For office use only	Team Control Number	For office use only
T1 ________	**2015001**	F1 ________
T2 ________		F2 ________
T3 ________		F3 ________
T4 ________		F4 ________

2015

The International Mathematical Modeling Challenge (IM^{2}C)
Summary Sheet
(Attach a copy of this page to the front of your solution paper.)

While most people have watched movies before, many do not know about the immense amounts of planning that go into the production of a film. Specifically, the creation of a filming schedule involves the consideration of multiple constraints. These include the availability of actors, resources, and locations, as well as the preparation of essential elements and the time it takes to shoot on each site. Our method takes these constraints as well as industry regulations into consideration, which no model in the literature to date has done.

We went through multiple strategies before settling on our final model. First we created a model that optimizes time and runs through all possible permutations. This uses logic easily portable into a programming language like Python. Next, we created a cost-optimizing model using Microsoft Excel which had a well-developed and versatile matrix component. We combined the powerful matrix setup of the Excel model with the programmatic aspects of the first model to create an enhanced, primary model.

Our main model uses Integral Linear Programming (ILP) with two heuristics we developed. The first allows for local optimization using pairwise interchange while the second maximizes matches between the actor/resource/prerequisite availability matrix and requirement matrix. This intelligently assigns filming days to calendar dates in a way that capitalizes on human and physical resources as efficiently as possible. We then use pairwise interchange to optimize cost without significantly increasing processing time by only considering cases that are likely to turn out to be both feasible and optimal. We confirm the validity of our model through test cases that showcase the capabilities of both our match maximization and pairwise interchange heuristics.

A very important consideration for the studio is the sensitivity of our model/schedule to unexpected problems, changes, or delays. Our model helps movie studios minimize the effect of such disruptions. We employ sensitivity analysis to determine that actor and resource availability rather than set construction and content generation requirements is the most sensitive part of the model. We can quantify this sensitivity for the movie studio, which has many applications. For example, when the studio insures different parts of their production, it knows how much value to assign to each one.

Another critical consideration is real-world conditions. One of our model's greatest strengths is its cognizance of real-world conditions, which allow it to function not just in a purely mathematical context, but in the context of the film industry, where the regulations of unions, for example, could render another model useless.

Our model thus quickly, efficiently, and accurately produces a schedule movie studios can feel confident using as they plan their future productions.

From Script to Screen: Scheduling the Production of a Movie

Team 2015001

Introduction

The glamour of the entertainment industry often hides the extensive amount of planning required for the filming of a movie. Multiple constraints must be met to construct the most optimal and feasible schedule. These constraints include:

- Availability of actors, locations, and resources
- Preparation of key elements, such as the constructions of sets, that must be completed before shooting certain scenes
- Time necessary to film at specific sites and locations
- Time for reshooting after the primary production period ends

But a movie studio must do much more than just satisfy these constraints: it must also do so in the most cost-effective way possible.

This problem can be classified as a resource constrained project scheduling problem (RCPSP) because it considers resources of limited availability and activities of known duration. RCPSP problems are considered NP-hard, meaning that a solution cannot be found in polynomial time [Artigues 2008]. This justifies the use of heuristics in order to come to good solutions in a reasonable amount of time.

In order to create a model that best satisfies realistic circumstances, it is essential to account for the real-world conditions of the industry. For example, the Screen Actors Guild (SAG), representing 160,000 movie actors all over the world, imposes very stringent regulations on movie producers. Most prominent actors are affiliated with SAG, and their affiliation requires them to only work with movie producers that sign agreements with SAG. As a result, the vast majority of movie companies sign these agreements to gain access to these actors and therefore must comply with SAG regulations. A model ignoring SAG requirements would therefore be of little use to anyone, and one of our model's strengths is its consideration of these requirements. These requirements are detailed in the Assumptions section.

Cheng, et al. [1993] pioneered the treatment of the actor availability problem by seeking to reduce talent hold time (that is, the time actors are paid but are simply being "held" and not used) and Shyu, et al. [2000] followed up on Cheng's work. But both models have substantial problems. Neither takes into account enough real-world conditions to make it practically useful, they can turn up empty-handed when a solution is there, and the computational power required to carry out the PVM model of Shyu is beyond the scope of the resources of a high school environment, as well as those of the department of a movie studio dedicated to scheduling.

Assumptions

1. The film is neither a documentary nor an animated film.

Justification: Documentaries and animated films require different resources and planning from what is presented in the problem. For example, actor availability is not especially relevant to documentaries, and set construction is irrelevant to animated films.

2. Screenwriting is finished, and we are only focusing on production.
   Justification: It would not be appropriate to design a schedule for the filming of a movie that has not yet been written. Considerations like actor availability and resource requirements change between the initial and final drafts of the script.
3. We are only concerned about the availability of lead actors, and the availability of secondary actors is immaterial. Similarly, there will not be a shortage of workers to do work like the preparation of sets.
   Justifications: 1. While a movie might need specific stars for popularity, the identity of the other actors is not important to the movie. Therefore, the movie studio can and will simply choose whichever actors for the secondary role are available, and as a result secondary actors are available all of the time from a mathematical standpoint. There are always agencies for large studios to find people to prepare sets and act in minor roles. 2. The problem asks us to consider the availability dates of the "stars" of the film only.
4. Actors are employed in "blocks" of time and paid during the entire duration of shooting, regardless of whether they work the whole time.
   Justification: This is a result of the SAG constraints mentioned in the introduction. Since the movie company in the problem is described as a "large studio," we make the assumption that it is bound by SAG rules (because it is virtually impossible to become a large studio without the help of guild-affiliated actors). The most relevant consideration in the SAG rules is that *actors must be paid for hold time*; that is, if an actor is needed on Monday, Tuesday, and Friday, the movie producer must pay the actor as if s/he had been working on Wednesday and Thursday, too, even if the actor did nothing for the producer during that time. SAG covers much of the world, although Bollywood actors in India, for example, organize under their own union, FWICE, and enjoy similar benefits, so we assume for the purposes of this problem that the proportion of actors not covered by SAG, FWICE, or other similar unions and guilds is negligible [Tellychakkar 2010].
5. Time needed to shoot the movie is related to the cost of production. Therefore, minimizing time saves money.
   Justification: As production drags on, more money is needed to pay for basic necessities such as food and housing as well as compensating for the crew members' time.
6. The filming of the scenes must be disjunctive, meaning two scenes cannot be filmed at the same time.
   Justification: There is only allowed to be one director for each movie [Directors' Guild of America 2011] and the director must be present at the filming of each scene so that he or she can direct the artistic and dramatic aspects of the film. Furthermore, the director is responsible for directing the actors and cinematographers and thus the director must always be on set to do so [Media College, nd.]. If there can only be one director and the director must be on set to direct, only one scene can be filmed at a time.
7. The amount of time required to film each scene is generally fixed before filming begins. Therefore, we should aim to shorten the the total production time by reducing the delay in between the filming of scenes and not the scenes themselves.
   Justification: The time required to film each scene is dependent upon the circumstances of the shot, which we cannot control. It is reasonable that the expected time for each scene is an accurate prediction for the actual filming time. Therefore, we take the time required to film each scene as constant.

Model

Overview:

The scheduling problem is not new to the literature; it has been attempted before. In this paper, we take two preliminary approaches to the problem presented and then combine them to produce a third, primary model. Of the two preliminary models, one optimizes time and takes an approach relying on computer programming while the other optimizes cost and takes a more algebraic approach using Microsoft Excel Solver. Both preliminary models are available for the studio to use (for instance, if optimizing time is more important than minimizing cost, the time-based model may be utilized), but we focus on the final, primary model, which is a synthesis of the two preliminary models. We combine the programming aspect of the time-based preliminary model with the matrix-based portions of the cost-based model to produce a new model based on integral linear programming using local optimization. We create two heuristics that vastly improve processing time while still ensuring we have a high chance of producing the most optimal solution.

Preliminary Model #1 (Time):

Question 1:

There are many significant factors that affect the total duration of production, including time spent traveling, time spent creating sets, and time windows during which certain resources are not available, creating windows of time where nothing can be done (lag time).

The problem asks us to consider the availability of 6 types of necessities: actors, sites, sets, resources, computer generated content, and physically constructed items. Given the nature of the availabilities, we can categorize them into 2 groups.

First, there are necessities that will be available in an on-and-off basis over time; these include actors, sites, and items such as helicopters, which can only be rented at certain times. We will label all such necessities, with the exception of actors, "resources." The second group of items will be termed "prerequisites," because they must be prepared before the scenes requiring them can be shot. Furthermore, they will always be available once enough time has elapsed. This category of necessities will include sets, computer generated content, and physically constructed props.

The expression that gives the total time for the production of a film is:

$$\Sigma\, time = \Sigma\, total\ time\ filming + \Sigma\, lag\ time$$

We wrote pseudocode for this model, which can be represented by the flowchart shown in Figure 1.

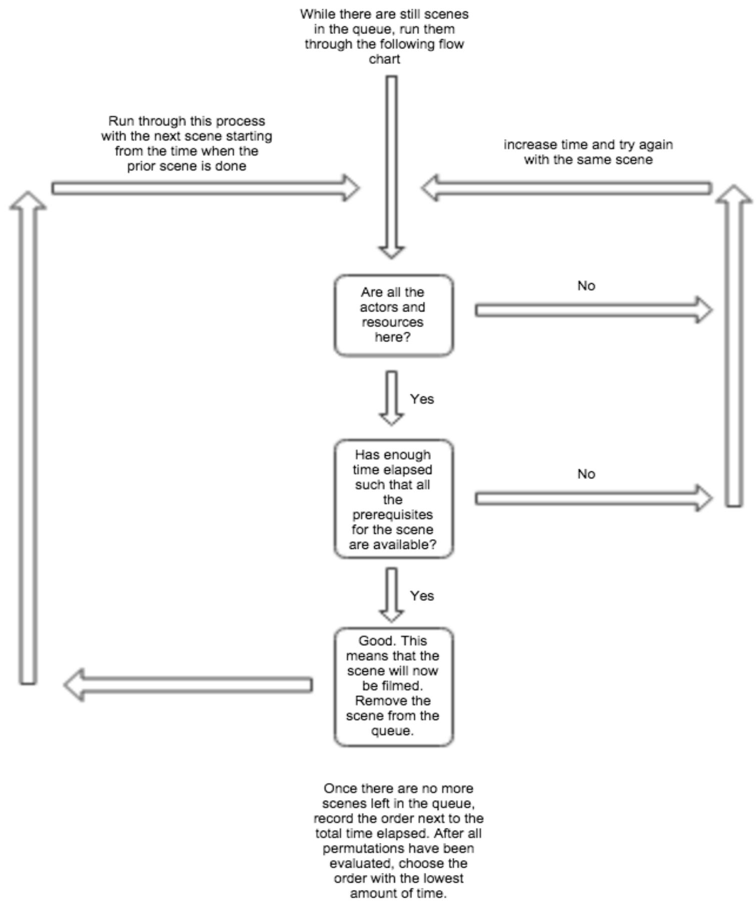


Figure 1

While the logic of this pseudocode holds no matter the number of scenes, actors, etc., using it at scale can become computationally impractical. The average movie has 40-60 scenes, and if there are 50 scenes, the algorithm would have to check 50! sequences. This would take very long for even the fastest computers.

This algorithm, once given all the required inputs, such as actor availability as a function of time, etc., will be able to effectively produce the most time-efficient order for filming the scenes by reducing lag time.

Let us apply this to a simple scenario to display the strengths of this model. Assume a movie has two scenes, two actors, and a single set that must be built. We assign random values to the availability functions and the times.

Let:

$$A_1(t) = \begin{cases} 0, & t = 2,3,5,7,9,10 \\ 1, & t = 1,4,6,8 \end{cases} \qquad R_1(t) = \begin{cases} 0, & t \text{ is odd} \\ 1, & t \text{ is even} \end{cases}$$

$N_{s1}=\{\}$
$P_{s1}=\{A_1\}$

$P_{s2}=\{A_1, R_1\}$
$N_{s2}=\{f_1\}$
$T_{f1}=3$
$q=\{s_1, s_2\}$

$T_{s1}=T_{s2}=1$
$t = 0$

If one films s_1 first, then the requirements of P_{s1} will be fulfilled on day 1, and the requirements for P_{s2}, including N_{s2}, will be fulfilled on day 4 because N_{s2} requires 3 preparation days. Thus, the value printed by the algorithm will be (s_1s_2, 4), with "4" indicating a four day shoot and "s_1s_2" indicating that s_1 should come before s_2.

However, if one starts with s_2, then the requirements of R_{s2} can only be fulfilled on day 4, leaving s_1 to be filmed on day 6 because the first day for which actor 1 is available after day 4 is day 6. This yields (s_2s_1, 6), with 6 indicating a six day shoot and "s_2s_1" indicating that s_2 comes before s_1.

Clearly, given the constraints of this scenario, filming s_1 first is advantageous. By finding the lag times of all permutations of scene filming order, the algorithm will eventually achieve the most time efficient sequence. This logic will hold as long as the number of permutations the computer must calculate is reasonable. Because the time it takes to film scenes is constant,

$$\text{delay time} = t - \text{total time filming}$$
$$\text{total time filming} = T_{s1}+T_{s2} = 2$$

The total delay time for this scenario would thus be 4 days. By minimizing *t*, we are minimizing delay time, so the lowest *t* will provide the lowest delay time (see Assumption 7).

A Gantt Chart for the optimal schedule would then resemble that shown in Figure 2:

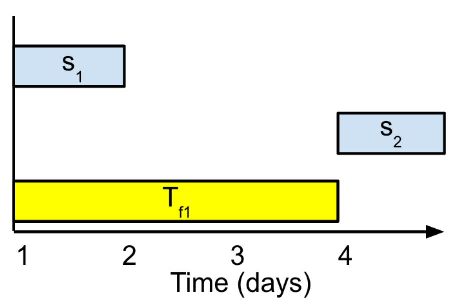


Figure 2

S_1 represents scene 1, and its location in the Gantt Chart (above t = 1) shows that it is filmed during day 1. T_{f1} represents the preparation time required for f_1, which is shown by its length in the chart ($\Delta t = 4 - 1 = 3$, meaning its preparation time is 3 days). Because S_2 cannot be filmed until f_1 is completed, S_2 must wait until the required preparation time (3 days) has elapsed. Therefore, S_2 appears above t = 4, after T_{f1} ends, and is filmed during day 4.

Question 2:

This model is also capable of effectively dealing with delays. If an actor or resource has an unexpected change in availability, it would need to be reflected in a new availability function. However, scenes that have already been filmed would not be affected and therefore would no longer remain in the queue for the new algorithm that takes the delays into account. Furthermore, the amount of time elapsed before the delay/change in availability must be taken into account when calculating whether preparation time for the prerequisites has already been fulfilled, assuming the prerequisites have been in development the whole time.

To illustrate, assume that after day 2, actor 2 injures himself and requires 4 days of rest in order to recover. His new availability function would be:

$$A_2(t) = \begin{cases} 0, & t = 1, 3, 4, 5, 6, 7, 9 \\ 1, & t = 2, 8, 10 \end{cases}$$

Now, by revising the algorithm to meet these new needs, we get:

$$t = 2$$
$$q = \{s_2\}$$

This means that two days have passed, and the only scene in the queue left to film is scene 2. Now, s_2 can only be shot only on day 8. The Gantt chart for this new scenario is shown in Figure 3.

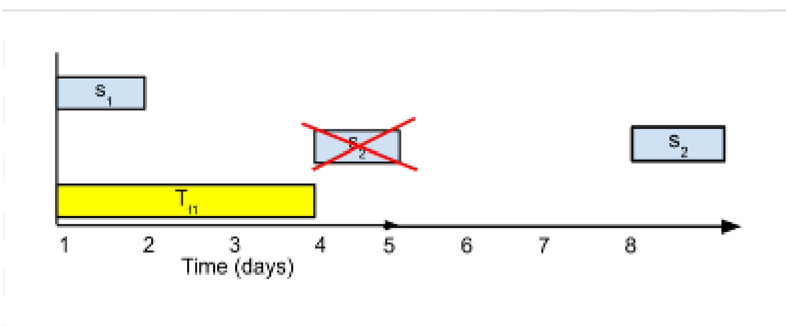


Figure 3

Notice that S_2 has been moved to $t = 8$ because 2 days have already elapsed (meaning the prerequisite is already ready), but the next day for which all requirements of S_2 are fulfilled is day 8 (since actor 2, required for S_2, is not ready until $t = 8$).

Question 3: The investigation of the most important constraints in the schedule is not directly related to the model itself. Thus, we leave a sensitivity analysis to our discussion of the primary model.

Preliminary Model #2 (Cost with Microsoft Excel):

Question 1:

In this model, we take the more standard approach of optimizing cost. Our first method for modeling the total cost involves creating a table of givens with actor availability, resource availability, actors required, filming locations, resources required, preparation time (for prerequisites) required, and average cost of changing filming locations.

We generate a matrix of the days each actor gets paid using Microsoft Excel functions. Each column of the matrix represents an actor and each row represents a day. The matrix is binary with 1 indicating a paid day and 0 indicating an unpaid day (all according to SAG constraints; see Assumption 4). For the first and last rows of the matrix only, the actor only gets paid if s/he is required to be filming that day (as opposed to all other days, in which the actor might be paid even if s/he is not working), so the function for those days is

=IF([cell reference of actor requirement for day 1]=1,1,0),

returning 1 if true and 0 if false.

From the second to the second-to-last row of the matrix, an actor gets paid for all days during which they film and days in between filming days (hold days). Therefore, the function becomes

=IF(OR([cell reference of actor requirement for day]=1,AND(SUM([cell reference of actor requirement on first day]:[cell reference of actor requirement day before])>=1,SUM(cell reference of actor requirement day after:cell reference of actor requirement on last day)>=1)),1,0),

again returning 1 if true (either the actor is required to work on that date or the actor is required to work during days both before and after that date, so must be paid anyway) and 0 if false.

The product of the matrix of actor pay days (dimensions: a x b, with a being the number of days and b being the number of actors) and the column matrix of actor pay rate (dimensions: b x 1, with b being the number of actors) yields the matrix showing the total cost of hiring actors each day (dimensions: a x 1). The sum of this column matrix is the total cost of hiring actors given a certain schedule.

To determine the additional cost resulting from changing locations, the movie studio enters the average cost for moving from one site to another. For the location column in our table, each number represents a different location. We create a column for moving costs based on the fact that each change in location (represented by a change in number in our model) entails added cost, so the Excel function is

=IF([cell reference for cell before current one in site column]<>[current cell reference in ,[average cost of a change in shooting location],0),

outputting the average cost of changing location if true (there is a change in location) and 0 if false (no change in location).

To determine if doing a day of filming on a certain date is possible, we must ensure that all required actors and resources are available and that any time requirement for prior prop construction/computer content generation has been met. In our table for actor availability, for each day, “1” denotes available while “0” denotes unavailable. In our table for actors required per day, “1” denotes required while “0” denotes not required. Thus, if the values of the cells in our table of actor availability are greater than or equal to those of the corresponding cells in our table of actors required, the actors are able to perform on the days in which they are needed to do so. Our tables of resource availability and resource requirements work the same way. To allow the preparation crew sufficient time to prepare what is needed, the number of days passed since the beginning of filming must be greater than or equal to the amount of time needed. Therefore, the binary Excel function for the possibility of filming on each day of the schedule is

=IF(AND([cell reference of first actor availability]>=[cell reference of first actor requirement],[cell reference of first resource availability]>=[cell reference of first resource requirement]**,[cell reference of necessary preparation time]<=ROW([current cell reference])-ROW([first cell reference])),1,0),*

**Adjustable for as many actors as needed*
***Adjustable for as many resources as needed*

This function returns 1 if true (all available and prepared) and 0 if false (any actor/resource unavailable or preparation time unfulfilled).

The total variable cost can be calculated by summing the actor costs per day and the costs from a change in location. Director costs and other fixed costs (costs that only vary by total time and not by circumstance) can be ignored because this model only re-orders by day and all permutations of a certain number of days amount to the same total time.

To optimize the order of filming days, we must minimize the total cost given that filming in the scheduled order is possible.

We include the sample table and test case in the Appendix. We find that for this case, solving manually for the minimal cost by computing each permutation results in the exact same answer given by the model: the studio should schedule filming in the order day 3, day 1, day 2, coming to a total cost of $2814.29. This manual validation proves that our Excel model works.

Question 2:

In order to accommodate for unexpected delays, values in actor and resource availability and preparation time can be altered according to the situation. For instance, if an actor injures himself/herself and breaks a prop in the process, both the actor and the resource (the prop) would be unavailable for the next few days. Furthermore, when considering the new optimal schedule using this model, one should disregard all days which have already been filmed. After all changes are made, the schedule should be optimized once more using our model.

To illustrate, assume actor 2 falls and breaks something required for day 3. This causes actor 2 to be unavailable for day 2 (the first row in the table because day 1 has been removed) and increases preparation time for day 3. Thus, the table would become that shown in Table 1 (compare this with the table detailed in the test case in the Appendix):

Availability					
Actor 1	Actor 2	Actor 3	Resource 1	Date	Preparation Time
1	0	0	0	2	0
0	1	1	1	3	4

Table 1

Notice that day 1 values have been removed to indicate that filming for that date has completed and is thus not affected by the accident.

Question 3: As with the first preliminary model, we leave discussion of this question to our analysis after our primary model.

Primary Model - Integral Linear Programming (ILP) with Match Maximization and Local Optimization/Pairwise Interchange Heuristics

Question 1:
<u>Matrices</u>
A = Actor, resource, and location availability matrix
R = Actor, resource, and location requirement matrix

C = Pay rate of each actor (row matrix)
D = Whether or not actor is paid on a given day (row matrix with binary entries)
Q = Total actor hiring cost (row matrix)
S = Cost of location changes (row matrix)

N = Resultant matrix with pre-filming and post-filming lag days
Y = Resultant matrix without pre-filming and post-filming lag days
Z = Row matrix of site numbers without lag days

<u>Variables</u>
m_a = number of actors
m_r = number of resources
m_p = number of prerequisites
x = total number of days for which actor/resource/prerequisite availability is known with $x > m$
m = total number of days filming
n = total number of days in production with all lag days, including lag days before any filming starts and after all filming ends
k = total cost to the studio that our schedule can control (cost to pay actors and cost to change location, plus a fixed daily cost for expenses like food, etc.)
p = the typical cost to change location, which varies widely depending on the size of the production, since moving a larger staff requires more money; we leave this for the studio to input
q = total number of days in production, including lag days during filming but excluding lag days before any filming starts and after all filming ends

<u>General</u>

Numbers are used to indicate fixed calendar days while letters are used to indicate flexible filming days. Thus, day 5 always happens before day 6 and after day 4. But days a, b, and c can happen in any order; they are simply tied to a certain scene, which in turn is tied to a certain location number and to certain actors. Based on actor/resource availability on different calendar dates, the lettered filming days can be moved to accommodate their schedules.

<u>Description of Model</u>

Where the film production problem has been attempted in the literature, a branch-and-bound approach has been used. This is because the film scheduling problem is an NP-hard problem (as proven by Cheng, et al.), so some kind of shortcut must be used to make solution of the problem computationally practical. Initially, we developed a model that uses the same branch-and-bound method as Cheng, along with the most attractive route and pairwise interchange heuristics he uses, but we realized that in many instances this model may turn up empty-handed where it should not: a solution might be possible, but the branch-and-bound method may prune the branch containing that solution early on, turning up with no solution later. To solve this problem, we abandoned the branch-and-bound method and the most attractive route heuristic, developed our own local optimization method, expanded on the pairwise interchange heuristic to better meet our needs, and developed a new match maximization heuristic. While Cheng's pairwise interchange heuristic simply switches the first and last nodes in the tree, our improved heuristic interchanges each day with every other day, minimizing our chances of missing the optimal solution. Our match maximization heuristic intends to maximize matches between actor availability and actor requirements: we make sure actors are available when they are needed, but we also try to ensure that we make the days that actors are not needed the days that they are also not available, since it is undesirable--possibly wasteful--to have an actor available but unneeded. By intelligently, rather than randomly, assigning matches, we reduce lag days and improve efficiency. Our local optimization method minimizes the number of schedules that must be tested while preserving accuracy, providing a greater chance of finding a solution, while still being operable on the resources a movie studio will have (the reduction in the number of schedules that needs to be tested means that the model does not require supercomputers to work).

While the branch-and-bound model using the most attractive route and basic pairwise interchange heuristics optimizes a schedule based on actor hold time only, our local optimization model accomplishes the same goals with more realistic application by incorporating many variables (e.g., actor availability, location of filming, and preparation time, etc.) that Cheng, for example, was unable to include:

> "*In general, a variety of other criteria will also be involved in the decision process...Other considerations such as restricted availability of some talent, setup costs for scenes, artistic constraints, and so on are ignored in this paper.*"

First, we generated matrices **A** and **R**. Matrix **A** has the dimensions $m_a + m_r + m_p$ by x. A sample matrix **A** with 2 actors, 1 resource, 1 prerequisite that is ready in 3 days, and 7 days of known availability might look like this:

$$\begin{array}{l} \\ \text{Actor 1} \\ \text{Actor 2} \\ \text{Resource 1} \\ \text{Prerequisite 1} \end{array} \begin{array}{c} \begin{array}{ccccccc} \text{Day 1} & \text{Day 2} & \text{Day 3} & \text{Day 4} & \text{Day 5} & \text{Day 6} & \text{Day 7} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Matrix 1

Recall that the days (1, 2, 3, 4, 5, 6, 7) represent set dates in that order. Actor 1 is only available on days 1, 2, 3, and 4; actor 2 is only available on days 2, 5, 6, and 7; resource 1 is only available on days 3, 4, and 5; and prerequisite 1 will be done in 3 days; thus, all entries after day 3 for the prerequisite will be 1's because it will never become "unready."

Matrix **R** has the dimensions $m_a + m_r + m_p + 1$ by m. The extra row represents the filming location number (nonbinary) for each day. A sample matrix **R** with 2 actors; 1 resource; 1 prerequisite that can be made ready in 3 days and is needed for day c; 5 days of filming; and 2 filming locations (location 1 and location 2) might look like this:

$$\begin{array}{l c} & \begin{array}{ccccc} \text{Day a} & \text{Day b} & \text{Day c} & \text{Day d} & \text{Day e} \end{array} \\ \begin{array}{l} \text{Actor 1} \\ \text{Actor 2} \\ \text{Resource 1} \\ \text{Prerequisite 1} \\ \text{Filming Location} \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 \end{bmatrix} \end{array}$$

Matrix 2

The days (a, b, c, d, e) represent filming days whose order we are trying to optimize.

We now insert our constraint:

$$A_{ij} \geq R_{ij} \text{ if } \quad i \in [1,\ m_a + m_r + m_p] \text{ and } j \in [1,\ n]$$

This constraint ensures that the actors are always available when they are needed on set. (For the parts of the matrix where i and j lie in the intervals given, the entries are binary: the entries in **A** are 1 when the actor/resource/prerequisite (A/R/P) is available and 0 when the A/R/P is not; similarly, for **R**, the entries are 1 when the A/R/P is needed and 0 when the A/R/P is not.) If the entry in **R** is 0, then it does not matter what the entry in **A** is: since the A/R/P is not needed, it is immaterial whether the A/R/P is available. If, however, the entry in **R** is 1 and the A/R/P is required, then the A/R/P must also be available. This constraint of our bounding function allow schedules with infeasible assignments to be discarded.

From here, we run our ILP method using total cost as our bounding function combined with the match optimization and pairwise interchange heuristics we developed to save time while increasing the probability of obtaining a reasonable answer.

We want to maximize the number of matches in order to capitalize on the actors and resources when they are available while trying to match the days they are not needed with the days they are not available.We first test each column (flexible lettered filming day) in **R** against each column (fixed numbered calendar date) in **A** and find whether the pairing fulfills all the constraints. If the constraint is not fulfilled, we discard that pairing. We then find the number of matches the remaining days have with the selected date. For example, when we compare row 1 of our sample matrix **R** (Matrix 2) with row 1 of our sample matrix **A** (Matrix 1), we see that $R_{13} = 0 = A_{15} = 0$, $R_{23} = 1 = A_{25} = 1$ and $R_{33} = 1 = A_{35} = 1$. All of the entries match in value. Therefore, the test of R_{j3} against A_{j5} has a match number of 3. We continue testing all columns of **R** against the first column in **A** and choose the column in **R** with the greatest number of matches to the first column in **A** as the temporary (it may later change with pairwise interchange) filming day for date 1. This is the maximizing matches heuristic. If there are multiple days that share the same (maximum) number of matches, we arbitrarily choose one day. We continue match maximization until all days in **R** are assigned to a date in **A**. If for a certain date in **A** none of the days in **R** are able to fulfill all of the constraints, we designate that day as a lag day during which no filming can occur. The resultant matrix, **N**, should have n columns (filming days + lag days).

We now create a function for calculating the total cost, k, which we must optimize. We will first trim matrix **N** to remove any lag days at the beginning or end of the matrix. For example, if a matrix **N** looks like this:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 & 0 \end{bmatrix}$$

Matrix 3

We will discard the first and last lag day columns to create the new matrix **Y** with dimensions $m_a + m_r + m_p + 1$ by q:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

Matrix 4

Let matrices **C** and **D** represent the pay rate of the actors and the days for which actors receive pay, respectively. Recall that hold time means actors receive pay from their first day of work until their last. Matrix **C** is a row matrix where each entry is the amount each actor earns per day. It includes extra zero terms (non-actor wages, so set to zero) such that its dimensions are 1 by $m_a + m_r + m_p$ to allow for multiplication with matrix **D**. Matrix **D** is a binary matrix (1 if the actor needs to be paid that day; 0 if not), and its entries are defined as:

$$D_{1j} = \begin{cases} 1, & Y_{ij} = 1 \text{ or } (\sum_{a=1}^{j-1} Y_{ia} \geq 1 \text{ and } \sum_{a=j+1}^{q} Y_{ia} \geq 1) \\ 0, & \text{otherwise} \end{cases} \quad \text{with } i \in [1, m_a] \text{ and } j \in [1, q]$$

Formula A

This indicates whether an actor is to be paid or not. If $Y_{ij} = 1$, then the actor is required, and then s/he must be paid. If, however, the sum of the (binary) entries before *and* after are both ≥ 1, then the actor is required both before and after the given day, and even though the actor is not required on that day, s/he must be paid anyway, making the entry in matrix **D** (whether the actor is to be paid or not) equal to 1. If the actor is not required on that date and is not required either before or after as well, then the actor need not be paid and the entry in matrix **D** is 0.

We then create two new variables. k represents the total cost to the studio that we can control, while p represents the average cost of changing locations. The model will take location change spending into account to avoid changing filming locations excessively, and if one location requires multiple days of filming, the model will try to make those days consecutive if possible to cut costs. We first generate a row matrix **Z** that contains the location numbers of the days of our schedule in order by the following function:

$$Z_{1j} = \begin{cases} \varnothing, & Y_{(m_a+m_r+m_p+1)j} = 0 \\ Y_{(m_a+m_r+m_p+1)j}, & \text{otherwise} \end{cases}$$

Formula B

Notice how there is no entry in matrix ***Z*** *when the* ***Y*** *= 0. matrix* ***Z*** *will be shorter than* ***Y****.*

We set p as a reasonable constant (the cost every time location is changed: for items like airline tickets, equipment moving costs, etc.), and define a new row matrix **S** that describes the number of site changes along with the cost of those changes. An element in **S** is defined as:

$$S_{1j} = \begin{cases} p, & Y_{1j} - Y_{1(j-1)} \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Formula C

This means that if the location is different from one column to the next, a cost p is incurred; otherwise, the location change cost is 0 because the movie is being filmed in the same place.

To calculate the total cost of film production, we first calculate the cost from changing filming location by calculating the sum of the entries in **S**. (Recall that for each change in location, a fixed cost p was entered in **S**. If there are two location changes, two entries in **S** will be p while the rest will be 0. Thus the sum of the entries in **S** will be $2p$.) We also need to incorporate constant costs per day (including director pay, crew-member pay etc.), with a being the average total cost of production per day and q being the number of days filming plus the number of lag days in the shooting period. We then need to calculate the total cost for hiring actors. The product of matrix **C** and matrix **D** (binary--whether or not the actor is paid--dimensions m_a by q) produces the total cost of hiring actors each day. We will call this the total cost matrix, **Q** (dimensions 1 by q). The sum of all of the elements in the row matrix **Q** is the total cost of hiring all actors. The sum of actor hiring costs, constant daily costs, and set moving costs models the total cost during the filming period.
Thus, we can define total cost, k, as such:

$$k = \sum_{j=2}^{n} (S_{1(j-1)}) + (a \times q) + \sum_{j=1}^{n} Q_{1j}$$

Formula D

Now that we have a feasible schedule and cost calculation, we apply the pairwise interchange heuristic. This avoids calculating the cost k for all permutations and entails minimal loss of accuracy. Suppose our preliminary schedule involves days [a, 0, b, c, 0, d] in that order ("0" indicates lag days). We would first interchange day a with all of the other days (including lag days) and verify that each interchange fulfills the constraints. We then calculate k for each qualified scenario. If [c, 0, b, a, 0, d] yielded the minimum k, for example, then we would discard all other scenarios and avoid spending all the processing power needed to compute the k for all of those (comparatively unfavorable) scenarios. We then interchange day b (notice that we skip over the lag days, since they are functionally non-unique) with all of the other days and find the configuration that fulfills the constraints with the minimum *k*. We repeat the process with c and d. We always start with filming days and interchange each filming day with all other days (including lag days). In order to increase our chances of reaching the optimal solution, we will run this interchange cycle (one cycle = interchanging each number with all other numbers once) up to 20 times (4 variables, 5 interchanges each), each time using the minimum cost schedule from the previous cycle as a standard. However, if the minimum cost converges before the 20 cycles are over, we can stop. These steps are illustrated in Figure 4.

Figure 4

This will ensure we test a small fraction of the total number of scenarios while still producing an optimal or nearly-optimal result.

Reshoots:

Reshoots, or pick-up shots, are shots or scenes shot after principal photography has concluded [Media College, nd]. Because the nature of reshoots is unpredictable, we would suggest that at least an extra week of pick-up shots be planned for all three models. Scheduling an extra week after production is over that would be flexible and subject to change is optimal because, according to *The Complete Film Production Handbook* [Honthaner 2010]:

> *"Reshoots are sometimes scheduled for shortly after the completion of principal photography or may be months later. They can last a day or two or a matter of weeks. They can entail a local shoot that's fairly routine and easy or require traveling, packing, shipping and working on distant locations."*

In other words, incorporating reshoot time into the main schedule (i.e., while filming is still happening) would be inappropriate because this is inconsistent with the real conditions of the film industry. As such, we simply add time to the end of the filming schedule for reshoots, just as it is done in the real world. It is also inappropriate to assume that we can model the amount of time needed for the reshoot, because this varies wildly, as evidenced by Honthaner's real-world experience. Instead, we simply assume a reasonable amount of time to add to the end of the schedule. It is important to note that the SAG regulations regarding hold time and actor pay do not apply to reshoots. Thus, this suggestion applies to all three models and actors are not paid for the period (perhaps many months long) between when filming ends and when reshoots begin: a new contract is created for the reshoot.

Our test case, provided in the appendix, shows a simple calculation and optimization of cost. There are three scenes to be filmed and the arrangement that suited them best spanned over 3 days and cost $7,800 to film. This demonstrates that our model is able to find a solution and then optimize cost.

The logic our model follows can be expanded to accept as many scenes as needed while still requiring minimal computing power, producing a viable schedule more often than existing models in the literature while taking far more factors into consideration and working at higher speed.

Question 2:

In the event of delay, we will remove the days for which filming has already finished from our availability and requirement matrices **A** and **R** and update the availability of the actors/resources or add required preparation time, depending on the cause of the delay. For example, if we find out on day 2 that preparation time for prerequisite 1 must be extended to 5 days due to an unexpected change in weather, matrix **A** would be described by Matrix 5:

$$\begin{array}{l} \\ \text{Actor 1} \\ \text{Actor 2} \\ \text{Resource 1} \\ \text{Prerequisite 1} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Day 2} & \text{Day 3} & \text{Day 4} & \text{Day 5} & \text{Day 6} & \text{Day 7} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Matrix 5 (Compare with matrix 1 to see the differences caused by delay)

while matrix **R** would be described by Matrix 6:

$$\begin{array}{l} \\ \text{Actor 1} \\ \text{Actor 2} \\ \text{Resource 1} \\ \text{Prerequisite 1} \\ \text{Filming Location} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Day b} & \text{Day c} & \text{Day d} & \text{Day e} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 2 \end{bmatrix} \end{array}$$

Matrix 6

With these two matrices updated, we would run the model again and generate a new schedule accounting for the change.

Question 3:

We use a simulation of our primary (ILP) model to determine the constraints that will cause the longest delays in the case of an unexpected problem. Utilizing the availability matrix **A** and the requirement matrix **R** from our model, we used the Oracle Crystal Ball program to analyze the sensitivity of the different parts of our model. This sensitivity analysis applies to all the models because it is more relevant to the final schedule than the algorithms that produced it.

In order to introduce uncertainty (the possibility of a problem occurring) to our availability matrix, we change the probability of an actor or resource being available from 1 to 0.8. Even if an actor is said to be available on a certain day, a problem could occur and therefore we are, in reality, never

100% confident that the actor will be able to film on the specified day. For the amount of preparation time (prerequisite) necessary for certain days of filming, we introduced uncertainty by incrementally increasing the probability of the prerequisite being fulfilled (see the last row of the matrix in matrix 7). In our example, the prerequisite is supposed to be done by Day 3 and not before, but to analyze sensitivity we posit that it has a low chance of being done on just Day 1, a higher chance on Day 2, and a 90% chance on Day 3 (by which point it is definitely supposed to be done - at least in theory). The whole matrix as we enter it into Crystal Ball to analyze sensitivity appears as follows:

$$\begin{bmatrix} 0.8 & 0 & 0.8 \\ 0.8 & 0.8 & 0 \\ 0 & 0.8 & 0.8 \\ 0.2 & 0.4 & 0.9 \end{bmatrix}$$

Matrix 7

We then create a requirement matrix **R** with the days for actors, resources, and any prerequisites must be present. The matrix below is an example requirement matrix.

$$\begin{array}{l c} & \begin{array}{ccc} \text{Day a} & \text{Day b} & \text{Day c} \end{array} \\ \begin{array}{l} \text{Actor 1} \\ \text{Actor 2} \\ \text{Resource 1} \\ \text{Prerequisite 1} \end{array} & \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Matrix 8

We create a Poisson distribution for lag time (the amount of days spent "at rest") if the constraints cannot be fulfilled on a certain day. Use of a Poisson distribution is appropriate here because a delay is always caused by an event; the Poisson models the probability of a certain number of events occurring. The mean of this Poisson distribution is 1 (probability of a 0-day delay is part of the distribution but not shown on the graph because it would mean no delay at all) because it is most likely that the constraints will be fulfilled the day after rescheduling. The Poisson distribution from Crystal Ball can be seen in Figure 5.

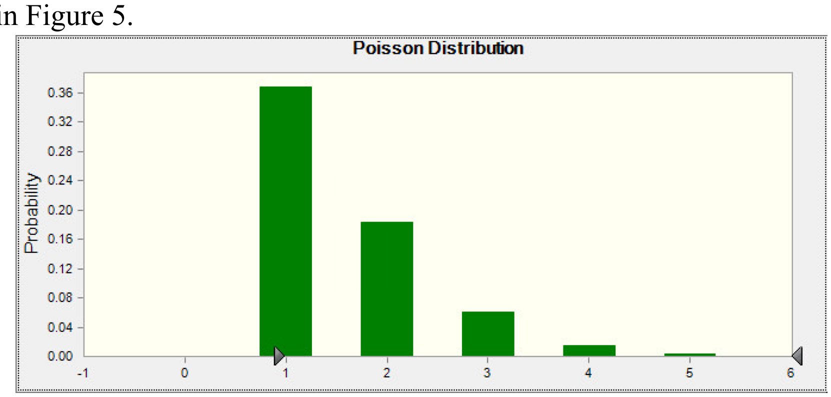


Figure 5

We see that it is most likely we will be able to film the following day, but there is a still a slight possibility that there are more lag days during which costs must be paid but no filming can be done.

To calculate the possible lag time (in days) if a problem were to occur for each day on the schedule, we use an Excel function to return 0 (no delay) if the corresponding entries in a certain row (certain day) of matrix **A** are greater than or equal to the corresponding entries in matrix **R** (because despite the problem, all required actors and resources are available, producing no delay) and a delay time chosen randomly based on the probability in the Poisson distribution (for example, there is a roughly 36% chance that the delay time chosen will be 1) if the greater than or equal to requirement is not fulfilled. (See figure 5)

In our 3-day, 2-actor, 1-resource, and 1-prerequisite example, the delay time distributions for day 1, day 2, day 3, and total delay with 100,000 runs can be seen in Figures 6, 7, 8, and 9.

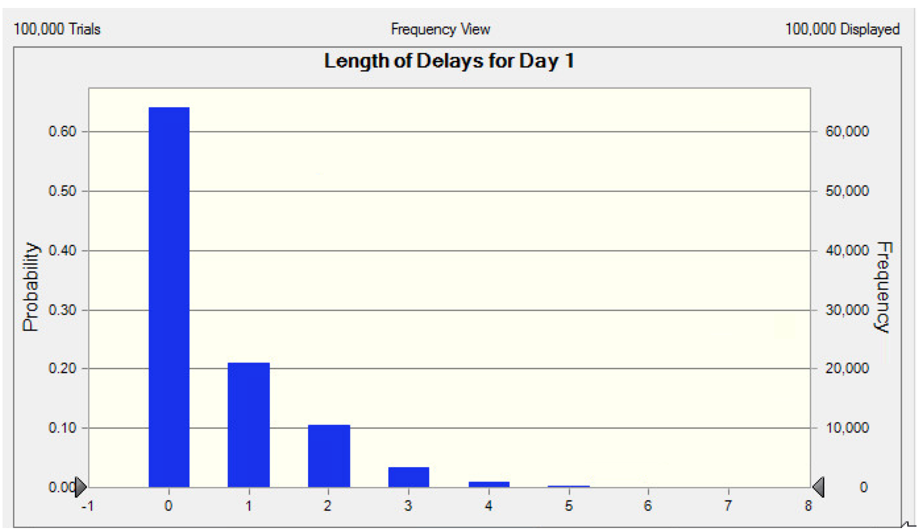


Figure 6

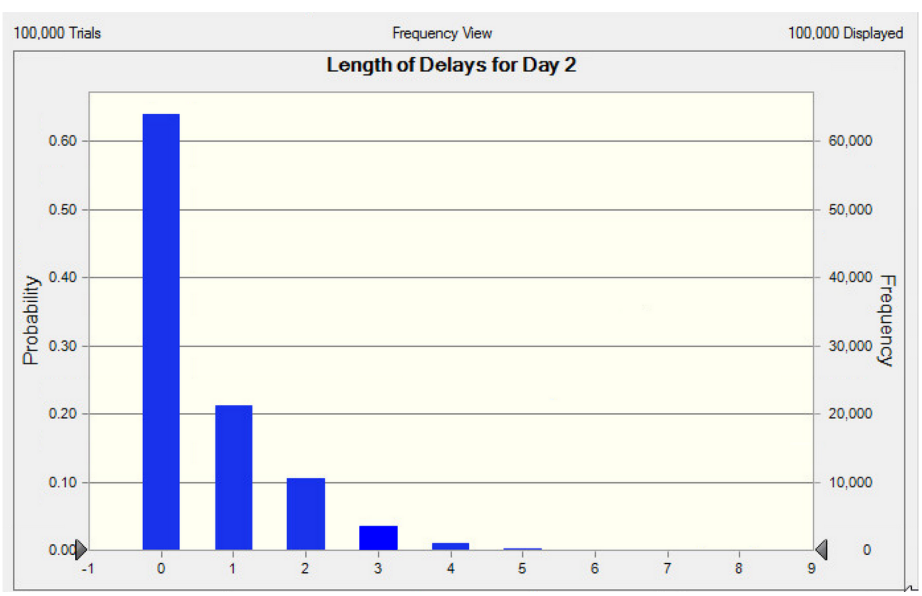


Figure 7

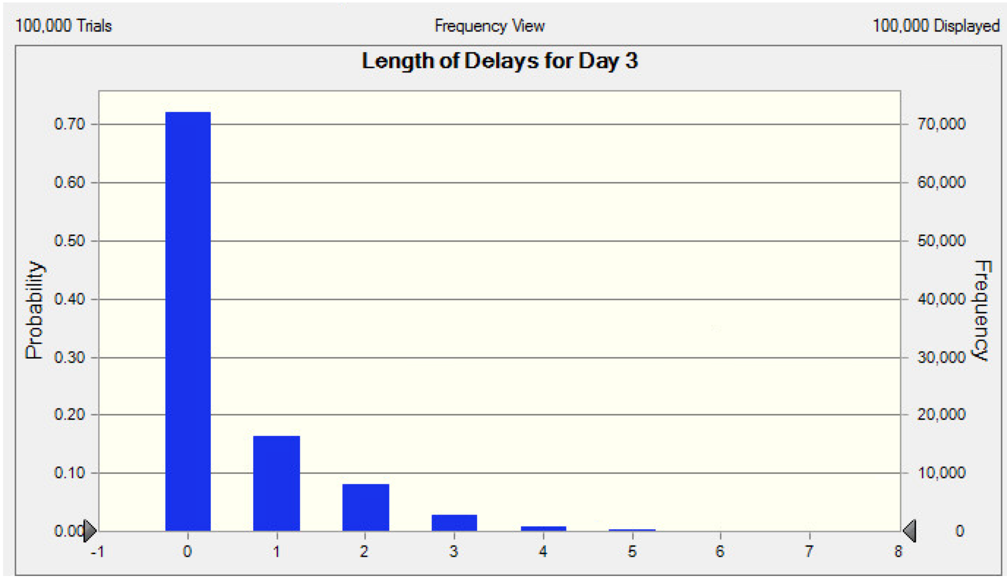


Figure 8

To generate the distribution of the total number of lag days from our example 3-day schedule, we sum the distributions of the number of lag days for each day in our schedule, producing the graph in Figure 9.

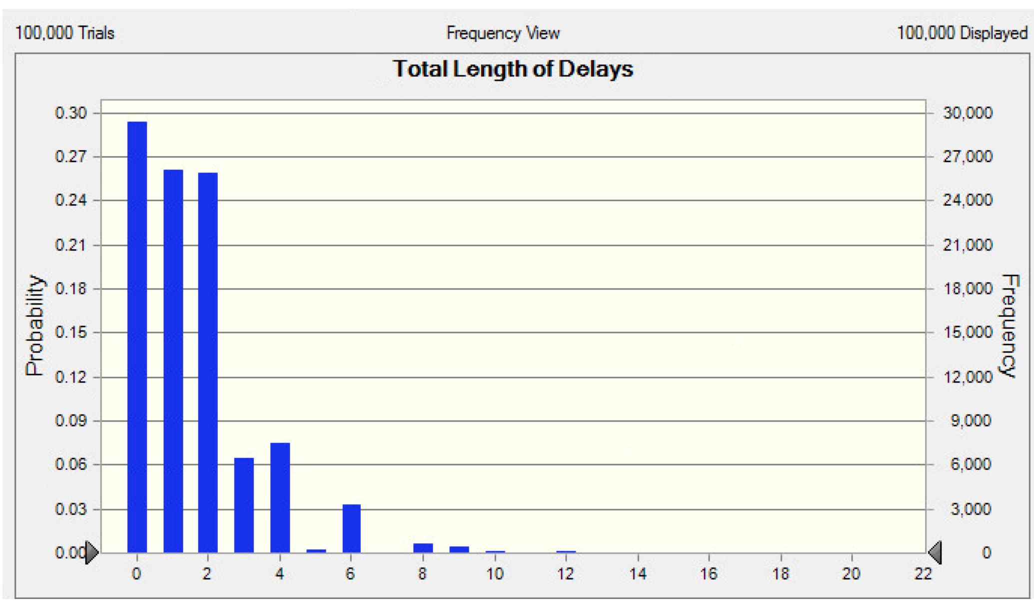


Figure 9

In order to analyze which constraints will cause the longest delays if a problem occurs, we generate the sensitivity analysis for the different variables that affect the length of the lag time:

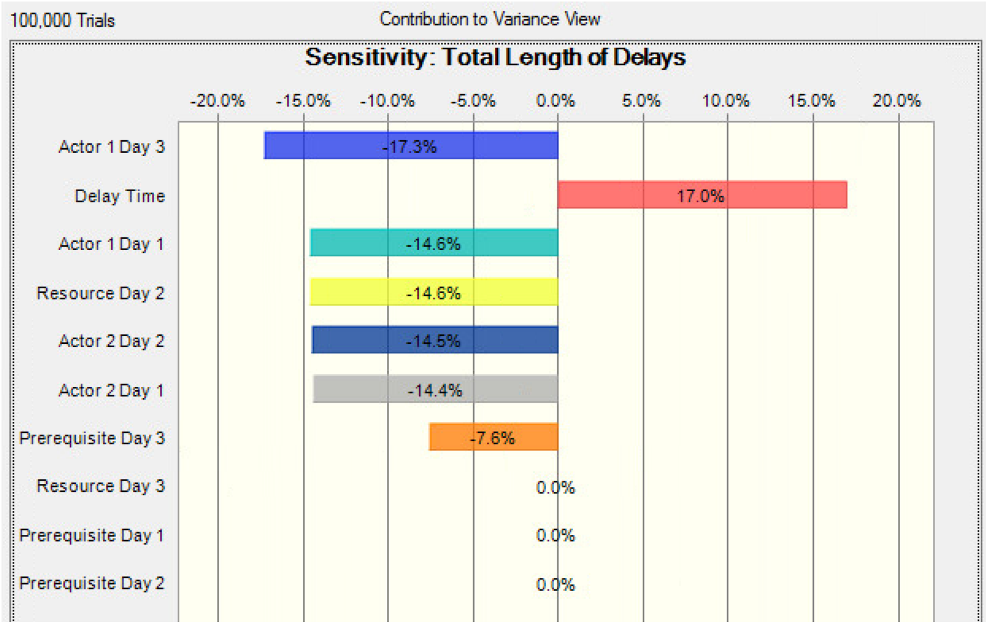


Figure 10

(*Note: a negative sensitivity indicates a negative correlation between that variable and the objective function and a positive sensitivity indicates a positive correlation between the variable and the objective function. Less availability means more delay time--a negative correlation--but more daily delay means more overall delay time--a positive correlation.*)

From the sensitivity analysis, we can see that changes in prerequisites account for a much smaller percentage change in the total length of delays than changes in actor and resource availability (which we group together as one type of cause for delay since both actor and resource availability are represented the same way in the availability and requirement matrices and are subject to the same constraints). The absolute sum of the percents responsible by prerequisites is only 7.6% while the absolute sum of the percents responsible by actor and resource availability is 92.4%. Because there is only 1 prerequisite and there are 3 actors/resources, we scale the percentages, dividing 92.4% by 3 to find to average percent responsible by each actor/resource - 30.8%, which is still much larger than the percentage of delay caused by a prerequisite (7.6%).

Therefore, from our simulation for the effects of possible delays on length of total delay time as well as the sensitivity analysis, we can conclude that availability constraints will cause the longest delay if a problem occurs and any time prerequisites will have a smaller effect on the length of delay. This sensitivity analysis has a variety of applications for the movie studio. If, for example, the studio wishes to insure itself against delay (which usually has substantial harmful financial consequences), the sensitivity analysis informs its decision as to how much to insure different types of accidents and delays. Clearly, it should spend more on its insurance policy for losses due to actor availability delays and less on insurance for preparation delays. If the studio is interested, our model quantifies exactly how likely each is projected to occur. Conversely, the model could be of use to insurance companies trying to determine how much they should charge for different movie insurance policies covering different types of damages.

Strengths and Weaknesses

We are confident in the ability of our model to produce a schedule for the studio that works well and will effectively account for real-world constraints and possible accidents. Our model's most salient strengths are its speed, accuracy, cognizance of real-world conditions, and reliability in finding a solution - when there is a solution to find. The model is quick because it employs heuristics to cut down on processing time, but it is still accurate, because the heuristics take into account every permutation reasonable and only save time by ignoring sub-optimal or entirely unreasonable permutations. Real-world conditions are also very important to our model, because we must never lose sight of the fact that this is something to be applied by a movie studio, not just examined in the interest of mathematics. Our research demonstrated that there are a number of different regulations movie studios are required to follow, such as the SAG requirement that actors must be paid during their "hold period." Models that do not consider these real-world regulations would be useless to a movie studio because they ignore conditions to which the studios are bound. Last, a model like that proposed by Cheng, while useful for the scope of the problem he considered, is less useful here. Our model accounts for the factors his model ignores while also reducing computing time and producing a solution more often. Most importantly, our maximizing matches heuristic makes it extremely unlikely that we will miss an optimal solution where a solution exists. Our test cases further demonstrate the effectiveness of our model. They show that our model can not only find a solution, but it can also identify the most optimal schedule. They stand up to scrutiny when subjected to manual inspection and produce reasonable schedules consistent with what would be seen in the real world. Importantly, the test cases run through the computer very quickly, ensuring that they are realistic for a movie studio to use.

While we are confident in the effectiveness of our model, we would be remiss not to address its imperfections. One of the most effective treatments of the scheduling problem harnesses the potential of parallel virtual machines (PVM) [Shyu, et al. 2000] to process far more permutations, possibly producing a more optimal answer. We chose not to pursue such a method because we do not think it would be at all reasonable to ask a movie studio to invest in expensive, high-powered computers just to create their own schedule, and we likewise lack access to such equipment. Our alternative thus produces a solution that is not always 100% optimal but is more relevant to real-world application in a movie studio. Furthermore, while our model is not accurate all of the time, it can produce the most optimal schedule most of the time. The model also can oversimplify in its assumption that the studio will never film more than one scene per day, which is not unrealistic.

Conclusion:

After thorough research of the topic literature and preliminary modeling, we adopted an ILP approach using local optimization with pairwise interchange and match maximization heuristics to generate a comprehensive film schedule that takes into consideration star/resource availability, location, the time to film various scenes, necessary preparation time, and extra time for reshoots. We also generated test cases for both our major and preliminary models that validate the applicability of our ILP, Excel, and time models to movie scheduling. In addition, our model accounts for delays caused by any of the constraints and sheds light on which variables in the model are the most sensitive; that is, which produce the largest effects if modified. This is of great use to the studio since, although nobody can control whether something might happen to an actor or resource, rendering it unavailable, we are able to quantify the magnitude of the projected disruption far in advance, allowing the movie studio to effectively plan ahead. Thus, we are confident that our model will be able to generate a filming schedule that can be of the most use to the movie studio in real-world conditions.

Appendix:

Preliminary Model #1 (Time) Test Case:

This is the pseudocode for testing scene 1 first in a queue of multiple scenes in preliminary model #1, explained by the flowchart in the main body of the report:

t_{sa} = time it takes to film scene a (s_a)
f_a = prerequisite a
T_{fa} = time it takes for f_a to be ready
T_{sa} = time it takes to film s_a
$A_a(t)$ = piecewise function for the availability of actor a (binary: 1 means available)
$R_{sa}(t)$ = piecewise function for the availability of resource a (binary)
P_{sa} = required actors/resources/sites of scene a (e.g., $\{A_1, A_2, R_1\}$)
N_{sa} = required prerequisites of scene a (e.g., $\{f_1, f_2\}$)
t = time elapsed
q = queue of all the scenes (e.g., $\{S_1,....S_n\}$)
n = total number of scenes in the movie

for all permutations of the scene order (n! times){
while (q has >0 scenes) {
t=0
##Comment: Sets the elapsed time at zero.
$P_{s1}=\{A_1, R_1, R_2\}$
##Comment: Describes which actors/resources/sites are needed.
$N_{s1}=\{f_1,f_3\}$
##Comment: Describes which prerequisites are needed.
if ($A_1(t)=R_2(t)=R_2(t)=1$ for $t \in (t, t+T_{s1})$ and $t>T_{f1}$ and $t>T_{f3}$){
##Comment: The equality checks that all the requisite actors and resources are available for the entire duration of the filming.
##Comment: The pseudocode then checks if enough time has elapsed for the prerequisite to be ready.
##Comment: The two inequalities ensure that all prerequisites are ready.
t+=T_{s1}
##Comment: Adds the time taken to film scene 1 to the total time elapsed
remove s_1 from the queue.
else{
$t = t + 1$
##Comment: A day has gone by, but nothing has happened, so add 1 day to the total elapsed time.
}
if (q has 0 scenes){
print (t)
##Comment: After all the scenes have been filmed, print the total amount of time that has elapsed. The objective function for this model is time, so this checks how well the objective function has been optimized.

```
            }
        }
}
```

Preliminary Model #2 (Cost with Microsoft Excel) Sample and Test Case:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Availability						Needed														
2	Actor 1	Actor 2	Actor 3	Resource 1	Day	Scene	Actor 1	Actor 2	Actor 3	Site	Resource	Preparation Time	Actor 1 paid days	Actor 2 paid days	Actor 3 paid days	Site/set movement cost	Actor pay per day	Can the scene be shot on this day	Actor cost per day	Total cost	Is this schedule do-able?
3	1	1	1	1	1	1	0	1	0	1	0	0	0	1	0	100	571.43	1	428.5714	63357.14	0
4	1	1	1	0	2	2	0	1	1	1	0	0	0	1	1	0	428.57	1	714.2857		
5	0	1	1	1	3	3	0	1	0	1	0	0	0	1	1	0	285.71	1	714.2857		
6	1	0	1	0	4	4	1	0	0	1	0	3	1	1	1	0		1	1285.714		
7	1	0	1	0	5	5	1	1	1	1	0	0	1	1	1	0		0	1285.714		
8	1	1	0	0	6	6	0	1	1	1	0	0	1	1	1	0		0	1285.714		
9	1	1	0	0	7	7	1	1	1	1	1	0	1	1	1	0		0	1285.714		
10	1	1	1	0	8	8	0	1	0	2	0	0	1	1	1	100		1	1285.714		
11	0	1	1	0	9	9	0	0	0	3	0	3	1	1	1	100		1	1285.714		
12	0	1	0	0	10	10	1	1	1	4	0	0	1	1	1	100		0	1285.714		
13	1	0	1	0	11	11	0	1	1	4	0	0	1	1	1	0		0	1285.714		
14	1	0	0	0	12	12	1	0	1	4	0	0	1	1	1	0		0	1285.714		
15	1	1	1	0	13	13	0	1	1	4	0	0	1	1	1	0		1	1285.714		
16	1	1	1	0	14	14	0	1	0	5	0	0	1	1	1	100		1	1285.714		
17	1	1	1	0	15	15	1	0	1	6	1	0	1	1	1	100		0	1285.714		
18	0	1	1	0	16	16	1	1	0	7	0	0	1	1	1	100		0	1285.714		
19	0	1	0	1	17	17	0	1	0	7	0	0	1	1	1	0		1	1285.714		
20	1	0	1	1	18	18	1	0	1	7	0	5	1	1	1	0		1	1285.714		
21	1	0	0	0	19	19	1	0	1	7	0	0	1	1	1	0		0	1285.714		
22	1	1	1	0	20	20	0	0	0	7	0	0	1	1	1	0		1	1285.714		
23	1	1	1	0	21	21	0	0	1	8	0	0	1	1	1	100		1	1285.714		
24	1	1	0	0	22	22	1	1	0	8	0	0	1	1	1	0		1	1285.714		
25	0	1	0	0	23	23	0	0	1	8	0	0	1	1	1	0		0	1285.714		
26	0	1	0	0	24	24	1	1	0	9	0	0	1	1	1	100		0	1285.714		
27	1	0	1	0	25	25	0	1	0	10	0	0	1	1	1	100		0	1285.714		
28	1	0	1	0	26	26	0	1	0	10	1	0	1	1	1	0		0	1285.714		
29	1	1	1	0	27	27	1	0	0	10	0	0	1	1	1	0		1	1285.714		
30	1	1	1	0	28	28	1	0	0	10	0	0	1	1	1	0		1	1285.714		
31	1	1	1	0	29	29	1	0	0	11	0	0	1	1	1	100		1	1285.714		
32	0	1	1	0	30	30	1	0	1	11	0	0	1	1	1	0		0	1285.714		
33	0	1	1	0	31	31	1	0	1	12	0	1	1	1	1	100		0	1285.714		
34	1	0	1	0	32	32	1	1	0	12	0	0	1	1	1	0		0	1285.714		
35	1	0	0	0	33	33	1	1	1	12	0	0	1	1	1	0		0	1285.714		
36	1	1	1	0	34	34	0	0	1	12	0	0	1	1	1	0		1	1285.714		
37	1	1	1	0	35	35	0	1	1	12	0	0	1	1	1	0		1	1285.714		
38	1	1	0	0	36	36	0	1	1	12	1	0	1	1	1	0		0	1285.714		
39	0	1	0	1	37	37	0	1	0	12	0	0	1	1	1	0		1	1285.714		
40	0	1	1	0	38	38	1	1	1	12	1	0	1	1	1	0		0	1285.714		
41	1	0	0	1	39	39	0	1	0	13	0	0	1	1	1	100		0	1285.714		
42	1	0	0	1	40	40	1	1	1	14	0	0	1	1	1	100		0	1285.714		
43	1	1	0	1	41	41	1	1	1	14	0	0	1	1	1	0		0	1285.714		
44	1	1	1	1	42	42	1	1	0	14	0	0	1	1	1	0		1	1285.714		
45	1	1	0	1	43	43	1	0	0	14	0	0	1	1	1	0		1	1285.714		
46	0	1	1	1	44	44	1	1	1	14	0	0	1	1	1	0		0	1285.714		
47	0	1	1	1	45	45	1	1	0	14	0	0	1	1	1	0		0	1285.714		
48	1	0	0	1	46	46	1	0	1	14	1	0	1	1	1	0		0	1285.714		
49	1	0	1	1	47	47	0	0	1	15	0	2	1	1	1	100		1	1285.714		
50	1	1	0	0	48	48	0	0	1	15	0	0	1	1	1	0		0	1285.714		
51	1	1	1	0	49	49	1	1	1	15	0	0	1	1	1	0		1	1285.714		
52	1	1	1	0	50	50	1	0	1	15	0	0	1	0	1	0		1	857.1429		

Table 1: Sample table for cost model

Example:

As an example for this model, we scheduled 3 days worth of shooting with 3 actors, 1 resource requirement, and 1 preparation time requirement. The requirements are shown in Table X:

Needed					
Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time
1	1	0	1	0	0
0	1	1	1	0	2
1	1	0	2	1	0

Table 2

We calculated total cost and filming possibility for all 6 permutations of the 3-day schedule with the following results shown in Tables 3-8.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Availability						Needed														
2	Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
3	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0	100	571.43	1	1000	3485.71	0
4	1	1	0	0	2	2	0	1	1	1	0	2	1	1	1	0	428.57	0	1285.714286		
5	0	1	1	1	3	3	1	1	0	2	1	0	1	1	0	100	285.71	0	1000		

Table 3

	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	1	1	1	0	1	0	0	1	1	0	100	571.43	1	1000	3200	0
1	1	0	0	2	3	1	1	0	2	1	0	1	1	1	0	428.57	0	1285.714286		
0	1	1	1	3	2	0	1	1	1	0	2	0	1	1	100	285.71	1	714.2857143		

Table 4

	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	2	0	1	1	1	0	0	0	1	1	100	571.43	1	714.2857143	2914.285714	0
1	1	0	0	2	1	1	1	0	1	0	2	1	1	0	0	428.57	0	1000		
0	1	1	1	3	3	1	1	0	2	1	0	1	1	0	100	285.71	0	1000		

Table 5

	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	2	0	1	1	1	0	2	0	1	1	100	571.43	1	714.2857143	3200	0
1	1	0	0	2	3	1	1	0	2	1	0	1	1	1	0	428.57	0	1285.714286		
0	1	1	1	3	1	1	1	0	1	0	0	1	1	0	100	285.71	0	1000		

Table 6

	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	3	1	1	0	2	1	0	1	1	0		571.43	1	1000	2814.285714	1
1	1	0	0	2	1	1	1	0	1	0	0	1	1	0	0	428.57	1	1000		
0	1	1	1	3	2	0	1	1	1	0	2	0	1	1	100	285.71	1	714.2857143		

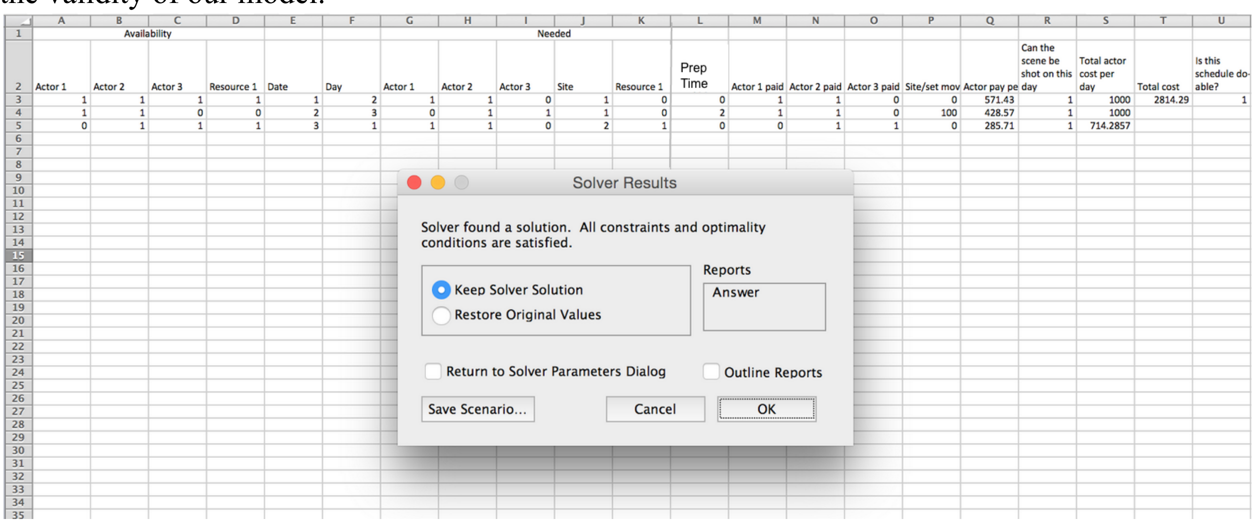
Table 7

	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Preparation Time	Actor 1 paid	Actor 2 paid	Actor 3 paid days	Site/set movemer	Actor pay per day	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	3	1	1	0	2	1	0	1	1	0	100	571.43	1	1000	3385.714286	0
1	1	0	0	2	2	0	1	1	1	0	2	1	1	1	0	428.57	0	1285.714286		
0	1	1	1	3	1	1	1	0	1	0	0	1	1	0	0	285.71	0	1000		

Table 8

As we can see, only 1 out of the 6 permutations (Table 7) is feasible (all actors and resources available and required preparation times met) with a total cost of $2814.29. Our model is scalable (able to calculate permutations for a larger number of scenes) by using a computer program calculate the total cost for all permutations and finding the minimum cost.

To check the validity of our model, we used Excel Solver to have the computer generate a solution to our problem. Using the VLOOKUP function to make re-ordering the filming days possible, we found the <u>exact same solution</u> as we did with the manual calculations, confirming the validity of our model.



	Availability							Needed												
Actor 1	Actor 2	Actor 3	Resource 1	Date	Day	Actor 1	Actor 2	Actor 3	Site	Resource 1	Prep Time	Actor 1 paid	Actor 2 paid	Actor 3 paid	Site/set mov	Actor pay pe	Can the scene be shot on this day	Total actor cost per day	Total cost	Is this schedule do-able?
1	1	1	1	1	2	1	1	0	1	0	0	1	1	0	0	571.43	1	1000	2814.29	1
1	1	0	0	2	3	0	1	1	1	0	2	1	1	0	100	428.57	1	1000		
0	1	1	1	3	1	1	1	0	2	1	0	0	1	1	0	285.71	1	714.2857		

Fig. 1: Solver solution page - same as solution from manual permutations

Primary Model Test Case:

For the test case, we begin with two matrices that the filmmakers must provide but that we generate randomly for the purposes of this example: the availability matrix for the actors and prerequisites, and the requirement matrix for each of the scenes.

A:

$$\begin{bmatrix}0&1&1&0&1&1\\1&0&0&1&0&1\\1&1&0&0&1&1\\0&0&0&1&1&1\end{bmatrix}$$

R:

$$\begin{array}{r}\\ \\ \\ \\ \text{Site:}\end{array}\begin{bmatrix}1&0&1\\1&1&0\\1&0&0\\1&1&0\\1&2&2\end{bmatrix}$$

After inputting these two matrices into the computer program, the following will be the output by utilizing the match maximization heuristic:

[5,3,2]

(because computer programming languages start from zero when counting rows and columns, add one to all the values in this array)

This means that the first scene will be filmed on the (5+1) = 6th day, the second scene will be filmed on the (3+1) = 4th day, and the third scene will be filmed on the (2+1) = 3rd day. From this information, we generate the preliminary matrix $\mathbf{N}_{\text{first}}$ below, placing the first column of **R** in the 6th column of $\mathbf{N}_{\text{first}}$, placing the second column of **R** in the 4th column of $\mathbf{N}_{\text{first}}$, and the third column of **R** in the 3rd column of $\mathbf{N}_{\text{first}}$, and the rest are lag days where the entire column is 0's.

$$\begin{bmatrix}0&0&1&0&0&1\\0&0&0&1&0&1\\0&0&0&0&0&1\\0&0&0&1&0&1\\0&0&2&2&0&1\end{bmatrix}$$

We will first transform **N** to **Y** by trimming the two beginning lag days:

$$\begin{bmatrix}1&0&0&1\\0&1&0&1\\0&0&0&1\\0&1&0&1\\2&2&0&1\end{bmatrix}$$

We then generate **D** from **Y** according to Formula A:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

We multiply **D** by **C**, generating matrix **Q**:

$$\begin{bmatrix} 100 & 300 & 300 & 300 \end{bmatrix}$$

The total actor cost is the sum of all entries in matrix **Q**, which equals

$$100 + 300 + 300 + 300 = 1000.$$

We now generate matrix **Z** (which shows the different site codes where filming occurs) from matrix **Y** to calculate the number of location changes according to formula B:

$$\begin{bmatrix} 2 & 2 & 1 \end{bmatrix}$$

Here, we can see that the first two days are filmed at location 2, while the last day is filmed at location 1. Thus, there is one location change.
We then generate matrix **S** from matrix **Z** according to Formula C. With a randomly selected *p* of 1000, **S** is:

$$\begin{bmatrix} 0 & 1000 \end{bmatrix}$$

The total cost of changing location is the sum of all entries in **S**, or 0 + 1000 = $1000. Now we can calculate the total cost of production by summing the location change cost, the total constant daily cost ($a*q$, where a is randomly assigned as 2000 and q, the number of columns in **Y**, = 4), and the total cost of hiring actors.

$$k = 1000+2000*4+1000 = \$10000$$

This is the matrix that incorporates the largest number of matches, meaning it is the matrix most likely to produce a solution. From here, we apply the pairwise interchange heuristic to improve the cost further. Eventually, the program shows that

[5,3,4]

is the most optimal for minimizing cost.

This creates an $\mathbf{N}_{best}$ matrix in which the first column of **R** is placed in the 6th column of $\mathbf{N}_{best}$, the second column of **R** is placed in the 4th column of $\mathbf{N}_{best}$, the third column of **R** is placed in the 5th column of $\mathbf{N}_{best}$, and the rest of the columns in $\mathbf{N}_{best}$ are lag days where the entire column is 0's.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 & 2 & 1 \end{bmatrix}$$

This matrix yields a total cost of $7800, according to our cost calculation Formula D.

Computer Code:

Program part 1 (Python):

```
from numpy import matrix
from numpy import linalg

A=matrix([[0,1,1,0,1,1],[1,0,0,1,0,1],[1,1,0,0,1,1],[0,0,0,1,1,1]])

print ('your availability matrix is')
print A

R=matrix([[1,0,1],[1,1,0],[1,0,0],[1,1,0],[1,2,2]])

print ('your requirement matrix is ')
print R

rj=0 #column number of the requirement matrix
i=0 # row#
aj=0 #column number of the availability matrix
m=4 #number of actors, resources, and prerequisites
an=6 #number of days given in the availability matrix
rn=3 #number of scenes needed to be shot

result =[]

for rj in range(0,rn):
    rj_score=0
    best_j=-1
    for aj in range (0,an):
        if aj in result:
            continue
        score=0
        for i in range (0,m):
            print i
            print rj,aj
            if A[i,aj]>R[i,rj]:
```

```
                score=-1
                break
            elif A[i,aj]==R[i,rj]:
                score+=1
        if rj_score<score:
            best_j=aj
    result.append(best_j)
    print 'this is result',rj, best_j
print result
```

Works Cited

Artigues, Christian, Sophie Demassey, and Emmanuel Néron. *Resource-constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. London: ISTE, 2008. Print.

Cheng, T. C. E., J. E. Diamond, and B. M. T. Lin. "Optimal Scheduling in Film Production to Minimize Talent Hold Cost." *Journal of Optimization Theory and Applications* 79.3 (1993): 479-92. Web. 20 Apr. 2015.

Chinneck, John W. "Integer/Discrete Programming via Branch and Bound." Carleton University Systems and Computer Engineering, 2010. Web. 20 Apr. 2015.

Directors Guild of America. "Directors' Minimum Conditions - Preparation, Production and Post-Production." Directors Guild of America, 2011. Web. 20 Apr. 2015.

Honthaner, Eve Light. *The Complete Film Production Handbook*. 4th ed. Burlington, MA: Focal, 2010. Print.

Media College. "The Film Director." Wavelength Media, n.d. Web. 20 Apr. 2015.

Media College. "Pickup Shots." Wavelength Media, n.d. Web. 20 Apr. 2015.

Oracle Crystal Ball. Vers. 11.1.2.1. Redwood City, CA: Oracle Corporation, 2011. Computer software.

Shyu, Shyong-Jian, and B. M. T. Lin. "An Application of Parallel Virtual Machine Framework to Film Production Problem." *Computers & Mathematics with Applications* 39.12 (2000): 53-62. Web. 20 Apr. 2015.

Tellychakar. "Producers and Artistes Discuss the FWICE' '12-hour for Workers' Rule." Ed. Anil Wanvari. Tellychakkar, 4 Sept. 2010. Web. 20 Apr. 2015.

Wild, Larry. "Film Production." Northern State University, 31 Dec. 2014. Web. 20 Apr. 2015.