

# Novel Approaches for Energy-Efficient Flexible Job-Shop Scheduling Problems

Nikolaos Rakovitis<sup>a</sup>, Dan Li<sup>a</sup>, Nan Zhang<sup>a</sup>, Jie Li<sup>a,\*</sup>, Liping Zhang<sup>b</sup>, Xin Xiao<sup>c</sup>

<sup>a</sup>Centre for Process Integration, Department of Chemical Engineering and Analytical Science, The University of Manchester, M13 9PL, United Kingdom

<sup>b</sup>Department of Industrial Engineering, Wuhan University of Science and Technology, Wuhan, Hubei, 430081 P. R. China

<sup>c</sup>Institute of Process Engineering, Chinese Academy of Sciences, P. R. China 100190  
 jie.li-2@manchester.ac.uk

In this work, two novel mixed-integer linear programming models for energy efficient scheduling of flexible job-shops with simultaneous consideration of machine switching off-on strategy are developed. While the first model is based on the unit-specific event-based approach, the other one uses the sequence-based approach. The computational results demonstrate that the proposed models, especially the unit-specific event-based model, are more robust and efficient than the existing models. To solve industrial-scale problems efficiently, a hybrid algorithm is developed through the combination of the existing eGEP algorithm and mathematical programming approach. The hybrid algorithm leads to up to 15 % more energy savings in comparison to the eGEP algorithm.

## 1. Introduction

Flexible job-shop scheduling problems have received significant attention due to their wide relation with many process industries (Chaudhry and Khan, 2015). Many approaches have been developed for such optimal scheduling to minimize operating cost or makespan without consideration of energy consumptions (Chaudhry and Khan, 2015). As reported, up to 65 % of total energy is consumed when machines are idle (Nguyen et al. 2019). The optimal schedule generated can lead to unnecessarily high energy consumption due to large machine idle periods and inefficient use of machine switching off-on strategy (Zhang et al., 2017).

Several approaches have been proposed for energy-efficient scheduling of flexible job-shop problems with simultaneous implementation of the machine switch off-on strategy (Meng et al., 2019). These approaches include the development of Mixed-Integer Linear Programming (MILP) models (Meng et al., 2019) and an efficient gene expression programming-based algorithm (eGEP) (Zhang et al., 2017). Although the MILP models can solve small-scale problems to optimality, they lead to intractable model size and require excessive computational time. They often fail to obtain a feasible solution for large-scale problems. The eGEP generates efficient dispatching rules for solving large-scale problems. However, it often generates solutions with higher energy consumption than the MILP models. No robust and efficient approach has been developed for the industrial-scale energy-efficient scheduling problem. In this work, two novel efficient MILP models are developed using the unit-specific event-based (Rakovitis et al., 2019) and sequence-based (Méndez and Cerdá, 2000) modelling approaches. These two models lead to smaller model size with tighter MILP relaxation and less computational time than the existing models. To solve large-scale problems efficiently, a hybrid algorithm through combination of the mathematical programming approach and the eGEP is proposed. It is shown that the hybrid algorithm always generates better solutions with up to 15 % additional energy savings than the eGEP. The hybrid algorithm is also superior to the proposed unit-specific event-based model for large-scale problems.

## 2. Problem description

Figure 1 illustrates a typical flexible job-shop facility. In total  $K$  ( $k = 1, 2, \dots, K$ ) jobs with  $L$  ( $l = 1, 2, \dots, L$ ) operations should be processed in  $J$  ( $j = 1, 2, \dots, J$ ) machines. Each job  $k$  contains  $L_k$  operations. An operation  $l$  in a job  $k$  can only start after all preceding operations in the same job finish. Each operation must be processed

exactly once in the horizon. A machine can be in operation, standby, or switch off-on mode. The energy consumption curve in a machine is depicted in Figure 2. Energy required is classified into direct energy, indirect energy, standby energy and switch off-on energy. While direct energy is consumed when a machine processes an operation, indirect energy is consumed for an operation in the facility but not directly from a machine. Standby energy is required when a machine remains on standby. Switch off-on energy is consumed when a machine is switched off and on. Given the necessary data including jobs, available machines, processing times, unit cutting power, unit unload power, switch off-on energy consumption and scheduling horizon, the scheduling problem is to determine optimal allocation, sequence and timings of operations, as well as optimal mode of each machine at a time. The objective is to minimize total energy consumption. All machines are switched off in the beginning.

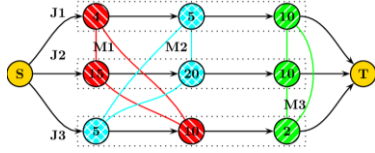


Figure 1: A typical flexible job-shop facility

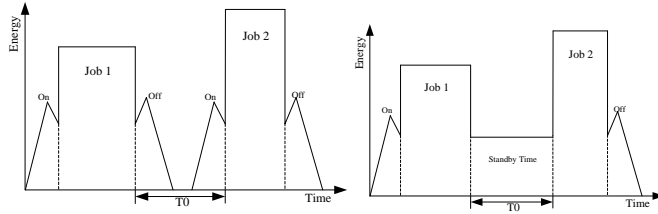


Figure 2: The energy consumption curve for a machine

### 3. Unit-specific event-based formulation (M1)

The process is represented by the state-task network (Kondili et al., 1993) where task  $i$  denotes an operation and state  $s$  denotes the relation between two tasks. A binary variable  $w_{i,j,n,n'}$  is used to denote if task  $i$  is processed on  $j$  from  $n$  to  $n'$ . A parameter  $\Delta n$  is the number of event points that a task is allowed to span over.

#### 3.1. Allocation constraints

At a time, machine  $j$  can process at most one task and all tasks must be processed exactly once in the horizon. A consuming task  $i'$  can start at event point  $n$  only if its related production task  $i$  ends at  $n$  or a previous  $n' < n$ .

$$\sum_{j \in \mathbf{I}_j} \sum_{n - \Delta n \leq n' \leq n} \sum_{n' \leq n'' \leq n + \Delta n} w_{i,j,n'',n'} = 1 \quad \forall j, n \quad (1)$$

$$\sum_{j \in \mathbf{I}_i} \sum_n \sum_{n \leq n' \leq n + \Delta n} w_{i,j,n,n'} = 1 \quad \forall i \quad (2)$$

$$\sum_{j \in \mathbf{I}_j} \sum_{n' \leq n \leq n' - \Delta n \leq n'' \leq n'} w_{i,j,n'',n'} \geq \sum_{n \leq n' \leq n + \Delta n} w_{i',j',n,n'} \quad \forall s \in S^{IN}, i \in \mathbf{I}_s^P, j', i' \in (\mathbf{I}_{j'} \cup \mathbf{I}_s^C), n \quad (3)$$

where  $\mathbf{I}_j$  contains tasks processed in  $j$ .  $\mathbf{I}_s^P$  contains tasks that “produce”  $s$ .  $\mathbf{I}_s^C$  includes tasks that “consume”  $s$ . The number of tasks processed on machine  $j$  must be within the minimum ( $N_j^{\min}$ ) and maximum ( $N_j^{\max}$ ) limits.

$$N_j^{\min} \leq \sum_{i \in \mathbf{I}_j} \sum_n \sum_{n \leq n' \leq n + \Delta n} w_{i,j,n,n'} \leq N_j^{\max} \quad \forall j \quad (4a,b)$$

#### 3.2. Duration constraints

The end time of unit  $j$  at  $n$  ( $T_{j,n}^f$ ) must equal the start time ( $T_{j,n}^s$ ) plus the duration ( $\alpha_{i,j}$ ) of the task processed.

$$T_{j,n}^f = T_{j,n}^s + \sum_{i \in \mathbf{I}_j} \sum_{n \leq n' \leq n + \Delta n} \alpha_{i,j} \cdot w_{i,j,n,n'} \quad \forall j, n \quad (5)$$

### 3.3 Standby energy constraints

A binary variable  $x_{j,n}$  denotes if  $j$  is on standby at  $n$ . The standby energy consumption ( $ES_{j,n}$ ) equals the idle time multiplying the unload power ( $PU_j$ ), which should not exceed the switch off-on energy consumption ( $EO_j$ ).

$$ES_{j,n} \leq EO_j \cdot x_{j,n} \quad \forall j, n \quad (6)$$

$$ES_{j,n} \leq (T_{j,n}^s - T_{j,n-1}^f) \cdot PU_j \quad \forall j, n > 1 \quad (7)$$

$$ES_{j,n} \geq (T_{j,n}^s - T_{j,n-1}^f) \cdot PU_j - \max_i(\alpha_{i,j}) \cdot PU_j \cdot (1 - x_{j,n}) \quad \forall j, n > 1 \quad (8)$$

### 3.4. Sequencing constraints

A machine  $j$  at event point  $(n + 1)$  must start after it finishes at  $n$ . A machine  $j$  at  $n$ , processing a “production” (“consumption”) task of state  $s$  must finish (start) before (after) the time that state  $s$  is available at  $n$  ( $T_{s,n}$ ).

$$T_{j,n+1}^s \geq T_{j,n}^f \quad \forall j, n < N \quad (9)$$

$$T_{s,n} \geq T_{j,n}^f - M \cdot (1 - \sum_{i \in (\mathbf{I}_j \cap \mathbf{I}_s^p)} \sum_{n-\Delta n \leq n' \leq n} w_{i,j,n',n}) \quad \forall s \in \mathbf{S}^{IN}, j, \sum_{i \in \mathbf{I}_j} \rho_{s,i} > 0, n \quad (10)$$

$$T_{s,n} \leq T_{j,n}^s + M \cdot (1 - \sum_{i \in (\mathbf{I}_j \cap \mathbf{I}_s^c)} \sum_{n \leq n' \leq n+\Delta n} w_{i,j,n',n}) \quad \forall s \in \mathbf{S}^{IN}, j, \sum_{j' \in (\mathbf{I}_{j'} \cap \mathbf{I}_s^c)} \sum_{i \in \mathbf{I}^R} \rho_{s,i'} > 0, n \quad (11)$$

The time that state  $s$  is available at event point  $n$  is before the time that is available at event point  $(n + 1)$ .

$$T_{s,n} \leq T_{s,n+1} \quad \forall s \in \mathbf{S}^{IN}, n \quad (12)$$

### 3.5. Makespan calculation

Makespan (denoted as  $MS$ ) is the earliest time, that all tasks have already been processed.

$$T_{j,n}^f \leq MS \quad \forall j, n = N \quad (13)$$

$$T_{s,n} \leq MS \quad \forall s \in \mathbf{S}^{IN}, n = N \quad (14)$$

### 3.6. Objective

The objective is to minimize the total energy consumption. The first and second term calculates direct energy, indirect energy, while the third term calculates standby energy and the switch off/on energy consumptions.

$$TEC = \sum_j \sum_{i \in \mathbf{I}_j} \sum_n \sum_{n \leq n' \leq n+\Delta n} w_{i,j,n',n} \cdot \alpha_{i,j} \cdot PC_{i,j} + \beta \cdot MS + \sum_{j,n>1} [ES_{j,n} + EO_j \cdot (1 - x_{j,n})] \quad (15)$$

## 4. Sequence-based mathematical formulation (M2)

A binary variable  $x_{k,l,k',l',j}$  denotes if an operation  $l$  in job  $k$  immediately precedes  $l'$  in job  $k'$  on  $j$ . Binary variables  $w_{k,l,j}$  denotes if  $l$  in job  $k$  is processed in  $j$  and  $XF_{k,l,j}$  denote if  $l$  in job  $k$  is first processed in  $j$ .

### 4.1. Allocation constraints

An operation is processed first or immediately preceded in  $j$ .  $\mathbf{KLJ}_{k,l,j}$  contains operations of job  $k$  processed in  $j$ .

$$\sum_{k' \neq l' \in \mathbf{KLJ}_{k',j}} \sum_{(k \neq k' \vee (k=k' \wedge l \neq l'))} x_{k',l',k,l,j} + XF_{k,l,j} = w_{k,l,j} \quad \forall j, k, l \in \mathbf{KLJ}_{k,l,j} \quad (16)$$

Constraint (17) ensures that an operation  $l$  that is not processed in  $j$  should not precede any operation in  $j$ . At most one operation can be processed first in  $j$  indicated in constraint (18). Constraint (19) ensures that an operation is processed exactly once. Constraints (20) impose min/max limits on the number of operations in  $j$ .

$$\sum_{k'} \sum_{l' \in \mathbf{KL}_{k',j}} \sum_{(k \neq k' \vee (k = k' \wedge l \neq l'))} x_{k,l,k',l',j} \leq w_{k,l,j} \quad \forall j, k, l \in \mathbf{KLJ}_{k,l,j} \quad (17)$$

$$\sum_k \sum_{l \in \mathbf{KLJ}_{k,j}} XF_{k,l,j} \leq 1 \quad \forall j \quad (18)$$

$$\sum_{j \in \mathbf{J}_{k,l}} w_{k,l,j} = 1 \quad \forall k, l \in \mathbf{L}_k \quad (19)$$

$$N_j^{\min} \leq \sum_k \sum_{l \in \mathbf{KLJ}_{k,j}} w_{k,l,j} \leq N_j^{\max} \quad \forall j \quad (20a,b)$$

A machine that does not process is in either the standby mode ( $y_{k,l,j} = 1$ ) or the switch on-off mode ( $z_{k,l,j} = 1$ ).

$$y_{k,l,j} + z_{k,l,j} = w_{k,l,j} \quad \forall j, k, l \in \mathbf{KLJ}_{k,l,j} \quad (21)$$

#### 4.2. Sequencing constraints

An operation  $l'$  ( $l' = l + 1$ ) should start after operation  $l$  in the same job finishes (constraint 22). Constraints 23-24 indicate if operation  $l$  immediately precedes  $l'$  in  $j$ , the start time of  $l'$  should equal the finish time of  $l$  plus the idle time ( $CT_{k,l,j}$ ).

$$T_{k,l'} \geq T_{k,l} + \sum_{j \in \mathbf{J}_{k,l}} (\alpha_{k,l,j} \cdot w_{k,l,j}) \quad \forall k, l, l' \in \mathbf{L}_k, l' = l + 1 \quad (22)$$

$$T_{k',l'} \geq T_{k,l} + \sum_{j \in \mathbf{J}_{k,l}} (\alpha_{k,l,j} \cdot w_{k,l,j} + CT_{k,l,j}) - H \left( 1 - \sum_{j \in (\mathbf{J}_{k,l} \cap \mathbf{J}_{k',l'})} x_{k,l,k',l',j} \right) \quad \forall (k, k', l, l') \in \mathbf{KL} \quad (23)$$

$$T_{k',l'} \leq T_{k,l} + \sum_{j \in \mathbf{J}_{k,l}} (\alpha_{k,l,j} \cdot w_{k,l,j} + CT_{k,l,j}) + H \left( 1 - \sum_{j \in (\mathbf{J}_{k,l} \cap \mathbf{J}_{k',l'})} x_{k,l,k',l',j} \right) + H \left( 1 - \sum_{j \in \mathbf{J}_{k,l}} y_{k,l,j} \right) \quad \forall (k, k', l, l') \in \mathbf{KL} \quad (24)$$

where set  $\mathbf{KL}$  is defined as:  $\mathbf{KL} = \{(k, l, k', l') | l \in \mathbf{L}_k, l' \in \mathbf{L}_{k'}, k' \neq k \text{ or } (k = k' \text{ and } l \neq l')\}$ .

#### 4.3. Standby energy calculation

$$ES_{k,l} = \sum_{j \in \mathbf{J}_{k,l}} CT_{k,l,j} \cdot PU_j \quad \forall k, l \in \mathbf{L}_k \quad (25)$$

$$CT_{k,l,j} \leq \min(H, \frac{EO_j}{PU_j}) \cdot y_{k,l,j} \quad \forall j, k, l \in \mathbf{KLJ}_{k,l,j} \quad (26)$$

#### 4.4. Tightening constraints

$$T_{k,l} + \sum_{j \in \mathbf{J}_{k,l}} (w_{k,l,j} \cdot \alpha_{k,l,j} + CT_{k,l,j}) \leq MS \quad \forall k, l \in \mathbf{L}_k, l = L \quad (27)$$

$$\sum_k \sum_{l \in \mathbf{KLJ}_{k,j}} (w_{k,l,j} \cdot \alpha_{k,l,j} + CT_{k,l,j}) \leq MS \quad \forall j \quad (28)$$

#### 4.5. Objective

$$z = \sum_j \sum_k \sum_{l \in \mathbf{KLJ}_{k,j}} (w_{k,l,j} \cdot \alpha_{k,l,j} \cdot PC_{k,l,j}) + \beta \cdot MS + \sum_k \sum_{l \in \mathbf{L}_k} [ES_{k,l}] + \sum_j \sum_k \sum_{l \in \mathbf{KLJ}_{k,j}} [EO_j \cdot z_{k,l,j}] \quad (29)$$

#### 4.6. Bounds

An operation  $l$  in a job  $i$  should always start after the minimum time required for all previous operations.

$$T_{k,l} \geq \sum_{l' < l} \left\{ \min_j (\alpha_{k,l',j} + \beta_{k,l',j} \cdot B_{k,l',j}^{\min}) \right\} \quad \forall k, l, l' \in \mathbf{L}_k \quad (30)$$

## 5. Hybrid algorithm

The eGEP from Zhang et al. (2017) generates dispatching rules that are used to obtain schedules. However, the schedules may not be energy efficient. The main reason is that the dispatching rules can only effectively determine the operation allocation and sequence. A heuristic “assign as early as possible” is used to determine the timings of operations on machines, which can result in high standby and switch off-on energy. A better solution can be obtained using an efficient mathematical model to determine the optimal timings. Based on this, a hybrid algorithm through combination of the eGEP algorithm and the proposed sequence-based model is developed in which the eGEP algorithm first generates efficient dispatching rules to determine the allocation and sequence of operations. The proposed sequence-based model M2 then determines the best operation timings and best trade-off between indirect energy consumption and switching off-on energy consumption.

Table 1: Computational results for Examples 1-10

Example	Model	Event points	CPU time (s)	RMILP (h)	MILP (h)	Discrete Variables	Continuous Variables	Constraints	Gap (%)
Ex1	ZTWW	3	0.19	7.33	63.03	30	45	218	-
	MZSR	3 <sup>a</sup>	0.06	63.03	63.03	21	23	68	-
	M1	3	0.02	63.03	63.03	19	28	68	-
	M2	-	0.06	63.03	63.03	16	30	57	-
Ex2	ZTWW	3	0.19	24.58	220.74	63	84	596	-
	MZSR	4 <sup>a</sup>	0.05	214.74	220.74	37	35	114	-
	M1	3	0.02	220.74	220.74	30	44	123	-
	M2	-	0.05	220.74	220.74	30	46	94	-
Ex3	ZTWW	3	0.20	20.00	166.23	42	57	292	-
	MZSR	4 <sup>a</sup>	0.05	159.23	166.23	46	36	133	-
	M1	3	0.02	166.23	166.23	28	31	76	-
	M2	-	0.11	166.23	166.23	40	46	103	-
Ex4	ZTWW	5	2.9	19.79	219.46	150	183	1,946	-
	MZSR	7 <sup>a</sup>	0.19	212.97	219.46	90	56	244	-
	M1	5	0.11	218.97	219.46	77	80	274	-
	M2	-	0.14	218.97	219.46	91	72	195	-
Ex5	ZTWW	6	2.8	25.70	274.94	180	219	2,660	-
	MZSR	6 <sup>a</sup>	0.30	260.94	274.94	84	56	232	-
	M1	6	0.05	274.94	274.94	93	95	315	-
	M2	-	0.05	274.94	274.94	82	72	183	-
Ex6	MZSR	15 <sup>a</sup>	3,600	1,455.95	1,715.22	853	277	2,032	22.6
	M1	11	3,600	1,542.15	1,830.43	710	612	2,405	5.5
	M2	-	3,600	1,710.62	1,792.04	943	342	1,739	4.5
Ex7	M1	20	3,600	2,606.72	2,637.50	1,815	1,507	6,016	0.3
	M2	-	3,600	2,606.72	2,736.30	1,816	496	3,366	4.7
Ex8	M1	21	3,600	3,002.11	3,035.98	2,494	2,002	8,207	0.4
Ex9	M1	12	3,600	2,512.49	3,648.90	1,466	1,452	5,585	31.1
	M2	-	3,600	2,590.43	3,150.59	1,555	654	3,129	26.2
Ex10	M1	17	3,600	3,885.07	6,704.23	2,934	2,652	11,439	42.1
Ex11	M1	21	3,600	5,598.36	9,603.38	4,946	4,422	17,809	41.5

<sup>a</sup> Maximum number of positions of all machines.

## 6. Computational results

15 examples from Zhang et al. (2017) are solved to illustrate the capability of the proposed models and hybrid algorithm. While Examples 1-5 are small-scale, Examples 6-11 and 12-15 are medium-scale and large-scale. All examples are solved using CPLEX 12/GAMS 24.6.1. on a desktop computer with Intel® Core™ i5-2500 3.3 GHz and 8 GB RAM running Windows 7. The maximum computational time is one hour. The results are given in Table 1. Model M1 is able to generate optimal solutions for Examples 1-5 and best feasible solutions for Examples 6-11. It fails for Examples 12-15. Model M2 can also generate optimal solutions for Examples 1-5 and feasible solutions for Examples 6-7 and 9 only.

The performance of models M1 and M2 is also compared with the existing models (Zhang et al., 2017; Meng et al., 2019) denoted as ZTWW and MZSR, as shown in Table 1. From Table 1, both ZTWW and MZSR can generate optimal solutions for Examples 1-5. However, ZTWW cannot generate a feasible solution for Examples

6-15, whilst MZSR is only able to generate a feasible solution for Example 6. Models M1 and M2 lead to tighter LP relaxation and smaller model size for most examples than ZTWW and MZSR. As a result, M1 and M2 require less computational time to obtain the same or better solutions than ZTWW and MZSR. The Model M1 is more robust than M2 since it can generate solutions for more examples.

*Table 2: Comparative results for Examples 6-15 using different approaches*

	M1	eGEP	Hybrid Algorithm	
Examples	TEC (kW)	TEC (kW)	TEC (kW)	Examples
Ex6	1,830.43	1,975.75	1,943.83	Ex6
Ex7	2,637.50	3,046.21	2,890.21	Ex7
Ex8	3,035.98	3,700.86	3,468.03	Ex8
Ex9	3,648.90	4,082.95	3,715.50	Ex9
Ex10	6,704.23	6,640.15	5,671.46	Ex10
Ex11	9,603.38	7,173.32	6,662.99	Ex11
Ex12	-	10,108.14	9,098.46	Ex12
Ex13	-	10,939.20	9,726.36	Ex13
Ex14	-	10,339.46	9,112.79	Ex14
Ex15	-	10,751.25	9,459.67	Ex15

The hybrid algorithm is also used to solve Examples 6-15. The comparative results of the hybrid algorithm, model M1 and the eGEP algorithm are provided in Table 2. It seems that model M1 can generate better feasible solutions for medium-scale examples 6-9 compared to the eGEP and hybrid algorithms. However, when the problem size increases, model M1 generates worse solutions or fails in comparison to the eGEP and hybrid algorithms (see Examples 10-15). The hybrid algorithm always generates better solutions than the eGEP algorithm due to the reduction of machine idle time. A maximum improvement by 15 % can be achieved.

## 7. Conclusions

In this work, two novel mathematical models for energy-efficient scheduling of flexible job-shops were developed. A hybrid algorithm combining the eGEP with the mathematical programming model was proposed to solve large-scale problems efficiently. The computational results demonstrate that the proposed unit-specific event-based model is the most robust and efficient compared to the existing models. The hybrid algorithm always generated better solutions with up to 15 % more energy savings in comparison to the eGEP algorithm. It is also shown that the hybrid algorithm is superior to the proposed unit-specific event-based model for industrial-scale problems. In the future, the hybrid algorithm will be refined further to improve the solution quality.

## Acknowledgements

NR would like to acknowledge financial support from the postgraduate award by The University of Manchester. LZ appreciates financial support from the National Natural Science Foundation of China (51875420).

## References

- Chaudhry I.A., Khan A.A., 2015. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operations Research*, 23, 551-591.
- Kondili E., Pantelides C. C., Sargent R. W. H., 1993 A general algorithm for short-term scheduling of batch operations-I MILP formulation. *Computers and Chemical Engineering*, 17(2), 211-227
- Méndez C.A., Cerdá J., 2000. Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities. *Computers and Chemical Engineering*, 24(2-7), 369-376
- Meng L., Zhang C., Shao X., Ren Y., 2019. MILP models for energy-aware flexible job shop scheduling problem. *Journal of Cleaner Production*, 210, 710-723.
- Rakovitis N., Zhang N., Li J., Zhang L., 2019. A new approach for scheduling of multipurpose batch processes with unlimited intermediate storage policy. *Frontiers of Chemical Science and Engineering*, 13, 784-802
- Nguyen T.A.K., Tran Le N.H., Henri P., Museau M., 2019. Evaluating energy efficiency of production machine. *Applied Mechanics & Materials*, 889, 580-587.
- Zhang L., Tang Q., Wu Z., Wang F., 2017. Mathematical modelling and evolutionary generation of rule sets for energy-efficient flexible job shops. *Energy*, 138, 210-227.