Supplemental Material

Appendix A. Basic notations

Table S.1 Mathematical notations used in the learning and deteriorating functions

Notation	Description
p	The fixed processing time of all jobs
p_j	The normal processing time of job j
p_r	The normal processing time of job sequenced in position r
p_b	The normal processing time of batch b
p_{ij}	The normal processing time of job j on machine i
p_{ir}	The normal processing time of job on machine i sequenced in position r on machine i
p_{bj}	The normal processing time of job j in batch b
p_{gj}	The normal processing time of job j in group g
p_{gjr}	The normal processing time of job j sequenced in position r in group g
$p_{[j]}$	The actual processing time of job j
$p_{[r]}$	The actual processing time of job sequenced in position r
$p_{[b]}$	The actual processing time of batch b
$p_{[ij]}$	The actual processing time of job j on machine i
$p_{[jr]}$	The actual processing time of job j sequenced in position r
$p_{[ir]}$	The actual processing time of job on machine i sequenced in position r
$p_{[br]}$	The actual processing time of batch b sequenced in the rth batch
$p_{[gj]}$	The actual processing time of job j in group g
$p_{[ijr]}$	The actual processing time of job j on machine i sequenced in position r
$p_{[bjr]}$	The actual processing time of job j sequenced in position r in batch b
$p_{[gjr]}$	The actual processing time of job j sequenced in position r in group g
$p_{[gjr_1r_2]}$	The actual processing time of job j in group g sequenced in the r_1th position in the r_2th group
a	The common learning indicator
a_i	The learning indicator of jobs on machine i
a_j	The learning indicator of job j
a_{ij}	The learning indicator of job j on machine i
a_b	The learning indicator of jobs in batch b
a_g	The learning indicator of jobs in group g
B_b	The learning indicator of batch b
G_g	The learning indicator of group g
α	The common deteriorating indicator
α_j	The deteriorating indicator of job j
α_g	The deteriorating indicator of jobs in group g
α_{ij}	The deteriorating indicator of job j on machine i
α_{gj}	The deteriorating indicator of job j in group g
t	The starting time of the job
t_0	The initial time that a set of jobs is available for processing
t_j	The starting time of job j
t_r	The starting time of job sequenced in position r
t_{ij}	The starting time of job j on machine i
M	The incompressibility factor
M_{ij}	The incompressibility factor of job j on machine i
u_j	The amount of resource allocated to job j
u_{ij}	The amount of resource allocated to job j on machine i
u_{gj}	The amount of resource allocated to job j in group g
κ_j	The positive compression rate of job j
κ_{ij}	The positive compression rate of job j on machine i

Notation	Description
κ_{gj}	The positive compression rate of job j in group g
σ	The positive constant
ρ	The truncation parameter

Table S.2 Mathematical notations used in the objective functions

Notation	Description
w_j, w_r	The job weight
Γ_j	The per time unit cost associated with the resource allocation
C_j, C_r, C_{gj}	The job completion time
T_j, T_r, T_{ij}	The job tardiness
d_j, D_r	The job due date
R_j, R_r	The job release date
W_j	The job waiting time
L_j	The job lateness
E_j	The job earliness
C_{max}	The makespan
C^i_{max}	The load of machine i which can be expressed as 'TML'
T_{max}	The maximum tardinesss
E_{max}	The maximum earliness
L_{max}	The maximum lateness
$\sum C_j, \sum C_{gj}$	The total completion time which can be expressed as 'TC'
$\sum C_j^2$	The sum of quadratic job completion time
$\sum C_j^\delta$	The sum of the δ power of job completion time
$\sum C^i_{max}$	The total machine load
$\sum w_j C_j$	The total weighted completion time
$\sum w_j(1-e^{-\gamma C_j})$	The discounted total weighted completion time, where $\gamma \in \{0,1\}$ is a discount factor
$\sum T_j$	The total tardiness
$\sum E_j$	The total earliness
$\sum F_j$	The total flow time
$\sum U_j$	The number of tardy jobs
$\sum W_j$	The total waiting times which can be expressed as 'TW'
$\sum \Gamma_j u_j$	The total time cost associated with the resource allocation

Appendix B. Abbreviations

Table S.3 The abbreviations

Abbreviations	Explanations
B&B	Branch-and-bound
DP	Dynamic programming
GAs	Genetic algorithms
SA	Simulated annealing
SPT	The shortest processing time first
ARB	Any busy schedule
WSPT	The weighted shortest processing time first
WDSPT	The weighted discounted shortest processing time first
EDD	The earliest due date
ERD	The earliest ready date
RS	Random search
TS	Tabu search
MODES	Multi-objective differential evolution algorithms
MO-SADE	Multi-objective simulated annealing differential evolution
PSO	Particle swarm optimization
CSA	Cloud theory based simulated annealing
BBNP	Bounds-based nested partition
GSA	Gravitational search algorithm
VNS	Variable neighborhood search
CS	Cuckoo search
GSA-TS	Hybrid gravitational search algorithm and tabu search
VNS-GSA	Hybrid variable neighborhood search and gravitational search algorithm
QDE	Quantum differential evolutionary
CS-SADE	Cuckoo search and self-adaptive differential evolution
TADC	The total deviation of completion times
SDR	The smallest deterioration rate first
FPTAS	Fully polynomial-time approximation schemes
MVO	Multi-verse optimizer
H-DP	Hybrid algorithm combining heuristic with dynamic programming
ABC-TS	Artificial bee colony and tabu search
ABC	Artificial bee colony
BA	Bat algorithm
DE	Differential evolutionary

Abbreviations	Explanations
WSDR	The weighted smallest deterioration rate first
SC-VNS	Society and civilization algorithm with variable neighborhood search
SC	Society and civilization
BRKGA-DE	Biased random-key genetic algorithm and differential evolutionary
BRKGA	Biased random-key genetic algorithm
VNS–ASHLO	Variable neighborhood search and adaptive simplified human learning optimization
ASHLO	Adaptive simplified human learning optimization
p-s-d	Past-sequence-dependent

Appendix C. Complexity of problems

Table S.4 complexity

Problem	complexity	paper
$1\|p_{[j]} = p_j - a_j min\{n_j, n_{0j}\}\|L_{max}$	NP-hard	Cheng and Wang (2000)
$1\|p_{[jr]} = p_j - a_j r\|C_{max}$	O(n log$n$)	Bachman and Janiak (2004)
$1\|p_{[jr]} = p_j - ar\|C_{max}$	O(n log$n$)	
$1\|R_j, p_{[jr]} = p_j - a_j r\|C_{max}$	NP-hard	
$1\|p_{[jr]} = p_j r^a\|C_{max}$	O(n log$n$)	
$1\|R_j, p_{[jr]} = p_j r^a\|C_{max}$	NP-hard	
$F\|p_{[ijr]} = p_{ij}(\mu - \nu r)\|\sum w_j C_j$	NP-complete	Sun et al. (2013)
$1\|p_{[jr]} = p_j r^a\|\sum C_j$	O(n log$n$)	Biskup (1999)
$1\|p_{[jr]} = p_j r^a\|\sum(\delta_1 E_j + \delta_2 T_j + \delta_3 C_j)$	O(n^3)	
$1\|p_{[jr]} = p_j r^a\|\delta_1 \sum C_j + \delta_2 T_j$	NP-hard	Eren and Güner (2007)
$P2\|p_{[jr]} = p_j r^a\|\sum C_j$	O(n^4)	Mosheiov (2001)
$Pm\|p_{[ijr]} = p_{ij} r^a\|L_{max}$	NP-hard	Xu, Yin, and Li (2010)
$F2\|p_{[ijr]} = p_{ij} r^a\|\delta_1 \sum C_j + \delta_2 C_{max}$	NP-hard	Eren and Güner (2008);
$Fm\|R_j, p_{[ijr]} = p_{ij} f(r)\|C_{max}, \sum C_j, \sum C_j^2$	NP-hard	Bai et al. (2018)
$Fm, h_{ik}\|R_j, p_{[ijr]} = p_{ij} r^{a_i}\|\sum_{j=1}^{n} \pi_j T_j + \sum_{i=1}^{m} \sum_{k=1}^{K} \theta_{ik}(FM_{ik} - EM_{ik})$	NP-hard	Vahedi Nouri, Fattahi, and Ramezanian (2013)
$Fm\|p_{[ijr]} = p_{ij} r^a\|C_{max}, TEC$	-	Xin et al. (2021)
$F2\|p_{[ijr]} = (\frac{p_{ij} r^a}{u_{ij}})^\sigma\|\sum(\delta_1 E_j + \delta_2 T_j + \delta_3 d) + \delta_4 \sum_{i=1}^{2} \sum_{j=1}^{n} \Gamma_{ij} u_{ij}$	O(n^3)	Gao et al. (2018)
$F2\|p_{[ijr]} = p_{ij} max\{r^a, \rho\}\|\sum C_j$	NP-complete	Li et al. (2011a)
$Pm\|p_{[jr]} = p_j(M + (1-M) r^a)\|C_{max}$	NP-hard	Okolowski and Gawiejnowicz (2010)
$1\|p_{[jr]} = p_j(1 + \sum_{l=1}^{r-1} p_{[l]})^a\|\sum C_j$	O($n$log$n$)	Kuo and Yang (2006)
$1\|p_{[jr]} = p_j(1 + \sum_{l=1}^{r-1} p_{[l]})^a\|C_{max}, \sum C_j, \sum C_j^2, \sum w_j C_j, L_{max}$	-	Wang (2008)
$Fm\|p_{[ijr]}^h = p_{ij}^h(1 + \sum_{l=1}^{r-1} p_{i[l]}^h)^a\|\sum T_j$	NP-hard	Lin et al. (2017)

Problem	complexity	paper
$Fm\|p_{[ijr]}=p_{ij}(1+\sum_{l=1}^{r-1}p_{i[l]})^a\|\sum C_j$	NP-hard	Wu et al. (2018)
$F2\|p_{[jr]}=p_j(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^a\|C_{max},\sum C_j$	O($n$log$n$)	Koulamas and Kyparisis (2007)
$1\|p_{[jr]}^X=p_j^X(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^a\|\sum w_j^X C_j^X:C_{max}^A\leq V$	NP-hard	Wu (2014)
$1\|p_{[jr]}=p_j(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^a\|\sum C_j$	NP-hard	Wu, Hsu, and Lai (2011)
$1\|p_{[jr]}=p_j(\mu a^{\sum_{l=1}^{r-1}p_{[l]}}+\nu)\|C_{max},\sum C_j^\theta,\sum w_jC_j,L_{max}$	-	Wang, Sun, and Sun (2010)
$1\|p_{jr}=p_j(\mu a^{-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}}+\nu)\|\sum L_j,\sum T_j,\sum w_jC_j,\sum w_j(1-e^{-\gamma C_j}),L_{max}$	-	Ma, Shao, and Wang (2014)
$Fm\|p_{[ijr]}=p_{ij}(\mu a^{\sum_{l=1}^{r-1}p_{i[l]}}+\nu)\|\sum[\mu(M_i)\tau_c(M_i)+k\mu(M_i)\tau_d(M_i)]$	NP-complete	Liu, Shi, and Shi (2018)
$1\|p_{[jr]}=p_j(1+\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}\|C_{max},\sum C_j,\sum w_jC_j$	P	Wu and Lee (2008)
$1\|p_{[jr]}=p_j(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}\|C_{max},\sum C_j,\sum w_jC_j,L_{max}$	P	Cheng, Wu, and Lee (2008)
$Fm\|p_{[ijr]}=p_j(1-\frac{\sum_{l=1}^{r-1}p_{i[l]}}{\sum_{l=1}^{n}p_{il}})^{a_1}r^{a_2}\|C_{max},\sum C_j$	P	
$1\|p_{jr}=p_j(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1}a_2^{r-1}\|\sum C_j,\sum w_jC_j$	P	Low and Lin (2011)
$1\|p_{[jr]}=p_j(\mu a_1^{\sum_{l=1}^{r-1}w_lp_{[l]}}+\nu)a_2^{r-1}\|C_{max},\sum C_j,\sum w_jC_j,\sum C_j^\theta,L_{max}$	P	Bai, Wang, and Wang (2012)
$Fm\|p_{[ijr]}=p_{ij}(\mu a_1^{\sum_{l=1}^{r-1}p_{i[l]}}+\nu)a_2^{r-1}\|L_{max}$	NP-complete	He (2016)
$1\|b,T_{no},p_{[bjr]}=p_{bj}r^{a_b}\|C_{max}$	O($n$log$n$)	Yang and Kuo (2009)
$1\|b,T_{part},p_{[br]}=p_b^{B_b}\|C_{max}$	O($n$log$n$+N^3)	
$1\|b,T_{total},p_{[bjr]}=p_{bj}(r+\sum_{l=1}^{b-1}n_l)^{a_b}\|C_{max}$	O($\overline{n}$log$\overline{n}$+N^3)	
$1\|s-batch,p_{[jr]}=(p_j-\tau_j)r^a\|C_{max}$	-	Pei et al. (2018)
$1\|s-batch,p_{[jr]}=p_jr^a\|E_{max},\sum U_j$	O($n$log$n$)	Pei et al. (2019a)
$P\|s-batch,p_{[jr]}=p_jr^a\|E_{max},\sum U_j$	-	
$Fm\|g,prmu,p_{[gjr_1r_2]}=p_{gj}r_1^{a_1}r_2^{a_2}\|C_{max}$	P	Qin, Zhang, and Bai (2016)
$Fm\|g,prmu,p_{[gjr_1r_2]}=p_{gj}r_1^{a_1}r_2^{a_2}\|\sum C_{gj},\sum w_gC_{gj},L_{max}$	NP-hard	

Problem	complexity	paper
$1\|g, p_{[gjr_1r_2]} = (\frac{p_{gj}r_1^{a_1}r_2^{a_2}}{u_{gj}})^{\sigma}\|\delta_1 C_{max} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$	$O(n^3)$	
$1\|g, p_{[gjr_1r_2]} = p_{gj}r_1^{a_1}r_2^{a_2} - \kappa_{gj}u_{gj}\|\delta_1 C_{max} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$	$O(n\log n)$	Zhu et al. (2011)
$1\|g, p_{[gjr_1r_2]} = (\frac{p_{gj}r_1^{a_1}r_2^{a_2}}{u_{gj}})^{\sigma}\|\delta_1 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}C_{gj} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$	$O(n^3)$	
$1\|g, p_{[gjr_1r_2]} = p_{gj}r_1^{a_1}r_2^{a_2} - \kappa_{gj}u_{gj}\|\delta_1 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}C_{gj} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$	$O(n^3)$	
$1\|g, prmu, p_{[gjr_1r_2]} = (\frac{p_{gj}r_1^{a_1}r_2^{a_2}}{u_{gj}})^{\sigma}, \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj} \leq U\|C_{max}$	NP-hard	Lu et al. (2017)
$1\|g, p_{[gjr_1r_2]} = p_{gj}r_1^{a_1 log_2(1-\tau)a}r_2^{a_2 log_2(1-\tau)a}\|\delta_1 C_{max} + \delta_2 k(\tau)$	$O(n^3)$	Huo, Ning, and Sun (2018)
$1\|g, p_{[gjr_1r_2]} = p_{gj}r_1^{a_1 log_2(1-\tau)a}r_2^{a_2 log_2(1-\tau)a}, n_g = \bar{n}\|\delta_1 C_{gj} + \delta_2 k(\tau)$	$O(n^3)$	
$1\|p-batch, p_{[b]} = \max_{J_j \in batch\, b}\{p_j - min\{at_b, \rho\}\}\|C_{max}$	P	
$Pm\|p-batch, p_{[b]} = \max_{J_j \in batch\, b}\{p_j - min\{at_b, \rho\}\}\|C_{max}$	NP-hard	Liu et al. (2020)
$1\|g, s-indep, p_{[gjr]} = p_{gj}(1+\sum_{l=1}^{r-1}p_{g[l]})^{a_g}\|C_{max}$	$O(n\log n)$	
$1\|g, s-dep, p_{[gjr]} = p_{gj}(1+\sum_{l=1}^{r-1}p_{g[l]})^{a_g}\|C_{max}$	$O(Q^3 + n\log n)$	Kuo (2012)
$1\|g, s-indep, p_{[gjr]} = p_{gj}(1+\sum_{l=1}^{r-1}p_{g[l]})^{a_g}\|\sum C_j$	P	
$1\|g, s-dep, p_{[gjr]} = p_{gj}(1+\sum_{l=1}^{r-1}p_{g[l]})^{a_g}\|\sum C_j$	P	
$1\|g, p_{[gjr]} = p_{gj}(1 - \frac{\sum_{g=1}^{r_2-1}s_{[g]} + \sum_{g=1}^{r_2-1}\sum_{j=1}^{n_g}p_{[g][j]}}{\sum_{g=1}^{Q}s_g + \sum_{g=1}^{Q}\sum_{j=1}^{n_g}p_{gj}})^{a}\|C_{max}$	P	Liu, Lee, and Wu (2008)
$1\|g, p_{[gjr_1]} = p_{gj}f_{1g}(\sum_{l=1}^{r_1-1}p_{g[l]})f_{2g}(r_1)\|C_{max}, \sum C_{gj}, \sum w_{gj}C_{gj}, \sum w_{gj}(1-e^{-\gamma C_{gj}})$	$O(n\log n)$	Yin et al. (2013)
$1\|g, p_{[gjr_1r_2]} = p_{gj}(1 - \frac{\sum_{l=1}^{r_1-1}p_{g[l]}}{\sum_{l=1}^{n_g}p_{gl}})^{a_1}a_2^{r_2-1}\|C_{max}, \sum C_{gj}$	$O(n\log n)$	Low and Lin (2012)
$1\|p_{[j]} = \alpha_j t\|C_{max}, \sum F_j, \sum w_jC_j, \sum L_j, L_{max}, T_{max}, \sum U_j$	P	Mosheiov (1994)
$1\|R_j = t_0, q_j, p_{[j]} = \alpha_j t\|V_{max}$, max $w_jV_j, \sum w_jV_j$ (V_j is delivery completion time of job j)	$O(n\log n)$	Zou (2014)
$1\|R_j, q_j, p_{[j]} = \alpha_j t\|V_{max}$	NP-hard	
$1\|p_{[j]} = \alpha_j t\|\sum w_jW_j^{\theta}$	P	
$1\|p_{[j]} = \alpha_j t\|\sum w_jW_j^{2}$	$O(n\log n)$	
$1\|weakchains, p_{[j]} = \alpha_j t\|\sum w_jW_j^{\theta}$	$O(n\log n)$	Wang and Wang (2015)
$1\|strongchains, p_{[j]} = \alpha_j t\|\sum w_jW_j^{\theta}$	P	
$1\|sp-digraph, p_{[j]} = \alpha_j t\|\sum w_jW_j^{\theta}$	$O(n^2)$	

Problem	complexity	paper
$Pm\|p_{[j]}=\alpha_j t_j\|C_{max},\sum L_j,\sum C_j$	NP-hard	Ji and Cheng (2009)
$Pm\|p_{[j]}=\alpha_j t\|C_{max}(S)+\sum_{\bar{S}} e_j$ $Pm\|p_{[j]}=\alpha_j t, r_j=t_0\|\sum w_j C_j+\sum_{\bar{S}} e_j$	NP-hard NP-hard	Li and Yuan (2010)
$Fm\|p_{[ij]}=\alpha_{ij} t\|C_{max}$ $Om\|p_{[ij]}=\alpha_{ij} t\|C_{max}$ $Jm\|p_{[ij]}=\alpha_{ij} t\|C_{max}$	NP-complete NP-complete NP-hard	Mosheiov (2002)
$F2\|p_{[ij]}=\alpha_i t, type-1chains\|C_{max}$ $F2\|p_{[ij]}=\alpha_i t, type-2chains\|C_{max}$	P NP-hard	Zhao and Tang (2012)
$F2\|p_{[ij]}=\alpha_{ij} t\|\sum C_j : C_{max}$	P	Cheng et al. (2014)
$1\|p_{[j]}=p_j+\alpha_j t\|L_{max}$	NP-complete	Bachman and Janiak (2000)
$1\|p_{[j]}=p_j+\alpha_j t_j\|\sum w_j C_j$	NP-hard	Bachman, Janiak, and Kovalyov (2002)
$1\|p_{[j]}=p_j+\alpha t_j, p_{[j]}=$ $\lambda p_j+\alpha(t_j-C_r-A)\|C_{max},\sum C_j$	$O(n^2 logn)$	Sun and Geng (2019)
$Rm\|p_{[ijr]}=p_{ij}+\alpha_i t_{ir}, nr, ma\|\sum C_j,\sum C^i_{max}$	O(n^{2m+2})	Hsu et al. (2013)
$Pm\|p_{[j]}=p_j+\alpha_j \eta_j\|C_{max}$	NP-hard	Woo and Kim (2018)
$F\|prmu, p_{[ij]}=p_{ij}+\alpha_{ij} t\|\sum max\{C_j-d_j,0\}$	NP-hard	Wang, Huang, and Wang (2019)
$Fm\|prmu, p_{[ij]}=p_{ij}(\mu+\nu t)\|\sum T_j$	NP-hard	Bank et al. (2012a)
$Fm\|prmu, p_{[ij]}=p_{ij}(\mu+\nu t)\|\sum T_j$	NP-hard	Bank et al. (2012b)
$F2\|p_{[ij]}=p_{ij}(\mu+\nu t)\|\sum C_j$	NP-hard	Ng et al. (2010)
$1\|p_{[j]}=p_j+\alpha_j t-\kappa u_j\|\delta_1 C_{max}+\delta_2 TC+$ $\delta_3 TADC+\delta_4\sum \Gamma_j u_j$ $1\|p_{[j]}=p_j+\alpha_j t-\kappa u_j\|\delta_1 C_{max}+\delta_2 TW+$ $\delta_3 TADW+\delta_4\sum \Gamma_j u_j$	O(n^3) -	Wei, Wang, and Ji (2012)
$1\|p_{[j]}=(\frac{\kappa_j}{u_j})^\sigma+\alpha_j t_j\|C_{max}+\theta\sum \Gamma_j u_j$ $1\|p_{[j]}=(\frac{\kappa_j}{u_j})^\sigma+\alpha_j t_j, C_{max}\le \hat{C}_j\|\sum \Gamma_j u_j$ $1\|p_{[j]}=(\frac{\kappa_j}{u_j})^\sigma+\alpha_j t_j, \sum C_j\le TC\|\sum \Gamma_j u_j$	O($n$log$n$) O($n$log$n$) O($n$log$n$)	Li and Wang (2018)
$1\|p_{[jr]}=$ $p_j(1+\sum_{l=1}^{r-1} logp_{[l]})^a\|C_{max},\sum C_j,\sum C_j^2,\sum T_j, L_{max}$	P	Cheng, Lee, and Wu (2011)
$Rm\|p_{[ijr]}=p_{ij}f_{ij}(r), ma\|TC$	O(n^{m+k+2})(k is the upper bound of the total maintenance frequencies)	Yang (2013)

Problem	complexity	paper
$Pm\|p_{[ijr]}=p_{ij}f_{ij}(r),ma\|TC$	O(n^{m+k}log$n$)	
$1\|p_{[gjr]}=p_{gj}(1+\alpha)^{r-1},DRMs\|C_{max}$	O(n^4)	Zhang et al. (2018)
$1\|p_{[gjr]}=p_{gj}(1+\alpha)^{r-1},DRM\|C_{max}$	O(n^2logn)	
$Pm\|p_{[gijr]}=p_{gij}(1+\alpha)^{r-1},DRMs\|C_{max}$	NP-hard	
$1\|p_{[jr]}=$ $p_jf(r)-\kappa_ju_j\|\sum_{v=1}^{V}\sum_{j\in V_v}(\delta_1E_j+\delta_2T_j+\delta_3D_v+\Gamma_ju_j)$	$O(n^3)$	Yang, Lee, and Guo (2013)
$1\|p_{[jr]}=$ $(\frac{p_jf(r)}{u_j})^{\sigma}\|\sum_{v=1}^{V}\sum_{j\in V_v}(\delta_1E_j+\delta_2T_j+\delta_3D_v+\Gamma_ju_j)$	$O(nlogn)$	
$1\|p-batch,p_{[j]}=\alpha_jt,R_j,\alpha=\infty\|C_{max}$	O($n$log$n$)	Li et al. (2011b)
$1\|p-batch,p_{[j]}=\alpha_jt,R_j,\alpha<n\|C_{max}$	NP-hard	
$1\|p-batch,p_{[j]}=\alpha_jt\|C_{max}+W$	NP-hard	Kong et al. (2020b)
$M\to C\|s-batch,p_{[j]}=\alpha_jt,buffer\|C_{max}$	-	Pei et al. (2015)
$M\to C\|s-batch,p_{[j]}=\alpha_jt\|C_{max}$	-	
$1\|s-batch,p_{[j]}=\alpha_jt,s_{sd}\|C_{max}$	O($n$log$n$)	Pei et al. (2017)
$1\|s-batch,p_{[j]}=\alpha_jt,s_{sd}\|E_{max}$	O(n^2log$n$)	
$1\|s-batch,p_{[j]}=\alpha_jt,s_{sd}\|\sum U_j$	O(n^2log$n$)	
$P\|s-batch,p_{[j]}=p_j+\alpha t-\kappa u_j\|C_{max}$	NP-hard	Pei et al. (2019b)
$1\|g,p_{[j]}=\alpha_jt\|\sum w_jU_j$	NP-hard	Lee and Lu (2012)
$1\|g,p_{[gj]}=\alpha_{gj}t\|\sum w_{gj}C_{gj},f_{max}$	P	Wang and Liu (2014)
$1\|g,p_{[gj]}=p_{gj}+\alpha_{gj}t\|C_{max}$	P	Lee and Wu (2010)
$1\|g,p_{[gj]}=p_{gj}-\alpha_{gj}t\|C_{max}$	P	
$1\|g,p_{[gj]}=p_{gj}(\mu+\nu t)\|C_{max},\sum w_{gj}C_{gj}$	P	Wang, Lin, and Shan (2008)
$1\|p_{[jr]}=$ $\alpha_jtr^a\|C_{max},\sum F_j,\sum w_jF_j,\sum L_j,L_{max},\sum U_j$	-	Lee (2004)
$1\|p_{[jr]}=$ $(p+\alpha_jt)r^a\|C_{max},\sum F_j,\sum w_jF_j,\sum L_j,L_{max},\sum U_j$	-	
$1\|p_{[jr]}=\alpha_j(\mu+$ $\nu t)r^a\|C_{max},\sum C_j,\sum C_j^{S.4},\sum w_jC_j,L_{max},\sum U_j$	-	Wang, Jiang, and Wang (2009)
$Rm\|p_{[ijr]}=(p_{ij}+\alpha t)r^a\|\delta_1TC+\delta_2TADC+\delta_3TML$	O(n^{m+2})	Wang and Wang (2014)
$Rm\|p_{[ijr]}=$ $(p_{ij}+\alpha t)r^a\|\delta_1TW+\delta_2TADW+\delta_3TML$	O(n^{m+2})	
$F\|p_{[ijr]}=(p_{ij}+\alpha_{ij}t_{ij})r^{a_{ij}}\|C_{max},\sum T_j$	NP-hard	Fu et al. (2018)

Problem	complexity	paper
$1\|p_{[jr]} = p_j(f(t)+\beta r^a)\|C_{max}, \sum C_j, \sum C_j^2, \sum w_jC_j, L_{max}$	P	Wang (2007)
$1\|p_{[jr]} = p_j max\{r^{a_j}, \rho\} + \alpha t\|C_{max}, \sum C_j, \sum W_j, TADC, TADW, \sum(\delta_1 E_j + \delta_2 T_j + \delta_3 d_j)$	O($n$log$n$)	Niu, Wang, and Yin (2015)
$Pm\|p_{[jr]} = p_j r^{a_j} + \alpha t\|\delta_1 TC + \delta_2 TADC$ $Pm\|p_{[jr]} = p_j r^{a_j} + \alpha t\|\delta_1 TW + \delta_2 TADW$	O(n^{m+2}) P	Huang, Wang, and Ji (2014)
$Pm\|p_{[jr]} = p_j(M+(1-M)r^a)+\alpha t\|C_{max}, \sum C_j$	P,NP-hard	Ji et al. (2016)
$1\|p_{[jr]} = p_j(1+\sum_{l=1}^{r-1} p_{[l]})^\alpha r^a\|C_{max}, \sum C_j, \sum C_j^2, \sum w_jC_j, L_{max}$	P	Sun (2009)
$1\|p_{[jr]} = p_j(1+\sum_{l=1}^{r-1} logp_{[l]})^\alpha r^a\|C_{max}, \sum C_j, \sum C_j^2, \sum w_jC_j, \sum T_j, L_{max}$	P	Cheng, Lee, and Wu (2010)
$1\|p_{[jr]} = p_j(\frac{p+\sum_{l=1}^{r-1} p_{[l]}}{p+\sum_{l=1}^{n} p_l})^\alpha r^a\|C_{max}, \sum C_j, \sum C_j^\theta, \sum w_jC_j, \sum U_j, L_{max}$	P	Yin et al. (2010)
$Pm\|PM, p-batch, p_{[ijr_1r_2]} = p_j r_1^a + \alpha_i t\|C_{max}$	-	Kong et al. (2020a)
$1\|s-batch, p_{[ijr]} = p_j(1+\sum_{l=1}^{r-1}\sum_{\varphi=1}^{n} p_\varphi x_{i[\varphi][l]})^\alpha r^a\|C_{max}$	NP-hard	Pei et al. (2021)
$1\|g, p_{[gjr]} = p_{gj}(\mu_1 G_g^{r-1}+\nu_1)(\mu_2 t+\nu_2), \sum u_g \leq U\|C_{max}$ $1\|g, p_{[gjr]} = p_{gj}(\mu_1 G_g^{r-1}+\nu_1)(\mu_2 t+\nu_2), C_{max} \leq V\|\sum u_g$	O($n$log$n$) $max\{O(nlogn, O(ng(n)))\}$	Huang, Wang, and Wang (2011)
$1\|g, p_{gjr} = (p_{gj}+\alpha_g t)r^a\|C_{max}, \sum C_j$	O($n$log$n$)	He and Sun (2015)
$1\|s-batch, g, p_{[gjr]} = p_{gj} max\{r^{a_g}, \rho\} + \alpha t, R_g = t_0\|C_{max}$ $1\|s-batch, g, p_{[gjr]} = p_{gj} max\{r^{a_g}, \rho\} + \alpha t, R_g\|C_{max}$	O($n$log$n$) -	Fan et al. (2018)

Appendix D. Scheduling models

Model 1.1 (Biskup 1999)

$$\min \sum(\delta_1 E_j + \delta_2 T_j + \delta_3 C_j)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{jr} = 1, \quad r = 1, \cdots, n, \qquad (1.1\ \text{a})$$

$$\sum_{r=1}^{n} x_{jr} = 1, \quad j = 1, \cdots, n, \qquad (1.1\ \text{b})$$

$$x_{jr} \in \{0, 1\}, \quad j, r = 1, \cdots, n, \quad (1.1\ \text{c})$$

In Biskup (1999), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[jr]} = p_j r^a$; the equation $x_{jr} = 1$ indicates that job j is scheduled in position r, and $x_{jr} = 0$ otherwise; δ_1, δ_2, and δ_3 are the per time unit penalties for the earliness, the tardiness and the completion time.

Constraint (1.1 a) represents that only one job can be scheduled in position r. Constraint (1.1 b) represents that each job can be scheduled only once.

Model 1.2 (Eren and Güner 2007)

$$\min \delta_1 \sum C_r + \delta_2 \sum T_r$$

$$\text{s.t.} \quad (1.1\ \text{a})\text{-}(1.1\ \text{c})$$

$$p'_{[jr]} = \sum_{j=1}^{n} x_{jr} p_{[jr]}, \quad r = 1, \cdots, n, \quad (1.2\ \text{a})$$

$$D_r = \sum_{j=1}^{n} x_{jr} d_j, \quad r = 1, \cdots, n, \quad (1.2\ \text{b})$$

$$C_r \geq C_{r-1} + p'_{[jr]}, \quad r = 1, \cdots, n, \quad (1.2\ \text{c})$$

$$T_r \geq C_r - D_r, \quad r = 1, \cdots, n, \quad (1.2\ \text{d})$$

In Eren and Güner (2007), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[jr]} = p_j r^a$; δ_1 and δ_2 are the weights.

Constraints (1.2 b)-(1.2 d) represent the constraints of the due date, the completion time, and the tardiness of job sequenced in position r.

Model 1.3 (Mosheiov 2001)

$$\min \sum C_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ijr} = 1, \quad i = 1, \cdots, m, r = 1, \cdots, n_i, \quad (1.3\ \text{a})$$

$$\sum_{i=1}^{m} \sum_{r=1}^{n_i} x_{ijr} = 1, \quad j = 1, \cdots, n, \quad (1.3\ \text{b})$$

$$x_{ijr} \in \{0, 1\}, \quad i = 1, \cdots, m, j = 1, \cdots, n, r = 1, \cdots, n_i, \quad (1.3\ \text{c})$$

In Mosheiov (2001), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = p_{ij}r^a$; the equation $x_{ijr} = 1$ represents that job j is scheduled on machine i in position r, and $x_{jr} = 0$ otherwise; n_i is the number of jobs scheduled on machine i.

Constraint (1.3 a) represents that only one job can be scheduled on machine i in position r. Constraint (1.3 b) represents that each job can be scheduled only once.

Model 1.4 (Xu, Yin, and Li 2010)

min L_{max}

s.t. (1.3 a)-(1.3 c)

$$p'_{[ijr]} = \sum_{j=1}^{n} x_{ijr}p_{[ijr]}, \quad i = 1, \cdots, m, r = 1, \cdots, n_i, \quad (1.4\text{ a})$$

$$D_{ir} = \sum_{j=1}^{n} x_{ijr}d_{ij}, \quad i = 1, \cdots, m, r = 1, \cdots, n_i, \quad (1.4\text{ b})$$

$$C_{ir} \geq C_{i,r-1} + p'_{[ijr]}, \quad i = 1, \cdots, m, r = 1, \cdots, n_i, \quad (1.4\text{ c})$$

$$L_{max} \geq C_{ir} - D_{ir}, \quad i = 1, \cdots, m, r = 1, \cdots, n_i, \quad (1.4\text{ d})$$

In Xu, Yin, and Li (2010), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = p_{ij}r^a$.

Constraint (1.4 b) represents the due date of job j on machine i in position r. Constraints (1.4 c)-(1.4 d) represent the constraints of the completion time and the maximum lateness, respectively.

Model 1.5 (Eren and Güner 2008)

min $\delta_1 \sum C_r + \delta_2 C_{max}$

s.t. (1.1 a)-(1.1 c)

$$A_r = \sum_{j=1}^{n} x_{jr}p_{[1jr]}, \quad r = 1, \cdots, n, \quad (1.5\text{ a})$$

$$B_r = \sum_{j=1}^{n} x_{jr}p_{[2jr]}, \quad r = 1, \cdots, n, \quad (1.5\text{ b})$$

$$C_{max} = \sum_{r=1}^{n} X_r + \sum_{r=1}^{n} B_r, \quad (1.5\text{ c})$$

$$X_r = t_r + A_r + Y_r - C_{r-1}, \quad r = 1, \cdots, n, \quad (1.5\text{ d})$$

$$t_r \geq t_{r-1} + A_{r-1}, \quad r = 1, \cdots, n, \quad (1.5\text{ e})$$

In Eren and Güner (2008), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = p_{ij}r^a$; p_{1j} and p_{2j} are the normal processing time of job j on the first machine and the second machine, respectively; δ_1 and δ_2 are the weights; X_r is the idle time on the second machine for job in position r; Y_r is the duration between the completion time of job in position r at the first machine and its starting time at the second machine.

Constraints (1.5 a)-(1.5 b) represent the actual processing time of job j in position r on the first machine and the second machine, respectively. Constraint (1.5 c) denotes the makespan. Constraint (1.5 d) denotes the idel time on the second machine for the job in position r. Constraint (1.5 e) represents the constraint of the job starting time on the first machine.

Model 1.6 (Bai et al. 2018)

min C_{max}, $\sum C_j$, $\sum C_j^2$

s.t. (1.1 a)-(1.1 c)

$$R_j + \sum_{r=1}^{n} x_{jr} p_{[1jr]} \leq C_{1j}, \qquad j = 1, \cdots, n, \qquad (1.6\text{ a})$$

$$C_{ir} - C_{i-1,r} \geq \sum_{j=1}^{n} x_{jr} p_{[ijr]}, \quad i = 1, \cdots, m+1, r = 1, \cdots, n, \quad (1.6\text{ b})$$

$$C_{ir} - C_{i,r-1} \geq \sum_{j=1}^{n} x_{jr} p_{[ijr]}, \quad i = 1, \cdots, m, r = 1, \cdots, n+1, \quad (1.6\text{ c})$$

$$R_j \geq 0, p_{ij} \geq 0, C_{ij} \geq 0, \qquad i = 1, \cdots, m, j, r = 1, \cdots, n, \quad (1.6\text{ d})$$

In Bai et al. (2018), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = p_{ij} f(r)$, $f : [1, +\infty) \to (0, 1]$ is a non-increasing function with $0 < f(1) \leq 1$ and $f(r) \geq f(r+1)$.

Constraints (1.6 a)-(1.6 c) denotes the constraints of the job completion time.

Model 1.7 (Vahedi Nouri, Fattahi, and Ramezanian 2013)

min $\sum_{j=1}^{n} \pi_j T_j + \sum_{i=1}^{m} \sum_{k=1}^{K} \theta_{ik}(FM_{ik} - EM_{ik})$

s.t. (1.3 a), (1.3 c)

$$\sum_{r=1}^{n} x_{ijr} = 1, \qquad \forall i, \forall j \qquad (1.7\text{ a})$$

$$\sum_{j=1}^{n} C_{ij,r+1} \geq \sum_{j=1}^{n} C_{ijr} + \sum_{j=1}^{n} x_{ij,r+1} p_{[ij,r+1]}, \quad r = 1, \cdots, n-1, \forall i \quad (1.7\text{ b})$$

$$\sum_{r=1}^{n} C_{ijr} - \sum_{r=1}^{n} x_{ijr} p_{[ijr]} - FM_{ik} + z_{ijk}\phi \geq 0, \quad \forall i, \forall j, \forall k \quad (1.7\text{ c})$$

$$FM_{ik} - t_{ik} - \sum_{r=1}^{n} C_{ijr} + (1 - z_{ijk})\phi \geq 0 \qquad \forall i, \forall j, \forall k \qquad (1.7\text{ d})$$

$$EM_{ik} \leq FM_{ik} \leq \phi M_{ik}, \qquad \forall i, \forall k \qquad (1.7\text{ e})$$

$$\sum_{r=1}^{n} C_{i+1,jr} \geq \sum_{r=1}^{n} C_{ijr} + \sum_{j=1}^{n} x_{i+1,jr} p_{[i+1,jr]}, \quad i = 1, \cdots, m-1, \forall j \quad (1.7\text{ f})$$

$$\sum_{r=1}^{n} C_{1jr} \geq R_j + \sum_{j=1}^{n} x_{1jr} p_{[1jr]}, \qquad \forall j \qquad (1.7\text{ g})$$

$$C_{ijr} \leq \phi x_{ijr} \qquad \forall i, \forall j, \forall r \qquad (1.7\text{ h})$$

$$T_j \geq \sum_{r=1}^{n} C_{mjr} - d_j, \qquad \forall j \qquad (1.7\text{ i})$$

In Vahedi Nouri, Fattahi, and Ramezanian (2013), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = p_{ij} r^{a_i}$; ϕ is a large positive number; K is the number of maintenance activities; t_{ik} is the execution time of the kth maintenance activity

PM_{ik}; y_{ikr} is the binary parameter, $y_{ikr}=1$ if PM_{ik} is performed on machine i after the processing of rth job in the sequence, and $y_{ikr}=0$ otherwise.

Constraints (1.7 b)-(1.7h) denote the constraints of the job completion time in consideration of learning effects, maintenance activities, and the release dates. Constraint (1.7 e) denotes the tardiness of job j.

Model 1.8 (Zhu et al. 2011)

min $\delta_1 C_{max} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$

$\delta_1 \sum_{g=1}^{Q}\sum_{j=1}^{n_g} C_{gj} + \delta_2 \sum_{g=1}^{Q}\sum_{j=1}^{n_g}\Gamma_{gj}u_{gj}$

s.t. $\sum_{r_2=1}^{Q} x_{gr_2}=1, \quad g=1,\cdots,Q, \qquad (1.8\text{ a})$

$\sum_{g=1}^{Q} x_{gr_2}=1, \quad r_2=1,\cdots,Q, \qquad (1.8\text{ b})$

$x_{gr_2}\in\{0,1\}, \qquad g,r_2=1,\cdots,Q, \quad (1.8\text{ c})$

In Zhu et al. (2011), the related notations and constraints of the above model are defined as follows.

The actual processing time functions are $p_{[gjr_1r_2]}=(\frac{p_{gj}r_1^{a_1}r_2^{a_2}}{u_{gj}})^{\sigma}$ and $p_{[gjr_1r_2]}=p_{gj}r_1^{a_1}r_2^{a_2}-\kappa_{gj}u_{gj}$; r_1 is the position of job j scheduled in group g; r_2 is the position of group g; Q is the number of groups; the equation $x_{gr_2}=1$ indicates that group g is scheduled in position r_2, and $x_{gr_2}=0$ otherwise.

Constraint (1.8 a) represents that each group can be scheduled only once. Constraint (1.8 b) represents that only one group can be scheduled in position r_2.

Model 1.9 (Lu et al. 2017)

min $C_{max}|\sum_{g=1}^{Q}\sum_{j=1}^{n_g}u_{gj}\le U$

s.t. (1.8 a)-(1.8 c)

In Lu et al. (2017), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[gjr_1r_2]}=(\frac{p_{gj}r_1^{a_1}r_2^{a_2}}{u_{gj}})^{\sigma}$; $U>0$ is given and denotes the total available resource.

Model 1.10 (Huo, Ning, and Sun 2018)

min $\delta_1 C_{max} + \delta_2 k(\tau)$

$\delta_1 \sum_{g=1}^{Q}\sum_{j=1}^{n_g} C_{gj} + \delta_2 k(\tau)$

s.t. (1.8 a)-(1.8 c)

In Huo, Ning, and Sun (2018), the related notations and constraints of the above model are defined as follows.

$p_{[gjr_1r_2]}=p_{gj}r_1^{a_1log_2(1-\tau)a}r_2^{a_2log_2(1-\tau)a}$; $0\le\tau<1$ is the percentage reduction of standard learning indicator a; $k(\tau)$ shows the investment cost.

Model 2.1 (Cheng et al. 2014)

$$
\begin{aligned}
&\min \textstyle\sum_{r=1}^{n} C_{2r} \\
&\text{s.t.} \quad (1.1\text{ a})\text{-}(1.1\text{ c}) \\
&\qquad C_{1r} = C_{1,r-1}(1+\textstyle\sum_{j=1}^{n} x_{jr}\alpha_{1j}) \quad 1 \le r \le n \qquad (2.1\text{ a}) \\
&\qquad C_{2r} \ge C_{1r}(1+\textstyle\sum_{j=1}^{n} x_{jr}\alpha_{2j}) \quad 1 \le r \le n \qquad (2.1\text{ b}) \\
&\qquad C_{2r} \ge C_{2,r-1}(1+\textstyle\sum_{j=1}^{n} x_{jr}\alpha_{2j}) \quad 1 \le r \le n \qquad (2.1\text{ c}) \\
&\qquad C_{max}^{M} \ge C_{2,n} \qquad (2.1\text{ d}) \\
&\qquad C_{i,r} \ge 1, C_{1,0} = C_{2,0} = 1 \quad i = 1,2, 1 \le j \le n \qquad (2.1\text{ e})
\end{aligned}
$$

In Cheng et al. (2014), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ij]} = \alpha_{ij}t$; C_{ir} denotes the completion time of job in the position r on machine i; C_{max}^{M} denotes the optimal value of the makespan of Mosheiov's algorithm (Mosheiov 2002); α_{1j} and α_{2j} represent the deteriorating indicators of job j on machine 1 and machine 2, respectively.

Constraints (2.1 a)-(2.1c) represent the constraints of the completion time of job j in position r on machine 1 and machine 2. Constraint (2.1d) denotes the constraint of makespan.

Model 2.2 (Hsu et al. 2013)

$$
\begin{aligned}
&\min \textstyle\sum C_j, \sum C_{max}^{i} \\
&\text{s.t.} \quad (1.3\text{ a})\text{-}(1.3\text{ c})
\end{aligned}
$$

In Hsu et al. (2013), the actual processing time functions are $p_{[ijr]} = p_{ij}r^{\alpha_{ij}}$, $p_{[ijr]} = p_{ij} + \alpha_i t_{jr}$, and $p_{[ijr]} = p_{ij} + \alpha_{ij}r$.

Model 2.3 (Woo and Kim 2018)

$$
\begin{aligned}
&\min C_{max} \\
&\text{s.t.} \quad \sum_{j' \in J}\sum_{k \in K} x_{j'jk} = 1 \qquad \forall j \in J \qquad (2.3\text{ a}) \\
&\qquad \sum_{\substack{j' \in J \\ j \ne j'}} x_{j'jk} \le \sum_{j' \in J} x_{jj'k} \qquad \forall j \in J, \forall k \in K \qquad (2.3\text{ b}) \\
&\qquad \sum_{j \in J} x_{j'jk} \le 1 \qquad \forall k \in K \qquad (2.3\text{ c})
\end{aligned}
$$

$$\sum_{i\in I} y_{ik} \le 1 \qquad \forall k \in K \qquad (2.3\text{ d})$$

$$(1+\alpha_j)\eta_j + p_j - \eta_{j'} \le \pi(1-\sum_{k\in K} x_{j'jk}) \qquad \forall j,j' \in J, j=j' \qquad (2.3\text{ e})$$

$$C_{ik} \le \pi y_{ik} \qquad \forall i \in I, \forall k \in K \qquad (2.3\text{ f})$$

$$(1+\alpha_j)\eta_j + p_j - \sum_{i\in I} C_{ik} \le \pi(1-\sum_{j'\in J} x_{j'jk}) \qquad \forall j \in J, \forall k \in K \qquad (2.3\text{ g})$$

$$\sum_{k\in K} C_{ik} + \gamma(\sum_{k\in K} y_{ik} - 1) \le C_i \qquad \forall i \in I \qquad (2.3\text{ h})$$

$$C_i \le C_{max} \qquad \forall i \in I \qquad (2.3\text{ i})$$

$$C_{max} \ge 0, C_i \ge 0, C_{ik} \ge 0, \eta_j \ge 0 \qquad \forall j,j' \in J, \forall i \in I, \forall k \in K \qquad (2.3\text{ j})$$

$$x_{jj'k} \in \{0,1\}, y_{ik} \in \{0,1\} \qquad \forall j,j' \in J, \forall k \in K \qquad (2.3\text{ k})$$

In Woo and Kim (2018), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[j]} = p_j + \alpha_j \eta_j$; η_j is the gap between the starting time of job j and a recent rate-modifying activity; J denotes a set of jobs; I denotes a set of machines; K denotes a set of buckets; $x_{j'jk} = 1$ if job j' precedes job j in bucket k, and $x_{j'jk} = 0$ otherwise; $y_{ik} = 1$ represents that bucket k is processed in machine i, and $y_{ik} = 0$ otherwise.

Constraints (2.3 a)-(2.3 c) represent the rules of the job assignment. Constraint (2.3d) shows the assignment of the bucket. Constraint (2.3e) denotes the precedence relationship of jobs. Constraints (2.3f)-(2.3 g) show the completion time of a potential bucket. Constraints (2.3 h)-(2.3 i) denote the constraints of each machine.

Model 2.4 (Wang, Huang, and Wang 2019)

$$\min \sum T_j$$

$$\text{s.t.} \quad t_{ij} + p_{[ij]} \le t_{i+1,j} \qquad i=1,\cdots,m-1, j=1,\cdots,n \qquad (2.4\text{ a})$$

$$t_{ij'} + p_{[ij']} \le t_{ij} + \phi \times (1-x_{j'j}) \qquad i=1,\cdots,m, j,j'=1,\cdots,n, j\neq j' \qquad (2.4\text{ b})$$

$$x_{j'j} + x_{jj'} \le 1 \qquad i=1,\cdots,m, j,j'=1,\cdots,n, j\neq j' \qquad (2.4\text{ c})$$

$$C_{ij} \ge 0, t_{ij} \ge 0 \qquad i=1,\cdots,m, j=1,\cdots,n \qquad (2.4\text{ d})$$

$$x_{j'j} \in \{0,1\} \qquad j',j=1,\cdots,n \qquad (2.4\text{ e})$$

In Wang, Huang, and Wang (2019), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ij]} = p_{ij} + \alpha_{ij} t_{ij}$; ϕ is an infinite number; $x_{j'j} = 1$ if job j is followed by job j' immediately, and $x_{j'j} = 0$ otherwise.

Constraints (2.4 a)-(2.4 b) represent the constraints of job starting time. Constraint (2.4 c) denotes the jobs' order relation.

Model 2.5 (Yang 2013)

min $\sum C_j$

s.t. (1.3 a)-(1.3 c)

In Yang (2013), the actual processing time function is $p_{[ijr]} = p_{ij} f_{ij}(r)$, where $f_{ij}(r)$ is the deteriorating function of job in position r.

Model 2.6 (Zhang et al. 2018)

min $C_{max}, \sum C_j, \sum(\delta_1 E_j + \delta_2 T_j + \delta_3 D)$

s.t. (1.1 a)-(1.1 c)

In Zhang et al. (2018), the actual processing time function is $p_{[gjr]} = p_{gj}(1+\alpha)^{r-1}$.

Model 2.7 (Yang, Lee, and Guo 2013)

min $\sum_{v=1}^{V} \sum\limits_{j \in V_v} (\delta_1 E_j + \delta_2 T_j + \delta_3 D_v + \Gamma_j u_j)$

s.t. (1.1 a)-(1.1 c)

In Yang, Lee, and Guo (2013), the related notations and constraints of the above model are defined as follows.

The actual processing time functions are $p_{[jr]} = p_j f(r) - \kappa_j u_j$ and $p_{[jr]} = (\frac{p_j f(r)}{u_j})^{\sigma}$; $\delta_1, \delta_2, \delta_3 > 0$ are the unit time penalties of job earliness, tardiness, and due date, respectively; V is the number of due dates, v is the index of the due date, and V_v is the set of jobs with due date D_v.

Model 2.8 (Pei et al. 2015)

min C_{max}

s.t. $\sum_{b=1}^{N} x_{jb} = 1$	$j = 1, \cdots, n$	(2.8 a)
$\sum_{j=1}^{n} x_{jb} \le c$	$b = 1, \cdots, N$	(2.8 b)
$C_{1b} = t_{1b} + \prod\limits_{j=1+\sum_{l=1}^{b-1} n_l}^{\sum_{l=1}^{b} n_l} (1+\alpha_j)$	$b = 1, \cdots, N$	(2.8 c)
$t_{1,b+1} = C_{1b}$	$b = 1, \cdots, N-1$	(2.8 d)
$t_{2,b+1} \ge C_{1b} + T$	$b = 1, \cdots, N-1$	(2.8 e)
$C_{2b} = t_{2b} + \frac{T}{2}$	$b = 1, \cdots, N$	(2.8 f)
$C_{1b} - C_{1b'} + \Phi y_{bb'} - p_{[bj]} \ge 0$	$b, b' = 1, \cdots, N$	(2.8 g)
$C_{2b} - C_{2b'} + \Phi z_{bb'} - T \ge 0$	$b, b' = 1, \cdots, N$	(2.8 h)
$C_{max} \ge C_{2b}$	$b, b' = 1, \cdots, N$	(2.11 i)
$x_{jb}, y_{bb'}, z_{bb'} \in \{0, 1\}$	$j = 1, \cdots, n, b, b' = 1, \cdots, N$	(2.8 j)

In Pei et al. (2015), the related notations and constraints of the above model are defined as follows.

The actual processing time functions are $p_{[j]} = \alpha_j t$ and $p_{[bj]} = t \prod_{j \in batch-b} (1+\alpha_j) - t$; $x_{jb} = 1$ if job j is assigned to batch b, and $x_{jb} = 0$ otherwise; $y_{bb'} = 1$ and $z_{bb'} = 1$ if batch b precedes batch b' during the production stage and the transportation stage, respectively, and $y_{bb'} = 0$ and $z_{bb'} = 0$ otherwise; C_{1b} and C_{2b} represent the completion time of batch b during the production stage and the transportation stage, respectively; t_{1b} and t_{2b} represent the starting time of batch b during the production stage and the transportation stage, respectively; Φ is a large number; T is the round-trip time between the manufacture and customer; c means the capacity of machine and vehicle.

Constraints (2.8 g)-(2.8 h) represent that there is no overlap between two batches at any two stages.

Model 2.9 (Pei et al. 2015)

min C_{max}

s.t. (2.8 a)-(2.8 c), (2.8 e)-(2.8 j)

$$t_{1,b+1} = max\{C_{1b}, t_{1b} + T\} \qquad b = 1, \cdots, N-1 \quad (2.9\text{ a})$$

The related notations and constraints of the above model are the same as those of Model 2.8, and the constraint of the starting time $t_{1,b+1}$ has changed.

Model 3.1 (Wang and Wang 2014)

min $\delta_1 TC + \delta_2 TADC + \delta_3 TML$

$$\delta_1 TW + \delta_2 TADW + \delta_3 TML$$

s.t. (1.3 a)-(1.3 c)

In Wang and Wang (2014), the actual processing time function is $p_{[ijr]} = (p_{ij} + \alpha t) r^a$. The variables $\delta_1, \delta_2, \delta_3 \geq 0$ are the given weights.

Model 3.2 (Wang and Wang 2014)

min $\delta_1 TC + \delta_2 TADC + \delta_3 TML$

$$\delta_1 TW + \delta_2 TADW + \delta_3 TML$$

s.t. (1.3 b)-(1.3 c)

$$\sum_{j=1}^{n} x_{ijr} \leq 1 \quad i = 1, \cdots, m, r = 1, \cdots, n \quad (3.2\text{ a})$$

The related notations and constraints of the above model are the same as those of Model 3.1, where the actual processing time function is $p_{[ijr]} = (p_{ij} + \alpha t) r^a$.

Constraint (3.2 a) shows the phenomenon that no job is assigned in the position r on machine i.

Model 3.3 (Fu et al. 2018)

$$\min\ C_{max}, \sum T_j$$

$$\text{s.t.}\quad (1.1\text{ a})\text{-}(1.1\text{ c})$$

$$t_{ir} + p_{[ir]} \leq t_{i+1,r} \quad i=1,\cdots,m-1, r=1,\cdots,n \quad (3.3\text{ a})$$

$$t_{ir} + p_{[ir]} \leq t_{i,r+1} \quad i=1,\cdots,m, r=1,\cdots,n-1 \quad (3.3\text{ b})$$

$$C_{max} \geq C_j \quad j=1,\cdots,n \quad (3.3\text{ c})$$

$$C_j \geq C_{ij} \quad i=1,\cdots,m, j=1,\cdots,n \quad (3.3\text{ d})$$

$$t_{ij} \geq 0, C_{ij} \geq 0 \quad i=1,\cdots,m, j=1,\cdots,n \quad (3.3\text{ e})$$

In Fu et al. (2018), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $p_{[ijr]} = (p_{ij} + \alpha_{ij} t_{ij}) r^{a_{ij}}$. The parameter C_j represents the completion time of job j on the last machine.

Constraints (3.3 a)-(3.3 b) represent the constraints of the job starting time. Constraints (3.3 c)-(3.3 d) show the constraints of the job completion time.

Model 3.4 (Niu, Wang, and Yin 2015)

$$\min\ C_{max}, \sum C_j, \sum W_j, TADC, TADW$$

$$\sum_{j=1}^{n} (\delta_1 E_j + \delta_2 T_j + \delta_3 D_j)$$

$$\text{s.t.}\quad (1.1\text{ a})\text{-}(1.1\text{ c})$$

In Niu, Wang, and Yin (2015), the actual processing time function is $p_{[jr]} = p_j max\{r^{a_j}, \rho\} + \alpha t$; the variables $\delta_1, \delta_2, \delta_3 \geq 0$ are the given weights.

Model 3.5 (Huang, Wang, and Ji 2014)

$$\min\ \delta_1 TC + \delta_2 TADC$$

$$\delta_1 TW + \delta_2 TADW$$

$$\text{s.t.}\quad (1.1\text{ a})\text{-}(1.1\text{ c})$$

In Huang, Wang, and Ji (2014), the actual processing time function is $p_{[jr]} = p_j r^{a_j} + \alpha t$; the variables $\delta_1, \delta_2 \geq 0$ are the given weights.

Model 3.6 (Yusriski et al. 2016)

min $\sum_{b=1}^{N}(\sum_{l=1}^{b}(s+T_{[l]}Q_{[l]})-s)Q_{[b]}$

s.t. $\sum_{b=1}^{N} Q_{[b]}=n,$ (3.6 a)

$\sum_{b=1}^{N} T_{[b]}Q_{[b]}+(N-1)s\leq d,$ (3.6 b)

$t_{[1]}+T_{[1]}Q_{[1]}=d,$ (3.6 c)

$Q_{[b]}\geq 1$ and integer, (3.6 d)

$1\leq N\leq n$ and integer, (3.6 e)

In Yusriski et al. (2016), the related notations and constraints of the above model are defined as follows.

The actual processing time function is $T_{[b]} = max\{p(1+\sum_{l=b}^{N} Q_{[l+1]})^{-log(a)/log(2)}, \rho\} + \mu(\sum_{l=b}^{N} T_{[l+1]}Q_{[l+1]}/\alpha)^{\beta}$; $T_{[b]}$ is the bth batch processing time, $b=1,\cdots,N$; N is the number of batches; n is the number of jobs; s is the setup time of batch; $Q_{[b]}$ is the number of jobs in batch b; d denotes the due date; $t_{[b]}$ is the starting time of bth batch.

The objective function is to minimize the total flow time. Constraints (3.6 b)-(3.6c) denote the constraints of due date.

Model 3.7 (Yusriski et al. 2018)

min $\sum_{b=1}^{N}(\{\sum_{l=1}^{b}(s+T_{[l]}Q_{[l]})-s-T_{[b]}Q_{[b]}\}\delta_1 Q_{[b]}+\delta_2 T_{[b]}Q_{[b]}^2)$

s.t. (3.6 a)-(3.6 e)

The related notations and constraints of the above model are the same as those of Model 3.6, where the processing time function $T_{[b]}$ is too complex to show in this appendix; δ_1 and δ_2 denote the unit inventory holding cost for a part in the completed batches and in-process batches, respectively. The objective function is to minimize the total inventory holding cost.

Appendix E. Other studies on learning and deteriorating effects

Appendix E.1. Extensions of pure non-linear learning function $p_{[jr]} = p_j r^a$

In manufacturing scenarios, it is highly unrealistic that the job processing time drops to zero precipitously with the increase of already processed jobs. Hence, a truncation parameter of $\rho \in (0,1)$ was introduced into $p_{[ijr]} = p_{ij} r^a$, namely, $p_{[ijr]} = p_{ij} max\{r^a, \rho\}$ with $a \leq 0$. This learning function was usually applied in two-machine flowshop scheduling problems (Li et al. 2011a, Cheng et al. 2013, Wang et al. 2013). Li et al. (2011a) proposed a B&B algorithm and three SA algorithms to minimize $\sum C_j$. Cheng et al. (2013) designed a B&B algorithm and three genetic algorithms (GAs) to minimize C_{max}. For six general performance criteria, Wang et al. (2013) proposed SPT, WSPT, and WDSPT algorithms, etc.

Since the part of job processing time is limited by some conditions and it cannot be shorten, DeJong's learning function $p_{[jr]} = p_j(M + (1-M)r^a)$ and some improved functions were proposed in parallel-machine and flowshop environments, as shown in Figure S.1. The parameter $M \in [0,1]$ denotes the incompressibility factor, that is, the incompressibility of the job processing time. DeJong's learning function $p_{[jr]} = p_j(M + (1-M)r^a)$ was applied in parallel-machine makespan minimization problem (Okołowski and Gawiejnowicz 2010, Hidri and Jemmali 2020). Then, combining with truncated effect, Amirian and Sahraeian (2014) presented a modified DeJong's learning function $p_{[ijr]} = p_{ij}(M + (1-M)max\{r^a, \rho\})$. Amirian and Sahraeian (2016) further improved the function in consideration of operator's prior experience and machine-based learning indicator, see $p_{[ijr]} = p_{ij}(M_{ij} + (1-M_{ij})(\rho + (B_j + r)^{a_i})$, where B is abstracted from *Stanford-B* learning curve (Fogliatto and Anzanello 2011), $M_{ij} \in [0,1]$, $a_i \leq 0$, and $\rho + (B_j + r)^{a_i} \leq 1$. As Figure S.1 shows, the constraints (AE1-1c)-(AE1-1d) of the proposed Model AE1-1 define the lateness and release date of each job. Regarding algorithms, Okołowski and Gawiejnowicz (2010) proposed two B&B algorithms and two greedy heuristics, while Amirian and Sahraeian (2016) and Amirian and Sahraeian (2014) designed multi-objective differential evolution (MODES) and multi-objective simulated annealing differential evolution (MO-SADE) algorithms. In Hidri and Jemmali (2020), two types of heuristic algorithms were proposed based on dispatching rules with new enhancement methods and exact solutions, respectively.

In the chemical industry, the job processing time can be compressed if extra costs are paid to increase catalysts (Wang and Cheng 2005). Then, scheduling problems with position- and resource-based learning effects were studied. As shown in Figure S.2, it is found that the single-machine and no-wait two-machine flowshop cases are all formulated as common due date assignment models, see Models AE1-2, AE1-3, AE1-4, and AE1-5. Moreover, all of them proposed polynomial time algorithms to solve these problems. In the initial research, a linear resource consumption function $p_{[jr]} = p_j r^a - \kappa_j u_j$ with $0 \leq u_j < \frac{p_j n^a}{\kappa_j}$ and a convex resource consumption function $p_{[jr]} = (\frac{p_j r^a}{u_j})^\sigma$ with $u_j > 0$ were proposed (Wang, Wang, and Wang 2010). In these two functions, u_j is the amount of resource allocated to the job j, κ_j is the positive compression rate, and σ is a positive constant. It is found that $p_{[ijr]} = (\frac{p_{ij} r^a}{u_{ij}})^\sigma$ with

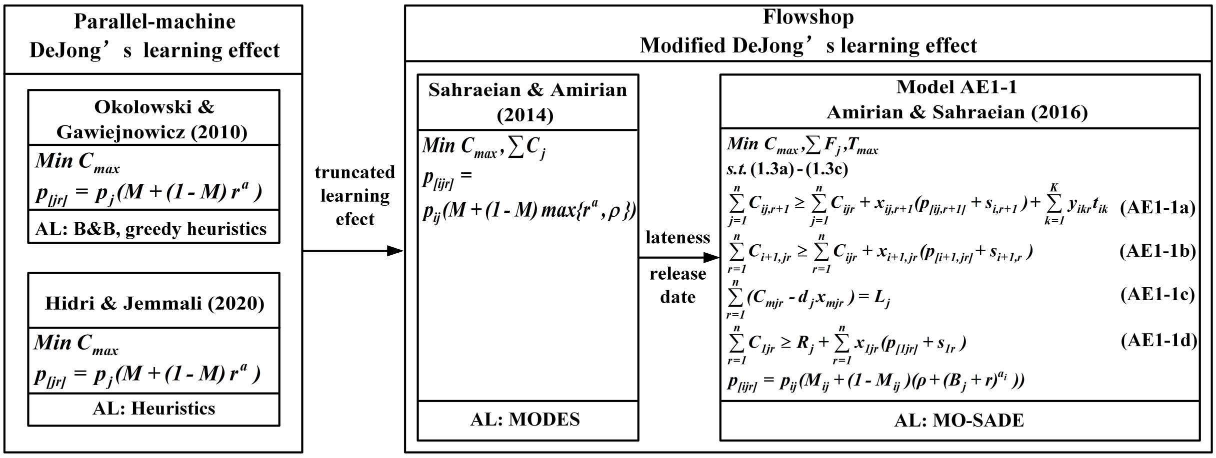

Figure S.1 Models of problems on DeJong's learning effects

$a \leq 0$ and $u_{ij} > 0$ was popular in no-wait two-machine flowshop scheduling problems (Gao et al. 2018, Geng, Wang, and Bai 2018, Tian et al. 2018, Liu and Feng 2014, Sun et al. 2018). Particularly, both Geng, Wang, and Bai (2018) and Tian et al. (2018) took into account total resource constraints $\sum_{i=1}^{2}\sum_{j=1}^{n}\Gamma_{ij}u_{ij} \leq U$, where Γ_{ij} denotes the cost related to the resource allocation per unit time and U is the upper bound of the resource cost.

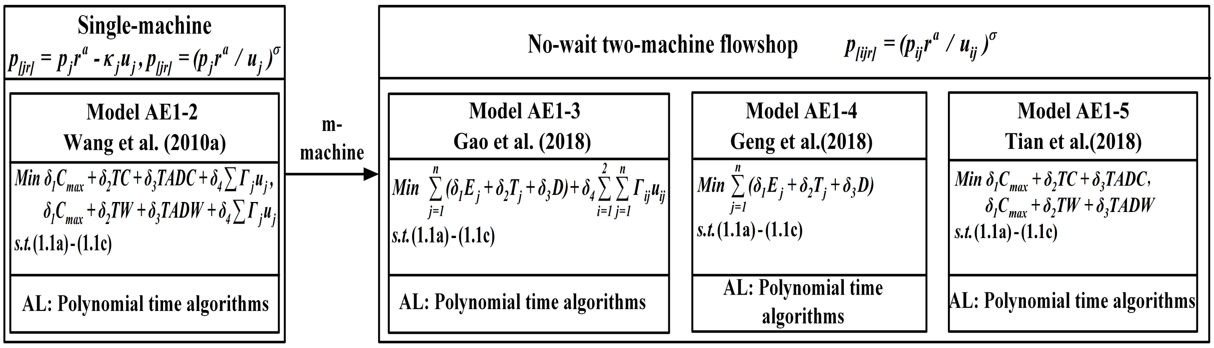

Figure S.2 Models of problems with non-linear functions considering resource allocation

Appendix E.2. Extensions of linear starting time-dependent functions with fixed processing time $p_{[j]} = p_j + \alpha t$

Actually, the deteriorating function $p_{[ij]} = p_{ij}(\mu + \nu t)$ with constant number $\mu, \nu \geq 0$ and deteriorating indicator $\alpha_{ij} \leq 0$ can be regarded as another representation of $p_{[ij]} = p_{ij} + \alpha_{ij} t$, which is common in the flowshop cases. In the context of m-machine flowshop cases, Bank et al. (2012a) proposed PSO and SA algorithms for minimizing $\sum T_j$. Besides meta-heuristic algorithms, Bank et al. (2012b) and Ng et al. (2010) both utilized B&B algorithms to solve two-machine flowshop scheduling problems, with the objectives to minimize L_{max} and $\sum w_j C_j$, respectively. In addition to flowshop scheduling, the total deviation of completion time minimization problems were solved by heuristic algorithms in the single-machine environment, where deteriorating function is $p_{[j]} = p_j(\mu + \nu t_j)$ (Li et al. 2009).

Considering the compression of job processing time in realistic situations, two resource-dependent deteriorating functions $p_{[j]} = p_j + \alpha_j t - \kappa u_j$ and $p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha_j t$ were presented in single-machine scheduling problems, where $\alpha_j > 0$, $\sigma > 0$. The term u_j is the amount of resource allocated to the job j, and κ_j is a positive parameter, denoting the workload of job j. Given the linear function $p_{[j]} = p_j + \alpha t - \kappa u_j$, Wei, Wang, and Ji (2012) utilized assignment models to solve two multi-objective problems. The expressions for objectives are $\delta_1 C_{max} + \delta_2 TC + \delta_3 TADC + \delta_4 \sum \Gamma_j u_j$ and $\delta_1 C_{max} + \delta_2 TW + \delta_3 TADW + \delta_4 \sum \Gamma_j u_j$. Additionally, in order to solve scheduling problems with convex function $p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha_j t$, Li and Wang (2018) and Liu et al. (2019) both designed $O(nlogn)$-time algorithms. The former studied three problems $1|p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha_j t_j|C_{max} + \theta \sum \Gamma_j u_j$, $1|p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha_j t_j, C_{max} < \hat{C}| \sum \Gamma_j u_j$, and $1|p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha_j t_j, \sum C_j < \hat{TC}| \sum \Gamma_j u_j$, where θ is a given number. The latter addressed two problems $1|p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha t, \sum \Gamma_j u_j < U|\lambda$ and $1|p_{[j]} = (\frac{\kappa_j}{u_j})^\sigma + \alpha t, \lambda < \phi| \sum \Gamma_j u_j$, where U and ϕ is the upper bound of total resource cost and schedule cost, respectively. The parameter λ is a set of objective functions including C_{max}, $\sum C_j$, $\sum W_j$, etc. Given convex function $p_{[j]} = (\frac{p_j}{u_j})^\sigma + \alpha t$, Liu, Yao, and Jiang (2020) investigated a bi-criteria scheduling problems where the first objective is to minimize scheduling cost and the second objective is to minimize resource consumption cost. They proposed common due-date assignment and slack due-date assignment methods.

Appendix E.2. Extensions of learning-deterioration function $p_{[jr]} = (p_j + \alpha_j t) r^a$

Apart from single-machine cases studied by Ceylan (2014), most papers focused on parallel-machine scheduling problems in this field. Ceylan (2014) proposed the learning-deterioration function $p_{[r]} = p_r + (\alpha \times C_{r-1}) r^a$. Additionally, there were three papers on earliness and tardiness minimization scheduling problems with $p_{[r]} = (p_r + \alpha \times C_{r-1}) r^a$ (Toksarı and Güner 2008, 2009, 2010). They all proposed mixed non-linear integer programming models considering various constraints, see Models AE3-1, AE3-2, and AE3-3 in Figure S.3. Specifically, in Models AE3-1 and AE3-3, the constraint $C_{ir} = C_{i,r-1} + \sum_{j'=1}^n \sum_{j=1}^n (s_{j'j} x_{ij'r} x_{ij,r+1}) + p_{[ir]}$ shows the actual job completion time considering sequence-dependent setup times $s_{j'j}$. This is different from the constraint $C_{ir} = C_{i,r-1} + p_{[ir]}$ of Model AE3-2. Furthermore, due to the up and down in machine speed and breakdowns, Arık and Toksarı (2018) investigated multi-objective fuzzy problems with four learning-deterioration functions, i.e., $p_{[r]} = (p_r + \alpha_1 \times C_{r-1}) r^a$, $p_{[r]} = (p_r + \alpha_1 \times C_{r-1}^{\alpha_2}) r^a$, $p_{[r]} = (p_r + \alpha_1 \times C_{r-1})(1 + \sum_{l=1}^{r-1} p_{[l]})^a$, and $p_{[r]} = (p_r + \alpha_1 \times C_{r-1}^{\alpha_2})(1 + \sum_{l=1}^{r-1} p_{[l]})^a$. Thereinto, $\alpha_1 > 0$ and $\alpha_2 > 0$ are linear and non-linear deteriorating indicator, respectively. Under the fuzzy environment, they built Model AE3-4 based on the fuzzy setting, and depicted the relationship among completion time, earliness, tardiness, and due date, see Figure S.3. A local search algorithm with different solution techniques was designed for these problems.

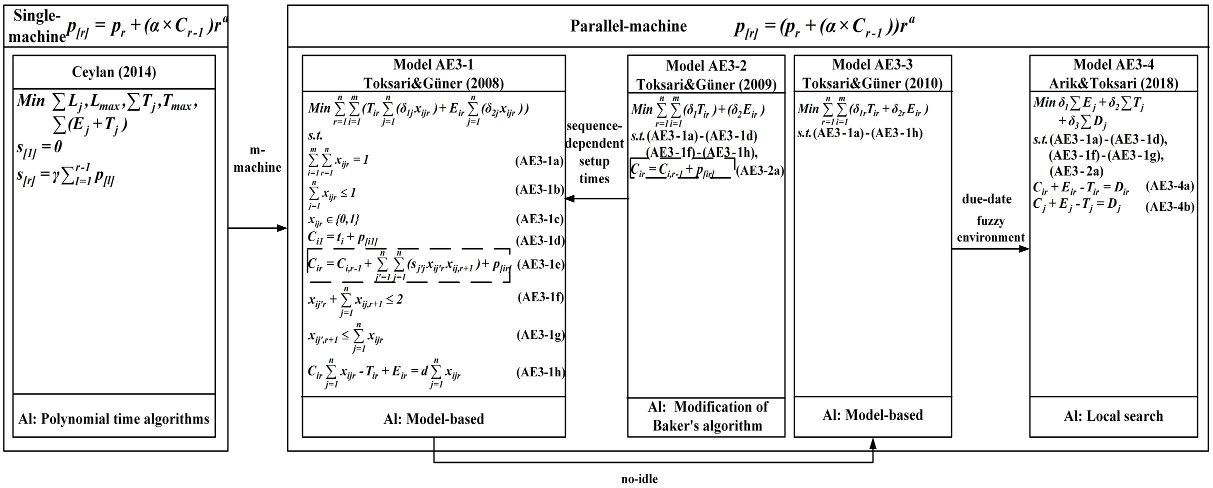

Figure S.3 Models of problems with functions based on $p_{[r]} = p_r + (\alpha \times C_{r-1})r^a$ and $p_{[r]} = (p_r + (\alpha \times C_{r-1}))r^a$

References

Amirian, H., and R. Sahraeian. 2014. "Multi-objective differential evolution for the flow shop scheduling problem with a modified learning effect." *International Journal of Engineering Transactions B: Applications* 27 (9).

Amirian, H., and R. Sahraeian. 2016. "A hybrid differential evolution for general multi-objective flow shop problem with a modified learning effect." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 230 (12): 2275–2285.

Arık, O. A., and M. D. Toksarı. 2018. "Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects." *International Journal of Production Research* 56 (7): 2488–2505.

Bachman, A., and A. Janiak. 2000. "Minimizing maximum lateness under linear deterioration." *European Journal of Operational Research* 126 (3): 557–566.

Bachman, A., and A. Janiak. 2004. "Scheduling jobs with position-dependent processing times." *Journal of the Operational Research Society* 55 (3): 257–264.

Bai, D., M. Tang, Z. H. Zhang, and E. D. Santibanez-Gonzalez. 2018. "Flow shop learning effect scheduling problem with release dates." *Omega* 78: 21–38.

Bai, J., M. Z. Wang, and J. B. Wang. 2012. "Single machine scheduling with a general exponential learning effect." *Applied Mathematical Modelling* 36 (2): 829–835.

Bank, M., S. F. Ghomi, F. Jolai, and J. Behnamian. 2012a. "Application of particle swarm optimization and simulated annealing algorithms in flow shop scheduling problem under linear deterioration." *Advances in Engineering Software* 47 (1): 1–6.

Bank, M., S. F. Ghomi, F. Jolai, and J. Behnamian. 2012b. "Two-machine flow shop total tardiness scheduling problem with deteriorating jobs." *Applied Mathematical Modelling* 36 (11): 5418–5426.

Biskup, D. 1999. "Single-machine scheduling with learning considerations." *European Journal of Operational Research* 115 (1): 173–178.

Ceylan, H. 2014. "Due date single machine scheduling problems with nonlinear deterioration and learning effects and past sequence dependent setup times." *Mathematical Problems in Engineering* 2014.

Cheng, M., P. R. Tadikamalla, J. Shang, and S. Zhang. 2014. "Bicriteria hierarchical optimization of two-machine flow shop scheduling problem with time-dependent deteriorating jobs." *European Journal of Operational Research* 234 (3): 650–657.

Cheng, T C E, Wenchiung Lee, and Chinchia Wu. 2011. "Single-machine scheduling with deteriorating jobs and past-sequence-dependent setup times." *Applied Mathematical Modelling* 35 (4): 1861–1867.

Cheng, T. C. E., C. C. Wu, J. C. Chen, W. H. Wu, and S. R. Cheng. 2013. "Two-machine flowshop scheduling with a truncated learning function to minimize the makespan." *International Journal of Production Economics* 141 (1): 79–86.

Cheng, T. E., W. C. Lee, and C. C. Wu. 2010. "Scheduling problems with deteriorating jobs and learning effects including proportional setup times." *Computers & Industrial Engineering* 58 (2): 326–331.

Cheng, T. E., and G. Wang. 2000. "Single machine scheduling with learning effect considerations." *Annals of Operations Research* 98 (1-4): 273–290.

Cheng, T. E., C. C. Wu, and W. C. Lee. 2008. "Some scheduling problems with sum-of-processing-times-based and job-position-based learning effects." *Information Sciences* 178 (11): 2476–2487.

Eren, T., and E. Güner. 2007. "A bicriteria scheduling problem with a learning effect: Total completion time and total tardiness." *INFOR: Information Systems and Operational Research* 45 (2): 75–81.

Eren, T., and E. Güner. 2008. "A bicriteria flowshop scheduling with a learning effect." *Applied Mathematical Modelling* 32 (9): 1719–1733.

Fan, W., J. Pei, X. Liu, P. M. Pardalos, and M. Kong. 2018. "Serial-batching group scheduling with release times and the combined effects of deterioration and truncated job-dependent learning." *Journal of Global Optimization* 71 (1): 147–163.

Fogliatto, F. S., and M. J. Anzanello. 2011. "Learning curves: the state of the art and research directions." *Learning curves: Theory, models, and applications* 3–21.

Fu, Y., J. Ding, H. Wang, and J. Wang. 2018. "Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in Industry 4.0-based manufacturing system." *Applied Soft Computing* 68: 847–855.

Gao, F., M. Liu, J. J. Wang, and Y. Y. Lu. 2018. "No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times." *International Journal of Production Research* 56 (6): 2361–2369.

Geng, X. N., J. B. Wang, and D. Bai. 2018. "Common due date assignment scheduling for a no-wait flowshop with convex resource allocation and learning effect." *Engineering Optimization* 1–23.

He, H. 2016. "Minimization of maximum lateness in an m-machine permutation flow shop with a general exponential learning effect." *Computers & Industrial Engineering* 97: 73–83.

He, Y., and L. Sun. 2015. "One-machine scheduling problems with deteriorating jobs and position-dependent learning effects under group technology considerations." *International Journal of Systems Science* 46 (7): 1319–1326.

Hidri, L., and M. Jemmali. 2020. "Near-optimal solutions and tight lower bounds for the parallel machines scheduling problem with learning effect." *RAIRO-Operations Research* 54 (2): 507–527.

Hsu, C. J., M. Ji, J. Y. Guo, and D. L. Yang. 2013. "Unrelated parallel-machine scheduling problems with aging effects and deteriorating maintenance activities." *Information Sciences* 253: 163–169.

Huang, X., M. Z. Wang, and P. Ji. 2014. "Parallel machines scheduling with deteriorating and learning effects." *Optimization Letters* 8 (2): 493–500.

Huo, J. Z., L. Ning, and L. Sun. 2018. "Group Scheduling with General Autonomous and Induced Learning Effect." *Mathematical Problems in Engineering* 2018.

Ji, M., and T. E. Cheng. 2009. "Parallel-machine scheduling of simple linear deteriorating jobs." *Theoretical Computer Science* 410 (38): 3761–3768.

Ji, M., X. Tang, X. Zhang, and T. E. Cheng. 2016. "Machine scheduling with deteriorating jobs and DeJong's learning effect." *Computers & Industrial Engineering* 91: 42–47.

Kong, Min, Xinbao Liu, Jun Pei, Hao Cheng, and Panos M Pardalos. 2020a. "A BRKGA-DE algorithm for parallel-batching scheduling with deterioration and learning effects on parallel machines under preventive maintenance consideration." *Annals of Mathematics and Artificial Intelligence* 88 (1): 237–267.

Kong, Min, Xinbao Liu, Jun Pei, Zhiping Zhou, and Panos M Pardalos. 2020b. "Parallel-batching scheduling of deteriorating jobs with non-identical sizes and rejection on a single machine." *Optimization Letters* 14 (4): 857–871.

Koulamas, C., and G. J. Kyparisis. 2007. "Single-machine and two-machine flowshop scheduling with general learning functions." *European Journal of Operational Research* 178 (2): 402–407.

Kuo, W. H. 2012. "Single-machine group scheduling with time-dependent learning effect and position-based setup time learning effect." *Annals of Operations Research* 196 (1): 349–359.

Kuo, W. H., and D. L. Yang. 2006. "Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect." *European Journal of Operational Research* 174 (2): 1184–1190.

Lee, W. C. 2004. "A note on deteriorating jobs and learning in single-machine scheduling problems." *International Journal of Business and Economics* 3 (1): 83.

Lee, W. C., and Z. S. Lu. 2012. "Group scheduling with deteriorating jobs to minimize the total weighted number of late jobs." *Applied Mathematics and Computation* 218 (17): 8750–8757.

Lee, W. C., and C. C. Wu. 2010. "A note on optimal policies for two group scheduling problems with deteriorating setup and processing times." *Computers & Industrial Engineering* 58 (4): 646–650.

Li, D. C., P. H. Hsu, C. C. Wu, and T. E. Cheng. 2011a. "Two-machine flowshop scheduling with truncated learning to minimize the total completion time." *Computers & Industrial Engineering* 61 (3): 655–662.

Li, L., and J. J. Wang. 2018. "Scheduling jobs with deterioration effect and controllable processing time." *Neural Computing and Applications* 29 (11): 1163–1170.

Li, S., C. T. Ng, T. E. Cheng, and J. Yuan. 2011b. "Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan." *European Journal of Operational Research* 210 (3): 482–488.

Li, S., and J. Yuan. 2010. "Parallel-machine scheduling with deteriorating jobs and rejection." *Theoretical Computer Science* 411 (40): 3642–3650.

Li, Y., G. Li, L. Sun, and Z. Xu. 2009. "Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times." *International Journal of Production Economics* 118 (2): 424–429.

Lin, W. C., C. C. Wu, K. Yu, Y. H. Zhuang, and S. C. Liu. 2017. "On the use of genetic algorithm for solving re-entrant flowshop scheduling with sum-of-processing-times-based learning effect to minimize total tardiness." *Intelligent Automation & Soft Computing* 1–11.

Liu, H. C., W. C. Lee, and C. C. Wu. 2008. "A note on single-machine group scheduling problem with a general learning function." *Journal of Statistics and Management Systems* 11 (4): 645–652.

Liu, L., Z. Shi, and L. Shi. 2018. "Minimization of total energy consumption in an m-machine flow shop with an exponential time-dependent learning effect." *Frontiers of Engineering Management* 5 (4): 487–498.

Liu, S., X. Liu, J. Pei, P. M. Pardalos, and Q. Song. 2020. "Parallel-batching machines scheduling problem with a truncated time-dependent learning effect via a hybrid CS-JADE algorithm." *Optimization Methods and Software* 35 (1): 116–141.

Liu, W., C. Jiang, J. B. Wang, and Y. Y. Lu. 2019. "Single-machine scheduling with simultaneous considerations of resource allocation and deteriorating jobs." *The Computer Journal* 62 (1): 81–89.

Liu, W., Y. Yao, and C. Jiang. 2020. "Single-machine resource allocation scheduling with due-date assignment, deterioration effect and position-dependent weights." *Engineering Optimization* 52 (4): 701–714.

Liu, Y., and Z. Feng. 2014. "Two-machine no-wait flowshop scheduling with learning effect and convex resource-dependent processing times." *Computers & Industrial Engineering* 75 (1): 170–175.

Low, C., and W. Y. Lin. 2011. "Minimizing the total completion time in a single-machine scheduling problem with a learning effect." *Applied Mathematical Modelling* 35 (4): 1946–1951.

Low, C., and W. Y. Lin. 2012. "Single machine group scheduling with learning effects and past-sequence-dependent setup times." *International Journal of Systems Science* 43 (1): 1–8.

Lu, Y. Y., J. B. Wang, P. Ji, and H. He. 2017. "A note on resource allocation scheduling with group technology and learning effects on a single machine." *Engineering Optimization* 49 (9): 1621–1632.

Ma, H., C. Shao, and X. R. Wang. 2014. "Single machine scheduling with exponential learning functions." *Optimization Letters* 8 (4): 1273–1285.

Mosheiov, G. 1994. "Scheduling jobs under simple linear deterioration." *Computers & Operations Research* 21 (6): 653–659.

Mosheiov, G. 2001. "Parallel machine scheduling with a learning effect." *Journal of the Operational Research Society* 52 (10): 1165–1169.

Mosheiov, G. 2002. "Complexity analysis of job-shop scheduling with deteriorating jobs." *Discrete Applied Mathematics* 117 (1): 195–209.

Ng, C. T., J. B. Wang, T. E. Cheng, and L. L. Liu. 2010. "A branch-and-bound algorithm for solving a two-machine flow shop problem with deteriorating jobs." *Computers & Operations Research* 37 (1): 83–90.

Niu, Y. P., J. Wang, and N. Yin. 2015. "Scheduling problems with effects of deterioration and truncated job-dependent learning." *Journal of Applied Mathematics and Computing* 47 (1-2): 315–325.

Okołowski, D., and S. Gawiejnowicz. 2010. "Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect." *Computers & Industrial Engineering* 59 (2): 272–279.

Pei, J., B. Cheng, X. Liu, P. M. Pardalos, and M. Kong. 2019a. "Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time." *Annals of Operations Research* 272 (1-2): 217–241.

Pei, J., X. Liu, W. Fan, P. M. Pardalos, and S. Lu. 2019b. "A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers." *Omega* 82: 55–69.

Pei, J., X. Liu, B. Liao, P. M. Pardalos, and M. Kong. 2018. "Single-machine scheduling with learning effect and resource-dependent processing times in the serial-batching production." *Applied Mathematical Modelling* 58: 245–253.

Pei, J., X. Liu, P. M. Pardalos, W. Fan, and S. Yang. 2017. "Scheduling deteriorating jobs on a single serial-batching machine with multiple job types and sequence-dependent setup times." *Annals of Operations Research* 249 (1-2): 175–195.

Pei, J., P. M. Pardalos, X. Liu, W. Fan, and S. Yang. 2015. "Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan." *European Journal of Operational Research* 244 (1): 13–25.

Pei, J., Q. Song, B. Liao, X. Liu, and P. M. Pardalos. 2021. "Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration." *Annals of Operations Research* 298: 407–444.

Qin, H., Z. H. Zhang, and D. Bai. 2016. "Permutation flowshop group scheduling with position-based learning effect." *Computers & Industrial Engineering* 92: 1–15.

Sun, L. 2009. "Single-machine scheduling problems with deteriorating jobs and learning effects." *Computers & Industrial Engineering* 57 (3): 843–846.

Sun, L. H., K. Cui, J. H. Chen, J. Wang, and X. C. He. 2013. "Some results of the worst-case analysis for flow shop scheduling with a learning effect." *Annals of Operations Research* 211 (1): 481–490.

Sun, X., and X. N. Geng. 2019. "Single-machine scheduling with deteriorating effects and machine maintenance." *International Journal of Production Research* 57 (10): 3186–3199.

Sun, X., X. N. Geng, J. B. Wang, and F. Liu. 2018. "Convex resource allocation scheduling in the no-wait flowshop with common flow allowance and learning effect." *International Journal of Production Research* 1–19.

Tian, Y., M. Xu, C. Jiang, J. B. Wang, and X. Y. Wang. 2018. "No-wait resource allocation flowshop scheduling with learning effect under limited cost availability." *The Computer Journal* 62 (1): 90–96.

Toksarı, M. D., and E. Güner. 2008. "Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach." *The International Journal of Advanced Manufacturing Technology* 38 (7-8): 801–808.

Toksarı, M. D., and E. Güner. 2009. "Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration." *Computers & Operations Research* 36 (8): 2394–2417.

Toksarı, M. D., and E. Güner. 2010. "Parallel machine scheduling problem to minimize the earliness/tardiness costs with learning effect and deteriorating jobs." *Journal of Intelligent Manufacturing* 21 (6): 843–851.

Vahedi Nouri, B., P. Fattahi, and R. Ramezanian. 2013. "Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities." *International Journal of Production Research* 51 (12): 3501–3515.

Wang, D., M. Z. Wang, and J. B. Wang. 2010. "Single-machine scheduling with learning effect and resource-dependent processing times." *Computers & Industrial Engineering* 59 (3): 458–462.

Wang, H., M. Huang, and J. Wang. 2019. "An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs." *Journal of Intelligent Manufacturing* 30 (7): 1–10.

Wang, J. B. 2007. "Single-machine scheduling problems with the effects of learning and deterioration." *Omega* 35 (4): 397–402.

Wang, J. B. 2008. "Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect." *Computers & Industrial Engineering* 55 (3): 584–591.

Wang, J. B., L. Lin, and F. Shan. 2008. "Single-machine group scheduling problems with deteriorating jobs." *The International Journal of Advanced Manufacturing Technology* 39 (7-8): 808–812.

Wang, J. B., L. H. Sun, and L. Y. Sun. 2010. "Scheduling jobs with an exponential sum-of-actual-processing-time-based learning effect." *Computers & Mathematics with Applications* 60 (9): 2673–2678.

Wang, J. B., and J. J. Wang. 2015. "Single-machine scheduling problems with precedence constraints and simple linear deterioration." *Applied Mathematical Modelling* 39 (3): 1172–1182.

Wang, J. J., and Y. J. Liu. 2014. "Single-machine bicriterion group scheduling with deteriorating setup times and job processing times." *Applied Mathematics and Computation* 242: 309–314.

Wang, X., and T. E. Cheng. 2005. "Single machine scheduling with resource dependent release times and processing times." *European Journal of Operational Research* 162 (3): 727–739.

Wang, X. Y., and J. J. Wang. 2014. "Scheduling deteriorating jobs with a learning effect on unrelated parallel machines." *Applied Mathematical Modelling* 38 (21-22): 5231–5238.

Wang, X. Y., Z. Zhou, X. Zhang, P. Ji, and J. B. Wang. 2013. "Several flow shop scheduling problems with truncated position-based learning effect." *Computers & Operations Research* 40 (12): 2906–2929.

Wei, C. M., J. B. Wang, and P. Ji. 2012. "Single-machine scheduling with time-and-resource-dependent processing times." *Applied Mathematical Modelling* 36 (2): 792–798.

Woo, Y. B., and B. S. Kim. 2018. "Matheuristic approaches for parallel machine scheduling problem with time-dependent deterioration and multiple rate-modifying activities." *Computers & Operations Research* 95: 97–112.

Wu, C. C., J. Y. Chen, W. C. Lin, K. Lai, S. C. Liu, and P. W. Yu. 2018. "A two-stage three-machine assembly flow shop scheduling with learning consideration to minimize the flowtime by six hybrids of particle swarm optimization." *Swarm and evolutionary computation* 41: 97–110.

Wu, C. C., P. H. Hsu, and K. Lai. 2011. "Simulated-annealing heuristics for the single-machine scheduling problem with learning and unequal job release times." *Journal of Manufacturing Systems* 30 (1): 54–62.

Wu, C. C., and W. C. Lee. 2008. "Single-machine scheduling problems with a learning effect." *Applied Mathematical Modelling* 32 (7): 1191–1197.

Wu, W. H. 2014. "Solving a two-agent single-machine learning scheduling problem." *International Journal of Computer Integrated Manufacturing* 27 (1): 20–35.

Xin, X., Q. Jiang, C. Li, S. Li, and K. Chen. 2021. "Permutation flow shop energy-efficient scheduling with a position-based learning effect." *International Journal of Production Research* Forthcoming Available at: https://doi.org/10.1080/00207543.2021.2008041.

Xu, D., Y. Yin, and H. Li. 2010. "Comments on "A note on minimizing maximum lateness in an m-machine scheduling problem with a learning effect"." *Applied mathematics and computation* 217 (2): 939–943.

Yang, D. L., and W. H. Kuo. 2009. "A single-machine scheduling problem with learning effects in intermittent batch production." *Computers & Industrial Engineering* 57 (3): 762–765.

Yang, S. J. 2013. "Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time." *Applied Mathematical Modelling* 37 (5): 2995–3005.

Yang, S. J., H. T. Lee, and J. Y. Guo. 2013. "Multiple common due dates assignment and scheduling problems with resource allocation and general position-dependent deterioration effect." *The International Journal of Advanced Manufacturing Technology* 67 (1-4): 181–188.

Yin, N., J. B. Wang, D. Wang, L. Y. Wang, and X. Y. Wang. 2010. "Deteriorating jobs and learning effects on a single-machine scheduling with past-sequence-dependent setup times." *The International Journal of Advanced Manufacturing Technology* 46 (5-8): 707–714.

Yin, Y., C. C. Wu, W. H. Wu, and J. C. Chen. 2013. "Single-machine group scheduling with a general learning effect." *European Journal of Industrial Engineering* 7 (3): 350–369.

Yusriski, R, TMAA Samadhi, AH Halim, et al. 2016. "An integer batch scheduling model for a single machine with simultaneous learning and deterioration effects to minimize total actual flow time." In *IOP conference series: materials science and engineering*, Vol. 114, 012073. IOP Publishing.

Yusriski, R, TMAA Samadhi, AH Halim, et al. 2018. "An integer batch scheduling model considering learning, forgetting, and deterioration effects for a single machine to minimize total inventory holding cost." In *IOP Conference Series: Materials Science and Engineering*, Vol. 319, 012038. IOP Publishing.

Zhang, X., W. H. Wu, W. C. Lin, and C. C. Wu. 2018. "Machine scheduling problems under deteriorating effects and deteriorating rate-modifying activities." *Journal of the operational research society* 69 (3): 439–438.

Zhao, C., and H. Tang. 2012. "Two-machine flow shop scheduling with deteriorating jobs and chain precedence constraints." *International Journal of Production Economics* 136 (1): 131–136.

Zhu, Z., L. Sun, F. Chu, and M. Liu. 2011. "Single-machine group scheduling with resource allocation and learning effect." *Computers & Industrial Engineering* 60 (1): 148–157.

Zou, J. 2014. "Single machine scheduling problems with delivery times under simple linear deterioration." *Applied and Computational Mathematics* 3 (3): 85.