Research on Some Inverse Scheduling Problems

Hongtruong Pham\textsuperscript{1}, Thilinh Pham\textsuperscript{2}, Dinhchuc Tran\textsuperscript{3}

\textsuperscript{1,2}Thai Nguyen University of Economics and Business Administration, ThaiNguyen, Vietnam

Corresponding Author: Thilinh Pham

ABSTRACT: In this paper, we summarize some results about the inverse scheduling problem of the total weighted completion time problem on single machines and the inverse scheduling problem of the total completion time objective on parallel machines in which the processing times are minimally adjusted, so that the given schedule is satisfying the necessary conditions and sufficient conditions for the scheduling problem and becomes optimal with respect to the adjusted processing times or weights. We have obtained the mathematical programming formulations for this inverse scheduling problem with different norms and provided efficient solution algorithms.

KEYWORDS: Scheduling, Inverse Problem, Completion Time, Parallel Machine, Single Machine.

I. INTRODUCTION

In the recent past and in the recent year, many authors studied the inverse optimisation in scheduling refers to the situation. There are a large number of articles on inverse optimisation in scheduling refers to the situation. Lun and Cariou [4]; Lun et al. [5], derived the processing times or the weights of the jobs can be adjusted depending on the deployment of such resources as quay cranes to load/discharge containers on/from the ship and trucks to transport containers between the quayside and the container yard, so that the scheduling criterion (e.g., the total weighted completion time, which is summary measure of the waiting times of the jobs or the inventory level in the shop) is minimised with respect to the adjusted processing times or weights. However, the resulting value of the scheduling criterion may be higher than the original value of the scheduling criterion, which is undesirable. Therefore we impose in this paper the constraint that the resulting value of the scheduling criterion based on the adjusted parameters should not be greater than the value of the scheduling criterion based on the original parameters.

II. THEINVERSE SCHEDULING PROBLEM OF THE TOTAL COMPLETION TIME OBJECTIVE ON IDENTICAL PARALLEL MACHINES

In the forward scheduling problem, consider an arbitrary \( n \)-jobs \( \{ J_1, J_2, \ldots, J_n \} \) should be processed by \( m \) parallel machines \( \{ M_1, M_2, \ldots, M_m \} \). There are no precedence constraints between the jobs. Each job \( J_j \) (\( j = 1, 2, \ldots, n \)) has processing time \( p_j \) (\( j = 1, 2, \ldots, n \)). All jobs are available at time zero. For any schedules, assume that on machine \( M_i \) (\( i = 1, 2, \ldots, m \)), \( n_i \) jobs (\( J_{i,1}, J_{i,2}, \ldots, J_{i,n_i} \)) are...
consecutively processed. So on machine $M_i (i = 1, 2, ..., m)$, the completion time of job $s$ is $C_{i,s}$ and the total completion time will be:

$$
\sum_{s=1}^{n_i} C_{i,s} = \sum_{s=1}^{n_i} \sum_{l=1}^{s} p_{i,l} = \sum_{s=1}^{n_i} s p_{i,n_i - s + 1}.
$$

The total completion time on $m$ machines $\sum_{j=1}^{m} C_j$ will be:

$$
\sum_{j=1}^{m} C_j = \sum_{i=1}^{m} \sum_{s=1}^{n_i} C_{i,s} = \sum_{i=1}^{m} \sum_{s=1}^{n_i} s p_{i,n_i - s + 1}.
$$

As we know it is well-known Hongtruong Truong et al. [2] proved following the result:

As schedule $\sigma = (J_1, J_2, ..., J_n)$ is optimal for problem $Pm \parallel \sum C_j$ if and only if for any given $S_a, S_b (a, b \in \{1, 2, ..., k\}, a < b)$, there are $S_a < S_b$ and $p_i < p_j$ for any $J_i \in S_a, J_j \in S_b$, where $S_1 = \{J_1, J_2, ..., J_h\}$,

$S_2 = \{J_{h+1}, J_{h+2}, ..., J_{h+m}\}$

$S_{k+1} = \{J_{(k-1)m+h+1}, J_{(k-1)m+h+2}, ..., J_{km+h}\}$

In the inverse scheduling problem $Pm \mid INV \mid \sum C_j$, given a feasible schedule $\sigma$ of the scheduling problem $Pm \parallel \sum C_j$, without loss of generality we assumethat $\sigma = (J_1, J_2, ..., J_n)$, then the total completion time on $m$ machines $\sum_{j=1}^{m} C_j$ will be:

$$
\sum_{j=1}^{m} C_j = \sum_{i=1}^{m} \sum_{s=1}^{n_i} s p_{i,n_i - s + 1}
$$

$$
= \sum_{j=(k-1)m+h+1}^{km+h} \sum_{j=1}^{(k-1)m+h} p_j + 2 \sum_{j=(k-1)m+h+1}^{(k+1)m+h} \sum_{j=1}^{h} p_j + ...
$$

$$
+ k \sum_{j=h+1}^{m+h} \sum_{j=1}^{h} p_j + (k+1) \sum_{j=1}^{h} p_j.
$$

where, $n = km + h, k \in \{1, 2, ..., m\}, h \in \{0, 1, 2, ..., m-1\}$.

The problem $Pm \mid INV \mid \sum C_j$ is solved by determining the minimum total adjustable perturbation to
the processing time \( p = (p_1, p_2, \ldots, p_n)^T \) to become \( \overline{p} = (\overline{p}_1, \overline{p}_2, \ldots, \overline{p}_n)^T \), so that the given schedule \( \sigma \) satisfies the necessary and sufficient conditions for optimality of the problem \( P_m \| \sum_{j=1}^{n} C_j \) and becomes optimal with respect to \( \overline{p} = (\overline{p}_1, \overline{p}_2, \ldots, \overline{p}_n)^T \). Thus, we can formulate the scheduling problem

\[
P_m \mid INV \sum_{j=1}^{n} C_j
\]

as a mathematical programming problem:

\[
\min \| \overline{p} - p \|
\]

s.t. \( \overline{p}_j \leq p_j \) for any \( J_i \in S_i \),

\[
J_i \in S_{i+1}, S_i \prec S_{i+1} (l = 1, 2, \ldots, k)
\]

(1)

where \( p_j \) is the new minimally perturbed processing time of job \( j \) \((j = 1, 2, \ldots, n)\).

For above inverse schedule problem, we have different models under three types of norms: \( l_1 \) – norm, \( l_2 \) – norm, \( l_{\infty} \) – norm.

1. The inverse problem \( P_m \mid INV \sum_{j=1}^{n} C_j \) under \( l_2 \) – norm

For \( l_2 \) – norm, the formula (1) can be written as

\[
\min \frac{1}{2} \sum_{j=1}^{n} (\overline{p}_j - p_j)^2
\]

s.t. \( \overline{p}_j \leq p_j \) for any \( J_i \in S_i \),

\[
J_i \in S_{i+1}, S_i \prec S_{i+1} (l = 1, 2, \ldots, k)
\]

(2)

\[
\overline{p}_j \geq 0, (j = 1, 2, \ldots, n).
\]

The problem (2) is equivalent to

\[
\min \ f(\overline{p}) = \frac{1}{2} \overline{p}^T \overline{p} - \overline{p}^T p + \frac{1}{2} p^T p
\]

s.t. \( A \overline{p} \geq 0 \) \( (j = 1, 2, \ldots, n) \).

Where

\[
A = \begin{bmatrix} M \\ N \end{bmatrix} \in \mathbb{R}^{(k-1)m^2 + mh \times n},
\]

\[
M = (a_{1,1}, a_{1,2}, \ldots, a_{1,m}, a_{2,1}, a_{2,2}, \ldots, a_{2,m}, a_{h,1}, \ldots, a_{h,1}, a_{h,2}, \ldots, a_{h,m}) \in \mathbb{R}^{l_{\text{inv}} \times n},
\]

\[
V = (v_{1,1}, v_{1,2}, \ldots, v_{1,m}) \in \mathbb{R}^{(k-1)m \times n},
\]

\[
W = (w_{1,1}, w_{1,2}, \ldots, w_{1,m}) \in \mathbb{R}^{(k-1)m \times n},
\]

\[
X = (x_{1,1}, x_{1,2}, \ldots, x_{1,m}) \in \mathbb{R}^{(k-1)m \times n},
\]

\[
Y = (y_{1,1}, y_{1,2}, \ldots, y_{1,m}) \in \mathbb{R}^{(k-1)m \times n},
\]

\[
Z = (z_{1,1}, z_{1,2}, \ldots, z_{1,m}) \in \mathbb{R}^{(k-1)m \times n},
\]

\[
\mathbb{R} = \{ (r(1), r(2), \ldots, r(n)) | r(i) \geq 0, (i = 1, 2, \ldots, n) \}
\]
\[
N = (b_{h+1, h+1}, \ldots, b_{h+1, h+2m}, \ldots, b_{h+m, h+m+1}, \ldots, b_{h+(k-1)m, h+(k-1)m+1}, \ldots), \quad a_{xy} = (0, \ldots, 0, -\frac{1}{x-y}, 0, \ldots, 0, 1)
\]
\[
x = 1, 2, \ldots, h \text{ and } y = 1, 2, \ldots, m.
\]

Since \( f(\overline{p}) \) is convex function and \( D = \{ p \mid A\overline{p} \geq 0, \overline{p} \geq 0 \} \) is is convex set, the problem (3) is convex quadratic programming. So, its Kuhn-Tucker conditions (4) is the necessary and sufficient conditions for the optimal formula (3).

\[
\begin{align*}
& p - p - A^T \lambda - \mu = 0, \\
& A\overline{p} \geq 0, \\
& \lambda^T (A\overline{p}) = 0, \\
& \mu^T \overline{p} = 0, \\
& \overline{p}, \lambda, \mu \geq 0.
\end{align*}
\]
in which \( \lambda \in R^{(k-1)m^2 + nh+1} \), \( \mu \in R^{(k-1)m^2 + nh+1} \).

By Wolfe algorithm of quadratic programming (D. Goldfarb and A. Idnani [1]), we can easily solve of above Kuhn-Tucker conditions. Thus we can obtain the optimal solution of problem (2).

2. The inverse problem \( P_m \mid INV \sum_{j=1}^{n} C_j \text{ under } l_1 - \text{norm} \)

For \( l_1 - \text{norm} \), the problem (1) can be written as follows:

\[
\min \sum_{j=1}^{n} |p_j - \overline{p_j}|
\]
\[
\text{s.t.} \quad \overline{p_i} \leq \overline{p_j} \quad \text{for any } J_i \in S_i, \\
\quad J_i \in S_{i+1}, S_i < S_{i+1} (i = 1, 2, \ldots, k) \\
\quad \overline{p_j} \geq 0, (j = 1, 2, \ldots, n).
\]

From (5) is a non-linear programming problem. Let...
\[
\begin{align*}
\alpha_j &= \frac{1}{2}[[\bar{p}_j - p_j] + (\bar{p}_j - p_j)] \\
\beta_j &= \frac{1}{2}[[\bar{p}_j - p_j] - (\bar{p}_j - p_j)].
\end{align*}
\] (6)

By (6), we have

\[
\begin{align*}
[p_j - p_j] &= \alpha_j + \beta_j \\
\bar{p}_j &= \alpha_j - \beta_j + p_j \quad (j = 1, 2, ..., n).
\end{align*}
\]

Thus problem (5) is converted to the linear program as follows:

\[
\min \sum_{j=1}^{n} (\alpha_j + \beta_j)
\]

s.t. \[\alpha_i - \beta_i + p_i \leq \alpha_j - \beta_j + p_j \]
for any \( J_i \in S_i, J_j \in S_{i+1} \),

\[ S_i \times S_{i+1} \quad (l = 1, 2, ..., k) \] \quad (7)

\[
\begin{align*}
\alpha_j - \beta_j + p_j &\geq 0 \quad (j = 1, 2, ..., n) \\
\alpha_j, \beta_j &\geq 0 \quad (j = 1, 2, ..., n)
\end{align*}
\]

According to linear programming (7) we can obtain optimal solution \( \alpha_j \) and \( \beta_j \) (\( j = 1, 2, ..., n \)). By \( \bar{p}_j = \alpha_j - \beta_j + p_j \), we find \( \bar{p}_j \) (\( j = 1, 2, ..., n \)).

3. The inverse problem \( Pm \mid INV \mid \sum_{j=1}^{n} C_j \) under \( l_{\infty} \) - norm

For \( l_{\infty} \) - norm, the mathematical program (1) of the inverse scheduling problem is

\[
\min_{1 \leq j \leq n} |\bar{p}_j - p_j|
\]

s.t. \( \bar{p}_i \leq \bar{p}_j \) for any \( J_i \in S_i \),

\[ J_i \in S_{i+1}, S_i \times S_{i+1} \quad (l = 1, 2, ..., k) \]

\[ |\bar{p}_j| \geq 0, \quad (j = 1, 2, ..., n) \]

and is rewritten into

\[
\min \theta
\]

s.t. \( |\bar{p}_j - p_j| \leq \theta \quad (j = 1, 2, ..., n) \)

\[ \bar{p}_i \leq \bar{p}_j \] for any \( J_i \in S_i \),

\[ J_i \in S_{i+1}, S_i \times S_{i+1} \quad (l = 1, 2, ..., k) \]

By similarly transforms

\[ |\bar{p}_j| \geq 0, \quad (j = 1, 2, ..., n) \].
$$\alpha_j = \frac{1}{2} \left[ (\bar{p}_j - p_j) + (\bar{p}_j - p_j) \right] \quad (j = 1, 2, \ldots, n).$$

$$\beta_j = \frac{1}{2} \left[ (\bar{p}_j - p_j) - (\bar{p}_j - p_j) \right] \quad (j = 1, 2, \ldots, n).$$

Problem (9) is converted to the form of linear programming as follows

$$\begin{align*}
\text{min } & \theta \\
\text{s.t. } & \alpha_j + \beta_j \leq \theta \quad (j = 1, 2, \ldots, n) \\
& \alpha_i - \beta_i + p_i \leq \alpha_j - \beta_j + p_j \\
& \text{for any } J_i \in S_i, \\
& J_j \in S_{j+1}, S_j \prec S_{j+1} \quad (l = 1, 2, \ldots, k) \quad (10) \\
& \alpha_j - \beta_j + p_j \geq 0 \quad (j = 1, 2, \ldots, n) \\
& \alpha_j, \beta_j \geq 0 \quad (j = 1, 2, \ldots, n)
\end{align*}$$

Similarly, we can easily solve above linear programming. Thus, we can find $\bar{p}_j$ from the formula

$$\bar{p}_j = \alpha_j - \beta_j + p_j \quad (j = 1, 2, \ldots, n).$$

### III. CONCLUSION

In this paper, we have summarized some research results on the inverse scheduling problem

$$1 \mid \text{INV} \sum_{j=1}^{n} w_j C_j$$

and the inverse scheduling problem

$$\text{Pm} \mid \text{INV} \sum_{j=1}^{n} C_j$$

in which the processing times $p = (p_1, p_2, \ldots, p_n)^T$ are minimally adjusted, so that the given schedule $\sigma$ is satisfying the necessary and sufficient conditions for optimality of the scheduling problem

$$1 \| \sum_{j=1}^{n} C_j$$

and $\text{Pm} \| \sum_{j=1}^{n} C_j$ and becomes optimal with respect to $\bar{p} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n)^T$. We have also produced their mathematical programming formulations and developed efficient solution algorithms, respectively.

### REFERENCES


American Institute of Mathematical Sciences 3.


