APEM journal

Advances in Production Engineering & Management
Volume 12 | Number 1 | March 2017 | pp 62–74
https://doi.org/10.14743/apem2017.1.240

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

A genetic regulatory network-based sequencing method for mixed-model assembly lines

Lv, Y.[a]**, Zhang, J.**[a,*]**, Qin, W.**[a]

[a]School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Mixed-model sequencing to minimize work overload at stations is regarded as one of the most concerned optimization problems in assembly lines manufacturing a variety of product models simultaneously. A novel sequencing method based on the genetic regulatory network is proposed to solve this problem. First, genes, gene regulation equations and gene expression procedures are developed in the network based on its similarity with the mixed-model sequencing problem. Each two-state gene represents a binary decision variable of the mathematical model. The gene regulation equations describe decision variable interactions in the constraints and objectives. The gene expression procedure depends on the regulation equations to generate solutions, in which the value of each decision variable is indicated by the expression state of the related gene. Second, regulatory parameter optimization in the regulation equations minimizes the work overload at stations. The effectiveness of the proposed method is validated through experiments consisting of reference instances and industrial instances. The experimental results demonstrate that this method outperforms other methods in large-scale instances.

ARTICLE INFO

Keywords:
Assembly line
Mixed-model sequencing
Work overload
Genetic regulatory network
Genetic algorithm

**Corresponding author:*
zhangjie@sjtu.edu.cn
(Zhang, J.)

Article history:
Received 11 October 2016
Revised 16 January 2017
Accepted 17 January 2017

1. Introduction

Mixed-model assembly lines (MMALs) are widely applied to the industrial engineering world because they can assemble various models of products in a facultative sequence while reducing setup times [1]. Although it is possible to implement any model sequence, the model sequence causes different economic impacts in the actual environment [2]. For instance, different models require diverging processing times at stations to complete the specific assembly operations that realize customized function requirements [3]. The cycle time defines the standard time to process a product at a station, which is typically the average of the processing times of different models weighted by the model demands [4]. The processing times required in complex operations are thus greater than the cycle time, while those required in simple operations are less than the cycle time [5, 6]. If some complex operations are processed continuously by using a specific station, then assembly tasks may not be completed before the operators have reached the down-stream station border, which is regarded as a work overload situation. Although utility workers or line stoppages are adopted to deal with the work overload situation, these lead to additional costs [7]. The mixed-model sequencing (MMS) problem is thus addressed to minimize the work overload at stations, in which different models of products are arranged by performing alternately complex operations and simple ones at each station [8].

The MMS problem has attracted a lot of attention because of its complexity and practical value. Various types of MMS methods including exact solution procedures, heuristic procedures and meta-heuristics have been proposed by scholars. However, the existing methods can hardly achieve high-quality solution when the problem is from large-scale instances and requires acceptable computational time. This study aims to fill in this gap by proposing a novel method based on the genetic regulatory network (GRN). In a GRN, gene states are the same as decision variable values in the MMS problem. Gene regulations describe the interconnection between genes, which have an analogous function with constraints. The gene expression procedure governed by gene regulations determines gene states iteratively, which is similar to a heuristic sequencing procedure. Based on these similarities, genes are first defined in the GRN to represent decision variables. Second, gene regulation equations are developed to express not only hard constraints in the mathematical model, but also soft constraints derived from certain sequencing rules. The importance of soft constraints is weighted by regulatory parameters in the equations. Third, the gene expression procedure is designed to indicate a heuristic procedure that is specified by regulatory parameters. Finally, the regulatory parameters are optimized to obtain the optimal solutions with minimum work overload at stations.

Thereupon, the key contribution is the extension of GRN applications to assembly line scheduling. A series of computational experiments are conducted to validate the effectiveness of this GRN-based method. The remainder of this paper is organized as follows. A relevant literature review of MMS methods and GRNs is offered in Section 2. A mathematical model of MMS problems is presented in Section 3. A GRN-based sequencing method is given in Section 4. Section 5 contains the experimental results and discussions. Conclusions and future research directions are discussed in Section 6.

2. Literature review

2.1 Mixed-model sequencing problem

In terms of sequencing in MMALs, Boysen et al. [9] provided an integrated review to discuss three fundamental approaches, i.e., MMS, car sequencing and level scheduling. Of these approaches, the MMS methods aim at minimizing sequence-dependent work overload based on a detailed scheduling in MMALs, which have been widely investigated by researchers [10]. These MMS methods can be further classified into four major classes: branch-and-bound computation, exact solution procedure, heuristic procedure, and meta-heuristic [11-13]. The focus of the existing literature was on heuristic procedure and meta-heuristic because it is impractical to implement other methods for large-scale instances [14-16]. For instance, Cano et al. [17] used a scatter search method that selects from 20 priority rules to generate hyper heuristic procedures. Gujjula et al. [18] proposed a heuristic method based on Vogel's approximation method to address large-scale MMS problems. Cortez and Costa [19] developed a set of fast constructive heuristics, two local search procedures and a meta-heuristic to deal with a specific problem featured with worker-dependent processing times. Well-known meta-heuristics such as genetic algorithm and ant colony algorithm have also been employed to solve a variety of MMS problems [20-25]. In general, most heuristic methods used greedy priority rules to construct rapidly a model sequence by appending iteratively alternative models into it. These methods chose products with minimum objective function values at each iteration, which led to intensive increases of the work overload in the last part of the model sequence. Hence, these approaches based on heuristic procedures could obtain good solutions rather than optimal ones. Although other heuristic methods were considered to improve the solution quality (e.g., the work overload increases caused by remaining model copies were taken into account at each iteration), they resulted in the computational effort to be dramatically increased for large-scale problems. Alternatively, meta-heuristic methods could find optimal solutions or near-optimal solutions by globally searching among all the feasible ones. However, the global search caused the computational effort to be increased with the number of model copies as well as the number of stations.

Consequently, it is difficult to generate a near-optimal model sequence with an acceptable computational effort for real large-scale instances in the actual manufacturing environment. In this paper, we propose a novel GRN-based method to solve MMS problems. The critical factors of this method include the description of the MMS problem by using a GRN and the integration of certain validated sequencing rules in the GRN. Based on such a GRN, the proposed method can solve the MMS problem more efficiently than meta-heuristics without compromising the solution quality, especially for large-scale instances.

2.2 Genetic regulatory network

The GRN is a structured network that describes the regulation of gene expression in cells [26]. It has been widely applied by biologists to investigate the dynamic changes of cell morphologies, and has become a hot topic in the past few years. A GRN has at least the following three elements in common: genes, gene regulations and gene expression procedure [27]. Each gene has two alternative states (i.e., the expressed state and the unexpressed state). If a gene is in the expressed state, it has regulatory effects on the states of other genes, which is the primary form of gene regulations. Based upon these regulations, gene expression procedure iteratively converts certain genes in the unexpressed state into ones in the expressed state if there are enough positive regulatory effects on these genes. Since a cell is mainly composed of copied components (e.g., mRNAs and proteins), gene expression procedure finally determines the cell's morphology in accordance with genes in the expressed states [28]. Various formalisms have already been employed to describe GRNs, for instance, Bayesian networks, directed graphs, partial differential equations, Boolean networks, qualitative differential equations, stochastic equations, and rule-based formalisms [29].

3. Problem description

The following assumptions are taken into consideration when constructing the mathematical model:

- The assembly line is a 'moving line' in which the conveyor moves at a constant speed;
- The length of a station is a fixed one (measured by the product passing time), and neighboring stations do not overlap;
- Products are equi-spaced on the line by launching each other after a constant time interval, which is equivalent to the cycle time;
- The operation processing time at a station is not longer than the length of this station;
- The impact of unfinished works on operations at succeeding stations is not taken into consideration;
- Operators return with infinite velocity to the subsequent product;
- The model changeover time is included in the operation processing time;
- To facilitate the presentation, the notations listed in Table 1 are used in the development of the mathematical model. The mathematical model takes the following form.

$$\text{Minimize} \sum_{t=1}^{T} \sum_{k=1}^{K} w_{tk} \tag{1}$$

$$\text{S.T.} \sum_{t=1}^{T} x_{tm} = d_m \quad \forall m \tag{2}$$

$$\sum_{m=1}^{M} x_{tm} = 1 \quad \forall t \tag{3}$$

$$s_{1k} = 0 \quad \forall k \tag{4}$$

$$s_{tk} \geq 0, w_{tk} \geq 0 \quad \forall t, k \tag{5}$$

$$w_{tk} \geq s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} - l_k \quad \forall t, k \tag{6}$$

$$s_{t+1,k} \geq s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} - w_{tk} - c \quad \forall t, k \tag{7}$$

Table 1 Problem's notations

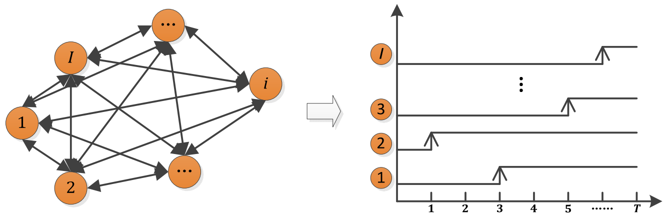
Notations		Definitions
Sets		
	$\{1, \dots, t, \dots, T\}$	Set of products in the model sequence
	$\{1, \dots, k, \dots, K\}$	Set of stations
	$\{1, \dots, m, \dots, M\}$	Set of models
Parameters		
	d_m	Demand for model m in the production plan
	c	Cycle time
	T	Total demand for products, $T = \sum_{m=1}^{M} d_m$
	l_k	Length of station k (time unit)
	p_{mk}	Operation processing time of model m at station k
Variables		
	s_{tk}	Starting time for assembling the tth product at station k
	w_{tk}	Extra operation processing time for the t-th product at station k (work overload time)
	x_{tm}	Binary variable: 1, if the t-th product belongs to model m; 0, otherwise

The objective of the mathematical model is to minimize the total work overload at stations. The constraints in Eq. 2 ensure that the model sequence satisfies the demand for each model. Eq. 3 makes sure that a station cannot process more than one product at the same time within each production cycle. Eq. 4 represents the initial state of stations. Eq. 5 and Eq. 6 ensure that all the operations should be processed within boundaries of their related stations. If operations on a product cannot be finished within boundaries of their related stations, then a work overload occurs. Eq. 7 makes sure that operators at station k can start processing product $(t + 1)$ after they have completed the operations on product t or product t has left station k.

4. Genetic regulatory network-based sequencing method

4.1 Mapping between the mathematical model and the genetic regulatory network

In the mathematical model presented in Section 3, the binary decision variables, the constraints and the solution are similar to the genes, the gene regulations and the gene expression procedure in a GRN, respectively. Based on these similarities, the model can be represented by using a GRN (shown in Fig. 1): (1) each gene represents a decision variable; (2) gene regulations describe constraints; (3) gene expression procedure generates solutions.



Component	i	i ⟷ i'	i:
MMS Problem	decision variable	constraints	model sequence (decision variable value: 0→1)
GRN	gene	gene regulations	gene expression procedure (gene state: unexpressed→expressed)

Fig. 1 Mapping between the MMS problem and the GRN

In general, the differential equation method is the most suitable method to develop gene regulation equations because it can represent gene regulations in a quantitative form. In the GRN, first, regulation equations are used to describe all the constraints, and the gene expression procedure is developed based upon the regulation equations to obtain feasible solutions. Second, some soft constraints related to sequencing rules are integrated in the regulation equations to decrease work overload at stations. Third, regulatory parameters are optimized to integrate reasonable sequencing rules and thus minimize work overload at stations. Consequently, as shown in Fig. 2, the GRN-based method contains two parts:

- A GRN is developed based on the mathematical model and certain sequencing rules;
- Regulatory parameters are optimized by using a genetic algorithm in order to minimize the work overload.

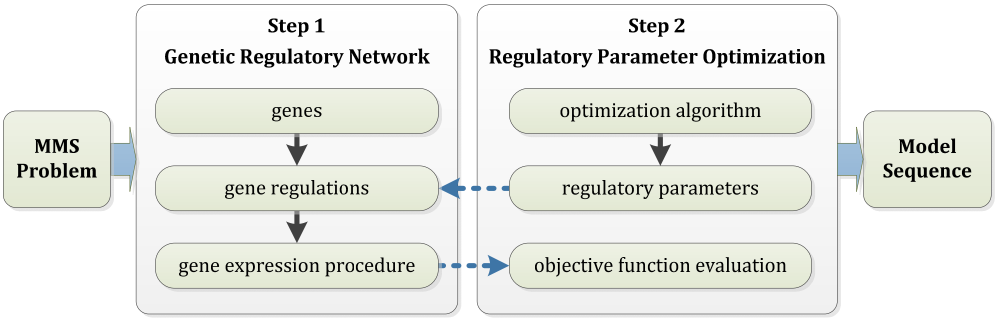

MMS Problem
Step 1
Genetic Regulatory Network
genes
gene regulations
gene expression procedure
Step 2
Regulatory Parameter Optimization
optimization algorithm
regulatory parameters
objective function evaluation
Model Sequence

Fig. 2 Outline of the GRN-based method

4.2 Genetic regulatory network establishment

According to decision variables of the mathematical model, genes $\{\theta_{tm} \mid t = 1,2, \dots, T, m = 1,2, \dots, M\}$ are generated. Each gene θ_{tm} indicates that a product of model m is assigned to the tth position of the model sequence. Moreover, the regulation equation in Eq. 8 is developed to express constraints.

$$v_i = f_i(x_1(t), x_2(t), \cdots, x_I(t), \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_E) \tag{8}$$

where I represents the number of genes in a GRN, $x_i(t)\ \forall i \in \{1,2, \cdots, I\}$ is a binary variable that is equal to 1 if gene θ_i is in the expressed state at time t, otherwise, it is equal to 0, $f_i: \mathbb{R}^{(I+E)} \to \mathbb{R}$ is a nonlinear function, v_i represents the inhibition coefficient to convert gene θ_i to the expressed state, $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_E$ are regulatory parameters. In terms of Eq. 2 and Eq. 3, the regulation equation first describes following constraints of each position in the model sequence:

(1) A model can be selected when other models have not been selected yet.
(2) A model can be selected when the demand for this model has not been satisfied at former positions.

Moreover, soft constraints related to the study of Cano et al. [17] and the study of Dörmer et al. [30] are also included in the regulation equation:

(3) A model can be selected if it causes the least work overload at stations.
(4) A model can be selected if it leads to the least idle time at stations.
(5) A model can be selected if its production ratio best matches its demand ratio in the production plan.

No sequencing procedure could satisfy all the soft constraints completely, and each unsatisfied case might increase the work overload at stations. The regulation equation of gene θ_{tm} is thereby developed as follows:

$$v_{tm} = \varepsilon_1 \sum_{k=1}^{K} \phi(s_{tk} + p_{mk} - l_k) + \varepsilon_2 \sum_{k=1}^{K} \phi(c - s_{tk} - p_{mk})$$

$$+\varepsilon_3 K^{\varepsilon_4} \left| \frac{\sum_{t=1}^{T} x_{tm} + 1}{\sum_{t=1}^{T} \sum_{m=1}^{m} x_{tm} + 1} - \frac{d_m}{T} \right| + H\left(\sum_{m=1}^{M} x_{tm}\right) + H\left(\sum_{t=1}^{T} x_{tm} + 1 - d_m\right) \tag{9}$$

where v_{tm} represents the inhibition coefficient to gene θ_{tm}, $H(x)$ is a step function satisfying $H(x) = 0\ (x < 0)$ and $H(x) = +\infty\ (x \geq 0)$, $\phi\,(x)$ is a piecewise function satisfying $\phi\,(x) = 0\ (x < 0)$ and $\phi\,(x) = x\ (x \geq 0)$, ε_1, ε_2, ε_3, ε_4 represent regulatory parameters that combine regulation segments derived from different constraints. The first three terms of the right side of Eq. 9 indicate the inhibition to gene θ_{tm} owing to soft constraints (3) to (5), respectively. The last two terms of the right side of Eq. 9 describe constraints (1) and (2), respectively. Constraints in Eq. 4 to Eq. 7 are embodied in the calculation of s_{tk}:

$$s_{t+1,k} = \begin{cases} 0 & \text{if} \quad s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} \leq c \\ s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} - c & \text{if} \quad c < s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} \leq l_k \\ l_k - c & \text{if} \quad l_k < s_{tk} + \sum_{m=1}^{M} x_{tm} p_{mk} \end{cases} \tag{10}$$

Based on the regulation equation, the expression procedure of gene θ_{tm} is also developed. As shown in Table 2, at each discrete time $t \in \{1,2,\cdots,T\}$, the inhibition coefficient v_{tm} is calculated for genes $\{\theta_{tm} \mid m = 1,2,\ldots,M\}$ and the gene with minimum v_{tm} is converted to the expressed state. When $t > T$, the model sequence is obtained based on the gene states $\{x_{tm} \mid t = 1,2,\ldots,T,$ $m = 1,2,\ldots,M\}$.

Table 2 Pseudo codes of gene expression procedure

```
//initialization
for t ← 1 to T do
  for m ← 1 to M do
    $x_{tm}$ ← 0              //all the genes are initialized in the unexpressed state
  next;
next;

//gene expression circulation
for k ← 1 to K do
  $s_{1k}$←0                  // initialization of stations
next;
for t ← 1 to T do             //discrete time
  $m_0$ ← 1, $v_0$ ← +∞       //index of the gene with minimum $v_{tm}$
  for m ← 1 to M do
    calculate $v_{tm}$ in Eq. 9      //calculate inhibition coefficients
    if $v_{tm}$ < $v_0$ then         //compare inhibition coefficients
      $m_0$ ← m               //update index
      $v_0$ ← $v_{tm}$        //update the minimum inhibition coefficient
    end if;
  next;
  $x_{tm_0}$ ← 1              //convert the gene with minimum $v_{tm}$ to the expressed state
  for k ← 1 to K do
    calculate $s_{t+1,k}$ in Eq. 10   //update station status
  next;
next;
```

4.3 Regulatory parameter optimization

Based on the GRN established in Section 4.2, a solution to the MMS problem is obtained when regulatory parameter values are determined. In the genetic algorithm illustrated in Fig. 3, first, the initial population is formed by N individuals that are generated randomly. Each individual represents a feasible solution to the MMS problem. A chromosome is a sequence of real numbers to indicate these regulatory parameters (i.e., ε_1, ε_2, ε_3 and ε_4). Letting the current generation to be r, the individuals from the current population $P(r)$ are then selected based on their fitness values. The selected ones will receive the operations of crossover and mutation to generate new individuals. During these operations, each individual has a specific possibility (denoted by *PMutation*) to reinitialize the value of a randomized position in its chromosome, whereas each pair of individuals have a possibility denoted by *PCrossover* to change values of first two positions between their chromosomes. The new population $P(r+1)$ for the next generation $r+1$ is then formed by the newly generated individuals. If the best fitness value in current generation is not better than that in the previous generation, then the search process is terminated.

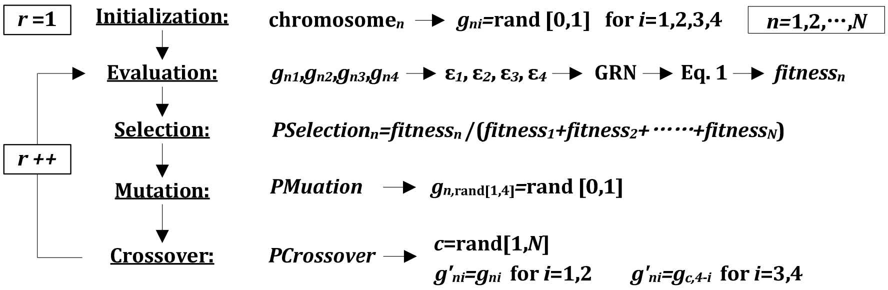

r =1
Initialization:
chromosome$_n$ → g_{ni}=rand [0,1] for i=1,2,3,4
n=1,2,···,N
Evaluation:
g_{n1},g_{n2},g_{n3},g_{n4} → ε$_1$, ε$_2$, ε$_3$, ε$_4$ → GRN → Eq. 1 → fitness$_n$
Selection:
PSelection$_n$=fitness$_n$ / (fitness$_1$+fitness$_2$+······+fitness$_N$)
r ++
Mutation:
PMuation → $g_{n,\mathrm{rand}[1,4]}$=rand [0,1]
Crossover:
PCrossover → c=rand[1,N]
g'_{ni}=g_{ni} for i=1,2 g'_{ni}=$g_{c,4-i}$ for i=3,4

Fig. 3 Regulatory parameter optimization by using a real coded genetic algorithm (RCGA)

5. Computational experiment

In this section, the GRN-based method is applied to reference instances [31] in order to validate its effectiveness. Table 3 presents the processing time structures. These five structures provide processing times as well as station lengths at four stations. Table 4 lists production plans included in these instances. These 45 production plans declare the demand for products of four models, and are divided into five blocks based on models' demand ratios. In addition, the GRN-based method is also applied to industrial instances collected from a powertrain plant [32]. This line is composed of 21 stations that are with the same station length. Table 5 presents processing times of nine engine models at these stations. Forty-six production plans composed of different demand structures are listed in Table 6. These production plans are divided into two blocks based on their total demands.

In addition, an integer coded genetic algorithm (ICGA) [34, 35], a Cplex v11.1 solver and the scatter search based hyper-heuristic (HH) method proposed by Cano et al. [17] are also used to solve these problems. The ICGA encodes the model sequence directly by using a sequence of integer numbers $g_{n1}g_{n2},\cdots,g_{nT}$, in which g_{nt} ($\forall t \in \{1,2,\cdots,T\}$) represents the model of the tth product in the model sequence. Basic steps of this ICGA are illustrated in Fig. 4. The Cplex solver uses a single-processor license to obtain optimal solutions for small-scale instances. To avoid unpredictable computational times for large-scale instances, its CPU time is limited to 7200 s. The HH method uses the HH2-IP10% procedure because it obtains the best results in a series of comparative experiments in Cano et al. [17]. Because the GRN-based method, the ICGA and the HH method depend on the stochastic search procedure, they are repeated 30 times for each instance to obtain the mean results of objective function values and CPU times.

Table 3 Processing time structures in reference instances

Structure		m=1	m=2	m=3	m=4	l_k
1	k=1	92	97	103	108	108
	k=2	103	98	104	95	105
	k=3	101	105	99	95	106
	k=4	95	104	96	105	106
2	k=1	91	80	107	114	115
	k=2	120	105	88	87	120
	k=3	90	113	117	100	120
	k=4	100	107	86	114	115
3	k=1	111	114	83	98	115
	k=2	120	113	85	87	120
	k=3	85	100	115	110	115
	k=4	82	94	119	115	120
4	k=1	113	114	82	95	115
	k=2	119	113	85	87	120
	k=3	115	112	84	94	115
	k=4	116	118	87	81	120
5	k=1	115	104	89	95	115
	k=2	99	119	98	87	120
	k=3	104	100	114	85	115
	k=4	96	102	87	118	118

Table 4 Production plans in reference instances

Block		d_1	d_2	d_3	d_4
1	P_1	13	1	1	1
	P_2	1	13	1	1
	P_3	1	1	13	1
	P_4	1	1	1	13
2	P_5	7	7	1	1
	P_6	7	1	7	1
	P_7	7	1	1	7
	P_8	1	7	7	1
	P_9	1	7	1	7
	P_{10}	1	1	7	7
3	P_{11}	5	5	3	3
	P_{12}	5	3	5	3
	P_{13}	5	3	3	5
	P_{14}	3	5	5	3
	P_{15}	3	3	5	5
	P_{16}	3	3	4	4
	P_{17}	4	4	4	4
4	P_{18}	5	5	5	1
	P_{19}	5	5	1	5
	P_{20}	5	1	5	5
	P_{21}	1	5	5	5
5	P_{22}	1	3	5	7
	P_{23}	1	3	7	5
	P_{24}	1	5	3	7
	P_{25}	1	5	7	3
	P_{26}	1	7	3	5
	P_{27}	1	7	5	3
	P_{28}	3	1	5	7
	P_{29}	3	1	7	5
	P_{30}	3	5	1	7
	P_{31}	3	5	7	1
	P_{32}	3	7	1	5
	P_{33}	3	7	5	1
	P_{34}	5	1	3	7
	P_{35}	5	1	7	3
	P_{36}	5	3	1	7
	P_{37}	5	3	7	1
	P_{38}	5	7	1	3
	P_{39}	5	7	3	1
	P_{40}	7	1	3	5
	P_{41}	7	1	5	3
	P_{42}	7	3	1	5
	P_{43}	7	3	5	1
	P_{44}	7	5	1	3
	P_{45}	7	5	3	1

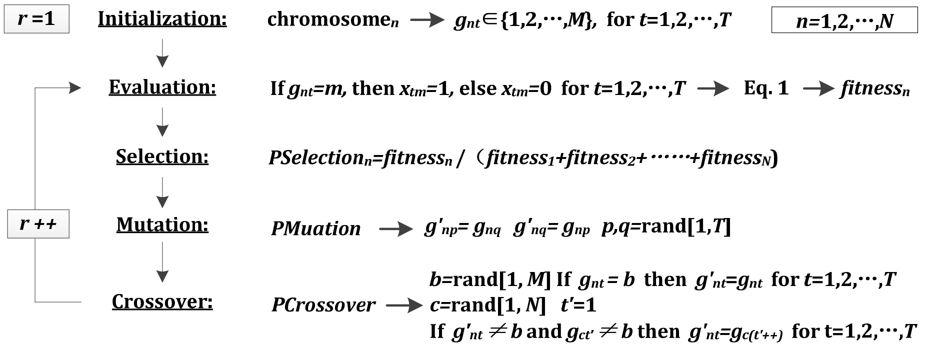

r =1
Initialization:
chromosome$_n$ → g_{nt}∈{1,2,⋯,M}, for t=1,2,⋯,T
n=1,2,⋯,N
Evaluation:
If g_{nt}=m, then x_{tm}=1, else x_{tm}=0 for t=1,2,⋯,T → Eq. 1 → fitness$_n$
Selection:
PSelection$_n$=fitness$_n$ / （fitness$_1$+fitness$_2$+⋯⋯+fitness$_N$)
r ++
Mutation:
PMuation → g'_{np}= g_{nq} g'_{nq}= g_{np} p,q=rand[1,T]
Crossover:
PCrossover → b=rand[1, M] If g_{nt} = b then g'_{nt}=g_{nt} for t=1,2,⋯,T
c=rand[1, N] t′=1
If g'_{nt} ≠ b and $g_{ct'}$ ≠ b then g'_{nt}=$g_{c(t'++)}$ for t=1,2,⋯,T

Fig. 4 Basic steps to solve the MMS problem by using an ICGA

Table 5 Processing times in industrial instances

	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	l_k
k=1	104	100	97	92	100	94	103	100	101	195
k=2	103	103	105	107	101	108	106	102	110	195
k=3	165	156	164	161	148	156	154	164	155	195
k=4	166	175	172	167	168	167	168	156	173	195
k=5	111	114	114	115	117	117	115	111	111	195
k=6	126	121	122	124	127	130	120	121	134	195
k=7	97	96	96	93	96	89	94	101	92	195
k=8	100	97	95	106	94	102	103	102	100	195
k=9	179	174	173	178	178	171	177	171	174	195
k=10	178	172	172	177	178	177	175	173	175	195
k=11	161	152	168	167	167	166	172	157	177	195
k=12	96	106	105	97	101	100	96	104	96	195
k=13	99	101	102	101	99	101	96	102	99	195
k=14	147	155	142	154	146	143	154	153	155	195
k=15	163	152	156	152	153	152	154	156	156	195
k=16	163	185	183	178	169	173	172	182	171	195
k=17	173	179	178	169	173	178	174	175	175	195
k=18	176	167	181	180	172	173	173	168	184	195
k=19	162	150	152	152	160	151	155	148	167	195
k=20	164	161	157	159	162	160	162	158	157	195
k=21	177	161	154	168	172	170	167	149	169	195

Table 6 Production plans in industrial instances

Block		d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
6	P_1	30	30	30	30	30	30	30	30	30
	P_2	30	30	30	45	45	23	23	22	22
	P_3	10	10	10	60	60	30	30	30	30
	P_4	40	40	40	15	15	30	30	30	30
	P_5	40	40	40	60	60	8	8	7	7
	P_6	50	50	50	30	30	15	15	15	15
	P_7	20	20	20	75	75	15	15	15	15
	P_8	20	20	20	30	30	38	38	37	37
	P_9	70	70	70	15	15	8	8	7	7
	P_{10}	10	10	10	105	105	8	8	7	7
	P_{11}	10	10	10	15	15	53	53	52	52
	P_{12}	24	23	23	45	45	28	28	27	27
	P_{13}	37	37	36	35	35	23	23	22	22
	P_{14}	37	37	36	45	45	18	18	17	17
	P_{15}	24	23	23	55	55	23	23	22	22
	P_{16}	30	30	30	35	35	28	28	27	27
	P_{17}	30	30	30	55	55	18	18	17	17
	P_{18}	60	60	60	30	30	8	8	7	7
	P_{19}	10	10	10	90	90	15	15	15	15
	P_{20}	20	20	20	15	15	45	45	45	45
	P_{21}	60	60	60	15	15	15	15	15	15
	P_{22}	20	20	20	90	90	8	8	7	7
	P_{23}	10	10	10	30	30	45	45	45	45
7	P_{24}	60	60	60	60	60	60	60	60	60
	P_{25}	60	60	60	90	90	45	45	45	45
	P_{26}	20	20	20	120	120	60	60	60	60
	P_{27}	80	80	80	30	30	60	60	60	60
	P_{28}	80	80	80	120	120	15	15	15	15
	P_{29}	100	100	100	60	60	30	30	30	30
	P_{30}	40	40	40	150	150	30	30	30	30
	P_{31}	40	40	40	60	60	75	75	75	75
	P_{32}	140	140	140	30	30	15	15	15	15
	P_{33}	20	20	20	210	210	15	15	15	15
	P_{34}	20	20	20	30	30	105	105	105	105
	P_{35}	47	47	40	90	90	55	55	55	55
	P_{36}	74	73	73	70	70	45	45	45	45
	P_{37}	74	73	73	90	90	35	35	35	35
	P_{38}	47	47	40	110	110	45	45	45	45
	P_{39}	60	60	60	70	70	55	55	55	55
	P_{40}	60	60	60	110	110	35	35	35	35
	P_{41}	120	120	120	60	60	15	15	15	15
	P_{42}	20	20	20	180	180	30	30	30	30
	P_{43}	40	40	40	30	30	90	90	90	90
	P_{44}	120	120	120	30	30	30	30	30	30
	P_{45}	40	40	40	180	180	15	15	15	15
	P_{46}	20	20	20	60	60	90	90	90	90

An Intel(R) Core(TM) i7-2720QM CPU @ 2.20 GHz and 8 GB RAM based notebook computer is used to conduct the computational experiments. Table 7 presents genetic algorithm parameters used in the GRN-based method and the ICGA method. Table 8 lists the average objective function values (*Obj*) obtained and the average CPU times (T_{CPU}) spent by the GRN-based method, the ICGA, the HH method and the Cplex solver in each block. In this table, the number of feasible solutions for each instance is evaluated based on Eq. 11 [33].

$$N_f = \left(\sum_{m=1}^{M} d_m\right)!/\prod_{m=1}^{M}(d_m!) \tag{11}$$

d_m is the demand for model m in the production plan. The average number of feasible solutions in each block (N_{fb}) is calculated and also presented in this table as the indicator of problem scales.

Table 7 Genetic algorithm parameters in different methods

Method	Population size	Maximum generation	PMutation	PCrossover
RCGA	50	30	0.1	0.8
ICGA	200	50	0.1	0.8

Table 8 Experimental results of different blocks

Block	N_{fb}	GRN-based		ICGA		HH method		CPLEX solver	
		Obj	T_{CPU}, s	Obj	T_{CPU}, s	Obj	T_{CPU},s	Obj	T_{CPU}, s
1	3.4×10^{3}	247.7	0.4	245.2	0.6	248.6	0.5	245.2	4.2
2	8.2×10^{5}	136.4	0.5	136.7	0.7	137.1	0.6	135.0	39.3
3	3.8×10^{7}	64.5	0.6	68.3	0.9	64.6	0.8	64.3	281.3
4	1.2×10^{7}	100.4	0.6	96.5	1.2	100.7	1.1	96.2	190.8
5	5.8×10^{6}	114.6	0.6	115.2	0.9	114.7	1.5	113.2	112.2
6	5.8×10^{247}	403.3	107.5	497.2	297.2	419.4	138.1	238410.5	7200
7	7.8×10^{505}	856.0	201.1	924.6	578.4	875.6	216.5	245774.1	7200

As shown in Table 8, based on 'N_{fb}' column, two scenarios are considered. Block 1, 2, 3, 4 and 5 are composed of small-scale reference instances and Block 6 and 7 are composed of large-scale industrial instances.

In the small-scale reference instances, the Cplex solver obtains the best results, and the other methods obtain results close to the best ones. Specifically, the results in Block 1 reveal that the GRN-based method and the HH method cannot generate the optimal solutions in some instances owing to the predetermined sequencing rules, while the ICGA can obtain the optimal ones through global searching procedure when there are a few feasible solutions. However, the results in Block 2, 3, 4 and 5 reveal that ICGA fails to obtain the optimal solutions for some instances when the number of feasible ones is increased, while the GRN-based method and the HH method generate better solutions than the ICGA. This is because the sequencing rules integrated in the GRN enable the RCGA to search among good solutions rather than all the feasible ones in the regulatory parameter optimization procedure. Similarly, the HH method uses the scatter search to select from different combinations of sequencing rules and thus searches among good solutions too. However, the GRN-based method achieves better results because its weighted integration of commonly-used sequencing rules enables better searching capacity than the random combination of 20 priority rules in the HH method.

For the large-scale instances, the Cplex solver fails to obtain good results in the limited CPU time, while the other methods achieve better ones in a reasonable time. The results in Block 6 and 7 reveal that the ICGA can hardly find even near-optimal solutions in an enlarged solution space, while the GRN-based method and the HH method are better than the ICGA. In addition, the results also demonstrate the GRN-based method saves the CPU time. In comparison with the ICGA method, the GRN-based method optimizes four regulatory parameters rather than the whole model sequence to decrease computational effort. This regulatory parameter optimization also demonstrates better efficiency than the scatter search on 20 priority rules in the HH method.

Fig. 5 illustrates how the computational time of different methods changes with the increase of problem sizes. The CPLEX solver finds out the optimal solution by using a traversal procedure, for which the computational time increases significantly with a larger problem size. The GRN-based method, the HH method and the ICGA are based on random searching procedures that demonstrate lower increasing rates than the CPLEX solver. Moreover, the ICGA searches among all feasible solutions, whereas the GRN-based method and the HH method search among good solutions owing to the predetermined sequencing rules. For this reason, these two methods save the computational time than the ICGA.

Consequently, it can be noted that the GRN-based method provides an effective means to solve the MMS problem, especially for large-scale instances. In addition, this method is also potentially applied for MMS problems in the dynamic environment by using the predictive-reactive strategy. In this strategy, the GRN-based method first provides a production plan with the minimum work overload before line production, and gives the reactive schedule within a rolling window (containing 10~20 products) once the predetermined plan is interrupted by dynamic events such as machine failures or processing time variations. Because the small-sized MMS problems are regularly solved within 1 s by using the GRN-based method, this reactive schedule realizes real-time responses for the dynamic events. In this way, by using a predictive schedule to ensure the overall performance and employing reactive schedules to make quick responses, the GRN-based method will realize efficient production in the dynamic environment.

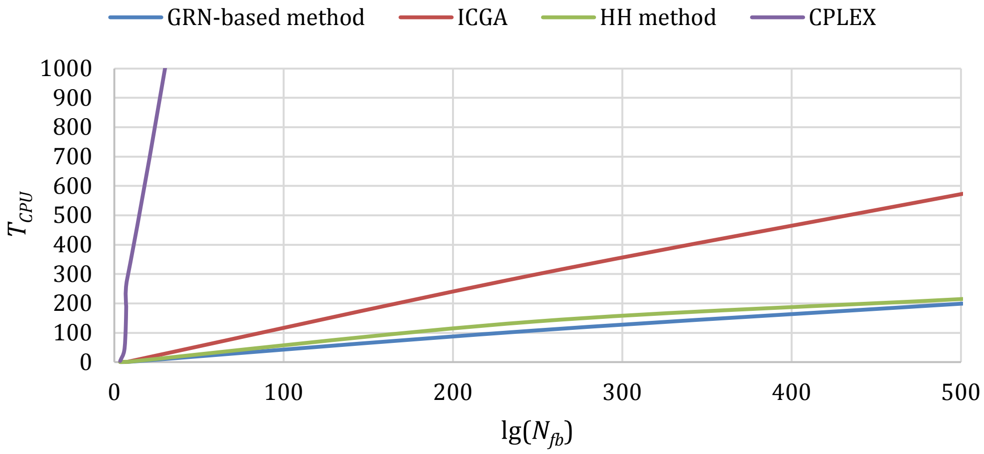


Fig. 5 Computational time with different problem sizes

6. Conclusion

This paper deals with the MMS problem in assembly lines to minimize work overload at stations. In terms of similarities between MMS and GRN, a novel MMS method based on the GRN is proposed. This method is applied to reference instances as well as industrial instances to validate its effectiveness. A Cplex solver, an ICGA and a HH method are used to benchmark the results. It is demonstrated that the GRN-based method realizes higher solution quality than other methods by integrating the sequencing rules reasonably, especially for large-sized problems. However, due to the regulatory parameter optimization that uses GA, rather than some well-designed mechanisms, the efficiency of this method requires further improvements. Thereupon, we will investigate a new parameter optimization mechanism in our future work. In addition, some other optimization problems also have the validated rules to determine its binary decision variables. The proposed method is thus potentially used for the problems in other areas, including the production scheduling problem in other manufacturing systems, the transportation scheduling problem in logistics industry and the medical device scheduling problem in healthcare industry. In our future work, we will develop new scheduling methods by extending the GRN-based approach to these areas.

Acknowledgement

This work was supported by the [National Natural Science Foundation of China] under Grant [number 51435009 & number U1537110].

References

[1] Moradi, H., Zandieh, M., Mahdavi, I. (2011). Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem, *International Journal of Production Research*, Vol. 49, No. 12, 3479-3499, doi: 10.1080/00207540903433882.

[2] Mukund Nilakantan, J., Ponnambalam, S.G. (2015). Robotic U-shaped assembly line balancing using particle swarm optimization, *Engineering Optimization*, Vol. 48, No. 2, 231-252, doi: 10.1080/0305215X.2014.998664.

[3] Lian, K., Zhang, C., Gao, L., Shao, X. (2012). A modified colonial competitive algorithm for the mixed-model U-line balancing and sequencing problem, *International Journal of Production Research*, Vol. 50, No. 18, 5117-5131, doi: 10.1080/00207543.2011.653453.

[4] Akpinar, S., Bayhan G.M. (2014). Performance evaluation of ant colony optimization-based solution strategies on the mixed-model assembly line balancing problem, *Engineering Optimization*, Vol. 46, No. 6, 842-862, doi: 10.1080/0305215X.2013.806915.

[5] Boysen, N., Fliedner, M., Scholl, A. (2009). Production planning of mixed-model assembly lines: Overview and extensions, *Production Planning & Control*, Vol. 20, No. 5, 455-471, doi: 10.1080/09537280903011626.

[6] Chan, F.T.S., Wong, T.C., Chan, L.Y. (2009). An evolutionary algorithm for assembly job shop with part sharing, *Computers & Industrial Engineering*, Vol. 57, No. 3, 641-651, doi: 10.1016/j.cie.2008.11.017.

[7] Boysen, N., Kiel, M., Scholl, A. (2011). Sequencing mixed-model assembly lines to minimise the number of work overload situations, *International Journal of Production Research*, Vol. 49, No. 16, 4735-4760, doi: 10.1080/00207543.2010.507607.

[8] Bautista, J., Alfaro, R., Batalla, C. (2015). Modeling and solving the mixed-model sequencing problem to improve productivity, *International Journal of Production Economics*, Vol. 161, 83-95, doi: 10.1016/j.ijpe.2014.11.018.

[9] Boysen, N., Fliedner, M., Scholl, A. (2009). Sequencing mixed-model assembly lines: Survey, classification and model critique, *European Journal of Operational Research*, Vol. 192, No. 2, 349-373, doi: 10.1016/j.ejor.2007.09.013.

[10] Wong, T.C., Chan, F.T.S., Chan, L.Y. (2009). A resource-constrained assembly job shop scheduling problem with Lot Streaming technique, *Computers & Industrial Engineering*, Vol. 57, No. 3, 983-995, doi: 10.1016/j.cie.2009.04.002.

[11] McMullen, P.R. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives, *Artificial Intelligence in Engineering*, Vol. 15, No. 3, 309-317, doi: 10.1016/S0954-1810(01)00004-8.

[12] Ding, F.-Y., He, J. (2008). A heuristic procedure for the automobile assembly-line sequencing problem considering multiple product options, *International Journal of Production Research*, Vol. 46, No. 20, 5827-5847, doi: 10.1080/ 00207540701381291.

[13] Franz, C., Hällgren, E.C., Koberstein A. (2014). Resequencing orders on mixed-model assembly lines: Heuristic approaches to minimise the number of overload situations, *International Journal of Production Research*, Vol. 52, No. 19, 5823-5840, doi: 10.1080/00207543.2014.918293.

[14] Rahimi-Vahed, A.R., Rabbani, M., Tavakkoli-Moghaddam, R., Torabi, S.A., Jolai, F. (2007). A multi-objective scatter search for a mixed-model assembly line sequencing problem, *Advanced Engineering Informatics*, Vol. 21, No. 1, 85-99, doi: 10.1016/j.aei.2006.09.007.

[15] Laili, Y., Tao, F., Zhang, L., Sarker, B.R. (2012). A study of optimal allocation of computing resources in cloud manufacturing systems, *The International Journal of Advanced Manufacturing Technology,* Vol. 63, No. 5, 671-690, doi: 10.1007/s00170-012-3939-0.

[16] Manavizadeh, N., Tavakoli, L., Rabbani, M., Jolai, F. (2013). A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority, *Journal of Manufacturing Systems*, Vol. 32 No. 1, 124-137, doi: 10.1016/j.jmsy.2012.09.001.

[17] Cano-Belmán, J., Ríos-Mercado, R.Z., Bautista, J. (2010). A scatter search based hyper-heuristic for sequencing a mixed-model assembly line, *Journal of Heuristics*, Vol. 16, No. 6, 749-770, doi: 10.1007/s10732-009-9118-2.

[18] Gujjula, R., Werk, S., Günther, H.-O. (2011). A heuristic based on Vogel's approximation method for sequencing mixed-model assembly lines, *International Journal of Production Research*, Vol. 49, No. 21, 6451-6468, doi: 10.1080/00207543.2010.527384.

[19] Cortez, P.M.C., Costa, A.M. (2015). Sequencing mixed-model assembly lines operating with a heterogeneous workforce, *International Journal of Production Research*, Vol. 53, No. 11, 3419-3432, doi: 10.1080/00207543.2014.987881.

[20] Akgündüz, O.S., Tunalı, S. (2010). An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem, *International Journal of Production Research*, Vol. 48, No. 17, 5157-5179, doi: 10.1080/00207540903117857.

[21] Shao, X., Wang, B., Rao, Y., Gao, L., Xu, C. (2010). Metaheuristic approaches to sequencing mixed-model fabrication/assembly systems with two objectives, *The International Journal of Advanced Manufacturing Technology*, Vol. 48, No. 9, 1159-1171, doi: 10.1007/s00170-009-2337-8.

[22] Wang, Y., Liu, J.H. (2010). Chaotic particle swarm optimization for assembly sequence planning, *Robotics and Computer-Integrated Manufacturing*, Vol. 26, No. 2, 212-222, doi: 10.1016/j.rcim.2009.05.003.

[23] Zhu, Q., Zhang, J. (2011). Ant colony optimisation with elitist ant for sequencing problem in a mixed model assembly line, *International Journal of Production Research*, Vol. 49, No. 15, 4605-4626, doi: 10.1080/00207543.2010.493534.

[24] Bautista, J., Cano, A., Alfaro, R. (2012). Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints. An application in Nissan's Barcelona Plant, *Expert Systems with Applications*, Vol. 39, No. 12, 11001-11010, doi: 10.1016/j.eswa.2012.03.024.

[25] Hamzadayi, A., Yildiz, G. (2013). A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines, *Computers & Industrial Engineering*, Vol. 66, No. 4, 1070-1084, doi: 10.1016/j.cie.2013.08.008.

[26] Pal, R., Bhattacharya, S., Caglar, M.U. (2012). Robust approaches for genetic regulatory network modeling and intervention: a review of recent advances, *IEEE Signal Processing Magazine*, Vol. 29, No. 1, 66-76, doi: 10.1109/MSP.2011.942818.

[27] Balasubramaniam, P., Banu, L.J. (2013). Robust state estimation for discrete-time genetic regulatory network with random delays, *Neurocomputing*, Vol. 122, 349-369, doi: 10.1016/j.neucom.2013.06.015.

[28] Sun, Y., Feng, G., Cao, J. (2009). Stochastic stability of Markovian switching genetic regulatory networks, *Physics Letters A,* Vol. 373, No. 18-19, 1646-1652, doi: 10.1016/j.physleta.2009.03.017.

[29] Dilão, R., Muraro, D. (2010). Calibration and validation of a genetic regulatory network model describing the production of the protein Hunchback in Drosophila early development, *Comptes Rendus – Biologies*, Vol. 333, No. 11-12, 779-788, doi: 10.1016/j.crvi.2010.09.003.

[30] Dörmer, J., Günther, H.-O., Gujjula, R. (2015). Master production scheduling and sequencing at mixed-model assembly lines in the automotive industry, *Flexible Services and Manufacturing Journal*, Vol. 27, No. 1, 1-29, doi: 10.1007/s10696-013-9173-8.

[31] Bautista, J., Cano, J. (2008). Minimizing work overload in mixed-model assembly lines, *International Journal of Production Economics*, Vol. 112, No. 1, 177-191, doi: 10.1016/j.ijpe.2006.08.019.

[32] Bautista, J., Cano, A. (2011). Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules, *European Journal of Operational Research*, Vol. 210, No. 3, 495-513, doi: 10.1016/j.ejor.2010.10.022.

[33] Tavakkoli-Moghaddam, R., Rahimi-Vahed, A.R. (2006). Multi-criteria sequencing problem for a mixed-model assembly line in a JIT production system, *Applied Mathematics and Computation*, Vol. 181, No. 2, 1471-1481, doi: 10.1016/j.amc.2006.02.033.

[34] Wong, C.S., Chan, F.T.S., Chung, S.H. (2012). A genetic algorithm approach for production scheduling with mould maintenance consideration, *International Journal of Production Research*, Vol. 50, No. 20, 5683-5697, doi: 10.1080/00207543.2011.613868.

[35] Shao, X., Li, X., Gao, L., Zhang, C. (2009). Integration of process planning and scheduling – A modified genetic algorithm-based approach, *Computers & Operations Research*, Vol. 36, No. 6, 2082-2096, doi: 10.1016/j.cor.2008.07.006.