

Research Article

Research on the Scheduling Problem of Movie Scenes

Yulian Liu,¹ Qiuji Sun ,¹ Xiaotian Zhang,^{1,2} and Yiwei Wu ¹

¹School of Management, Shanghai University, Shanghai, China

²Oulu Business School, University of Oulu, Finland

Correspondence should be addressed to Yiwei Wu; wyw0044@shu.edu.cn

Received 28 November 2018; Accepted 10 January 2019; Published 3 February 2019

Guest Editor: Xinchang Wang

Copyright © 2019 Yulian Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The scheduling problem of movie scenes refers to the reasonable ordering of the shooting sequence of the scene, thereby minimizing the total cost of the movie scenes. This paper considers the factors affecting the cost of movie scenes shooting in the real world and constructs an integer linear programming model. A tabu search based method (TSBM) and a particle swarm optimization based method (PSOBM) are designed to solve larger-scale problems. Numerical experiments show that both TSBM and PSOBM are suitable for solving small-scale movie scenes scheduling problem. Moreover, through the comparison experiments between TSBM and PSOBM, it is verified that the TSBM can effectively solve the large-scale movie scenes scheduling problem.

1. Introduction

In the real world, the sequence of movie scenes shoot is not the final sequence. Therefore, we do this research in order to find the optimal sequence. The movie scenes scheduling problem can be described as follows. Let $S = \{s_1, s_2, \dots, s_n\}$ represent a set of n scenes and $A = \{a_1, a_2, \dots, a_m\}$ denote a set of m actors. Let $D = \{d_1, d_2, \dots, d_n\}$ show the duration of n scenes and $P = \{p_1, p_2, \dots, p_m\}$ mean the wage of m actors in one day. The actor shot from the first scene belongs to his last scene. Meanwhile, the actors are in the crew. During this period, the actor will be paid every day. Even in some days, they may not be shot. The movie scenes scheduling problem is how to reduce the transfer cost in consideration of actors with different wages so as to minimize the total cost.

The movie scenes scheduling problem is a multifactor optimization problem. Most literatures study the scheduling problem at the same location, taking no account of the impact of different locations on the sequence of the scene. It is very likely that the transfer cost of the scene location will be generated. In this paper, the location of the movie scenes is considered. In addition, this paper also considers different wage of actors and different scenes shooting duration in order to construct an integer linear programming model. Then, this paper designs a tabu search based method (TSBM) and a particle swarm optimization based method (PSOBM) to solve

large-scale problems, comparing them through a series of numerical experiments.

The remainder of this paper is organized as follows. Section 2 is related works. In Section 3, we build an integer programming model for the movie scenes scheduling problem. Then, we establish a TSBM in Section 4. Section 5 develops a PSOBM to handle the movie scenes scheduling problem. The experiment results and settings are shown in Section 6. In Section 7 we conclude the whole paper.

2. Related Works

During the movie scenes shooting process, many factors affect the order in which movie scenes are shot, such as the cost of the actors, the duration of the scenes shot, and the duration of the actors. In addition, cost control is one of the issues that must be considered. Optimizing the sequence of scenes in the movie scenes shooting process, reducing the waiting time for the actors, and minimizing the total cost of the crew are the overall goals.

The movie scenes scheduling problem was proposed by Cheng et al. [1]. Considering the different daily wage of actors, the shooting duration of each scene was the same value. A joint model was proposed to minimize the waiting time of actors as well as the total minimized costs. And they used the branch and bound algorithm and heuristic algorithm

to solve large-scale problems. Nordström and Tufekci [2] applied hybrid genetic algorithm to solve the problem of movie scenes scheduling based on the research of Cheng et al. [1] and verified that the proposed algorithm is better than the heuristic algorithm of Cheng et al. [1]. Fink and Voß [3] solved the movie scenes scheduling problem as a special application of the general pattern ordering problem and used simulated annealing algorithm and TS to solve this problem. Bomsdorf and Derigs [4] put forward a conceptual model to solve the scheduling problem in the movie scenes shooting process and used a heuristic algorithm to generate a feasible scheme. Finally, a decision support system is proposed and the effectiveness of the decision support system is proved.

Some scholars assumed that each scene shooting duration is different, and different actors have different daily wages. de la Banda et al. [5] described the scheduling problem. In their study, actor's daily wage was different. And scenes may have different durations. They proposed the basic dynamic programming algorithm, through the preprocessing, search limits, to determine the upper and lower bound to improve the algorithm. Preprocessing mainly includes two simplified methods. On the one hand, referring to the same actors in the two scenes, you can combine the two scenes shooting duration and reduce the calculation. On the other hand, if any actor appears in only one scene, then the cost of the actor can be calculated as a fixed cost. Kochetov [6] studied the movie scenes scheduling problem in movie shooting of the column transformation problem that can be reconstructed into a binary matrix and solved it using the improved greedy algorithm and metaheuristic algorithm of iterative local search. Liang et al. [7] proposed a branch and bound algorithm to solve the movie scenes scheduling problem. They used a new lower bound function to delete redundant search nodes. Based on de la Banda et al. [5], Qin et al. [8] proposed an enhanced branch and bound algorithm to solve the movie scenes scheduling problem. They used an integer linear programming model and branch and bound algorithm to solve the problem. By preprocessing, double-ended search determines the lower bound, and cache search states and dominant rules enhance the branch and bound algorithm. The preprocessing is similar to that of de la Banda et al. [5]. Experiments results show that the improved branch and bound algorithm has better performance than the dynamic programming algorithm of de la Banda et al. [5]. Cheng et al. [9] presented two methods to improve the lower bound of the movie scenes scheduling problem. The result can give a partial timetable.

Most of studies consider the cost of the actor, nevertheless, a small number of studies, not only consider the cost of the actor, but also consider the operating costs of crew. Wang et al. [10] studied the minimization of movie costs, including the cost of the actors as well as the cost of operating the crew. They used the next fitting algorithm and the first fitting reduction algorithm to solve this problem.

Scheduling in movie shooting process is essentially a scheduling problem, which can draw lessons from the scheduling problem. There are also many studies that resemble the scheduling problem in movie. For example, the

problem of program rehearsal is optimized through reasonable scheduling, so as to reduce the waiting time and minimize the total cost. Therefore, we can learn from such research ideas and methods. Gregory et al. [11] solved the rehearsal scheduling problem by planning and model checking, setting the daily wage of actors differently, and the rehearsal duration of each program was determined. Sakulsom and Tharmmaphornphilas [12] proposed a two-stage method to solve the problem of rehearsal of music programs. They set the same actor daily wage, and the rehearsal duration of each program was determined.

From the above study, we can see that most scholars consider the cost of actors from the perspective of actors. They consider the cost of actors, including the daily wage of actor which is different, and each scene has the same shooting duration. As well as the daily wage of actor is different, each scene shooting duration for different values. Some scholars not only consider the cost of actors, but also consider the operational costs of the crew. Few scholars consider the transfer cost between scenes. Different movie scenes may be taken at different locations, so there is a high potential for transfer costs between the scene and the scene transition during movie scenes shooting process.

3. Model Formulation

Above all, the movie scenes scheduling problem is a scheduling problem. Zhen [13] studied the scheduling problem of tactical berth allocation problem. The first deterministic model established in the paper considers the certainty of the period. The research in this paper considers the certainty of the shooting duration. The model building process refers to the deterministic model in Zhen [13]. This paper establishes an integer programming model to solve the movie scenes scheduling problem considering the different wage of actors, different shooting duration of actors, and the transfer cost between scenes. The model mainly consists of two parts, one takes the cost of the actors into account, and the other considers the transfer cost between scenes.

Before formulating the model for the movie scenes scheduling problem, we list the notation used in the paper below.

3.1. Notations

Indices and Sets

i, j : Index of scenes.

k : Index of an actor.

m, n : Index of shooting locations.

K : Set of all actors; $k \in K$.

G : Set of all shooting locations; $m, n \in G$.

S : Set of all scenes. Set two virtual scenes 0 and T , in which 0 indicates the scene where the crew begins to shoot and T indicates the scene the crew shot in the end.

$i, j \in S$; $S^0 = S \cup \{0\}$; $S^T = S \cup \{T\}$; $\bar{S} = S \cup \{0, T\}$.

Parameters

p_k : The daily wage of actor k .

s_{ik} : Set to one if actor k appears in scene i , and zero otherwise.

x_{ijk} : Set to one if actor k appears in scene i and scene j , and zero otherwise.

c_{ij} : The cost of moving from scene i to scene j .

u_i : The number of days required to shoot scene i .

Decision Variables

δ_{ijk} : Binary variable; it equals one if the actor k first appears in scene i and then appears in scene j , and zero otherwise; note that it is not required that scene i and scene j are adjacent to each other.

α_{ijk} : Binary variable; it equals one if the actor k appears in two adjacent scenes i and j in which scene j is the next scene of scene i , and zero otherwise.

β_{ijk} : Integer; the time interval of actor k plays between scene i and scene j . For the actor k , it first appeared in the scene i , followed by the scene j .

γ_{ij} : Binary variable; it equals one if the scene j is the next scene of scenes i , and zero otherwise.

ε_i : Integer; the time that scene i finishes.

3.2. Mathematical Model.

$$\text{Minimize } \sum_{i \in \bar{S}} \sum_{j \in \bar{S}} \sum_{k \in K} \beta_{ijk} p_k + \sum_{j \in S} \sum_{i \in S} \gamma_{ij} c_{ij} \quad (1)$$

$$\sum_{j \in S^T} \gamma_{0j} = 1 \quad (2)$$

$$\sum_{i \in S^0} \gamma_{iT} = 1 \quad (3)$$

$$\sum_{j \in S^T} \gamma_{ij} = 1 \quad \forall i \in S, i \neq j \quad (4)$$

$$\sum_{j \in S^0} \gamma_{ji} = 1 \quad \forall i \in S, i \neq j \quad (5)$$

$$\gamma_{ij} + \gamma_{ji} \leq 1 \quad \forall i \in \bar{S}, j \in \bar{S} \quad (6)$$

$$\varepsilon_j - \varepsilon_i - u_j \leq M(1 - \gamma_{ij}) \quad (7)$$

$$\forall i \in S^0, j \in S^T$$

$$\sum_{j \in S} \alpha_{0jk} = 1 \quad \forall k \in K \quad (8)$$

$$\sum_{j \in S} \alpha_{jTk} = 1 \quad \forall k \in K \quad (9)$$

$$\delta_{ijk} + \delta_{jik} = x_{ijk} \quad (10)$$

$$\forall i, j \in \bar{S}, i \neq j, \forall k \in K$$

$$\alpha_{ijk} \leq \delta_{ijk} \quad \forall k \in K, \forall i, j \in \bar{S} \quad (11)$$

$$s_{ik} \leq \sum_{j \in S^T} \alpha_{ijk} \quad \forall i \in S^0, \forall k \in K \quad (12)$$

$$s_{ik} \leq \sum_{j \in S^0} \alpha_{jik} \quad \forall i \in S^T, \forall k \in K \quad (13)$$

$$\varepsilon_j - \varepsilon_i + M(\alpha_{ijk} - 1) \leq \beta_{ijk} \quad (14)$$

$$\forall i, j \in S, i \neq j, \forall k \in K$$

$$u_j + M(\alpha_{0jk} - 1) \leq \beta_{0jk} \quad (15)$$

$$\forall j \in S, \forall k \in K$$

$$M(\alpha_{jTk} - 1) \leq \beta_{jTk} \quad \forall j \in S, \forall k \in K \quad (16)$$

$$s_{jk} - 0.5 \leq 0.5 \left(\sum_{i \in \bar{S}} \alpha_{ijk} + \sum_{i \in \bar{S}} \alpha_{jik} \right) \leq s_{jk} \quad (17)$$

$$\forall j \in \bar{S}, i \neq j, \forall k \in K$$

$$M(\delta_{ijk} - 1) \leq \varepsilon_j - \varepsilon_i \quad (18)$$

$$\forall i \in S^0, \forall j \in S^T, i \neq j, \forall k \in K$$

$$\varepsilon_j - u_j \leq M(2 - \alpha_{0jk} - \gamma_{ij}) + \varepsilon_i \quad (19)$$

$$\forall i \in S^0, \forall j \in S^T, \forall k \in K$$

$$\varepsilon_j - u_j \leq M(2 - \alpha_{jTk} - \gamma_{ij}) + \varepsilon_i \quad (20)$$

$$\forall i \in S^0, \forall j \in S^T, \forall k \in K$$

$$0 \leq \varepsilon_j - \varepsilon_i - u_j + M(1 - \alpha_{ijk}) \quad (21)$$

$$\forall i \in S^0, \forall j \in S^T, \forall k \in K$$

$$0 \leq \varepsilon_i \leq d \quad \forall i \in \bar{S} \quad (22)$$

$$\beta_{ijk} \geq 0 \quad \forall i, j \in \bar{S}, \forall k \in K \quad (23)$$

$$\delta_{ijk}, \alpha_{ijk}, \gamma_{ij} \in \{0, 1\} \quad \forall i, j \in \bar{S}, \forall k \in K \quad (24)$$

Objective function (1) minimizes the total costs of the movie scenes scheduling problem. The total costs include the wage of actors and the transfer cost of scenes. Constraints (2) and (3) guarantee that each scene has a start point 0 and an end point T . Constraints (4) and (5) ensure that there is only one scene before and after each scene. Constraint (6) states the priority relationship of scene i and scene j . Constraint (7) links the two decision variables γ_{ij} and ε_i . Constraints (8) and (9) ensure that each actor k starts with scene 0 and ends with scene T . Constraints (10) and (11) define the relation of α_{ijk} , x_{ijk} , and δ_{ijk} . Constraints (12) and (13) guarantee that for actor k , each scene before and after there is only one scene. Constraints (14), (15), and (16) define β_{ijk} . Constraint (17) ensures that actor k completes the shooting scenes according

to the sequence decided by α_{ijk} . Constraint (18) links the two decision variables δ_{ijk} and ε_i . Constraints (19), (20), and (21) link the two decision variables α_{ijk} and ε_i . Constraints (22), (23), and (24) define the decision variable.

The above model can be solved by ILOG CPLEX solver. However, ILOG CPLEX solver can only solve small-scale movie scenes scheduling problems. For some large-scale movie scenes scheduling problems, the ILOG CPLEX solver cannot solve very well. Therefore, this paper proposes a TSBM to handle the movie scenes scheduling problems under large-scale situation.

4. Tabu Search Based Method

4.1. Introduction to the Tabu Search. At present, many scholars have used tabu search (TS) algorithms to solve optimization problems. Xue et al. [15] studied the problem of local container shipping under the new operating mode, defining that the tractor can be removed from its associated trailer and assigned to a new mission. Aiming at this problem, the TS is proposed, which uses numerical experiments to evaluate the performance of the algorithm and the advantages of the new operation mode. Shen [16] designed a TS with complex neighborhood structure for job shop problems with sequence-related setup time. The method of inserting is used to define the neighborhood function without a simple exchange. The test results show that the proposed TS is superior to other algorithms. Qin et al. [17] studied a multicycle inspector scheduling problem, which is an extension of the multistroke vehicle routing problem with time windows, and then solved using the TS. In addition, a constrained backpack model is proposed, which can generate an upper limit of the problem, thereby improving the efficiency of the solution. Peng et al. [18] combined the TS process into the path relinking framework to study a new algorithm to solve the job shop scheduling problem. The experimental results show that the new algorithm has a good effect on the solution quality and computational efficiency. Li and Alidaee [19] use the TS to solve the black and white traveling salesman problem. The improved TS is built on a new efficient neighborhood structure, which utilizes the arrangement of black and white traveling salesman problems and the backpack function. Moreover, the TS is embedded as a heuristic process to improve the upper bound of the mixed integer linear programming equation. Li and Gao [20] use a hybrid algorithm of genetic algorithm and TS to solve the shop scheduling problem, in order to minimize the completion time. The genetic algorithm with powerful global search ability is used to explore the TS with good local search ability. The experimental results show that the hybrid algorithm has good search ability and can balance the intensification and diversification. Shahvari and Logendran [21] improved the TS for a batch scheduling problem on a set of uncorrelated parallel machines with the goal of minimizing the linear combination of total weighted completion time and total weighted delay. Therefore, we design a TSBM to solve the integer linear programming model. Wei et al. [22] studied the partner selection problem from the perspective of supplier network global optimization. The study established

an optimal combination model of supplier selection to choose the right suppliers for improving the efficiency of the supply chain. A dynamic programming-TS and an ant colony algorithm were developed to solve the model, respectively.

TS is a global progressive optimization algorithm. Factors affecting search performance in TS include initial solution, special rules, neighborhood structure, tabu list, candidate set, and termination rule. For the amnesty rules, if the target value of a solution is better than any of the best candidate solutions, it can be special. And if all objects are contraindicated, then a solution with the smallest evaluation value is selected. It is possible to specifically target objects that have a large impact on the target value. For the size of the candidate set, the excessive calculation increases the computational memory and computation time and too small will cause the premature fall into the local optimum. For termination criteria, it is to make the search terminate the search process in some cases. For the tabu list, it has a short-term memory function that prohibits searching for previously searched content. For the tabu length, the purpose is to prevent the search from appearing in an infinite loop. When other variables are unchanged, the shorter the tabu length, the less memory usage, and the larger the scope of the lifting, that is, the upper limit of the search range. But it is easy to fall into local optimum earlier. If the length of the tabu is too long, it will be time-consuming.

4.2. The Frame of Tabu Search

4.2.1. The Basic Flow of the TSBM. The basic flow of the TSBM is as follows.

Step 1. Determine an initial feasible solution and its objective function value. Set the tabu list to empty and set the length of the tabu list. Meanwhile, let the initial solution as the current solution x_{now} . In addition, define n as the number of iterations. Set $N = 1$, $nowL = 0$.

Step 2. When n reaches the maximum number of iterations N , end the algorithm and output the optimal solution; otherwise, go to Step 3.

Step 3. If $nowL < L$, the neighborhood function that generates the current solution x_{now} will generate a number of neighborhood solutions, and go to Step 5; otherwise, go to Step 4;

Step 4. Use the diversification strategy to generate a new current solution x_{now} , and set $nowL = 0$, go to Step 3.

Step 5. Sort the candidate solutions according to the fitness value, and judge from the candidate solution with the smallest fitness value. If it is not in the tabu list, directly replace it with the current solution to become x_{now} ; if it is in the tabu list and meet the exemption criteria, let it as x_{now} and go to Step 7. Otherwise, go to Step 6.

Step 6. If the candidate solution value is in the tabu list and does not meet the exemption criteria, certain courage criterion is given. If the candidate solution value satisfies the

courage criteria, it is also selected as the current solution x_{now} . Go to Step 7.

Step 7. Update the global optimal x_{best} , and put the current solution into the tabu list. If the tabu list exceeds the set length, release the tabu element that firstly put into the tabu list. When x_{best} has not changed, $nowL = nowL + 1$, $n = n + 1$, go to Step 2.

4.2.2. Initial Solution. The initial solution has great influence on the algorithm search performance because the TSBM is mainly based on the neighborhood search. Therefore, we firstly consider generating a number of feasible solutions for CPLEX solver randomly and select the optimal fitness value of the solution as the initial solution.

Each x represents one possible order of scene, i.e., the variable γ_{ij} is known. If x is given, then we select the optimal fitness value of the solution as the initial solution. For example, if there are 5 scenes, elements in x are 1-4-5-3-2, which means the order of scene is 1-4-5-3-2.

4.2.3. Neighborhood Structure. The neighborhood structure is critical in tabu search because it determines the form and number of neighborhood solutions of the current solution and the relationship between the solutions.

The neighborhood is generated by 2-swap and 2-opt methods. The 2-swap method is to generate neighborhood as explained below. On the basis of the current solution x_{now} , the order of the two scenes is randomly exchanged. For example, x_{now} is 1-4-5-3-2, and we exchange first and third scene to get a neighborhood, i.e., 5-4-1-3-2. However, the method of generating the neighborhood using the 2-opt method is random two scenes for 2-opt exchange. And 2-opt means that from the selected starting scene to the ending scene, all scenes are reversed to generate new neighborhoods. For example, x_{now} is 1-4-5-3-2, and the two random scenes are the second and fifth scenes, so a new neighborhood is 1-2-3-5-4.

4.2.4. Tabu List. The tabu list has a short-term memory function which prohibits the items that are searched before. TSBM uses tabu list to prevent the search process being caught in the local optimal.

The neighborhood exchange method and the exchanges of two points will be added to the tabu list. For example, x_{now} is 1-4-5-3-2, and we exchange the first and the third scene to get a neighborhood, i.e., 5-4-1-3-2. Use the 2-swap method, and exchange the first and the third scene, then add 1-1-3 three numbers to the tabu list. Besides, set x_{now} is 1-4-5-3-2, and two random scenes are the second and the fifth scenes, so the new neighborhood is 1-2-3-5-4. And we use the 2-opt method, and the random number is 2 and 5, then add 2-2-5 to the tabu list. This shows that when determining whether a candidate solution is in the tabu list, you need to know which method is used, and which two points should be random.

4.2.5. Diversification Strategy. In order to avoid the algorithm getting into local optimum prematurely, a diversified strategy is adopted in this paper. If the optimal value does not change after a certain number of iterations, we will adopt a diversified

strategy to regenerate the current solution and make the algorithm jump out of the local minimum. In addition, the method of generating the current solution is the same as that of generating the initial solution, i.e., it is generated randomly.

5. Particle Swarm Optimization Based Method

5.1. Introduction to the PSO. PSO is a new evolutionary algorithm proposed by Kennedy and Eberhart [23] to optimize continuous nonlinear functions. At present, PSO has been widely used to solve problems such as scheduling, combinatorial optimization, and supply chain network design. Garg and Sharma [24] use PSO to solve system assignment problems. Many of these goals are system reliability and design costs. Above all, a fuzzy nonlinear programming model is constructed, and then the particle swarm optimization algorithm is designed to solve the proposed problem. Hamta et al. [25] studied the assembly line balance problem of a single model. This considers the minimization of cycle time, total equipment cost, and smoothness index. A new method combining PSO and variable domain search is used to solve the proposed problem. Zhong et al. [26] use the PSO to solve the multiobjective hull assembly line balance problem. The goal is to minimize cycle time, multisite responsibility, and dynamic and static load balancing. In the particle swarm algorithm, the new method is used to encode the particles. Yao et al. [27] considered the carton heterogeneous vehicle routing problem at the collection station, constructed a mathematical model, and solved the model using the improved PSO. For improving the PSO, the local search is optimized in the algorithm. Experiments show that both the proposed model and the algorithm are feasible. Khatami et al. [28] constructed a PSO for flame detection, which enabled image detection techniques for flame detection. PSO can distinguish the different colors, and the color conversion matrix can detect different color flames. Hu et al. [14] established a mixed integer programming model for the joint optimization of multibatch production and vehicle routing problems involving a pickup. A two-stage hybrid heuristic algorithm is proposed to solve this model. The PSO was used in the second stage to allocate vehicles to each production batch. The results of numerical experiments validate the efficiency of the proposed algorithm.

PSO finds the best solution by simulating the process of bird flocking and sharing information between birds and birds. Each particle can change its position and speed, and the position of the particle is a viable solution in the search space. The speed of the particles determines the direction and distance of the flight of bird. Other particles will search for the direction of the current optimal particle. Each particle remembers, follows the current optimal particle, and searches in the solution space. But the process of each iteration is not completely random. If a better solution is found, it will be used as a basis to find the next solution to find the global optimal solution. The advantage of PSO is that the algorithm is easy to implement and there are not many parameters to adjust.

TABLE 1: The performances of the proposed solution methods.

Case id	TSBM		PSOBM		CPLEX		Gap	Gap1
	OBJ_T	T_T (s)	OBJ_P	T_P (s)	OBJ_C	T_C (s)		
10-3-3-1	2960	14	3040	137	2960	8	0.00%	2.70%
10-3-3-2	3440	21	3980	154	3440	13	0.00%	15.70%
10-3-3-3	4440	7	4630	125	4440	22	0.00%	4.28%
11-3-3-1	4600	42	5010	178	4600	97	0.00%	8.91%
11-3-3-2	4810	7	5130	156	4810	104	0.00%	6.65%
11-3-3-3	4410	43	4630	147	4410	115	0.00%	4.99%
11-5-3-1	5700	14	5860	215	5690	48	0.20%	2.99%
11-5-3-2	6890	46	6930	246	6890	165	0.00%	0.58%
11-5-3-3	6710	21	7060	197	6710	170	0.40%	5.22%
13-5-4-1	22230	57	26430	236	22130	1167	0.00%	19.43%
13-5-4-2	32160	46	33790	318	31800	2850	1.00%	6.26%
13-5-4-3	20180	26	23570	115	20080	6613	0.50%	17.38%
15-5-4-1	36680	51	38800	53	N.A	>7200	N.A	N.A
15-5-4-2	22320	74	31300	39	N.A	>7200	N.A	N.A
15-5-4-3	32800	113	38430	137	N.A	>7200	N.A	N.A
					Avg.		0.18%	7.92%

5.2. *The Frame of Particle Swarm Optimization.* The flow of the PSOBM is as follows.

Step 1. Set the initial iteration number $G = 1$ and the number of particles to H . The position of the particle represents a feasible solution, and the velocity of the particle represents the search direction. The position and velocity of the initial particle swarm are randomly generated.

Step 2. Calculate the fitness value of each particle.

Step 3. Find the individual optimal and global optimal values of each particle. If it is better than the current individual extremum, it is updated; if the best of all the individual extremum of the particles is better than the current global optimal, the global optimal is updated.

Step 4. Use the particle velocity and position formula to update the velocity and position of the particle.

Step 5. If one of the two termination conditions is reached, the iteration is stopped and the optimal solution is output; otherwise, go to Step 2.

Among the above steps, the two termination conditions in Step 5 are as follows. For the first, the number of iterations reaches the maximum number of iterations. And the global optimal value is no longer updated in the number of iterations.

6. Numerical Experiments

6.1. *Experiment Settings.* In this section, we apply the proposed model and methods to solve the movie scenes scheduling problem. The movie scenes scheduling problem is a scheduling problem. The numerical experiments in this paper

mainly refer to the experiments of Zhen [29] and Zhen et al. [30] to analyze the proposed method and model.

The experimental cases in this paper are set as follows. For the cost c_{ij} transferred from scene i to scene j , they are randomly generated within the range (0, 10000). Moreover, it is necessary to satisfy $c_{ij} = c_{ji}$. For the parameter q_k (i.e., the daily wage of actor k), they are randomly generated within a range between 80 dollars/day and 1000 dollars/day. For the parameter s_{ik} (i.e., set to one if the actor k appears in scene i) and x_{ijk} (i.e., set to one if the actor k appears in scene i , scene j), they are mainly generated based on the actor k . Numerical experiments are performed on a PC (Intel (R), 3.4 GHz; Memory, 128 G). The models are implemented by the CPLEX12.6, and the methods are coded by C# (Visual Studio 2015).

6.2. *Performances of the Proposed Solution Methods.* This paper designs a TSBM and PSOBM to solve the proposed integer linear programming model. Compare the performance of the proposed solution with the results obtained by directly using the CPLEX solver. Table 1 shows the comparison of the experimental results of TSBM, PSOBM, and CPLEX solvers under the same parameters. In the ‘‘Case id’’ column of Table 1, the first three values represent the scene, the actor, and the location, and the fourth value represents the index of the different cases at the same scale (scene, actor, and location). The difference between the three cases in each group size is mainly due to the difference in the length of the scene. In this paper, OBJ_T , OBJ_P , and OBJ_C show the objective value of TSBM, PSOBM, and CPLEX. And $Gap = (OBJ_T - OBJ_C)/OBJ_C$, $Gap1 = (OBJ_P - OBJ_C)/OBJ_C$.

It can be seen from Table 1 that under the small-scale problems, the TSBM has a shorter solution time than CPLEX, and the average difference rate is 0.18%. The PSOBM has a longer solution time, and the average difference rate is

TABLE 2: Comparison between TSBM and PSOBM.

Case id	TSBM		PSOBM		Gap2
	OBJ _T	T _T (s)	OBJ _P	T _P (s)	
15-5-4-1	36680	51	38800	53	5.78%
15-5-4-2	22320	74	31300	39	40.23%
15-5-4-3	32800	113	38430	137	17.16%
20-5-6-1	30320	122	46330	102	52.80%
20-5-6-2	35260	127	53840	81	52.69%
20-5-6-3	32090	214	47350	163	47.55%
30-10-10-1	227730	677	341930	813	50.15%
30-10-10-2	225000	616	380560	898	69.14%
30-10-10-3	213140	654	241730	946	13.41%
30-20-10-1	468200	1295	572380	1547	22.25%
30-20-10-2	439530	1240	668980	1435	52.20%
30-20-10-3	515170	1293	594190	1636	15.34%
40-20-15-1	897070	2176	1092090	2807	21.74%
40-20-15-2	692040	2112	984500	2749	42.26%
40-20-15-3	763560	2034	978900	3006	28.20%
				Avg.	35.39%

7.92%. Therefore, in the small-scale movie scene scheduling problem, the TSBM can obtain good results in a shorter time than the PSOBM. Based on the above analysis, it can be concluded that for small-scale problems, the proposed TSBM is a better way to obtain a satisfactory solution.

6.3. Comparison between TSBM and PSOBM. For small-scale problems, CPLEX, TSBM, and PSOBM can obtain good results. But the CPLEX takes a long time. In addition, the result of TSBM and PSOBM takes a short time. Therefore, for large-scale problems, the validity of the TSBM and PSOBM needs to be further studied experimentally. In this test, the result of the comparison between TSBM and PSOBM is listed in Table 2. And $Gap2 = (OBJ_P - OBJ_T) / OBJ_T$.

The results in Table 2 show that the average difference between the experimental results obtained by the TSBM and the PSOBM is 35.39%. And as the size of the experimental case increases, the time required for problem solving continues to increase. Experimental results show that TSBM is more suitable for solving medium-scale movie scenes scheduling problem.

7. Conclusions

This paper mainly studies the optimization problem of movie scenes scheduling by considering transfer costs and certain shooting duration. The purpose is to minimize the total cost of actors and transfer costs included in the shooting scene of the crew, considering the different wage of the actors, the duration of the scene shooting, and the transfer costs between scenes. An integer linear programming model is built, and TSBM and PSOBM are developed to solve the proposed model. Being compared with other related research work, the major contribution of this study includes two aspects:

(1) Being different from the general movie scenes scheduling problem, the proposed model in this paper considers the transfer costs and certain shooting duration.

(2) We propose different methods of TSBM and PSOBM for the movie scenes scheduling problem with different scale.

The numerical experiments indicate both TSBM and PSOBM are suitable for solving small-scale problems. For large-scale problems, the TSBM can get better results.

For future study, we will consider finding a better optimization method in TSBM to extend our work.

Data Availability

The data used in this research is generated randomly. Detailed data generation method is introduced in Section 6.1.

Conflicts of Interest

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

References

- [1] T. C. Cheng, J. E. Diamond, and B. M. Lin, "Optimal scheduling in film production to minimize talent hold cost," *Journal of Optimization Theory and Applications*, vol. 79, no. 3, pp. 479–492, 1993.
- [2] A.-L. Nordström and S. Tufekci, "A genetic algorithm for the talent scheduling problem," *Computers & Operations Research*, vol. 21, no. 8, pp. 927–940, 1994.
- [3] A. Fink and S. Voß, "Applications of modern heuristic search methods to pattern sequencing problems," *Computers & Operations Research*, vol. 26, no. 1, pp. 17–34, 1999.
- [4] F. Bomsdorf and U. Derigs, "A model, heuristic procedure and decision support system for solving the movie shoot scheduling problem," *OR Spectrum*, vol. 30, no. 4, pp. 751–772, 2008.

- [5] M. Garcia de la Banda, P. J. Stuckey, and G. Chu, "Solving talent scheduling with dynamic programming," *INFORMS Journal on Computing*, vol. 23, no. 1, pp. 120–137, 2011.
- [6] Y. Kochetov, "Iterative local search methods for the movie scene scheduling problem," in *Proceedings of the 1st International Symposium and Balkan Conference on Operational Research*, pp. 282–288, Thessaloniki, Greece, September 2011.
- [7] X. Liang, Z. Zhang, H. Qin, S. Guo, and A. Lim, "A Branch-and-Bound Algorithm for the movie scene scheduling problem," in *Proceedings of the 27th International Conference on Modern Advances in Applied Intelligence, Part I*, vol. 8481, pp. 208–217, Springer, New York, NY, USA, 2014.
- [8] H. Qin, Z. Zhang, A. Lim, and X. Liang, "An enhanced branch-and-bound algorithm for the talent scheduling problem," *European Journal of Operational Research*, vol. 250, no. 2, pp. 412–426, 2016.
- [9] T. C. Cheng, B. M. Lin, and H.-L. Huang, "Talent hold cost minimization in film production," *Journal of Industrial and Management Optimization*, vol. 13, no. 1, pp. 223–235, 2017.
- [10] S.-Y. Wang, Y.-T. Chuang, and B. M. T. Lin, "Minimizing talent cost and operating cost in film production," *Journal of Industrial and Production Engineering*, vol. 33, no. 1, pp. 17–31, 2016.
- [11] P. Gregory, A. Miller, and P. Prosser, "Solving the rehearsal problem with planning and with model checking," *College of Science and Engineering School of Computing Science*, vol. 103, no. 3, pp. 441–446, 2004.
- [12] N. Sakulsom and W. Tharmmaphornphilas, "Scheduling a music rehearsal problem with unequal music piece length," *Computers & Industrial Engineering*, vol. 70, no. 1, pp. 20–30, 2014.
- [13] L. Zhen, "Tactical berth allocation under uncertainty," *European Journal of Operational Research*, vol. 247, no. 3, pp. 928–944, 2015.
- [14] H. Hu, J. Mo, and C. Ma, "Optimization of vehicle routing with pickup based on multibatch production," *Discrete Dynamics in Nature and Society*, vol. 2018, Article ID 2804589, 9 pages, 2018.
- [15] Z. Xue, C. Zhang, W.-H. Lin, L. Miao, and P. Yang, "A tabu search heuristic for the local container drayage problem under a new operation mode," *Transportation Research Part E*, vol. 62, no. 1, pp. 136–150, 2014.
- [16] L. Shen, "A tabu search algorithm for the job shop problem with sequence dependent setup times," *Computers & Industrial Engineering*, vol. 78, pp. 95–106, 2014.
- [17] H. Qin, W. Ming, Z. Zhang, Y. Xie, and A. Lim, "A tabu search algorithm for the multi-period inspector scheduling problem," *Computers & Operations Research*, vol. 59, pp. 78–93, 2015.
- [18] B. Peng, Z. Lü, and T. C. E. Cheng, "A tabu search/path relinking algorithm to solve the job shop scheduling problem," *Computers & Operations Research*, vol. 53, no. 53, pp. 154–164, 2015.
- [19] H. Li and B. Alidaee, "Tabu search for solving the black-and-white travelling salesman problem," *Journal of the Operational Research Society*, vol. 67, no. 8, pp. 1061–1079, 2016.
- [20] X. Li and L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, vol. 174, pp. 93–110, 2016.
- [21] O. Shahvari and R. Logendran, "An enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes," *Computers & Operations Research*, vol. 77, pp. 154–176, 2017.
- [22] C.-T. Wei, H. Zuo, C.-B. Jiang, and S.-f. Li, "Modeling multilevel supplier selection problem based on weighted-directed network and its solution," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 8470147, 12 pages, 2017.
- [23] J. Kennedy and R. Eberhar, "Particle swarm optimization," in *Proceedings of the IEEE International Conference*, pp. 1942–1948, IEEE, 1995.
- [24] H. Garg and S. P. Sharma, "Multi-objective reliability-redundancy allocation problem using particle swarm optimization," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 247–255, 2013.
- [25] N. Hamta, S. M. T. Fatemi Ghomi, F. Jolai, and M. Akbarpour Shirazi, "A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect," *International Journal of Production Economics*, vol. 141, no. 1, pp. 99–111, 2013.
- [26] Z. Yuguang, A. Bo, and Z. Yong, "A PSO algorithm for multi-objective hull assembly line balancing using the stratified optimization strategy," *Computers & Industrial Engineering*, vol. 98, pp. 53–62, 2016.
- [27] B. Yao, B. Yu, and P. Hu, "An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot," *Annals of Operations Research*, vol. 242, no. 2, pp. 303–320, 2016.
- [28] A. Khatami, S. Mirghasemi, A. Khosravi, C. P. Lim, and S. Nahavandi, "A new PSO-based approach to fire flame detection using K-Medoids clustering," *Expert Systems with Applications*, vol. 68, pp. 69–80, 2017.
- [29] L. Zhen, "Modeling of yard congestion and optimization of yard template in container ports," *Transportation Research Part B: Methodological*, vol. 90, pp. 83–104, 2016.
- [30] L. Zhen, E. P. Chew, and L. H. Lee, "An integrated model for berth template and yard template planning in transshipment hubs," *Transportation Science*, vol. 45, no. 4, pp. 483–504, 2011.

