

IMMC 2015

I Abstract (Summary)

In the analysis of film scheduling, we based our two models on three instrumental factors: resource availability, time requirements, and dependency. Both models use a directed acyclic graph to represent the filmmaking process, requiring resources available for limited periods of time.

First, the Complete Search Model offers a comprehensive approach applicable in general cases, including special cases. A complete search algorithm is used to find an effective filming schedule, by repeatedly choosing plausible nodes and exploring the feasibility. If rejected, the program moves to the next possible node. Source Code (2011) is used as an example to offer insight on the algorithm, where significant resources, characters, sets and scenes are first identified, after which a filming schedule measured in time units is generated. The algorithm is then applied.

The Complete Search Model is advantageous due to its realistic considerations. Its practicality ensures that it is a working model in many cases. However, the time required to derive an effective schedule based on this model depends on the limited time duration given for the film to be completed within.

The Rarity Model focuses on the fact that there may be limited resources during construction. Weightings are first assigned to every node in the graph, based on the range of time available for work, in time units, derived through an arbitrary assignment. This is followed by a renormalisation where the weights are adjusted accordingly, in relation to the node's position in the graph and dependency. From the set of all possible nodes, the rarity of each node is calculated, and the one with the highest rarity will be picked. The program is then run, as showcased in the example of Annabelle (2014).

The Rarity Model is advantageous as it considers the priority of events, assigning weights to determine the relative order. However, it is important to note that the program is unable to determine (or rather, will take an unrealistic amount of time) impossible cases, such as an impractical time duration.

Indisputably, unforeseen circumstances will occur throughout the course of filming, leading to delay in production time and affecting the filming schedule in return. Therefore, we have introduced modifications through three ways: adding additional nodes to the graph in case of refilming or extra takes, changing the resource availability table should there be a change in character or resource availability, and deleting completed nodes and any other edges connected to these nodes should there be unanticipated changes midway through the filming process.

In addition, constraints are determined based on the availability of resources during construction and filming, such as cast presence, sites, budget limitations, and sudden modifications to the original script. To do so, the constraints are ranked to identify what causes the most delay. After applying this to Annabelle using the Rarity Model, we obtained results to generalise patterns that indicate the most significant constraints for generic cases.

1. Restatement

1.1 Introduction

Oftentimes, filming schedules are instrumental in the development of movies, taking into consideration resource availability, time constraints, and dependency. Moreover, unforeseen circumstances such as a sudden change in weather that hinders work progress, or something as mundane as wardrobe malfunctions have the potential to affect the shooting schedule of the entire team. Therefore, it is of utmost importance that we develop a working model closely aligned to the availability of resources, while at the same time adhering to a flexibility should there be last minute changes. We also aim to identify and propose working solutions to cater to constraints causing considerable delays in filming.

1.2 Problem Outline

In this problem, we are required to develop a model which generates a reasonable and flexible schedule given the following types of constraints:

1. Resource availability (availability of stars, resources needed for construction or filming, manpower, sites, etc. under the umbrella of resources);
2. Time (amount of time required for construction or filming); and
3. Dependency (there exist scenes required to be shot in a sequential order, after certain preparations such as constructions).

The model should also be able to accommodate unpredictable changes during the filming such as delay or addition of scenes, change in the availability of resources.

1.3 Assumptions

The models take into consideration the following assumptions:

1. Duration of filming takes into account time taken for travel, break time, multiple takes, makeup, setting up, wardrobe fitting, extras, props, etc.
2. The work of construction of one set or filming of one scene has to be continuous. That is to say, one task that lasts for multiple days cannot be broken into separate parts.
3. There is only one director during the entire filming process, and the director must be present during the filming of every scene. However, the director does not necessarily have to be present during the constructions.
4. Availability dates of all resources throughout the entire duration of filming is known before the schedule is made.
5. Integer units for time are assumed.

2. Model

We have developed two models to generate the filming schedule, both using a directed acyclic graph model to represent the filmmaking process. For both models, filming of each scene requires certain types of resources that are only available during one or multiple periods of time.

2.1 Complete Search Model

2.1.1 Definition of Variables

e_i	last day for resource i to be available
i_r	total types of resources
j_e	total number of events
N_j	event (scene/construction)
p_k	number of resources of type k needed
r_k	resource of type k
s_i	first day for resource i to be available
t_j	time required for event j to be completed
T	current time
u	upper bound for total time required

In the first model, the filmmaking procedure is represented by a directed acyclic graph where each node represents an event (including the start, end, constructions and filming of scenes). The edges indicate the dependency of the events. For example, an edge pointing from node A to B indicates that B can only start after A is finished. For each node N_j , the event needs t_j units of times to be finished.

Resources and their availabilities are not represented in the graph; instead, they are recorded in a separate table. As there may be multiple available periods of time for certain resources (e.g. the casts), we denote the first available period of resource r_i as (s_i, e_i) , the second period as (s'_i, e'_i) , and so on.

2.1.2 Assumptions

Availability dates for all the resources, as well as the total timespan for the whole film production, are assumed to be reasonable such that small delays will not result in insufficient time for all the events to be finished.

2.1.3 Algorithm

A complete search algorithm is used here to find an effective filming schedule:

1. An upper bound u is determined based on the film studio's estimation.
2. Create a list L containing the start time of each node (if it is known). This list will be empty at the beginning. Create a process list P consisting of all the ongoing events (at $T=0$, P is empty). Set $T=1$ and start the algorithm.
3. Check if any event in P is ending at time T . Mark all events in P that end at the current time, T , as finished and remove these from P .
4. Compare the current time T with the upper bound
 - a. If $T \leq u$, continue to step 4;

- b. Else
 - i. if any of the nodes have not been marked as finished, return to the previous step where a node is chosen. Remove the start times of that node from L . The node chosen previously should not be chosen again at the same time unit. Another node should be chosen, or if no other nodes can be chosen, the algorithm will move on to the next time unit, $T=T+1$, without choosing any of the nodes.
 - ii. if all nodes are marked, the algorithm ends. The contents of list L will be printed out.
5. Check the possibility to choose node $N_j, j=1,2,...,j_e$ at this unit of time. It is possible to choose node N_j only when these criteria are fulfilled
 - a. the parent node(s) of N_j is completed or N_j has no incoming edges;
 - b. resources required for N_j are available. That is to say, for every resource required r_k , where p_k number of that resource is needed, check whether the amount of r_k available at time T is $\geq p_k$.
 - c. N_j is not in P and has not been marked as finished.
6. From the set of all possible nodes $A=\{N_j, N_j, ..., \}$ identified in step 4, choose a node and update the availability of the remaining resources.
 - a. If set A is nonempty, update L by setting the start time for the node as T and repeat step 5-6;
 - b. else if set A is empty, move on to the next time unit i.e. $T = T + 1$ and repeat steps 3-6.

2.1.3.1 Source Code (2011)

Source Code (2011) [1] is a science-fiction movie that narrates the story of a soldier who wakes up in someone else's body and discovers that he is part of a government program to identify the person behind the train bombing incident.

To demonstrate an example of the algorithm, we investigate the process of filming Source Code (2011). Firstly, we develop a directed graph, where the movie is broken down into 9 scenes. Scenes requiring construction of set are highlighted in blue, while filming scenes are highlighted in purple. Red is used to label the number of days approximated for the filming to take place. Note that the train in the film is a set construction, so scenes filmed in the train does not have to take place in the train station.

Sites	Cockpit	Interrogation
Train station	Train	Talking scene
Plaza	Car/ Van	Stevens unconscious
Characters	Rutledge's office	Chase Frost
Stevens		Christina chasing Stevens
Christina		Talking to Christina
Goodwin		Christina alone
Rutledge		Finding Frost
Derek Frost		Stevens and Christina walking to plaza
Supporting characters		

Table A: Source Code (2011) Compilation of Scenes

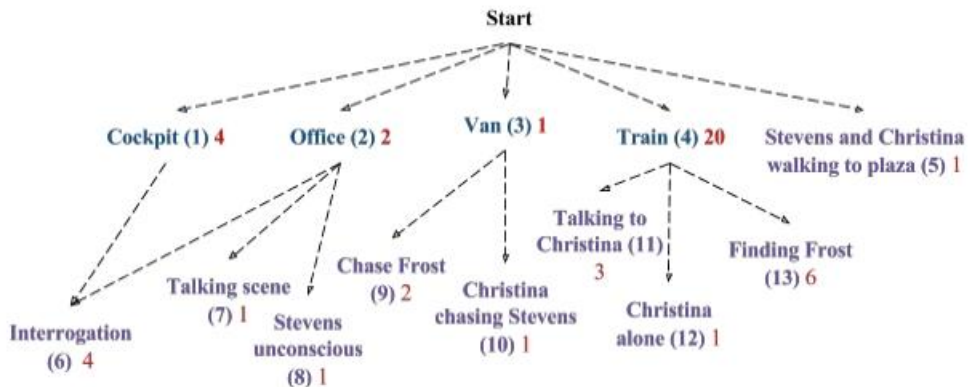


Figure B: Directed acyclic graph for Source Code

Note: Numbers in brackets indicate labels for the nodes. Numbers in red indicate time taken (in units) for event to be completed.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41		
Stevens																																											
Christina																																											
Goodwin																																											
Rutledge																																											
Derek Frost																																											
Train Station																																											
Plaza																																											
Supporting																																											

Figure C: Availability dates of resources for Source Code

Scene	Characters Involved
Interrogation	Stevens, Goodwin