European Journal of Operational Research 250 (2016) 412–426

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

An enhanced branch-and-bound algorithm for the talent scheduling problem

Hu Qin[a], Zizhen Zhang[b,*], Andrew Lim[c], Xiaocong Liang[d]

[a] School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, China
[b] School of Mobile Information Engineering, Sun Yat-sen University, Guangzhou, China
[c] Department of Industrial & Systems Engineering National University of Singapore 1, Engineering Drive 2, Singapore 117576
[d] Department of Computer Science, Sun Yat-Sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 8 November 2014
Accepted 1 October 2015
Available online 22 October 2015

Keywords:
scheduling
Talent scheduling
Branch-and-bound
Dynamic programming
Dominance rules

ABSTRACT

The talent scheduling problem is a simplified version of the real-world film shooting problem, which aims to determine a shooting sequence so as to minimize the total cost of the actors involved. In this article, we first formulate the problem as an integer linear programming model. Next, we devise a branch-and-bound algorithm to solve the problem. The branch-and-bound algorithm is enhanced by several accelerating techniques, including preprocessing, dominance rules and caching search states. Extensive experiments over two sets of benchmark instances suggest that our algorithm is superior to the current best exact algorithm. Finally, the impacts of different parameter settings, algorithm components and instance generation distributions are disclosed by some additional experiments.

1. Introduction

The scenes of a film are not generally shot in the same sequence as they appear in the final version. Finding an optimal sequence in which the scenes are shot motivates the investigation of the talent scheduling problem, which is formally described as follows. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n scenes and $A = \{a_1, a_2, \ldots, a_m\}$ be a set of m actors. All scenes are assumed to be shot on a given location. Each scene $s_j \in S$ requires a subset $a(s_j) \subseteq A$ of actors and has a duration $d(s_j)$ that commonly consists of one or several days. Each actor a_i is required by a subset $s(a_i) \subseteq S$ of scenes. We denote by Π the permutation set of the n scenes and define $e_i(\pi)$ (respectively, $l_i(\pi)$) as the earliest day (respectively, the latest day) in which actor i is required to be present on location in the permutation $\pi \in \Pi$. Each actor $a_i \in A$ has a daily wage $c(a_i)$ and is paid for each day from $e_i(\pi)$ to $l_i(\pi)$ regardless of whether he (or she) is required in the scenes. The objective of the talent scheduling problem is to find a shooting sequence (i.e., a permutation $\pi \in \Pi$) of all scenes that minimizes the total paid wages.

Table 1 presents an example of the talent scheduling problem, which is reproduced from de la Banda, Stuckey, and Chu (2011). The information of $a(s_j)$ and $s(a_i)$ is determined by the $m \times n$ matrix M shown in Table 1(a), where cell $M_{i,j}$ is filled with an "X" if actor a_i participates in scene s_j and with a " · " otherwise. Obviously, we can obtain $a(s_j)$ and $s(a_i)$ by $a(s_j) = \{a_i | M_{i,j} = X\}$ and $s(a_i) = \{s_j | M_{i,j} = X\}$, respectively. The last row gives the duration of each scene and the rightmost column gives the daily cost of each actor. If the shooting sequence is $\pi = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}\}$, we can get a matrix $M(\pi)$ shown in Table 1(b), where in cell $M_{i,j}(\pi)$ a sign "X" indicates that actor a_i participates in scene s_j and a sign "–" indicates that actor a_i is waiting at the filming location. The cost of each scene is presented in the second-to-last row and the total cost is 604. The cost incurred by the waiting status of the actors is called *holding cost*, which is shown in the last row of Table 1(b). The optimal solution of this instance is $\pi^* = \{s_5, s_2, s_7, s_1, s_6, s_8, s_4, s_9, s_3, s_{11}, s_{10}, s_{12}\}$ whose total cost and holding cost are 434 and 53, respectively.

The talent scheduling problem was originated from Adelson, Norman, and Laporte (1976) and Cheng, Diamond, and Lin (1993). Adelson et al. (1976) introduced an orchestra rehearsal scheduling problem, which can be viewed as a restricted version of the talent scheduling problem with all actors having the same daily wage. They proposed a simple dynamic programming algorithm to solve their problem. Cheng et al. (1993) studied a film scheduling problem in which all scenes have identical duration. They first showed that the problem is NP-hard even if each actor is required by two scenes and the daily wage of each actor is one. Next, they devised a branch-and-bound algorithm and a simple greedy hill climbing heuristic to solve

* Corresponding author. Tel.: +86 13826411848; fax: +852 34420189.
E-mail addresses: tigerqin1980@gmail.com, tigerqin@hotmail.com (H. Qin), zhangzizhen@gmail.com (Z. Zhang), lim.andrew@cityu.edu.hk (A. Lim), yurilxc@gmail.com (X. Liang).

http://dx.doi.org/10.1016/j.ejor.2015.10.002
0377-2217/

Table 1
An example of the talent scheduling problem reproduced from de la Banda et al. (2011).

(a) The matrix M for an instance of the talent scheduling problem.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	$c(a_i)$
a_1	X	·	X	·	·	X	·	X	X	X	X	X	20
a_2	X	X	X	X	X	·	X	·	X	·	X	·	5
a_3	·	X	·	·	·	·	X	X	·	·	·	·	4
a_4	X	X	·	·	X	X	·	·	·	·	·	·	10
a_5	·	·	·	X	·	·	·	X	X	·	·	·	4
a_6	·	·	·	·	·	·	·	·	·	X	·	·	7
$d(s_j)$	1	1	2	1	3	1	1	2	1	2	1	1	

(b) The matrix $M(\pi)$ corresponding to a solution π of the instance.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	$c(a_i)$
a_1	X	–	X	–	–	X	–	X	X	X	X	X	20
a_2	X	X	X	X	X	–	X	–	X	–	X	·	5
a_3	·	X	–	–	–	–	X	X	·	·	·	·	4
a_4	X	X	–	–	X	X	·	·	·	·	·	·	10
a_5	·	·	·	X	–	–	–	X	X	·	·	·	4
a_6	·	·	·	·	·	·	·	·	·	X	·	·	7
Cost	35	39	78	43	129	43	33	66	29	64	25	20	604
Holding cost	0	20	28	34	84	13	24	10	0	10	0	0	223

their problem. Later, Smith (2003) applied constraint programming to solve both the problems introduced by Adelson et al. (1976) and Cheng et al. (1993). In her subsequent work, namely Smith (2005), she accelerated her constraint programming approach by caching search states.

The talent scheduling problem we study in this article was first formally described by de la Banda et al. (2011). This problem is a generalization of the problems introduced by Adelson et al. (1976) and Cheng et al. (1993), where scenes may have different durations and actors may have different wages. However, it is a simplified version of the movie shoot scheduling problem (MSSP) introduced by Bomsdorf and Derigs (2008). In the MSSP, we need to deal with a couple of practical constraints, such as the precedence relations among scenes, the time windows of each scene, the resource availability, and the working time windows of actors and other film crew members. Recently, Liang, Zhang, Qin, Guo, and Lim (2014) proposed a branch-and-bound algorithm to solve the talent scheduling problem and achieved better results than de la Banda et al. (2011).

In literature, there exist several meta-heuristics developed for the problem introduced by Cheng et al. (1993). Nordström and Tufekci (1994) provided several hybrid genetic algorithms for this problem and showed that their algorithms outperform the heuristic approach in Cheng et al. (1993) in terms of both solution quality and computation speed. Fink and Voß (1999) treated this problem as a special application of the general pattern sequencing problem, and implemented a simulated annealing algorithm and several tabu search heuristics to solve it.

The talent scheduling problem is a very challenging combinatorial optimization problem. The current best exact approach by de la Banda et al. (2011) can only optimally solve small- and medium-size instances. In this paper, we propose an enhanced branch-and-bound algorithm for the talent scheduling problem, which uses the following two main techniques:

- *Dominance rules.* When a partial solution represented by a node in the search tree can be dominated by another partial solution, this node need not be further explored and can be safely discarded.
- *Caching search states.* The talent scheduling problem can be solved by dynamic programming algorithm (see de la Banda et al. (2011)). It is beneficial to incorporate the dynamic programming states into the branch-and-bound framework by a memoization technique. In the branch-and-bound tree, each node is related to a dynamic programming state. If the search process explores a certain node whose already confirmed cost is not smaller than the value of its corresponding cached state, this node can be pruned.

There are three main contributions in this paper. Firstly, we formulate the talent scheduling problem as a mixed integer linear programming model so that commercial mathematical programming solvers can be applied to the problem. Secondly, we propose an enhanced branch-and-bound algorithm whose novelties include a new lower bound, caching search states and two problem-specific dominance rules. Thirdly, we achieved the optimal solutions for more benchmark instances by our algorithm. The experimental results show that our branch-and-bound algorithm is superior to the current best exact approach by de la Banda et al. (2011).

The remainder of this paper is organized as follows. In Section 2, we present the mixed integer linear programming model for the talent scheduling problem. Next, we describe our branch-and-bound algorithm in Section 3, including the details on a double-ended search strategy, the computation of the lower bound, a preprocessing step, the state caching process and the dominance rules. The computational results are reported in Section 4, where we used our algorithm to solve over 200,000 benchmark instances. Finally, we conclude our study in Section 5 with some closing remarks.

2. Mathematical formulation

The talent scheduling problem is essentially a permutation problem. It tries to find a permutation (i.e., a schedule) $\pi = (\pi(1), \ldots, \pi(n)) \in \Pi$, where $\pi(k)$ is the kth scene in permutation π, such that the total cost $C(\pi)$ is minimized. The value of $C(\pi)$ is computed as:

$$C(\pi) = \sum_{i=1}^{m} c(a_i) \times \big(l_i(\pi) - e_i(\pi) + 1\big)$$

We set the parameter $m_{i,j} = 1$ if $M_{i,j} = X$ and $m_{i,j} = 0$ otherwise. The total holding cost can be easily derived as:

$$H(\pi) = \sum_{i=1}^{m} c(a_i) \times \left(l_i(\pi) - e_i(\pi) + 1 - \sum_{j=1}^{n} m_{i,j} d(s_j)\right)$$

Apparently, for this problem minimizing the total cost is equivalent to minimizing the total holding cost (de la Banda et al., 2011).

We create two dummy scenes s_0 and s_{n+1} to represent the first and the last scenes to be shot, namely, $\pi(0) = s_0$ and $\pi(n+1) = s_{n+1}$. The starting days for shooting s_0 and s_{n+1} are equal to zero and $\sum_{j=1}^{n} d(s_j) + 1$, respectively. The durations of s_0 and s_{n+1} are both equal to zero. The talent scheduling problem can be formulated into an integer linear programming model using the following decision variables: