Proceeding of the 11th World Congress on Intelligent Control and Automation
Shenyang, China, June 29 - July 4 2014

A Modified RBM-Based Estimation of Distribution Algorithm for Steelmaking-Continuous Casting Scheduling[*]

Shenglong Jiang, Min Liu, and Jinghua Hao

Department of Automation
University of Tsinghua
Beijing, P.R. China
sh.l.jiang.jx@gmail.com

***Abstract* – In order to solve steelmaking-continuous casting scheduling problem which is considered as a critical module of planning and scheduling in modern iron and steel enterprise, this paper propose a data-based evolutionary algorithm. Firstly, a mixed integer programming model considering practical constraints and multi-objectives is provided. Secondly, we develop a modified estimation of distribution algorithm based on restricted Boltzmann machine, which incorporates a backward list scheduling method, a solution representation with job permutation within each batch, and a local search method used to enhance the performance of this algorithm. To verify the proposed approach, a number of instances from real-world industrial data are generated. The final numerical results show that the proposed data-based algorithm is efficient and effective to solve practical steelmaking-continuous casting scheduling problems.**

***Index Terms* - RBM; EDA; steelmaking; hybrid flow shop; scheduling.**

I. INTRODUCTION

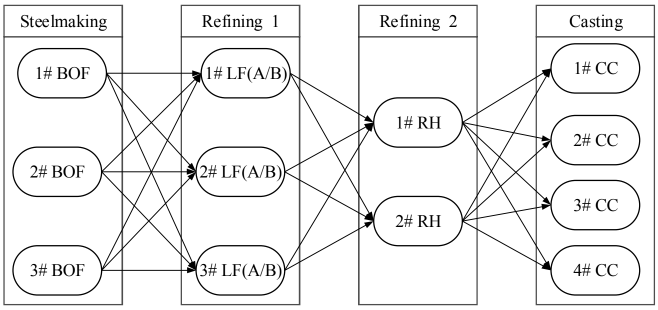

Fig. 1 A steelmaking-continuous casting manufacturing process

Steelmaking-continuous casting (SM-CC) manufacturing process usually is the cornerstone and bottleneck section in the modern iron and steel enterprise. SM-CC scheduling problem is to determine job sequence, machine assignment and timetable from the first stage to the last stage. Fig.1 shows the SM-CC manufacturing process of a steelmaking plant in the eastern China. It consists of three types of stages: (1) steelmaking stage, including three Basic Oxygen Furnaces (BOF); (2) two refining stages, including three double-satiation ladle furnaces (LF) and two Ruhrstahl-Hausen (RH) vacuum refining furnaces, respectively; (3) continuous casting stage, including three continuous casters (CC).

SM-CC scheduling problem, which is regarded as a special hybrid flow shop scheduling problem (HFSSP) with continuity constraints at the last stage [1], is an important research topic both in the academic and industrial community. All the methods solving SM-CC scheduling problem can be divided into three types: exact, heuristics and meta-heuristic methods.

Inspired by the just-in-time (JIT) principle, Tang [1] developed a non-linear model which was converted into linear programming model. Bellabdaoui and J.Teghem [2] proposed a mixed-integer linear programming formulation focusing on a real-world application. Both of them can be solved by standard software packages easily. Xuan and Tang [3] decomposed a scheduling problem formulated with integer programming into batch-based sub-problems by Lagrangian relaxation method. Furthermore, they developed a dynamic programming approach to solve those sub-problems. Wang X et al [4] optimized a preliminary schedule considering the same usage rate of machines and minimization of time conflict, and then eliminated time conflict with a linear programming model. Pacciarelli and Pranzo [5] proposed an alternative graph model to describe all the constraints, and a beam search algorithm to solve the model. Atighehchian [6] developed a hybrid algorithm, which determined job sequence and allocation by ant colony optimization (ACO) method, and a timetabling algorithm for all jobs on assigned machines through a non-linear programming model. Tang and Wang [7] presented an improved particle swarm optimization (PSO) algorithm to solve a HFSSP with steelmaking industrial background. To optimize total waiting time in the SM-CC scheduling problem, Zhao et al. [8] presented a taboo search (TS) algorithm to optimize the sub-problem including job sequence and machine assignment decision, and used a linear programming for decoding and calculating start time of all operations. According to some problem-specific characteristics, Pan and Wang [9] presented a job-permutation-based artificial bee colony (ABC) algorithm combined with several improved heuristic procedures. In this

[*] This work is supported by the National Natural Science Foundation of China (Nos. 61025018, 60834004, 61021063, 61104172), the National Key Basic Research and Development Program of China (2009CB320602), and the National Science and Technology Major Project of China (2011ZX02504-008).

978-1-4799-5825-2/14/$31.00 ©2014 IEEE

paper, the SM-CC scheduling problem is decomposed into parallel machine scheduling problem and HFSSP, which is different from the job sequence, machine assignment and timetabling sub-problems reported in most existing literatures.

The rest of this paper is organized as follows. In section II, the formalized model of SM-CC scheduling problem studied in this paper is provided. Section III will describe the modified restricted Boltzmann machine-based estimation of distribution algorithm (RBM-EDA) solution methodology. In section IV, numerical experimental results are reported. Finally, some conclusions are given.

II. Problem Formulation

A. Notation

i Stage index, where $i \in \{1, 2 \cdots, g\}$.

j Job index, where $j \in \{1, 2 \cdots, n\}$.

l Batch index, where $l \in \{1, 2, \cdots h\}$, and the batch sequence is predefined.

$m_{i,k}$ The k^{th} machine at the i^{th} stage, where nk_i is the number of machines at the i^{th} stage.

$\Omega_{j,l}$ The j^{th} job within the l^{th} batch.

Ω_l Job set of the l^{th} batch.

n_l Number of jobs in the l^{th} batch, and $n = \sum_{l=1}^{h} n_l$.

$p_{i,j}$ Processing time of job j at the i^{th} stage.

u_l Setup time of the l^{th} batch before processing.

t_{i_1,i_2} Transport time from stage i_1 to stage i_2 .

$s_{i,j}$ Start time of job j at the i^{th} stage.

$c_{i,j}$ Completion time of job j at the i^{th} stage.

d_l Start time of the l^{th} batch given by the planner.

γ_1 Cost coefficient of the total waiting time inter-stages.

γ_2 Cost coefficient of the total tardiness/earliness to the predefined start time.

L "Very large" positive constant.

B. Mathematic Model

1) Decision variables:

$s_{i,j}$ The start time of job j at stage i .

$$x_{i,j,k} = \begin{cases} 1 & \text{if job } j \text{ is assigned to the machine } k \text{ at stage } i. \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{i,j_1,j_2} = \begin{cases} 1 & \text{if job } j_1 \text{ precedes job } j_2 \text{ at stage } i. \\ 0 & \text{otherwise.} \end{cases}$$

2) Objectives:

With the above notations, two objectives of the SM-CC scheduling problem studied in this paper can be formulated as follows.

$$f_1 = \gamma_1 \sum_{j=1}^{n} \sum_{i=1}^{g-1} \left(s_{i+1,j} - t_{i,i+1} - c_{i,j} \right) \tag{1}$$

$$f_2 = \gamma_2 \sum_{l=1}^{h} \left(max\left(0, s_{g,\Omega_{1,l}} - d_l \right) + max\left(0, d_l - s_{g,\Omega_{1,l}} \right) \right) \tag{2}$$

$$minimize \ f = f_1 + f_2 \tag{3}$$

3) Constraints:

$$\sum_{k=1}^{nk_i} x_{i,j,k} = 1 \qquad \forall i, \forall j \tag{4}$$

$$y_{i,j_1,j_2} + y_{i,j_2,j_1} = 1 \quad \forall i, \forall j_1, \forall j_2, j_1 \neq j_2 \tag{5}$$

$$c_{i,j} = s_{i,j} + p_{i,j} \qquad \forall i, \forall j \tag{6}$$

$$s_{i,j_2} - c_{i,j_1} - \left(3 - x_{i,j_1,k} - x_{i,j_2,k} - y_{ij_1j_2}\right) L \geq 0 \quad \forall i, \forall j_1, \forall j_2, j_1 \neq j_2 \tag{7}$$

$$s_{i+1,j} - c_{i,j} \geq t_{i,i+1} \quad \forall i \in \{1, 2 \ldots, g-1\}, \forall j \tag{8}$$

$$\sum_{k=1}^{nk_g} \sum_{j=1}^{n_l} x_{g,\Omega_{j,l},k} = n_l \qquad \forall l \tag{9}$$

$$c_{g,\Omega_{j,\ l}} = s_{g,\Omega_{j+1,l}} \qquad \forall l, \forall j \in \{1, 2 \ldots, n_l - 1\}, \forall k \in \{1, 2 \ldots, nk_g\} \tag{10}$$

$$c_{g,\Omega_{j1,l1}} - (2 - x_{g,\Omega_{j_1,l_1},k} - x_{g,\Omega_{j_2,l_2},k}) L \leq s_{g,\Omega_{j_1,l_2}} - u_{l_2} \quad \forall l_1, \forall l_2, l_1 < l_2, \forall j_1 \in \Omega_{l_1}, \forall j_2 \in \Omega_{l_2}, k \in \{1, 2 ..., nk_g\} \tag{11}$$

In the aforementioned formulation, objective function (3) is made up of two terms: the total cost of cumulative waiting time of inter-stages and cumulative batch earliness/tardiness. Constraint (4) ensures that a job is processed on only one machine at each stage. Constraint (5) ensures the unique precedence relationship between any two different jobs at each stage. Constraint (6) defines the completion time as an auxiliary variable of any job at each stage. Constraint (7) guarantees that a machine cannot process more than one job at a time. Constraint (8) defines the successive operation between two adjacent stages for the same job. Constraint (9) ensures that all jobs in the same batch must be assigned to the same machine at the last stage. Constraint (10) ensures that the two adjacent jobs in same batch must be continuously processed. Constraint (11) ensures enough setup time before starting a batch.

It is obvious that the presented model is a nonlinear programming problem, which contains many binary, integer and continuous variables. And the nonlinearity comes from the equation (2). In order to eliminate this nonlinearity, we replace constraint (12) and (13) with another equation (14) [1].

$$T_l = max\left(0, s_{g,\Omega_{1,l}} - d_l\right) \qquad \forall l \tag{12}$$

$$E_l = max\left(0, d_l - s_{g,\Omega_{1,l}}\right) \qquad \forall l \tag{13}$$

$$T_l - E_l = s_{g,\Omega_{1,l}} - d_l \qquad \forall l \tag{14}$$

III. The Modified RBM-EDA Algorithm

The main problem of using mathematic model stated above as an efficient solver is its model structure with a great number of binary variables $x_{i,j,k}$ and y_{i,j_1,j_2} . Since these problems are usually NP-Complete, which means that no method can solve this kind of problems within accepted computing time. Unlike most literatures which decomposed SM-CC scheduling problem into two sub-problems, sequence job and assign machine at first, and then timetabling, we divide SM-CC scheduling problem into two sub-problems: parallel machine scheduling problem at last stage and HFSSP of upstream stages, because it can be considered as a special HFSSP with continuity constraints at the last stage. We propose a modified RBM-EDA method to solve the first sub-problem, and design a Backward List Scheduling (BLS) method for the second sub-problem.

A. A Brief View of Modified RBM-EDA

Recently, estimation of distribution algorithms (EDAs) have received more attentions, since its promising prospect in evolutionary computation field [10]. These algorithms explore the search space by building a probabilistic model from a set of better candidate solutions, and the probabilistic model will be used to sample new solutions. So far, there exist a large number of papers solving scheduling problems with EDAs, such as job shop [11], flow shop [12], flexible job shop [13][14], hybrid flow shop [15][16], and so on. To explain how to employ different probabilistic models, Shih-Hsin Chen [10] compared different types of probabilistic models and pointed out the future research directions of EDAs solving scheduling problem. Since the dependencies among decision variables are in favour of search for the optimum in the solution space, Vim and Tan proposed a novel EDA via training restricted Boltzmann machine (RBM) and constructing probability matrix by the energy-model directly[17][18].

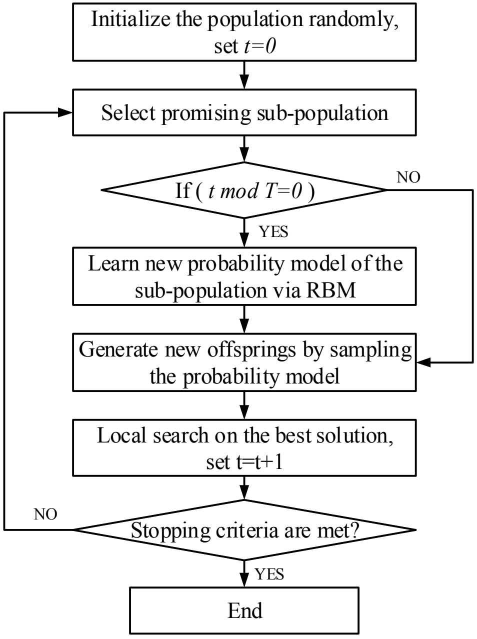
Initialize the population randomly, set t=0
Select promising sub-population
If (t mod T=0)
NO
YES
Learn new probability model of the sub-population via RBM
Generate new offsprings by sampling the probability model
Local search on the best solution, set t=t+1
NO
Stopping criteria are met?
YES
End

Fig. 2 The flowchart of RBM-EDA

To solve SM-CC scheduling problem, we introduce an EDA with modifying RBM training procedure proposed by Vim and Tan, and the detail will be described in section 3.4. The general framework of the modified RBM-EDA is illustrated in Fig. 2. Since the training procedure of RBM is time-consuming, we construct the new probability model every T generations, where T is bigger than 1. To enhance the performance of the algorithm, a local search procedure is performed at each generation.

B. Solution Representation and Initialization

The input of SM-CC scheduling problem is the batch set Ω with n jobs that is given by planning module of MES. The start time of each batch is also predefined, but job order and machine assignment of each batch need to be determined. According to those characters of SM-CC scheduling problem, a job permutation of each batch which denotes a solution of SM-CC scheduling problem is proposed. Fig. 3 shows a representation example of 3 batches which contain 5 jobs respectively. In the 1st batch, the job sequence is $[1,3,2,4,5]$ in the 2nd batch, the job sequence is $[3,2,4,5,1]$ and the job sequence in the 3rd batch is $[4,3,2,1,5]$.

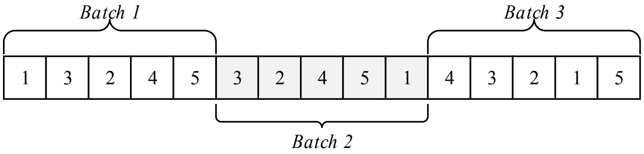

Fig. 3 Representation of SM-CC scheduling solution

Unlike the standard EDA beginning with a population of initial individuals by an evenly distributed probability matrix, a randomly shuffle procedure is employed for initializing population.

Step 1 Generate an initial individual $\pi(0)$ whose job permutation in each batch is sorted with longest processing time (LPT) rule, and let the counter $Ck = 1$.

Step 2 If $Ck < N$, randomly choose a batch from $\pi(0)$, and randomly produce a new job permutation in the selected batch. Else, exit this procedure.

Step 3 If the new individual is same as previous one, then skip this repetitive solution. Else, set $Ck = Ck + 1$, put it into the initial population.

Step 4 Go to Step 2.

C. BLS Method for Upstream Stages Scheduling Problem

After the job permutation of each batch is determined, the solution can be decoded as follows.

Step 1 Parallel machine scheduling at the last stage.

Step 1.1 Set $i = g$, calculate the release time rh_l of the l^{th} batch at the last stage, where

$$rh_l = \max\left(d_l , \sum_{i=1}^{g-1} (p_{i,\Omega_{1,l}} + t_{i,i+1}) \right)$$

Step 1.2 For batch $l = 1$ to h , do

Assign the first available machine $m_{g,k}$ to the l^{th} batch and calculate start time of each job in the l^{th} batch, as follows.

$$s_{g,\Omega_{j,l}} = max\left(AT_{g,k} + u_k, rh_l\right) + \sum_{p=0}^{p-1} p_{g,\Omega_{p,l}}$$

Where $AT_{g,k}$ is the available time of machine $m_{g,k}$ when a batch assigned. And its initial value is 0.

Step 2 Backward scheduling for upstream stages.

While $i != 1$ set $i = i-1$, do.

Step 2.1 Calculate the latest completion time of each job at the i^{th} stage according to the start time of the job j a*t the* $(i+1)^{th}$ *s*tage.

$$LCT_{i,j} = s_{i+1,j} - t_{i,i+1}$$

Step 2.2 Sort all jobs at the i^{th} stage according to $LCT_{i,j}$ in descending order, then select machine with the latest start time for job j .

$$m_{i,k} = arg\left(\max_{k\in\{1,2...nk_i\}}\left(LCT_{i,j} - p_{i,j}, RAT_{i,k} - p_{i,j}\right)\right)$$

Where $RAT_{i,k}$ is the reverse available time of the machine at the i^{th} stage. And its initial value is positive infinity.

Step 3 Right shift.

If $min\left(s_{i,j}\right) < 0$, then right shift all operations for $-min\left(s_{i,j}\right)$ units.

D. Probability Model Construct and Sampling

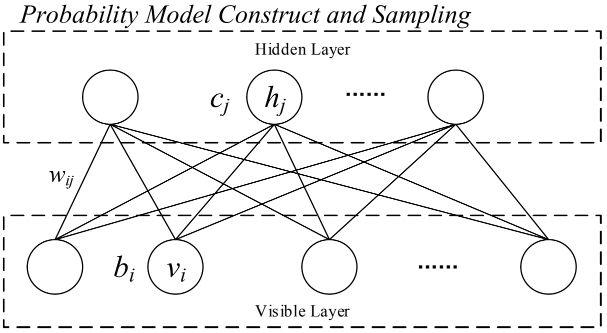

Fig. 4 The structure of RBM

RBM (Fig. 4) is a free energy-based neural network with binary nodes that can learn a probability distribution from large-scale input data set. Recently, with the development of efficient training algorithms for RBM (i.e., contrastive divergence, CD), it has been widely applied in machine learning fields, such as feature extraction, dimensionality reduction, classification, collaborative filtering, and so on. Since those advantages of RBM, H Tang and VA proposed an RBM-based EDA [17] [18]. After the CD training algorithm is executed, the parameters of RBM, weights, biases, and hidden states, are obtained, the visible states can be estimated by trained model. So, we encode probability of each decision variable by RBM model, of which conditional probability is constructed by unobserved latent variables. This procedure is different from energy statistics method proposed at [17]. According to this idea, the probability model of EDA is obtained as follows.

Step 1 Convert each solution into binary vector. Since the visible units of classic RBM are binary, job j at i^{th} position within the l^{th} batch should be converted into a binary vector with length of n_l . For example, the job permutation $[3,1,2]$ in some batch is transferred to $[0,0,1,1,0,0,0,1,0]$.

Step 2 Construct different h RBM models corresponding to h batches, and train them with CD-k algorithm respectively.

Step 3 Construct probability model as follows.

Firstly, calculate the conditional probability of a visible unit being one as follows.

$$p\left(v_i = 1 \mid h\right) = \sigma\left(\sum_{i=1}^{n} w_{ij}h_j + b_i\right)$$

Where $\sigma(x) = 1/(1+e^{-x})$ is a sigmoid activation function. Then re-convert the binary representation into integer representation, and compute the probability distribution of the jobs of the l^{th} batch at generation t , denoted as $Pr_t(\pi_l)$, as follows.

$$Pr_t(\pi_{l,i} = j) = \frac{p(\pi_{l,i} = j \mid h)}{\sum_{i=1}^{n_l} p(\pi_{l,i} = j \mid h)}$$

Finally, the probability matrix is constructed as follows.

$$Pr_t(\pi_l) = \begin{bmatrix} Pr(\pi_{l,1} = 1) & \cdots & Pr(\pi_{l,n_l} = 1) \\ \vdots & \ddots & \vdots \\ Pr(\pi_{l,1} = n_l) & \cdots & Pr(\pi_{l,n_l} = n_l) \end{bmatrix}$$

Generate new candidate solutions with sampling $\Pr_t(\pi)$ as follows:

Step 1 Generate a random number $rand_i$ between 0-1 with uniform distribution.

Step 2 Set $\pi_{l,i} = 0$, which means no job is selected at the i^{th} position of the l^{th} batch ;

Step 3 For $p = 1$ to n_l .

If $\sum_{k=1}^{p-1} \Pr_t(\pi_{l,i} = j) \le rand_i \le \sum_{k=1}^{p} \Pr_t(\pi_{l,i} = j)$ and $j \notin \pi_l$, then $\pi_{l,i} = p$;else $p = 1$.

E. Local Search Procedure

To enhance the exploitation ability of the RBM-EDA, three types of local search operators are employed to produce some neighbouring solutions, including swap, shift and inversion, which are described as follows:

1) $swap(l,x,y)$ Swaps two randomly selected elements in the sub-vector of the l^{th} batch.

2) $shift(l,x,y)$ Randomly chooses two different elements from the sub-vector of the l^{th} batch, and insert the latter before the former.

3) $inversion(l,x,y)$ Inverts a randomly selected sub-sequence between $[x,y]$ or $[y,x]$ in the sub-vector of the l^{th} batch.

In each local search step, the above three operators are performed several times separately, and the new solution replaces the old one only when it has a better objective. It is noted that one of above operators which are applied on the best solution in the current population will change to another one while the objective not improved for a certain number of iterations.

IV. Computational Experiments

In this section, the effectiveness and efficiency of proposed algorithm is tested on several real-world scheduling instances which are generated from statistics data in MES database. The predefined start time of each batch at the last stage is estimated by the equation as follows.

$$d_l = max\left(AT_{g,k}+u_l, \sum_{i=1}^{g-1}\left(p_{i,\Omega_{1,l}}+t_{i,i+1}\right)\right)$$

Where $AT_{g,k}$ denotes the available time of machine $m_{g,k}$. Noted that this method leads to a relatively compact start time of each batch, because it ignores constraints of upstream stages. The processing time of each job at each stage is randomly generated according to the statistic result from the database. The processing times on BOFs, LFs, RHs and CCs are stochastic integer numbers generated with uniform distributions, $U(30,30)$, $U(30,50)$, $U(25,40)$, $U(30,40)$ respectively. The transport times from the first to the last stage are $t_{1,2}=10$, $t_{2,3}=10$, $t_{3,4}=10$ and $t_{2,4}=12$. Because the RH stage can be skipped if the steel grade of jobs is not required. The skipping rate of RH stage is set to 0.7 according to the statistical results. The setup time of each batch $u_l=60$. The constant coefficients $\gamma_1=0.1$ and $\gamma_2=1.0$. According to the machine capacity of BOFs, the maximum output of the steelmaking plant only comes to about 144 jobs in one day. So, for experimental evaluation and comparison, we generate 10 problem instances with 10 batches, and each batch includes 10~20 jobs subject to technological constraints.

To evaluate the modified RBM-EDA, two algorithms have been applied for the regular HFSSP for comparison. These are slope index-based heuristic (SIBH) proposed by YING [19], and standard EDA (sEDA). The relative percentage deviation (RPD) which is considered to be another performance measure of algorithms is given as follows.

$$RPD = \frac{f_{SIBH}-f_{a\lg o}}{f_{SIBH}}\times 100\%$$

TABLE I
COMPARISON OF TEST ALGORITHMS

Instance	SIBH	sEDA		RBM-EDA	
		obj	*RPD* (%)	*obj*	*RPD* (%)
1.	998.4	867.8	13.1	834.6*	16.4
2.	1014.5	874.6	13.8	845.6*	16.6
3.	1042.1	849.2*	18.5	853.6	18.1
4.	1008.5	875.7	13.2	861.5*	14.6
5.	1115.6	894.9	19.8	872.2*	21.8
6.	970.4	819.0	15.6	799.3*	17.6
7.	1103.2	936.7	15.1	895.9*	18.8
8.	1118.2	937.6	16.2	897.3*	19.8
9.	1058.8	892.9	15.7	862.4*	18.5
10.	976.4	867.7	11.1	843.8*	13.6

Table I shows that the computational results comparison among test algorithms, including SIBH, sEDA and RBM-EDA. Both sEDA and RBM-EDA exceed the performance of SIBH. And RBM-EDA improves the performance of sEDA slightly in most situations. Since probability distribution of the decision variables in the solution space is more complex than a simple distribution function, such as uniform, Gaussian distribution, and the RBM model have learned and utilized some favourable features of dominant solutions, the RBM-EDA is more likely to access to the optimal solution.

V. Conclusion

In this paper, we studied a real-world SM-CC scheduling problem, which is aim to find an optimal feasible solution to minimize the total waiting time and total earliness/tardiness of all batches' start time guaranteeing the practical constraints. Additionally, we designed a modified RBM-EDA which employed a machine learning method in guiding search procedure to find the optimal job permutation of each batch, a BLS procedure which is used to timetabling all operations at upstream stages, and a local search for improving the solution's quality in each generation. Experimental results show that the proposed algorithm can obtain satisfactory solution in acceptable time for industrial applications.

References

[1] ANG L, LIU J, RONG A, et al. A mathematical programming model for scheduling steelmaking-continuous casting production [J]. European Journal of Operational Research, 2000, 120(2): 423–435.

[2] Bellabdaoui a., Teghem J. A mixed-integer linear programming model for the continuous casting planning [J]. International Journal of Production Economics, 2006, 104(2): 260-270.

[3] Xuan H, Tang L. Scheduling a hybrid flowshop with batch production at the last stage [J]. Computers & operations research, 2007, 34(9): 2718-2733.

[4] Wang X, Chai T, Zheng B, et al. Two-stage optimized scheduling method and application for steelmaking and continuous casting[C].IEEE, 2010: 6109-6114.

[5] Pacciarelli D, Pranzo M. Production scheduling in a steelmaking-continuous casting plant [J]. Computers & chemical engineering, 2004, 28(12): 2823-2835.

[6] Atighehchian A, Bijari M, Tarkesh H. A novel hybrid algorithm for scheduling steel-making continuous casting production [J]. Computers & operations research, 2009, 36(8): 2450-2461.

[7] Tang L, Wang X. An Improved Particle Swarm Optimization Algorithm for the Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time in Process Industry [J]. Control Systems Technology, IEEE Transactions on, 2010, 18(6): 1303-1314.

[8] Zhao Y, Jia F, Wang G. A hybrid tabu search for steelmaking-continuous casting production scheduling problem [J]. Advanced Control of, Proceedings of the 2011 4th International Symposium on Advanced Control of Industrial Processes, Thousand Islands Lake, Hangzhou, P.R. China, May 23-26, 2011: 535-540.

[9] Pan, Q.-K., Wang, L., Mao, K., Zhao, J.-H., & Zhang, M. (2012). An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. IEEE Transactions on Automation Science and Engineering, 1–16.

[10] CHEN S, CHEN M. Addressing the advantages of using ensemble probabilistic models in Estimation of Distribution Algorithms for scheduling problems [J]. International Journal of Production Economics, 2013, 141(1): 24–33.

[11] ZHANG R, WU C. A double-layered optimisation approach for the integrated due date assignment and scheduling problem [J]. International Journal of Production Research, 2012, 1(1): 5–22.

[12] CHEN S, CHANG P, CHENG T C E, et al. A Self-guided Genetic Algorithm for permutation flowshop scheduling problems [J]. Computers and Operation Research, Elsevier, 2012, 39(7): 1450–1457.

[13] WANG S, WANG L, ZHOU G, et al. A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem [J]. Computers & Industrial Engineering, 2012, 62(4): 917–926.

[14] WANG S, WANG L, LIU M, et al. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem [J]. International Journal of Production Economics, 2013, 145: 387–396.

[15] Salhi A, Rodríguez J A V, Zhang Q. An estimation of distribution algorithm with guided mutation for a complex flow shop scheduling problem[C]. Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, 2007: 570-576.

[16] WANG S, WANG L, LIU M, et al. An enhanced estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machines [J]. The International Journal of Advanced Manufacturing Technology, 2013, 68(9-12): 2043–2056.

[17] Tang H, Shim V A, Tan K C, et al. Restricted Boltzmann machine based algorithm for multi-objective optimization[C]. Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, 2010: 1-8.

[18] Shim V A, Tan K C, Tan K K. A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem[C]. Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, 2012: 1-8.

[19] YING K-C, LIN S-W. Scheduling multistage hybrid flowshops with multiprocessor tasks by an effective heuristic [J]. International Journal of Production Research, 2009, 47(13): 3525–3538.