

Evolved Discrete Harmony Search Algorithm for Multi-objective No-wait Flow Shop Scheduling Problem

Xie Guang
 Hainan University Sanya College
 Sanya, China, 572022
 lovesnowy31@163.com

Li Junqing
 School of Computer Science
 Liaocheng University
 Liaocheng, China, 252059
 lijunqing@lcu.edu.cn

Abstract—In this paper, an evolved discrete harmony search (EDHS) is proposed. Firstly, a job-permutation-based encoding scheme is applied to enable the continuous harmony search algorithm to be used in all sequencing problems. Additional, a new method is proposed to generate new solutions, while an efficient approach is developed to update the archive set of the non-dominated solutions during the search process. Finally, computational simulation results based on the well-known benchmarks show that the proposed EDHS algorithm is superior to hybrid differential evolution algorithm in terms of searching quality, diversity level and efficiency.

Keywords- Harmony search, No-wait flow shop, Multi-object optimization

I. INTRODUCTION

The no-wait flow shop scheduling problem is one of the most typical problems with strong engineering background. In the no-wait flow shop scheduling problem, each of n jobs consists of m operations and each one will be processed on m machines in the sequence continuously. Thus, in order to meet the no-wait requirement the start of a job on the first machine must be delayed. In the past decades, efforts have been dedicated to solving the problems with single objective. However, the multi-objective no-wait flow shop scheduling problem has a wider practical application. Therefore, it is more important to develop the technologies and approaches for this problem. With the advent of just in time, we find Pareto optimal solutions for no-wait flow shop scheduling problems with the minimize the maximum completion time, the total flow time and the maximum tardiness criteria.

Harmony Search (HS) is a heuristic optimization method for complex continuous nonlinear functions which is music-inspired. The harmony in music is analogous to the solution vector. The musician's improvisations are looked as local and global search schemes. Due to its simplicity, easy implementation and quick convergence, the HS algorithm has gained much attention and a wide range of successful applications.

II. MULTI-OBJECT NO-WAIT FLOW SHOP SCHEDULING PROBLEM

The problem considered in this research is described as follows. Each job $j \in J = \{1, 2, \dots, n\}$ will be through each machine $i \in M = \{1, 2, \dots, m\}$ in a sequence. Suppose that a job permutation $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ represents a schedule of

jobs to be processed. Let the processing time of job j on machine i be $p(\pi_j, k)$. Let $e(\pi_{j-1}, \pi_j)$, $d(\pi)$ be the minimum delay on the first machine between the start of jobs π_{j-1} and π_j and due date respectively. The formula is given as follows:

$$e(\pi_{j-1}, \pi_j) = p(\pi_{j-1}, 1) + \max\left\{0, \max_{2 \leq k \leq m} \left[\sum_{h=2}^k p(\pi_{j-1}, h) - \sum_{h=1}^{k-1} p(\pi_j, h) \right] \right\} \quad (1)$$

The completion time of job π_j on machine m can be computed by the following formula :

$$C(\pi_j, m) = \sum_{k=1}^m p(\pi_j, k) \quad (2)$$

$$C(\pi_j, m) = \sum_{i=2}^j e(\pi_{i-1}, \pi_i) + \sum_{k=1}^m p(\pi_j, k) \quad j = 2, 3, \dots, n \quad (3)$$

The makespan $C_{\max}(\pi)$, maximum tardiness $T_{\max}(\pi)$ and total flow time $TF(\pi)$ can be computed by the following formula :

$$f_1(\pi) = C_{\max}(\pi) = C(\pi_n, m) = \sum_{j=2}^n e(\pi_{j-1}, \pi_j) + \sum_{k=1}^m p(\pi_n, k) \quad (4)$$

$$f_2(\pi) = T_{\max}(\pi) = \max_{j=1}^n (\max\{0, C(\pi_j, m) - d(\pi_j)\}) \quad (5)$$

$$f_3(\pi) = TF(\pi) = \sum_{j=1}^n C(\pi_j, m) = \sum_{j=2}^n (n+1-j)e(\pi_{j-1}, \pi_j) + \sum_{j=1}^n \sum_{k=1}^m p(\pi_j, k) \quad (6)$$

III. THE EVOLVED DISCRETE HS(EDHS)

A EDHS algorithm which is evolved in the discrete space is proposed. Let HMS , $HMCR$, PAR represent harmony memory size, harmony memory considering rate, pitch adjusting rate respectively. $X^i = [x_1^i, x_2^i, \dots, x_n^i]$, denote the i th harmony solution. The detail of the EDHS algorithm is explained as follows.

A. update solution

The initial solutions are generated based on job-permutation. According to the method, the solutions can be updated as follows:

$$x^i = HMCR \otimes g(x^r, x^b) = \begin{cases} g(x^r, x^b) & Rnd < HMCR \\ x^r & otherwise \end{cases} \quad (7)$$

x^b is one of the pareto solutions. $g()$ represents two-point crossover with the probability of $HMCR$. The x^r denotes the temporary solution which is generated by

learning harmony solutions. The $h()$ represents insert operation.

The disturbance operation as follows:

$$x' = PAR \otimes h(x') = \begin{cases} h(x') & Rnd < PAR \\ x' & otherwise \end{cases} \quad (8)$$

(1) Two-point crossover

In the two-point crossover, a block of jobs from x^b is determined by two-cut points randomly. This block is either moved to the right or left corner of the solution. Then the offspring permutation is filled out with the remaining jobs from x^r . This procedure will always produce two distinctive offspring from the same two parents. In this paper, the better offspring is chosen.

(2) The procedure of generate x^r

The x^r is a temporary solution which is generated by learning all the harmony individuals. One of the solutions in the non-dominated solutions is selected to be x^b . The detail of the procedure is shown in the fig 1:

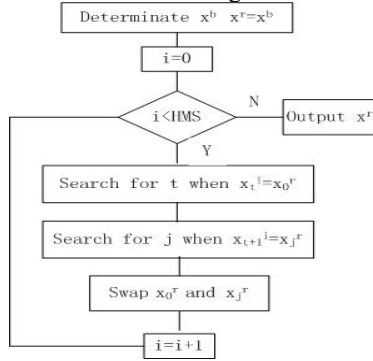


Figure 1. generate x^r

B. The Procedure of EDHS

The HS algorithm generates one solution once. In order to improve its efficiency, the algorithm generates ten solutions once. Based on the above design, the procedure of the EDHS algorithm is summarized as follows.

- Step 1: Set the parameters such as HMS , $HMCR$, PAR and Initialize solutions.
- Step 2: Evaluate each solution. Initialize AS.
- Step 3: $i = 0$
- Step 4: Generate new solution using the method of Section 3.1.
- Step 5: if $i < 10$, go back to step 4.
- Step 6: Update AS and the first ten solutions with new solution.
- Step 7: Update solutions which is dominated by new solution.
- Step 8: If a stopping criterion is met, output AS, otherwise go back to Step 3.

IV. COMPUTATIONAL RESULTS

A. Comparison Setup

In order to test the algorithm proposed in this paper, 31 well-known flow shop benchmark instances [1-3] are modified for the problems. The related data is given as follows: $HMCR = 0.95$, $PAR = 0.1$, $HMS = n$. Maximum iterating times $t = 10 * m * n$. The EDHS algorithm is coded in C++. All experiments are conducted on a Intel(R) Core(TM) i3 CPU, 2GHZ PC with 2GB memory. For each instance, we carried out 20 independent replications and calculate average distance (Dl_r), Number of non-dominated solutions ($N_{NDS}(S_j)$), Ratio of non-dominated solutions ($R_{NDS}(S_j)$), due date ($d(\pi)$) [4]. Let S_j ($j = 1, 2$) to be the non-dominated solution set, which obtained by algorithm J , S^* denote the reference solution set, which obtained by the formula $S^* = \cup S_j$. Dl_r can be used to evaluate the spread and distribution of S_j to S^* . $N_{NDS}(S_j)$ is the total number of those solutions in S_j that are not dominated by any other solution in S^* . $R_{NDS}(S_j)$ can be used to evaluate the quality of the solutions in S_j .

B. Computational Results

We compare our EDHS algorithm with HDE [5]. The HDE algorithm is applied to the problems under consideration by modifying its objective evaluation and its parameters were fixed at the same ones as given in their papers. The average value (AVG), minimum value (MIN), maximum value (MAX), and the standard deviation (SD) of these metrics in 20 replications are also calculated as the statistics for the performance measures. The results are shown in Tab. 1, Tab. 2 and Tab. 3.

(1) It is clear from Table 1 that the EDHS algorithm is the better one in terms of the overall mean average, min, max and standard deviation of D1R in comparison to HDE. For all instances, EDHS algorithm yields significantly better results. So, we can say that the EDHS algorithm can nearly find all the reference solutions for these instances.

(2) The EDHS algorithm has the advantage of the HDE algorithm in that the EDHS algorithm produces much better overall mean AVG, MIN, MAX and SD of N_{NDS} and R_{NDS} than the HDE algorithm from table 2 and table 3. The two metrics generated by the EDHS algorithm is much higher than the HDE algorithm. This means that for these instances, almost every solution of the HDE algorithm is dominated by some solutions of the EDHS algorithm. So it is concluded that the EDHS algorithm can produce much more non-dominated solutions with good performances than the HDE

V. CONCLUSION

This paper addresses to solve multi-object no-wait flow shop scheduling problem to minimize the maximum completion time, the total flowtime and the maximum tardiness .We presented a discrete harmony search optimization algorithm which worked in the discrete domain by employing a job-permutation-based encoding scheme. a new method of producing new solutions. In order to obtain a good approximate Pareto-optimal set, the EDHS algorithm dynamically updated an archive set of the non-dominated solutions during the search process. Computational simulations and comparisons demonstrated the superiority of the proposed algorithms in terms of solution quality.

REFERENCES

[1] Carlier J. Ordonnancements a contraintes disjonctives. *RAIRO Recherche Operationelle*[J] 1978;12:333-51.
 [2] Reeves C. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*[J], 1995,22:5-13.
 [3] Heller J. Some numerical experiments for an M×J flow shop and its decision-theoretical aspects. *Operations Research*[J], 1960;8:178-84.
 [4] PAN Quan-ke, WANG Ling, QIAN Bin. A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. *Computers & Operations Research*[J], 2009, 36(8): 2498 - 2511.
 [5] QIAN Bin, WANG Ling, HUANG De-xian, WANG Wan-liang, WANG Xiong. An effective hybrid DE-based algorithm for multi-objective flowshop scheduling with limited buffers. *Computers & Operations Research*[J] 36 (2009) 209 – 233.

TABLE I. AVERAGE DISTANCE Dl_r OF THE EDHS AND HDE ALGORITHMS

instance		HDE				EDHS			
Name	n×m	AVG	MIN	MAX	SD	AVG	MIN	MAX	SD
Car01	11×5	25.77	15.89	38.36	5.66	9.97	5.37	22.49	4.76
Car02	13×4	20.59	14.45	26.27	3.52	9.33	5.00	22.02	3.77
Car03	12×5	17.42	12.00	26.51	4.06	9.21	5.65	20.12	3.41
Car04	14×4	40.77	25.86	53.21	6.67	14.89	4.77	38.16	7.71
Car05	10×4	15.81	10.04	24.96	3.54	7.90	5.42	20.56	3.69
Car06	8×9	6.95	3.40	18.66	3.68	1.33	0.00	7.38	1.89
Car07	7×7	6.94	0.90	19.05	4.27	0.92	0.00	9.97	2.78
Car08	8×8	9.75	4.31	17.75	3.39	1.54	0.22	9.07	2.29
Hel1	100×10	224.75	196.61	259.67	18.39	7.65	0.00	38.93	10.35
Hel2	20×10	37.89	23.34	49.13	6.56	15.25	10.03	28.75	4.74
Rec01	20×5	34.34	23.78	49.97	7.51	12.20	5.73	29.21	5.55
Rec03	20×5	39.41	24.06	51.98	7.58	7.33	0.31	20.91	5.60
Rec05	20×5	51.56	28.24	73.97	11.28	15.59	9.16	33.08	6.41
Rec07	20×10	50.29	26.62	74.76	14.13	9.37	3.24	22.30	5.41
Rec09	20×10	55.88	41.11	83.77	11.69	7.77	0.00	28.09	6.66
Rec11	20×10	50.27	21.10	75.93	15.63	7.30	1.41	16.99	4.39
Rec13	20×15	103.84	58.79	137.57	21.70	12.21	0.00	31.74	9.68
Rec15	20×15	90.20	26.09	128.24	28.57	14.36	9.16	20.28	3.34
Rec17	20×15	56.36	33.20	73.54	11.36	7.61	0.00	26.80	7.57
Rec19	30×10	98.82	57.63	141.87	24.11	11.43	0.00	39.47	11.52
Rec21	30×10	67.22	42.05	85.63	11.63	12.59	4.66	30.79	7.98
Rec23	30×10	67.93	30.08	94.79	17.54	8.99	1.89	20.98	5.55
Rec25	30×15	78.32	35.17	116.48	16.96	10.62	0.28	29.05	7.73
Rec27	30×15	39.97	29.27	51.97	5.73	15.46	9.51	27.36	5.79
Rec29	30×15	454.94	248.83	702.10	126.09	96.43	0.00	166.24	59.86
Rec31	50×10	151.05	108.52	205.02	25.81	10.79	0.00	36.61	8.99
Rec33	50×10	104.06	48.57	146.47	24.00	13.36	0.00	27.83	6.56
Rec35	50×10	88.61	61.63	113.63	14.67	12.26	0.00	25.09	8.83
Rec37	75×20	219.53	141.76	274.89	35.89	13.00	0.00	28.56	7.10
Rec39	75×20	177.77	134.25	223.09	25.99	12.51	0.00	38.73	10.05
Rec41	75×20	96.26	72.05	119.69	14.28	13.10	3.46	27.80	6.16
Mean		83.33	51.60	114.80	17.16	12.98	2.75	30.50	7.94

TABLE II. NUMBER OF NON-DOMINATED SOLUTIONS N_{NDS} OF THE EDHS AND HDE ALGORITHMS

instance		HDE				EDHS			
Name	n×m	AVG	MIN	MAX	SD	AVG	MIN	MAX	SD
Car01	11×5	2.05	0.00	8.00	2.65	19.45	5.00	27.00	5.11
Car02	13×4	3.20	0.00	11.00	3.21	20.50	9.00	27.00	5.30
Car03	12×5	5.45	0.00	24.00	5.99	26.00	6.00	40.00	8.49
Car04	14×4	0.90	0.00	4.00	1.17	15.60	7.00	19.00	2.91

Car05	10×4	6.00	1.00	13.00	3.96	25.85	9.00	31.00	5.99
Car06	8×9	10.65	3.00	14.00	2.66	16.85	9.00	18.00	2.11
Car07	7×7	13.80	8.00	19.00	2.95	19.15	9.00	21.00	2.76
Car08	8×8	16.60	10.00	25.00	3.98	32.70	16.00	36.00	4.74
Hel1	100×10	0.00	0.00	0.00	0.00	16.25	9.00	19.00	3.60
Hel2	20×10	0.75	0.00	6.00	1.77	18.25	11.00	25.00	3.74
Rec01	20×5	0.40	0.00	2.00	0.75	19.00	7.00	25.00	3.96
Rec03	20×5	0.50	0.00	2.00	0.76	31.05	16.00	40.00	8.03
Rec05	20×5	0.25	0.00	3.00	0.72	9.95	8.00	17.00	1.99
Rec07	20×10	0.80	0.00	10.00	2.35	18.15	9.00	24.00	3.53
Rec09	20×10	0.00	0.00	0.00	0.00	18.60	5.00	25.00	5.18
Rec11	20×10	0.25	0.00	2.00	0.64	22.20	10.00	30.00	4.75
Rec13	20×15	0.05	0.00	1.00	0.22	14.05	9.00	17.00	2.67
Rec15	20×15	0.25	0.00	4.00	0.91	9.10	6.00	11.00	1.71
Rec17	20×15	0.25	0.00	3.00	0.72	22.80	13.00	30.00	4.47
Rec19	30×10	0.00	0.00	0.00	0.00	18.55	8.00	26.00	5.19
Rec21	30×10	0.05	0.00	1.00	0.22	11.60	8.00	17.00	2.66
Rec23	30×10	0.20	0.00	2.00	0.62	17.20	13.00	20.00	2.14
Rec25	30×15	0.05	0.00	1.00	0.22	18.35	7.00	26.00	5.26
Rec27	30×15	0.45	0.00	6.00	1.36	23.95	8.00	30.00	5.38
Rec29	30×15	0.15	0.00	2.00	0.49	13.55	7.00	20.00	4.73
Rec31	50×10	0.10	0.00	2.00	0.45	15.10	8.00	21.00	3.68
Rec33	50×10	0.15	0.00	3.00	0.67	18.70	14.00	24.00	2.74
Rec35	50×10	0.60	0.00	4.00	1.35	22.20	10.00	30.00	5.20
Rec37	75×20	0.00	0.00	0.00	0.00	12.00	7.00	16.00	1.95
Rec39	75×20	0.00	0.00	0.00	0.00	25.45	13.00	35.00	6.11
Rec41	75×20	0.10	0.00	2.00	0.45	20.50	15.00	33.00	4.75
Mean		2.06	0.71	5.61	1.33	19.12	9.39	25.16	4.22

TABLE III. RATIO OF NON-DOMINATED SOLUTIONS R_{NDS} OF THE EDHS AND HDE ALGORITHMS

instance		HDE				EDHS			
Name	$n \times m$	AVG	MIN	MAX	SD	AVG	MIN	MAX	SD
Car01	11×5	0.16	0.00	1.00	0.27	0.96	0.56	1.00	0.12
Car02	13×4	0.17	0.00	0.58	0.15	0.95	0.76	1.00	0.07
Car03	12×5	0.27	0.00	0.83	0.28	0.87	0.40	1.00	0.19
Car04	14×4	0.06	0.00	0.29	0.08	0.98	0.83	1.00	0.05
Car05	10×4	0.28	0.04	0.60	0.17	0.93	0.69	1.00	0.08
Car06	8×9	0.70	0.19	0.93	0.16	0.99	0.82	1.00	0.04
Car07	7×7	0.87	0.56	1.00	0.11	0.93	0.64	1.00	0.08
Car08	8×8	0.78	0.58	0.94	0.11	0.99	0.89	1.00	0.03
Hel1	100×10	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00
Hel2	20×10	0.05	0.00	0.50	0.13	0.99	0.93	1.00	0.02
Rec01	20×5	0.03	0.00	0.20	0.05	0.99	0.88	1.00	0.03
Rec03	20×5	0.03	0.00	0.14	0.05	0.99	0.91	1.00	0.02
Rec05	20×5	0.02	0.00	0.23	0.05	0.99	0.82	1.00	0.04
Rec07	20×10	0.04	0.00	0.53	0.12	0.99	0.75	1.00	0.06
Rec09	20×10	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00
Rec11	20×10	0.03	0.00	0.40	0.10	1.00	0.95	1.00	0.01
Rec13	20×15	0.00	0.00	0.04	0.01	1.00	1.00	1.00	0.00
Rec15	20×15	0.03	0.00	0.44	0.10	1.00	1.00	1.00	0.00
Rec17	20×15	0.01	0.00	0.14	0.03	0.99	0.90	1.00	0.03
Rec19	30×10	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00
Rec21	30×10	0.00	0.00	0.05	0.01	1.00	1.00	1.00	0.00
Rec23	30×10	0.02	0.00	0.22	0.06	1.00	1.00	1.00	0.00
Rec25	30×15	0.00	0.00	0.06	0.01	1.00	0.92	1.00	0.02
Rec27	30×15	0.03	0.00	0.46	0.10	1.00	1.00	1.00	0.00
Rec29	30×15	0.02	0.00	0.33	0.08	1.00	1.00	1.00	0.00
Rec31	50×10	0.01	0.00	0.10	0.02	1.00	1.00	1.00	0.00
Rec33	50×10	0.01	0.00	0.20	0.04	1.00	1.00	1.00	0.00
Rec35	50×10	0.04	0.00	0.27	0.09	1.00	1.00	1.00	0.00
Rec37	75×20	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00
Rec39	75×20	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00
Rec41	75×20	0.01	0.00	0.15	0.03	1.00	1.00	1.00	0.00
Mean		0.12	0.04	0.34	0.08	0.99	0.89	1.00	0.03