

Scheduling Hybrid Flow Shops by an Improved Memetic Algorithm

Wang Binggang

Research Institute of Business Administration
Henan University of Urban Construction
Pingdingshan, China, 467044

Abstract—This study is concerned about how to optimize the production sequences to minimize the makespan in hybrid flow shops with limited intermediate buffers. An improved memetic algorithm (IMA) is proposed for solving the NP-hard problem. In this algorithm, the method of generating the initial population, the local search procedures, the selection, crossover and mutation operators are newly designed. Optimization results are compared between the IMA and other three scheduling algorithms proposed in previous literature. Comparison results show that the IMA proposed in this paper is feasible and efficient for scheduling hybrid flow shops with limited intermediate buffers.

Keywords--Scheduling, Hybrid flow shop, Buffer, Memetic algorithm, Makespan

I. INTRODUCTION

The scheduling problems considered in this paper can be described as follows: The hybrid flow line has S stages. There are m_s identical parallel machines in each stage, and there is at least one stage, $m_s > 1$. Between every two successive stages, there exists a buffer with limited size, and jobs obey the FIFO (first in first out) rule in each buffer. There have n jobs are to be sequentially processed on through the first stage to the last one in the flow line in the same order. If there is no buffer space available for a completed job on one machine in one stage, then this job must remain on that machine until either a buffer space or one machine in the next stage becomes available. At any time, no job can be processed on more than one machine, while no machine can process more than one job simultaneously. The task is to determine a whole schedule to minimize the makespan.

In order to maximize the utilization of hybrid flow lines, some researchers have investigated the scheduling problem and, for different optimization goals, proposed some methods, mainly including: construction method [1], heuristic algorithm [2], mixed integer programming method [3], tabu search method [4][5], Lagrangian relaxation method [6], genetic algorithm based method [7], et. al. Gupta has proved that the scheduling problem in hybrid flow shop is an NP-hard problem even if the flow line consists of only two stages and there has more than one machine at either stage [8].

Though some research work has done on scheduling problem in hybrid flow shops with limited intermediate buffers, to our best knowledge, memetic algorithm, an efficient method for solving optimization problem in different fields [9-11], has not been adopted to solve this

problem. So we propose a newly-designed memetic algorithm to solve the NP-hard problem in this paper.

II. PROCEDURES OF IMPROVED MEMETIC ALGORITHM (IMA)

The procedures of the proposed algorithm are described as follows.

- Step 1: Generate an initial population, $P(0)$, with size L , and let generation number $g=0$.
- Step 2: Calculate each individual's fitness value in $P(g)$.
- Step 3: Selection operations.
- Step 4: Crossover operations.
- Step 5: Local search.
- Step 6: Mutation operations.
- Step 7: Local search.
- Step 8: Elitist strategy.
- Step 9: $g=g+1$, if the maximum generation number, G , is reached, then output the best solution, otherwise, go to Step 2.

III. IMPLEMENTATION OF IMA

A. Encoding method

The letter encoding method is employed to represent the feasible production sequence for the first stage in the hybrid flow shop, i.e., using one unique letter to represent one type of parts.

B. Initial population

The following steps are taken to guarantee the solutions' quality in the initial population.

Step 1: Generate the first feasible production sequence for the first stage in the flow line by the NEH method, which is one of the most effective heuristics for scheduling flow shop scheduling problems.

Step 2: Use the SPT rule, which is another effective method for scheduling flow shop scheduling problems, to generate the second feasible production sequence.

Step 3: The rest ($L - 2$) individuals are generated randomly.

C. Makespan calculation

The event-driven method described in Ref. [7], one of our previous study, is employed to determine the makespan for one production sequence of the first stage in hybrid flow shops.

D. Fitness value

The makespan value is used directly as the fitness value of a feasible production sequence for the first stage in the flow line.

E. Selection

The following procedures are adopted to ensure that the individuals with higher quality have more chance to enter into next generation and, at the same time, to maintain the diversity of solutions in a population.

Step 1: Rank all the individuals in descending order according to each individual's fitness value.

Step 2: Select a determined number of solutions, from the best to the worst, to enter into the next generation directly. The quantity selected is controlled by a preset proportion value, P_s .

Step 3: For each of the remaining $L \times (1 - P_s)$ individuals, the local search procedures described in section 3.8 are performed.

F. Crossover

The partial mapping crossover operator (PMX) is employed for crossover operations. Two offspring individuals will be generated by each crossover operation. After that, among the two parent and the two offspring individuals, select the best two according to their fitness values to replace the original two parent individuals. The two parent individuals selected for performing crossover operation are controlled by the adaptive crossover probability, P_{c_j} . The procedures for adjusting crossover probability are described as follows.

Step 1: If $j=1$, then, P_{c1} will be determined randomly from the given interval for crossover probability, I_c , otherwise, go to step 2.

Step 2: Calculate $y = v_j / v_{j-1} - 1$, if $y \geq 1$, P_{c_j} will be selected randomly from I_c , else, $P_{c_j} = U_c + 0.0025g$.

Where, v_j , v_{j-1} is the objective function value of individual j and $j-1$, respectively, U_c is the upper bound of I_c .

G. Mutation

The swap mutation operator (SWAP) is adopted for performing mutation operation on each selected individual, controlled by the adaptive mutation probability, P_{m_j} . An offspring individual will be generated by each mutation operation on the parent individual. Then, compare the parent and offspring individual according to their fitness values, if the offspring individual is better than its parent individual, then replace the parent individual with the offspring individual, otherwise, keep the parent individual unchanged. The procedures for adjusting mutation probability are described as follows.

Step 1: If $j=1$, then, P_{m1} will be determined randomly from the given interval for mutation probability, I_m , otherwise, go to step 2.

Step 2: Calculate $y = v_j / v_{j-1} - 1$, if $y \geq 1$, P_{m_j} will be selected randomly from I_m , else, $P_{m_j} = U_m + 0.0035g$.

Where, U_m is the upper bound of I_m .

H. Local search

Local search is a search method that iteratively examines the set of points in a neighborhood of the current solution and replaces the current solution with a better neighbor if any exists. Local search procedures are as follows.

Step 1: Select a production sequence from the population randomly.

Step 2: Generate its neighborhood with 12 solutions by inverse mutation operator (INV).

Step 3: Calculate the fitness values of the original individual and all the generated individuals and compare with each other. If the best individual in the neighborhood is better than the original individual, then replace the original individual with the best one in the neighborhood, otherwise, keep the original individual unchanged.

I. Elitist strategy

To keep the elitist individual, the best-so-far solution obtained will be used to replace the worst individual in the temporary population, $P_t(g)$, after Step 7 of IMA in every generation.

IV. CASE STUDY AND DISCUSSIONS

In order to test the performance of the algorithm proposed in this paper, for the same optimization problems, we compare the results obtained by the IMA and other three algorithms presented in previously published papers [1] [4] [12]. In our computational experiments, the hybrid flow line consists of three stages. There are 2, 3 and 3 machines in the three stages respectively. The buffer size between stage 1 and stage 2 is 3 and it is also 3 between stage 2 and stage 3. Thirteen kinds of parts (A~M) can be processed in this production line. Production plans and process time are shown in Table I. The IMA proposed in this paper is coded in C++ and implemented on a PC (Intel (R) Celeron (R) M, CPU 1.70 GHz, 512MB). Numerous computational effort was devoted to test the sensitivity of algorithm parameters. The selected algorithm parameters are as follows: $L=50$, $I_c=[0.55-0.70]$, $I_m=[0.10-0.20]$, $P_s=0.5$, $G = 50$. Computational results are listed in Table II. t_{CPU} is the computational time and C_{max} is the value of makespan.

From computational results listed in Table II, it is clear that all the results obtained by the IMA are better than those achieved by WLA and RITM. By comparing IMA with TS-H1, we can find that, for production 2 and 4, the same results are obtained by both the two methods, and for other four production plans, all of the IMA's results are better than those obtained by TS-H1. Moreover, the computational time of IMA and TS-H1 is almost the same. So, the IMA proposed in this paper is feasible and efficient for solving the scheduling problem in hybrid flow shops with limited intermediate buffers.

V. SUMMARIES

With the aim of minimizing the makespan, an improved memetic algorithm is proposed in this study to solve the scheduling problem in hybrid flow shops with limited

intermediate buffers. In this algorithm, the method of generating the initial population and the local search procedures are newly designed, the selection scheme, the adaptive crossover and mutation operators are presented. Over the same computation data, computational experiments are conducted to compare the IMA and other three algorithms proposed in previous literature. The feasibility and efficiency are shown by the comparison results.

REFERENCES

[1] WITTROCK R J. An Adaptable Scheduling Algorithm for Flexible Flow Lines. *Operational Research* [J], 1988, 33(4), PP: 445-453.

[2] SAWIK T J. Scheduling Flexible Flow Lines with No In-Process Buffers. *International Journal of Production Research* [J], 1995, 33(5), PP: 1357-1367.

[3] SAWIK T. Mixed Integer Programming for Scheduling Flexible Flow Lines with Limited Intermediate Buffers. *Mathematical and Computer Modelling* [J], 2000, 31, PP: 39-52.

[4] WARDONO B, FATHI Y. A Tabu Search Algorithm for the Multi-stage Parallel Machine Problem with Limited Buffer Capacities. *European Journal of Operational Research* [J], 2004, 155, PP: 380-401.

[5] WANG Xian-peng, TANG Li-xin. A Tabu Search Heuristic for the Hybrid Flow Shop Scheduling with Finite Intermediate Buffers. *Computers & Operations Research* [J], 2009, 36, PP: 907-918.

[6] TANG Li-xin, XUAN Hua. Lagrangian Relaxation Algorithms for Real-Time Hybrid Flow Shop Scheduling with Finite Intermediate Buffers. *Journal of the Operational Research Society* [J], 2006, 57, PP: 316-324.

[7] WANG Bing-gang, Rao Yun-qing, Shao Xin-yu, XU Chi. Solving Multi-stage Parallel Machine Problem with Limited Intermediate Buffers (in Chinese). *Journal of Huazhong University of Science and Technology* [J], 2009, 37(5), PP: 86-89.

[8] GUPTA J N D. Two-Stage Hybrid Flow Shop Scheduling Problem. *Operations Research Society* [J], 1988, 39, PP: 359-364.

[9] GAO Liang, QIAN Wei-rong, LI Xin-yu, WANG Jun-feng. Application of Memetic Algorithm in Assembly Sequence Planning. *International Journal of Advanced Manufacturing Technology* [J], 2010, 49, PP: 1175-1184.

[10] FRUTOS M, OLIVERA A C, TOHME F. A Memetic Algorithm Based on a NSGAI Scheme for the Flexible Job-Shop Scheduling. *Annals of Operations Research* [J], 2010, 181, PP: 745-765.

[11] BANSAL R, SRIVASTAVA K. A Memetic Algorithm for the Cyclic Antibandwidth Maximization Problem. *Soft Computing* [J], 2011, 15, PP: 397-412.

[12] SAWIK T J. A Scheduling Algorithm for Flexible Flow Lines with Limited Intermediate Buffers. *Applied Stochastic Models and Data Analysis* [J], 1993, 9, PP: 127-138.

TABLE I. PRODUCTION PLAN AND PROCESS TIME

Part Type	Process Time (s)			Production Plan					
	s=1	s=2	s=3	1	2	3	4	5	6
A	39	12	15	12	0	0	0	0	0
B	13	29	55	1	0	0	0	0	0
C	22	57	61	26	0	0	14	23	20
D	234	40	1	0	0	0	2	0	0
E	39	26	81	0	6	7	4	0	1
F	13	71	55	0	14	20	16	0	0
G	143	67	1	1	4	0	0	3	1
H	0	29	15	7	0	0	0	0	0
I	26	40	75	0	6	0	0	0	5
J	18	60	35	0	4	0	0	0	0
K	22	71	41	4	0	0	0	0	0
L	13	71	55	0	4	0	0	0	0
M	61	47	35	0	0	11	0	14	3
Total (pcs)				51	38	38	36	40	30

TABLE II. COMPUTATIONAL RESULTS

Production Plan	IMA		TS-H1[4]		WLA [1]	RITM [12]
	Cmax (s)	tCPU (s)	Cmax	tCPU (s)	Cmax	Cmax
1	774	125	776	121.17	907	837
2	774	48	774	44.81	972	870
3	776	35	777	32.52	885	838
4	775	26	775	24.72	940	922
5	963	22	969	19.28	964	999
6	669	18	675	13.73	686	674