# Optimization of Production Cost of Film and Its Solution Method

Xiao-jiao Wang*

School of Management, Shanghai University, Shanghai 200444, China

*vera_wxj@163.com

**Keywords:** Discrete particle swarm optimization, genetic algorithms, movie, scheduling optimization.
**Abstract.** Considering the difference between actors' fee, this paper constructed the model of movie talent scheduling problem to minimize the total costs of talents. According to the characteristic of the problem, this paper applies the genetic algorithm with elitist strategy and the discrete particle swarm optimization algorithm which combined genetic operations and can adjust the inertia weight dynamically to work out this problem in different sizes. Experimental results show that, the genetic algorithm with elitist strategy can obtain a better solution when the problem size is small, while the discrete particle swarm optimization is more efficient for the large scale problems.

## Introduction

Film production is a project which expends vast human and material resources, including a lot of scheduling work, for example talents, sites, and production tools. Among them, the rational choice in actors is undoubtedly important to the success of a film. Besides, the spending on talents has also become the core of the film production costs. In a survey of China's movie ticket cost shows that the film production costs have risen sharply which is largely due to the rising prices, especially the talents' costs. For example, the cost on movie stars accounts for about forty percent of the total production costs in the movie< Don't Go Breaking My Heart > [1] and < The Warlords> [2].Therefore, according to the requirements of different scenes in movie to make a reasonable scheduling to talents will not only improve the return on investment in film-making, but also boost the public's cultural consumption.

The talent scheduling problem was introduced by Cheng [3].They proved that the problem was strongly NP hard problem and the heuristic algorithm was more efficient in get the approximate optimal solution than the branch and bound optimization algorithm. Zhang [4] made the improvement on the branch and bound algorithm by adding the methods of pretreatment and advantage rule. de la Banda .M.G[5] used an optimized dynamic programming method which contained the pretreatment and a limited search boundary to solve this problem. However, these algorithms cannot avoid the "dimension disaster" when the problem size is increasing. Then, intellectual start research the heuristic algorithm to solve NP-hard problems like the combinatorial optimization problems. Jiao [6] applied GA to solve the flight scheduling problem. Tang [7] applied PSO to solve the pre- warning satellite scheduling problem. All these intellectuals also made some improvements on GA and PSO algorithms which included changing the way of coding or altering the motion equation of swarms and so on. Therefore, this article will apply the GA and DPSO algorithms to solve the talent scheduling problem with the target of minimizing the total talents' costs on the bases of ever improved heuristic algorithms. In addition, the optimization results of two algorithms will be compared and analyzed.

## Mathematical Model

During the film shooting, it is always divided into different scenes according to the script as well as the realistic field of shooting. Then, the director will arrange the shooting task to talents on the basis of different scenes. Because the shooting activities for each actor cannot be carried out continuously, the producer shall compensate some actors for the waiting time of non-shooting activities. In this paper, we assume that the compensation expense for each actor is equal to the normal reward. When the number of scenes and actors is increasing, the spending on total talents will be the main part of a film production costs. Therefore, the key to the talents scheduling problem is the optimization of the scenes

 

de la Banda, M. G[5] has summarized the proposed problem in his research. The relevant variables in the model as follows:

m —— The number of scenes. $i \in \{1,2,3,...m\}$

S —— A set of scene, $S = \{s_1, s_2, s_3, ... s_m\}$

$u_i$ —— Shooting duration of each scene

n —— The number of talents. $j = \{1,2,3,...n\}$

A —— A set of talent. $A = \{a_1, a_2, a_3, ... a_n\}$

$c_j$ —— Cost of each talent per day.

$a_j(s)$ ——A set of scene which a talent j participate. $s \in S$

For each talent, the complete shooting process is from the first scene to the last shooting activity he or she performed according to the shooting sequence arranged by the director. For example, a set of scenarios is R. $e_j(R)$ and $f_j(R)$ are first and last scene numbers respectively which talent j participates in this scene shooting sequence. Total work time of talent j is calculated and total spending of actors is represented by C which is our optimization objective are calculated in Eq.1 and Eq.2:

$$T_j = \sum_{i=e_j(R)}^{f_j(R)} u_i \qquad e_j(R) \in a_j(s),\ f_j(R) \in a_j(s) \tag{1}$$

$$C = \sum_{j=1}^{n} (c_j \bullet \sum_{i=e_j(R)}^{f_j(R)} u_i) \qquad e_j(R) \in a_j(s),\ f_j(R) \in a_j(s) \tag{2}$$

Therefore, the core of the talent scheduling problem is to find a satisfied shooting sequence to minimize the total costs of talents.

## Application of Genetic Algorithm

Genetic algorithm (GA) is a global optimization algorithm which is based on the imitation of the natural biological evolution process. GA searches a better solution by the fitness value of individual. In the process of programming, the most important three parts are the encoding, fitness function, and the design of genetic operators. In addition, this paper adds an elitist strategy in the process of programming to improve the optimization effect of the GA. The algorithm operations as follows:

### Encoding

Talent scheduling problem is a discrete combination optimization problem in essence. The key to solve this problem is how to arrange the sequence of scenes. This paper encodes the chromosome to integer. Taking the number of scene as the gene on the chromosome, then each chromosome represents a sequence of scenes, also a solution.

For example, a chromosome $S_1$={1,3,5,2,6,4,8,10,7,9} means the shooting sequence in the problem of 10 scenes and 5 actors.

### Fitness function

In this paper, the reciprocal of the total costs of talents is used as fitness function. Therefore, when total costs calculated by formula 2 is smaller, the fitness value is larger, and then the individual chromosome is more likely to be retained in the iteration.

### Genetic operations

#### Selection

Selection operation is an important step of the algorithm, which reflects the tendency of individual evolution. This paper chooses the roulette wheel method for it is easy to implement. In this method, the probability of the individual to be chosen is equal to the fitness value of this individual accounts for the proportion of all the individuals in the population.

 

**Crossover**

Crossover operation in genetic algorithm promotes the generation of the optimal solution. This paper will generate two points randomly to make crossover operation of individuals. For example, two individuals are $S_1$={1,2,3,4,5,6,7,8,9,10} and $S_2$={2,4,7,3,8,10,1,5,6,9}.There are two points generated randomly in [1, 9], which are 3 and 7. Through the crossover operation, two new individuals are $S_1$*={2,3,8,4,5,6,7,10,1,9}and $S_2$*={2,4,5,3,8,10,1, 6,7,9}.

The selection of parameter is very important in crossover operation. It is not conducive to obtain a satisfied solution that the cross probability is too small or too great. Therefore, it is appropriate to take 0.5~0.9 for cross probability [8-9]. In this paper, the crossover rate Pc is selected as 0.8 after the large experiment tests.

**Mutation**

Mutation operation is a kind of local random search, which keeps the population diversity and prevents the non - mature convergence effectively. In this paper, the position of two genes in chromosome will be traded to mutate. For example, the individual mentioned above is $S_1$={2,4,7,3,8,10,1,5,6,9}. If two random points are 5 and 8, the new individual after mutation operation is $S_1$*={2,4,7,3,5,10,1,8,6,9}.

In general, the mutation probability is small, because a great variation rate will lead to instability. But it cannot play the role of maintaining population diversity if this parameter is too small. Therefore, it is appropriate to take 0.01~0.1for it [8-9]. After the large experiment tests, the mutation rate Pm is selected as 0.04.

**Elitist strategy**

During the iterative process of traditional genetic algorithm, the optimal individual often cannot survive through a series of crossover and mutation operations. Hence, the elitist strategy helps to retain the good individuals and enhance the efficiency of optimization.

## Application of Particle Swarm Optimization

Particle swarm optimization algorithm (PSO) develops based on the intelligence of the biological groups. Each particle has the position and velocity in PSO. They take a movement by seeking the previous best positions of themselves and the position of optimal individual in the group to search globally.

**Primary process of PSO**

Assuming all particles move in a space of n dimensions, the optimal position of each particle is pbest, and the position of globally optimal particle is gbest. $Y_i = (y_{i1}, y_{i2}, \ldots y_{in})$ is the position of particle i, and its velocity is $V_i = (v_{i1}, v_{i2}, \ldots v_{in})$. $pbest_i = (p_{i1}, p_{i2}, \ldots p_{in})$ is the optimal position of particle i and $gbest = (g_1, g_2, \ldots g_n)$ is the position of the optimal individual in the group. Then, from the generation t to t+1, the updated position and velocity equation of particle i in dimension jcan be represented in Eq.3 and Eq.4 [10]:

$$y_{ij}(t+1) = y_{ij}(t) + v_{ij}(t+1) \tag{3}$$

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1 (p_{ij}(t) - y_{ij}(t)) + c_2 r_2 (g_{ij}(t) - y_{ij}(t)) \tag{4}$$

In Eq.4, is the inertia weight coefficient which represents how much effect that previous velocity on the velocity updating of particle i. $C_1$ and $C_2$ are cognitive factor and social factor of particle which represents the degree of bias of learning from the best location of itself or the best location in the group. $r_1$ and $r_2$ are random numbers, $r_1 \in [0,1]$, $r_2 \in [0,1]$.

**Discrete Particle Swarm Optimization**

This paper decides to apply DPSO which combines the crossover and mutation operations in GA to promote particles' motion for it is difficult to describe the velocity and position of particles by applying the traditional PSO [11],

 

Among the information of updating velocity of particles, the influence from previous velocity is determined by $w$. When $w$ increases, the global search ability of the particles will be enhanced, but in contrast the local search ability of particles will increase [12]. Therefore, a dynamic adjustment mechanism for $w$ is shown in Eq.5.

$$w = \frac{fitness(i)}{fitnesspbest_i} \tag{5}$$

In addition, $C_1$ and $C_2$ control the updating position of particles. This paper takes the reciprocal of particles as the judging criteria for the operation and makes $C_1$ equal to $C_2$ to keep the consistency of the cognition and learning community of particles. The iterative steps of the DPSO are as follows:

1). Set the population, iterative generation, and initial positions of particles.

2). Calculate the fitness value of particles and initialize their individual extreme values and the global extreme value.

3). Generate a random number r, and compare the value of r and $w$ of each particle. When r is greater than $w$, particle takes mutation operation, otherwise the position is invariant.

4). Generate random numbers $r_1$ and $r_2$. Each particle will take the cross operation with the optimal position of its own if $r_1 > \frac{1}{c_1}$, then it takes the cross operation with the optimal position in global.

5). Calculate the updated fitness value of each particle, and update the individual extreme value and the global extreme value.

6). Return the third step to carry out the iterative operations until iteration generation is reached.

## Experimental Analysis

This paper programs by visual studio Microsoft 2008 software, and the programming environment is Windows 7 operating system. Four experiments are set for comparing the optimization results of two algorithms, and these experiments have different number of actors and scenarios. The populations of the two algorithms are 200, and the iteration generation is 500. When two algorithms run 20 times respectively, the data obtained under four scenarios are compared in Table 1, and the convergence trends of algorithms are also compared in Fig.1 to Fig4 when they nearby their mean values.

**Table 1 Data comparison of algorithms under four scenarios**

| | GA | | | DPSO | | |
|---|---|---|---|---|---|---|
| **Scale** | Min | Mean | Time/s | Min | Mean | Time/s |
| **10_5** | 424 | 424.7 | 1.67 | 456 | 456.1 | 2.04 |
| **20_10** | 1364 | 1428.2 | 2.89 | 1444 | 1495.5 | 3.37 |
| **30_15** | 3192 | 3300.5 | 4.14 | 3193 | 3378.7 | 5.67 |
| **40_20** | 8351 | 8597.9 | 6.60 | 7829 | 8356.9 | 7.58 |

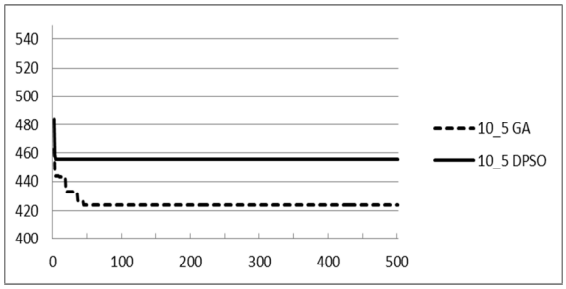


**Fig.1 Comparison of the convergence trend on 10_5**

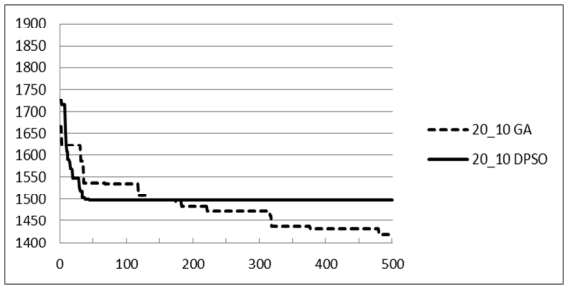


**Fig.2 Comparison of the convergence trend on 20_10**

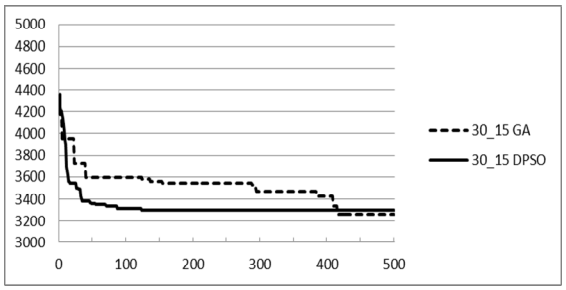


**Fig.3 Comparison of the convergence trend on 30_15**

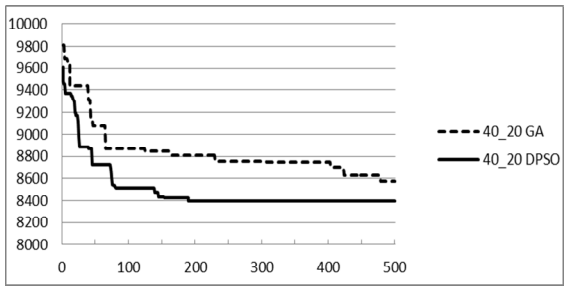


**Fig.4 Comparison of the convergence trend on 40_20**

By comparing the experiment results, some conclusions can be obtained as follows.

1. In Table 1, GA has an advantage over DPSO in getting the optimal value, mean value, as well as the computing time in the first three scenarios. However, the gap between GA and DPSO decreases on optimal value and mean value with the expansion of problem' scale. DPSO is obviously superior to the experimental results of GA on 40_20.

2. In figure 1 to 4, the convergence rate of DPSO is faster, yet the GA iteration process is more stable. In the first scenarios, DPSO is too early to fall into the local optimum, so the result is not ideal. When the number of scenarios increases to 20, even 30, DPSO is obviously better than the GA within 100 generations. Moreover, when they converge to their optimal solutions, the iteration number of DPSO is significantly less than GA. In Figure 4, the effect of DPSO is better than GA on 40_20.

Obviously, it is the different operating mechanisms between GA and DPSO lead to the varied results. When the scale of the problem is large, the dynamic inertia weight can balance the global and local search ability of DPSO, so that it cannot get into local optimal solution too early. On the contrary, DPSO is too early to fall into the local optimal solution for the affect of dynamic inertia weight is not obvious in a small scale problem, so the advantage of GA is more prominent.

## Conclusion

According to the characteristics of film making, this paper applies GA and DPSO to solve the talent scheduling problem which aims at minimize the total costs of the participating actors to optimize the film production costs ultimately. The experimental results show that the optimization ability of the algorithm is related to the size of the problem. GA has a better solution on the problem of small scale; nevertheless DPSO is more efficient on the larger scale problems.

## References

[1] Mengwei Jiang, "Actors' fee accounted for forty percent of all costs, then who was scared off by the rising movie' admission" [N], Beijing Business Daily, Mar.2012.(In Chinese)

[2] Liujie Yan, Xuewen Ren, "The current situation and development strategy of chinese film industry" [J], Youth, Sep.2012, vol.02, pp.8-9.(In Chinese)

[3] T. C. E. Cheng, J. E. Diamond, and B. M. T. Lin, "Optimal scheduling in film production to minimize talent hold cost" [J], Journal of Optimization Theory and Application, 1993, vol.79, no.3, pp.479-492.

[4] Zizhen Zhang, Hu Qin, Xiaocong Liang, Andrew Lim and Andrew Lim, "An enhanced branch-and-bound algorithm for the talent scheduling problem"[C], Cornell University Library, 2014, pp.1-33.(In Chinese)

[5] Maria Garcia de la Banda., P. J. Stuckey, and Geoffrey Chu, "Solving talent scheduling with dynamic programming" [J], INFORMS Journal on Computing, 2011, vol.23, no.1, pp. 120 – 137.

 

[6] Xiaobing Jiao, Xiangdong Fei, and Zehui Xie, “Research on arrival flights landing sequence based on improved genetic algorithm” [J], Computer Technology and Development, 2014, vol.24, no.2, pp.246-249. (In Chinese)

[7] Shaoxun Tang, Zhao Yi, and Xueshan Luo, “An improved particle swarm optimization algorithm for early warning satellites scheduling problems” [J], System Engineering, 2012, vol.30, no.1, pp.116-121. (In Chinese)

[8] Shuai Yuan , Bradley Skinner, Shoudong Huang, Dikai Liu, “A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms” [J], European Journal of Operational Research, 2013, vol.228, pp.72-82. (In Chinese)

[9] Xiaoping Wang, “Genetic algorithm--Theory application and software implementation” [M], Xi-an: Xi'an Jiao Tong University Press, 2002, pp.28-49. (In Chinese)

[10] Feng Qian, “Particle swarm optimization and its application on industry” [M]. Beijing: Science Press, 2013, pp.29-34. (In Chinese)

[11] Ashoka Varthanan Perumal, N. Murugan, and G. Mohan Kumar,“A discrete PSO approach for generating an integrated multi-plant aggregate production-distribution plan” [J].International Journal of Knowledge-based and Intelligent Engineering Systems, 2013, vol.17, pp.195–207.

[12] Wen Li, Tiebin Wu, Quanyou Zhao, and Lingxiang Li, “Improved algorithm of chaotic particle swarm and its application in TSP” [J/OL], Computer Application and Research, 2015, vol.32, http://www.cnki.net/kcms/detail/51.1196.TP.20150130.1142.047.html.(In Chinese)

 