Robust Optimization of Performance Scheduling Problem under Accepting Strategy

Hui Ding, Yuqiang Fan*, Weiya Zhong

School of Management, Shanghai University, Shanghai, China
Email: *fyqshu@shu.edu.cn

How to cite this paper: Ding, H., Fan, Y.Q. and Zhong, W.Y. (2018) Robust Optimization of Performance Scheduling Problem under Accepting Strategy. Open Journal of Optimization, 7, 65-78. https://doi.org/10.4236/ojop.2018.74004

1. Introduction

In the planning stage of a performance, many programs will sign up for the performance, and each program bring different revenue and different cost. The program group needs to comprehensively consider the performance revenue and cost, and decides which programs to be accepted. Each program requires multiple performers, and a performer can also participate in multiple programs. If the programs which the same performer participates in are not consecutively arranged, the waiting cost will occur. Therefore, after determining the set of the accepted programs, the reasonable scheduling of these accepted programs to increase the performance revenue is the main concern of the decision maker.

Scholars have done extensive researches on program performance scheduling and the similar film production scheduling problem. Cheng et al. [1] first men-
tioned scheduling problems in the film production process. They assumed that
the shooting duration of each scene is a certain value, considering the actor
waiting cost problem during film shooting. They proved that the problem is
strong NP-hard, and used the branch and bound algorithm to solve the mini-
cum cost problem. Nordström and Tufekci [2] proposed several hybrid algo-
rithms which use limited pairwise interchange procedure within the simple ge-
etic algorithm framework to solve the problem proposed by Cheng et al. [1].
Their algorithms outperformed in terms of quality of solution and computational
time. Bomsdorf and Derigs [3] presented the movie shooting scheduling problem
and formulated a conceptual model. And they proposed a meta-heuristic algo-
rithm to generate a timeline for film. Stuckey et al. [4] applied a dynamic pro-
gramming to the scheduling problem to minimize the cost of the talent. They
assumed that the actor’s appearance fee is different, and the shooting time of
each scene is a certain value, showed a number of ways to improve the dynamic
programming solution by preprocessing and restricting the search. Wang et al. [5]
generalized a scheduling model by incorporating the performers waiting cost
and operating cost in film shooting. They used the next fit (NF) algorithm and
the first fit decreasing (FFD) algorithm to allocate scenes to work days so as to
provide initial solutions for further improvements. Dynamic programming, ite-
rated local search, and tabu search are adopted to constitute the second-phase
improvement procedures. Qin et al. [6] formulated the talent scheduling prob-
lem as an integer linear programming model and designed an improved branch
and bound method to deal with it. Sakulsom and Tharmmaphornphilas [7] stu-
died a music performance scheduling problem. The objective is to minimize the
total number of days that all performers have to show up, and sequence the music
pieces within each day to minimize the total waiting time of the performers. They
proposed a 2-stage methodology to schedule music pieces, which is a combination
of a cell formation technique and an integer-programming model.

Based on the above research results, it can be concluded that the film shooting
or program rehearsal duration of the predecessors is assumed to be a certain
value. In the actual program performances, due to factors such as staff absence,
equipment failure, performance effects, etc., the duration of each program is
uncertain. Therefore, the results obtained by the deterministic research method
are greatly deviated from the actual situation. Zhen et al. [8] proposed the per-
formance scheduling problem of mixed duration, and expressed the perfo-
rmance time as the sum of the certain performance time and the uncertain ad-
justment time. They assumed that the uncertain adjustment time obeyed the
normal distribution. The objective is to minimize the total waiting cost of the
performers. However, in the actual situations, a large amount of data is often
required to obtain a distribution function of a random parameter. This paper
breaks through this limitation and uses the robust optimization method to ex-
press the uncertain duration as a continuous bounded interval. We only need to
know the upper and lower bound of the program performance duration.
Robust optimization is an effective method to solve the uncertain problem and has been widely used. Wang and Tang [9] proposed a two-stage robust optimization method for interval-type surgical scheduling problem, effectively reducing the adverse effects of service time uncertainty on hospital revenue. Xu et al. [10] built a robust scheduling model for homogeneous parallel machines based on the min-max regret criterion under the condition that only knew the processing time interval. Qiu et al. [11] used the robust optimization method to solve the order policy of the integrated supply chain and the contract coordination policy of the distributed supply chain under the min-max regret value criterion. Zhang et al. [12] built an emergency rescue network based on the scenario of min-max regret value criteria, constructed a robust optimization model, and transformed it into a mixed integer programming model, and they used the scenario relaxation algorithm to solve this model.

This paper considers the uncertain duration program performance scheduling problem under accepting strategy (recorded as P0), the remainder is organized as follows. In Section 2, we use a simple example to describe the program performance scheduling problem and introduce the application of the min-max robust optimization method in this paper. In the case of determining the set of the accepted performance programs, the decision maker’s risk preference coefficient [13] [14] is introduced and a robust performance scheduling model (RPSM) is built for these accepted programs in Section 3. Then we transform the RPSM into a 0 - 1 mixed linear programming model to minimize the performance cost, and based on the algorithm for solving RPSM, the algorithm H of P0 is constructed to determine the accepted or rejected programs and the performance sequence of these accepted programs in Section 4. Finally we use Matlab software to carry out numerical experiments, verify the actual performance of algorithm H, and compare the influence of decision maker’s risk preference on performance cost.

2. Problem Description

In the actual performance, due to limitations of time or layout and so on, decision maker needs to consider program revenue and cost in a comprehensive manner, select a part of the programs from many registration programs to perform, and arrange the performance sequence of these accepted programs. Accepting a program will generate revenue, and rejecting a program will occur penalty cost. After determining the set of the accepted performance programs, multiple performers will participate in the performance. If there is no comprehensive arrangement for a performer who participates in different programs, the performer will generate waiting time and waiting cost. The objective of this paper is to find a solution that maximizes the overall revenue of the program group, including determining the set of the accepted performance programs and scheduling these accepted performance programs. The overall revenue of the program group consists of the following two parts: 1) the revenue value of the
accepted programs and the penalty cost of the rejected programs. 2) the performance cost of the performers at the performance scene, including the appearance fees and the waiting cost.

Let’s look at the following example. A program group receives 6 programs \( \{s_1, s_2, s_3, s_4, s_5, s_6\} \). These programs are performed by 3 performers \( \{a_1, a_2, a_3\} \), and the performers participation list is shown in Table 1 (In this paper, √ indicates that the performer participates in the program, and × indicates that the performer does not participate in the program). The performance duration of the programs \( \{s_1, s_2, s_3, s_4, s_5, s_6\} \) is \([3, 5, 5, 8, 4, 6] \), respectively, the performance revenue from the accepted programs \( \{s_1, s_2, s_3, s_4, s_5, s_6\} \) is \([175, 100, 145, 155, 110, 160] \), respectively, and the penalty cost of rejecting them is \([30, 20, 33, 40, 25, 35] \), respectively.

Assume that the program group decides to accept the programs \( \{s_1, s_3, s_4, s_6\} \) to perform, the performance revenue from these accepted programs \( \{s_1, s_3, s_4, s_6\} \) is 635, and the penalty cost for these rejected programs \( \{s_2, s_5\} \) is 45. After determining the set of the accepted programs, in order to improve efficiency and reduce cost, we assume that performers show up on time before the first program they play starts, leave immediately after the last program they play finishes. In this paper, the robust optimization method is used to arrange the sequence of these accepted programs \( \{s_1, s_3, s_4, s_6\} \), which is based on the principle of “min-max” [15]. “Min-max” means that after formulating a program performance scheduling scheme, due to the uncertainty of the performance duration, the overall revenue of the program group may have multiple possibilities. We calculate the maximum performance cost for each possible scheduling scheme, and then select the least one in these maximum costs. Table 2 lists the two scheduling schemes (the blank space in the table indicates that the performer has finished performance and left the scene). \( t_1, t_2, t_3 \) and \( t_4 \) indicate the time series. In the scheme 1, \( a_1 \) is idle in \( t_3 \), waiting time is the performance duration of \( s_4 \). \( a_2 \) and \( a_3 \) are idle in \( t_2 \), waiting time is the performance duration of \( s_3 \). The

| Table 1. Performer participation list. |
|---|
| \( s_1 \) | \( s_2 \) | \( s_3 \) | \( s_4 \) | \( s_5 \) | \( s_6 \) |
| \( a_1 \) | √ | × | √ | × | √ |
| \( a_2 \) | × | √ | × | √ | × |
| \( a_3 \) | √ | × | × | √ | × |

| Table 2. Program performance schemes. |
|---|
| scheme 1 | \( t_1 \) | \( t_2 \) | \( t_3 \) | \( t_4 \) | scheme 2 | \( t_1 \) | \( t_2 \) | \( t_3 \) | \( t_4 \) |
| \( s_1 \) | √ | √ | × | √ | \( a_1 \) | √ | × | √ | √ |
| \( s_2 \) | × | √ | √ | | \( a_2 \) | √ | √ | | |
| \( s_3 \) | × | × | √ | | \( a_3 \) | √ | √ | | |
| \( s_4 \) | | | | | | | | | |
maximum performance cost under scheme 1 is calculated as $c_1$. In the scheme 2, only $a_i$ is idle at $t_1$, and the waiting time is the performance duration of $s_0$, and the maximum performance cost is calculated as $c_2$. Our robust optimization method is to calculate the maximum performance cost of all feasible solutions, then pick the solution with the least cost in the maximum performance cost to determine the performance sequence of these accepted programs. Based on the above, the overall revenue of the program group is obtained by comprehensively considering the revenue of the accepted programs and the penalty cost of the rejected programs.

3. Performance Scheduling Model

3.1. Model Hypothesis

1) $n$ performers $A = \{a_1, a_2, \cdots, a_n\}$ participate in $k$ programs. If the performers $a_i$ participates in the program $s_j$, defining $w_{ij}$ to be 1, 0 otherwise, $\forall i \in \{1, 2, \cdots, n\}, \forall j \in \{1, 2, \cdots, k\}$. The program group needs to select several programs from $k$ registration programs to perform. Suppose that the number of accepted programs is $m$ ($m$ is part of our decision). $T = \{1, 2, \cdots, m\}$ represents the set of time series, arranged from small to large, each program occupies a time series.

2) The duration of program $q_j$ is an uncertain number, $q_j \in [l_{q_j}, u_{q_j}]$, $l_{q_j}$, $u_{q_j}$ value is known.

3) Unit time waiting cost of performer $a_i$ is $l_{c_i}$, unit time appearance fee of performer $a_i$ is $u_{c_i}$, $\forall i \in \{1, 2, \cdots, n\}$.

4) Performer $a_i$ shows up on time before the first program he play starts, leave immediately after the last program he play finishes.

5) The revenue of the accepted program is $p_j$, the penalty cost of the rejected program is $f_j$, $\forall j \in \{1, 2, \cdots, k\}$. The objective function of this paper is the revenue of accepted programs minus the penalty cost of rejected programs, and then minus the performance cost of these accepted programs. The performance cost of the accepted programs is the appearance fees and waiting cost of the performers at the scene.

3.2. Robust Performance Scheduling Model of the Accepted Program (RPSM)

After determining the set of the accepted performance program, it is important to properly schedule these accepted programs and reduce the performance cost so that the program group can achieve a better overall revenue. Without loss of generality, let’s assume that the accepted programs are $\{s_1, s_2, \cdots, s_m\}$, the decision variables are as follows:

- $x_{jt} = 1$ if program $s_j$ performs in time series $t$, 0 otherwise.
- $y_{it} = 1$ if performer $a_i$ performs in time series $t$, 0 otherwise.
- $a_{jt} = 1$ if performer $a_i$ arrives at the scene before time series $t$ (including $t$), 0 otherwise.
\( l_a = 1 \) if performer \( a_i \) leaves the scene after time series \( t \) (including \( t \)), 0 otherwise.

\( d_a = 1 \) if performer \( a_i \) waits at the scene in time series \( t \), 0 otherwise.

Let \( \theta_j = \frac{q_j - lq_j}{uq_j - lq_j} \) be the degree which the performance duration \( q_j \) of program \( s_j \) deviates from the lower bound \( lq_j \), \( \theta_j \in [0,1] \), \( \forall j \in \{1,2,\cdots,m\} \).

In this paper, the idea of Bertsimas and Sim [13] [14] is used to control the conservative degree of robust optimal scheduling model, and the risk preference coefficient \( \mu \) of decision maker is introduced, \( \sum_{j=1}^{m} \theta_j = \frac{\sum_{j=1}^{m} q_j - lq_j}{uq_j - lq_j} \leq \mu \), \( 0 \leq \mu \leq m \), indicating that up to \( \mu \) programs which duration reaches the upper bound at the same time. The \( \mu \) value is given by the decision maker in advance, and the more conservative the decision maker is, the larger the \( \mu \) value is. Therefore, the uncertain set of the program duration \( q_j \) is expressed as:

\[
\left\{ q_j \mid lq_j \leq q_j \leq uq_j, \sum_{j=1}^{m} \frac{q_j - lq_j}{uq_j - lq_j} \leq \mu, \forall j \in \{1,2,\cdots,m\} \right\}
\]  

We build the following min-max robust performance scheduling model for the accepted programs (RPSM):

\[
\min \left\{ \max \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} c_i \cdot d_{ij} + \sum_{j=1}^{m} q_j \cdot x_{ji} + \sum_{i=1}^{n} \sum_{j=1}^{m} w_i \cdot y_{ji} \right\} \right\}
\]  

s.t.

\[ \sum_{j=1}^{m} x_{ji} = 1, \ \forall j \in \{1,2,\cdots,m\} \]  

\[ \sum_{j=1}^{m} y_{ji} = 1, \ \forall t \in T \]  

\[ y_a - \sum_{j=1}^{m} x_{ji} \cdot w_{ij} = 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ a_a + l_a - h_a \leq 1, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ a_{i-1} - a_i \leq 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in \{1,2,\cdots,m-1\} \]  

\[ l_{i-1} - l_i \leq 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in \{1,2,\cdots,m-1\} \]  

\[ -a_a - l_a \leq -1, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ y_a - a_a \leq 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ y_a - l_a \leq 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ h_a - d_a - y_a \leq 0, \ \forall i \in \{1,2,\cdots,n\}, \ \forall t \in T \]  

\[ lq_j \leq q_j \leq uq_j, \ \forall j \in \{1,2,\cdots,m\} \]  

\[ \sum_{j=1}^{m} \frac{q_j - lq_j}{uq_j - lq_j} \leq \mu \]
Equation (2) is the minimum performance cost in the worst case of the uncertain set, including the waiting cost and appearance fees of the performers at the scene. Equation (3) and Equation (4) force to schedule only one program at one time series and assign each program to only one time series. Equations (5)-(9) indicate the presence of performers in each time series. Equation (10) and Equation (11) indicate that each performer shows up on time before the first program he play starts, leave immediately after the last program he play finishes. Equation (12) determines if the performer is in a wait state at a time series. Equation (13) and Equation (14) constitute an uncertain set of program performance duration. Equation (15) indicates that the decision variables are a 0 - 1 variable.

4. Solve Model

This paper studies the uncertain duration program performance scheduling problem under accepting strategy (P0). Firstly, we decide to accept some programs from the registration programs; secondly, we build a robust performance scheduling model for the accepted programs (RPSM) to get the program performance sequence and performance cost; Finally, based on the above, we consider the revenue of the accepted programs and the penalty cost of the rejected programs comprehensively, and determine a feasible scheduling scheme for P0 to maximize the performance revenue.

4.1. Solve the Robust Performance Scheduling Model of the Accepted Programs (RPSM)

We observe the Equations (2)-(15) of RPSM, finding that only the objective function (2), Equation (13) and Equation (14) are affected by the uncertainty of duration $q_j$. Therefore, RPSM can be expressed as a two-stage robust optimization model. The decision variables of the first stage are $x_{ji}, y_{ji}, a_{ji}, l_{ji}, d_{ji}, \forall i \in \{1,2,\cdots,n\}, \forall j \in \{1,2,\cdots,m\}, \forall t \in T$ (15)

$$x_{ji}, y_{ji}, a_{ji}, l_{ji}, d_{ji} \in \{0,1\}, \forall i \in \{1,2,\cdots,n\}, \forall j \in \{1,2,\cdots,m\}, \forall t \in T$$ (15)

The constraints are Equations (3)-(12) and Equation (15).

And:

$$\min F\left(\left\{q_j\right\}_{j=1,2,\cdots,m}\right)$$ (16)

The constraints are Equations (3)-(12) and Equation (15).

And:

$$F\left(\left\{q_j\right\}_{j=1,2,\cdots,m}\right) = \max \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} l_{ji} \cdot d_{ji} \cdot \sum_{j=1}^{m} q_j \cdot x_{ji} + \sum_{i=1}^{n} u_{ci} \sum_{j=1}^{m} q_j \cdot w_{ij} \right\}$$ (17)

s.t.

$$l_{qj} \leq q_j \leq u_{qj}, \forall j \in \{1,2,\cdots,m\}$$ (18)

$$\sum_{j=1}^{m} \frac{q_j - l_{qj}}{u_{qj} - l_{qj}} \leq \mu$$ (19)
Noticing that (17)-(19) is a linear programming problem for \( q_j \), which is equivalent to the following form:

\[
\max \left\{ \sum_{j=1}^{m} q_j \cdot \left( \sum_{i=1}^{n} \sum_{t=1}^{T} l_{ct} \cdot d_{it} \cdot x_{jt} + \sum_{i=1}^{n} u_{ct} \cdot w_{ij} \right) \right\} 
\]

\[
\text{s.t.} \\
-q_j \leq -l_{qj}, \quad \forall j \in \{1, 2, \ldots, m\} \\
q_j \leq u_{qj}, \quad \forall j \in \{1, 2, \ldots, m\} \\
\sum_{j=1}^{m} \frac{q_j}{u_{qj} - l_{qj}} \leq \mu + \sum_{j=1}^{m} \frac{l_{qj}}{u_{qj} - l_{qj}} 
\]

Since the feasible domain of the linear programming problem is bounded, according to the strong dual theory \([16]\), it can be known that the original maximization problem can be equivalent to the minimization problem. We define \( \rho_j \) as the dual variable of \( q_j \) in Equations (21)-(23), \( \forall j \in \{1, 2, \ldots, 2m+1\} \). Then the dual programming problem of (20)-(23) is as the following:

\[
\min \sum_{j=1}^{m} \left( -l_{qj} \right) \cdot \rho_j + \sum_{j=m+1}^{2m} u_{qj} \cdot \rho_j + \rho_{2m+1} \cdot \left( \mu + \sum_{j=1}^{m} \frac{l_{qj}}{u_{qj} - l_{qj}} \right) 
\]

\[
\text{s.t.} \\
-\rho_j + \rho_{m+j} + \frac{1}{u_{qj} - l_{qj}} \cdot \rho_{2m+1} = \sum_{i=1}^{n} \sum_{t=1}^{T} l_{ct} \cdot d_{it} \cdot x_{it} + \sum_{i=1}^{n} w_{ij} \cdot u_{ct} \cdot x_{it}, \quad \forall j \in \{1, 2, \ldots, m\} \\
\rho_j \geq 0, \quad \forall j \in \{1, 2, \ldots, 2m+1\} 
\]

Bringing the Equations (24)-(26) to the RPSM to get the equivalent model RPSM1:

\[
\min \sum_{j=1}^{m} \left( -l_{qj} \right) \cdot \rho_j + \sum_{j=m+1}^{n} u_{qj} \cdot \rho_j + \rho_{2m+1} \cdot \left( \mu + \sum_{j=1}^{m} \frac{l_{qj}}{u_{qj} - l_{qj}} \right) 
\]

\[
\text{s.t.} \\
\sum_{i=1}^{n} x_{it} = 1, \quad \forall i \in \{1, 2, \ldots, m\} \\
\sum_{j=1}^{m} x_{jt} = 1, \quad \forall t \in T \\
y_{it} - \sum_{j=1}^{m} x_{jt} \cdot w_{ij} = 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T \\
a_{it} + l_{it} - h_{it} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T \\
a_{it} - a_{it+1} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in \{1, 2, \ldots, m-1\} \\
l_{it+1} - l_{it} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in \{1, 2, \ldots, m-1\} \\
-a_{it} - l_{it} \leq -1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T \\
y_{it} - a_{it} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T 
\]
\[
y_a - l_a \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(36)

\[
h_i - d_i - y_a \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(37)

\[-\rho_j + \rho_{\text{com}} + \frac{1}{u_{aq_j} - l_{aq_j}} \cdot \rho_2 = \sum_{i=1}^n \sum_{t=1}^m c_{ij} \cdot d_{ij} - \sum_{j=1}^n w_{ij} \cdot u_{ci}, \quad \forall j \in \{1, 2, \ldots, m\}\]  
(38)

\[
\rho_j \geq 0, \quad \forall j \in \{1, 2, \ldots, 2m + 1\}
\]  
(39)

\[
x_{ij}, y_a, h_a, l_a, d_a \in [0, 1], \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall j \in \{1, 2, \ldots, m\}, \quad \forall t \in T
\]  
(40)

It is observed that the Equation (38) contains nonlinear part \(\sum_{j=1}^n \sum_{t=1}^m c_{ij} \cdot d_{ij} \cdot x_{ij}\).

In order to convert the nonlinear constraints into linear constraints, we introduce variable \(\Delta_{ij}\), let \(\Delta_{ij} \geq d_{ij} - x_{ij} - 1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall j \in \{1, 2, \ldots, m\}, \quad \forall t \in T\). Then we can obtain the following 0-1 mixed linear programming model RPSM2:

\[
\min z_j = \frac{1}{\mu} \left( -l_{aq_j} \right) \cdot \rho_j + \sum_{j=1}^n u_{aq_j} \cdot \rho_j + \rho_{\text{com}} \left( \mu + \sum_{j=1}^n \frac{l_{aq_j} - l_{aq_j}}{u_{aq_j} - l_{aq_j}} \right)
\]  
(41)

s.t.

\[
\sum_{j=1}^n x_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, m\}
\]  
(42)

\[
\sum_{j=1}^n x_{ij} = 1, \quad \forall t \in T
\]  
(43)

\[
y_{ij} - \sum_{j=1}^n x_{ij} \cdot w_{ij} = 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(44)

\[
a_{ij} + l_{ij} - h_{ij} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(45)

\[
a_{ij} - a_{ij} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(46)

\[
l_{ij} - l_{ij} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(47)

\[
-a_{ij} - l_{ij} \leq -1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(48)

\[
y_{ij} - a_{ij} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(49)

\[
y_{ij} - l_{ij} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(50)

\[
h_{ij} - d_{ij} - y_{ij} \leq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall t \in T
\]  
(51)

\[-\rho_j + \rho_{\text{com}} + \frac{1}{u_{aq_j} - l_{aq_j}} \cdot \rho_2 = \sum_{i=1}^n \sum_{t=1}^m c_{ij} \cdot d_{ij} - \sum_{j=1}^n w_{ij} \cdot u_{ci}, \quad \forall j \in \{1, 2, \ldots, m\}\]  
(52)

\[
x_{ij} + d_{ij} - \Delta_{ij} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall j \in \{1, 2, \ldots, m\}, \quad \forall t \in T
\]  
(53)

\[
\rho_j \geq 0, \quad \forall j \in \{1, 2, \ldots, 2m + 1\}
\]  
(54)

\[
x_{ij}, y_{ij}, h_{ij}, l_{ij}, d_{ij}, \Delta_{ij} \in [0, 1], \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall j \in \{1, 2, \ldots, m\}, \quad \forall t \in T
\]  
(55)

It is observed that the difference between the above two models is the Equation (38) in RPSM1 and the Equation (52), Equation (53) in RPSM2. However, RPSM1 and RPSM2 have the same optimal solution and objective function value.
The reason is the coefficients symbol of \( j \rho \), \( m_j \rho + 21 m \rho + \) in Equation (41) and Equation (52) are the same (positive or negative simultaneously). So the change of the variable value in Equation (52) will have the same effect in Equation (41). From Equation (53), it can be known that when at least one of \( d_i \) and \( x_j \) is 0, \( \Delta_i' \) can be 1 or 0. However, in the process of finding the optimal solution, \( \Delta_i' \) should be 0 or 1 as much as possible, because if \( \Delta_i' = 1 \), then in the Equation (52), \( \rho_j \) decreases or \( m_j \rho + 21 m \rho + \) increases, in the corresponding Equation (41), \( \rho_j \) decreases or \( m_j \rho + 21 m \rho + \) increases, the objective value increases (when other variables remain unchanged). The performance of this solution is worse than the solution corresponding to \( \Delta_i' = 0 \). So when at least one of \( d_i \) and \( x_j \) is 0, \( \Delta_i' = 0 \), only when \( d_i = x_j = 1 \), \( \Delta_i' = 1 \). In summary, variable \( \Delta_i' \) is introduced to transform the nonlinear programming RPSM1 into linear programming RPSM2 successfully.

### 4.2. Performance Scheduling Algorithm H

We denote \( \overline{p_j} = \frac{p_j}{lq_j + wq_j} \), and sort the registration programs from large to small according to \( \overline{p_j} \) value. Without loss of generality, we assume that the program sequence is \( \{s_1, s_2, \cdots, s_{\tau}\} \). The algorithm H for solving P0 is described as follows:

**Algorithm H:**

**Step 1.** Select a set of program number \( \{m_1, m_2, \cdots, m_{\tau}\} \) that may be accepted, \( 1 \leq m_1, m_2, \cdots, m_{\tau} \leq k \). \( \tau = 1 \).

**Step 2.** The number of the accepted programs is \( \tau \), the set of performance program is \( \{s_1, s_2, \cdots, s_{\tau}\} \).

**Step 2.1.** Build a robust performance scheduling model RPSM of these accepted programs. Through the method in Section 4.1, we can get the performance sequence of \( \{s_1, s_2, \cdots, s_{\tau}\} \) and the performance cost \( z_1 \).

**Step 2.2.** Calculate the difference between the revenue of \( \tau \) accepted programs and the penalty cost of \( (k - \tau) \) rejected programs, namely \( z = \sum_{j=1}^{m_\tau} p_j - \sum_{j=\tau+1}^{k} f_j \), and get the performance revenue \( z(m_\tau) = z_2 - z_1 \).

**Step 3.** \( \tau = \tau + 1 \). If \( \tau \leq x \), go back step 2, otherwise go step 4.

**Step 4.** Denote \( \overline{m} = \arg \max \{z(m_1), z(m_2), \cdots, z(m_{\tau})\} \), select program set \( \{s_1, s_2, \cdots, s_{\overline{m}}\} \) to perform, get the maximum performance revenue \( z(\overline{m}) \) and the performance sequence.

### 5. Numerical Experiment

We use Matlab 2017b software for numerical experiments. The experiment was carried out under the Windows 10 Professional 64-bit i5-3230M 8GB RAM operation environment. RPSM2 is a 0 - 1 mixed linear programming model, so it can be solved by the Intlinprog function in Matlab 2017b.

Although RPSM2 contains \((n + 1)m^2 + 5mn + 2m + 1\) variables,
\( nm^2 + 8mn + 3m - 2n \) constraints, after observation, we find that the zero element in the coefficient matrix of the constraints is the majority, which is a sparse matrix [17]. Its density is the ratio of non-zero elements to total elements in the matrix, namely
\[
\frac{(2n+4)m^2 + (18n+3)m - 4n}{(n+1)m^2 + 5mn + 2m + 1}\left[ nm^2 + 8mn + 3m - 2n \right].
\]
And as the values of \( m \) and \( n \) increase, the density of the RPSM2 coefficient matrix decreases sharply. For sparse matrices, Matlab only stores non-zero element values and their positions. Therefore, we use the sparse feature of RPSM2 coefficient matrix to reduce the variable storage space of the computer and improve the running speed of the procedure.

The parameter setting of the problem \( P_0 \) is as the following:

1) There are 15 registration programs. These programs require 30 performers to participate in. The relationship between the performers and the programs is a 0 - 1 matrix, defined as \( W = \left( w_{ij} \right)_{30 \times 15} \). The value of each element is generated by Matlab according to the random uniform probability.

2) The duration \( q_{js} \) of program \( s_j \) is a bounded interval value, \( \forall j = \{1, 2, \cdots, 15\} \). The lower bound \( l_{jq} \) of \( q_{js} \) obeys the random uniform distribution between the interval \([3, 5]\), and the upper bound \( u_{jq} \) obeys the random uniform distribution between the interval \([6, 10]\).

3) The unit time waiting cost \( l_{ic} \) obeys the random uniform distribution between the interval \([10, 20]\), and the unit time appearance fee \( u_{ic} \) obeys the random uniform distribution between the interval \([50, 80]\).

4) The revenue \( p_{js} \) of the accepted program \( s_j \) obeys the random uniform distribution between the interval \([5000, 20000]\), and the penalty cost \( f_{js} \) of the rejected program \( s_j \) obeys the random uniform distribution between the interval \([800, 1000]\).

Due to time and layout restrictions, the decision maker decides to accept 8 to 13 programs, namely \( m \in \{8, 9, 10, 11, 12, 13\} \). In order to explain the influence of decision maker’ risk preference on performance cost, \( \mu = 0, \frac{1}{2}, m \) is selected to conduct experiments, which represent that decision maker is extremely preferences, moderate risk preferences and very conservative.

The numerical experiment results are shown in Table 3. The first column is the number of the accepted programs, and the second column is the overall performance revenue \( z \), performance cost \( z_1 \) and procedure running time when the decision maker prefers the risk. Columns 3 and 4 and so on. (A negative value in the table indicates that the performance revenue is negative).

In Table 3, as the number of accepted programs increases, the variables and constraints in the model increase correspondingly. However, our procedure gets the scheduling scheme within the acceptable time, indicating that the algorithm H has a robust practicality in the actual performance scheduling. The horizontal axis of Figure 1 is the \( \mu \) value, indicating risk preference degree of the decision
### Table 3. Numerical experiment results.

| \(m\) | \(\mu = 0\) | \(\mu = \frac{m}{2}\) | \(\mu = m\) |
|---|---|---|---|
| | \(z\) | \(z_1\) | time/s | \(z\) | \(z_1\) | time/s | \(z\) | \(z_1\) | time/s |
| 8 | 73,541 | 44,978 | 54 | 36,863 | 81,656 | 57 | 15,598 | 102,921 | 53 |
| 9 | 77,901 | 54,175 | 70 | 37,835 | 94,241 | 124 | 13,150 | 118,926 | 144 |
| 10 | 81,488 | 61,789 | 73 | 37,340 | 105,937 | 141 | 9158 | 134,119 | 283 |
| 11 | 84,608 | 70,084 | 152 | 37,433 | 117,259 | 194 | 4131 | 150,561 | 230 |
| 12 | 89,063 | 74,499 | 220 | 36,589 | 126,973 | 197 | −651 | 164,213 | 433 |
| 13 | 91,288 | 79,870 | 223 | 29,452 | 141,706 | 350 | −7779 | 178,937 | 437 |

#### Figure 1. Relationship between performance cost and risk preference of decision maker.

The chart shows the relationship between performance cost \(z\) and the risk preference \(\mu\). It can be seen from this figure that in the case of determining the set of the accepted programs, the performance cost increases with the increase of \(\mu\), namely if the decision maker avoids the risk, the performance cost will increase and the performance revenue will reduce. Therefore, decision maker can obtain an ideal performance scheduling solution based on their own risk preference.

### 6. Conclusions

This paper studies a problem of uncertain duration performance scheduling under accepting strategy, the accepted programs will bring in revenue, and the rejected programs will produce corresponding penalty cost. After determining the set of the accepted performance programs, the decision maker’s risk preference coefficient is introduced, and the min-max robust performance scheduling model of these accepted programs is built, and then it is transformed into a 0-1 mixed linear programming model to minimize the performance cost. Based on this, we combined with the revenue of the accepted programs and the penalty cost of the rejected programs, the algorithm H for solving the performance scheduling problem under accepting strategy is proposed, which provides a reference for decision maker to choose the ideal program scheduling scheme.
This article does not limit the sequence of program performance, but in the actual world, the decision maker will arrange a program at the opening or finale. Or depending on the type of program, some programs must not be adjacent. In addition, multi-objective functions can also be studied, such as maximizing overall revenue on the basis of ensuring that the performance cost does not exceed the budget.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Cheng, T.C.E., Diamond, J.E. and Lin, B.M.T. (1993) Optimal Scheduling in Film Production to Minimize Talent Hold Cost. *Journal of Optimization Theory and Applications*, **79**, 479-492. https://doi.org/10.1007/BF00940554
[2] Nordström, A.L. and Tufekci, S. (1994) A Genetic Algorithm for the Talent Scheduling Problem. *Computers & Operations Research*, **21**, 927-940. https://doi.org/10.1016/0305-0548(94)90021-3
[3] Bomsdorf, F. and Derigs, U. (2008) A Model, Heuristic Procedure and Decision Support System for Solving the Movie Shoot Scheduling Problem. *OR Spectrum*, **30**, 751-772. https://doi.org/10.1007/s00291-007-0103-6
[4] Garcia De La Banda, M., Stuckey, P.J. and Chu, G. (2011) Solving Talent Scheduling with Dynamic Programming. *INFORMS Journal on Computing*, **23**, 120-137. https://doi.org/10.1287/ijoc.1090.0378
[5] Wang, S.Y., Chuang, Y.T. and Lin, B.M.T. (2016) Minimizing Talent Cost and Operating Cost in Film Production. *Journal of Industrial and Production Engineering*, **33**, 17-31.
[6] Qin, H., Zhang, Z.Z., Lim, A., *et al.* (2016) An Enhanced Branch-and-Bound Algorithm for the Talent Scheduling Problem. *European Journal of Operational Research*, **250**, 412-426. https://doi.org/10.1016/j.ejor.2015.10.002
[7] Sakulsom, N. and Tharmmaphornphilas, W. (2014) Scheduling a Music Performance Problem with Unequal Music Piece Length. *Computers & Industrial Engineering*, **70**, 20-30. https://doi.org/10.1016/j.cie.2013.12.017
[8] Zhen, L., Liu, B. and Wang, W.C. (2017) Scheduling of Performance with Mixing Piece Length. *Journal of Systems Management*, **26**, 850-856.
[9] Wang, Y. and Tang, J.F. (2016) A Two-Stage Robust Optimization Method for Solving Surgery Scheduling Problem. *Journal of Systems Engineering*, **31**, 431-440.
[10] Xu, X.Q., Cui, W.T., Lin, J. and Qian, Y.J. (2013) Robust Identical Parallel Machines Scheduling Model Based on Min-Max Regret Criterion. *Journal of Systems Engineering*, **28**, 729-737.
[11] Qiu, R.Z. and Huang X.Y. (2011) A Robust Supply Chain Coordination Model Based on Minimax Regret Criterion. *Journal of Systems Management*, **20**, 296-302.
[12] Zhang, L., Chen, T. and Huang, J. (2014) Emergency Network Model and Algorithm Based on Min-Max Regret Robust Optimization. *Chinese Journal of Management Science*, **22**, 131-139.
[13] Bertsimas, D. and Sim, M. (2004) The Price of Robustness. *Operations Research*, **52**, 641-679.
35-53. https://doi.org/10.1287/opre.1030.0065

[14] Bertsimas, D. and Sim, M. (2003) Robust Discrete Optimization and Network Flows. Mathematical Programming, 98, 49-71. https://doi.org/10.1007/s10107-003-0396-4

[15] Charalambous, C. and Conn, A.R. (1978) An Efficient Method to Solve the Minimax Problem Directly. SIAM Journal on Numerical Analysis, 15, 162-187. https://doi.org/10.1137/0715011

[16] Yao, E.Y., He, Y. and Cheng, S.P. (2001) Mathematical Planning and Combination Optimization. Zhejiang University Press, Hangzhou, 31-35.

[17] Liu, H. (2013) MATLAB R2012a Completely Self-Study. Electronic Industry Press, Beijing.