $h = h + 1$ and return to Step 11

Step 14

If $S \neq \{\varnothing\}$, set $l = l + 1$ and return to Step 6, else continue to Step 15

Step 15

Select the schedule with the smallest makespan (MS = min (MS_1, ..., $MS_{maxiter}$))

IV. Computational Results

This section presents computational results when our proposed algorithm was tested using two well known instances from the literature. The first instance, adopted from Nasr and Elsayed [23] (NE for short), is a small problem, which consists of four jobs and six machines. The second one, taken from Brandimarte [24], is a large problem that has ten jobs and six machines. In order to evaluate our proposed algorithms, we compare our results with some published results from literature.

Table 1 shows the comparison of our results with other authors. The first column is the instance name. The second and third columns are the number of jobs and the number of machines for each data set, respectively. The fourth column is our results by the proposed algorithm. The remaining columns are the best results from [23], [33], [31], [30], [32], and [34]. The bold numbers in Table 1 refer to best makespan found for each instance and the shaded cells indicate that some authors do not conducted experiments for the instances.

Based on the results in Table 1, our proposed algorithm generally provides competitive results compared to the other publications. For example, in the case of NE data set, our algorithm yields better solution compared to [23] for one unit of time and gives similar makespan to the remaining published results.

TABLE I

DETAILS OF COMPARISON OF ALL RESULTS

Data Sets	*n*	*m*	Makespan						
			The Proposed Algorithm	Nasr and Elsayed [23]	Prassetiyo *et al.* [33]	Moon *et al.* [31]	Pezella *et al.* [30]	Gao *et al.* [32]	Saleh *et al.* [34]
NE	4	6	**17**	18	**17**	**17**			**17**
MK-01	10	6	**40**				**40**	**40**	**40**

V. CONCLUSIONS

In this paper the job shop scheduling problem with alternative routings is studied. A variable neighbourhood descent based approach is proposed and its performance is evaluated using well-known problem instances. Two data sets are used. The small one contains four independent jobs and six machines and the large one has ten jobs and six machines. The results from both instances show that our proposed methods produce favourable outcomes when compared against those in the literature.

The following research directions may be worthy of investigation in the future. The work can be extended by hybridizing VND and other metaheuristics such as GRASP or threshold accepting algorithm to enhance the local search. An overview of heuristic search in general can be found in [38]. Another direct modification of the existing problem is a case of dynamic situation where jobs can arrive at a system at any time.

REFERENCES

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, W. H. Freeman and Company, 1979.

[2] J. Carlier and E. Pinson, "An algorithm for solving the job shop problem", *Management Science*, Vol. 35, No. 29, pp. 164–176, 1989.

[3] P. Brucker, B. Jurisch, and B. Sievers, B., "A branch and bound algorithm for job-shop scheduling problem", *Discrete Applied Mathematics*, 49, pp. 105–127, 1994.

[4] I. Sabuncuoglu and M. Bayiz, "Job shop scheduling with beam search", *European Journal of Operational Research*, 118, pp. 390–412, 1999.

[5] O. Holthaus and C. Rajendrant, "New dispatching rules for scheduling in a job shop – an experimental study", *Int J Adv Manuf Tech*, 13, pp.148-153, 1997.

[6] J. Adams, E. Balas, and D. Zawack, "The shifting bottleneck procedure for job shop scheduling", *Management Science*, Vol. 34, pp. 391–401, 1988.

[7] S. Dauzer-Peres and J. B. Lasserre, "A modified shifting bottleneck procedure for job-shop scheduling", *International Journal of Production Research*, Vol. 31, No. 4, pp. 923–932, 1993.

[8] H. Wenqia and Y. Aihua, "An improved shifting bottleneck procedure for the job shop scheduling problem", *Computers & Operations Research*, 31, pp. 2093–2110, 2004.

[9] E. D. Taillard, "Parallel taboo search techniques for the job shop scheduling problem", *Journal on Computing*, Vol. 6, No.2, pp. 108-117, 1993.

[10] E. Nowicki and C. Smutnicki, "A Fast taboo search algorithm for the job shop problem", *Management Science*, Vol. 42, No. 6, pp. 797-813, 1996.

[11] P. J . M. Van Laarhoven, E. H. L. Aarts, and J . K. Lenstra, "Job shop scheduling by simulated annealing. *Operations Research*, 40, pp. 113–125. 1992.

[12] S. G. Ponnambalam, N. Jawahar, and P. Aravindan, "A simulated annealing algorithm for job shop scheduling", *Production Planning & Control*, Vol. 10, No. 8, pp. 767 –777, 1999.

[13] M. Sevkali, and M. E. Aydin, " Variable neighbourhood search for job shop scheduling problems", *Journal of Software*, Vol. 1, No. 2, pp. 34-39, 2006.

[14] S. Binato, W. Hery, D. Loewenstern, and M. G. C. Resende, "A GRASP for job shop scheduling", In C. Ribeiro dan P. Hansen (editors), *Essays and Survery on Metaheuristics*, Kluwer Academic Publishers, pp. 59-79, 2001.

[15] L. Wang and D. Z. Zheng, "A modified genetic algorithm for job shop scheduling", *Int J Adv Manuf Tech*, 20, pp. 72–76, 2002.

[16] T.K. Liu, J. T. Tsai, and J. H. Chou, "Improved genetic algorithm for the job-shop scheduling problem, *Int J Adv Manuf Tech*, 27, pp. 1021–1029, 2006.

[17] S. Van der zwaan, and C. Marques, "Ant colony optimisations for job shop scheduling", in *Proc. of the Third Workshop on Genetic Algorithm and Artificial Life (GAAL 99)*, 1999.

[18] K. C Ying and C. J. Liao, "An ant colony system approach for scheduling problems", *Production Planning & Control*, Vol. 14, No. 1, pp. 68–75, 2003.

[19] J. Blazewicz, W. Domschke, and E. Pesch, "The job shop scheduling problem: conventional and new solution techniques", *European Journal of Operation Research*, 93, pp. 1-33, 1996.

[20] A. Jain and S. Meeran, "Deterministic job shop scheduling: past, present and future", *European Journal of Operation Research*, 113, pp. 390-434, 1999.

[21] P. Fattahi, M. Saidi-Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems", *J Intell Manuf*, 18, pp. 331–342, 2007.

[22] R. Alvarez-Valdes , A. Fuertes , J. M. Tamarit , G. Giménez, and R. Ramos, "A heuristic to schedule flexible job-shop in a glass factory", *European Journal of Operational Research*, Vol. 165 , pp. 525–534, 2005.

[23] N. Nasr, and E. Elsayed, "Job shop scheduling with alternative machines", *International Journal of Production Research*, Vol. 28, pp. 1595–1609, 1990.

[24] P. Brandimarte, "Routing and scheduling in a flexible job shop by taboo search", *Annals of Operations Research*, 41, pp. 157–183, 1993.

[25] M. Saidi-Mehrabad and P. Fattahi, "Flexible job shop scheduling with tabu search algorithm", *Int J Adv Manuf Tech*, 32, pp. 563–570, 2007.

[26] M. Ennigrou and K. Ghédira, "New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach", *Auton Agent Multi-Agent Syst*, 17, pp. 270–287, 2008.

[27] A. Baykasoğlu, "Linguistic-based meta-heuristic optimization model for flexible job shop scheduling", *International Journal of Production Research*, Vol. 40, No. 17, pp. 4523-4543, 2002.

[28] K. Jansen, M. Mastrolilli, and R. Solis-Oba, "Approximation algorithms for flexible job shop problems", *International Journal of Foundations of Computer Science*, Vol. 16, No. 2, pp. 361-379, 2005.

[29] M. F. Hussain and S. B. Joshi, "A genetic algorithm for job shop scheduling problems with alternate routing", in *Proc. IEEE. International Conference* 3, pp. 2225-2230, 1998.

[30] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job shop scheduling problem", *Computers & Operations Research*, 35, pp. 3202-3212, 2008.

[31] I. Moon, S. Lee, and H. Bae, Genetic algorithms for job shop scheduling problems with alternative routings", *International Journal of Production Research*, Vol. 46, No. 10, 15, pp. 2695–2705, 2008.

[32] J. Gao, L. Sun and M. Gen, "A hybrid genetic and variable neighbourhood descent algorithm for flexible job shop scheduling problems", *Computers & Operations Research*, 35, pp. 2892-2897, 2008.

[33] H. Prasetyo, E. Zaini, M. Luis, and D. Irawan, "Model for job shop scheduling with alternative routing using ant colony system to minimize makespan", in *Proc. of the National Conference on Quality Improvements of Manufacturing and Service systems*, 2005. (in Indonesian).

[34] A. Saleh, M. Luis, and N. Susanti, "Job Scheduling Problems with alternative routings using greedy randomized adaptive search to minimize makespan", in *Proc. Ritekra National Conference*, pp. 71-78, 2010. (in Indonesian).

[35] N. Mladenovic and P. Hansen, "Variable neighborhood search", *Computers & Operations Research*, 24 (11), pp. 1097–100, 1997.

[36] P. Hansen and N. Mladenovic, "Variable neighborhood search: principles and applications", *European Journal of Operational Research*, 130, pp. 449–67, 2001.

[37] N. Mladenovic and P. Hansen, "Variable Neighbourhood Search", in F. Glover and G. Kochenberger (editors): *Handbook of Metaheuristic*, Kluwer, Dordrecht, pp. 145-184, 2003,

[38] S. Salhi, "Heuristic search: the science of tomorrow. *Paper presented at OR 48 Conference*, University of Bath, Bath, 2006.