Southeast-Asian J. of Sciences Vol. 2, No. 2(2013) pp. 165-170

MIXED HEURISTIC METHODS FOR PERMUTATION FLOWSHOP SCHEDULING

Titikan Moonsan[*] and **Tawun Remsungnen**[†]

[*] *Department of Mathematics, Faculty of Science*
Khon Kaen University, Khon Kaen, Thailand
e-mail: mtitikan@udru.ac.th

[†] *Faculty of Applied Science and Engineering*
Nong Khai Campus, Khon Kaen Univ., Nong Khai, Thailand
e-mail: rtawun@kku.ac.th

Abstract

The decision criteria and the reverse NEH have been proposed for improvement of construction heuristic NEH methods. Then mixed construction and meta-heuristic methods NEH-DE is proposed to obtain the optimal solutions of permutation flowshop scheduling problem. The standard test instances of Taillard are evaluated to test the qualities of methods. The new optimal solution is found for one problem.

1 Introduction

Because of many economic and industrial applications,the flowshop scheduling problem (FSP) has been intensively studied with diverse of assumptions, objective functions and implementing various optimization techniques. The regular flowshop problem consists of two main elements: (1) a set of m machines and (2) a set of n jobs to be processed on the machines. All jobs have the same ordering of machines for their processes sequence. Each job can be processed on one and only one machine at a time and only once on each machine. Each machine can process only one job at a time. Operations are not preemptable and set-up times of operations are independent of the sequences and therefore can be included in the processing time. The FSP is that the processing order is the same on each machine and the objective function is optimal. Among

Key words: construction NEH, meta heuristic, flowshop, inverse NEH, mixed NEH-DE, Taillard instances.

desired objectives, makespan or completion time (C_{max}) minimization has attracted a lot of attention. Though the problem with minimize C_{max} on two machines can be solved in time by using the famous Johnson's rule, while the general problem has been proved to be strongly NP-complete [1].

Many approximate algorithms have been developed to find best solutions in a short time. These algorithms can be classified into two categories, construction hueristic method and improvement metahueristic method.The construction hueristic method is an algorithm that determines one or some solutions in a finite length time. The solution finds it fast and is good but not necessarily optimal. As for several construction heuristics methods for FSP had been developed in relatively early decades, e.g., heuristics by Palmer [2], Campbell et al. [3], and Nawaz et al. (denoted by NEH) [4] and its improvements. The metaheuristics method is formally defined as an iterative generation process which guides a subordinate heuristic by combining smart concepts and strategies for searching in the solutions space in order to find efficiently optimal solutions [Osman and Laporte 1996]. There are several metaheuristics, such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS) algorithm and differential evolution (DE) algorithm [5, 6]. The DE algorlithm is a new evolutionary computational method which has become a new research focus because of its outstanding performance in the first contest of IEEE evolutionary computational.

In this study, some new strategies that combining between construction heuristics method and metaheuristics methods have been proposed for the FSP with the objective of makespan.

2 Models and Calculations

2.1 Problem Formulation

Let $\pi = \pi(1)\pi(2)...\pi(n)$ be member of possible solutions space. The processing time of job j on machine i is denoted by $p(i, j)$. The completion time of job j at position $\pi(j)$ on machine i denoted by $C_{i,\pi(j)}$ is calculated as follows:

$$C_{i,\pi(j)} = max(C_{i-1,\pi(j)}, C_{i,\pi(j-1)}) + p(i, j), \tag{1}$$

where $i = 1, ..., m$, $j = 1, ..., n$, $C_{0,\pi(j)} = 0$ and $C_{i,\pi(0)} = 0$. Then the C_{max} is the completion time of the last job on the last machine, $C_{m,\pi(n)}$. The total jobs flowtime and total machines operating time are denoted by C_{tf} and C_{ot} and are obtained as $\sum_{j=1,..,n}(C_{m,j})$ and $\sum_{i=1,..,m}(C_{n,i})$, respectively.

Let Π denote the set of all such permutations. The FSP then can be formulated such that:

$$C_{max}(\pi^*) = min_{\pi\in\Pi} C_{max}(\pi). \tag{2}$$

2.2 Algorithms

Algorithm 1 NEH: algorithm is as follow steps:
(1) Arrange the n jobs in descending order of the sum of processing times ($T_j = \sum_{i=1,...,m} p(i,j)$).
(2) Take the first two jobs and schedule them in order to minimize the partial makespan as if there were only these two jobs. Choose the best one to be a partial best sequence, π^2_{best}.
For $k = 3$ to n do:
(3) Insert the k-th job into π^{k-1}_{best} at the place, among the k possible ones, which minimizes the partial makespan. Choose the best one to be a partial best sequence, π^k_{best}.
End

Consider step (2) and step (4) of algorithm 1, which sometime there exist more than one best sequence with respect to C_{max}. The algorithm just select the first or last or random one which can lead to the different final solutions. In this study the using of C_{tf} and C_{ot} as second and third criteria is proposed and denoted as NEH3C. Since the NEH algorithm do develop the solution from partial sequences, the order of jobs in k-1 previous partial sequence is fixed and the solution should be improved if the interchage of this order is allowed. In this study, the reverse NEH with full sequance (rfNEH3C) algorithm is proposed as follows:

Algorithm 2 rfNEH3C:
(1) Give the initial sequence and set it be π^n_{best}. Obtain the C_{max} of π^n_{best} and set it be C^{best}_{max}. Set $k = n - 1$.
While $k > 1$ do:
(2) Take out the k-th job and insert it into k-1 positions at the place among previous k-1 jobs, which minimizes the makespan. Choose the best one to be a full best sequence π^k_{best} and obtain it best C^k_{max}.
(3) If $C^k_{max} < C^{best}_{max}$ then set $\pi^n_{best} = \pi^k_{max}$, $C^{best}_{max} = C^k_{max}$ and $k = n - 1$ else $k = k - 1$.
End.

The construction heuristic NEH3C and rfNEH3C with using of three criteria, C_{max}, C_{tf} and C_{ot} algorithms are used in the combination with metaheuristic DE with hill climbing algorithm [7, 8]. This construction-meta-heuristic combined algoritm is denoted by NEHxHDE and is outlined as follows:

Algorithm 3 NEH3CxHDE:
(1) Set the number of population (NP), number of generation (NG), interation number of hill climbing (NH).
(2) **RAND:** Generate an inition soluion, $X^0 = (\pi^0_1, \pi^0_2, ..., \pi^0_{NP})$, by uniform

random from permutation space of {1,2,...,n}.
(3) **NEH3C:** Perform NEH3C algorithm without step (1), in order to obtain the initial $F(X^0) = (C_{max}\pi_1^0, C_{max}\pi_2^0, ..., C_{max}\pi_{NP}^0)$.
(4) **DE+HILL:** Perform HDE alorithm with NP, NG, and NH parameters. Set X^0 and $F(X^0)$ as an initial propulation and its fitness value in order to obtain X^{NG} and $F(X^{NG})$.
(5) **rfNEH3C:** Perform rfNEH3C algorithm using X^{NG} and $F(X^{NG})$ in order to obtain X^f and $F(X^f)$. The solution is $\pi^* = min_{i=1,NP}C_{max}(\pi_i^f)$.

2.3 Numerical Experiment

The Taillard's benchmark sets of 5 machines with 20 jobs (I5x20) and 10 machines with 20 jobs (I10x20) in total of 20 instances are used for testing the algorithm. The NP, NG and NH parameters are set to 40 ($2 \times n$), 100 ($5 \times n$) and 20 (n), respectively.

3 Results and Discussions

Results reported in this paper will focus on preliminary analysis of efficiency of the NEH3CxHDE algorithm in order to use the obtained information for improving. The table of obtained C_{max} for two sets of Taillard's bencmarks is shown as Figure 1. In I5x20 set,two new upper bounds C_{max} are obtained for instances 5 after and 7, while all other instances the equal number with Taillard upper bounds are obtained. For instance 5, the solution is obtained after applying of HILL step while intance 7, the solution is obtained after applying of DE step. In I10x20 only half number of instances are obtained the upper bounds equal to Tailllard' numbers. Only instance 2 of I10x20 is found the improving of solution of HILL step by rfNEH3C step. This result may ead to the unnecessary use of rfNEH3C step or it should be rearrange to another position. Figure 2, shows two examples of the evolution of solutions after each steps of NEH3CxHDE algorithm. It is for sure that a big jump of solultions form RAND to NEH3C. Then the solutions are significantly improved by DE step. If the solution is not yet found, the solutions are agian improved by HILL step. However, the rfNEH3C step is not requred, base on these results. Note that, getting two soulutions that give new upper bounds of Taillad's instances is notable success in this work.However there are still much more theoretical and experimental works required for modifying and improving of the NEH3CxHDE algorithm.

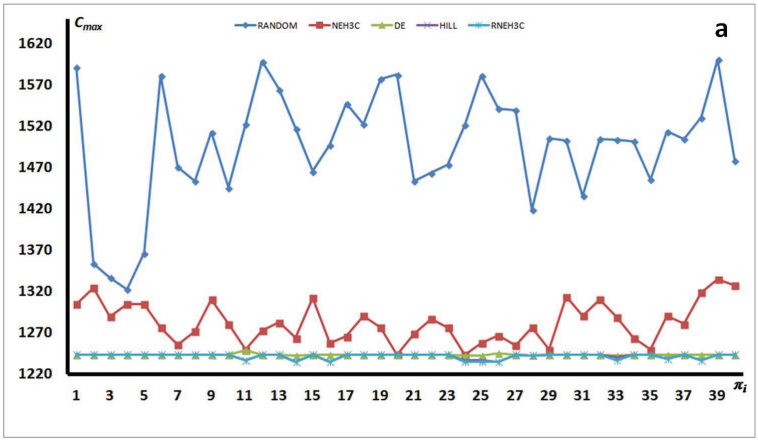
References

[1] M. R. Garey, D. D. Jonhson, R. Sethi, *The complexity of flowshop and jobshop scheduling.*, Math. Oper. Res., 1 (1976), 117-129.

20 Jobs x 5 Machines							
i	UB	LB	RAND	NEH3C	DE	HILL	rfNEH3C
1	1278	1232	1390	1286	1278	1278	1278
2	1359	1290	1412	1365	1359	1359	1359
3	1081	1073	1249	1089	1081	1081	1081
4	1293	1268	1418	1314	1298	1293	1293
5	1236	1198	1323	1244	1243	1235	1235
6	1195	1180	1312	1195	1195	1195	1195
7	1239	1226	1374	1251	1234	1234	1234
8	1206	1170	1341	1224	1206	1206	1206
9	1230	1206	1360	1255	1230	1230	1230
10	1108	1082	1164	1113	1108	1108	1108

20 Jobs x 10 Machines							
i	UB	LB	RAND	NEH3C	DE	HILL	rfNEH3C
1	1582	1448	1757(1)	1640(1)	1586	1582	1582
2	1659	1479	1854(1)	1697(2)	1676	1660	1659
3	1496	1407	1645(1)	1529(1)	1505	1500	1500
4	1378	1308	1547(1)	1412(1)	1385	1379	1379
5	1419	1325	1558(1)	1459(1)	1422	1419	1419
6	1397	1290	1591(1)	1425(1)	1424	1405	1405
7	1484	1388	1630(1)	1526(2)	1484	1484	1484
8	1538	1363	1788(1)	1577(1)	1552	1543	1543
9	1397	1290	1591(1)	1441(1)	1412	1400	1400
10	1593	1472	1720(1)	1635(1)	1600	1593	1593

Figure 1: The obtained C_{max} after each steps of NEH3CxHDE for sets of 20 jobs with 5 machines and 20 jobs with 10 machines. The numbers that are lower than (red) and equal to (blue) Taillard's upper bounds are obtained, respectively.



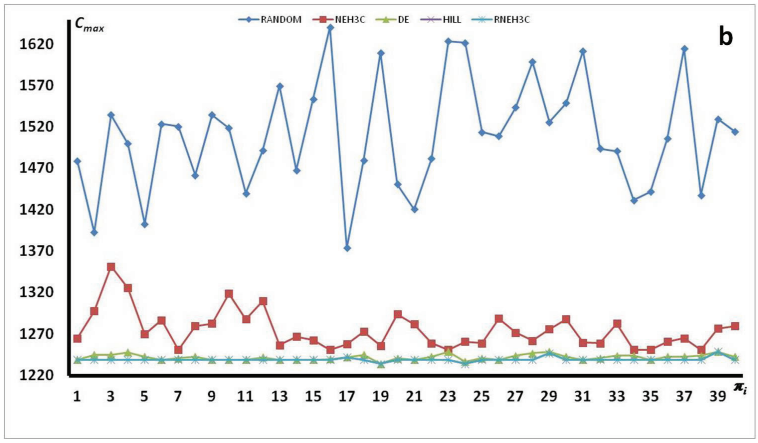


Figure 2: The improvement of C_{max} after applied each steps of NEH3CxHDE for instances 5(a) and 7(b) of Taillard's 5 machines and 20 jobs benchmark.

[2] D.S. Palmer,*Sequencing jobs through a multistage process in the minimum total time: a quick method of obtaining a near optimum.* Oper. Res. Quart., 16 (1965), 101-107.

[3] H. G. Campbell, R. A. Dudek, M.L Smith, *A heuristic algorithm of the n-job, m-machine sequencing problem.* Manag. Sci., 16 (1970), B630-B63.

[4] M. Nawaz, E. Enscore, I. Ham, *A heuristic algorithm for the m-machine, n-job flowshop sequencing problem.* OMEGA, Inter. J. of Manage. Sci., 11 (1983), 91-95.

[5] R. Storn, K. Price, *Differential evolution Ca simple and efficient heuristic for global optimization over continuous spaces.*, J. Glob. Opt., 11 (1997), 341-359.

[6] A. Josiah, O. Fred, *Differential evolution algorithm for solving multi-objective crop planning model.*, Agri. Water Manage., 97 (2010), 848-856.

[7] W. Xiang, X. Guoyi, *Hybrid Differential Evolution Algorithm for TravelingSalesman Problem.*, Proc. Eng., 15 (2011), 2716-2720.

[8] Y. C. He, et al., *Differential evolution algorithm with position-order encoding for solving traveling salesman problem.*, J. Comp. App., 3 (2007).