engineering proceedings

MDPI

Proceeding Paper

Differential Evolution Algorithm to Solve the Parallel Batch Processing Machine Scheduling Problem with Multiple Jobs †

Xue Zhao, Yarong Chen *, Mudassar Rauf * and Chen Wang

School of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou 325035, China; zx2710750937@163.com (X.Z.); wangchen32130@163.com (C.W.)

* Correspondence: yarongchen@126.com (Y.C.); mudassarrauf16@gmail.com (M.R.)

† Presented at the Third International Conference on Advances in Mechanical Engineering 2023 (ICAME-23), Islamabad, Pakistan, 24 August 2023.

Abstract: We conducted this study with the aim of resolving the scheduling problem of parallel batch processing machines (PBPM) with different capacity constraints and different energy consumption per unit of time, as well as jobs with different processing times, arrival times, delivery dates and sizes, with the goal of simultaneously minimizing the maximum completion time, ET and total energy consumption. The IUDRLM rule is used to batch and sort jobs, and a decomposition-based multi-objective differential evolution algorithm MODE/D is proposed. Simulation experiments are performed to compare the performance of the proposed algorithm to those of existing algorithms. The proposed MODE/D algorithm outperformed NSGA-III in terms of NR value (0.96) and IGD (6.6) measures.

Keywords: parallel batch processor; multi-objective scheduling; DE algorithm; energy consumption

check for updates

Citation: Zhao, X.; Chen, Y.; Rauf, M.; Wang, C. Differential Evolution Algorithm to Solve the Parallel Batch Processing Machine Scheduling Problem with Multiple Jobs. *Eng. Proc.* **2023**, *45*, 22. https://doi.org/10.3390/engproc2023045022

Academic Editors: Mohammad Javed Hyder, Muhammad Mahabat Khan, Muhammad Irfan and Manzar Masud

Published: 11 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The parallel batch processing machine scheduling problem (PBPMSP) is essential to production scheduling problems. Therefore, studying the PBPMSP is of great significance in the context of intelligent manufacturing.

Extensive literature can be found on PBPMSP. However, the literature on PBPMSP, such as its different capacity constraints and energy consumption, is relatively scarce. Majumder et al. [1] minimized the maximum completion time PBPMSP and designed a cuckoo algorithm based on the discrete Levy Flights strategy to improve local search ability and enhance solution diversity. Wang et al. [2] considered the deterioration effect constraint of the machine and proposed an improved algorithm based on the drosophila algorithm to solve the problem. Zhou et al. [3] studied the scheduling problem affecting parallel batch processors using dynamic arrival and usage time pricing schemes. They designed a multi-objective DE algorithm to solve large-scale problems. Li et al. [4] generated individuals through heuristic rules and adopted an angle-based environmental selection strategy to select individuals. They designed an improved algorithm based on a genetic algorithm to minimize both delay and total pollution emission costs.

It can be surmised from the above literature that research on PBPMSPs mainly focuses on minimizing the completion time, total process time and other single objectives under the same capacity constraints of batch processors. Therefore, this article studies the multi-objective PBPMSP by placing different capacity constraints on batch processors and varying energy consumption per unit of time, and it proposes a multi-objective differential evolution (MODE/D) algorithm.

2. Materials and Methodology

2.1. Problem Description

The problem studied in this paper is described as follows: n jobs needed to be processed on m PBPM. The job had the different processing time p_j, arrival time r_j, job size s_j and delivery time d_j. The capacity constraints Q_i of PBPM were different, and the energy consumption per unit of time varied due to machine updates. At the same time, three optimization objectives involved in minimizing total lead time/lag time ET, maximum completion time C_{max} and total energy consumption TEC were considered.

The scheduling problem was denoted as $P_m \left| Q_i, p_j, r_j, d_j, s_j \right| C_{max}, ET, TEC$; the main decisions included determining the batch mode of the job assigned to the machine and the processing sequence of the batches. The basic assumptions to be met were as follows: (1) The sum of all the job sizes in the batch B_b could not exceed the capacity constraint of the batch machine. (2) The batch processor could not be interrupted while processing. (3) The batch arrival time was determined based on the latest arrival time of the job in the batch; the batch processing time was equal to the maximum processing time of the job in the batch.

2.2. Multi-Objective Differential Evolution Algorithm Based on Decomposition

The DE algorithm is a simple, rapid and efficient global optimization algorithm that has been successfully applied in communication and scheduling. This article combined the characteristics of the PBPM multi-objective scheduling problem, decomposed multi-objective problems into single-objective subproblems and introduced the DE algorithm to design the MODE/D algorithm.

2.2.1. Initial Solution Generation

In this paper, the scheduling solution of the 2 × n array representation problem was generated via double-layer coding. The first layer represented the random number generated via each job gene in the interval [−1, 1], which constituted the job-ordering layer. The second layer represented the machines assigned to the job and randomly generated an integer value in the interval [1, m] for each job, with m representing the number of machines.

2.2.2. Decoding and Target Value Calculation

Zhou et al. [5] designed three heuristic batch rules of first row start (FRS), minimum distance start (MDS) and updated distance (UD). In this paper, UD batch rules were selected to enable improvement according to the characteristics of PBPMSP, and the improved UD-right-light moving (IUDRLM) decoding rules were designed.

Direct use of the UD rules could not optimize target values other than C_{max}. For PBPM multi-objective scheduling problems considering artifacts, if two jobs p_j, r_j and d_j were very close, and the sum of s_j of the two jobs was less than or equal to Q_i, the two jobs were assigned to the same batch. The IUD rule distance designed in this paper was calculated via Formula (1).

$$d(j,w) = \begin{cases} \sqrt{\alpha\left(p_j - p_w\right)^2 + \beta\left(r_j - r_w\right)^2 + \gamma\left(d_j - d_w\right)^2} & s_j + s_w \le Q_i \text{ and } j < w \\ \infty & s_j + s_w \le Q_i \text{ or } \quad j \ge w \end{cases} \quad (1)$$

We used the IUDRLM rule to sort all jobs in batches, before using randomly generated 0–1 variables to sequentially decide whether to perform left and right movement operations on each processing batch/block and, finally, obtain the scheduling solution.

2.2.3. Differential Mutation

The job-ordering layer used the formula $V_i^t = X_a^t + F \times \left(X_b^t - X_c^t\right)$ to perform differential mutation operations, where F was the shrinkage factor, and the value of F could

be determined through pre-experiments. The job distribution machine layer adopted the method of two-point variation and carried out the variation operation on the job distribution machine layer of individual X_a^t, randomly generated two-position numbers, and it exchanged the machine information of the corresponding gene position of the job distribution machine layer to obtain the job distribution machine layer after the mutation. Assuming that F = 0.5, the mutation operation shown in Figure 1 was used.

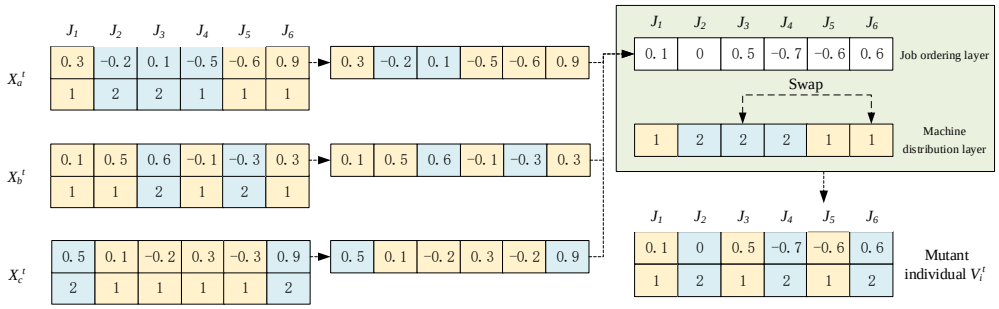


Figure 1. Differential mutation operation.

2.2.4. Crossover

In the interval [0, 1], n random numbers were randomly generated to form a 1 × n-dimensional array *rand*. When the value contained in each position in the *rand* array was less than or equal to the cross probability CR, the gene at the position of the test individual U_i^t directly inherited the corresponding position gene of the mutant individual V_i^t. Otherwise, the gene at the position of the test individual U_i^t directly inherited the gene at the corresponding position of the target individual X_i^t.

2.2.5. Multi-Objective Solution Set Update Based on Decomposition

After the difference variation and cross-operation, the test individual U_i^t was obtained, and the three target values of the test individual U_i^t were calculated. The aggregation function decomposition method was used to update the neighborhood and accelerate the convergence rate of the population.

2.3. Data Generation

The experimental problem associated with data generation is shown in Table 1.

Table 1. The scale and parameter range of the experimental problem.

Instance	n	m	p_j	r_j	d_j	s_j
Small	10, 20	2	U[1, 100]	U[0, 50]	$d_j = p_j + U[d_{min}, d_{min} + \rho \times P]/2$, $d_{min} = P \times (\tau - \rho/2)$, $P = \sum_{j=1}^{n} p_j$, $\tau = 0.5$, $\rho = 0.05$	U[1, 10]
Medium	40, 60	2				
Large	80, 100	3				

Six sets of experiments were generated for different artifacts and machine combinations, with each set generating 10 test data for use in experiments, resulting in a total of 60 test data.

3. Analysis of Results

The NSGA-III algorithm was selected as the comparison algorithm, and the algorithm's performance was evaluated using the Nondomination rate (NR) index and the inverted generation distance (IGD) index. The larger the NR value, the smaller the IGD value, as well as the better the comprehensive performance of the algorithm. This study's experimental results are shown in Table 2.

Table 2. NR and IGD values of MODE/D and NSGA-III algorithms.

Instance	n	m	NR		IGD	
			MODE/D	NSGA-III	MODE/D	NSGA-III
Small	10	2	**0.82**	0.47	**6.3**	19.2
	20	2	**1.00**	0.00	**0.0**	161.7
Medium	40	2	**1.00**	0.00	**0.0**	671.5
	60	2	**1.00**	0.00	**0.0**	1547.7
Large	80	3	**1.00**	0.00	**0.0**	978.1
	100	3	**0.96**	0.04	**33.3**	1026.2

Table 2 shows that for six instances of three scales, the NR values of the MODE/D algorithm are greater than those of the NSGA-III algorithm, while the IGD values are all smaller than those of the NSGA-III algorithm. This result indicates that the performance of the MODE/D algorithm is superior to that of the NSGA-III algorithm. As the MODE/D algorithm can decompose the multi-objective optimization problem into a single objective subproblem, it can achieve a better distribution and convergence effect on the Pareto frontier.

4. Conclusions

In this paper, the PBPM multi-objective optimization problem was studied, which simultaneously minimizes the three objectives of C_{max}, ET and TEC. The MODE/D algorithm was designed by decomposing multi-objective subproblems into single-objective subproblems and introducing the DE algorithm. The comprehensive performance of the algorithm was evaluated through simulation experiments, and the final results showed that the designed MODE/D algorithm was better than the comparison algorithm NSGA-III. In future research, we will consider the impact of dynamic events and other dynamic scheduling methods to better meet the actual production needs of enterprises.

Author Contributions: Conceptualization, X.Z. and Y.C.; methodology, X.Z. and M.R.; software, X.Z. and C.W.; validation, X.Z. and C.W.; formal analysis, X.Z. and Y.C.; writing—review and editing, Y.C. and M.R.; visualization, M.R.; supervision, X.Z.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: The National Natural Science Foundation of China, grant number [No.51705370].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Majumder, A.; Laha, D.; Suganthan, P.N. A hybrid cuckoo search algorithm in parallel batch processing machines with unequal job ready times. *Comput. Ind. Eng.* **2018**, *124*, 65–76. [CrossRef]
2. Wang, R.; Jia, Z.; Li, K. Scheduling parallel-batching processing machines problem with learning and deterioration effect in fuzzy environment. *J. Intell. Fuzzy Syst.* **2021**, *40*, 12111–12124. [CrossRef]
3. Zhou, S.; Li, X.; Du, N.; Pang, Y.; Chen, H. A multi-objective differential evolution algorithm for parallel batch processing machine scheduling considering electricity consumption cost. *Comput. Oper. Res.* **2018**, *96*, 55–68. [CrossRef]
4. Li, K.; Zhang, H.; Chu, C.; Jia, Z.; Chen, J. A bi-objective evolutionary algorithm scheduled on uniform parallel batch processing machines. *Expert Syst. Appl.* **2022**, *204*, 117487. [CrossRef]
5. Zhou, S.; Chen, H.; Xu, R.; Li, X. Minimising Makespan on a single batch processing machine with dynamic job arrivals and non-identical job sizes. *Int. J. Prod. Res.* **2014**, *52*, 2258–2274. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.