Supplementary material for Mosheiov and Pruwer, "On the minmax Common-Due-Date problem: extensions to position-dependent processing times, job-rejection, learning effect, uniform machines and flowshops", Engineering Optimization, 2020.

***Minmax CON Algorithms*:**

Algorithm $\boldsymbol{Minmax_CON_p_{jr}}$:

Input: $p_{jr},\ j,r = 1,\dots,n;\ \alpha,\beta,\gamma$.

Main:

$optcost = \infty$.

For $f = 1,\dots,n$, solve $AP(f)$ to obtain the job sequence q, C_{min} and C_{max}.

If $\alpha \le \gamma$ (*Case* 1):

$d = d_1 = \beta C_{max}/(\beta+\gamma),\ cost = \beta\gamma C_{max}/(\beta+\gamma)$.

If $\alpha > \gamma$ and $\beta C_{max}/(\beta+\gamma) \le (\beta C_{max} + \alpha C_{min})/(\beta+\alpha)$ (*Case* 2):

$d = d_1 = \beta C_{max}/(\beta+\gamma),\ cost = \beta\gamma C_{max}/(\beta+\gamma)$.

If $\alpha > \gamma$ and $\beta C_{max}/(\beta+\gamma) > (\beta C_{max} + \alpha C_{min})/(\beta+\alpha)$ (*Case* 3):

$d = d_2 = (\beta C_{max} + \alpha C_{min})/(\beta+\alpha),\ \ cost = \beta\alpha(C_{max} - C_{min})/(\beta+\alpha)$.

If $cost < optcost$ then: $q^* = q;\ C^*_{max} = C_{max};\ C^*_{max} = C_{max}\ d^* = d;\ optcost = cost$.

Running time: The total running time required for solving $AP(f)$ is clearly $O(n^3)$. $AP(f)$ is solved n times. Hence the total running time of the algorithm is $O(n^4)$.

Algorithm $\boldsymbol{Minmax_CON_Rejection}$:

Input: $p_j, j = 1,\dots,n;\ e_j\ \ j = 1,\dots,n;\ \alpha,\beta,\gamma$.

Main:

$optcost = \sum_{j=1}^{n} e_j$. // The cost when all jobs are rejected //

For $f = 1,\dots,n,\ n_p = 1,\dots,n$:

If $\alpha \le \gamma$ (*Case* 1)

Solve $AP(f, n_P, Case1)$ to obtain the job sequence q, C_{min} and C_{max}.

If $cost < optcost$ then:

$q^* = q;\ C^*_{min} = C_{min};\ C^*_{max} = C_{max};\ d^* = d_1;\ optcost = cost$.

Else ($\alpha > \gamma$) (*Case* 2) // Assume that Case 2 is relevant //

Solve $AP(f, n_P, Case2)$ to obtain the job sequence q, C_{min} and C_{max}.

Check intersection points:

If $(d_1 = \frac{\beta C_{max}}{\beta+\gamma} \leq \frac{\beta C_{max}+\alpha C_{min}}{\beta+\alpha} = d_2)$ // Case 2 is relevant //

If $cost < optcost$ then:

$q^* = q;\ \ C^*_{min} = C_{min};\ \ C^*_{max} = C_{max};\ \ d^* = d_1;\ \ optcost = cost.$

Else ($Case$ 3)

Solve $AP(f, n_P, Case3)$ to obtain the job sequence q, C_{min} and C_{max}.

If $cost < optcost$ then:

$q^* = q;\ \ C^*_{min} = C_{min};\ \ C^*_{max} = C_{max};\ \ d^* = d_2;\ \ optcost = cost.$

Running time: The running time required for solving $P(f, n_P, Case1), AP(f, n_P, Case2)$ and $AP(f, n_P, Case3)$ is clearly $O(n^3)$. It should be solved for all f and n_P values, i.e. $O(n^2)$ times. Hence the total running time of the algorithm is $O(n^5)$.

Algorithm $\boldsymbol{Minmax_CON_linear_deterioration}$:

Input: $\boldsymbol{a_j, b_j;\ \ j = 1, \dots, n;\ \ \alpha, \beta, \gamma}$.

Main:

$optcost = \infty$.

Sort the jobs in a non-decreasing order of a_j/b_j. Renumber the jobs accordingly.

For $f = 1, \dots, n$, assign job f to the first position,

to obtain the sequence: $(f, 1,2, \dots, f-1, f+1, f+2, \dots, n)$.

$C_{min} = a_f$.

$C_{max} = \sum_{i=1}^{n} a_i \prod_{j=i+1}^{n}(1 + b_j)$

If $\alpha \leq \gamma$ (*Case* 1):

$d = d_1 = \beta C_{max}/(\beta + \gamma),\ \ cost = \beta\gamma C_{max}/(\beta + \gamma).$

If $\alpha > \gamma$ and $\beta C_{max}/(\beta + \gamma) \leq (\beta C_{max} + \alpha C_{min})/(\beta + \alpha)$ (*Case* 2):

$d = d_1 = \beta C_{max}/(\beta + \gamma),\ \ cost = \beta\gamma C_{max}/(\beta + \gamma).$

If $\alpha > \gamma$ and $\beta C_{max}/(\beta + \gamma) > (\beta C_{max} + \alpha C_{min})/(\beta + \alpha)$ (*Case* 3):

$d = d_2 = (\beta C_{max} + \alpha C_{min})/(\beta + \alpha),\ \ \ cost = \beta\alpha(C_{max} - C_{min})/(\beta + \alpha).$

If $cost < optcost$ then: $q^* = q;\ \ C^*_{max} = C_{max};\ \ C^*_{max} = C_{max}\ \ d^* = d;\ \ optcost = cost.$

Running time: The running time required for sorting the jobs is $O(nlogn)$. Computing C_{max} in each of the n iteration is done in $O(n)$ time. Hence the total running time of the algorithm is $O(n^2)$.