Job Scheduling and Allocation on Heterogeneous Servers with Multiple Surgery Rooms

Yuheng Cai Shuzhe Xiao Yuguo Zhong

February 6, 2018

1 Introduction

1.1 Problem Description

Surgery planning has been a widely studied problem in areas of job scheduling and allocations[1]. The problem can be described as patients entering a hospital and needing to go through several processes to complete their treatment. Here, we extend this problem to model surgery planning under the setting of heterogeneous servers with multiple surgery rooms. For our problem, we pre-assign each patient a sequence of servers to go through, and then sequentially allocate patients to multiple servers efficiently when the service duration is highly uncertain.

1.2 Assumptions

To make the model more concise and efficient, some assumptions were made before modeling:

- Each server has at least one OR.
- The sequence of servers patients need to go through is pre-given as a set
- Patients' no-show, delay and cancellation are not considered.
- Operating rooms under the same server are identical and interchangeable
- Job duration is independent of ORs and job position
- The unit cost of waiting time, idling time and overtime only depends on server types (i.e. the unit cost of different rooms of the same server should be the same).
- Each room is available for only one surgery at a time and each surgery is conducted in exactly one room.

1.3 Objective & Method

There are two main objectives for this problem, each of which will be determined stage-wise. Our first objective is to determine optimal opening decision of operating rooms and consequently allocation of patients to these ORs. After achieving this, we then try to find out optimal start time of surgery and intervals between each surgery, while balancing conflicting objectives including waiting, idling, and overtime cost.

To optimally solve this problem, we integrate allocation and scheduling into one problem, and model this problem in two stages. Some decision variables such as idling time between surgeries depend on the realization of surgery duration that follows from certain probability distribution; however, we will initially solve the first-stage problem that includes only allocation decisions that are independent of realization of such scenario. And all other decision problems that involve realization of different distribution scenarios will be solved in the stage stage. For the purpose of best simulating reality, we will randomly generate data from Log-Normal distribution and compare its results with data from Normal distributions.

2 Literature Review

We have conducted literature reviews on some related papers, from which we have also borrowed some ideas for implementation of our model. Here

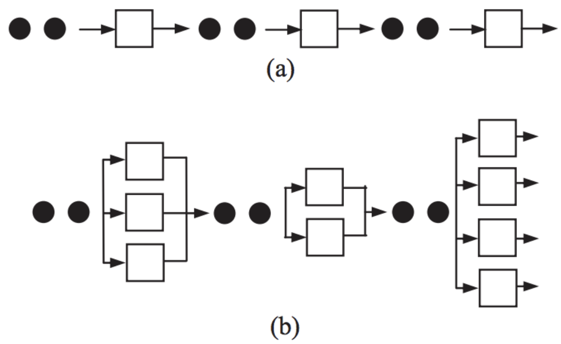

Figure 1: example of queuing system from [2]: (a) multi-stage/single-server system; (b) multi-stage/ multi-server system.

we provide a brief summary of work and paper that helped to the modeling and other parts of our work.

2.1 Surgery planning under uncertainty

The primary literature focus on this area is to model single operating room under very similar constraints as those in our problem. P.P.Wang [3] assumed that job duration are iid random variables following Coxian distribution. G. C. Kaandorp and G. Koole [4] modeled job duration as from Exponential distribution with different means and patient can only arrive at an interval of ten minutes. In P. M. Vanden Bosch and D. C. Dietz papers, [5] [6] the authors assumed discrete scheduled starting times and included penalties for waiting time and overtime. They used Phase Type and Lognormal distributions to model the three duration distributions. In Deng's paper [7][8] three binary decision variables were used to decide opening of operating room, job assignment and job sequence. We have also incorporated such objective and several constraints in our model, but we extend the complexity of those decisions by including multiple ORs and sequential server process for patient. In Denton's paper[9], they showed a sequencing rule to generate substantial reduction in time cost based on the surgery duration time. One of our objectives is also to make sequential decisions to minimize various time cost.

2.2 Multi-server/multi-room system

Another essential part of our model is to focus on the multi-server with multi-OR setting (Figure 4). It is very common to have multiple surgeries in different ORs for the same type of server at the same time. E. Weiss [2] get the optimal sequence when there are only two surgeries and concluded that this may not apply to surgeries over two. In Zhang's work[10], they developed a dynamic surgery assignment model in multi-OR setting. Their model defined starting time of surgery according to surgeon's arrival time, but we primarily focus on the patients' side and do it according to the arrival time of patients. In Granja's work[2], they came up with a concept of a static multi-stage/multi-server system with precedence relations that are helpful to the formulation of several of our model's constraints. However, paper related to multi-server scheduling problem is not very common.

3 Model Formulation

In this section, we describe our model as a two-stage linear programming model aiming to solve the scheduling and allocation problem of a medical institution with multiple stages and multiple rooms.

The quality of a schedule and allocation plan can be judged by a weighted sum of expected total cost of waiting time, idling time and overtime. Overtime with respect to a planned opening session is important because late completion of a job can have adverse effect on the institution's overall performance. We will give overtime a heavier weight to our model.

3.1 Notation

We denote I the set of patients to be scheduled and J be all servers(stages). J_i', a subset of J, is set of server sequences that patient i needs to complete. L_j is set of ORs for server j, and P_{ij} denotes the set of tasks that precedes server j of patient i.

Uncertainty is denoted by a scenario ω that defines the vector of random job duration by $d_{ij}(\omega)$. The opening session(total length of opening) of server j room l is t_{jl}. Unit cost of overtime, waiting time, and idling time of server j are α_j, β_j, γ_j. OR opening cost c_j is also introduced.

For scheduling part, we have 6 decision variables. The waiting and idling time of patient i directly following patient m in server j room l are represented by $w_{imjl}(\omega)$ and $u_{imjl}(\omega)$, respectively. Both of them depend on random duration through scenario, ω. $o_{jl}(\omega)$ indicates the period that exceeds planned closure time of server j room l. x_{ijl} defines the scheduled interval for surgery i in server j room l. We also introduce s_{ijl}, meaning start time of job i in server j of l^{th} room and δ_{imjl} denotes the planned start time of surgery i.

For allocation part, we have 3 decision variables that are all binary. v_{jl} equals 1 if server i room l is open, 0 otherwise. y_{ijl} equals to 1 if assigning job i to server j of l^{th} room, 0 otherwise. z_{imjl} equals to 1 if job i is arranged directly following job m in server j room l, otherwise 0;

v_{jl}, y_{ijl} and z_{imjl} are all binary variables denoting sequencing and allocating decisions. The formulation includes first stage binary decision variables representing sequencing decisions which must be defined in advance of knowing the outcomes of the random surgery duration.

3.2 Problem Formulation

Given above definitions, the first stage stochastic optimization problem can be formed as:

$$\min \sum_{j\in J}\sum_{l\in L_j} c_j v_{jl}$$

$$\sum_{l\in L_j} y_{ijl} = 1 \qquad \forall i\in I, j\in J_i' \qquad (1)$$

$$y_{ijl} \le v_{jl} \qquad \forall i\in I, j\in J_i' \qquad (2)$$

$$\sum_{i\in I} y_{ijl} \ge v_{jl} \qquad \forall j\in J_i', l\in L \qquad (3)$$

$$v_{jl}, y_{ijl} \in \{0,1\} \qquad \forall i,m\in I, j\in J, l\in L_j$$

Constraint (1) guarantees that each stage j of job i is assigned to one and only one certain room. Constraint (2) limits that job should be allocated to open rooms. Constraint (3) means that if a room is open, there has to be at least one job allocated to that room (otherwise this room should not be opened for economical concern).

After obtaining opening and allocation decisions, we incorporate scheduling part. The stochastic optimization problem can be written as minimization of the weighted sum of the expectation of waiting, idling, and overtime plus the opening cost of rooms as follows:

$$\min \sum_{\omega\in\Omega} p^{\omega}\Big(\sum_{i\in I}\sum_{m\in I}\sum_{j\in J}\sum_{l\in L_j}(\beta_j w_{imjl}(\omega) + \gamma_j u_{imjl}(\omega)) + \sum_{j\in J}\sum_{l\in L_j}\alpha_j o_{jl}(\omega)\Big)$$

$$\sum_{i\in I}\sum_{m\in I} z_{imjl} \ge \sum_{i\in I} y_{ijl} - 1, \forall j\in J, l\in L \qquad (4)$$

$$y_{ijl} \ge z_{imjl} \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (5)$$

$$\text{ijl-}y_{mjl} + z_{imjl} \le 1 \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (6)$$

$$y_{mjl} - y_{ijl} + z_{imjl} \le 1 \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (7)$$

$$z_{imjl} + z_{mijl} \le 1 \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (8)$$

$$\sum_{i\in I} z_{imjl} \le 1 \quad \forall m\in I, j\in J, l\in L_j \qquad (9)$$

$$\sum_{m\in I} z_{imjl} \le 1 \quad \forall i\in I, j\in J, l\in L_j \qquad (10)$$

$$w_{imjl}(\omega) \le M y_{ijl} \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (11)$$

$$u_{imjl}(\omega) \le M y_{ijl} \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (12)$$

$$\delta_{imjl} \le M y_{ijl} \quad \forall i,m\in I, i\neq m, j\in J, l\in L_j \qquad (13)$$

$$\sum_{m\in I}(w_{mijl}(\omega) - w_{imjl}(\omega) - u_{imjl}(\omega) + \delta_{mijl} - \delta_{imjl}) = d_{ij}(\omega) * y_{ijl} \quad \forall i\in I, j\in J, l\in L_j \qquad (14)$$

$$o_{jl}(\omega) - \sum_{i\in I}\sum_{m\in I} u_{imjl}(\omega) = \sum_{i\in I} d_{ij}(\omega) * y_{ijl} - t_{jl} \quad \forall j\in J, l\in L_j \qquad (15)$$

$$\sum_{m\in I} w_{imjl} = s_{ijl} - \sum_{m\in I}\delta_{imjl} \quad \forall i\in I, j\in J, l\in L_j \qquad (16)$$

$$s_{ijl} \ge s_{ikl} + d_{ik}(\omega), \quad \forall i\in I, j\in J_i', k\in P_{ij}, l\in L_j \qquad (17)$$

$$s_{ijl} \ge s_{mjl} - M(1 - z_{imjl}) + d_{mj}(\omega), \quad \forall i,m\in I, i\neq m, j\in J_i', l\in L_j \qquad (18)$$

$$\sum_{m\in I}\delta_{imjl} \ge \sum_{m\in I}\delta_{imkl} + d_{ik}(\omega) \quad \forall i\in I, j\in J_i', k\in P_{ij}, l\in L_j \qquad (19)$$

$$\sum_{m\in I}(\delta_{mijl} - \delta_{imjl}) \ge -M(1 - \sum_{m\in I} z_{mijl}) + d_{ij}(\omega) \quad \forall i\in I, j\in J_i', l\in L_j \qquad (20)$$

$v_{jl}, y_{ijl}, z_{imjl} \in \{0,1\} \quad \forall i, m \in I, i \neq m, j \in J, l \in L_j$ (21)

$w_{imjl}, u_{imjl}, o_{jl}, x_{ijl}, s_{ijl}, \delta_{imjl} \geq 0 \quad \forall i, m \in I, i \neq m, j \in J, l \in L_j$ (22)

For sequencing part, we have constraint (4)-(10). Constraint (4) defines the relationship between number of sequencing and number of allocation. Constraint (5) means that if job m is not assigned to server j room l the corresponding z should be 0. Constraint (6) - (8) decides the sequence in accordance with allocation. Constraint (9) and (10) means that only one or no job can be assigned to immediately following or preceding job i.

For scheduling part, we have constraints (11)-(20). Constraints (11) - (13) require that waiting ,idling and planned start time between surgeries be zero unless job has been assigned to this job. Constraint (14) - (16) balances between waiting time, idling time and overtime. Constraint (17) and (19) means the planned start time of current stage of a particular patient has to be larger than any stage preceding this stage plus their respective duration. Constraint (18) and (20) restrict that if patient i and m are at the same room and m is just ahead of i, the start time of job i has to be larger than the start time of job m plus its duration.

4 Model Advances and Solution Algorithm

For time and technical issues, the model is directly solved using random data in ampl with solver cplex. Therefore, the solution algorithm is provided by ampl, where the first-stage is solved using integer linear-programming methods and stage-stage solved with stochasticity.

4.1 Two-stage problem

For our model, the most difficult part lies in that we consider multiple servers with multiple rooms. At the same time, patients can have different servers (combinations) to go through. To optimally solve this problem, we divide the problem into two stages, where the allocation and OR opening decision is solved in stage one, and the rest scheduling part is solved in stage two. Similar to Deng's approach[7], we only included allocation variables and corresponding constraints in the first stage. Then, at second stage, solutions obtained from the first stage are plugged in as pre-defined parameters and the sequencing and scheduling problem among all patients is solved. For second stage problem, Deng[7][8] efficiently solved chance-constrained models using advanced algorithms.

4.2 Patient scheduling conflict

There arises another big challenge of avoiding schedule conflict of a patient. That means patient cannot start his next job until he finishes his current job. To solve this problem, we introduce the planned start time and actual start time for a single patient. Instead of using time interval, we chose planned start time difference for scheduling purpose.

5 Experimental Design and Computational Result

5.1 Parameter Choice

This model contains various parameters that need to be chosen carefully. The first set of parameters that need to be considered are waiting, idling, and overtime cost, and we set their ratio to be 1:1:5, a choice that is widely used in real setting. Then we set total daily OR opening length to be eight hours (480 minutes) based upon our understanding about general working hours. In the end, the operating cost of surgery room is set differently for each type, which should be reasonable because complexity and difficulty of surgery differs for different types. Sensitivity analysis will also be performed later to test the influence of parameters on the final outcome.

5.2 Testing Data

To test validity and correctness of our model, we first used a small designed instance with five patients and four servers, and each patient needs to go through at least three servers. Using this instance, we obtained an sequencing allocation decision as in Table 1 and

Server 1	Server 2	Server 3	Server 4
Job 4	Job 2	Job 3	Job 5
Job 1	Job 3	Job 2	Job 1
Job 5	Job 4	Job 1	Job 2
Job 2	Job 5	Job 4	

Table 1: Sequencing allocation decision

a minimum objective cost of 2422, where the overtime cost, which costs mostly, is minimized to zero for server 1 and 2. This example result shows that the model is successful in planning server scheduling by minimizing amount of overtime, while making optimal sequential decisions for each patient.

5.3 Result Comparison of Different Distributions

After verifying that the model works and produces reasonable result, we decided to generate random instances that follow empirical probability distributions. We referenced the surgery time table from [7] and [11], and generated random surgery duration for four different types of servers following mean and standard deviation of Normal and Log-Normal distribution.

Before experimenting, we first fixed the number of servers and number of patients, but the set of servers that patients need to go through is randomly drawn. Then using same set of parameters as before, and generating random numbers for testing for three times, we took average of three outcomes and obtained results in Table2 for four types of servers and Table3 for each test case and average. We see that data generated from Log-Normal distribution results in much higher total daily cost, and the amount of overtime is also much higher than that from Normal distribution.

For random data generated from both distributions, ampl provides a very efficient solution algorithm that costs no more than few seconds to solve, and therefore the CPU time is negligible and almost same for both data.

5.4 Sensitivity Analysis

We are also interested in testing how choice of parameters affects the result and which one affects the result mostly. Therefore, we conducted sensitivity analysis on following parameters using the initially created small instance and obtained result in Table4.

	J1	J2	J3	J4
Normal	14.66	52.84	11.23	4.44
Log-Normal	324.35	192.64	74.48	1239.61

Table 2: Average Overtime Outcome Generated

	Average	T1	T2	T3
Normal	2179.93	2010.22	2392.14	2137.42
Log-Normal	20103.22	18122.33	22773.94	19313.40

Table 3: Average and Single Objective Value

Based on this result, it is clear that as the daily opening time of ORs becomes smaller, the total cost increases because the amount of overtime increases significantly and produces a very high cost. When the OR opening length is fixed, it is also clear that total costs get higher when overtime penalty is high. We also observe that decreasing waiting time to 0 does contribute to reduction of total cost, but not in a large scale. Furthermore, when the daily opening session is long enough for OR, total cost becomes fixed since overtime all reduces to 0 for each OR. In conclusion, the length of OR opening plays a crucial role in minimizing total cost due to its limitation on overtime; however, in reality, it is unlikely and unreasonable to extend OR opening to 24 hours, and it is more important to come up with some methods to reduce overtime cost, which affects the outcome more than other two costs.

6 Conclusion

6.1 Model Conclusion

In this report, we developed a two-stage multi-server/multi-OR model for surgery allocation and

Total time/Cost ratio (over:idle:waiting)	10:1:1	5:1:0	5:1:1
480 Min	3232	2114	2422
720 Min	2102	2102	2102
300 Min	9024	5114	5428

Table 4: Sensitivity Analysis of Results

planning. This model involves decisions that need to be made sequentially: opening of ORs, assignment of patient to server, sequencing of patients at the same type of server, and lastly scheduling of surgery start time. By deriving optimal strategy for each decision, the model eventually minimizes conflicting outcomes such as overtime and waiting time.

Stochastic models that consider uncertainty in surgery duration offer the potential for significant improvement to daily OR schedules

- results from uncertain duration can provide potential instruction for health institutions in terms of daily OR schedules
- scheduling time and sequencing decisions are sensitive to opening session and cost units.
- Overtime can be greatly shortened and avoided if we set high ratio of overtime cost to waiting(or idling) time cost. However, this may also lead to higher cost when overtime cannot be avoided and the ratio is too high.
- Without incorporating overtime cost in first stage, the results show that only one OR for each server is needed. The reason can be that opening cost of a room is very high. This can change as we increase the number of patients as well as job duration.

6.2 Future Work

Even though the model is able to make reasonable allocation and scheduling plannings, several improved works could be done in the future. First of all, we could split total time horizon into discrete time blocks, and also incorporate this part into scheduling planning. In this way, the system might work more efficiently in terms of saving idling and waiting costs by forcing each block to be occupied. Additionally, we can use real data and analyze generated results, thus better understanding how to help minimize total cost. Eventually, we could add a human cost variable and take human effort into consideration.

6.3 Group Member Contributions

All group members were very involved in each part of the project and preparing the written report. Below are specifics regarding what each group member performed (note: although one task is assigned to just one person, input/suggestions from other members were crucial for all stages of analysis and writing)

Yuheng Cai - Primarily focused on the modeling part and wrote the parts of report related to these topics and proposed advanced ideas to the model formulation. Additionally contributed to every part of the project.

Shuzhe Xiao - Performed experimental design, literature review, and wrote related parts of report. Also contributed to modeling parts and other works as well.

Yuguo Zhong - Performed coding in ampl and generating test instances. Also provided work and suggestion to other parts of the project.

References

[1] Diwakar Gupta and Brian Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819, 2008.

[2] Conceição Granja, Bernardo Almada-Lobo, Filipe Janela, João Seabra, and A Mendes. An optimization based on simulation approach to the patient admission scheduling problem using a linear programing algorithm. *Journal of biomedical informatics*, 52:427–437, 2014.

[3] P Patrick Wang. Optimally scheduling n customer arrival times for a single-server system. *Computers & Operations Research*, 24 (8):703–716, 1997.

[4] Guido C Kaandorp and Ger Koole. Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229, 2007.

[5] Peter M Vanden Bosch and Dennis C Dietz. Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research*, 4(1):15–25, 2001.

[6] Peter M Vanden Bosch and Dennis C Dietz. Minimizing expected waiting in a medical ap-

pointment system. *Iie Transactions*, 32(9): 841–848, 2000.

[7] Yan Deng, Siqian Shen, and Brian Denton. Chance-constrained surgery planning under uncertain or ambiguous surgery duration. *Available at SSRN*, 2432375, 2014.

[8] Yan Deng and Siqian Shen. Decomposition algorithms for optimizing multi-server appointment scheduling with chance constraints. *Mathematical Programming*, 157(1):245–276, 2016.

[9] Brian Denton, James Viapiano, and Andrea Vogl. Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health care management science*, 10(1):13–24, 2007.

[10] Zheng Zhang and Xiaolan Xie. Simulation-based optimization for surgery appointment scheduling of multiple operating rooms. *IIE Transactions*, 47(9):998–1012, 2015.

[11] Serhat Gul, Brian T Denton, John W Fowler, and Todd Huschka. Bi-criteria scheduling of surgical services for an outpatient procedure center. *Production and Operations management*, 20(3):406–417, 2011.

Appendix: AMPL Code

```
set Job := 1 2 3 4 5;
set Server := 1 2 3 4;
set Room [1] := 1;
set Room [2] := 1;
set Room [3] := 1;
set Room [4] := 1;
set Sequence [1] := 1 3 4;
set Sequence [2] := 2 3 1 4;
set Sequence [3] := 3 2;
set Sequence [4] := 1 2 3;
set Sequence [5] := 4 1 2;

set Past [1,1] :=;
set Past [1,3] := 1;
set Past [1,4] :=1 3;
set Past [2,2] :=;
set Past [2,3] :=2;
set Past [2,1] :=2 3;
set Past [2,4] :=2 3 1;
set Past [3,3] :=;
set Past [3,2] :=3;
set Past [4,1] :=;
set Past [4,2] :=1;
set Past [4,3] :=1 2;
set Past [5,4] :=;
set Past [5,1] :=4;
set Past [5,2] :=4 1;

set Noseq [1] := 2;
set Noseq [2] := ;
set Noseq [3] := 1 4;
set Noseq [4] := 4;
set Noseq [5] := 3;
param op_cost :=
1   4
2   1
3   3
4   5;

param o_cost:= 1 100 2 100 3 100 4 100;
param i_cost:= 1 1 2 1 3 1 4 1;
param w_cost:= 1 1 2 1 3 1 4 1;

param d:
    1 2 3 4:=
1   59.90  0 45.89 18.32
2   49.36 33.33 53.72 19.31
3   0 7.40 10.01 0
4   45.02 14.81 42.32 0
5   57.11 17.30 0 49.12;

param y [*,*,1]
:   1   2   3   4    :=
1   1   0   1   1
2   1   1   1   1
3   0   1   1   0
4   1   1   1   0
5   1   1   0   1

;

param t[1,*]:= 1 90
[2,*]:= 1 120
[3,*]:= 1 100
[4,*]:= 1 150;
```

Figure 2: AMPL code for instance data

```
set Job;
set Server;
set Room {Server};
set Sequence {Job};
set Noseq {Job};

param or_cost {j in Server};

var v {j in Server, l in Room[j]} integer >= 0, <= 1;

var y {i in Job, j in Server, l in Room [j]} binary;

minimize total_cost:
sum{j in Server}(sum{l in Room[j]}(or_cost[j] * v[j,l]));

subject to c1{i in Job, j in Sequence[i]}:
sum{l in Room[j]} y[i,j,l] = 1;

subject to c2{i in Job, j in Noseq[i]}:
sum{l in Room[j]} y[i,j,l] = 0;

subject to c3{i in Job, j in Sequence[i], l in Room[j]}:
y[i,j,l] <= v[j,l];

subject to c4{j in Server, l in Room[j]}:
sum{i in Job}y[i,j,l] >= v[j,l];
```

Figure 3: AMPL code for stage1 model

```
#set of patient
set Job;
#set of all servers(stages)
set Server;
#set of room for server j
set Room {Server};
#process that the patient i need to go through
set Sequence {Job};
#set of server that patient i won't go to
set Noseq {Job};
#set of server that patient i go before go to server j
set Past {Job, Server};

#1 if patient i go to server j's room l
param y {i in Job, j in Server, l in Room [j]} ;
#duration of operation of patient i in server j
param d {Job, Server};
#opening session of server j room l
param t {j in Server, l in Room[j]};
#cost of overtime in server j
param o_cost {Server};
#cost of waiting time in server j
param w_cost {Server};
#cost of idle time in server j
param i_cost {Server};
#opening cost for server j
param op_cost {Server};

#1 if patient i is arranged directly ahead of job m in server j room l
var z {i in Job, m in Job, j in Server, l in Room [j]} binary;
#idling time of patient i directly following m in server j room l
var u {i in Job, m in Job, j in Server, l in Room[j]} >= 0;
#overtime of server j's room l
var o {j in Server, l in Room [j]} >= 0;
#waiting time of patient i directly following m in server j room l
var w {i in Job, m in Job, j in Server, l in Room [j]} >= 0;
#start time of patient i in server j's room l
var s {i in Job, j in Server, l in Room [j]} >= 0;
#
var sigma {i in Job, m in Job, j in Server, l in Room [j]} >= 0;
minimize total_cost:
sum {i in Job}
    (sum {m in Job}
        (sum {j in Server}
            (sum {l in Room [j]}(w_cost[j]*w[i,m,j,l] + i_cost[j]*u[i,m,j,l]))))
+ sum{j in Server}
     (sum{l in Room[j]}(o_cost[j]*o[j,l]));

subject to c1{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
y[i,j,l] >= z[i,m,j,l];

subject to c3{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
y[i,j,l] - y[m,j,l] + z[i,m,j,l] <= 1;

subject to c4{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
y[m,j,l] - y[i,j,l] + z[i,m,j,l] <= 1;

subject to c5{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
z[i,m,j,l] + z[m,i,j,l] <= 1;

subject to c6{i in Job,j in Server, l in Room[j]}:
sum{m in Job: m<>i} z[i,m,j,l] <= 1;

subject to c7{m in Job,j in Server, l in Room[j]}:
sum{i in Job: i<>m} z[i,m,j,l] <= 1;

subject to c8{j in Server, l in Room[j]}:
sum{i in Job}y[i,j,l] <= sum{i in Job}(sum{m in Job: m<>i}(z[i,m,j,l])) + 1;

subject to c10{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
w[i,m,j,l] <= 100000*y[i,j,l];

subject to c11{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
u[i,m,j,l] <= 100000*y[i,j,l];

subject to c12{i in Job, j in Server, l in Room[j], m in Job: m<>i}:
sigma[i,m,j,l] <= 100000*y[i,j,l];

subject to c13{i in Job, j in Server, l in Room[j]}:
sum{m in Job: m<>i}(w[m,i,j,l] - w[i,m,j,l] - u[i,m,j,l])  +
(sum{m in Job: m<>i}(sigma[i,m,j,l]))= d[i,j]*y[i,j,l] - (sum{m in Job: m<>i}(sigma[m,i,j,l]));

subject to c14{j in Server, l in Room[j]}:
o[j,l] - sum{i in Job}(sum{m in Job}(u[i,m,j,l])) = sum{i in Job}(d[i,j]*y[i,j,l]) - t[j,l];
subject to c15{i in Job, j in Sequence[i], k in Past[i,j], l in Room[j], l1 in Room[k]}:
s[i,j,l] >= s[i,k,l1] + d[i,k];

subject to c16{i in Job, j in Sequence[i], l in Room[j], m in Job: m<>i}:
s[i,j,l] >= s[m,j,l] - 100000*(1 - z[i,m,j,l]) + d[m,j];

subject to c17{i in Job, j in Sequence[i], k in Past[i,j], l in Room[j]}:
sum{m in Job: m <> i}(sigma[i,m,j,l] - sigma[i,m,k,l]) >= d[i,k];

subject to c18{i in Job, j in Sequence[i], l in Room[j]}:
sum{m in Job: m <> i}(sigma[m,i,j,l]) >= sum{m in Job: m <> i}(sigma[i,m,j,l]) -
100000*(1- sum{m in Job: m <> i}(z[m,i,j,l])) + d[i,j];

subject to c19{i in Job, j in Sequence[i], l in Room[j]}:
s[i,j,l] >= sum{m in Job: m <> i}(sigma[i,m,j,l]);

subject to c20{i in Job, j in Sequence[i], l in Room[j]}:
sum{m in Job: m <> i}w[i,m,j,l] = s[i,j,l] - sum{m in Job: m <> i}sigma[i,m,j,l];
```

Figure 4: AMPL code for stage2 model