

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

12-18-2015

Stochastic Goal Programming and a Metaheuristic for Scheduling of Operating Rooms

Justin William Britt
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Britt, Justin William, "Stochastic Goal Programming and a Metaheuristic for Scheduling of Operating Rooms" (2015). *Electronic Theses and Dissertations*. 5629.

<https://scholar.uwindsor.ca/etd/5629>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Stochastic Goal Programming and a Metaheuristic
for Scheduling of Operating Rooms

by

Justin Britt

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2015

©2015 Justin Britt

Stochastic Goal Programming and a Metaheuristic
for Scheduling of Operating Rooms

by

Justin Britt

APPROVED BY:

M. El-Masri
Nursing

R. Lashkari
Industrial & Manufacturing Systems Engineering

A. Azab, Co-Advisor
Industrial & Manufacturing Systems Engineering

M. F. Baki, Co-Advisor
Odette School of Business

November 27, 2015

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Health care systems in Canada provide benefits to patients but have issues with costs and wait lists. Long wait lists negatively affect patients' welfares. This in turn can increase costs because conditions can develop into more complicated ones over time. Operating rooms in a hospital are responsible for a significant portion of both costs and benefits; therefore, finding ways to use them more efficiently can reduce both the waste of tax dollars and the lengths of wait lists and can improve patients' welfares. In this research, a stochastic weighted goal programming model is proposed to perform elective surgery scheduling under uncertainty of both surgical durations and patient lengths of stay. The model generates a Master Surgical Schedule that schedules surgical teams in operating room blocks in a way that minimizes four objectives, which are the deviations between the targeted number of surgeries and the actual number of surgeries performed, the deviations between the targeted number of hours for surgeries and the actual number of hours used for surgeries, the maximum expected number of patients in the recovery ward over the course of the planning horizon, and the difference between the maximum and minimum expected numbers of patients in the recovery ward over the course of the planning horizon. In addition, the impact of cancellations on the schedule is taken into account. A simulated annealing meta-heuristic is developed to find near-optimal solutions. Discrete event simulation is used for validation and to demonstrate the system of operating rooms and recovery ward beds to relevant stakeholders in the health care sector.

ACKNOWLEDGEMENTS

I would like to acknowledge my co-advisors, Dr. Ahmed Azab and Dr. Mohammed Fazle Baki, for allowing me to work on this research. Furthermore, I would like to thank them for their guidance and suggestions. I would also like to acknowledge my committee members, Dr. Maher El-Masri and Dr. Reza Lashkari, for providing invaluable constructive criticism.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iii
ABSTRACT	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF APPENDICES	xiii
LIST OF ACRONYMS	xiv
CHAPTER 1: INTRODUCTION	1
1.1 Health Care and Insurance in Canada	1
1.2 Industrial Engineering Methodologies	2
1.2.1 Goal Programming and Stochastic Programming	2
1.2.2 Optimization Methods	3
1.2.3 Simulated Annealing	4
1.3 Scheduling Elective Surgeries Under Uncertainty	5
1.3.1 Types of Hospitals	6
1.3.2 Types of Surgical Patients	6
1.3.3 Elective Patient Flows	7

1.3.4	Uncertainties	8
1.3.5	Operating Room Scheduling	8
1.3.6	Importance of Viewing a Hospital as a System	8
1.3.7	Engineering Problem and Purpose of Research	9
1.3.8	Related Engineering Problems	9
1.3.9	Importance of Research	10
1.4	Thesis Organization	11
CHAPTER 2: LITERATURE REVIEW		12
2.1	Overview	12
2.2	Literature Relevant to OR Scheduling Problems	13
2.2.1	Job Shop and Flow Shop Approaches	13
2.2.2	Markov Chain, Markov Decision Process, Newsvendor Model, and Queue- ing Theory Approaches	15
2.2.3	Mathematical Programming Approaches	17
2.2.4	Simulation Approaches	22
2.2.5	Heuristics and Metaheuristics Approaches	23
2.3	Contributions of This Research	25
CHAPTER 3: MATHEMATICAL MODEL		26
3.1	Assumptions	26
3.2	Model Components	27
3.2.1	Indices	27

3.2.2	Deterministic Parameters	28
3.2.3	Stochastic Parameters	29
3.2.4	Deterministic Variables	31
3.2.5	Deviation Variables	31
3.2.6	Stochastic Variables	32
3.2.7	Constraints	32
3.2.8	Scenarios, Realizations, Goals, and Objective Functions	35
3.3	Cancellations	36
CHAPTER 4: METAHEURISTIC		37
4.1	Motivation, Purpose, and Importance	37
4.2	Simulated Annealing Metaheuristic for Operating Room Scheduling	37
4.2.1	Initialization and Physical Meanings of the Parameters	37
4.2.2	Initial Schedule Generation	38
4.2.3	Inner Loop	41
4.2.4	Outer Loop	42
CHAPTER 5: DISCRETE EVENT SIMULATION		44
5.1	Motivation, Purpose, and Importance	44
5.2	Discrete Event Simulation in ProModel	45
5.2.1	Locations, Entities, and Entity Flow	45
5.2.2	Arrivals	45
5.2.3	Processing	46

5.2.4	Attributes	46
5.2.5	Resources and Scheduling	46
5.2.6	Screenshot of the Discrete Event Simulation Model	46
CHAPTER 6: NUMERICAL EXAMPLE		49
6.1	Data	49
6.2	Mathematical Model	52
6.2.1	Numerical Results	52
6.2.2	Recovery Ward Utilization Graphs	54
6.2.3	Variables and Constraints	56
6.2.4	Optimality Gaps	56
6.2.5	Example Master Surgical Schedule	57
6.3	Simulated Annealing	61
6.3.1	Preliminary Numerical Results	62
6.3.2	Design of Experiments	64
6.4	Discrete Event Simulation	66
6.4.1	Simulation Results	68
CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH		71
7.1	Conclusions	71
7.2	Future Research	73
REFERENCES		76

APPENDICES	82
VITA AUCTORIS	223

LIST OF TABLES

Table 1	Simulated Annealing - Loop Order Example - Empty Schedule	40
Table 2	Simulated Annealing - Loop Order Example - Partial Schedule 1	40
Table 3	Simulated Annealing - Loop Order Example - Partial Schedule 2	40
Table 4	Simulated Annealing - Loop Order Example - Partial Schedule 3	41
Table 5	Simulated Annealing - Loop Order Example - Full Feasible Schedule	41
Table 6	Assignment of Surgical Teams and Blocks to Surgical Specialties	49
Table 7	Surgical Duration and Frequency Values [Adapted from Van Houdenhoven et al. (2007)]	51
Table 8	Mathematical Model - Underutilization of Operating Rooms and Utilization of Recovery Ward Beds	53
Table 9	Mathematical Model - Optimality Gaps	57
Table 10	Mathematical Model - Example Master Surgical Schedule - Week 1	58
Table 11	Mathematical Model - Example Master Surgical Schedule - Week 2	59
Table 12	Mathematical Model - Example Master Surgical Schedule - Week 3	60
Table 13	Mathematical Model - Example Master Surgical Schedule - Week 4	61
Table 14	Simulated Annealing - Preliminary Numerical Results	63
Table 15	Simulated Annealing - Factor Values	64
Table 16	Simulated Annealing - 2^4 Factorial Designed Experiment	65
Table 17	Simulation - Utilization of Operating Rooms	69

LIST OF FIGURES

Figure 1	Process Diagram of Elective Patient Flows [Adapted and Modified from Ballard and Kuhl (2006, p.434)]	7
Figure 2	Simulated Annealing - Initialization and Initial Schedule Generation	39
Figure 3	Simulated Annealing - Loops	43
Figure 4	Simulation - Screenshot of Model	48
Figure 5	Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before Cancellations, All Scenarios	54
Figure 6	Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before Cancellations	55
Figure 7	Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - After Cancellations	55
Figure 8	Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before and After Cancellations	56
Figure 9	Simulated Annealing - Normal Effects Plot	66
Figure 10	Simulation - Expected Recovery Ward Utilization	70

LIST OF APPENDICES

APPENDIX 1 - MATHEMATICAL MODEL - XPRESS CODE (SWGPM.MOS)	83
APPENDIX 2 - MATHEMATICAL MODEL - XPRESS DATA FILE (~DATA/ DATAFILE.DAT)	94
APPENDIX 3 - SIMULATED ANNEALING METAHEURISTIC - MATLAB CODE	120
MAIN SIMULATED ANNEALING CODE (SIMULATEDANNEALING.M)	120
FUNCTION TO CHECK FOR FEASIBILITY (ISFEASIBLE.M)	128
FUNCTION TO CALCULATE ENERGY (ENERGYFUNCTION.M)	131
FUNCTION TO GENERATE LENGTH OF STAY PROBABILITIES (GENER- ATELENGTHOFSTAYPROBABILITIES.M)	134
FUNCTION TO GENERATE SURGICAL DURATIONS (GENERATESURGICALDURATIONS.M)	135
APPENDIX 4 - MATLAB DATA FILES	136
DATA FILE WITH REAL WORLD SURGICAL DURATIONS (~ DATA/D.TXT)	136
DATA FILE WITH REAL WORLD SURGICAL FREQUENCIES (~ DATA/F.TXT)	136
DATA FILE WITH OMEGA CONSTRAINTS (~ DATA/OMEGA.TXT)	137
DATA FILE WITH PARAMETERS (~ DATA/PARAMETERS.TXT)	138
DATA FILE WITH PI CONSTRAINTS (~ DATA/PI.TXT)	138
APPENDIX 5 - DISCRETE EVENT SIMULATION - PROMODEL CODE	140

LIST OF ACRONYMS

CPU	Central Processing Unit
DSS	Decision Support System
MSS	Master Surgery Schedule or Master Surgical Schedule
OR	Operating Room
ORUU	Operating Room Underutilization
OT	Operating Theatre
PACU	Post-Anesthesia Care Unit
SA	Simulated Annealing
VBA	Visual Basic for Applications

CHAPTER 1: INTRODUCTION

In Chapter 1, background information relevant to the problem is provided. The specific engineering problem being considered and its importance are then explained.

1.1 Health Care and Insurance in Canada

Developed countries typically have some form of universal public health coverage. Canada is no exception to this. Health care and health insurance in Canada are complicated to some extent by jurisdictional issues. Although the federal government provides some transfer funding for health care and has some powers because of the Canada Health Act and other laws, health care and health insurance are primarily provincial and territorial issues; therefore, there is no single Canadian health care system (Deber, 2003, pp. 20-21). In each province and territory, people who meet residency and other eligibility requirements can register for public health insurance, which covers many common medical issues (Deber, 2003, p. 20). Some procedures are not covered by provincial health care plans; thus, some patients may have to rely on private sources, such as private insurance provided by employers, to pay for treatments (Deber, 2003, p. 22).

Although universal public health coverage is available, some Canadians still choose to travel abroad for medical treatment. This suggests that some Canadians may not have confidence in their public health care systems. Determining why these people choose to leave can provide some insights into how health care systems can be improved from the perspective of patients and potential patients. According to Snyder et al. (2011), Canadians engage in medical tourism primarily because of wait lists (pp. e142-e143). Reducing the time that patients have to wait for surgeries therefore is an important goal to achieve.

From the perspective of taxpayers, provincial and territorial governments, and the federal government, the main issue with Canadian style health care is and will continue to be funding. If all of the provincial and territorial budgets in Canada were combined, then

approximately 38% of the money would be spent on health care (Canadian Institute for Health Information, 2015c, p. 18). There are three main factors that will continue to increase the cost of health care in Canada: an increasing population, an aging population, and a desire for cutting-edge procedures. Canada currently has one of the highest population growth rates in the developed world (Statistics Canada, 2014, p. 14). Clearly, the increased number of people will strain the health care system; therefore, provincial and territorial governments will have to devote more funding to health care. The median age of Canadians in 2014 was nearly 10 years higher than in 1984; furthermore, the over-65 and over-85 age cohorts in Canada have population growth rates that are significantly higher than the 15-64 cohort (Statistics Canada, 2014, pp. 56-59). The aging population will strain health care systems; consequently, funding will have to be increased. Finally, according to Snyder et al. (2011), the secondary reason that Canadians engage in medical tourism is because other countries have procedures that are not available in Canada (p. e143). If provincial and territorial governments want their residents to retain confidence in their health care systems, then they will have to fund more of these procedures in order to prevent Canadians from travelling to other countries to procure health care.

Industrial Engineering methodologies can be used to help fix some of these issues. The following section explains the concepts used in this proposed thesis.

1.2 Industrial Engineering Methodologies

In this section, relevant Industrial Engineering methodologies are explained.

1.2.1 Goal Programming and Stochastic Programming

Goal programming is a form of mathematical programming that makes it possible to consider multiple objectives. A goal is set for each objective, and the deviation between each objective and its goal are minimized. There are three common types of goal programming: weighted, lexicographic, and min-max (Tamiz et al., 1998, p. 570). In weighted goal programming, each objective is multiplied by a weight assigned to it, and the overall objective function, which is the Archimedian sum of all these, is minimized (Tamiz et al., 1998, p.

570). In lexicographic goal programming, the objectives are assigned priorities; next, they are ranked by priority from highest to lowest; subsequently, the first objective is minimized by itself, and a constraint is set after the optimization to prevent the next optimization from obtaining a worse result; finally, this procedure is repeated for all of the objectives (Tamiz et al., 1998, p. 570). In min-max goal programming, the maximum difference between any goal and its objective is minimized (Tamiz et al., 1998, p. 570).

Stochastic programs have two or more stages. In a two-stage problem, first stage decisions are made under uncertainty of second stage parameters (Kall & Wallace, 1994, p. 9). In the second stage, there are a finite or infinite number of scenarios that contain realizations of second stage parameters (Shapiro et al., 2009, p. 30). If there are a finite number of scenarios, then the overall objective function is the sum of the first stage costs and the expected value of the objective functions from the second stage (Shapiro et al., 2009, p. 4).

1.2.2 Optimization Methods

There are two main types of optimization methods: exact and approximate. An exact optimization method will find the best feasible solution whereas an approximate method will just find a feasible solution (Talbi, 2009, p.18).

There are different types of exact optimization methods. Branch and bound, branch and cut, and branch and price are commonly used exact optimization methods that use tree structures to enumerate solutions until the optimal solution is found (Talbi, 2009, p. 19). Other examples of exact methods are dynamic programming, constraint programming, and iterative deepening algorithms. (Talbi, 2009, p.19).

Approximate optimization methods are divided into two classes: approximation algorithms and heuristic algorithms. An approximation algorithm is specific to a problem being solved and is designed to find a solution with a certain level of optimality within a certain amount of time; a heuristic algorithm finds a near-optimal solution but does not have any guarantees with respect to the level of optimality or the amount of time required to find the solution

(Talbi, 2009, pp. 21-23).

Heuristic algorithms are also divided into two classes: specific heuristics and metaheuristics. A specific heuristic can only solve a specific problem whereas a metaheuristic contains a higher level heuristic that controls one or more lower level heuristics (Talbi, 2009, p. 21).

The appropriate optimization method depends on the problem being considered. If the problem is small, then an exact method, such as branch and bound, usually will be appropriate. However, large problems often can not be solved using exact methods. This is referred to as intractability (Gigerenzer, 2008, p. 21). Approximate methods should be used for large problems. Approximation algorithms and specific heuristics are problem specific and typically obtain solutions that are far from optimal; for these reasons, they are not particularly useful (Talbi, 2009, pp. 21-23). Metaheuristics are not problem specific and can find near-optimal solutions reasonably quickly; accordingly, they are useful for solving large problems (Talbi, 2009, p. 23).

1.2.3 Simulated Annealing

Simulated annealing (SA) is a metaheuristic proposed by Kirkpatrick et al. (1983) and Černý (1985) that is inspired by the actual process of annealing. At the start of the metaheuristic, a random initial solution is generated (Nahar et al., 1986, p. 294). The outer loop of the metaheuristic starts at some initial "temperature" (Nahar et al., 1986, p. 294). A new neighbouring solution is then found (Nahar et al., 1986, p. 294). If the new solution is feasible and better than the current solution, then it is accepted (Nahar et al., 1986, p. 294). If it is feasible and worse, then it may or may not be accepted (Kirkpatrick et al., 1983, pp. 672-673). A condition developed by Metropolis et al. (1953), which uses a random number, is used to decide whether the worse solution should be accepted; this aspect of the metaheuristic prevents it from getting stuck in local optima (Kirkpatrick et al., 1983, pp. 672-673). More new neighbouring solutions are found at that temperature until the inner loop length is reached; subsequently, the temperature in the outer loop is lowered; next, this

process repeats until the final temperature is reached (Nahar et al., 1986, p. 294). SA is a stochastic metaheuristic; ergo, running it over the same problem multiple times will obtain different results (Kirkpatrick et al., 1983, p. 674).

In an SA metaheuristic, there are four parameters: initial temperature, final temperature, cooling rate, and inner loop length. These parameters are important because they control the metaheuristic. However, they do not have any physical meanings outside of the context of the metaheuristic itself. The initial temperature needs to be high enough so that nearly 100% of solutions are accepted (Kirkpatrick et al., 1983, p. 675). This allows the metaheuristic to diversify. The lower temperature needs to be low enough to ensure that only improved solutions will be accepted near the end of a run of the metaheuristic. This allows the metaheuristic to intensify towards a good solution. The inner loop length controls the number of iterations at each temperature, and the cooling rate controls the number of temperature levels at which solutions are considered. Both of these parameters therefore impact the run time of the metaheuristic.

1.3 Scheduling Elective Surgeries Under Uncertainty

In this section, the engineering problem being considered and the purpose of this research are explained.

In a hospital that has an Operating Theatre (OT), there are different types of patients who require different types of surgeries. The flow of a particular patient in the hospital depends on the type of surgery the patient requires. A surgical team, which consists of one or more surgeons, nurses, and other medical professionals, can use an operating room (OR) in the OT to perform a surgery on a patient. The surgical teams and the patients need to be scheduled into the ORs.

1.3.1 Types of Hospitals

Different jurisdictions can classify hospitals into various categories. In Ontario, Regulation 964, which is associated with the Public Hospitals Act, classifies hospitals into 22 alphabetic groups (2001). The most relevant hospital classifications are groups A, B, and C, which correspond to general teaching hospitals, general hospitals with more than 100 beds, and general hospitals with fewer than 100 beds, respectively (2001).

It should be noted that in addition to ORs in general hospitals, there are trauma centers that specialize in emergent surgeries. Ontario currently has nine trauma centers for adult patients (Gomez et al., 2014, p. E177).

1.3.2 Types of Surgical Patients

There are three main cases of surgical patients: emergent, urgent, and elective. Emergent surgical patients need to be treated immediately or may suffer serious health problems or death (Christensen & Kockrow, 2014, p. 18). Urgent surgical patients also will suffer if they are not treated, but they are able to wait for some period of time (Christensen & Kockrow, 2014, p. 18). Elective surgical patients can wait significant periods of time without suffering significantly; consequently, these patients can decide when they wish to be operated on. (Christensen & Kockrow, 2014, p. 18). Elective surgical patients can be further split into two different groups. An elective inpatient needs to stay in a bed in the hospital after being treated (Christensen & Kockrow, 2014, p. 18). An elective outpatient does not need to stay in a bed and can be discharged from the hospital shortly after the surgery (Christensen & Kockrow, 2014, p. 18). All types of elective surgical patients clearly are different than emergent and urgent ones, but the distinction between emergent and urgent surgical patients is less clear. Dexter and O'Neill (2001) consider an emergent case to be one where a surgical patient needs to be treated by a surgical team that is already in the hospital whereas an urgent case is one where the patient can wait for a surgical team to arrive from outside the hospital (p. 664).

1.3.3 Elective Patient Flows

Different types of elective patients will flow through different parts of the hospital. Figure 1, which is adapted and modified from Ballard and Kuhl (2006), is a process diagram that shows these paths. Note that an elective inpatient requires a full examination before an operation; an elective outpatient does not require the examination (Ballard & Kuhl, 2006, p. 435). Also, note that an elective outpatient leaves shortly after being operated on and thus does not recover in the ward (Ballard & Kuhl, 2006, p. 435). In addition to these intra-hospital patient flows, inter-hospital patient flows can occur. In some cases, patients, especially the ones needing elective procedures, may be transferred between hospitals for a variety of reasons (Robinson et al., 2009).

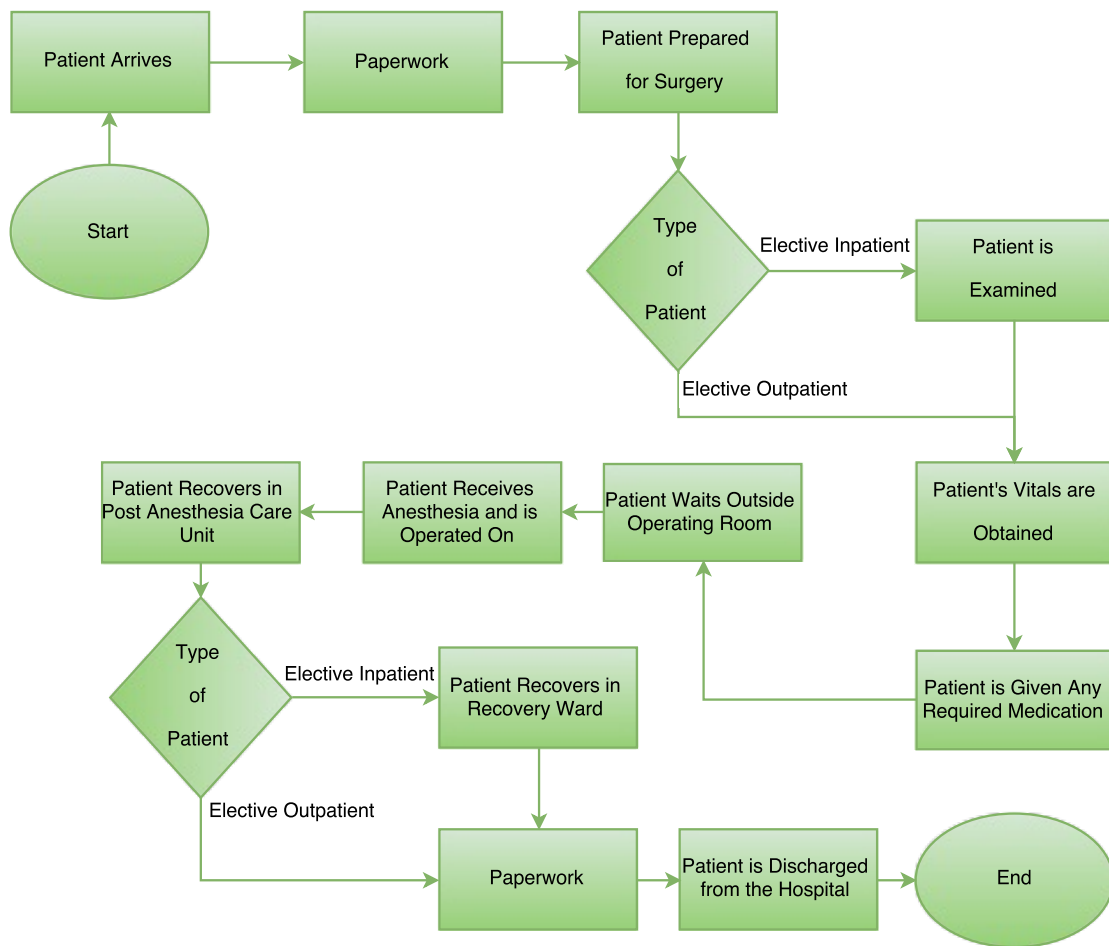


Figure 1: Process Diagram of Elective Patient Flows [Adapted and Modified from Ballard and Kuhl (2006, p.434)]

1.3.4 Uncertainties

A hospital can collect data to determine how long it takes to perform surgeries or how long patients stay in the recovery ward. It is also possible to use other means to obtain relevant data. For example, it is possible to estimate surgical durations in Ontario by using billing data from anesthesiologists (Redelmeier et al., 2008). However, these data values are averages. The actual length of an individual surgery may be longer or shorter than the average. Similarly, it is not known in advance exactly how long a given patient will recover in the ward. These uncertainties in surgical durations and patient lengths of stay need to be accounted for when scheduling is performed.

1.3.5 Operating Room Scheduling

OR scheduling has multiple decisions that need to be made over different time frames. In tactical scheduling, surgical teams have to be scheduled within a planning horizon, which might be a week or a month. There are three main ways to do this: block scheduling, open scheduling, and modified block scheduling (Fei et al., 2010, p. 222). In block scheduling, each OR is divided into time blocks for the duration of the planning horizon (Chaabane et al., 2008, p. 173). Surgical teams are assigned to blocks by following some criteria (Chaabane et al., 2008, p. 173). The schedule generated by following this process is the Master Surgical Schedule (MSS), which can be re-used cyclically (Tanfani & Testi, 2010, p. 106). In open scheduling, surgical teams are not guaranteed specific time blocks (Chaabane et al., 2008, pp. 172-173). Instead, a surgical team requests to use a particular OR on a certain day (Chaabane et al., 2008, pp. 172-173). Scheduling assignments are performed based on the order in which the requests were received (Chaabane et al., 2008, pp. 172-173). Modified block scheduling uses aspects of both block scheduling and open scheduling (Chaabane et al., 2008, p. 173).

1.3.6 Importance of Viewing a Hospital as a System

The research is primarily concerned with the scheduling of ORs. However, it is important to view a hospital as a system. Pam Bush, who has over 20 years of experience managing various units in hospitals, is now the Clinical Director Stakeholder Engagement, Informa-

tion Services, and also is a Perioperative Coach at The Ottawa Hospital, stresses this by explaining that "operating rooms have always been the focus of attention, due to their resource intensity. However, many perioperative leaders do not appreciate that one inefficient perioperative unit will negatively impact the others because of interdependencies. For example if the [Post-Anesthesia Care Unit (PACU)] does not have clinically based discharge criteria, patients will block PACU beds, bringing the ORs to a standstill. Likewise an operating room schedule that does not evenly distribute resource requirements across the week, will cause chaos in the surgical day care unit, the ORs and PACU" (Ontario Ministry of Health and Long-Term Care, 2010, p. 33).

1.3.7 Engineering Problem and Purpose of Research

The purpose of this research is to investigate the problem of designing a system of ORs and recovery ward beds in a way that optimally allocates blocks to surgical teams. This involves generating an MSS for elective inpatients that optimizes utilization of both the ORs and the beds in the recovery ward under uncertainties of both surgical durations in the ORs and patient lengths of stay in the recovery ward. Only elective surgeries are considered. Emergent and urgent patients are ignored. The ORs are optimized by minimizing both deviations from targets set by the hospital during strategic scheduling and OR underutilization. The recovery ward is optimized by minimizing both the maximum expected number of patients in the recovery ward and the difference between the maximum and minimum expected number of patients in the recovery ward. The solution to this problem can help maximize the welfares of patients and alleviate some of the aforementioned issues, such as long wait lists and high costs, that are negatively affecting health care systems in Canada.

1.3.8 Related Engineering Problems

This research focuses on a system of ORs and beds. However, the problem being considered here could easily be modified into other engineering problems that consider the designs of other systems in hospitals, such as diagnostic imaging departments.

1.3.9 Importance of Research

Computers are now ubiquitous. However, health care sectors in developed countries, especially Canada, have not fully adopted computers and related information technologies. In Canada, fewer than 20% of primary care physicians and 30% of secondary care physicians regularly use electronic health records to exchange information with other doctors (Protti, 2015, p. 422). Many hospitals in Canada and other countries still use manual techniques to schedule surgeries (Blake et al., 2002; Roland et al., 2010). The schedules used by these hospitals therefore may not be utilizing the ORs and the recovery ward beds in an efficient manner. Using the ORs and the recovery ward beds inefficiently leads to higher costs and longer wait lists.

Long wait lists are a problem both because they negatively affect patients' welfares and because the health of a given patient might deteriorate while waiting for a surgery. This can also lead to higher costs because relatively simple conditions can quickly develop into major conditions. For example, in Ontario in fiscal year 2012-2013, surgery for esophagitis had an estimated average direct cost to a hospital of \$3913; a major intervention to the esophagus, which could be necessary if esophagitis is not treated quickly enough, had an estimated average cost to a hospital of \$42796 (Canadian Institute for Health Information, 2015d). Note that both cost values are averages across all age groups. This is just one example, but it shows how simple conditions that could have been treated for not much money can develop into major conditions that are more expensive. In Canada, hospitals are expected to spend nearly \$65 billion in 2015 (Canadian Institute for Health Information, 2015b). Clearly, there is a significant opportunity to reduce costs and potentially save millions of dollars by implementing this research.

Another factor to consider is that manual scheduling techniques are time consuming. For example, the manager of an operating theatre at a hospital in Belgium had to devote an entire work day every week to manually generate feasible advance and allocation schedules

(Roland et al., 2010, p. 219). Health care professionals have more important things to do; for this reason, the amount of time they spend on developing schedules should be minimized. This research can resolve this issue.

This research therefore is not some unnecessary luxury. Maximizing patients' welfares should be a priority for all hospitals. Hospitals need to adopt this research in order to reduce the length of wait lists, prevent the health of patients from getting even worse, and reduce costs.

1.4 Thesis Organization

In this thesis, there are seven chapters, which are organized as follows.

In this chapter, an overview of both health care and insurance in Canada is included. Also, this chapter has an overview of relevant Industrial Engineering methodologies. Finally, the specific engineering problem being considered is explained.

In chapter 2, a review of literature relevant to OR scheduling problems is provided.

In chapter 3, a stochastic weighted goal programming model is proposed for creating an optimal Master Surgical Schedule under uncertainty of both surgical durations and patient lengths of stay.

In chapter 4, a simulated annealing metaheuristic is developed and used to quickly obtain near-optimal solutions.

In chapter 5, a discrete event simulation model is created and used to demonstrate and validate the design of the system of ORs and recovery ward beds.

In chapter 6, a numerical example is provided.

Finally, chapter 7 contains conclusions and a discussion about future research.

CHAPTER 2: LITERATURE REVIEW

In chapter 2, a review of literature on operating room (OR) scheduling is provided.

2.1 Overview

According to Testi et al. (2007), OR scheduling can be divided into three hierarchical phases: strategic, tactical, and operational. In the strategic phase, long term decisions regarding case mix are made. These decisions involve the allocation of OR time and capacity to different surgical specialties. In a hospital in Ontario, strategic decisions need to account for the needs of patients, the needs of surgeons who perform surgeries at the hospital, priorities in the community, and other priorities developed by strategic decision makers (Ontario Ministry of Health and Long-Term Care, 2010, p. 8). In the tactical phase, a Master Surgical Schedule (MSS) is developed. The MSS is used to assign the different surgical specialties to time blocks in the OR schedule. Typically, the MSS will have a planning horizon of a few weeks or months and can be used on a cycle until new strategic decisions are made (Tànfani & Testi, 2010, p. 106). Alternatives to block scheduling include modified block scheduling and open scheduling. In the operational phase, the overall Surgical Scheduling Problem is split into two sub-problems: advance scheduling and allocation scheduling. In advance scheduling, which is done before the day of surgery, actual surgeries are assigned into blocks in the MSS. In allocation scheduling, which is performed on the actual day of surgery, the sequence of events needed to perform each procedure are determined and scheduled into the corresponding block.

Abdelrasol et al. (2013) propose a framework that also uses a three phase approach for OR scheduling. In the first stage, a stochastic dynamic programming model is used to make the strategic case mix decisions. In the second stage, modified block scheduling is used to both allocate elective surgery cases to ORs and to route emergency patients. In addition, simulation is used for sequencing. In the third stage, a system dynamics model is used to

improve the overall efficiency.

There are a variety of approaches available for scheduling ORs at strategic, tactical, and operational phases. Before reviewing these approaches, it is important to reiterate that the three phases proposed by Testi et al. (2007) are hierarchical. Some articles in the literature only consider scheduling at one phase whereas others try to integrate one or more phases. In addition, it should be noted that these phases do not have standard definitions in the literature; thus, an article on tactical scheduling might not use the same definition of the word tactical as Testi et al. (2007).

2.2 Literature Relevant to OR Scheduling Problems

This section contains a literature review of OR scheduling approaches. For other literature reviews on OR scheduling, see Cardoen et al. (2010), Guerriero and Guido (2011), and May et al. (2011).

2.2.1 Job Shop and Flow Shop Approaches

Two common manufacturing systems are job shops and flow shops. Both systems consist of multiple machines, which are also referred to as stages, that are used to complete jobs (Johnson, 1954). In a job shop, different jobs require the use of different machines in different orders; in a flow shop, all jobs use the same machines in the same order (Sule, 2008, p. 10). There are two main types of flow shops: classical and hybrid. In a classical flow shop, there is only one machine in each stage (Allaoui & Artiba, 2004, p. 433). In a hybrid flow shop, at least one stage contains multiple machines; if a stage has more than one machine, then they operate in parallel (Linn & Zhang, 1999, p. 57). Job shops and flow shops can both be affected by blocking; this is the situation where a finished job must stay in a machine because there is no storage space available outside of the machine (Sule, 2008, p. 13). Scheduling techniques for both manufacturing systems often minimize the makespan, which is the total amount of time it takes to finish all jobs (Sule, 2008, p. 10). All of the approaches in this subsection involve the application of manufacturing scheduling techniques to OR scheduling problems.

Pham and Klinkert (2008) apply two mixed integer programs of multi-mode blocking job shops to the elective surgery case scheduling problem. The first model schedules elective surgeries, and the second inserts three types of add-on surgeries: emergency surgeries, urgent surgeries, and additional elective surgeries. This online job shop scheduling approach can be used for both allocation and external resource, but not advance, scheduling of inpatients and outpatients in both public and private hospitals. The authors provide a numerical example of a daily schedule for two days but do not use actual data to validate their model.

Guinet and Chaabane (2003) propose a linear program for advance scheduling and a two-stage hybrid flow shop model for allocation scheduling. The first stage of the flow shop accounts for the surgical duration in the OR plus time spent cleaning after the procedure, and the second stage accounts for time spent in the recovery area. In Chaabane et al. (2008), this model is compared to a different model under both block and open scheduling. Jebali et al. (2006) consider a similar problem as Guinet and Chaabane (2003) and Marques et al. (2012) and develop a deterministic mixed integer linear program for advance and allocation scheduling that is solved using a two stage method. In the first stage, surgeries are assigned to an OR in a way that minimizes three things: surgery lead time, OR overutilization, and OR underutilization (ORUU). In the second stage, which is similar to the flow shop model in Guinet and Chaabane (2003), the operations that must be performed are planned in a way that minimizes overtime.

Fei et al. (2010) develop two models for advance and allocation scheduling within an open scheduling system. The first model is an integer linear program of a set partitioning model that assigns surgery dates over a one week period. The second model is a two-stage hybrid flow shop model that creates daily schedules. The OR is the first stage of the flow shop, and the recovery room is the second stage. A novelty of this model is that recovery time can be split between the OR and the recovery room.

A more complicated flow shop model is proposed by Augusto et al. (2010). They use a four-

stage hybrid flow shop model with blocking to investigate elective allocation scheduling within an open scheduling system. The resources considered are OR beds, transporters, and recovery beds. A novelty of this model is that patients can recover in OR beds if recovery beds are unavailable. However, resource costs are not taken into account. They suggest testing the robustness of the model against uncertainties in surgery durations and patient recovery times.

Xiang et al. (2013) develop a mathematical model of a flexible job shop that minimizes the makespans of the surgeries. All variables, aside from times, are 0-1. Unlike the models in Fei et al. (2010) and Augusto et al. (2010), their model does not consider surgeons as being interchangeable because each surgeon has different skills and abilities. In Xiang et al. (2015), this model is extended to make nurses non-interchangeable.

2.2.2 Markov Chain, Markov Decision Process, Newsvendor Model, and Queueing

Theory Approaches

Many articles in this category use variants of the newsvendor model; this is a classic model for inventory management that involves the tradeoff between procuring too many copies of a perishable object (e.g. newspapers), which results in overage costs, and buying too few, which results in underage (i.e. stockout) costs (Porteus, 2002, p. 7-17). In OR scheduling problems, the newsvendor model is relevant because capacity, which is perishable, needs to be allocated. Olivares et al. (2008) consider the problem of allocating capacity to specific elective surgery cases in a cardiac department. They apply a structural estimation approach to a newsvendor model to evaluate the balance between allocating too much time for a case versus not allocating enough time. Zonderland et al. (2010) also propose a newsvendor model for the capacity allocation aspect of strategic scheduling. This model minimizes cancellations of elective surgeries because of preemptions caused by semi-urgent surgeries while also minimizing the time scheduled for semi-urgent surgeries. This minimizes ORUU. Their model has three components: a queueing model that determines how much

capacity is needed to perform all semi-urgent surgeries, another queueing model that evaluates elective surgery cancellations versus ORUU, and a Markov Decision Process solved using policy iteration. Choi and Wilhelm (2014b) also consider capacity allocation but instead develop a non-linear stochastic model to maximize excess revenue by minimizing three expected costs: costs associated with failing to provide service to patients, costs associated with OR overutilization, and costs associated with ORUU. This base model is then developed into four models. Two of these models are based on the newsvendor model, and the other two linearize the base model into stochastic integer programs. Choi and Wilhelm (2014a) instead consider scheduling at the tactical level by formulating two mathematical models for creating block surgical schedules for an intermediate time period. The first is a newsvendor model that determines the block duration, and the second is a sequential newsvendor model, which is novel, that determines the block sequence. They also enhance their models to account for patients who fail to show up. Instead of overbooking, they accomplish this by considering patient no-shows, who follow a discrete probability distribution, and then recalculating the optimal block duration.

Tancrez et al. (2009) establish a method for both strategic and tactical scheduling that uses a Markov Chain approach to investigate the impact of stochastic operating times and disruptions caused by emergency surgeries. Their model minimizes overtime and has two types of ORs: dedicated rooms for emergencies and versatile rooms that can handle both elective and emergency patients. They use a simulation package to validate their model. Tancrez et al. (2013) improve this model by including the recovery ward. A case study of a hospital in Belgium is also included.

Gerchak et al. (1996) propose using an infinite horizon Markov Decision Process and a dynamic stochastic programming model to create an advance schedule that maximizes the number of elective surgeries that can be performed while still being able to service emergency patients. Both elective and emergency surgeries have uncertain demands. Their

model finds the optimal policy that maximizes the expected profit. Although this model is primarily for Canadian hospitals that do not use OR block schedules, it could be applied to hospitals in other countries.

Astaraky and Patrick (2015) use an infinite horizon Markov Decision Process and an approximate dynamic programming model, which uses simulation and approximate policy iteration, for the advance scheduling of elective and emergency surgeries with the objective of minimizing three things: surgery lead time, OR overtime, and ward congestion. Unlike Gerchak et al. (1996), they create the advance schedule a variable number of days in advance. Parizi and Ghate (2015) extend the model in Astaraky and Patrick (2015) by including overbooking, cancellations and no-shows.

2.2.3 Mathematical Programming Approaches

Denton et al. (2007) develop a two-stage stochastic mixed integer model, which considers uncertainties in surgery durations, that simultaneously sequences and schedules elective surgeries on a daily basis. The model works under both block and open scheduling. In the case study, they use real world data from a non-profit hospital in the United States.

Generation of a Master Surgical Schedule (MSS) is a problem considered by many authors. Blake et al. (2002) propose an integer linear programming model for creating an MSS that minimizes both the underutilization of allocated hours and the differences between weekly schedules. They present a case study that uses data from a hospital in Toronto. Santibáñez et al. (2007) extend the models in Blake et al. (2002) and Blake and Donald (2002) by considering the Master Surgical Schedule Problem in the context of a group of hospitals in British Columbia. Their mixed integer linear program is flexible in a manner that objectives can be turned into constraints; thus, the model can be used to design different cyclic MSS that influence a variety of performance measures across the entire group of hospitals. Testi and Tànfani (2009) also consider the tactical Master Surgical Schedule Problem but also include the operational Surgical Case Assignment Problem. They develop an integer linear

program that minimizes costs associated with patients' welfares. They use data from the Department of General Surgery at an Italian hospital to verify their model.

Multiple authors consider the problem of optimizing and balancing bed utilization. Beliën and Demeulemeester (2007) propose a non-linear program, which is linearized into a mixed integer linear program, for creating a cyclic MSS that minimizes the total expected bed shortage in the ward. The number of patients in an OR block and lengths of stay are both stochastic. This model is incorporated into a decision support system in Beliën et al. (2009). Yahia et al. (2014) propose a mixed integer linear program for creating an MSS that levels both bed occupancy and nurse workloads. The approach for creating the MSS is similar to Testi et al. (2007) and Mannino et al. (2012). Surgery duration and lengths of stay are both deterministic. Surgeons' preferences are considered in a novel way. The model is solved using Lingo, which results in a small optimality gap. A case study using data from a non-profit hospital in Egypt is used to demonstrate the model. van Oostrum et al. (2008) develop a two-phase stochastic mixed integer linear program to create a cyclic MSS for elective procedures that are performed frequently. Less frequent elective procedures and all emergencies are ignored. Surgery durations are stochastic. They propose extending their model by including the uncertainty of the demand for beds.

A similar problem is considered by Li et al. (2015). They develop a deterministic mixed integer goal programming model for tactical elective surgery scheduling. The model optimizes the utilization of the ORs and the recovery ward. It also considers the impact of cancellations at different levels on the schedule. They propose modifying the model to account for uncertainties in surgery durations and patient lengths of stay.

Capacity scheduling has been investigated by many authors. Adan and Vissers (2002) develop an integer linear program for strategic and tactical capacity scheduling that satisfies patient throughput targets while also minimizing the deviation between actual and targeted resource utilization values. The resources that are included in the model are beds, operating

theater capacity, nursing capacity, and intensive care beds. Vissers et al. (2005) investigate a similar problem but consider different resources (operating theater time, medium care beds, intensive care beds, and intensive care nurses), focus on tactical scheduling, and apply their model to a cardiothoracic department, which is more likely to have emergency patients than the orthopedics department used in Adan and Vissers (2002). The mixed-integer linear program in Vissers et al. (2005) creates a cyclic MSS for elective surgeries, which are classified into different categories, and reserves space for emergencies. Each patient category has different average resource requirements. In Adan et al. (2009), this model is extended by using stochastic lengths of stay in both medium and intensive care units. However, emergencies are ignored; hence, the MSS only takes elective surgeries into account. In Adan et al. (2011), this model is extended to include emergency patients, a weighted goal programming approach, and operational scheduling. Also, this model incorporates cancellations by scanning through the categories of patients. However, they suggest developing a better approach for dealing with cancellations. Dellaert et al. (2015) also extend the tactical scheduling model in Adan et al. (2009) by determining mathematical relationships of probability distributions for both waiting times and the utilization of resources. The probability distributions can then be used to see how a given MSS will affect both waiting times and resource utilization. In addition, they develop novel strategies to improve both of these performance measures. They, unlike Adan et al. (2011), ignore emergencies.

Other authors use probability distributions to solve OR scheduling problems. Vanberkel et al. (2011a) develop a model that uses binomial distributions to represent patient discharges. Next, these discharge values are used to calculate bed utilization. Finally, pair-wise swaps in the MSS are made to improve the schedule. Vanberkel et al. (2011b) use the same approach but include mathematical relationships of probability distributions associated with the number of patients in the ward, patient admissions, patient discharges, and the number

of patients who have been recovering in the ward for a certain number of days.

Jebali and Diabat (2015) propose a two-stage stochastic mixed integer program with recourse for advance scheduling. Their model considers elective patients, but not emergency patients, and accounts for uncertainties in surgery durations and lengths of stay in both the intensive care unit and the ward. They use the data from Adan et al. (2011) for their case study.

Marques et al. (2012) investigate the Elective Case Scheduling Problem at a hospital in Lisbon and develop an integer linear program for both advance and allocation scheduling that maximizes OR utilization. The model considers elective patients at four different priority levels and ambulatory patients. Also, outpatients and maternity patients are ignored because the hospital in the case study does not provide those services. Their model is solved in two stages. In the first stage, elective surgeries are planned. In the second phase, scheduling for the ambulatory patients, who have a dedicated OR, is performed. In Marques et al. (2014), the same model is extended to maximize both OR utilization and patient throughput.

Agnetis et al. (2014) develop two integer programs for the decomposition of tactical and operational elective surgery scheduling. The first models the process of creating a weekly MSS as a minimum cost flow problem. The second models advance scheduling as a multiple knapsack problem. These problems are solved sequentially. San Giuseppe Hospital in Italy is used as a case study. Agnetis et al. (2012) use the same hospital to study the impact of using a weekly MSS, which is created using an integer linear program, that changes over a one year period.

Conforti et al. (2010) propose a deterministic mixed integer program for both tactical and operational scheduling of elective surgeries under a block scheduling system. Their model has four objectives: maximization of OR utilization for all ORs, maximization of the num-

ber of procedures performed on high priority patients, maximization of the satisfaction of preferences of each surgical specialty, and minimization of the underutilization of each surgical specialty. Also, they use preliminary data from an Italian hospital to perform a small case study.

Aringhieri, Landa, Soriano, et al. (2015) develop a 0-1 linear program for block scheduling and advance scheduling over a one week planning horizon. Unlike the approaches used by Agnetis et al. (2014) and Conforti et al. (2010), the scheduling phases in this model are performed simultaneously instead of sequentially. They calculate the complexity of the OR scheduling problem being considered. This model only considers elective patients because there are dedicated resources for emergencies. Patients can use care beds on the weekend. A case study with scenarios generated from real world data is included.

Aringhieri, Landa, and Tànfani (2015) propose a 0-1 linear program for advance scheduling of elective surgeries over some given planning horizon and within a given MSS. Their model levels ward bed utilization by maximizing the quantity of ward beds being used at the lowest possible bed utilization level. This also maximizes OR utilization.

Determining how to allocate resources to different types of surgeries is considered by a few authors. Kuo et al. (2003) propose a linear program for the strategic case mix problem that maximizes revenues for surgeons at Duke University Medical Center. The authors assume that beds downstream to the ORs are always available. Hans et al. (2008) consider a similar problem at tactical and operational phases.

Sier et al. (1997) develop a non-linear mixed integer program for daily elective and emergency surgery scheduling that minimizes conflicts while prioritizing younger children over older children and adults, higher priority cases over lower priority ones, and longer operations over shorter ones.

2.2.4 Simulation Approaches

The problem of scheduling patient admissions is considered by Granja et al. (2014) in the context of a radiology department in Portugal. They develop a simulation model of the process that patients go through and use it to find bottlenecks. They also use the simulation model to determine how a different configuration would affect the efficiency of the process. Their model does not consider the impact of scheduling patients in different orders and does not provide any information related to costs and benefits.

Zhang et al. (2009) propose a mixed integer programming model with a finite horizon for tactical scheduling. Their model, which can prioritize emergency patients over elective patients and can constrain the usage of OR resources, determines the master surgical schedule that minimizes the length of stay for patients. However, their model ignores uncertainties associated with surgery durations and patient arrivals. In addition to the mixed integer programming model, they use a simulation model to find both the average length of stay for patients and OR utilization.

Ivaldi et al. (2003) use a discrete event simulation model to manage a wait list. The model considers both elective and emergent surgeries; a constraint reserves beds for emergent cases. A case study, which uses data collected over a two year period from a hospital in San Genoa, Italy, is included. The simulation model can be used to investigate the impact of increasing the amount of time available for surgeries on the length of the wait list.

Banditori et al. (2013) develop a mixed integer linear program for creating an MSS for elective surgeries that considers due dates and has a waiting list that can be managed. Emergency patients are ignored because they do not use the same resources as elective patients. They use a simulation model to evaluate the ability of the MSS to handle variations in both patient lengths of stay and surgery durations. Cappanera et al. (2014) expand this model by proposing a multi-objective mixed integer program that maximizes patient throughput and resource leveling while still being able to adapt to uncertainties associated

with patient lengths of stay and surgery durations. The resources that are considered and leveled are surgeons, ORs, and ward beds. Also, an additional leveling criterion not considered by Banditori et al. (2013) is included. They again use a simulation model to see how the model responds to the uncertainties.

Ferrand et al. (2014) use a simulation model to investigate different methods of handling elective emergent surgeries. In general hospitals, there are two different approaches for dealing with emergent surgeries. In the first approach, an OR is reserved for emergent surgeries. In the second approach, emergent and urgent surgeries can be scheduled into any of the ORs. These two approaches can be described as the focused approach and the flexible approach, respectively. It is also possible to use a combination of both approaches. It is unclear which approach is better. According to Ferrand et al. (2014), a partially flexible approach reduces wait times for both elective and emergent surgeries. Heng and Wright (2013) develop a mathematical model and show that having a dedicated OR for emergent surgeries in a children's hospital in Toronto decreases wait times for low priority emergent surgeries, decreases cancellations of elective surgeries, and decreases the total overtime associated with elective surgeries. However, Wullink et al. (2007) use a mathematical model to compare the two approaches and find that the flexible approach reduces wait times for emergent surgeries, reduces overtime, and increases OR utilization.

2.2.5 Heuristics and Metaheuristics Approaches

A variety of approaches using heuristics and/or metaheuristics exist in the literature. Some of these approaches focus on tactical scheduling. Blake and Donald (2002) consider the same hospital as Blake et al. (2002) but develop a post-solution heuristic to find better solutions. van Oostrum et al. (2008) use a column generation heuristic to solve their model that generates cyclic MSS. Beliën and Demeulemeester (2007) develop a repetitive mixed integer program heuristic, a quadratic mixed integer program heuristic, and a simulated annealing metaheuristic to solve their model. Li et al. (2015) extend the work of Beliën and

Demeulemeester (2007) but only use a commercial solver to solve the model. They suggest developing a metaheuristic that can be used to solve larger scheduling problems. Adan et al. (2011) and Yahia et al. (2014) also suggest developing heuristics and/or metaheuristics to solve their models. Aringhieri, Landa, Soriano, et al. (2015) include a novel two-level tabu search metaheuristic that simultaneously performs both block scheduling and advance scheduling. Conforti et al. (2010) also consider both tactical and operational scheduling and use a genetic algorithm.

Many of the approaches using heuristics and/or metaheuristics involve operational scheduling problems. Sier et al. (1997) use a simulated annealing metaheuristic to solve their model. Van Houdenhoven et al. (2007) compare various approaches, including some that use heuristics, for scheduling surgical cases. Pham and Klinkert (2008), like Li et al. (2015), suggest developing a metaheuristic in order to make it possible to solve large scheduling problems. They specifically recommend a tabu search metaheuristic. Guinet and Chaabane (2003) use a Hungarian primal-dual heuristic to solve their models that perform advance and allocation scheduling. Fei et al. (2010) use a heuristic based on column generation to solve their advance scheduling model. Furthermore, the two models for advance scheduling and allocation scheduling are also solved using a hybrid genetic algorithm. Augusto et al. (2010) and Parizi and Ghate (2015) both use Lagrangian relaxation heuristics to solve their models. Riise and Burke (2011) apply a local search heuristic to an operational scheduling model. Vijayakumar et al. (2013) develop a first fit decreasing heuristic to solve their mixed integer dual bin-packing model; surgical cases are assigned priorities and are then assigned by order of priority. Silva et al. (2015) extend Pham and Klinkert (2008), Riise and Burke (2011), and Vijayakumar et al. (2013) by considering specialized human resources who can perform multiple surgeries at the same time; a relax-and-fix heuristic and a linear programming based heuristic are used to solve the model. Xiang et al. (2013) and Xiang et al. (2015) use a modified ant colony optimiza-

tion metaheuristic to solve their operational scheduling models. They also show that their metaheuristic outperforms a different scheduling model that uses discrete event simulation. Denton et al. (2007) use several heuristics to solve their mathematical programming model that schedules elective surgeries on a daily basis. Aringhieri, Landa, and Tànfani (2015) develop reduced and adaptive variable neighborhood search metaheuristics to solve their advanced scheduling model. Marques et al. (2012) use a genetic algorithm to solve large instances of their model for advanced and allocation scheduling.

2.3 Contributions of This Research

The literature review shows important trends in research related to OR scheduling. Many articles consider uncertainties in surgical durations and/or uncertainties in patient lengths of stay. Authors of other articles suggest extending their models to account for these uncertainties. Some articles use simulation approaches to test, demonstrate, and validate models. Many articles contain heuristics and/or metaheuristics. Simulated annealing, genetic algorithm, tabu search, and ant colony optimization seem to be the most common types of metaheuristics used. Furthermore, some authors suggest developing metaheuristics in order to solve larger problem instances.

The aim of this research is to extend the deterministic mixed integer goal programming model from Li et al. (2015), which optimizes both ORs and recovery ward utilization, by doing the following:

1. Consider uncertain surgical durations in the ORs
2. Consider uncertain patient lengths of stay in the recovery ward
3. Develop a simulation model to demonstrate and validate the mathematical model
4. Develop a metaheuristic to get a near-optimal solution in a reasonable amount of time

CHAPTER 3: MATHEMATICAL MODEL

In this chapter, a stochastic weighted goal programming model is proposed for creating an optimal Master Surgical Schedule (MSS) that maximizes the utilization of operating rooms (OR) and balances the utilization of beds in the recovery ward under uncertainty of both surgical durations and patient lengths of stay. The MSS is only for elective surgeries.

The model is designed to be as general as possible and is not tailored to any specific hospital. However, different hospitals can vary in size and in structure. Some aspects of the model thus might be more relevant at certain hospitals. For example, a given orthopedic surgical team at a large hospital in Toronto may only perform knee surgeries whereas a given orthopedic surgical team at a smaller hospital in Windsor may perform many different types of surgery. Therefore, the concept of using patient mix clusters is relevant at the hospital in Windsor but may not be as relevant at the hospital in Toronto. Also, the large hospital in Toronto has more ORs, more surgical teams, more surgical specialties, and more recovery ward beds.

3.1 Assumptions

In this model, a number of assumptions are made to simplify the problem.

1. Various strategic decisions are made by hospital administrators in advance of any tactical scheduling. These strategic decisions affect the number of patients a surgical team will operate on in a planning horizon, the number of blocks assigned to surgical teams in a planning horizon, the number of blocks assigned to surgical specialties in a planning horizon, and the maximum patient lengths of stay. Note that strategic decisions can change over time.
2. Elective patients for all surgical specialties are selected from wait lists that always have available supplies of patients who require surgeries.

3. Emergent and urgent patients are not operated on in these ORs.
4. Patients and all surgical team members always arrive on time.
5. A selected patient is operated on by a surgical team from the appropriate surgical specialty. The duration of a surgery is not known with certainty in advance. The patient is then transferred to a bed in the ward to recover.
6. Patients may pass through the Post-Anesthesia Care Unit (PACU) in between the ORs and the recovery ward. However, this unit is ignored because different surgical specialties may have different requirements and policies regarding how patients should be recover from anesthesia.
7. Surgeries can only be scheduled on weekdays because all ORs are closed on Saturdays and Sundays.
8. The recovery ward is always open and always has a surplus of beds.
9. The amount of time that a given patient stays in the recovery ward is not known with certainty in advance.
10. Scheduled surgeries can be cancelled. All cancellations are made after the schedule is generated.

3.2 Model Components

In this section, the parameters, variables, constraints, and objective functions used in the model are explained.

3.2.1 Indices

The following parameters are used to index other parameters.

- | | |
|-----|--|
| i | $i = 1, 2, \dots, m$ indexes the surgical teams. |
| j | $j = 1, 2, \dots, n$ indexes the operating rooms. |
| k | $k = 1, 2, \dots, T$ indexes the days of the planning horizon. |

- l $l = 0, 1, 2, \dots, h_i$ indexes the days of the patient lengths of stay in the recovery ward.
- e $e = 1, 2, \dots, E_i$ indexes the types of surgery associated with each surgical team i .
- r $r = 1, 2, \dots, R$ indexes the surgical specialties.

3.2.2 Deterministic Parameters

In the strategic scheduling phase, the following parameters are set over the planning horizon for each surgical team i .

- N_i Each surgical team i should perform surgeries on N_i patients.
- L_i Each surgical team i should perform surgeries in at least L_i blocks.
- H_i Each surgical team i should perform surgeries in at most H_i blocks.
- h_i Each surgical team i should perform surgeries on patients in assigned blocks. After surgeries are completed, patients can recover in beds. The maximum patient length of stay in the recovery ward for patients operated on by surgical team i should not exceed h_i days.

In the strategic scheduling phase, the following parameters are set over the planning horizon for each surgical specialty r .

- S_r Each surgical team i is assigned to one of R surgical specialties. The set of surgical teams in each surgical specialty is S_r .
- B_r Each surgical specialty r is assigned B_r blocks.

The following parameters are also used.

- n The Operating Theatre contains n ORs.

m	There are m surgical teams.
T	The planning horizon has T days.
R	Each surgical team i is assigned to one of R surgical specialties.
E_i	Each surgical team i performs E_i types of surgery. Each type of surgery is a cluster. This represents the patient mix.
v	Each block in an OR is v hours long.
O_{ie}	Each cluster e of each surgical team i requires an overhead time multiplier O_{ie} to account for the time needed to set up before the procedure and to clean up after the procedure.
f_{ie}	Each cluster e of each surgical team i represents a number of procedures that is a fraction f_{ie} of the total number of procedures associated with all clusters in surgical team i .
Π_{ij}	An OR j may or may not be available for a surgical team i . If the OR j is available for surgical team i , then $\Pi_{ij} = 0$. If the OR j is not available for surgical team i , then $\Pi_{ij} = 1$.
Ω_{ik}	A day k in the planning horizon may or may not be available for a surgical team i . If the day k is available for surgical team i , then $\Omega_{ik} = 0$. If the day k is not available for surgical team i , then $\Omega_{ik} = 1$.

3.2.3 Stochastic Parameters

The following stochastic parameters are used.

NS	There are NS scenarios. Each scenario has realizations of surgical durations and patient lengths of stay.
ω	$\omega = 1, 2, \dots, NS$ indexes the scenarios.

- D_{ie}^ω and \tilde{D}_{ie} In scenario ω , if a surgery associated with cluster e of surgical team i is performed, then the surgical duration will be D_{ie}^ω , where D_{ie}^ω is a realization of the stochastic parameter \tilde{D}_{ie} in scenario ω .
- p_{il}^ω and \tilde{p}_{il} In scenario ω , if a surgery is performed by surgical team i , then the patient will stay in the recovery ward for l days with probability p_{il}^ω , where p_{il}^ω is a realization of the stochastic parameter \tilde{p}_{il} in scenario ω .
- A_i^ω In scenario ω , a surgery associated with cluster e that is performed by surgical team i has a duration of D_{ie}^ω . This duration accounts for the time it takes to perform the surgery but does not account for the time it takes to set up for the surgery and the time it takes to clean the OR after the surgery. In scenario ω , the total expected surgical duration A_i^ω across all clusters e for surgical team i is the sum of the products of the surgical durations D_{ie}^ω , the time multipliers O_{ie} for setting up before the surgery and cleaning after the surgery, and the relative fractions f_{ie} at which surgeries are performed. A_i^ω for a given surgical team i and for a given scenario ω equals $\sum_{i=1}^m \sum_{e=1}^{E_i} D_{ie}^\omega O_{ie} f_{ie}$.
- n_i^ω The length of a block is v hours. This limits how many surgeries can be performed by a surgical team in a block. In scenario ω , each surgical team i can perform surgeries on an expected maximum of $n_i^\omega = \left\lfloor \frac{v}{A_i^\omega} \right\rfloor$ patients in a block.
- Φ_{il}^ω In scenario ω , if surgical team i uses a block today, then the recovery ward is expected to have a cumulative total of Φ_{il}^ω patients operated on by surgical team i in l days. Patient counts are performed every day at 18:00. If a patient leaves the hospital before that time, then that patient is not counted on that day. Therefore, this parameter counts how many

nights each patient from surgical team i stays in the recovery ward. If a maximum length of stay h_i is longer than the length T of the planning horizon, then the recovery ward will have patients operated on by surgical team i from multiple planning horizons. Φ_{il}^ω in a given scenario ω equals $\sum_{\tau=0}^{\lfloor h_i/T \rfloor} \sum_{l'=l+T\tau+1}^{h_i} n_i^\omega P_{il'}^\omega$.

3.2.4 Deterministic Variables

x_{ijk} is a 0-1 decision variable that represents the scheduling decisions.

$$x_{ijk} = \begin{cases} 1 & \text{if a decision is made to assign OR } j \text{ to surgical team } i \text{ on day } k, \\ 0 & \text{if OR } j \text{ is not assigned to surgical team } i \text{ on day } k. \end{cases}$$

3.2.5 Deviation Variables

These deviation variables are used in the goal constraints and are minimized.

$d_{1i}^{\omega-}$ In scenario ω , $d_{1i}^{\omega-}$ is the deviation between the target number of surgeries N_i and the actual number of surgeries performed by surgical team i , where the actual number of surgeries is less than N_i .

$d_{1i}^{\omega+}$ In scenario ω , $d_{1i}^{\omega+}$ is the deviation between the target number of surgeries N_i and the actual number of surgeries performed by surgical team i , where the actual number of surgeries is greater than N_i .

$d_{2ijk}^{\omega-}$ In scenario ω , $d_{2ijk}^{\omega-}$ is the deviation between the total number of hours available for surgical team i in OR j on day k and the actual number of hours used by surgical team i in OR j on day k , where the actual number of hours used is less than the total number of hours available.

3.2.6 Stochastic Variables

These stochastic variables account for the utilization of the recovery ward, which is downstream to the ORs.

I_k^ω On day k of the planning horizon in scenario ω , the recovery ward is expected to have I_k^ω patients.

I_{max}^ω Over the course of all days k in the planning horizon in scenario ω , the recovery ward is expected to have a maximum of I_{max}^ω patients.

I_{min}^ω Over the course of all days k in the planning horizon in scenario ω , the recovery ward is expected to have a minimum of I_{min}^ω patients.

3.2.7 Constraints

The following constraints are used.

Constraint 1

A surgical team i may prefer not to be scheduled in OR j . If $\Pi_{ij} = 1$, then the surgical team i cannot be assigned to OR j . If $\Pi_{ij} = 0$, then the surgical team i can be assigned to OR j .

Therefore, the product of Π_{ij} and x_{ijk} must sum to 0 for all days k .

$$\sum_{i=1}^m \sum_{j=1}^n \Pi_{ij} x_{ijk} = 0 \quad \forall k = 1, 2, \dots, T \quad (1)$$

Constraint 2

A surgical team i may prefer not to be scheduled on day k . If $\Omega_{ik} = 1$, then the surgical team i cannot be assigned to day k . If $\Omega_{ik} = 0$, then the surgical team i can be assigned to day k . Therefore, the product of Ω_{ik} and x_{ijk} must sum to 0 for all OR j .

$$\sum_{i=1}^m \sum_{k=1}^T \Omega_{ik} x_{ijk} = 0 \quad \forall j = 1, 2, \dots, n \quad (2)$$

Constraint 3

Each block can be assigned to at most one surgical team i . This constraint prevents multiple surgical teams from being assigned to the same OR j on the same day k .

$$\sum_{i=1}^m x_{ijk} \leq 1 \quad \forall j = 1, 2, \dots, n \quad (3)$$

$$\forall k = 1, 2, \dots, T$$

Constraint 4

A surgical team i can be allocated at most one block per day. This constraint prevents a surgical team from being assigned to multiple ORs on the same day k .

$$\sum_{j=1}^n x_{ijk} \leq 1 \quad \forall i = 1, 2, \dots, m$$

$$\forall k = 1, 2, \dots, T$$
(4)

Constraint 5

In the strategic phase of scheduling, each surgical team i is allocated between L_i and H_i blocks to be used over the planning horizon.

$$L_i \leq \sum_{j=1}^n \sum_{k=1}^T x_{ijk} \leq H_i \quad \forall i = 1, 2, \dots, m$$
(5)

Constraint 6

In the strategic phase of scheduling, each surgical specialty r is allocated B_r blocks to be used over the planning horizon.

$$B_r = \sum_{i \in S_r} \sum_{j=1}^n \sum_{k=1}^T x_{ijk} \quad \forall r = 1, 2, \dots, R$$
(6)

Constraint 7

In the strategic phase of scheduling, each surgical team i is assigned a target number of surgeries N_i to be performed over the planning horizon.

$$n_i^\omega \sum_{j=1}^n \sum_{k=1}^T x_{ijk} + d_{1i}^{\omega-} - d_{1i}^{\omega+} = N_i \quad \forall i = 1, 2, \dots, m$$

$$\forall \omega = 1, 2, \dots, NS$$
(7)

Constraint 8

All ORs j should be in use at all times.

$$(A_i^\omega n_i^\omega - v) x_{ijk} + d_{2ijk}^{\omega-} = 0 \quad \forall i = 1, 2, \dots, m$$

$$\forall j = 1, 2, \dots, n$$

$$\forall k = 1, 2, \dots, T$$

$$\forall \omega = 1, 2, \dots, NS$$
(8)

Constraint 9

The Master Surgical Schedule is cyclical, so the recovery ward on day k at 18:00 in scenario ω is expected to have some patients from the current planning horizon and some from the previous horizon.

$$I_k^\omega = \sum_{i=1}^m \sum_{l=1}^k \sum_{j=1}^n \Phi_{i(k-l)}^\omega x_{ijl} + \sum_{i=1}^m \sum_{l=k+1}^T \sum_{j=1}^n \Phi_{i(k+T-l)}^\omega x_{ijl} \quad \forall k = 1, 2, \dots, T \quad (9)$$

$$\forall \omega = 1, 2, \dots, NS$$

Constraint 10

All I_k^ω must be between I_{min}^ω and I_{max}^ω .

$$I_{min}^\omega \leq I_k^\omega \leq I_{max}^\omega \quad \forall k = 1, 2, \dots, T \quad (10)$$

$$\forall \omega = 1, 2, \dots, NS$$

Constraint 11

x_{ijk} is a 0-1 variable.

$$x_{ijk} \in \{0, 1\} \quad \forall i = 1, 2, \dots, m$$

$$\forall j = 1, 2, \dots, n \quad (11)$$

$$\forall k = 1, 2, \dots, T$$

Constraints 12-18

All variables must be non-negative.

$$x_{ijk} \geq 0 \quad \forall i = 1, 2, \dots, m$$

$$\forall j = 1, 2, \dots, n \quad (12)$$

$$\forall k = 1, 2, \dots, T$$

$$d_{1i}^{\omega-} \geq 0 \quad \forall i = 1, 2, \dots, m \quad (13)$$

$$\forall \omega = 1, 2, \dots, NS$$

$$d_{1i}^{\omega+} \geq 0 \quad \forall i = 1, 2, \dots, m \quad (14)$$

$$\forall \omega = 1, 2, \dots, NS$$

$$\begin{aligned}
d_{2ijk}^{\omega-} &\geq 0 \quad \forall i = 1, 2, \dots, m \\
&\quad \forall j = 1, 2, \dots, n \\
&\quad \forall k = 1, 2, \dots, T \\
&\quad \forall \omega = 1, 2, \dots, NS
\end{aligned} \tag{15}$$

$$\begin{aligned}
I_k^{\omega} &\geq 0 \quad \forall k = 1, 2, \dots, T \\
&\quad \forall \omega = 1, 2, \dots, NS
\end{aligned} \tag{16}$$

$$I_{max}^{\omega} \geq 0 \quad \forall \omega = 1, 2, \dots, NS \tag{17}$$

$$I_{min}^{\omega} \geq 0 \quad \forall \omega = 1, 2, \dots, NS \tag{18}$$

3.2.8 Scenarios, Realizations, Goals, and Objective Functions

In the model, there are two stages. In the first stage, the scheduling decisions x_{ijk} are made under uncertainty of both surgical durations and patient lengths of stay. In the second stage, there are NS scenarios. In each scenario ω , there is a realization D_{ie}^{ω} of the surgical duration parameter \tilde{D}_{ie} and a realization p_{il}^{ω} of the length of stay probability parameter \tilde{p}_{il} .

The weighted goal programming model uses four goals in all scenarios. The four goals and their importance are as follows.

Goal 1 is the sum of deviations between the targeted number of surgeries and the actual number of surgeries performed. Minimizing goal 1 ensures that strategic targets are met.

$$\text{Goal 1: } g_1^{\omega} = \sum_{i=1}^m d_{1i}^{\omega-}$$

Goal 2 is the sum of the deviations between the targeted number of hours for surgeries and the actual number of hours used for surgeries. Minimizing goal 2 ensures that the ORs are used as much as possible.

Goal 2: $g_2^\omega = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^T d_{2ijk}^{\omega-}$

Goal 3 is the maximum expected number of patients in the recovery ward over the course of the planning horizon. Minimizing goal 3 ensures that the number of recovery ward beds required is as low as possible.

Goal 3: $g_3^\omega = I_{max}^\omega$

Goal 4 is the difference between the maximum and minimum expected numbers of patients in the recovery ward over the course of the planning horizon. Minimizing goal 4 ensures that the bed utilization in the ward is balanced.

Goal 4: $g_4^\omega = I_{max}^\omega - I_{min}^\omega$

Each scenario is a deterministic problem. In each scenario, each goal g is multiplied by a factor w_g . The overall objective function, which is minimized, is the expected value of all the weighted goals associated with the scenarios.

Minimize $E_{\omega=1,2,\dots,NS} [w_1 \sum_{i=1}^m d_{1i}^{\omega-} + w_2 \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^T d_{2ijk}^{\omega-} + w_3 I_{max}^\omega + w_4 (I_{max}^\omega - I_{min}^\omega)]$

3.3 Cancellations

Cancellations happen after all other decisions have been made. For every scheduling assignment (i.e. $x_{ijk} = 1$) a value c_{ijk} is generated from some given distribution. If a given c_{ijk} value fails to meet some condition (e.g. it is too far from the mean of the distribution from which the c_{ijk} values were generated), then the assignment is cancelled.

CHAPTER 4: METAHEURISTIC

In this chapter, a simulated annealing metaheuristic is developed and applied to the model from chapter 3.

4.1 Motivation, Purpose, and Importance

The mathematical model is a mixed integer program that generates a Master Surgical Schedule (MSS). There are a variety of commercial solvers available that can solve this kind of problem. These programs typically use a branch and bound search to find an exact solution to the problem. This search technique uses a tree structure and can be exponential in terms of complexity (Jeroslow, 1974). Different hospitals may have different numbers of surgical teams, operating rooms (ORs), and/or planning horizon lengths. All of these affect the size of the problem, which can range from thousands to hundreds of thousands of variables and constraints. If the problem is moderately sized, then a commercial solver can be used to generate an MSS. However, if the problem is large, then the commercial solver may take too long and/or utilize too much memory.

A way of finding a near-optimal schedule is to use a metaheuristic; note that near-optimal does not mean exact. In this chapter, a simulated annealing metaheuristic is developed because it is necessary for solving large scheduling instances.

4.2 Simulated Annealing Metaheuristic for Operating Room Scheduling

There are four main components in the metaheuristic: initialization, initial schedule generation, an inner loop, and an outer loop. The components of the metaheuristic are explained in detail in the following subsections.

4.2.1 Initialization and Physical Meanings of the Parameters

In the initialization component, values for the parameters in metaheuristic are set. The parameters are the initial temperature, the final temperature, the cooling rate, and the maximum number of accepted solutions per inner loop. Note that the parameters have no physical meanings outside of the context of the metaheuristic itself. These values can af-

fect both the speed of the metaheuristic and the quality of the schedule obtained; hence, the values of the parameters should be determined through experimentation. Furthermore, the energy function, which is the same as the objective function in the mathematical model, is defined. Figure 2 is a flowchart of both the initialization and initial schedule generation components; this flowchart connects to Figure 3.

4.2.2 Initial Schedule Generation

An initial schedule that can be improved is needed as a starting point. The schedule needs to be feasible because an infeasible schedule cannot be implemented. The number of blocks available in the schedule first needs to be calculated. The schedule is then generated by looping through the surgical teams, the operating rooms (ORs), and the days in the planning horizon and making assignments until no more blocks are available. After each assignment is made, the partially completed schedule is run through a feasibility function, which contains the constraints, to ensure that the assignment does not violate any of the constraints. If a constraint is violated, then the assignment is rejected.

The order in which the assignments are made is important for two reasons. The first reason is that constraint 5 in the model from chapter 3 has a lower bound and an upper bound for each surgical team i . Before the metaheuristic can make assignments to any surgical team, it first satisfies all of the lower bounds associated with constraint 5. The second reason that the assignment order matters is because some blocks are available to more surgical teams than others. A small example can be used to demonstrate this. Consider the process of developing a schedule for an operating theater that has five surgical teams and five ORs. There are various constraints that need to be satisfied; thus, some assignments cannot be made. Table 1 shows the initial state of the schedule before any assignments are made. If an OR contains a letter x , then it is unavailable to that surgical team.

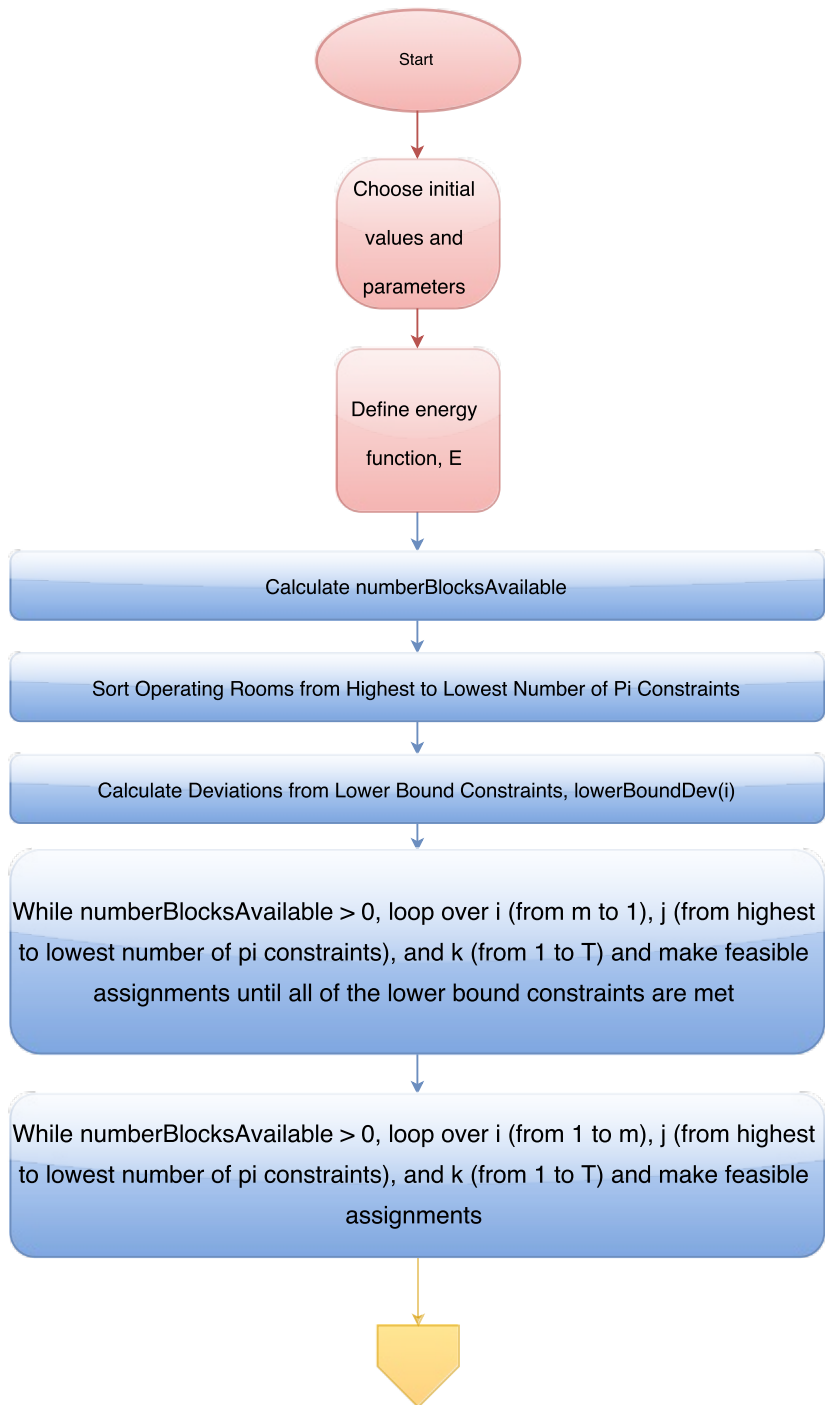


Figure 2: Simulated Annealing - Initialization and Initial Schedule Generation

		j				
		1	2	3	4	5
i	1			x		
	2					x
	3		x			x
	4					x
	5			x		x

Table 1: Simulated Annealing - Loop Order Example - Empty Schedule

Potential assignments are made into the schedule by looping over surgical teams i and ORs j and then running the schedule through a feasibility function. First, try assigning surgical team 1 to OR 1.

		j				
		1	2	3	4	5
i	1	1		x		
	2					x
	3		x			x
	4					x
	5			x		x

Table 2: Simulated Annealing - Loop Order Example - Partial Schedule 1

The schedule is run through a feasibility function and does not violate any constraints; as a result, this assignment is accepted. Now try assigning surgical team 2 to OR 2.

		j				
		1	2	3	4	5
i	1	1		x		
	2		1			x
	3		x			x
	4					x
	5			x		x

Table 3: Simulated Annealing - Loop Order Example - Partial Schedule 2

Again, it passes the feasibility test; thus, the assignment is accepted. If this process is repeated for surgical teams 3 and 4, then this feasible partial schedule is obtained.

		j				
		1	2	3	4	5
i	1	1		x		
	2		1			x
	3		x	1		x
	4				1	x
	5			x		x

Table 4: Simulated Annealing - Loop Order Example - Partial Schedule 3

OR 5 is only available to surgical team 1. However, surgical 1 is already assigned to OR 1. Therefore, an assignment for OR 5 cannot be made; consequently, a full schedule cannot be generated. This is why the looping order matters. If the number of x constraints is taken into account before the looping process, then the following feasible schedule can be obtained.

		j				
		1	2	3	4	5
i	1			x		1
	2		1			x
	3		x	1		x
	4				1	x
	5	1		x		x

Table 5: Simulated Annealing - Loop Order Example - Full Feasible Schedule

In the actual model, constraint 1, which prevents the assignment of certain surgical teams i to certain OR j , is the equivalent of the x constraints in the small example. The SA metaheuristic therefore calculates the number of Π_{ij} constraints per OR and then loops over j from the highest to lowest number of Π_{ij} constraints. This allows a full schedule to be generated.

4.2.3 Inner Loop

In the inner loop, random values of i , j , k , j_2 , and k_2 are chosen. If $i = 30$, then $i_2 = 1$. Otherwise, $i_2 = i + 1$. If $x_{ijk} = x_{i_2j_2k_2}$, then new values of i , j , k , i_2 , j_2 , and k_2 are chosen. Otherwise, the values of x_{ijk} and $x_{i_2j_2k_2}$ are swapped. The new schedule is run through the

feasibility function to ensure that it does not violate any constraints and thus is feasible. If it is infeasible, then the inner loop starts over. If it is feasible, then the new energy of the schedule is calculated. If the new energy is less than the current energy, then the new schedule and new energy are accepted as the current schedule and current energy, respectively. If the new energy is greater than or equal to the current energy, then a random real number RN is generated from the uniform distribution $U(0, 1)$. If $e^{-\frac{\text{New Energy} - \text{Current Energy}}{\text{Current Temperature}}} > RN$, then the new schedule and new energy are accepted as the current schedule and current energy, respectively, even though this schedule is worse. Otherwise, the schedule is rejected. The acceptance of some, but not all, worse solutions allows the metaheuristic to escape from local optima. The inner loop repeats until the maximum number of solutions per inner loop L has been reached. Figure 3 is a flow chart that shows the inner and outer loops.

4.2.4 Outer Loop

In the outer loop, the temperature starts at an initial value T_i . The temperature is decreased by multiplying the current temperature by the cooling rate α that was set during initialization. After the temperature is decreased, the metaheuristic returns to the inner loop. This process repeats until the current temperature is less than the final temperature T_f .

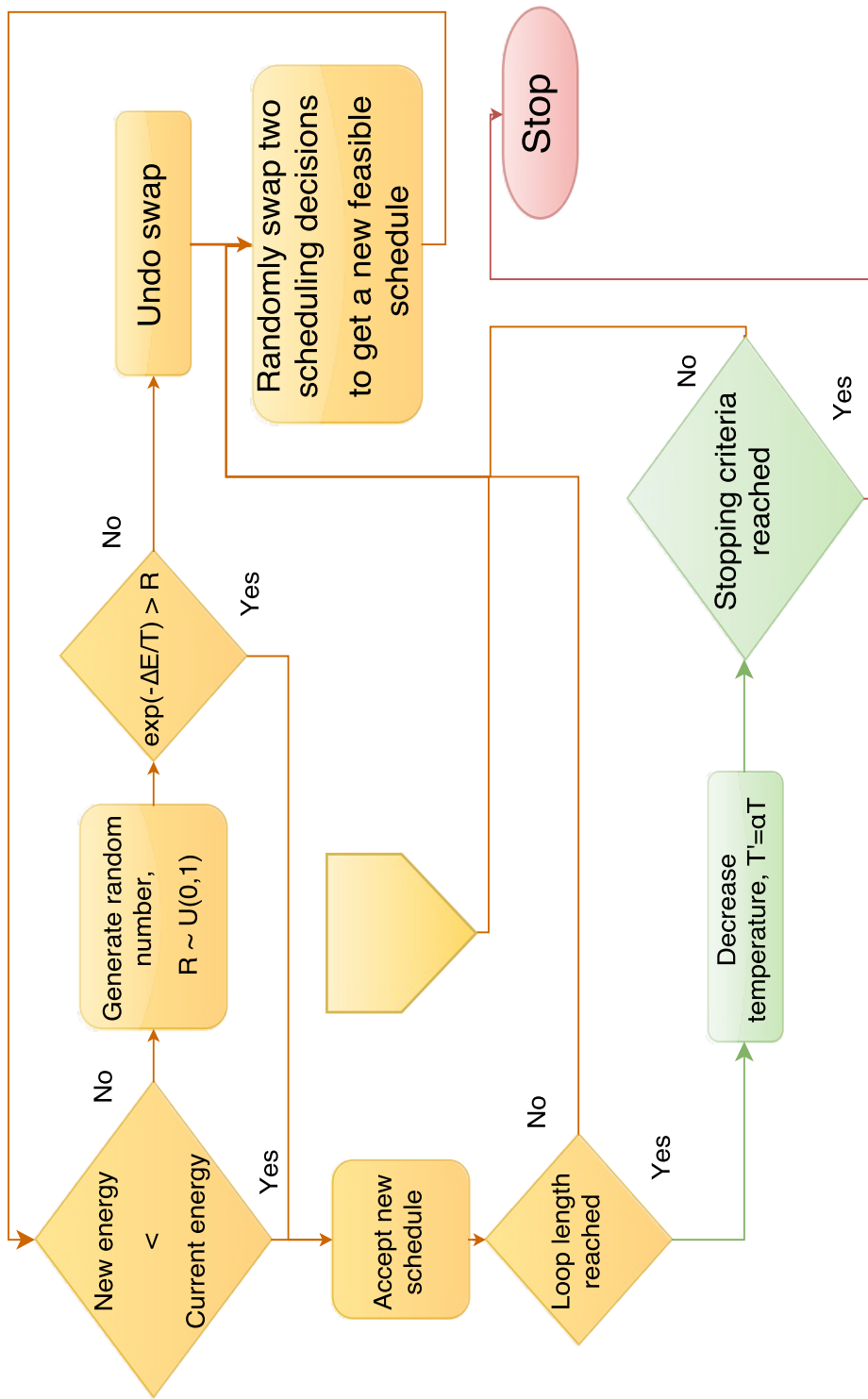


Figure 3: Simulated Annealing - Loops

CHAPTER 5: DISCRETE EVENT SIMULATION

In this chapter, a discrete event simulation model is developed.

5.1 Motivation, Purpose, and Importance

The mathematical model in chapter 3 and the simulated annealing (SA) metaheuristic in chapter 4 are useful for people who have a background in operations research. However, they may not make sense to others. Discrete event simulation models can be understood by nearly anyone. Therefore, the main purpose of the discrete event simulation model is to demonstrate the impact that schedules have on the operating rooms (ORs) and the recovery ward beds. Schedules are not generated using the discrete event simulation model. The schedules can be generated using either the mathematical model or the simulated annealing (SA) metaheuristic. In addition, the schedules can be generated manually.

The discrete event simulation model is necessary because it can be used to show that the schedules generated by the mathematical model and the SA metaheuristic use the system of ORs and recovery ward beds more efficiently than manually generated schedules. This can be done by generating a schedule using a manual technique, generating a second schedule using the mathematical model, and generating a third schedule using the SA metaheuristic. Each schedule can then be simulated. The discrete event simulation model produces results for utilizations of both the ORs and the recovery ward beds. These results can be used to compare the performance of the three schedule generation techniques. This is an important part of convincing relevant stakeholders in the health care sector that this research can lead to actual improvements.

Furthermore, the mathematical model has a variety of assumptions that may not be completely realistic. The discrete event simulation model can be used to change some of the less realistic assumptions.

5.2 Discrete Event Simulation in ProModel

ProModel is a commercial discrete event simulation package. A model in ProModel has four primary aspects: locations, entities, arrivals, and processes. In addition to these primary aspects, there are a variety of other aspects that can be included in models. Relevant aspects are explained in the following subsections. Furthermore, a screenshot of the model used in the numerical example is provided.

5.2.1 Locations, Entities, and Entity Flow

Patients are the only entities that flow through the system. Patients are selected from wait lists. A patient who enters the system is placed in the Arrival Buffer and then proceeds to the Arrival Area. Next, the patient is placed in the Surgery Buffer. Subsequently, the patient is moved to the appropriate OR. After the surgery is complete, the patient is moved to the Recovery Buffer. The patient then is moved again to the Initial Recovery Area, which represents the Post-Anesthesia Care Unit. The patient is then transported to the Recovery Ward Buffer. The next location for the patient is the Recovery Ward. Finally, the patient eventually is discharged from the hospital and enters a Discharge Buffer before being removed from the system.

All buffers in the system have infinite capacities and only exist to avoid blockages during patient flows.

5.2.2 Arrivals

If surgical team i is scheduled on day k , then a batch of patients corresponding to that surgical team arrives on that day. For each batch, a random scenario value is generated. Each batch is then split into the appropriate number of patients for that scenario. All patients arrive before the block actually starts because they need to check in and be prepared for surgery. This means that some patients might have to wait for a few hours, but that is not a problem because a comfortable waiting area is available for patients who are scheduled to be operated on later in the day.

5.2.3 Processing

The simulation model has a process that corresponds to each physical non-buffer location; accordingly, there are four processes: Arrival, Surgery, Initial Recovery, and Recovery.

At the Arrival Process, the patients check in and fill out any necessary forms. Each patient is then prepared for surgery. This process always finishes before the ORs open at the start of the day.

At the Surgery Process, each patient is operated on by the appropriate surgical team in the appropriate OR. The duration of each surgery is not known in advance.

At the Initial Recovery Process, the patients recover until they are well enough to be transported to beds in the actual recovery ward.

At the Recovery Process, patients recover in beds until they are ready to be discharged. The length of stay for each patient is not known in advance.

5.2.4 Attributes

Each patient is assigned attributes that keep track of which scenario the patient falls under, which surgical team that patient belongs to, which OR the patient is assigned to, and which day the patient is operated on by the surgical team.

5.2.5 Resources and Scheduling

There are two resources that are used in the simulation: surgical teams and ORs. Surgical teams are needed during the Surgery Process. The schedule for surgical teams can be generated using either the mathematical model in chapter 3 or the simulated annealing metaheuristic in chapter 4; the schedule can also be generated manually. Each OR location is assigned a corresponding resource.

5.2.6 Screenshot of the Discrete Event Simulation Model

Figure 4 is a screenshot of the model. The desk is the Arrivals Area. In the numerical example in chapter 6, it is assumed that the hospital has nine ORs; thus, the nine surgical tables are the ORs. The stretchers are the Initial Recovery Area. The beds are the Recovery Ward. The boxes are buffers, and the targets show where the entities are placed. The arrows

are the routes for the patients.



Figure 4: Simulation - Screenshot of Model

CHAPTER 6: NUMERICAL EXAMPLE

In this chapter, a numerical example is provided.

6.1 Data

In the strategic phase of scheduling, multiple decisions are made. In this case, the hospital is hypothetical and does not exist yet; therefore, reasonable values are chosen to demonstrate how the model functions. It would be trivial to use different values to see how the performance would change. All random data generation, except for random numbers associated with cancellations, is performed in MATLAB with a Mersenne Twist random number generator using a seed value of 5. This is important because it ensures consistency, which makes it possible to compare results. Cancellations occur after optimization is completed; thus, consistency is not important for those values.

The hospital will have thirty surgical teams ($m = 30, i = 1, 2, \dots, 30$) and nine operating rooms (ORs) ($n = 9, j = 1, 2, \dots, 9$). Each block within the twenty-eight day planning horizon ($T = 28, k = 1, 2, \dots, 28$) will be eight hours long ($v = 8$). Each surgical team will have no more than eight clusters ($E = 8, e = 1, 2, \dots, 8$). There will be six surgical specialties ($R = 6, r = 1, 2, \dots, 6$): General, Gynecology, Plastic, Ear-Nose-Throat, Orthopedic, and Urology. Each surgical specialty will be assigned six blocks for every surgical team within the specialty. Table 6 shows how the surgical teams and blocks will be assigned to surgical specialties.

Surgical Specialty r	Surgical Teams i	Number of Blocks B_r
General	1-6	36
Gynecology	7-11	30
Plastic	12-14	18
Ear-Nose-Throat	15-16	12
Orthopedic	17-22	36
Urology	23-30	48

Table 6: Assignment of Surgical Teams and Blocks to Surgical Specialties

Note that the assignments in Table 6 are for the entire 28 day planning horizon. All ORs are closed on Saturdays and Sundays, and there are no holidays or other days off in this numerical example; therefore, the 28 day planning horizon has 20 weekdays available. There are 9 operating rooms; thus, the schedule has $(9 \frac{\text{blocks}}{\text{day}}) (20 \text{ days}) = 180$ blocks available. Each block is 8 hours long. Therefore, there are $(9 \text{ ORs})(8 \frac{\text{hours}}{\text{OR}}) = 72$ hours per day available for surgeries that are divided across 6 surgical specialties. Thus, each surgical specialty is assigned an average of $\frac{72}{6} = 12$ hours per surgical specialty per day; this corresponds to an average of $\frac{12}{8} = 1.5$ blocks per surgical specialty per day.

Targets are set for each surgical team. Over the course of the planning horizon, each surgical team should perform surgeries in between L_i and H_i blocks. Values for L_i are generated by sampling integers from the uniform distribution $U(1,5)$, and values for H_i are generated by sampling integers from the uniform distribution $U(6,20)$. Also, each surgical team should operate on N_i patients. Values for N_i are generated by sampling integers from the uniform distribution $U(20,35)$.

For each surgical team, a maximum patient length of stay h_i is generated by sampling integers from the uniform distribution $U(0,5)$. Also, for each surgical team, length of stay probabilities for each length of stay in $0,1,\dots,h_i$ are generated by sampling terms from a normal distribution with certain parameters, summing those values, and then dividing each generated term by the sum. This ensures that the length of stay probabilities for a given surgical team will always sum to one. It should be noted that generating terms from a normal distribution and then dividing by their sum does not necessarily result in terms that are normally distributed. A different array of length of stay probabilities is obtained for each scenario by using different pairs of parameters in the normal distribution.

Van Houdenhoven et al. (2007) provide real-world data from a teaching hospital in the Netherlands for surgical durations D_{ie} in hours and the relative frequencies f_{ie} at which these surgeries are performed. Table 7 shows these values for all six surgical specialties.

Surgical Specialty r	Parameter	Cluster e of surgical team i							
		1	2	3	4	5	6	7	8
General	D_{ie} (hr)	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
	f_{ie} (%)	8	4	12	19	20	3	25	9
Gynecology	D_{ie} (hr)	1.33	0.87	1.22	1.63	2.08	2.60	3.55	N/A
	f_{ie} (%)	2	14	19	25	32	2	6	0
Plastic	D_{ie} (hr)	1.98	1.05	1.37	1.87	2.32	3.12	7.20	N/A
	f_{ie} (%)	5	4	17	21	22	11	10	0
Ear-Nose-Throat	D_{ie} (hr)	1.70	0.67	1.08	1.70	2.12	3.03	4.23	9.15
	f_{ie} (%)	4	33	19	12	14	8	5	6
Orthopedic	D_{ie} (hr)	1.78	1.02	1.38	1.82	2.67	3.32	4.85	N/A
	f_{ie} (%)	9	10	18	21	21	16	5	0
Urology	D_{ie} (hr)	2.02	0.98	1.23	1.70	2.53	3.83	6.42	N/A
	f_{ie} (%)	3	5	30	15	17	21	8	0

Table 7: Surgical Duration and Frequency Values [Adapted from Van Houdenhoven et al. (2007)]

An array of surgical durations is generated for each scenario. For each cluster e of surgical team i , the mean surgical duration across all scenarios is the real-world surgical duration value from Table 7. The f_{ie} values from Van Houdenhoven et al. (2007) are not modified. The values for overhead time factors O_{ie} are generated by sampling real numbers from the uniform distribution $U(1.01, 1.5)$.

Each array of patient lengths of stay is a realization of \tilde{p}_{il} . Similarly, each array of surgical durations is a realization of \tilde{D}_{ie} . It is assumed that each realization is equally likely. A stochastic scenario tree is generated by making combinations of realizations of patient lengths of stay and surgical durations. This results in a two-stage scenario tree with sixteen equally likely scenarios. Any number of scenarios could be used, but sixteen is chosen to limit the amount of computational time required to create the scenario tree.

The objective function for the deterministic problem in each scenario ω has four goals that are multiplied by weights to obtain a weighted objective function. The overall objective function for the stochastic weighted goal program is the expected value of the objective functions from all sixteen scenarios. Different sets of weights can be used for the goals.

However, according to Li et al. (2015), the weighted goal programming model for this kind of problem produces similar schedules regardless of which weights are used. Therefore, all of the goals in the objective function are multiplied by a weight of 1.

The scheduling decisions x_{ijk} are made in the first stage under uncertainty and are scenario-independent. All other variables are determined in the second stage and are scenario-dependent.

Cancellations are made after optimization is complete. For each scheduling decision x_{ijk} , a value c_{ijk} is generated by sampling a real number from the normal distribution $N(50, 2)$. If a c_{ijk} value falls outside two standard deviations from the mean, then the surgery is cancelled and the surgical decision x_{ijk} is changed from 1 to 0. After cancellations are made, all other variables are recalculated to determine the impact of cancellations.

6.2 Mathematical Model

The 64 bit version of the commercial solver Xpress 7.7 is used to implement the stochastic weighted goal programming model from chapter 3. The model is run on a laptop with an Intel Core i5-3210 2.50 GHz Central Processing Unit (CPU) and 12.0 GB of memory. The Xpress code and the data file are included in the appendices. Note that some of the names of parameters, variables, and constraint in the code may not align directly with the names in the mathematical model in chapter 3.

The results from Xpress can be analyzed numerically and graphically to see how well the schedule generated by the model achieves these goals. In this section, numerical results from the stochastic weighted goal programming model are presented. Also, graphs are created to show how the recovery ward is utilized. Optimality gap values are included. Finally, a Master Surgery Schedule generated by the mathematical model is shown.

6.2.1 Numerical Results

Table 8 shows the expected values and standard deviations of expected values for the recovery ward utilization variables. It also shows the percentage of time that the ORs are underutilized before and after cancellations.

The utilization percentage for an OR is the total amount of time used for setting up for surgeries, performing surgeries, and cleaning up after surgeries divided by the total amount of time that the OR both is available and has the necessary staff available to perform surgeries and related tasks (Dexter et al., 1999, p. 7). The OR underutilization (ORUU) percentage is the utilization percentage subtracted from 100%. In this case, the ORUU percentage is calculated as the expected value across all scenarios of $\frac{100 * \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^T d_{ijk}^{\omega-}}{\text{Number of blocks available} * (vT)}$.

Cancellations need to be accounted for. Assume a surgery has a duration of five hours and is scheduled in an eight hour long block. The underutilization and underutilization percentages therefore are three hours and 37.5%. If the block were cancelled, then the ORUU and ORUU percentages would be zero hours and 0% because an unused block cannot underutilize the OR. Therefore, the ORUU percentage actually decreases after cancellations.

	$E_{\omega}(I_{\max}^{\omega})$	$E_{\omega}(I_{\max}^{\omega} - I_{\min}^{\omega})$	$\text{StdDev}(E_{\omega}(I_k^{\omega}))$	$E_{\omega}(\text{ORUU}^{\omega})(\%)$
Before Cancellations	24.86	19.17	6.33	18.82
After Cancellations	24.46	19.59	6.08	17.78

Table 8: Mathematical Model - Underutilization of Operating Rooms and Utilization of Recovery Ward Beds

The results in Table 8 show that the variation in the expected number of patients in the recovery ward is small. This means that the recovery ward is being used efficiently. For all ORs across Canada, the OR utilization percentage during working days is estimated to be 70%; only 12% of Canadian ORs are utilized more than 80% of the time during working days (Canadian Institute for Health Information, 2015a). The results in Table 8 also show that the ORUU percentage is under 20% before and after cancellations; therefore, this is reasonable compared to the utilization of actual ORs in Canada. ORs are expensive to operate; hence, it is crucial to keep the ORUU percentage as low as possible.

6.2.2 Recovery Ward Utilization Graphs

Utilization of the recovery ward beds can be analyzed graphically. However, the graphs need to be generated from the results associated with the mathematical model. The procedure for creating graphs from the Xpress results is as follows. Xpress returns values for all sixteen scenarios. Figure 5 shows the expected number of patients in the recovery ward before cancellations are made for all of the scenarios.

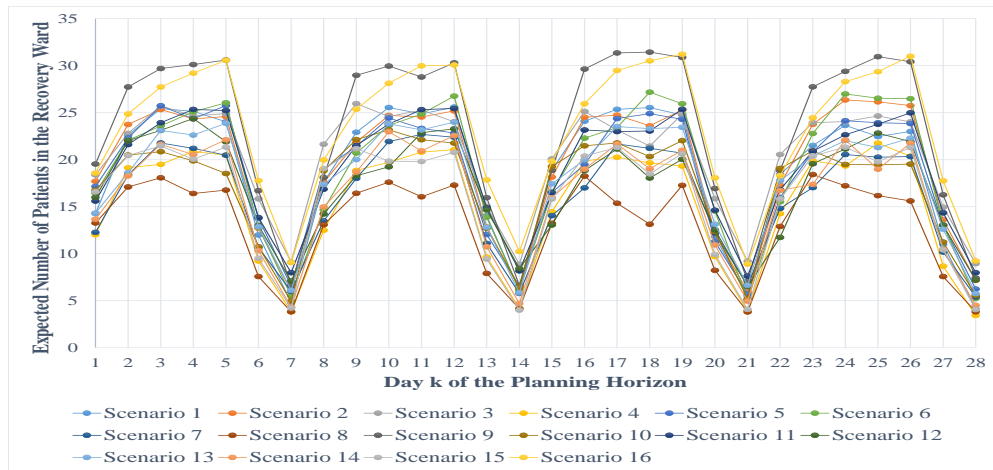


Figure 5: Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before Cancellations, All Scenarios

The overall objective function uses the expected value of all the weighted goals associated with the scenarios. Figure 6 shows the expected value of the number of patients in the recovery ward before cancellations are made.

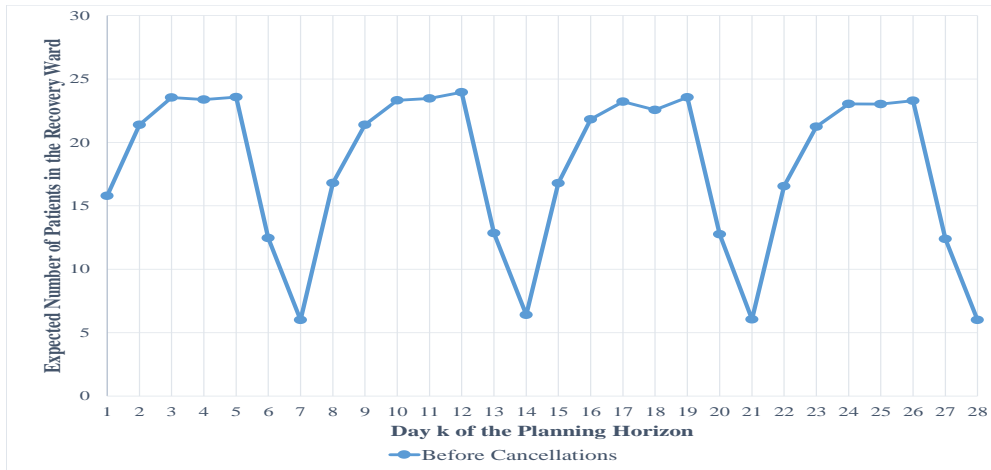


Figure 6: Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before Cancellations

Figure 7 shows the expected number of patients in the recovery ward after cancellations are made.

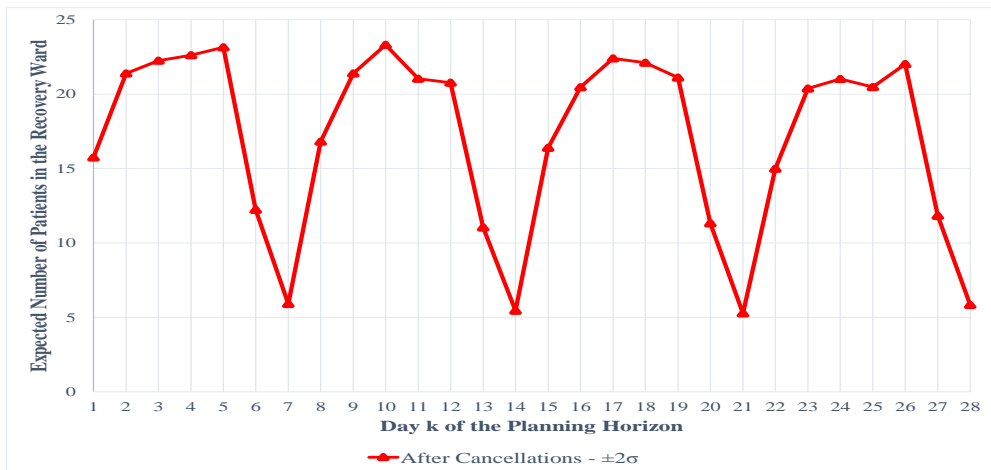


Figure 7: Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - After Cancellations

Figure 8 combines Figure 6 and Figure 7.

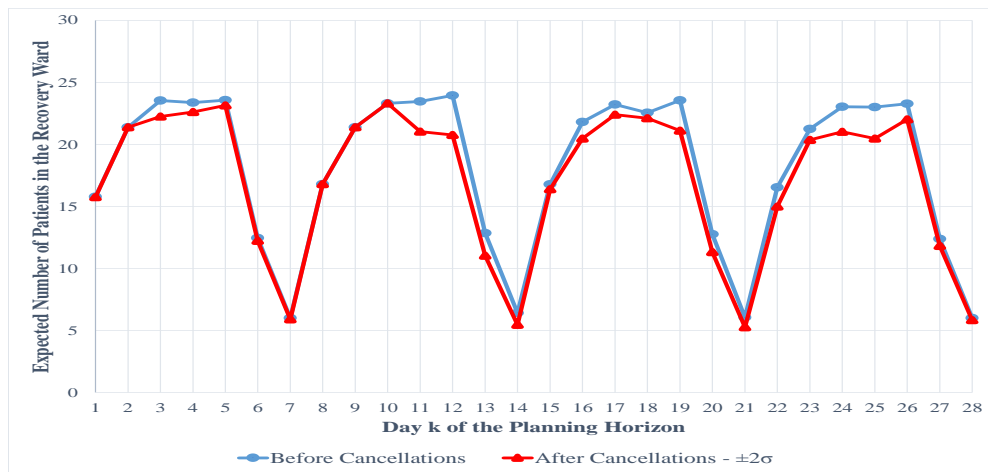


Figure 8: Mathematical Model - Expected Number of Patients in the Recovery Ward Over Planning Horizon - Before and After Cancellations

Figure 8 shows that the expected number of patients in the recovery ward is reasonably flat during weekdays. This means that the recovery ward bed utilization is balanced even though the scheduling decisions were made under uncertainty of both surgical durations and patient lengths of stay. Therefore, the recovery ward is being used efficiently, which is the purpose of the last two goals in the stochastic weighted goal programming model. After cancellations are made, the bed utilization is still reasonably balanced.

6.2.3 Variables and Constraints

The mathematical model has 129960 variables and 123979 constraints.

6.2.4 Optimality Gaps

For mixed integer programs, Xpress solves the linear relaxation of the problem, uses heuristics to find initial integer feasible solutions, and then uses a branch-and-bound search technique to find integer feasible solutions; thus, there are optimality gaps (FICO, 2014, pp. 20-26). A maximum search time of 3600 seconds is used. Table 9 shows the number of integer feasible solutions found, the times at which the first and last integer feasible solutions are found, and the relative optimality gaps.

Initial Relative Optimality Gap (%)	Number of Integer Feasible Solutions Found	Final Solution Time (s)	Final Relative Optimality Gap (%)
0.55	11	714	0.42

Table 9: Mathematical Model - Optimality Gaps

A Master Surgical Schedule (MSS) is generated in about 12 minutes. An MSS can be re-used on a cycle until the strategic decisions change (Tànfani & Testi, 2010, p. 106). Therefore, this amount of time is acceptable.

6.2.5 Example Master Surgical Schedule

Tables 10, 11, 12, and 13 show the MSS generated by the mathematical model; thus, these tables are displaying the x_{ijk} variables that equal 1. Each block in the schedule starts at 0800 and ends at 1600.

		Day						
		1	2	3	4	5	6	7
Operating Room	1	Surgical Team 15	Surgical Team 28	Surgical Team 2	Surgical Team 26	Surgical Team 14	OR Closed	OR Closed
	2	Surgical Team 5	Surgical Team 30	Surgical Team 4	Surgical Team 5	Surgical Team 1	OR Closed	OR Closed
	3	Surgical Team 17	Surgical Team 18	Surgical Team 14	Surgical Team 23	Surgical Team 12	OR Closed	OR Closed
	4	Surgical Team 26	Surgical Team 17	Surgical Team 21	Surgical Team 1	Surgical Team 15	OR Closed	OR Closed
	5	Surgical Team 8	Surgical Team 14	Surgical Team 24	Surgical Team 7	Surgical Team 18	OR Closed	OR Closed
	6	Surgical Team 28	Surgical Team 10	Surgical Team 11	Surgical Team 6	Surgical Team 9	OR Closed	OR Closed
	7	Surgical Team 10	Surgical Team 16	Surgical Team 25	Surgical Team 19	Surgical Team 6	OR Closed	OR Closed
	8	Surgical Team 12	Surgical Team 21	Surgical Team 13	Surgical Team 12	Surgical Team 25	OR Closed	OR Closed
	9	Surgical Team 21	Surgical Team 22	Surgical Team 19	Surgical Team 29	Surgical Team 26	OR Closed	OR Closed

Table 10: Mathematical Model - Example Master Surgical Schedule - Week 1

		Day							
		8	9	10	11	12	13	14	
Operating Room	1	Surgical Team 9	Surgical Team 10	Surgical Team 30	Surgical Team 25	Surgical Team 23	OR Closed	OR Closed	
	2	Surgical Team 5	Surgical Team 30	Surgical Team 4	Surgical Team 5	Surgical Team 3	OR Closed	OR Closed	
	3	Surgical Team 7	Surgical Team 21	Surgical Team 11	Surgical Team 7	Surgical Team 15	OR Closed	OR Closed	
	4	Surgical Team 25	Surgical Team 16	Surgical Team 21	Surgical Team 29	Surgical Team 12	OR Closed	OR Closed	
	5	Surgical Team 17	Surgical Team 17	Surgical Team 23	Surgical Team 1	Surgical Team 18	OR Closed	OR Closed	
	6	Surgical Team 10	Surgical Team 14	Surgical Team 2	Surgical Team 26	Surgical Team 9	OR Closed	OR Closed	
	7	Surgical Team 15	Surgical Team 4	Surgical Team 5	Surgical Team 19	Surgical Team 14	OR Closed	OR Closed	
	8	Surgical Team 8	Surgical Team 19	Surgical Team 25	Surgical Team 12	Surgical Team 26	OR Closed	OR Closed	
	9	Surgical Team 21	Surgical Team 28	Surgical Team 24	Surgical Team 23	Surgical Team 1	OR Closed	OR Closed	

Table 11: Mathematical Model - Example Master Surgical Schedule - Week 2

		Day						
		15	16	17	18	19	20	21
Operating Room	1	Surgical Team 7	Surgical Team 21	Surgical Team 5	Surgical Team 28	Surgical Team 14	OR Closed	OR Closed
	2	Surgical Team 5	Surgical Team 30	Surgical Team 2	Surgical Team 1	Surgical Team 1	OR Closed	OR Closed
	3	Surgical Team 28	Surgical Team 16	Surgical Team 18	Surgical Team 19	Surgical Team 9	OR Closed	OR Closed
	4	Surgical Team 11	Surgical Team 26	Surgical Team 7	Surgical Team 15	Surgical Team 18	OR Closed	OR Closed
	5	Surgical Team 15	Surgical Team 28	Surgical Team 21	Surgical Team 6	Surgical Team 26	OR Closed	OR Closed
	6	Surgical Team 9	Surgical Team 14	Surgical Team 1	Surgical Team 5	Surgical Team 16	OR Closed	OR Closed
	7	Surgical Team 10	Surgical Team 10	Surgical Team 28	Surgical Team 2	Surgical Team 27	OR Closed	OR Closed
	8	Surgical Team 17	Surgical Team 17	Surgical Team 11	Surgical Team 20	Surgical Team 12	OR Closed	OR Closed
	9	Surgical Team 25	Surgical Team 19	Surgical Team 4	Surgical Team 29	Surgical Team 25	OR Closed	OR Closed

Table 12: Mathematical Model - Example Master Surgical Schedule - Week 3

		Day						
		22	23	24	25	26	27	28
Operating Room	1	Surgical Team 17	Surgical Team 17	Surgical Team 21	Surgical Team 5	Surgical Team 14	OR Closed	OR Closed
	2	Surgical Team 5	Surgical Team 30	Surgical Team 30	Surgical Team 1	Surgical Team 1	OR Closed	OR Closed
	3	Surgical Team 10	Surgical Team 26	Surgical Team 2	Surgical Team 3	Surgical Team 26	OR Closed	OR Closed
	4	Surgical Team 11	Surgical Team 21	Surgical Team 6	Surgical Team 26	Surgical Team 9	OR Closed	OR Closed
	5	Surgical Team 25	Surgical Team 12	Surgical Team 25	Surgical Team 20	Surgical Team 6	OR Closed	OR Closed
	6	Surgical Team 15	Surgical Team 10	Surgical Team 11	Surgical Team 7	Surgical Team 12	OR Closed	OR Closed
	7	Surgical Team 7	Surgical Team 4	Surgical Team 17	Surgical Team 19	Surgical Team 18	OR Closed	OR Closed
	8	Surgical Team 9	Surgical Team 19	Surgical Team 28	Surgical Team 12	Surgical Team 15	OR Closed	OR Closed
	9	Surgical Team 28	Surgical Team 28	Surgical Team 24	Surgical Team 23	Surgical Team 25	OR Closed	OR Closed

Table 13: Mathematical Model - Example Master Surgical Schedule - Week 4

6.3 Simulated Annealing

The SA metaheuristic is implemented in MATLAB R2014b and is run on the same laptop.

The MATLAB code and the data files are included in the appendices.

There are four parameters in the metaheuristic for which values need to be chosen: cooling rate (α), initial temperature (T_i), final temperature (T_f), and the number of solutions accepted per inner loop (L). The metaheuristic is first run over a variety of parameter com-

binations to see which ones provide reasonable results in a reasonable amount of time. Design of Experiments is then used to determine whether the values of parameters have a significant impact on the energy values.

6.3.1 Preliminary Numerical Results

Different combinations of parameters can be used in the metaheuristic. Choices for parameter values are completely arbitrary. Each combination is run ten times. Table 14 shows preliminary results. The first four columns are the parameter values. The next three columns are the best, average, and worst solutions obtained for each combination of parameter values; the values of these solutions come from the energy function, which is the same as the objective function in the mathematical model. The energy function is being minimized, so lower values are better. The final column shows the standard deviation of the ten runs for each combination of parameter values.

Solutions From Ten Runs of Each Combination							
Cooling Rate	T_i	T_f	L	Best	Average	Worst	Standard Deviation
0.85	1e5	1	1	762.469	767.647	772.653	3.477
0.85	1e4	1	1	762.919	767.417	774.706	3.466
0.85	2e3	1	1	763.044	766.568	773.687	3.357
0.85	1e3	1	1	762.635	766.754	771.918	3.082
0.90	1e5	1	1	759.876	767.081	770.935	3.650
0.90	1e4	1	1	761.382	766.225	772.804	3.627
0.90	2e3	1	1	761.938	767.077	771.344	3.390
0.90	1e3	1	1	759.267	765.849	775.609	4.263
0.95	1e5	1	1	757.998	761.784	765.989	2.824
0.95	1e4	1	1	758.960	763.847	767.517	3.164
0.95	2e3	1	1	758.927	765.254	771.522	4.292
0.95	1e3	1	1	761.024	767.305	770.889	3.059
0.85	1e5	1	2	760.405	766.184	770.489	3.526
0.85	1e4	1	2	760.172	766.568	769.561	2.933
0.85	2e3	1	3	761.448	766.912	773.436	3.785
0.85	1e3	1	3	759.223	767.037	771.838	3.968
0.85	1e5	1	10	754.656	759.982	764.272	3.111
0.85	1e8	1e-3	1	747.394	749.704	753.372	1.687
0.85	1e5	1e-2	1	754.975	760.493	764.446	2.863

Table 14: Simulated Annealing - Preliminary Numerical Results

Some insights can be gained from these preliminary numerical results. Average results are approximately the same regardless of which parameters are used. The exception to this is when an extremely low final temperature is used. Results for the best solutions behave differently. In general, using higher cooling rates takes longer but usually results in lower best solutions. Similarly, a lower final temperature typically will obtain a lower best result. Finally, using a higher number of solutions per inner loop also obtains lower results for the best values but requires more time.

6.3.2 Design of Experiments

In the designed experiment, each parameter is a factor. Each factor is assigned a high value (1) and a low value (-1). The choice of values is arbitrary but is based on the preliminary numerical results. Values that produce reasonable results in a reasonable amount of time are used. For the cooling rate α , high and low values of 0.95 and 0.85, respectively, are used. Note that the high cooling rate value actually causes slower cooling than the low value. For the initial temperature T_i , high and low values of 100 000 000 and 1 000, respectively, are used. For the final temperature T_f , values of 1 and 0.1, respectively, are used. For the number of solutions per inner loop L , high and low values of 3 and 1, respectively are used. Table 15 summarizes the values of the factors.

Level	Factor			Solutions Per Inner Loop
	Cooling Rate	Initial Temperature	Final Temperature	
High (1)	0.95	100 000 000	1	3
Low (-1)	0.85	1 000	0.1	1

Table 15: Simulated Annealing - Factor Values

A randomized 2^4 full factorial design without replication and without repeats is used for the experiment. This is justifiable because the standard deviation values in Table 14 are low. Table 16 shows the order in which the runs are conducted and the results from the experiment.

Order		Factor				
Standard	Run	Cooling Rate	Initial Temperature	Final Temperature	L	Energy Value
4	1	1	1	-1	-1	752.9992
6	2	1	-1	1	-1	767.797
13	3	-1	-1	1	1	768.3791
9	4	-1	-1	-1	1	753.7186
3	5	-1	1	-1	-1	765.0815
2	6	1	-1	-1	-1	752.8883
16	7	1	1	1	1	758.8771
5	8	-1	-1	1	-1	768.9311
11	9	-1	1	-1	1	754.053
8	10	1	1	1	-1	770.9658
15	11	-1	1	1	1	766.4768
12	12	1	1	-1	1	749.1089
14	13	1	-1	1	1	763.3421
1	14	-1	-1	-1	-1	762.2911
7	15	-1	1	1	-1	766.9889
10	16	1	-1	-1	1	749.443

Table 16: Simulated Annealing - 2^4 Factorial Designed Experiment

Figure 9 is a normal effects plot generated by Minitab 16.0. It shows that the final temperature (Factor C, T_f) has a large positive effect on the energy value (i.e. objective function value). In other words, as the final temperature increases, the energy value also increases. The cooling rate (Factor A, α), the number of solutions per inner loop (Factor D, L), and the interaction between cooling rate, final temperature, and solutions per inner loop (Interaction ACD) have small negative effects on the energy value. If a factor with a negative effect (e.g. final temperature) is increased, then the response (i.e. energy value) decreases. The initial temperature (Factor B, T_i) and all other factor interactions do not have significant effects on the energy value.

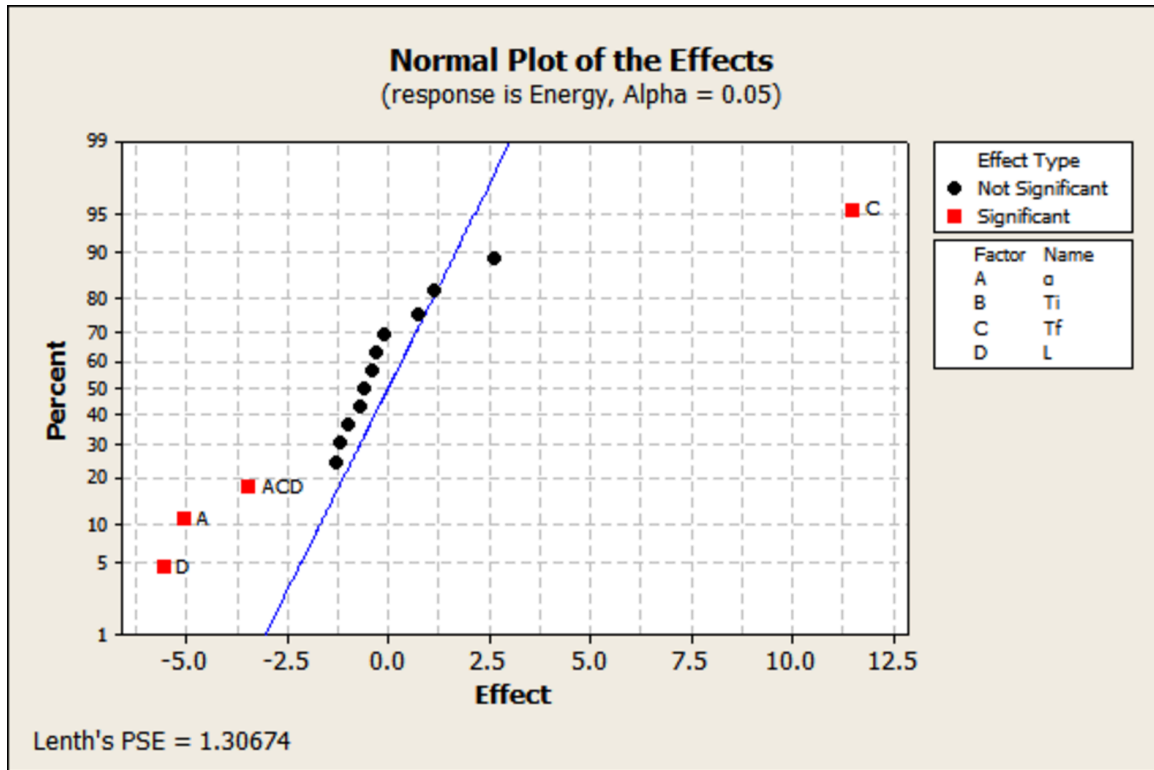


Figure 9: Simulated Annealing - Normal Effects Plot

The results of the preliminary experiment and the designed experiment are used to suggest values for the parameters. The initial temperature does not have a significant effect on the energy value; therefore, T_i should be set to 1 000 in order to reduce the amount of computational time required to find a schedule. α , T_f , and L should be set to 0.95, 0.1, and 3, respectively.

6.4 Discrete Event Simulation

999 replications of a 28 day planning horizon are performed in ProModel. February of 2010 is used because it has 28 days and starts on Monday February 1. However, any 28 day period could be used.

The Master Surgical Schedule (MSS) generated by either the mathematical model or the metaheuristic can be used to schedule surgical teams. In this case, the MSS from the mathematical model (see Tables 10, 11, 12, and 13) is used. Each day at 0600 an entity called PatientBatch for each surgical team scheduled to that day arrives at the Arrival buffer. Each

PatientBatch is assigned attributes to keep track of the associated surgical team, OR, and day of surgery. Also, each PatientBatch is randomly assigned a scenario value by selecting an integer from 1 to 16. In different scenarios, a given surgical team can operate on a different number of patients because the surgical durations are different. Each PatientBatch entity is split into the appropriate number of Patient entities for the assigned scenario. All attributes are then transferred to the Patient entities. This procedure ensures that each surgical team receives an appropriate number of patients in the correct OR on the correct day. All PatientBatch entities exit the system after the Patient entities are generated.

OR resources and surgical team resources are both scheduled on weekdays from 0800 to 1600. All patients arrive and are available for their surgeries before 0800.

All buffers have infinite capacities. The Arrival Area has one unit with an infinite capacity. Each OR, Initial Recovery Area, and Recovery Ward has a capacity of one patient. There are nine ORs, twenty-five Initial Recovery Area units, and sixty Recovery Ward units.

Some of the locations, such as the Arrival Area and the Initial Recovery Area, and the processes that occur at those locations are not relevant to the problem being considered but are added to the simulation to enhance its realism. It is assumed that the Arrival Process follows the normal distribution $N(60, 3)$ minutes and that the Initial Recovery Process follows the normal distribution $N(75, 5)$ minutes.

Process times in the ORs and in the Recovery Ward do matter. The simulation uses the same surgical duration values and length of stay probabilities as the mathematical model and the simulated annealing metaheuristic. The OR process times follow the weighted surgical duration parameter A_i^{ω} . These values are exported from MATLAB and are loaded into the simulation. Similarly, the probability distributions for patient lengths of stay, which are also generated in MATLAB, are loaded into the simulation. The Recovery Ward process time for a patient is randomly selected from the appropriate probability distribution. The

patient then stays in the Recovery Ward for the selected number of days.

Buffers, except for the final one where patients are discharged, do not have any processing times. The processing time for discharges follows the normal distribution $N(30, 1)$ minutes.

Any cancellations are made after the MSS is generated; therefore, cancellations are not included in the simulation.

6.4.1 Simulation Results

For the 28 day planning horizon, the number of patients operated on has a mean of 399.16 patients and a standard deviation of 5.69 patients. The time that a patient spends in the system has a mean of 2210.43 minutes and a standard deviation of 111.56 minutes. The breakdown of the average time is 27.31 minutes in waiting (standard deviation = 0.38 minutes), 2036.33 minutes in operation (standard deviation = 111.72 minutes), and 146.78 minutes in blocking (standard deviation = 2.36 minutes). The blocking happens because all patients arrive at 0600 and then wait for their surgeries after being processed at the Arrival Area. All other blocking is eliminated by the buffers.

Table 17 shows utilization values for the ORs. The ORs are open for 8 hours per day for 20 days in the planning horizon; thus, the scheduled time per OR in the planning horizon is $(8 \frac{\text{hours}}{\text{day}}) (20 \text{ days}) = 160 \text{ hours}$. The work time is the number of minutes that an OR is used for over the course of the planning horizon. The OR underutilization (ORUU) values can be obtained by subtracting each % utilization value from 100%. Note that these values are similar to the ones associated with the mathematical model in Table 8.

Operating Room	Scheduled Time Per OR (hr)	Work Time (min)		Time Per Usage (min)		% Utilization	
	Mean	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
1	160	7459.02	229.39	173.58	4.50	77.70	2.39
2	160	7614.15	354.39	216.35	4.59	79.31	3.69
3	160	7835.56	214.24	161.64	4.16	81.62	2.23
4	160	7932.92	224.05	164.83	4.15	82.63	2.33
5	160	7721.60	224.32	174.54	4.62	80.43	2.24
6	160	7969.55	235.15	156.66	3.72	83.02	2.45
7	160	7740.83	257.22	168.10	4.29	80.63	2.68
8	160	7816.90	211.48	164.43	4.40	81.43	2.20
9	160	7589.75	223.77	183.91	4.72	79.06	2.33

Table 17: Simulation - Utilization of Operating Rooms

Figure 10 is a graph of the expected utilization of recovery ward beds. This graph can be compared to the equivalent graph from the mathematical programming model in Figure 6. Note that there is no warm-up period in the simulation. In other words, there are no spillover patients from January 2010 because the simulation started on February 1, 2010. The recovery ward beds are being used efficiently in the simulation.

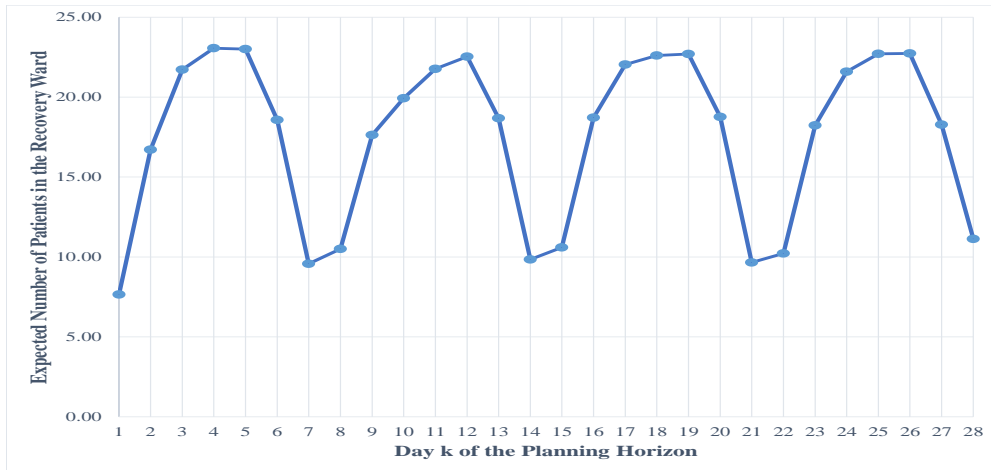


Figure 10: Simulation - Expected Recovery Ward Utilization

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH

In this chapter, conclusions and ideas for future research are presented.

7.1 Conclusions

In this research, the problem of designing a system of operating rooms (ORs) and recovery ward beds by allocating OR blocks to surgical teams in an optimal manner is considered. Viewing a hospital as a system is important because problems in one area of a hospital can affect other areas. A stochastic weighted goal programming model is proposed to schedule elective surgeries under uncertainty of both surgical durations and patient lengths of stay. The model generates a Master Surgical Schedule (MSS) that allocates blocks to surgical teams in a way that optimizes the utilization of both the ORs and the recovery ward beds. Four goals are used in the model: the deviations between the targeted number of surgeries and the actual number of surgeries performed, the deviations between the targeted number of hours for surgeries and the actual number of hours used for surgeries, the maximum expected number of patients in the recovery ward over the course of the planning horizon, and the difference between the maximum and minimum expected numbers of patients in the recovery ward over the course of the planning horizon. Using a stochastic weighted goal programming model to generate an MSS that optimizes both the ORs and the recovery ward beds while accounting for both the uncertainties in surgical durations and patient lengths of stay is an innovation that is present in this research.

Xpress, which is a commercial mathematical programming package, is used to solve the model. The numerical example shows that the implementation of the mathematical model in Xpress generates an MSS with an optimality gap of 0.42%; the branch and bound search runs for 3600 seconds. The expected OR utilization is approximately 80%. The maximum expected number of patients in the recovery ward is minimized, and the expected number of patients in the recovery ward is balanced over the course of the planning horizon. Fur-

thermore, the results from the model are still sensible after cancellations are made. It is concluded that stochastic weighted goal programming is a reasonable way of scheduling surgical teams into ORs while accounting for uncertainties in surgical durations and patient lengths of stay. It should be noted that large scheduling instances likely will have issues with tractability.

The issues with time and memory can be managed by using a metaheuristic. Therefore, a simulated annealing (SA) metaheuristic is developed. In the numerical example, the SA metaheuristic uses less memory than the Xpress implementation of the mathematical model and requires less time; however, the schedules generated by the metaheuristic are near-optimal. Design of Experiments is used to investigate the impact of parameter values in the metaheuristic on the quality of the solutions obtained. It is found that the final temperature parameter has a large positive effect on the energy value, which is the objective function value and thus represents the quality of a schedule. The cooling rate, the inner loop length, and the interaction between the cooling rate, the final temperature, and the inner loop length all have negative effects on the energy value.

A discrete event simulation is developed for the purposes of demonstrating and validating the system of ORs and recovery ward beds. The discrete event simulation model can use an MSS generated by either the mathematical model or the SA metaheuristic. It is shown that the discrete simulation model has similar results as the mathematical model; it should also have similar results as the metaheuristic, but this is not shown in this research. The discrete event simulation model is an important part of demonstrating the value of this research to relevant stakeholders in the health care sector.

The most important conclusion is that both the Xpress implementation of the mathematical model and the SA metaheuristic make it possible for a computer to generate MSS that are near-optimal. Therefore, this research optimizes the system of ORs and recovery ward beds in a hospital; this can reduce costs, make wait lists shorter, and positively affect the

welfares of patients. Furthermore, it can save time for schedulers in the health care sector; however, it must be developed properly in order to be adopted in the health care sector.

7.2 Future Research

There are many ways in which this research could be extended.

The mathematical model, the simulated annealing metaheuristic, and the discrete event simulation model can be incorporated into a Decision Support System (DSS) software package that allows the scheduler to generate MSS without having to understand the underlying mathematics and optimization techniques. Different hospitals may have different needs; thus, the DSS could be customized. The DSS would be based in Microsoft Excel and would contain Visual Basic for Applications (VBA) code to analyze the size of the scheduling problem. Extremely small problems could be solved directly in Excel by using the built in solver package. The overhead would be low. Unfortunately, the Excel solver package can only manage 200 decision variables; ergo, most problems would have to be solved using a commercial solver (Fylstra et al., 1998, p. 32). Some problems may be too large to solve in a commercial solver; hence, these could be solved using the simulated annealing metaheuristic. The Excel file would contain more VBA code to display the schedule to the scheduler. Also, the schedule would be sent to the simulation model to show how the schedule would function in the operating theatre. The simulation model would also generate data regarding the utilization of ORs and recovery ward beds. The schedule in Excel could be interactive. The scheduler would have the option of making swaps in the schedule. The simulation model could then be run again to show the impacts of the swaps. Eventually the scheduler would accept an MSS that would be implemented.

More real world data could be obtained by collaborating with relevant agencies and organizations, such as the Local Health Integration Networks in Ontario, the Community Care Access Centre, and the Institute for Clinical Evaluative Sciences. Also, data from the Ontario Health Insurance Plan and equivalent plans from other provinces could be collected

and used. Furthermore, it would be useful to interact with OR managers in order to consider any special needs or wants that they might have and to ensure that the research is accurately representing real operating theatres. Furthermore, this research should be tested in the real world in using a clinical approach. This could be accomplished by finding a hospital that is willing to implement the approach in this research over a long time period. In the first planning horizon, the hospital could continue to use its existing approach for OR scheduling. The hospital would also use the approach in this research to generate a Master Surgical Schedule (MSS). Performance measures, such as costs, wait list lengths, and measures of patients' welfares, would need to be established in order to allow for comparisons between this approach and the existing approach. Fixler and Wright (2013) propose a scorecard based approach for evaluating the efficiency of ORs. In subsequent planning horizons, the hospital would stop using the existing approach and would exclusively use the approach in this research. At the end of each planning horizon, the performance measures would be used to compare the result of the current planning horizon to results from previous ones. Many factors would need to be controlled for to ensure that changes are due to the scheduling approach.

This research considers the scheduling of surgical teams into ORs, which is a tactical problem. The mathematical model could be extended to make strategic decisions and to schedule surgical cases.

The mathematical model has a number of limitations. In addition, it has assumptions that might not be completely realistic. Different hospitals might be structured differently, and a given hospital can vary significantly on a day to day basis. The model does not consider any of these differences or variances. One assumption is that beds are always available in the recovery ward. In actual hospitals, recovery wards do not have infinite capacities (Beliën et al., 2006, p. 346). A lack of beds would cause cancellations in the proposed operational scheduling model. It is also assumed that all surgical teams within a given surgical specialty

are equally competent. For each surgical team, the relationship between the competence of that surgical team and the length of stays for patients operated on by that surgical team should be derived and incorporated into the mathematical model. Furthermore, it is assumed that all patients operated on by a given surgical team are interchangeable. However, different patients may have certain characteristics that affect surgical durations and/or patient lengths of stay. For example, if the same surgical team performs the same procedure on an 85 year old patient and a 25 year old patient, then the 85 year old patient probably will have a longer length of stay. These variations between patient types should be considered in the mathematical model. Also, some hospitals schedule children before adults; thus, the proposed operational scheduling model should do this (Meskens et al., 2013, p. 651). In addition, emergent surgeries, urgent surgeries, and patient transfers should be considered by the proposed operational case scheduling model. Furthermore, the impact of these on the tactical scheduling should be considered.

REFERENCES

- Abdelrasol, Z. Y., Harraz, N., & Eltawil, A. (2013). A proposed solution framework for the operating room scheduling problems. In *Proceedings of the world congress on engineering and computer science* (Vol. 2, pp. 1149–1157).
- Adan, I., Bekkers, J., Dellaert, N., Jeunet, J., & Vissers, J. (2011). Improving operational effectiveness of tactical master plans for emergency and elective patients under stochastic demand and capacitated resources. *European Journal of Operational Research*, 213(1), 290–308. doi: 10.1016/j.ejor.2011.02.025
- Adan, I., Bekkers, J., Dellaert, N., Vissers, J., & Yu, X. (2009). Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. *Health care management science*, 12(2), 129–141. doi: 10.1007/s10729-008-9080-9
- Adan, I., & Vissers, J. (2002). Patient mix optimisation in hospital admission planning: a case study. *International journal of operations & production management*, 22(4), 445–461. doi: 10.1108/01443570210420430
- Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2012). Long term evaluation of operating theater planning policies. *Operations Research for Health Care*, 1(4), 95–104. doi: 10.1016/j.orhc.2012.10.001
- Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems. *Health care management science*, 17(1), 49–59. doi: 10.1007/s10729-013-9244-0
- Allaoui, H., & Artiba, A. (2004). Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering*, 47(4), 431–450.
- Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., & Testi, A. (2015). A two level meta-heuristic for the operating room scheduling and assignment problem. *Computers & Operations Research*, 54, 21–34. doi: 10.1016/j.cor.2014.08.014
- Aringhieri, R., Landa, P., & Tànfani, E. (2015). Assigning surgery cases to operating rooms: A vns approach for leveling ward beds occupancies. *Electronic Notes in Discrete Mathematics*, 47, 173–180. doi: 10.1016/j.endm.2014.11.023
- Astaraky, D., & Patrick, J. (2015). A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling. *European Journal of Operational Research*, 245(1), 309–319. doi: 10.1016/j.ejor.2015.02.032
- Augusto, V., Xie, X., & Perdomo, V. (2010). Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering*, 58(2), 231–238. doi: 10.1016/j.cie.2009.04.019
- Ballard, S. M., & Kuhl, M. E. (2006). The use of simulation to determine maximum capacity in the surgical suite operating room. In *Proceedings of the 2006 winter simulation conference* (pp. 433–438). doi: 10.1109/WSC.2006.323112
- Banditori, C., Cappanera, P., & Visintin, F. (2013). A combined optimization–simulation approach to the master surgical scheduling problem. *IMA Journal of Management Mathematics*, dps033. doi: 10.1093/imaman/dps033

- Beliën, J., & Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2), 1185–1204. doi: 10.1016/j.ejor.2005.06.063
- Beliën, J., Demeulemeester, E., & Cardoen, B. (2006). Visualizing the demand for various resources as a function of the master surgery schedule: A case study. *Journal of medical systems*, 30(5), 343–350. doi: 10.1007/s10916-006-9012-5
- Beliën, J., Demeulemeester, E., & Cardoen, B. (2009). A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of scheduling*, 12(2), 147–161. doi: 10.1007/s10951-008-0086-4
- Blake, J. T., Dexter, F., & Donald, J. (2002). Operating room managers use of integer programming for assigning block time to surgical groups: a case study. *Anesthesia & Analgesia*, 94(1), 143–148. doi: 10.1213/00000539-200201000-00027
- Blake, J. T., & Donald, J. (2002). Mount sinai hospital uses integer programming to allocate operating room time. *Interfaces*, 32(2), 63–73. doi: 10.1287/inte.32.2.63.57
- Canadian Institute for Health Information. (2015a). *Exploring occupancy through administrative data: A test case using operating rooms*. https://secure.cihi.ca/free_products/or_occupancy_en.pdf.
- Canadian Institute for Health Information. (2015b). *National health expenditure database*.
- Canadian Institute for Health Information. (2015c). *National health expenditure trends, 1975 to 2015*. https://www.cihi.ca/sites/default/files/document/nhex_trends_narrative_report_2015_en.pdf.
- Canadian Institute for Health Information. (2015d). *Patient cost estimator*. https://www.cihi.ca/en/pce_results_en.xlsx.
- Cappanera, P., Visintin, F., & Banditori, C. (2014). Comparing resource balancing criteria in master surgical scheduling: A combined optimisation-simulation approach. *International Journal of Production Economics*, 158, 179–196. doi: 10.1016/j.ijpe.2014.08.002
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3), 921–932. doi: 10.1016/j.ejor.2009.04.011
- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 41–51. doi: 10.1007/BF00940812
- Chaabane, S., Meskens, N., Guinet, A., & Laurent, M. (2008). Comparison of two methods of operating theatre planning: application in belgian hospital. *Journal of Systems Science and Systems Engineering*, 17(2), 171–186. doi: 10.1007/s11518-008-5074-x
- Choi, S., & Wilhelm, W. E. (2014a). An approach to optimize block surgical schedules. *European Journal of Operational Research*, 235(1), 138–148. doi: 10.1016/j.ejor.2013.10.040
- Choi, S., & Wilhelm, W. E. (2014b). On capacity allocation for operating rooms. *Computers & Operations Research*, 44, 174–184. doi: 10.1016/j.cor.2013.11.007
- Christensen, B., & Kockrow, E. (2014). *Adult health nursing*. St. Louis, Missouri: Elsevier Health Sciences.

- Conforti, D., Guerriero, F., & Guido, R. (2010). A multi-objective block scheduling model for the management of surgical operating rooms: new solution approaches via genetic algorithms. In *2010 IEEE Workshop on Health Care Management* (pp. 1–5). doi: 10.1109/WHCM.2010.5441264
- Deber, R. B. (2003). Health care reform: Lessons from Canada. *American Journal of Public Health, 93*(1), 20–24. doi: 10.2105/AJPH.93.1.20
- Dellaert, N., Cayiroglu, E., & Jeunet, J. (2015). Assessing and controlling the impact of hospital capacity planning on the waiting time. *International Journal of Production Research, 1*–12. doi: 10.1080/00207543.2015.1051668
- Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health care management science, 10*(1), 13–24. doi: 10.1007/s10729-006-9005-4
- Dexter, F., Macario, A., Traub, R. D., Hopwood, M., & Lubarsky, D. A. (1999). An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time. *Anesthesia & Analgesia, 89*(1), 7–20. doi: 10.1213/00000539-199907000-00003
- Dexter, F., & O'Neill, L. (2001). Weekend operating room on call staffing requirements. *AORN Journal, 74*(5), 664–671. doi: 10.1016/S0001-2092(06)61765-1
- Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering, 58*(2), 221–230. doi: 10.1016/j.cie.2009.02.012
- Ferrand, Y. B., Magazine, M. J., & Rao, U. S. (2014). Partially flexible operating rooms for elective and emergency surgeries. *Decision Sciences, 45*(5), 819–847. doi: 10.1111/dec.12096
- FICO. (2014). *Xpress-optimizer reference manual release 26.01*.
- Fixler, T., & Wright, J. G. (2013). Identification and use of operating room efficiency indicators: the problem of definition. *Canadian Journal of Surgery, 56*(4), 224–226. doi: 10.1503/cjs.020712
- Fylstra, D., Lasdon, L., Watson, J., & Waren, A. (1998). Design and use of the Microsoft Excel solver. *Interfaces, 28*(5), 29–55. doi: 10.1287/inte.28.5.29
- Gerchak, Y., Gupta, D., & Henig, M. (1996). Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science, 42*(3), 321–334. doi: 10.1287/mnsc.42.3.321
- Gigerenzer, G. (2008). Why heuristics work. *Perspectives on Psychological Science, 3*(1), 20–29. doi: 10.1111/j.1745-6916.2008.00058.x
- Gomez, D., Alali, A. S., Haas, B., Xiong, W., Tien, H., & Nathens, A. B. (2014). Temporal trends and differences in mortality at trauma centres across Ontario from 2005 to 2011: a retrospective cohort study. *Canadian Medical Association Open Access Journal, 2*(3), E176–E182. doi: 10.9778/cmajo.20140007
- Granja, C., Almada-Lobo, B., Janela, F., Seabra, J., & Mendes, A. (2014). An optimization based on simulation approach to the patient admission scheduling problem: Diagnostic imaging department case study. *Journal of Digital Imaging, 27*(1), 33–40. doi: 10.1007/s10278-013-9626-3

- Guerriero, F., & Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health care management science*, *14*(1), 89–114.
- Guinet, A., & Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, *85*(1), 69–81. doi: 10.1016/S0925-5273(03)00087-2
- Hans, E., Wullink, G., Van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, *185*(3), 1038–1050. doi: 10.1016/j.ejor.2006.08.022
- Heng, M., & Wright, J. G. (2013). Dedicated operating room for emergency surgery improves access and efficiency. *Canadian Journal of Surgery*, *56*(3), 167. doi: 10.1503/cjs.019711
- Ivaldi, E., Tanfani, E., & Testi, A. (2003). Simulation supporting the management of surgical waiting lists. *Rapporto tecnico DIEM*, *21*, 2003.
- Jebali, A., Alouane, A. B. H., & Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, *99*(1), 52–62. doi: 10.1016/j.ijpe.2004.12.006
- Jebali, A., & Diabat, A. (2015). A stochastic model for operating room planning under capacity constraints. *International Journal of Production Research*(ahead-of-print), 1–19. doi: 10.1080/00207543.2015.1033500
- Jeroslow, R. G. (1974). Trivial integer programs unsolvable by branch-and-bound. *Mathematical Programming*, *6*(1), 105–109. doi: 10.1007/BF01580225
- Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval research logistics quarterly*, *1*(1), 61–68.
- Kall, P., & Wallace, S. (1994). *Stochastic programming*. Wiley.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.
- Kuo, P. C., Schroeder, R. A., Mahaffey, S., & Bollinger, R. R. (2003). Optimization of operating room allocation using linear programming techniques. *Journal of the American College of Surgeons*, *197*(6), 889–895. doi: 10.1016/j.jamcollsurg.2003.07.006
- Li, X., Rafaliya, N., Baki, M. F., & Chaouch, B. A. (2015). Scheduling elective surgeries: the tradeoff among bed capacity, waiting patients and operating room utilization using goal programming. *Health care management science*, 1–22. doi: 10.1007/s10729-015-9334-2
- Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & industrial engineering*, *37*(1), 57–61.
- Mannino, C., Nilssen, E. J., & Nordlander, T. E. (2012). A pattern based, robust approach to cyclic master surgery scheduling. *Journal of Scheduling*, *15*(5), 553–563. doi: 10.1007/s10951-012-0275-z
- Marques, I., Captivo, M. E., & Pato, M. V. (2012). An integer programming approach to elective surgery scheduling. *OR spectrum*, *34*(2), 407–427. doi: 10.1007/s00291-011-0279-7
- May, J. H., Spangler, W. E., Strum, D. P., & Vargas, L. G. (2011). The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, *20*(3), 392–405. doi: 10.1111/j.1937-5956.2011.01221.x

- Meskens, N., Duvivier, D., & Hanset, A. (2013). Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems*, 55(2), 650–659. doi: 10.1016/j.dss.2012.10.019
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), 1087–1092. doi: 10.1063/1.1699114
- Nahar, S., Sahni, S., & Shragowitz, E. (1986). Simulated annealing and combinatorial optimization. In *Proceedings of the 23rd acm/ieee design automation conference* (pp. 293–299).
- Olivares, M., Terwiesch, C., & Cassorla, L. (2008). Structural estimation of the newsvendor model: an application to reserving operating room time. *Management Science*, 54(1), 41–55. doi: 10.1287/mnsc.1070.0756
- Ontario Ministry of Health and Long-Term Care. (2010). *Reference guide and toolkit for improvements in perioperative practice in ontario*. <http://www.health.gov.on.ca/en/pro/programs/ecfa/docs/guide.pdf>.
- O. reg. 321/01. (2001). <http://www.ontario.ca/laws/regulation/90096>.
- Parizi, M. S., & Ghate, A. (2015). *Multi-class, multi-resource advance scheduling with no-shows, cancellations and overbooking*. <http://ssrn.com/abstract=2550560>. doi: 10.2139/ssrn.2550560
- Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185(3), 1011–1025. doi: 10.1016/j.ejor.2006.03.059
- Porteus, E. (2002). *Foundations of stochastic inventory theory*. Stanford University Press.
- Protti, D. (2015). Missed connections: The adoption of information technology in canadian healthcare. *CD Howe Institute Commentary*, 422. doi: 10.2139/ssrn.2587406
- Redelmeier, D. A., Thiruchelvam, D., & Daneman, N. (2008). Introducing a methodology for estimating duration of surgery in health services research. *Journal of clinical epidemiology*, 61(9), 882–889. doi: 10.1016/j.jclinepi.2007.10.015
- Riise, A., & Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics*, 17(4), 389–414. doi: 10.1007/s10732-010-9139-x
- Robinson, V., Goel, V., MacDonald, R. D., & Manuel, D. (2009). Inter-facility patient transfers in ontario: do you know what your local ambulance is being used for? *Healthcare Policy*, 4(3), 53.
- Roland, B., Di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58(2), 212–220. doi: 10.1016/j.cie.2009.01.005
- Santibáñez, P., Begen, M., & Atkins, D. (2007). Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a british columbia health authority. *Health care management science*, 10(3), 269–282. doi: 10.1007/s10729-007-9019-6
- Shapiro, A., Dentcheva, D., & Ruszczykii, A. (2009). *Lectures on stochastic programming: Modeling and theory*. Philadelphia: Society for Industrial and Applied Mathematics.
- Sier, D., Tobin, P., & McGurk, C. (1997). Scheduling surgical procedures. *Journal of the Operational Research Society*, 48(9), 884–891. doi: 10.1057/palgrave.jors.2600441

- Silva, T. A., de Souza, M. C., Saldanha, R. R., & Burke, E. K. (2015). Surgical scheduling with simultaneous employment of specialised human resources. *European Journal of Operational Research*, 245(3), 719–730. doi: 10.1016/j.ejor.2015.04.008
- Snyder, J., Crooks, V. A., Johnston, R., & Kingsbury, P. (2011). What do we know about canadian involvement in medical tourism? a scoping review. *Open Medicine*, 5(3), e139-e148.
- Statistics Canada. (2014). *Annual demographic estimates: Canada, provinces and territories*. <http://www.statcan.gc.ca/pub/91-215-x/91-215-x2014000-eng.pdf>.
- Sule, D. R. D. R. (2008). *Production planning and industrial scheduling : examples, case studies, and applications*. Boca Raton: CRC Press.
- Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. Wiley Publishing.
- Tamiz, M., Jones, D., & Romero, C. (1998). Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of operational research*, 111(3), 569–581. doi: 10.1016/S0377-2217(97)00317-2
- Tancrez, J.-S., Roland, B., Cordier, J.-P., & Riane, F. (2009). How stochasticity and emergencies disrupt the surgical schedule. In *Intelligent patient management* (pp. 221–239). Springer. doi: 10.1007/978-3-642-00179-6_14
- Tancrez, J.-S., Roland, B., Cordier, J.-P., & Riane, F. (2013). Assessing the impact of stochasticity for operating theater sizing. *Decision Support Systems*, 55(2), 616–628. doi: 10.1016/j.dss.2012.10.021
- Tanfani, E., & Testi, A. (2010). A pre-assignment heuristic algorithm for the master surgical schedule problem (mssp). *Annals of Operations Research*, 178(1), 105–119. doi: 10.1007/s10479-009-0568-6
- Testi, A., & Tanfani, E. (2009). Tactical and operational decisions for operating room planning: Efficiency and welfare implications. *Health Care Management Science*, 12(4), 363–373. doi: 10.1007/s10729-008-9093-4
- Testi, A., Tanfani, E., & Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10(2), 163–172. doi: 10.1007/s10729-007-9011-1
- Vanberkel, P. T., Boucherie, R. J., Hans, E. W., Hurink, J. L., van Lent, W. A., & van Harten, W. H. (2011a). Accounting for inpatient wards when developing master surgical schedules. *Anesthesia & Analgesia*, 112(6), 1472–1479. doi: 10.1213/ANE.0b013e3182159c2f
- Vanberkel, P. T., Boucherie, R. J., Hans, E. W., Hurink, J. L., van Lent, W. A., & van Harten, W. H. (2011b). An exact approach for relating recovering surgical patient workload to the master surgical schedule. *Journal of the Operational Research Society*, 62(10), 1851–1860. doi: 10.1057/jors.2010.141
- Van Houdenhoven, M., van Oostrum, J. M., Hans, E. W., Wullink, G., & Kazemier, G. (2007). Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling. *Anesthesia & Analgesia*, 105(3), 707–714. doi: 10.1213/01.ane.0000277492.90805.0f
- van Oostrum, J. M., Van Houdenhoven, M., Hurink, J. L., Hans, E. W., Wullink, G., & Kazemier, G. (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR spectrum*, 30(2), 355–374. doi: 10.1007/s00291

-006-0068-x

- Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3), 583–591. doi: 10.1016/j.ejor.2012.09.010
- Vissers, J., Adan, I. J., & Bekkers, J. A. (2005). Patient mix optimization in tactical cardiothoracic surgery planning: a case study. *IMA Journal of Management Mathematics*, 16(3), 281–304. doi: 10.1093/imaman/dpi023
- Wullink, G., Van Houdenhoven, M., Hans, E. W., van Oostrum, J. M., van der Lans, M., & Kazemier, G. (2007). Closing emergency operating rooms improves efficiency. *Journal of Medical Systems*, 31(6), 543–546. doi: 10.1007/s10916-007-9096-6
- Xiang, W., Yin, J., & Lim, G. (2013). Modified ant colony algorithm for surgery scheduling under multiresource constraints. *Advances in Information Sciences and Service Sciences*, 5(9), 810.
- Xiang, W., Yin, J., & Lim, G. (2015). A short-term operating room surgery scheduling problem integrating multiple nurses roster constraints. *Artificial intelligence in medicine*, 63(2), 91–106. doi: 10.1016/j.artmed.2014.12.005
- Yahia, Z., Harraz, N., & Eltawil, A. B. (2014). Building master surgery schedules with leveled bed occupancy and nurse workloads. In *2014 IEEE conference on industrial engineering and engineering management* (pp. 89–93). doi: 10.1109/IEEM.2014.7058606
- Zhang, B., Murali, P., Dessouky, M., & Belson, D. (2009). A mixed integer programming approach for allocating operating room capacity. *Journal of the Operational Research Society*, 60(5), 663–673. doi: 10.1057/palgrave.jors.2602596
- Zonderland, M. E., Boucherie, R. J., Litvak, N., & Vleggeert-Lankamp, C. L. (2010). Planning and scheduling of semi-urgent surgeries. *Health Care Management Science*, 13(3), 256–267. doi: 10.1007/s10729-010-9127-6

APPENDIX 1 - MATHEMATICAL MODEL - XPRESS CODE (SWGPM.MOS)

```

1 model "Weighted Goal Programming Model"
2
3 !Load modules
4 uses "mmsp", "random"
5
6 !Location of data file
7 parameters
8   DataFile = 'Data/DataFile.dat'
9 end-parameters
10
11 declarations
12 !Declare a random seed value
13 randSeed: integer
14
15 !Declare index sets
16 i, j, k, l, e, r: range
17
18 !Declare the upper bounds of the index sets
19 m, n, T, h, E, R: integer
20
21 !Declare other parameters
22 v: real
23
24 L, H, N: array(i) of integer
25
26 B, B1, F2: array(r) of integer
27
28 D1, D2, D3, D4, D5, D6, D7, D8: array(i,e) of real
29 D9, D10, D11, D12, D13, D14, D15, D16: array(i,e) of real
30
31 p1, p2, p3, p4, p5, p6, p7, p8: array(i,l) of real
32 p9, p10, p11, p12, p13, p14, p15, p16: array(i,l) of real
33
34 O, f: array(i,e) of real
35
36 pi: array(i,j) of integer
37
38 OMEGA: array(i,k) of integer
39
40 !Declare 0-1 decision variables
41 x: array(i,j,k) of spvar
42

```

```

43 !Declare continuous decision variables
44 I: array(k) of spvar
45
46 I_max, I_min: spvar
47
48 !Declare deviation variables
49 dminus, dplus: array(i) of spvar
50
51 d2n: array(i,j,k) of spvar
52
53 !Define number of scenarios
54 numScenarios: integer
55
56 !Declare stochastic parameters
57 D: array(i,e) of sprand
58
59 p: array(i,l) of sprand
60
61 !Declare random expressions
62 maxni: array(i) of sprandexp
63
64 A: array(i) of sprandexp
65
66 PHI: array(i,l) of sprandexp
67
68 !Declare stages in scenario tree
69 Stages = "First","Second"
70 end-declarations
71
72 initializations from DataFile
73 randSeed numScenarios m n T h E R
74 end-initializations
75
76 !Set the random seed
77 setmtrandseed(randSeed)
78
79 !Define all index sets
80 i := 1..m
81 j := 1..n
82 k := 1..T
83 l := 0..28 !Note that l starts at 0
84 e := 1..E
85 r := 1..R
86
87 !Finalize the index sets

```

```

88 finalize(i)
89 finalize(j)
90 finalize(k)
91 finalize(l)
92 finalize(e)
93 finalize(r)
94
95 !Forward declarations of procedures
96 forward procedure assembleData
97 forward procedure calculateCancellations
98 forward procedure createBinaryDecisionVariables
99 forward procedure createContinuousVariables
100 forward procedure createDeviationVariables
101 forward procedure displayResults
102 forward procedure generateScenarioTree
103 forward procedure weightedGoalProgramming
104
105
106 !Call the procedures to create the decision variables
107 createBinaryDecisionVariables
108 createContinuousVariables
109 createDeviationVariables
110
111 !Read values from the data file
112 initializations from DataFile
113 v pi OMEGA
114 end-initializations
115
116
117 (!*****
118
119 Named Constraints, Parameters, and Calculations
120
121 *****!)
122 declarations
123 constraint1: array(k) of splinctr
124 constraint2: array(j) of splinctr
125 constraint3: array(j,k) of splinctr
126 constraint4: array(i,k) of splinctr
127 constraint5: array(i) of splinctr
128 constraint6: array(i) of splinctr
129 constraint7: array(r) of splinctr
130 constraint8: array(i) of splinctr
131 constraint9: array(i,j,k) of splinctr
132 constraint10: array(k) of splinctr

```

```

133     constraint11: array(k) of splinctr
134     constraint12: array(k) of splinctr
135 end-declarations
136
137 assembleData
138
139 !Generate the scenario tree
140 generateScenarioTree
141
142 forall(icount in i) A(icount) := sum(ecount in e)
143 (D(icount,ecount)*O(icount,ecount)*f(icount,ecount))
144
145 forall(icount in i) maxni(icount) := floor(v/A(icount))
146
147 forall(icount in i, lcount in l) PHI(icount,lcount) :=
148 sum(tau in 0..floor(h/T), lcount2 in lcount+tau*T+1..h)maxni(icount)*p(icount,lcount2)
149
150 !Constraints
151 forall(kcount in k) constraint1(kcount) :=
152 sum(icount in i, jcount in j) pi(icount, jcount) * x(icount, jcount, kcount) = 0
153
154 forall(jcount in j) constraint2(jcount) :=
155 sum(icount in i, kcount in k) OMEGA(icount, kcount) * x(icount, jcount, kcount) = 0
156
157 forall(jcount in j, kcount in k) constraint3(jcount, kcount) :=
158 sum(icount in i) x(icount, jcount, kcount) <= 1
159
160 forall(icount in i, kcount in k) constraint4(icount, kcount) :=
161 sum(jcount in j) x(icount, jcount, kcount) <= 1
162
163 forall(icount in i) do
164     constraint5(icount) := sum(jcount in j, kcount in k) x(icount, jcount, kcount) >= L(icount)
165     constraint6(icount) := sum(jcount in j, kcount in k) x(icount, jcount, kcount) <= H(icount)
166 end-do
167
168 forall(rcount in r) constraint7(rcount) :=
169 sum(icount in B1(rcount) .. F2(rcount), jcount in j, kcount in k)
170 x(icount, jcount, kcount) = B(rcount)
171
172 forall(icount in i) constraint8(icount) :=
173 maxni(icount)*(sum(jcount in j, kcount in k)
174 (x(icount, jcount, kcount))) + d1minus(icount) - d1plus(icount) = N(icount)
175
176 forall(icount in i, jcount in j, kcount in k) constraint9(icount, jcount, kcount) :=
177 (A(icount)*maxni(icount) - v)*(x(icount, jcount, kcount)) + d2n(icount, jcount, kcount) = 0

```

```

178
179 forall(kcount in k) constraint10(kcount) := I(kcount) =
180 sum(icont in i, jcount in j, lcount2 in l..kcount) PHI(icont, kcount-lcount2)*x(icont, jcount, lcount2) +
181 sum(icont in i, jcount in j, lcount2 in kcount..I) PHI(icont, kcount+I-lcount2)*x(icont, jcount, lcount2)
182
183 forall(kcount in k) constraint11(kcount) := I(kcount) <= I_max
184
185 forall(kcount in k) constraint12(kcount) := I(kcount) >= I_min
186
187 (!*****
188
189 Main part of program
190 *****
191 weightedGoalProgramming
192
193 (!*****
194
195 Weighted goal programming procedure
196
197 *****
198 procedure weightedGoalProgramming
199
200     declarations
201     objective: splinctr
202     end-declarations
203
204     !Objective function
205     objective := sum(icont in i)(dIminus(icont)) +
206     sum(icont in i, jcount in j, kcount in k) (d2n(icont, jcount, kcount)) +
207     I_max +
208     (I_max - I_min)
209
210     setparam("XPRS_MAXTIME", -3600)
211     minimize(objective)
212     displayResults
213 end-procedure
214
215 (!*****
216
217 Procedure for displaying the results
218
219 *****
220 procedure displayResults
221
222     writeln("*****Objective value before cancellations *****")

```

```

223 writeIn(getparam("xprs_mipobjval"))
224
225 writeIn("*****xijk values before cancellations *****")
226 forall(icontains in i, jcount in j, kcount in k | getsol(x(icontains,jcount,kcount)) = 1)
227 writeIn("x(",icontains, ", ", jcount, ", ", kcount, ") = ", getsol(x(icontains,jcount,kcount)))
228
229 forall(s in 1..numScenarios) do
230   writeIn("***** Sum of dminus(i) values before cancellations, Scenario", s, "*****")
231   writeIn(sum(icontains in i)(getsolinscen(dminus(icontains),s)))
232 end-do
233
234 forall(s in 1..numScenarios) do
235   writeIn("***** Sum of d2n(i,j,k) values before cancellations, Scenario", s, "*****")
236   writeIn("Underutilized time", (sum(icontains in i, jcount in j, kcount in k)
237     getsolinscen(d2n(icontains,jcount,kcount),s)))
238 end-do
239
240 forall(s in 1..numScenarios) do
241   writeIn("***** Percent underutilization values before cancellations, Scenario", s, "*****")
242   writeIn("Percent unutilized OR time", 100*(sum(icontains in i, jcount in j, kcount in k)
243     getsolinscen(d2n(icontains,jcount,kcount),s))/(v*n*(T-8)))
244 end-do
245
246 forall(s in 1..numScenarios) do
247   writeIn("***** I(k) values before cancellations, Scenario", s, "*****")
248   forall(kcount in k) writeIn(getsolinscen(I(kcount),s))
249 end-do
250
251 calculateCancellations
252
253 forall(s in 1..numScenarios) do
254   writeIn("***** Sum of d2n(i,j,k) values after cancellations, Scenario ", s, "*****")
255   writeIn("Underutilized time",
256     (sum(icontains in i, jcount in j, kcount in k | speval(x(icontains,jcount,kcount),s) = 1)
257     (v - speval(A(icontains),s)*speval(maxni(icontains),s))))
258 end-do
259
260 forall(s in 1..numScenarios) do
261   writeIn("***** Sum of d2n(i,j,k) values after cancellations, Scenario", s, "*****")
262   writeIn("Percent unutilized OR time",
263     100*(sum(icontains in i, jcount in j, kcount in k | speval(x(icontains,jcount,kcount),s) = 1)
264     (v - speval(A(icontains),s)*speval(maxni(icontains),s))/(v*n*(T-8))))
265 end-do
266
267 forall(s in 1..numScenarios) do

```

```

268 writeIn("***** I(k) values after Cancellations, Scenario", s, "*****")
269 forall(kcount in k) writeIn(speval(
270   sum(icontains in i, jcount in j, lcount2 in 1..kcount)
271   PHI(icontains-lcount2)*x(icontains,jcount,lcount2) +
272   sum(icontains in i, jcount in j, lcount2 in kcount+1..T)
273   PHI(icontains+lcount2)*x(icontains,jcount,lcount2),s))
274   end-do
275 end-procedure
276
277 (!*****
278
279 Variable creation procedures
280
281 *****
282 !Create the binary decision variables
283 procedure createBinaryDecisionVariables
284   forall(icontains in i, jcount in j, kcount in k) do
285     create(x(icontains,jcount,kcount))
286     x(icontains,jcount,kcount) is_binary
287   end-do
288 end-procedure
289
290 !Create the continuous decision variables
291 procedure createContinuousVariables
292   forall(kcount in k) create(I(kcount))
293 end-procedure
294
295 !Create the deviation variables
296 procedure createDeviationVariables
297   forall(icontains in i) do
298     create(dlplus(icontains))
299     create(dlminus(icontains))
300   end-do
301
302   forall(icontains in i, jcount in j, kcount in k) create(d2n(icontains,jcount,kcount))
303 end-procedure
304
305 (!*****
306
307 Scenario Tree Generation
308
309 *****
310 procedure generateScenarioTree
311   declarations
312     Probabilities : array (1..numScenarios) of real

```



```

313 end-declarations
314
315 Probabilities::[1/numScenarios,1/numScenarios,1/numScenarios, 1/numScenarios,
316 1/numScenarios,1/numScenarios,1/numScenarios, 1/numScenarios,1/numScenarios,
317 1/numScenarios,1/numScenarios,1/numScenarios,1/numScenarios,1/numScenarios,
318 1/numScenarios,1/numScenarios]
319
320 spsetstages (Stages)
321
322 forall(kcount in k) spsetstage(constraint1(kcount), "First")
323 forall(jcount in j) spsetstage(constraint2(jcount), "First")
324 forall(jcount in j, kcount in k) spsetstage(constraint3(jcount, kcount), "First")
325 forall(icont in i, lcount in l) spsetstage(constraint4(icont,kcount), "First")
326 forall(icont in i) spsetstage(constraint5(icont), "First")
327 forall(icont in i) spsetstage(constraint6(icont), "First")
328 forall(rcount in r) spsetstage(constraint7(rcount), "First")
329 forall(icont in i) spsetstage(constraint8(icont), "Second")
330 forall(icont in i, jcount in j, kcount in k) spsetstage(constraint9(icont, jcount, kcount), "Second")
331 forall(kcount in k) spsetstage(constraint10(kcount), "Second")
332 forall(kcount in k) spsetstage(constraint11(kcount), "Second")
333 forall(kcount in k) spsetstage(constraint12(kcount), "Second")
334
335 forall(icont in i, jcount in j, kcount in k) spsetstage(x(icont, jcount, kcount), "First")
336 forall(icont in i, ecount in e) spsetstage(D(icont, ecount), "Second")
337 forall(icont in i, lcount in l) spsetstage(p(icont, lcount), "Second")
338 forall(kcount in k) spsetstage(I(kcount), "Second")
339 spsetstage(I_max, "Second")
340 spsetstage(I_min, "Second")
341 forall(icont in i) spsetstage(dlminus(icont), "Second")
342 forall(icont in i) spsetstage(dlplus(icont), "Second")
343 forall(icont in i, jcount in j, kcount in k) spsetstage(d2n(icont, jcount, kcount), "Second")
344
345 !Create the tree
346 spcreatetree(numScenarios)
347
348 !Assign realizations of p(i,l) and D(i,e)
349
350 forall(icont in i, ecount in e) spsetrandatnode(D(icont, ecount), 1, D1(icont, ecount))
351 forall(icont in i, lcount in l) spsetrandatnode(p(icont, lcount), 1, p1(icont, lcount))
352
353 forall(icont in i, ecount in e) spsetrandatnode(D(icont, ecount), 2, D2(icont, ecount))
354 forall(icont in i, lcount in l) spsetrandatnode(p(icont, lcount), 2, p2(icont, lcount))
355
356 forall(icont in i, ecount in e) spsetrandatnode(D(icont, ecount), 3, D3(icont, ecount))
357 forall(icont in i, lcount in l) spsetrandatnode(p(icont, lcount), 3, p3(icont, lcount))

```

```

358 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 4, D4(icount,ecourt))
359 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 4, p4(icount,lcourt))
360
361
362 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 5, D5(icount,ecourt))
363 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 5, p5(icount,lcourt))
364
365 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 6, D6(icount,ecourt))
366 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 6, p6(icount,lcourt))
367
368 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 7, D7(icount,ecourt))
369 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 7, p7(icount,lcourt))
370
371 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 8, D8(icount,ecourt))
372 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 8, p8(icount,lcourt))
373
374 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 9, D9(icount,ecourt))
375 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 9, p9(icount,lcourt))
376
377 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 10, D10(icount,ecourt))
378 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 10, p10(icount,lcourt))
379
380 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 11, D11(icount,ecourt))
381 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 11, p11(icount,lcourt))
382
383 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 12, D12(icount,ecourt))
384 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 12, p12(icount,lcourt))
385
386 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 13, D13(icount,ecourt))
387 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 13, p13(icount,lcourt))
388
389 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 14, D14(icount,ecourt))
390 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 14, p14(icount,lcourt))
391
392 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 15, D15(icount,ecourt))
393 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 15, p15(icount,lcourt))
394
395 forall(icount in i, ecourt in e) spsetrandatnode(D(icount,ecourt), 16, D16(icount,ecourt))
396 forall(icount in i, lcourt in l) spsetrandatnode(p(icount,lcourt), 16, p16(icount,lcourt))
397
398 !Assign probabilities to branches
399 spsetprobcond(2,Probabilities)
400
401 !Generate the tree
402 spgentree

```

```

403 end-procedure
404
405 (!*****
406
407 Assemble Data
408
409 *****!)
410 procedure assembleData
411     initializations from DataFile
412     L H N O
413     D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16
414     p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16
415     f as "f1"
416     B1 as "B11"
417     F2 as "F21"
418     B as "B1"
419     end-initializations
420
421 end-procedure
422 (!*****
423
424 Cancellations
425
426 *****!)
427 procedure calculateCancellations
428
429     declarations
430     c: array(i,j,k) of real
431     mean: real
432     stdev: real
433     end-declarations
434
435     initializations from DataFile
436     mean stdev
437     end-initializations
438
439     !Generate a value of cij for all xijk by selecting from a normal distribution.
440     forall(icount in i, jcount in j, kcount in k) c(icount,jcount,kcount) := normal(mean,stdev)
441
442     !If the value of cij falls outside of the bounds, then xijk will be fixed to 0.
443     forall(icount in i, jcount in j, kcount in k) c(icount,jcount,kcount) <= (mean - 2*stdev) or
444     c(icount,jcount,kcount) >= (mean+2*stdev) spfix(x(icount,jcount,kcount),0)
445
446     !Display cancellations
447     writeln("***** Cancellations *****")

```

```
448     forall(icontains in i, jcount in j, kcount in k | c(icontains, jcount, kcount) <=
449         (mean - 2*stdev) or c(icontains, jcount, kcount) >= (mean+2*stdev))
450         writeIn("x(", icount, ", ", jcount, ", ", kcount, "): ", c(icontains, jcount, kcount), " = 0 ")
451     end-procedure
452
453 end-model
```

APPENDIX 2 - MATHEMATICAL MODEL - XPRESS DATA FILE (~DATA/DATAFILE.DAT)

```

1 !Random seed value
2 randSeed: 5
3
4 !Number of scenarios
5 numScenarios: 16
6
7 !Upper bounds for index sets
8 m: 30
9 n: 9
10 T: 28
11 h: 28
12 v: 8
13 R: 6
14 E: 8
15
16 N:[20 21 33 23 24 30 28 26 26 34 25 35 25 21 30 22 27 21 26 31 32 31 32 22 29 29 24 30 30 35]
17 L:[1 5 1 5 2 5 3 2 4 3 5 1 1 2 4 3 2 2 3 1 5 3 2 5 1 4 3 4]
18 H:[9 14 12 16 10 8 17 18 10 18 7 9 12 16 16 18 13 12 20 8 13 20 11 7 10 10 17 10 11 6]
19
20 B11:[1 7 12 15 17 23]
21 F21:[6 11 14 16 22 30]
22 B1:[36 30 18 12 36 48]
23
24 pi:[0 0 0 0 0 0 0 0 0
25 0 0 0 0 0 0 0 0 0
26 0 0 0 0 0 0 0 0 0
27 0 0 0 0 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0
29 0 1 0 0 0 0 0 0 1
30 0 1 0 0 0 0 0 0 1
31 0 1 0 0 0 0 0 0 1
32 0 1 0 0 0 0 0 0 1
33 0 1 0 0 0 0 0 0 1
34 0 1 0 0 0 0 0 0 1
35 0 1 0 0 0 0 0 0 1
36 0 1 0 0 0 0 0 0 1
37 0 1 0 0 0 0 0 0 1
38 0 1 0 0 0 0 0 0 1
39 0 1 0 0 0 0 0 0 1
40 0 1 0 0 0 0 0 0 1
41 0 1 0 0 0 0 0 0 0
42 0 1 0 0 0 0 0 0 0

```

```

43 0 1 0 0 0 0 0 0 0
44 0 1 0 0 0 0 0 0 0
45 0 1 0 0 0 0 0 0 0
46 0 1 0 0 0 0 0 0 0
47 0 1 0 0 0 0 0 0 0
48 0 1 0 0 0 0 0 0 0
49 0 1 0 0 0 0 0 0 0
50 0 1 0 0 0 0 0 0 0
51 0 1 0 0 0 0 0 0 0
52 1 1 1 0 1 1 0 0 0
53 0 0 0 1 0 0 1 1 1
54

```

```

55 omega: [1 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1
56 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 1
57 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1
58 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1
59 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
60 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
61 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1
62 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1
63 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1
64 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
65 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1
66 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1
67 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1
68 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
69 0 1 0 0 0 1 1 0 1 0 0 0 1 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1
70 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 0 1 1
71 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1
72 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
73 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1
74 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1
75 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
76 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 1
77 0 1 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1
78 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
79 0 1 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1
80 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 1
81 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
82 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1
83 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1
84 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1]

```

```

85
86 D1: [2.3173 0.7446 2.2956 2.8759 3.8257 3.2070 3.7625 8.0210
87 3.4796 0.8833 2.2587 2.2586 2.6717 4.0321 4.7326 5.9723

```

88 1.7038 1.3210 1.1107 1.8261 1.7691 2.5791 5.4525 3.8960
89 4.1565 1.3954 1.6853 2.7277 2.3129 2.5092 5.8354 6.2566
90 2.4947 0.6581 1.0788 1.4872 1.7912 3.9252 4.4539 4.3219
91 2.5375 1.1567 1.6341 2.1475 2.6922 2.9506 4.7939 3.3971
92 1.5174 0.4754 1.3031 1.4026 2.5217 3.6492 4.4615 0
93 1.2310 1.1292 0.6299 2.2318 1.4573 2.4759 3.9117 0
94 0.8574 1.0559 0.8278 2.0638 1.7825 2.5556 3.7492 0
95 0.8463 0.9741 1.1390 0.9440 2.8403 2.8394 4.0146 0
96 0.8672 1.3024 1.1323 0.7568 1.6091 2.7147 3.7169 0
97 2.6737 0.6448 1.5334 1.5747 1.3902 3.6321 6.2399 0
98 1.9178 1.0408 1.1499 2.1099 1.7620 4.3342 7.6834 0
99 1.2966 1.1982 1.6916 1.8868 1.9815 3.0361 7.6597 0
100 1.6916 0.5361 0.9663 1.6393 2.2734 2.3995 3.2762 8.6857
101 1.4712 0.8104 1.0314 2.3654 1.5598 3.3235 3.4926 5.8404
102 1.8449 0.9602 2.0882 2.5831 1.7495 4.1742 3.3657 0
103 1.3964 1.1592 1.5681 1.9055 3.6620 3.7478 3.9180 0
104 2.0268 1.0119 0.8461 2.5511 3.7432 5.2705 3.5831 0
105 1.9759 0.3947 2.0725 1.7499 2.9620 2.3582 3.7911 0
106 1.7972 0.8097 1.1811 1.0520 4.1368 3.2656 3.4545 0
107 1.6677 0.6867 0.9302 1.7455 2.1655 2.7414 3.9156 0
108 1.7541 0.9010 1.1636 2.0174 3.5691 2.8988 5.4100 0
109 1.4383 1.1282 1.4630 1.9644 2.5182 3.4189 6.6003 0
110 1.6318 1.0152 1.3424 2.0822 2.1652 3.5522 7.3678 0
111 1.1347 1.5308 1.0838 1.9629 2.7909 3.6708 6.7298 0
112 1.0705 0.9953 0.9358 2.6200 2.5914 3.5918 5.0396 0
113 2.8894 1.0933 1.3803 1.9310 3.2616 3.0142 8.5488 0
114 2.2151 1.2180 0.8661 1.8301 0.9485 3.7859 7.8162 0
115 0.9405 1.2198 1.3990 2.1633 2.9292 3.4481 5.7142 0]

116
117 D2: [2.3036 1.0673 1.4400 2.7063 2.1923 4.5825 3.5551 6.2214
118 2.1117 0.9154 1.2136 1.4661 2.5226 3.0288 5.4398 3.3969
119 2.9125 0.9735 0.8892 2.4945 2.5439 4.7532 3.8917 3.3326
120 2.0772 1.2095 2.0898 3.4496 2.3321 3.6615 3.5372 4.8369
121 1.6445 1.0228 2.5461 1.7821 3.3118 4.6515 3.3698 6.3941
122 2.5113 0.8949 2.2198 2.5457 3.6127 4.6390 6.6048 6.6878
123 1.3535 0.7639 1.1724 1.7634 2.9433 1.8130 3.1236 0
124 1.3847 0.6795 0.7920 1.7311 1.9119 2.4012 3.1595 0
125 1.7088 0.8004 1.1955 1.5136 2.6419 1.9401 4.5591 0
126 1.5710 0.9156 0.9679 1.9230 2.3736 1.9169 3.2888 0
127 0.7568 0.8977 1.1320 2.0730 2.6005 2.1824 3.5604 0
128 1.7395 1.0564 1.1685 1.4064 2.3437 4.1202 6.6140 0
129 2.0257 0.5071 1.9561 1.4228 2.6995 2.3076 2.7818 0
130 2.3079 0.8988 1.3777 1.8509 2.6080 2.4694 7.8432 0
131 2.1393 0.3315 0.8069 2.0306 2.0453 4.0885 3.3051 8.0558
132 1.5874 0.4372 1.0808 1.3357 2.3399 3.1571 3.6869 5.6817

133 2.0382 1.1705 1.5369 2.0402 3.5958 3.3524 6.3725 0
134 2.4231 1.0817 1.6741 1.5415 2.7121 4.1181 5.4140 0
135 1.2886 1.2449 0.9425 2.3725 2.6202 4.0764 4.6467 0
136 1.4581 0.9904 2.0387 1.5078 2.3536 2.8613 4.5906 0
137 1.5699 0.9453 1.2873 1.8477 2.9571 2.8700 1.2443 0
138 1.8051 1.3515 1.6648 2.3718 2.2989 2.6811 4.9976 0
139 3.2565 0.9559 1.6818 1.7207 3.0374 5.2443 4.9949 0
140 2.1398 1.1068 1.6293 2.0355 2.5127 5.0388 6.3106 0
141 2.8266 1.5678 1.3863 1.4690 2.6479 2.3836 6.6692 0
142 2.4931 0.7811 1.3980 1.5232 2.6727 3.9482 8.3583 0
143 2.0390 0.9004 1.3184 1.2721 2.9331 3.8359 4.7773 0
144 1.7940 1.0451 1.1781 1.5397 3.2623 5.1773 5.1520 0
145 1.7083 0.9473 0.7977 1.5527 2.9319 3.5988 6.9051 0
146 2.3703 0.9601 1.1916 1.5259 2.0545 3.1997 3.3752 0]
147
D3: [3.0081 1.3156 1.3275 3.0569 3.3628 4.3715 4.1266 4.3920
148 2.5170 0.9346 1.6501 2.5059 3.2834 3.2008 5.7869 3.0810
149 2.4210 1.5181 1.9738 2.4342 2.9332 4.9547 5.6921 6.1123
150 4.0752 0.8926 1.1118 2.2508 2.5911 3.9923 3.3730 8.0185
151 2.9958 1.0621 1.5283 2.2004 2.5831 4.4686 4.0606 4.2970
152 1.7744 0.7573 1.4003 2.6234 2.1151 3.8966 2.2859 5.1654
153 1.6966 1.0833 0.9157 1.4250 2.0627 2.4939 4.5432 0
154 1.1638 0.7943 1.3105 1.6362 3.2289 3.5430 4.8002 0
155 0.7630 0.7791 1.4088 2.3003 2.0696 1.5938 4.0495 0
156 1.1920 0.7136 1.2887 1.2597 2.3661 2.5048 3.9534 0
157 1.5215 0.7831 1.4880 0.9749 2.1511 3.3879 2.5010 0
158 2.8887 0.8997 0.9906 1.5862 1.6761 2.9059 7.2340 0
159 2.4221 1.3654 1.0871 1.3076 1.5496 2.7278 7.2900 0
160 2.0397 1.0828 1.0261 1.6659 2.4257 2.7394 7.6117 0
161 1.2962 0.9793 1.6178 1.2148 2.0818 3.2666 3.4984 0
162 1.6947 0.2523 0.9236 1.0843 2.4493 3.2440 3.7755 11.9197
163 1.2962 0.9793 1.6178 1.2148 2.0818 3.2666 3.4984 0
164 1.7398 0.5926 1.3410 2.2117 2.2223 3.6956 4.5390 0
165 2.2190 0.7593 1.7888 1.5568 1.8265 2.8189 5.6011 0
166 2.2529 1.0231 1.7759 2.1280 2.2137 3.5716 6.5526 0
167 2.0500 1.1779 1.4761 1.7295 3.0441 3.2371 6.4869 0
168 2.2572 1.3157 1.7053 2.0401 2.4892 3.0216 4.7826 0
169 1.5424 1.0555 0.2866 1.2755 2.5657 3.0809 5.3454 0
170 2.5329 0.2653 1.6098 1.7658 2.2840 4.3854 7.1894 0
171 1.7452 0.7825 0.7911 1.4915 2.5349 2.8504 5.9098 0
172 2.8616 1.3858 1.3784 1.8332 2.7865 3.8427 6.2737 0
173 2.6341 1.2674 1.2656 1.2811 1.6889 4.4513 2.0232 0
174 1.6186 1.1872 0.4574 1.8657 3.4409 3.0967 6.2763 0
175 2.7221 1.0186 1.7727 1.3711 2.1707 3.5404 5.3838 0
176 2.0720 1.4093 0.9339 1.9654 1.9577 3.3833 9.8419 0]
177

178 D4: [2.6261 0.7973 1.5039 2.1333 2.1267 3.2601 4.3550 2.6987
179 2.6990 0.9439 1.7641 2.3241 2.4378 4.9148 5.1610 6.5007
180 2.6721 0.7684 2.0406 1.7402 3.9554 3.3150 4.2893 7.4307
181 2.7214 0.8455 1.2378 2.2558 2.9251 2.7494 2.5978 2.4169
182 2.2181 0.8572 1.5432 2.5450 2.9035 2.2022 3.7114 5.9892
183 2.5534 1.5176 1.8581 2.1892 2.2798 3.6650 2.4203 3.9796
184 0.9870 0.9232 1.5098 1.6141 2.1393 1.8938 3.8644 0
185 1.5319 0.8938 1.4532 2.3308 1.8737 2.7757 2.13949 0
186 1.3395 0.9104 1.3290 1.2006 2.8971 1.1159 3.4773 0
187 1.0424 0.5905 1.4788 1.4325 2.7318 2.6629 3.0825 0
188 1.3558 1.1202 1.4835 2.2713 2.1034 0.6202 2.4925 0
189 1.5362 1.1084 1.8944 1.8795 3.0494 3.3172 8.3618 0
190 1.9569 0.9435 1.4524 2.3220 1.3730 2.1707 8.2483 0
191 1.9267 1.3518 1.1451 2.1355 2.9953 3.8768 8.9984 0
192 1.4644 0.6692 1.3343 1.6544 2.3949 3.1250 3.5180 16.1407
193 0.9262 0.8439 0.9580 2.0292 2.4085 4.4661 4.2976 10.6269
194 1.3824 0.8325 1.2415 1.9103 2.5255 4.5602 5.4097 0
195 2.5245 0.9352 1.2396 1.9288 2.0770 3.6246 5.3946 0
196 1.2865 1.2205 1.3650 1.7027 2.8233 3.6846 5.1707 0
197 1.5775 1.1722 1.2937 1.8694 2.4980 4.3594 4.2248 0
198 1.9571 0.9182 0.9742 1.0841 1.8747 4.1448 4.1669 0
199 1.6044 0.5175 1.1805 0.9098 3.6299 3.8417 4.8939 0
200 1.8009 1.5319 1.1600 1.9057 2.7185 3.7608 6.8070 0
201 1.2431 0.7080 0.6758 2.0164 1.6114 5.8964 6.3605 0
202 1.9707 0.8062 1.1986 2.6022 3.2581 3.8418 8.4042 0
203 2.0803 1.1741 0.9831 1.7012 3.6501 3.1552 5.6400 0
204 2.0254 0.6043 1.9123 1.1904 2.4289 2.5041 7.3081 0
205 1.8215 0.8433 1.2558 1.6864 1.5512 4.0074 6.6171 0
206 1.4516 1.2974 0.9357 0.9689 4.1820 2.3738 5.6593 0
207 1.9258 1.0685 0.8714 1.8383 2.7087 4.2752 5.7589 0]
208
209
210 D5: [1.8086 1.4044 1.3023 1.3424 4.5007 2.0959 4.4558 6.2837
211 2.5949 1.0789 1.2776 2.0811 3.8034 2.9238 3.2701 5.2408
212 2.1100 1.0225 1.3633 2.3885 4.2814 4.5938 5.3726 7.3294
213 1.1330 1.3181 2.0143 3.2584 2.3333 3.9020 4.8510 5.2287
214 1.6640 1.7755 1.7072 2.6904 2.9407 4.4588 3.3019 6.0410
215 2.3293 0.7121 1.3882 1.2075 2.3132 4.6200 5.3698 4.6051
216 1.4615 1.1073 1.2544 1.4482 2.3279 2.6653 3.5876 0
217 1.1988 1.0835 1.3115 1.5624 2.6988 3.2463 4.4870 0
218 1.3316 0.8587 1.3004 1.6599 1.7730 3.1521 3.7601 0
219 1.7389 0.7852 0.8024 0.9841 1.6735 2.4489 5.8680 0
220 1.8081 0.9267 1.2255 1.6925 2.7003 1.3061 3.1500 0
221 2.4932 0.8059 1.3483 1.9601 1.8508 2.7592 9.1338 0
222 1.8248 1.3021 0.9781 1.8164 2.1105 2.7234 5.7237 0

223 2.0765 0.7926 1.3346 1.8817 3.4147 3.8883 7.1746 0
 224 1.7383 0.2987 1.1036 1.2656 2.8150 4.2200 5.4480 9.9459
 225 1.6104 0.7966 0.9286 1.3831 2.4129 2.0015 5.4253 11.5019
 226 0.9968 0.8761 1.4489 1.9854 2.3103 3.4647 6.1439 0
 227 2.0341 1.1073 1.0108 2.3392 3.1501 3.0730 5.7624 0
 228 2.0939 1.1988 1.3334 1.7757 2.7348 3.5499 5.3518 0
 229 1.7085 1.2998 1.4862 1.7722 2.5040 1.3391 7.1248 0
 230 2.2774 0.5917 1.0536 2.2255 1.9649 3.0453 4.2913 0
 231 2.1551 1.1530 1.5095 1.7841 2.2934 2.5764 6.4521 0
 232 2.0250 0.8743 1.5875 1.9035 2.3161 3.5643 6.8492 0
 233 1.0578 0.9684 1.2955 0.9750 2.1510 2.5035 9.6689 0
 234 1.5534 0.8256 0.6682 2.7064 1.3538 4.1150 7.4634 0
 235 2.0396 1.4144 1.2384 1.0083 2.2946 5.8054 5.5282 0
 236 2.1339 1.3187 1.0450 2.0537 2.4396 5.1147 9.3216 0
 237 1.5213 1.1281 1.5422 1.9097 3.6273 3.8194 4.8078 0
 238 2.5229 0.8692 1.2392 1.5102 1.7828 3.7981 5.7316 0
 239 1.6614 0.9307 1.6761 1.8663 2.9133 4.5037 9.7266 0]
 240
 D6: [2.4299 1.3688 1.9204 2.6874 3.2631 3.1091 3.9126 6.4664
 241 2.1695 1.5135 1.7197 2.3861 2.9219 3.7076 3.8480 7.3674
 242 2.5426 1.0242 2.1839 2.3938 3.6502 3.3186 2.2127 5.5873
 243 2.8154 0.8618 1.8996 1.2509 2.9272 4.2377 4.4465 7.3801
 244 2.4283 1.1214 0.8419 2.0247 2.6692 3.3892 6.1868 5.8346
 245 2.8659 0.9941 1.3475 2.9419 3.3148 2.9559 2.2844 4.6927
 246 2.0404 0.5901 1.1087 1.9302 2.0010 2.1157 2.9950 0
 247 1.4480 1.2100 1.0498 1.9106 1.1082 2.6265 4.6232 0
 248 1.6007 0.8858 1.8748 1.1614 2.3708 3.7003 3.7768 0
 249 2.0103 0.9934 0.7718 1.6419 2.2157 2.3520 2.4100 0
 250 1.6270 0.9299 1.1609 2.4933 1.8094 1.9619 3.4945 0
 251 2.6452 0.7400 1.3488 1.8822 3.6234 2.1872 5.6692 0
 252 1.9617 0.8436 1.1810 1.7784 3.0710 2.6904 8.2344 0
 253 2.5072 1.1770 1.0618 1.6576 1.6876 3.9167 8.0061 0
 254 1.5947 0.7025 1.2178 1.8617 1.8960 2.7056 3.9361 9.9441
 255 1.6688 0.7606 0.9725 2.1994 2.3756 4.2608 5.8444 13.9264
 256 2.4757 1.4894 0.8595 1.7496 2.8334 3.0258 5.0001 0
 257 1.5585 1.5521 1.7235 1.7660 3.0463 4.1603 4.4995 0
 258 1.6182 0.7818 0.9887 2.2240 1.5914 2.8350 6.1909 0
 259 1.9417 0.7562 1.4395 1.3039 3.0246 3.8399 5.2749 0
 260 1.9108 1.1849 1.5985 1.7226 1.7748 3.8930 6.4771 0
 261 1.9581 1.3041 1.4210 2.6586 1.7581 2.9820 6.4612 0
 262 1.6177 1.0442 1.3343 1.4458 2.1609 5.0621 5.5496 0
 263 2.0993 1.0822 1.3502 1.1317 2.7675 4.4323 6.1223 0
 264 1.9961 1.2258 1.2393 1.2109 2.7812 5.2329 7.1361 0
 265 1.3463 0.7252 1.3549 2.5452 1.5239 4.3220 4.2427 0
 266 1.5343 0.9422 0.9618 2.6242 2.2142 4.8750 4.1469 0
 267

268 2.1428 0.7906 1.2774 1.7871 2.0815 4.0803 3.3541 0
 269 2.3591 0.9054 1.2160 1.7949 2.8352 4.3729 4.7232 0
 270 2.5610 0.5884 0.7491 2.2030 1.8600 3.4842 4.9155 0]
 271
 D7: [2.8900 1.2882 1.5227 1.3855 2.2993 3.9151 5.1432 6.0263
 272 1.6672 1.4667 1.5066 1.8674 3.0479 1.6909 4.2350 8.1658
 274 1.7367 1.6773 1.9651 3.0410 2.6759 1.7380 3.9418 5.7926
 275 1.5021 1.1606 1.3672 2.3470 3.4270 4.0530 4.0533 5.9007
 276 2.6004 1.2109 2.2900 2.3624 3.5122 4.5272 3.3822 7.4683
 277 1.7529 0.9831 1.7197 2.1429 3.0962 3.6293 5.8869 4.1834
 278 0.8703 0.8699 1.5801 1.7259 1.7686 3.4421 3.3638 0
 279 1.3198 0.7844 1.4443 2.0456 2.7162 2.4003 2.8390 0
 280 1.7594 0.8489 1.0799 1.8884 1.7608 2.8442 2.9900 0
 281 1.5544 0.9136 1.8198 1.7424 2.4168 2.8146 3.2972 0
 282 2.0215 1.1514 0.3631 1.9989 1.6506 3.2958 4.4267 0
 283 2.6093 1.2037 1.4052 1.5271 3.3483 3.8881 7.4408 0
 284 2.0843 1.1369 1.3065 2.4484 2.7527 3.2516 5.7509 0
 285 2.5913 1.1594 1.4696 1.9086 2.6454 2.2334 6.7907 0
 286 1.5673 0.6332 1.0770 0.8750 1.9304 1.9899 4.4970 8.2841
 287 1.5346 0.4548 0.8862 1.2942 2.5491 4.9715 4.0243 6.8074
 288 2.4668 0.8873 1.6175 1.8902 3.1160 4.3050 4.0828 0
 289 2.3434 1.1386 1.5905 1.7440 1.8813 3.8181 5.7380 0
 290 1.7432 0.8051 1.4178 2.0310 3.3170 3.0906 4.2194 0
 291 2.7387 1.1069 1.2877 1.9128 3.3491 4.0695 5.8159 0
 292 1.7522 1.0551 1.5071 2.8684 3.6315 3.1924 3.2522 0
 293 1.4538 1.0532 1.3487 1.5925 2.9509 5.5482 4.7454 0
 294 1.7976 0.7397 0.8486 2.0174 2.7619 3.8258 8.2783 0
 295 2.4468 0.7556 1.0654 1.5335 2.8036 3.7785 7.6586 0
 296 1.3420 0.7041 0.6183 1.2045 1.8897 4.0889 3.8036 0
 297 2.7456 0.6810 0.9706 2.1665 2.3855 4.7507 4.1974 0
 298 1.8398 1.0039 1.0701 1.0805 3.2297 4.5652 6.3719 0
 299 1.3704 0.7580 0.9002 1.9472 3.1314 2.6829 6.6296 0
 300 2.5037 0.9502 1.9102 2.1614 2.5420 2.6650 5.3493 0
 301 2.6631 1.4186 1.7121 1.4488 1.9422 5.3740 6.1727 0]
 302
 D8: [2.7982 1.1182 1.8836 1.6022 3.2312 3.8135 4.2530 5.5964
 303 1.9626 1.0796 2.5403 2.4314 2.6682 4.4716 4.2855 7.6165
 304 2.6632 1.2788 1.8123 2.0244 3.8795 3.1594 3.3376 3.5462
 305 2.0686 1.8747 1.7143 1.771 3.4463 3.5109 5.2523 5.4075
 306 2.7014 1.4494 1.0722 2.3700 4.0829 2.4591 2.5806 4.5716
 307 2.1557 1.4027 1.8212 3.5077 2.6545 4.3630 3.0783 5.6706
 308 1.2790 0.5435 0.9490 1.5776 1.5758 3.1117 3.6115 0
 309 1.4797 1.0680 1.6599 1.7914 3.1436 1.9666 3.4109 0
 310 1.6819 0.8201 1.2341 1.9627 1.9514 1.6581 3.9871 0
 311 1.1112 0.9619 1.1540 1.6735 1.6898 2.2055 3.1405 0
 312

313 1.2845 0.6119 1.4537 1.9919 1.9998 2.4161 4.0214 0
314 1.3288 1.5123 1.6204 1.7229 2.7450 3.7344 6.3281 0
315 2.0897 0.7301 1.6009 2.4340 2.7221 4.7561 7.2469 0
316 1.6092 0.7380 1.9943 2.2564 2.0125 1.9573 6.5648 0
317 1.8935 0.9715 1.1042 2.5128 1.8895 2.6709 3.1938 8.5543
318 2.0071 0.6202 1.2383 1.8125 2.6621 2.4018 2.6716 8.0201
319 2.2165 1.1175 1.2394 1.1405 2.2403 2.0191 4.0415 0
320 0.9457 1.1771 1.5855 1.8926 2.5680 3.3709 3.0404 0
321 1.5673 1.1943 1.5627 2.3029 2.6409 2.9781 5.4348 0
322 1.4689 0.9381 0.9698 1.8057 3.0961 4.0390 2.5039 0
323 1.4581 1.1838 1.6354 1.6776 2.0165 3.0579 3.0679 0
324 2.3199 1.2955 1.6425 2.4536 2.3617 3.8651 3.1285 0
325 2.3330 1.0559 1.2433 1.7776 1.8138 3.2431 7.1776 0
326 2.0044 1.1488 1.1817 2.5047 1.8005 4.4849 5.7934 0
327 2.8183 1.3606 1.8453 1.3980 2.2696 3.6706 5.6502 0
328 1.0493 1.0512 1.2840 1.2936 0.4916 3.3615 5.3150 0
329 1.9435 0.9175 1.0129 1.6801 1.5914 4.9867 4.5512 0
330 1.8754 0.9866 0.9477 1.1849 2.3546 3.9142 3.9931 0
331 2.1377 0.9703 1.4140 2.0452 2.3798 6.0492 9.0372 0
332 2.1638 0.7689 1.0799 1.9415 3.0215 4.1971 3.8235 0
333
334 D9: [3.0932 1.5746 1.2194 2.2979 1.9066 4.2361 4.2960 5.8473
335 2.9781 1.3878 1.6572 1.7274 2.7987 2.9939 5.0778 6.7837
336 2.9919 0.9192 1.6276 2.3124 2.7455 5.6039 4.4453 5.3724
337 2.8962 1.0297 1.6443 1.9954 1.7445 3.1754 4.1301 5.9746
338 2.4731 1.2426 1.3724 2.3699 3.2532 3.0256 3.7975 5.2230
339 2.5568 1.3285 0.5619 1.9553 1.4372 4.0838 3.2946 7.6567
340 1.2329 0.7950 1.3675 2.2341 1.8226 1.3954 2.5659 0
341 0.8028 1.2246 0.3404 1.0581 1.3949 2.0840 3.9393 0
342 1.5654 0.8189 1.3933 1.0690 2.1066 3.8945 3.4164 0
343 1.1484 0.8409 1.7410 2.1695 0.9871 3.3713 2.6741 0
344 1.0502 0.8803 1.4173 1.9061 2.2799 3.9242 3.5457 0
345 1.3525 1.4262 0.8004 2.1305 1.9760 2.4828 7.0427 0
346 2.0775 1.0187 1.0022 1.7340 2.3397 2.4060 6.8788 0
347 2.2274 1.1371 0.9167 1.8036 1.6452 3.5110 4.7420 0
348 1.8517 0.6217 1.0875 0.4384 2.5228 2.7315 5.0218 15.0672
349 1.9585 0.7100 1.3998 1.7556 1.8512 2.4617 4.7827 8.2022
350 2.0639 0.6583 1.7088 1.1323 2.3393 2.8517 3.4498 0
351 2.3766 1.0033 1.8209 1.7061 2.9924 3.5402 5.2940 0
352 1.8017 0.5602 1.6641 0.8550 3.9424 2.7137 3.9761 0
353 2.2015 1.3038 0.8903 1.3738 1.6440 3.5297 4.9764 0
354 1.9869 1.1225 1.6170 1.8258 2.5633 3.3895 4.1276 0
355 1.7425 0.8314 1.0281 1.0827 1.6385 2.9230 3.7049 0
356 1.3926 1.0296 1.7621 2.1419 2.7463 3.9560 6.5638 0
357 2.0980 1.5127 1.5976 1.5899 3.5240 3.3100 5.9706 0

358 2.4238 1.0019 1.2954 2.0779 2.6311 3.7967 6.6943 0
359 2.2436 0.7992 1.3334 2.0474 2.3398 4.3856 8.0057 0
360 2.2828 0.9306 1.2921 2.4650 2.1853 3.5525 7.1232 0
361 2.5243 0.7208 1.3592 1.7471 0.5051 2.1903 4.0445 0
362 1.3758 0.6917 1.4881 1.8902 2.7786 4.0818 5.2156 0
363 1.8919 0.6749 1.2971 1.8907 2.7726 2.2438 8.0264 0]
364
D10:[2.1614 1.3316 2.1129 2.2241 1.5196 3.4615 4.9322 8.5426
365 2.6261 0.9832 1.6366 2.5209 3.5371 3.6571 5.0451 6.2282
366 2.4442 1.3551 2.4817 1.9268 1.7290 2.2479 3.9935 5.0123
367 1.9264 0.7342 1.9233 1.7108 3.3873 3.9204 3.6783 6.7964
368 3.0089 1.0484 2.1533 1.5140 3.4780 3.6684 4.4925 6.3764
369 3.0656 1.0397 1.9915 2.1957 3.8988 4.9173 3.0587 6.9954
370 1.2598 0.8761 1.2774 1.7303 1.2425 3.0051 3.9090 0
371 1.0681 0.2707 1.5576 1.2591 2.7559 2.9191 3.2014 0
372 1.0309 0.6373 1.1690 1.7818 2.4335 2.9652 2.3974 0
373 0.9483 0.9047 1.0829 1.5075 1.5411 3.4604 2.0168 0
374 1.0575 1.1740 1.2037 2.0996 1.8299 2.7040 3.0112 0
375 1.1316 1.3815 1.2379 1.8950 2.2311 3.3044 11.0390 0
376 2.4469 0.8333 1.3357 2.1868 3.1171 3.9896 7.8874 0
377 2.2406 0.5583 1.9200 1.6463 1.5058 4.3146 6.4323 0
378 1.7152 0.4428 0.7138 1.5925 2.0071 3.0050 3.6480 9.8775
379 2.8808 0.9429 1.0903 2.8579 3.4538 3.8000 5.7781 0
380 1.0259 0.6789 0.8302 1.4149 2.5649 3.3898 5.6582 0
381 2.3740 1.1539 1.6328 1.8054 3.3402 2.4523 4.8546 0
382 1.1964 0.9455 1.0133 2.1700 1.7903 2.4158 3.7121 0
383 2.0972 1.2522 1.1198 1.3112 2.1577 3.1467 5.8404 0
384 1.5051 0.9931 1.1127 2.2389 2.8170 3.5348 4.7531 0
385 1.7803 0.9876 1.1580 1.7862 2.2286 3.7170 5.3586 0
386 2.0084 0.8685 1.1488 1.6546 3.9480 1.9942 6.5511 0
387 1.7750 0.7829 1.6648 2.2025 3.1762 2.9637 6.1419 0
388 2.4607 0.9967 1.2112 1.3985 2.4083 5.2271 4.5261 0
389 2.5145 0.6723 1.3010 2.1555 2.9871 0.4838 9.1944 0
390 1.9573 1.1346 1.0253 2.0410 4.1563 4.2015 6.3799 0
391 2.3424 0.7339 1.3208 2.0285 2.3807 3.6826 5.3793 0
392 2.2454 0.7236 1.0031 1.6379 2.4319 5.2158 6.7049 0]
393
D11:[2.3851 1.2033 2.5894 1.6495 3.5378 4.0536 6.1495 6.7710
394 2.5929 1.1014 1.3524 3.1833 2.2057 2.9233 3.1481 6.1986
395 3.0676 1.0964 1.4446 1.8859 3.9198 3.0007 3.5727 6.9085
396 2.4550 0.6875 1.0278 2.1424 3.4823 4.9825 2.6861 5.3070
397 0.8710 1.4640 2.0465 2.8529 2.3303 2.5730 4.2755 6.6741
398 3.2478 1.4305 2.1471 2.3306 2.6225 1.8970 5.7167 5.6957
399 1.2279 0.9547 1.4008 1.5450 1.9080 1.7420 2.8820 0
400

403 1.6419 0.9823 1.0049 1.4971 2.2290 1.7486 3.3167 0
 404 1.0858 0.8130 1.5731 0.9922 1.5258 3.1538 4.4344 0
 405 1.8024 1.1303 1.4621 1.5199 1.8123 2.4469 4.1481 0
 406 1.3131 0.7061 1.1367 0.9862 2.0971 3.5793 2.6992 0
 407 1.9071 0.6415 0.7861 2.2562 2.3762 3.2531 7.2713 0
 408 1.6928 1.1783 1.1606 2.1388 3.0906 4.2414 7.4008 0
 409 2.0851 1.5346 1.6194 1.8775 2.0058 2.5321 8.6327 0
 410 2.1725 0.6840 1.1777 1.9366 1.8726 2.3254 4.6789 8.1352
 411 0.9702 0.6915 0.9193 1.4147 2.11743 2.5079 4.1536 7.5546
 412 1.2745 1.2026 1.3058 2.1110 1.8324 3.1839 5.7012 0
 413 1.7272 1.0021 1.5133 0.7695 2.5928 3.2486 5.2462 0
 414 1.3701 1.1047 1.4964 1.5034 1.0645 4.3035 7.8735 0
 415 1.5597 1.3771 1.3152 1.8687 2.6418 3.5723 5.2384 0
 416 2.2707 0.5899 1.2416 1.9550 3.6246 3.5244 5.7459 0
 417 1.5527 1.3668 1.5826 1.5663 2.6311 3.8744 6.4212 0
 418 1.4956 0.7754 0.9861 1.5461 1.5332 3.5853 8.1750 0
 419 1.8464 1.0210 1.1982 1.9712 2.4413 2.8002 5.2331 0
 420 1.2613 0.8422 1.2026 1.2848 3.9188 3.8399 7.4841 0
 421 2.0258 0.7715 1.2378 1.7201 2.7384 5.5954 6.3249 0
 422 1.8735 1.1143 1.5806 1.5342 3.1843 3.2230 5.1825 0
 423 2.1282 0.9512 1.5208 1.3514 3.3545 3.9804 5.9235 0
 424 1.3875 1.1969 1.0095 2.1161 2.2407 2.5643 8.6256 0
 425 1.9643 1.1836 0.9162 1.6942 2.3203 3.8959 6.7805 0]

426
 427 D12:[2.4752 0.8889 1.3653 2.3507 2.5801 3.1986 4.7725 5.6514
 428 1.9504 0.7835 2.4283 2.5174 3.7133 2.2330 3.8893 7.2974
 429 2.3328 1.0972 1.3840 1.4186 1.6895 4.0423 4.6618 6.7106
 430 1.9101 1.0559 2.4037 2.3243 3.0502 3.5196 5.6283 4.9429
 431 3.8338 0.6980 1.5551 1.9170 1.4921 2.4858 6.0978 6.1774
 432 1.7480 1.0954 1.3209 1.6855 2.9333 3.4414 4.6544 6.7492
 433 0.9157 0.8446 0.8492 1.4091 1.9353 2.4158 4.2322 0
 434 1.3667 0.7051 1.3941 1.9478 1.1085 2.4639 2.2255 0
 435 1.1509 0.7984 0.8674 1.1208 2.1537 3.2006 2.7299 0
 436 1.1441 0.8447 0.8712 1.7866 2.8291 1.9232 3.4581 0
 437 1.0150 0.4789 1.5361 1.7460 2.3936 2.2873 4.3333 0
 438 2.0589 1.2677 2.0901 2.1564 2.5263 2.1712 6.2271 0
 439 1.1294 1.3887 1.7311 1.5892 2.6339 2.6333 6.9755 0
 440 1.1722 0.8608 1.2567 1.0113 2.5048 4.5040 8.7667 0
 441 1.2147 0.8471 1.2848 1.9266 1.5457 1.6207 5.7656 4.4617
 442 1.6939 0.7198 1.0232 1.1912 1.5062 3.4746 3.0580 8.7985
 443 1.6373 1.2250 0.7464 1.3005 2.4727 3.4269 3.7827 0
 444 1.2971 1.2405 1.1035 1.5989 2.7426 4.2647 2.7715 0
 445 2.1675 0.9887 1.7094 1.1018 2.3810 3.2877 5.4139 0
 446 2.1946 0.7589 1.2604 2.6691 3.1060 2.2592 4.7618 0
 447 1.2093 1.2700 1.3577 2.1831 2.11768 3.9353 5.3825 0

448 1.2363 0.5842 1.6299 1.1499 3.6835 3.5068 3.7511 0
449 2.4615 1.0633 1.5083 1.3617 1.5633 4.7347 6.7480 0
450 2.1338 0.7572 1.0591 1.5877 2.9797 4.2697 3.6885 0
451 2.4119 0.8297 1.2898 0.9619 2.0725 4.8892 6.7860 0
452 1.8058 0.8897 1.1775 2.0987 2.3917 3.2372 9.4374 0
453 1.9594 0.8423 1.2167 1.5049 2.6312 4.0489 9.3311 0
454 1.9519 1.3586 2.0221 1.1775 1.6545 2.8536 9.6237 0
455 2.0812 1.1897 0.8666 1.4927 2.9863 3.9917 7.0292 0
456 1.5439 1.4139 1.6117 0.9345 3.7265 1.7125 8.7979 0]

D13: [2.9524 0.6186 1.8623 2.2425 2.6355 3.6238 5.7230 5.4766
458 2.8549 0.9897 0.6461 2.2780 2.8103 4.2065 4.7696 6.8732
459 2.2058 0.9596 1.9649 3.0709 2.5072 3.1030 6.2607 6.7735
460 2.4454 1.0938 1.2220 2.3759 2.3313 5.1214 4.4051 3.8574
461 1.8292 1.1256 1.5278 1.9994 3.1658 4.3792 3.8218 5.0341
462 2.3878 1.1709 1.5176 2.6214 2.8635 3.2036 6.5723 6.9837
463 0.8140 1.0824 1.2556 2.2702 2.9602 3.4888 4.6497 0
464 0.8073 0.6356 1.3324 1.5728 2.3321 2.4047 3.7267 0
465 1.0160 0.8165 1.1138 1.7285 1.3841 1.9764 3.7633 0
466 1.2725 0.7958 1.3366 1.4659 2.8730 3.0425 3.3860 0
467 1.2189 0.8566 0.5064 1.7555 2.0748 4.0393 3.8764 0
468 3.2070 0.9758 0.9467 1.8264 2.7696 2.7433 8.0783 0
469 2.1578 1.2371 1.6965 1.5350 2.8252 3.2972 9.3830 0
470 2.5672 1.4170 0.9333 2.9959 3.2276 1.4578 7.0533 0
471 1.4083 0.7445 1.1160 1.7802 1.8093 3.8734 5.3948 9.0792
472 1.6106 0.7062 1.0141 1.7117 2.0968 3.3874 3.8319 10.8240
473 1.6255 0.8313 1.8443 1.8073 3.4383 2.3765 5.2709 0
474 1.8704 0.7061 1.4425 1.7598 1.8087 3.0224 4.2968 0
475 3.0087 0.9884 0.9362 2.3963 3.3815 4.0943 3.6433 0
476 1.0820 0.9014 1.4184 2.2550 2.4545 4.5591 6.8556 0
477 1.6849 1.1858 1.3100 2.3332 2.6831 3.3330 6.1804 0
478 2.2420 0.8475 1.1343 1.8123 2.4002 3.0629 4.9347 0
479 2.4315 0.6305 1.7529 1.9352 2.2781 4.2954 4.8991 0
480 2.1386 1.1270 1.5645 1.5264 2.2661 2.9553 6.3171 0
481 1.7462 0.9895 1.0059 0.7831 2.4292 3.4675 6.7183 0
482 1.6624 0.7980 1.0760 1.5751 2.1343 2.7857 7.0820 0
483 2.8613 0.7031 1.5218 1.5123 1.5987 3.9781 6.7863 0
484 2.8192 0.9417 0.8783 1.9320 2.4710 3.6127 7.0400 0
485 1.5532 0.4465 1.0649 1.0974 2.5033 3.7620 7.7031 0
486 2.0406 0.7625 1.3473 1.8860 2.6713 4.4820 4.9543 0]

D14: [2.5592 1.0596 1.2835 1.7881 2.3981 1.6146 5.4188 5.3929
488 3.1674 1.3899 1.6198 2.3517 2.2173 6.0826 4.2841 3.9562
489 3.1527 1.1108 1.0829 2.4774 3.4694 3.3189 5.0647 6.2825
490 2.1918 1.5489 1.9056 2.1041 3.5226 3.2573 6.3415 6.0122

493 3.9763 0.5894 2.1682 2.1147 2.7093 3.1181 3.4302 6.7334
494 3.1411 1.3038 1.5892 1.4898 2.3846 1.7839 4.4895 5.9404
495 1.7298 0.9518 1.4163 1.3024 2.1772 2.9500 3.1001 0
496 1.6545 0.9985 1.4155 1.2788 2.5729 3.2465 3.1568 0
497 1.0106 0.8978 0.9684 2.2098 1.2964 3.4784 4.2757 0
498 1.8832 0.8152 1.3538 2.2018 1.1347 3.1205 3.9967 0
499 1.7475 0.7772 1.2960 0.4120 1.5370 1.7388 4.4279 0
500 1.6459 1.1237 1.5527 1.9337 2.3043 3.5609 5.1227 0
501 2.5458 1.0469 1.1700 1.3149 1.4933 1.7001 7.0195 0
502 1.5744 0.8517 1.4043 1.6504 1.6615 3.7129 8.0088 0
503 1.7457 0.8905 1.3474 1.4249 1.8214 4.0219 4.0130 8.5712
504 2.2006 0.7425 1.0673 1.5344 2.2188 2.1664 4.4806 11.3547
505 1.4180 1.0319 0.9230 1.6913 2.3288 3.1291 5.0238 0
506 1.8066 1.1458 1.0590 2.5912 2.5509 2.4757 6.2181 0
507 1.2581 0.7073 1.2771 0.8319 2.3392 3.0401 5.1176 0
508 1.8387 1.0114 1.3134 1.5435 3.2682 3.7967 3.2244 0
509 1.9797 0.8883 1.2129 1.5232 2.6633 2.7827 6.1011 0
510 2.1770 1.0969 1.6355 1.7544 3.7296 3.3352 4.8393 0
511 2.0096 1.0221 1.4475 1.0007 3.0648 3.3307 5.7920 0
512 2.7409 1.2731 0.4682 2.0317 2.5610 4.0318 5.6442 0
513 2.5222 1.2901 1.4896 1.8896 2.5444 3.1921 6.6085 0
514 2.5011 0.9353 1.4292 1.5788 3.4834 3.9594 5.0728 0
515 1.5319 1.2851 0.5580 1.3513 3.3841 4.3639 5.6777 0
516 1.3877 1.0566 1.5804 0.9862 1.4501 4.0211 8.2475 0
517 2.4195 1.2268 1.0904 1.8110 2.6342 4.6351 8.0823 0
518 2.3255 0.8483 0.9908 1.7229 3.6993 2.9053 6.2074 0
519
520 D15: [2.3864 1.2860 1.3685 2.9866 2.8718 4.9646 2.0811 4.5223
521 2.9475 0.7836 1.7433 2.1226 2.6597 2.9488 5.4724 3.2492
522 1.8133 0.9127 1.6266 2.2810 2.1043 2.5594 3.6627 7.5478
523 2.1757 1.1165 1.6434 2.2775 2.8550 2.3098 5.1770 8.0361
524 2.1968 1.2729 2.0841 2.4957 2.3887 3.2961 6.8851 5.3921
525 2.1531 1.2512 2.1032 2.2414 3.9556 2.9719 5.2505 8.1017
526 1.3026 0.8370 1.3062 1.0652 1.5836 2.8880 3.0148 0
527 1.5799 0.5809 1.2157 0.6833 1.8614 2.8965 3.5288 0
528 1.7736 1.3346 1.1681 1.8489 2.2160 1.5439 3.3717 0
529 1.3182 0.7170 0.9877 2.1579 1.5789 1.4697 3.5568 0
530 1.0368 0.7565 1.8785 1.6631 2.1879 3.2334 4.8760 0
531 1.2238 1.3197 1.7047 2.2855 1.7479 3.3356 6.9313 0
532 0.8747 0.9916 1.6389 1.7346 1.8286 3.0571 7.7365 0
533 1.7013 0.7407 1.3150 1.2607 2.3491 2.9347 5.7175 0
534 1.2526 1.0098 1.1668 2.1822 2.6152 3.3799 3.40163 3.9889
535 1.9483 0.8073 1.1855 2.6235 2.2804 1.6737 4.0391 11.4377
536 1.2979 1.0552 1.2327 2.3629 3.8262 3.3815 6.5508 0
537 1.7799 0.8842 1.3477 2.0552 3.8453 2.9068 3.5471 0

538 1.9273 1.2156 1.1994 2.3363 2.8874 3.1500 3.1139 0
539 1.9376 1.4771 1.0823 1.6667 2.7872 3.3230 4.4170 0
540 1.4467 0.9564 1.5190 1.5214 2.3868 3.6321 3.6697 0
541 1.6907 1.0139 1.1744 1.7932 2.5523 2.7207 5.1261 0
542 2.3019 1.2339 0.9327 1.8781 3.2231 5.0997 8.9742 0
543 2.3287 0.7595 1.8470 1.5586 2.0414 4.9009 7.0513 0
544 2.0542 0.8108 1.6663 1.6021 2.1845 4.3306 3.3280 0
545 2.4412 1.1026 0.7735 1.6205 3.0994 0.7031 9.6848 0
546 2.3602 1.2484 1.2902 1.8858 2.5668 4.5453 7.0609 0
547 2.0394 1.0614 1.0205 1.8240 2.3278 5.2675 8.4489 0
548 1.3139 0.9885 1.3702 2.1793 2.1010 3.8579 4.3087 0
549 1.6066 1.0606 1.2950 0.6575 1.8683 5.4491 4.9040 0]
550
551 D16:[1.8054 0.8530 1.7227 2.6706 3.3487 3.2924 2.9831 5.6902
552 1.6811 1.6850 1.7056 1.9779 2.3011 3.7843 1.4747 5.6721
553 3.2297 0.8853 1.7688 2.2844 1.7465 4.5122 4.0683 5.9655
554 3.4500 1.0954 1.8298 2.3523 2.9318 1.8976 3.9270 7.2275
555 3.0637 1.3215 1.2048 3.2743 2.9880 4.1720 6.0722 7.0719
556 3.2193 0.8814 2.0998 2.1744 3.4261 3.7818 4.1588 7.0954
557 1.5916 1.2219 0.8538 1.6368 2.3103 2.5300 2.8958 0
558 1.6013 0.8796 1.6082 1.5432 0.8866 2.4011 4.0784 0
559 1.6046 0.8442 1.0165 1.5783 2.9168 2.8269 2.0622 0
560 0.6964 1.0235 1.2625 1.6698 2.2162 3.0206 4.5082 0
561 1.5986 0.5671 1.1063 1.2587 2.2556 2.2145 2.6669 0
562 1.2385 0.6928 1.4918 1.8972 1.1619 2.5242 6.4659 0
563 2.4720 1.2358 1.4729 2.0471 1.7613 3.6334 8.9590 0
564 1.7565 1.3010 1.4538 2.4309 2.4495 2.8354 5.1975 0
565 1.6354 0.5944 0.9324 2.3406 1.8442 2.7075 4.0293 9.0705
566 2.2343 0.6900 1.5328 1.6170 1.9389 2.5861 4.1198 6.2654
567 1.5645 1.0600 1.5792 1.3428 2.5759 2.8024 4.1283 0
568 1.6308 0.9152 1.2299 1.8953 2.3030 0.6637 6.2621 0
569 0.7291 1.3845 1.9195 1.7734 2.0864 1.7745 3.4085 0
570 1.3474 0.8635 1.4226 1.5235 3.0268 3.2263 4.5355 0
571 1.0330 1.1884 1.9887 2.2597 3.0641 2.6701 8.1113 0
572 1.1123 0.9091 1.3801 2.1663 3.3202 2.9046 4.6926 0
573 2.3197 0.7792 0.8266 1.4864 2.8991 1.8812 5.7971 0
574 2.0629 1.1978 0.5259 1.3530 2.2696 3.0792 6.5601 0
575 2.2412 0.8450 0.9762 2.2334 2.6229 5.0649 6.5546 0
576 1.4287 0.6434 1.7502 1.1266 3.2888 2.5299 6.3011 0
577 1.7161 0.9343 1.3978 0.9888 2.8255 3.1598 8.8239 0
578 2.4786 0.6230 1.3343 2.2890 1.8498 5.3604 7.6331 0
579 2.2263 1.0297 1.3180 1.3503 3.0823 4.5206 5.7706 0
580 2.3439 0.6485 1.6057 1.8238 1.6027 3.5102 7.0162 0]
581
582 O:[1.207 1.336 1.192 1.048 1.049 1.342 1.432 1.206

583 1.148 1.064 1.418 1.433 1.448 1.242 1.1391.036
584 1.453 1.303 1.034 1.092 1.368 1.222 1.0931.475
585 1.313 1.199 1.174 1.086 1.193 1.238 1.264 1.343
586 1.485 1.030 1.204 1.423 1.132 1.295 1.089 1.193
587 1.178 1.061 1.162 1.322 1.199 1.300 1.307 1.467
588 1.117 1.423 1.411 1.336 1.180 1.422 1.403 1.291
589 1.490 1.023 1.075 1.396 1.497 1.244 1.247 1.431
590 1.199 1.171 1.213 1.214 1.367 1.268 1.348 1.368
591 1.061 1.127 1.177 1.472 1.169 1.340 1.339 1.343
592 1.324 1.099 1.467 1.402 1.466 1.270 1.225 1.276
593 1.140 1.290 1.079 1.115 1.064 1.348 1.049 1.049
594 1.491 1.178 1.493 1.278 1.476 1.362 1.482 1.351
595 1.177 1.378 1.339 1.088 1.486 1.289 1.203 1.359
596 1.087 1.460 1.184 1.174 1.368 1.096 1.054 1.064
597 1.026 1.417 1.346 1.320 1.233 1.064 1.099 1.420
598 1.108 1.157 1.224 1.377 1.454 1.068 1.057 1.497
599 1.397 1.486 1.053 1.306 1.177 1.392 1.329 1.025
600 1.376 1.420 1.373 1.367 1.018 1.479 1.030 1.091
601 1.332 1.187 1.438 1.169 1.051 1.445 1.418 1.467
602 1.471 1.341 1.394 1.306 1.199 1.061 1.389 1.298
603 1.390 1.183 1.217 1.324 1.266 1.369 1.208 1.093
604 1.156 1.195 1.128 1.077 1.155 1.106 1.307 1.272
605 1.207 1.491 1.400 1.429 1.330 1.147 1.087 1.247
606 1.385 1.103 1.025 1.465 1.434 1.498 1.312 1.495
607 1.380 1.211 1.305 1.477 1.390 1.348 1.066 1.026
608 1.343 1.433 1.354 1.218 1.032 1.145 1.223 1.134
609 1.487 1.126 1.366 1.364 1.201 1.079 1.058 1.282
610 1.118 1.451 1.175 1.259 1.160 1.362 1.265 1.359
611 1.175 1.238 1.273 1.317 1.133 1.372 1.292 1.471]
612
613
614 f1:[0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
615 0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
616 0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
617 0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
618 0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
619 0.08 0.04 0.12 0.19 0.2 0.03 0.25 0.09
620 0.02 0.14 0.19 0.25 0.32 0.02 0.06 0.0
621 0.02 0.14 0.19 0.25 0.32 0.02 0.06 0.0
622 0.02 0.14 0.19 0.25 0.32 0.02 0.06 0.0
623 0.02 0.14 0.19 0.25 0.32 0.02 0.06 0.0
624 0.02 0.14 0.19 0.25 0.32 0.02 0.06 0.0
625 0.05 0.14 0.17 0.21 0.22 0.11 0.1 0.0
626 0.05 0.14 0.17 0.21 0.22 0.11 0.1 0.0
627 0.05 0.14 0.17 0.21 0.22 0.11 0.1 0.0

628 0.04 0.33 0.19 0.12 0.14 0.08 0.05 0.06
 629 0.04 0.33 0.19 0.12 0.14 0.08 0.05 0.06
 630 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 631 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 632 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 633 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 634 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 635 0.09 0.1 0.18 0.21 0.21 0.16 0.05 0.0
 636 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 637 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 638 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 639 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 640 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 641 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 642 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 643 0.03 0.05 0.3 0.15 0.17 0.21 0.08 0.0
 644
 p1:[0.331051746 0.333185348 0.335762906 0
 645 0.167641468 0.166902703 0.162785635 0.165680763 0.169982581 0.167006849 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 646 0.314014218 0.337276495 0.348709286 0
 648 0.248006347 0.256774441 0.243448534 0.251770678 0
 649 0.246079724 0.246469413 0.25370076 0.253750103 0
 650 0.339644776 0.327238725 0.333116498 0
 651 0.252279398 0.251230373 0.252213814 0.244276415 0
 652 0.52564529 0.47435471 0
 653 0.254224517 0.253193125 0.246591309 0.245991048 0
 654 0.2008332 0.197964282 0.201980749 0.20228349 0.196938279 0
 655 0.343321511 0.327307921 0.329370568 0
 656 1 0
 657 0.330939794 0.344793963 0.324266244 0
 658 0.505165817 0.494834183 0
 659 0.198445662 0.194261619 0.203421057 0.205041633 0.198830028 0
 660 0.161591757 0.173031248 0.163021898 0.168143446 0.166034591 0.168177058 0
 661 0.246931455 0.246205588 0.246416302 0.260446655 0
 662 0.251070845 0.243766303 0.247857667 0.257305185 0
 663 0.332380527 0.336599721 0.331019752 0
 664 0.250893142 0.240107236 0.250919758 0.258079865 0
 665 0.197782394 0.192459026 0.202550966 0.199938023 0.207269592 0
 666 0.251109982 0.256351157 0.24696982 0.245569041 0
 667 0.164925606 0.165668592 0.168710763 0.168909948 0.163513845 0.168271246 0
 668 0.163209621 0.173929073 0.165054203 0.15582686 0.174485444 0.167494798 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 669 0.496619346 0.503380654 0
 670 0.257199689 0.24414869 0.251767112 0.24688451 0
 671 0.254932788 0.250098984 0.244313095 0.250655134 0
 672 0.492114108 0.507885892 0

718 0.201207629 0.197084493 0.198042761 0.202895714 0.200769403 0
719 0.201765198 0.197451825 0.200307317 0.203007101 0.197468559 0
720 0.165729435 0.166578766 0.166553455 0.165641096 0.167609237 0.167888012 0
721 0.202021156 0.197945709 0.196489036 0.204046156 0.199497943 0
722 0.251853504 0.248700388 0.25038929 0.249056818 0
723 0.165916587 0.168865685 0.164584941 0.165638131 0.167355407 0.167639249 0
724 1 0
725 0.503270039 0.496729961 0
726 0.497645187 0.502354813 0
727 0.331311262 0.333545687 0.335143051 0
728 0.2477105 0.250095003 0.252505488 0.249689008 0
729 1 0
730 0.252529653 0.251635617 0.248555947 0.247278784 0
731 0.203367681 0.199858636 0.197550958 0.198792684 0.200430042 0
732 0.500669205 0.499330795 0
733 0.166166141 0.167801585 0.166486451 0.167587422 0.166867907 0.165090493 0
734 1 0
735 0.169172335 0.163248643 0.169325609 0.165894305 0.166414592 0.165944516 0
736 0.332320966 0.33440546 0.333273574 0
737
738 p4: [1 0
739 1 0
740 0.200428837 0.199536314 0.199873895 0.19921047 0.200950484 0
741 0.49720765 0.50279235 0
742 0.202082647 0.199421411 0.200906187 0.197152264 0.200437491 0
743 0.246434193 0.248838703 0.252222527 0.252504576 0
744 0.329905753 0.333742775 0.336351472 0
745 1 0
746 0.201587044 0.198048944 0.201434627 0.200073886 0.198855499 0
747 0.336067684 0.332669348 0.331262968 0
748 0.331253014 0.33388225 0.334864736 0
749 0.335405994 0.330713365 0.33388064 0
750 0.16551686 0.166663953 0.167758834 0.16621758 0.166657954 0.167184818 0
751 0.501858919 0.498141081 0
752 0.335676053 0.331607785 0.332716162 0
753 0.332090454 0.334992742 0.332916805 0
754 1 0
755 0.505875374 0.494124626 0
756 0.496021263 0.503978737 0
757 0.197438595 0.19725461 0.201035519 0.201805537 0.20246574 0
758 0.24989414 0.249127717 0.250483444 0.250494699 0
759 0.199372822 0.199735664 0.201810573 0.19686229 0.202218651 0
760 0.200999659 0.200751188 0.198035602 0.201062016 0.199151536 0
761 0.165610514 0.167304756 0.166904987 0.166716351 0.165137669 0.168325723 0
762 1 0

853 0.198909185 0.201141814 0.199699996 0.200495511 0.199753494 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 854 0.50375967 0.49624033 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 855 0.166267734 0.165488974 0.165709493 0.167587489 0.167466888 0.167479421 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 856 1 0
 857 0.166588661 0.167611534 0.166008161 0.168825637 0.165825272 0.165140735 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 858 0.493646444 0.506353556 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 859 0.197186683 0.201209136 0.196781614 0.204025805 0.200796762 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 860 0.249592475 0.248366981 0.253897979 0.248142564 0
 861
 p8:[1 0
 862 1 0
 863 1 0
 864 0.497520222 0.502479778 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 865 1 0
 866 0.168503598 0.168880539 0.165732098 0.166369867 0.164331372 0.166182526 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 867 1 0
 868 1 0
 869 1 0
 870 0.335880733 0.334218713 0.329900554 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 871 0.331650659 0.330367971 0.33798137 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 872 0.19866827 0.200437384 0.200888427 0.200348275 0.199657644 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 873 0.166351629 0.166546952 0.168271153 0.166597091 0.168024545 0.164208629 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 874 1 0
 875 0.251330995 0.249421417 0.250386576 0.248861011 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 876 1 0
 877 1 0
 878 0.494354348 0.505645552 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 879 0.169089791 0.166197353 0.165465545 0.168072667 0.1669225 0.164252145 0 0 0 0 0 0 0 0 0 0 0 0
 880 0.332422488 0.331600793 0.335976719 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 881 0.500779853 0.499220147 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 882 0.249217753 0.252536967 0.245004849 0.253240432 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 883 1 0
 884 0.495562662 0.504437338 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 885 0.503290505 0.496709495 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 886 0.503781882 0.496218118 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 887 0.497319793 0.502680207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 888 0.333336495 0.334864438 0.331799068 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 889 1 0
 890 0.332003386 0.332472764 0.33552385 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 891 0.330302957 0.331031922 0.338665121 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 892
 p9:[0.345134548 0.323393118 0.331472334 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 893 0.20198943 0.198940466 0.198625921 0.197896305 0.202547877 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 894 0.176313126 0.162881329 0.162581087 0.1676967 0.169893501 0.160634257 0 0 0 0 0 0 0 0 0 0 0 0 0
 895 0.337050567 0.330984483 0.33196495 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 896 0.197027284 0.196773182 0.199376916 0.202234355 0.204588263 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 897

APPENDIX 3 - SIMULATED ANNEALING METAHEURISTIC - MATLAB CODE

MAIN SIMULATED ANNEALING CODE (SIMULATEDANNEALING.M)

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %Set random seed
4 rng(5); %5
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8 %Data for model
9 fileId = fopen('Data/parameters.txt');
10 formatSpec = '%s m = %d n = %d T = %d h = %d v = %d R = %d E = %d numberScenarios = %d %*s';
11 parameters = cell2mat(textscan(fileID,formatSpec, ...
12     'delimiter', '\n', ...
13     'CollectOutput', true));
14 fclose(fileID);
15
16 m = parameters(1);
17 n = parameters(2);
18 T = parameters(3);
19 l = parameters(4);
20 h = parameters(5);
21 v = double(parameters(6));
22 R = parameters(7);
23 E = parameters(8);
24 numberScenarios = parameters(9);
25
26 omegaFilename = 'Data/omega.txt';
27 omegaDelimiterIn = ' ';
28 omegaHeaderLinesIn = 0;
29 omega = importdata(omegaFilename,omegaDelimiterIn,omegaHeaderLinesIn);
30
31 piFilename = 'Data/pi.txt';
32 piDelimiterIn = ' ';
33 piHeaderLinesIn = 0;
34 pi = importdata(piFilename,piDelimiterIn,piHeaderLinesIn);
35
36 DmeanFilename = 'Data/D.txt';
37 fFilename = 'Data/f.txt';
38 DmeanDelimiterIn = ' ';
39 DmeanHeaderLinesIn = 0;
40 Dmean = importdata(DmeanFilename,DmeanDelimiterIn,DmeanHeaderLinesIn);

```

```

41 f = importdata(Filename,DmeanDelimiterIn,DmeanHeaderLinesIn);
42
43 D = zeros(m,E,numberScenarios);
44
45 D = generateSurgicalDurations(Dmean,0.25,m,E,numberScenarios);
46
47 %Probabilities
48 p = zeros(m,l,numberScenarios);
49
50 for scenario = 1:4:16
51     p(:,scenario) = generateLengthOfStayProbabilities(0.2,0.005,m,l);
52 end
53
54 for scenario = 2:4:16
55     p(:,scenario) = generateLengthOfStayProbabilities(0.3,0.005,m,l);
56 end
57
58 for scenario = 3:4:16
59     p(:,scenario) = generateLengthOfStayProbabilities(0.5,0.005,m,l);
60 end
61
62 for scenario = 4:4:16
63     p(:,scenario) = generateLengthOfStayProbabilities(0.6,0.005,m,l);
64 end
65
66 %Block constraint
67 B1 = [ 1 7 12 15 17 23];
68 F2 = [ 6 11 14 16 22 30];
69 B = [36 30 18 12 36 48];
70
71 %Generate overhead time factors
72 O = zeros(m,E);
73
74 for i = 1:m
75     for e = 1:E
76         O(i,e) = unifrnd(1.01,1.5);
77     end
78 end
79
80 %Cancellation values
81 meanCancellations = 50;
82 stdDevCancellations = 2;
83
84 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
85 %Targets

```



```

86 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87 L = randi([1,5],m,1);
88 H = randi([6,20],m,1);
89 N = randi([20,35],m,1);
90 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91 %Calculations and Initial Schedule Generation
92 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93 A = zeros(m,scenario);
94 maxNumberPatients = zeros(m,scenario);
95 PHI = zeros(m,1,scenario);
96 for scenario = 1:numberScenarios
97     for i = 1:m
98         A(:,scenario) = sum(times(times(D(:, :,scenario),O),f),2);
99         maxNumberPatients(i,scenario) = floor(v/A(i,scenario));
100         for lCount = 1:l
101             PHI(i,lCount,scenario) = maxNumberPatients(i,scenario)*sum(p(i,lCount+1:l,scenario),2);
102         end
103     end
104     numberBlocksTotal = n*(T-8)
105     numberBlocksAvailable = numberBlocksTotal;
106     %Create an empty schedule
107     x = zeros(m,n,T);
108     lowerBoundDev = zeros(m);
109     upperBoundDev = zeros(m);
110     isInitialScheduleFeasible = 0;
111     numberAssignments = 0;
112     %Sort from highest to lowest number of pi constraints
113     piSum = sum(pi,1);
114     [piSumA,piSumB] = sort(piSum, 'descend');
115     %Generate the initial schedule
116     while isInitialScheduleFeasible == 0
117         while numberBlocksAvailable > 0
118             while (sum(sum(x,2),3) >= L) < m
119                 for i = 1:m
120                     lowerBoundDev(i) = L(i) - sum(sum(x(i, :, :),2),3);
121                 end
122             end
123         end
124     end

```

```

131 for i = m:-1:1
132 for k = 1:T
133     for j = piSumB
134         if(numberBlocksAvailable > 0 && lowerBoundDev(i) > 0)
135             if(numberBlocksAvailable > 0 && x(i,j,k) == 0)
136                 x(i,j,k) = 1;
137                 if(isFeasible(x, 'partialSchedule', m, n, T, l, h, v, R, E, ...
138                     numberScenarios, pi, omega, L, H, B, B1, F2, ...
139                     maxNumberPatients, N, A, PHI) == 1)
140                     numberBlocksAvailable = numberBlocksAvailable - 1;
141                     numberAssignments = numberAssignments + 1;
142                     lowerBoundDev(i) = lowerBoundDev(i) - 1;
143                 else
144                     x(i,j,k) = 0;
145                 end
146             end
147         end
148     end
149 end
150 end
151 end
152 end
153 while (sum(sum(x,2),3) > H) == 0 && numberBlocksAvailable > 0
154     for i = 1:m
155         upperBoundDev(i) = H(i) - sum(sum(x(i,:),:),2),3);
156     end
157 end
158 for i = 1:m
159     for k = 1:T
160         for j = piSumB
161             if(numberBlocksAvailable > 0 && upperBoundDev(i) > 0)
162                 if(numberBlocksAvailable > 0 && x(i,j,k) == 0)
163                     x(i,j,k) = 1;
164                     if(isFeasible(x, 'partialSchedule', m, n, T, l, h, v, R, E, ...
165                         numberScenarios, pi, omega, L, H, B, B1, F2, ...
166                         maxNumberPatients, N, A, PHI) == 1)
167                         numberBlocksAvailable = numberBlocksAvailable - 1;
168                         numberAssignments = numberAssignments + 1;
169                         upperBoundDev(i) = upperBoundDev(i) - 1;
170                     else
171                         x(i,j,k) = 0;
172                     end
173                 end
174             end
175         end

```

```

176         end
177     end
178     end
179     end
180     end
181     end
182
183     if(isFeasible(x, 'partialSchedule', m, n, T, l, h, v, R, E, ...
184         numberScenarios, pi, omega, L, H, B, B1, F2, ...
185         maxNumberPatients, N, A, PHI) == 1)
186         isInitialScheduleFeasible = 1;
187     else
188         isInitialScheduleFeasible = 0;
189     end
190 end
191
192 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
193 %Inner and Outer Loops
194 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
195 xInitial = x;
196
197 rng();
198
199 %Parameters
200 coolingRate = 0.95;
201 initialTemperature = 1000;
202 finalTemperature = 0.1;
203 maximumSolutionsAccepted = 3;
204
205 [energy, dMinusSum, d2nSum, l, IMax, IRange] = ...
206     energyFunction(x, m, n, T, l, h, v, R, E, ...
207     numberScenarios, pi, omega, L, H, B, B1, F2, ...
208     maxNumberPatients, N, A, PHI);
209 initialEnergy = energy;
210
211 currentEnergy = initialEnergy;
212 currentTemperature = initialTemperature;
213
214 NA = 0;
215 NATotal = 0;
216
217 while currentTemperature > finalTemperature
218     [energy, dMinusSum, d2nSum, l, IMax, IRange] = ...
219         energyFunction(x, m, n, T, l, h, v, R, E, ...
220         numberScenarios, pi, omega, L, H, B, B1, F2, ...

```

```

221     maxNumberPatients, N, A, PHI);
222     currentEnergy = energy;
223
224     NATotal = NATotal + NA;
225     NA = 0;
226
227     while NA < maximumSolutionsAccepted
228         i = randi([1,m]);
229         j = randi([1,n]);
230         k = randi([1,T]);
231
232         if(i == 30)
233             i2 = 1;
234         else
235             i2 = i + 1;
236         end
237
238         j2 = randi([1,n]);
239         k2 = randi([1,T]);
240
241         if(x(i,j,k) == 0 && x(i2,j2,k2) == 1)
242             status = 1;
243             x(i,j,k) = 1;
244             x(i2,j2,k2) = 0;
245
246         elseif(x(i,j,k) == 1 && x(i2,j2,k2) == 0)
247             status = 2;
248             x(i,j,k) = 0;
249             x(i2,j2,k2) = 1;
250         end
251
252         if(x(i,j,k) ~= x(i2,j2,k2))
253             if(isFeasible(x, 'fullSchedule', m, n, T, l, h, v, R, E, ...
254                 numberScenarios, pi, omega, L, H, B, B1, F2, ...
255                 maxNumberPatients, N, A, PHI) == 1)
256
257                 [energy, d1MinusSum, d2hSum, l, IMax, IRange] = ...
258                 energyFunction(x, m, n, T, l, h, v, R, E, ...
259                 numberScenarios, pi, omega, L, H, B, B1, F2, ...
260                 maxNumberPatients, N, A, PHI);
261                 newEnergy = energy;
262
263                 %See if the new schedule
264                 %is better or not
265                 if(newEnergy < currentEnergy)

```

```

266 NA = NA + 1;
267 %Accept new solution and swap values
268 currentEnergy = newEnergy;
269
270
271 elseif(newEnergy >= currentEnergy)
272     %Metropolis Criterion
273     if exp(-(newEnergy - currentEnergy)/ currentTemperature) > unifrnd(0,1)
274         %Accept new solution and swap values
275         NA = NA + 1;
276         currentEnergy = newEnergy;
277     else
278         %Reject new solution and don't swap values
279         if(status == 1)
280             x(i,j,k) = 0;
281             x(i2,j2,k2) = 1;
282         elseif(status == 2)
283             x(i,j,k) = 1;
284             x(i2,j2,k2) = 0;
285         end
286     end
287
288 else
289     %Reject due to infeasibility
290     if(status == 1)
291         x(i,j,k) = 0;
292         x(i2,j2,k2) = 1;
293     elseif(status == 2)
294         x(i,j,k) = 1;
295         x(i2,j2,k2) = 0;
296     end
297
298 else
299     end
300
301 currentTemperature = currentTemperature * coolingRate;
302 end
303
304 %Display results before cancellations
305 isFeasible(x, 'fullSchedule', m, n, T, l, h, v, R, E, ...
306 numberScenarios, pi, omega, L, H, B, B1, F2, ...
307 maxNumberPatients, N, A, PHI);
308 [energy, dlMinusSum, d2nSum, I, IMax, IRange] = ...
309 energyFunction(x, m, n, T, l, h, v, R, E, ...
310 numberScenarios, pi, omega, L, H, B, B1, F2, ...

```

```

311     maxNumberPatients, N, A, PHI);
312
313 %Cancellations
314 xBeforeCancellations = x;
315
316 c = zeros(m,n,T);
317 for i = 1:m
318     for j = 1:n
319         for k = 1:T
320             if(x(i,j,k) == 1)
321                 c(i,j,k) = normrnd(meanCancellations, stdDevCancellations);
322
323                 if((c(i,j,k) <= (meanCancellations - 2*stdDevCancellations) || ...
324                    (c(i,j,k) >= (meanCancellations + 2*stdDevCancellations)))));
325                     x(i,j,k) = 0;
326                 end
327             end
328         end
329     end
330 end
331
332 %Display results after cancellations
333 xAfterCancellations = x;
334 isFeasible(xAfterCancellations, 'fullSchedule', m, n, T, l, h, v, R, E, ...
335           numberScenarios, pi, omega, L, H, B, B1, F2, ...
336           maxNumberPatients, N, A, PHI);
337 [energyAfterCancellations, dMinusSumAfterCancellations, ...
338  dZnSumAfterCancellations, IAfterCancellations, ...
339  IMaxAfterCancellations, IRangeAfterCancellations] = ...
340  energyFunction(xAfterCancellations, m, n, T, l, h, v, R, E, ...
341               numberScenarios, pi, omega, L, H, B, B1, F2, ...
342               maxNumberPatients, N, A, PHI);
343 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
344 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

FUNCTION TO CHECK FOR FEASIBILITY (ISFEASIBLE.M)

```
1 function [feasibility] = isFeasible(x, scheduleType, m, n, T, l, h, v, R, E, ...
2 numberScenarios, pi, omega, L, H, B, B1, F2, ...
3 maxNumberPatients, N, A, PHI)
4
5 infeasibilityCounter = 0;
6
7
8 if(strcmp(scheduleType, 'partialSchedule') == 1)
9     %Constraints 1 and 2
10    for k = 1:T
11        for j = 1:n
12            for i = 1:m
13
14                if(x(i,j,k) == 1 && pi(i,j) == 1)
15                    infeasibilityCounter = infeasibilityCounter + 1;
16                end
17
18                if(x(i,j,k) == 1 && omega(i,k) == 1)
19                    infeasibilityCounter = infeasibilityCounter + 1;
20                end
21            end
22        end
23    end
24
25    %Constraint 3
26    constraint3 = sum(x(1:m, :, :), 1);
27    for j = 1:n
28        for k = 1:T
29            if(constraint3(:,j,k) > 1)
30                infeasibilityCounter = infeasibilityCounter + 1;
31            end
32        end
33    end
34
35    %Constraint 4
36    constraint4 = sum(x(:, 1:n, :), 2);
37    for i = 1:m
38        for k = 1:T
39            if(constraint4(i, :, k) > 1)
40                infeasibilityCounter = infeasibilityCounter + 1;
41            end
42        end
43    end
44 end
```

```

44 %Constraint 5
45 constraint5 = sum(sum(x,2),3);
46 for i = 1:m
47     if(constraint5(i) > H(i))
48         infeasibilityCounter = infeasibilityCounter + 1;
49     end
50 end
51
52 %Constraint 6
53 for r = 1:R
54     if(sum(sum(x(B1(r):F2(r),:),1),3),2) > B(r))
55         infeasibilityCounter = infeasibilityCounter + 1;
56     end
57 end
58
59 elseif(strcmp(scheduleType, 'fullSchedule') == 1)
60 %Constraints 1 and 2
61 for k = 1:T
62     for j = 1:n
63         for i = 1:m
64             if(x(i,j,k) == 1 && pi(i,j) == 1)
65                 infeasibilityCounter = infeasibilityCounter + 1;
66             end
67         end
68     end
69 end
70     if(x(i,j,k) == 1 && omega(i,k) == 1)
71         infeasibilityCounter = infeasibilityCounter + 1;
72     end
73 end
74 end
75 end
76
77 %Constraint 3
78 constraint3 = sum(x(1:m,:),1);
79 for j = 1:n
80     for k = 1:T
81         if(constraint3(:,j,k) > 1)
82             infeasibilityCounter = infeasibilityCounter + 1;
83         end
84     end
85 end
86
87 %Constraint 4
88 constraint4 = sum(x(:,1:n,:),2);

```



```

89 for i = 1:m
90     for k = 1:T
91         if(constraint4(i,:,k) > 1)
92             infeasibilityCounter = infeasibilityCounter + 1;
93         end
94     end
95 end
96
97 %Constraint 5
98 constraint5a = sum(sum(x,2),3);
99 for i = 1:m
100     if(constraint5a(i) > H(i))
101         infeasibilityCounter = infeasibilityCounter + 1;
102     end
103 end
104
105 constraint5b = sum(sum(x,2),3);
106 for i = 1:m
107     if(constraint5b(i) < L(i))
108         infeasibilityCounter = infeasibilityCounter + 1;
109     end
110 end
111
112 %Constraint 6
113 for r = 1:R
114     if(sum(sum(x(B1(r):F2(r),:,1),3),2) ~= B(r))
115         infeasibilityCounter = infeasibilityCounter + 1;
116     end
117 end
118
119 else
120     infeasibilityCounter = 1;
121 end
122
123 %Return the feasibility of the schedule
124 if(infeasibilityCounter == 0)
125     feasibility = 1;
126 else
127     feasibility = 0;
128 end
129 end

```

FUNCTION TO CALCULATE ENERGY (ENERGYFUNCTION.M)

```
1 function [energy, dlMinusSum, d2nSum, I, IMax, IRange] = ...
2 energyFunction(x, m, n, T, l, h, v, ...
3 R, E, numberScenarios, pi, omega, L, H, B, B1, F2, ...
4 maxNumberPatients, N, A, PHI)
5
6 %Constraint 7
7 dlPlus = zeros(m,numberScenarios);
8 dlMinus = zeros(m,numberScenarios);
9 for scenario = 1:numberScenarios
10     for i = 1:m
11         if(maxNumberPatients(i,scenario)*sum(sum(x(i,:,:),2),3)) > N(i))
12             dlPlus(i,scenario) = maxNumberPatients(i,scenario)*sum(sum(x(i,:,:),2),3) - N(i);
13         elseif(maxNumberPatients(i,scenario)*sum(sum(x(i,:,:),2),3)) < N(i))
14             dlMinus(i,scenario) = N(i) - maxNumberPatients(i,scenario)*sum(sum(x(i,:,:),2),3);
15         end
16     end
17 end
18
19 dlMinusSum = zeros(numberScenarios,1);
20 for scenario = 1:numberScenarios
21     dlMinusSum(scenario) = sum(dlMinus(:,scenario),1);
22 end
23
24 %Constraint 8
25 d2n = zeros(m,n,T,numberScenarios);
26 for scenario = 1:numberScenarios
27     for i = 1:m
28         for j = 1:n
29             for k = 1:T
30                 if(x(i,j,k) == 0)
31                     d2n(i,j,k,scenario) = 0;
32                 elseif(x(i,j,k) == 1 && v > A(i,scenario)*maxNumberPatients(i,scenario))
33                     d2n(i,j,k,scenario) = v - A(i,scenario)*maxNumberPatients(i,scenario);
34                 end
35             end
36         end
37     end
38 end
39
40 d2nSum = zeros(numberScenarios,1);
41 for scenario = 1:numberScenarios
42     d2nSum(scenario) = sum(sum(d2n(:,:,:),scenario),1,2),3);
43 end
```

```

44 %Constraint 9
45 partialI1 = zeros(T,numberScenarios);
46 partialI2 = zeros(T,numberScenarios);
47 for scenario = 1:numberScenarios
48     for i = 1:m
49         for j = 1:n
50             for k = 1:T
51                 for lCount = 1:k
52                     partialI1(k,scenario) = partialI1(k,scenario) + x(i,j,lCount)*PHI(i,k-lCount+1,scenario);
53                     end
54                 end
55                 for lCount = k+1:T
56                     partialI2(k,scenario) = partialI2(k,scenario) + x(i,j,lCount)*PHI(i,k-lCount+1+T,scenario);
57                     end
58                 end
59             end
60         end
61     end
62 end
63 end
64 I = zeros(T, numberScenarios);
65 for scenario = 1:numberScenarios
66     for k = 1:T
67         I(k,scenario) = partialI1(k,scenario) + partialI2(k,scenario);
68     end
69 end
70
71 %IMax, IMin, IRange
72 Isort = zeros(T, numberScenarios);
73 IMax = zeros(numberScenarios,1);
74 IMin = zeros(numberScenarios,1);
75 IRange = zeros(numberScenarios,1);
76
77 for scenario = 1:numberScenarios
78     Isort(:,scenario) = sort(I(:,scenario));
79     IMax(scenario) = Isort(T,scenario);
80     IMin(scenario) = Isort(1,scenario);
81     IRange(scenario) = IMax(scenario) - IMin(scenario);
82 end
83
84 subProblemObjectiveFunction = zeros(numberScenarios,1);
85 for scenario = 1:numberScenarios
86     subProblemObjectiveFunction(scenario) = d1MinusSum(scenario) + ...
87         d2nSum(scenario) + ...
88         IMax(scenario) + ...

```

```
89     (IMax(scenario) - IMin(scenario));
90 end
91
92 energy = sum(subProblemObjectiveFunction(:))/double(numberScenarios);
93
94 end
```

FUNCTION TO GENERATE LENGTH OF STAY PROBABILITIES (GENERATELENGTHOFSTAYPROBABILITIES.M)

```
1 function [p] = generateLengthOfStayProbabilities(pGenMean, pGenStdDev, m, l)
2
3 pGen = zeros(m,1);
4 pGen2 = zeros(m);
5 pGen3 = zeros(m,1);
6 hGen = randi([1,6],m,1);
7
8 for i = 1:m
9     for lCount = 1:hGen(i)
10        pGen(i,lCount) = normrnd(pGenMean, pGenStdDev);
11    end
12
13    pGen2(i) = sum(pGen(i,:),2);
14
15    for lCount = 1:hGen(i)
16        pGen3(i,:) = pGen(i,:) / pGen2(i);
17    end
18    p = pGen3;
19 end
20 end
```

FUNCTION TO GENERATE SURGICAL DURATIONS (GENERATESURGICALDURATIONS.M)

```
1 function [D] = generateSurgicalDurations (Dmean,DGenStdDev,m,E,numberScenarios)
2
3 DGen = zeros(m,E,numberScenarios);
4 DGen2 = zeros(m,E);
5 DGen3 = zeros(m,E,numberScenarios);
6
7 DGen = normrnd(1,DGenStdDev,[m,E,numberScenarios]);
8
9 for i = 1:m
10     for e = 1:E
11         for scenario = 1:numberScenarios
12             DGen2(i,e) = DGen2(i,e) + DGen(i,e,scenario);
13         end
14     end
15 end
16
17 for scenario = 1:numberScenarios
18     for e = 1:E
19         for i = 1:m
20             DGen3(i,e,scenario) = (DGen(i,e,scenario) / DGen2(i,e));
21         end
22     end
23 end
24
25 for scenario = 1:numberScenarios
26     for e = 1:E
27         for i = 1:m
28             DGen4(i,e,scenario) = Dmean(i,e)*double(numberScenarios)*DGen3(i,e,scenario);
29         end
30     end
31 end
32
33 D = DGen4;
34 end
35
```

APPENDIX 4 - MATLAB DATA FILES

DATA FILE WITH REAL WORLD SURGICAL DURATIONS (~ DATA/D.TXT)

1	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
2	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
3	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
4	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
5	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
6	2.50	1.12	1.67	2.25	2.85	3.55	4.37	5.85
7	1.33	0.87	1.22	1.63	2.08	2.60	3.55	0
8	1.33	0.87	1.22	1.63	2.08	2.60	3.55	0
9	1.33	0.87	1.22	1.63	2.08	2.60	3.55	0
10	1.33	0.87	1.22	1.63	2.08	2.60	3.55	0
11	1.33	0.87	1.22	1.63	2.08	2.60	3.55	0
12	1.98	1.05	1.37	1.87	2.32	3.12	7.2	0
13	1.98	1.05	1.37	1.87	2.32	3.12	7.2	0
14	1.98	1.05	1.37	1.87	2.32	3.12	7.2	0
15	1.70	0.67	1.08	1.70	2.12	3.03	4.23	9.15
16	1.70	0.67	1.08	1.70	2.12	3.03	4.23	9.15
17	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
18	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
19	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
20	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
21	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
22	1.78	1.02	1.38	1.82	2.67	3.32	4.85	0
23	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
24	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
25	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
26	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
27	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
28	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
29	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0
30	2.02	0.98	1.23	1.70	2.53	3.83	6.42	0

DATA FILE WITH REAL WORLD SURGICAL FREQUENCIES (~ DATA/F.TXT)

1	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09
2	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09
3	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09
4	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09
5	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09

6	0.08	0.04	0.12	0.19	0.2	0.03	0.25	0.09
7	0.02	0.14	0.19	0.25	0.32	0.02	0.06	0.0
8	0.02	0.14	0.19	0.25	0.32	0.02	0.06	0.0
9	0.02	0.14	0.19	0.25	0.32	0.02	0.06	0.0
10	0.02	0.14	0.19	0.25	0.32	0.02	0.06	0.0
11	0.02	0.14	0.19	0.25	0.32	0.02	0.06	0.0
12	0.05	0.14	0.17	0.21	0.22	0.11	0.1	0.0
13	0.05	0.14	0.17	0.21	0.22	0.11	0.1	0.0
14	0.05	0.14	0.17	0.21	0.22	0.11	0.1	0.0
15	0.04	0.33	0.19	0.12	0.14	0.08	0.05	0.06
16	0.04	0.33	0.19	0.12	0.14	0.08	0.05	0.06
17	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
18	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
19	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
20	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
21	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
22	0.09	0.1	0.18	0.21	0.21	0.16	0.05	0.0
23	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
24	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
25	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
26	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
27	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
28	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
29	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0
30	0.03	0.05	0.3	0.15	0.17	0.21	0.08	0.0

DATA FILE WITH OMEGA CONSTRAINTS (~ DATA/OMEGA.TXT)T

1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1
2	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1
4	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1	1
5	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1	1
6	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1
7	0	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
8	0	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	1
9	0	0	1	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1
10	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	1
11	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1
12	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1
13	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1
14	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
15	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	1	1	0


```
16 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 1 1
17 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1
18 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 1
19 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1
20 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1
21 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 1
22 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1
23 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
24 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1
25 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1
26 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 1
27 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1 1
28 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1
29 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1
30 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1
```

DATA FILE WITH PARAMETERS (~ DATA/PARAMETERS.TXT)

```
1 begin
2 m = 30
3 n = 9
4 T = 28
5 l = 29
6 h = 28
7 v = 8
8 R = 6
9 E = 8
10 numberScenarios = 16
11 end
```

DATA FILE WITH PI CONSTRAINTS (~ DATA/PI.TXT)

```
1 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0
6 0 1 0 0 0 0 0 0 1
7 0 1 0 0 0 0 0 0 1
8 0 1 0 0 0 0 0 0 1
9 0 1 0 0 0 0 0 0 1
```

10	0	1	0	0	0	0	0	1
11	0	1	0	0	0	0	0	1
12	0	1	0	0	0	0	0	1
13	0	1	0	0	0	0	0	1
14	0	1	0	0	0	0	0	1
15	0	1	0	0	0	0	0	1
16	0	1	0	0	0	0	0	1
17	0	1	0	0	0	0	0	1
18	0	1	0	0	0	0	0	0
19	0	1	0	0	0	0	0	0
20	0	1	0	0	0	0	0	0
21	0	1	0	0	0	0	0	0
22	0	1	0	0	0	0	0	0
23	0	1	0	0	0	0	0	0
24	0	1	0	0	0	0	0	0
25	0	1	0	0	0	0	0	0
26	0	1	0	0	0	0	0	0
27	0	1	0	0	0	0	0	0
28	0	1	0	0	0	0	0	0
29	1	1	1	0	1	1	0	0
30	0	0	0	1	0	0	1	1

APPENDIX 5 - DISCRETE EVENT SIMULATION - PROMODEL CODE

```

1 *****
2 *
3 * Formatted Listing of Model:
4 *
5 *
6 *****
7
8 Time Units: Hours
9 Distance Units: Meters
10
11
12 *****
13 * Locations
14 *
15 *****
16
17 Name Cap Units Stats Rules Cost
18 ArrivalBuffer inf 1 Time Series Oldest, First
19 Arrival inf 1 Time Series Oldest, First
20 PreSurgeryBuffer inf 1 Time Series Oldest, First
21 OperatingRoom1 1 1 Time Series Oldest, First
22 OperatingRoom2 1 1 Time Series Oldest, First
23 OperatingRoom3 1 1 Time Series Oldest, First
24 OperatingRoom4 1 1 Time Series Oldest, First
25 OperatingRoom5 1 1 Time Series Oldest, First
26 OperatingRoom6 1 1 Time Series Oldest, First
27 OperatingRoom7 1 1 Time Series Oldest, First
28 OperatingRoom8 1 1 Time Series Oldest, First
29 OperatingRoom9 1 1 Time Series Oldest, First
30 PreRecoveryBuffer inf 1 Time Series Oldest, Fewest
31 Recovery 1 25 Time Series Oldest, Fewest
32 Recovery.1 1 1 Time Series Oldest,
33 Recovery.2 1 1 Time Series Oldest,
34 Recovery.3 1 1 Time Series Oldest,
35 Recovery.4 1 1 Time Series Oldest,
36 Recovery.5 1 1 Time Series Oldest,
37 Recovery.6 1 1 Time Series Oldest,
38 Recovery.7 1 1 Time Series Oldest,
39 Recovery.8 1 1 Time Series Oldest,
40 Recovery.9 1 1 Time Series Oldest,
41 Recovery.10 1 1 Time Series Oldest,
42 Recovery.11 1 1 Time Series Oldest,
43 Recovery.12 1 1 Time Series Oldest,
44 Recovery.13 1 1 Time Series Oldest,
45 Recovery.14 1 1 Time Series Oldest,
46 Recovery.15 1 1 Time Series Oldest,
47 Recovery.16 1 1 Time Series Oldest,
48 Recovery.17 1 1 Time Series Oldest,
49 Recovery.18 1 1 Time Series Oldest,
50 Recovery.19 1 1 Time Series Oldest,
51 Recovery.20 1 1 Time Series Oldest,
52 Recovery.21 1 1 Time Series Oldest,
53 Recovery.22 1 1 Time Series Oldest,
54 Recovery.23 1 1 Time Series Oldest,
55 Recovery.24 1 1 Time Series Oldest,
56 Recovery.25 1 1 Time Series Oldest,
57 RecoveryWardBuffer inf 1 Time Series Oldest,

```

58	RecoveryWard	1	60	Time Series Oldest	, Fewest
59	RecoveryWard.1	1	1	Time Series Oldest	,
60	RecoveryWard.2	1	1	Time Series Oldest	,
61	RecoveryWard.3	1	1	Time Series Oldest	,
62	RecoveryWard.4	1	1	Time Series Oldest	,
63	RecoveryWard.5	1	1	Time Series Oldest	,
64	RecoveryWard.6	1	1	Time Series Oldest	,
65	RecoveryWard.7	1	1	Time Series Oldest	,
66	RecoveryWard.8	1	1	Time Series Oldest	,
67	RecoveryWard.9	1	1	Time Series Oldest	,
68	RecoveryWard.10	1	1	Time Series Oldest	,
69	RecoveryWard.11	1	1	Time Series Oldest	,
70	RecoveryWard.12	1	1	Time Series Oldest	,
71	RecoveryWard.13	1	1	Time Series Oldest	,
72	RecoveryWard.14	1	1	Time Series Oldest	,
73	RecoveryWard.15	1	1	Time Series Oldest	,
74	RecoveryWard.16	1	1	Time Series Oldest	,
75	RecoveryWard.17	1	1	Time Series Oldest	,
76	RecoveryWard.18	1	1	Time Series Oldest	,
77	RecoveryWard.19	1	1	Time Series Oldest	,
78	RecoveryWard.20	1	1	Time Series Oldest	,
79	RecoveryWard.21	1	1	Time Series Oldest	,
80	RecoveryWard.22	1	1	Time Series Oldest	,
81	RecoveryWard.23	1	1	Time Series Oldest	,
82	RecoveryWard.24	1	1	Time Series Oldest	,
83	RecoveryWard.25	1	1	Time Series Oldest	,
84	RecoveryWard.26	1	1	Time Series Oldest	,
85	RecoveryWard.27	1	1	Time Series Oldest	,
86	RecoveryWard.28	1	1	Time Series Oldest	,
87	RecoveryWard.29	1	1	Time Series Oldest	,
88	RecoveryWard.30	1	1	Time Series Oldest	,
89	RecoveryWard.31	1	1	Time Series Oldest	,
90	RecoveryWard.32	1	1	Time Series Oldest	,
91	RecoveryWard.33	1	1	Time Series Oldest	,
92	RecoveryWard.34	1	1	Time Series Oldest	,
93	RecoveryWard.35	1	1	Time Series Oldest	,
94	RecoveryWard.36	1	1	Time Series Oldest	,
95	RecoveryWard.37	1	1	Time Series Oldest	,
96	RecoveryWard.38	1	1	Time Series Oldest	,
97	RecoveryWard.39	1	1	Time Series Oldest	,
98	RecoveryWard.40	1	1	Time Series Oldest	,
99	RecoveryWard.41	1	1	Time Series Oldest	,
100	RecoveryWard.42	1	1	Time Series Oldest	,
101	RecoveryWard.43	1	1	Time Series Oldest	,
102	RecoveryWard.44	1	1	Time Series Oldest	,
103	RecoveryWard.45	1	1	Time Series Oldest	,
104	RecoveryWard.46	1	1	Time Series Oldest	,
105	RecoveryWard.47	1	1	Time Series Oldest	,
106	RecoveryWard.48	1	1	Time Series Oldest	,
107	RecoveryWard.49	1	1	Time Series Oldest	,
108	RecoveryWard.50	1	1	Time Series Oldest	,
109	RecoveryWard.51	1	1	Time Series Oldest	,
110	RecoveryWard.52	1	1	Time Series Oldest	,
111	RecoveryWard.53	1	1	Time Series Oldest	,
112	RecoveryWard.54	1	1	Time Series Oldest	,
113	RecoveryWard.55	1	1	Time Series Oldest	,
114	RecoveryWard.56	1	1	Time Series Oldest	,
115	RecoveryWard.57	1	1	Time Series Oldest	,
116	RecoveryWard.58	1	1	Time Series Oldest	,
117	RecoveryWard.59	1	1	Time Series Oldest	,
118	RecoveryWard.60	1	1	Time Series Oldest	,

```

119 Discharge      inf 1      Time Series Oldest, ,
120
121
122 *****
123      Entites
124 *****
125
126 Name      Speed (mpm)  Stats      Cost
127 -----
128 PatientBatch 50      Time Series
129 Patient      50      Time Series
130
131
132 *****
133      Resources
134 *****
135
136 Name      Units  Stats      Res      Ent      Search  Search  Path      Motion      Cost
137 -----
138 Surgical_Team_1  1  By Unit  None  Oldest
139      Empty: 50 mpm
140      Full: 50 mpm
141
142 Surgical_Team_2  1  By Unit  None  Oldest
143      Empty: 50 mpm
144      Full: 50 mpm
145
146 Surgical_Team_3  1  By Unit  None  Oldest
147      Empty: 50 mpm
148      Full: 50 mpm
149
150 Surgical_Team_4  1  By Unit  None  Oldest
151      Empty: 50 mpm
152      Full: 50 mpm
153
154 Surgical_Team_5  1  By Unit  None  Oldest
155      Empty: 50 mpm
156      Full: 50 mpm
157
158 Surgical_Team_6  1  By Unit  None  Oldest
159      Empty: 50 mpm
160      Full: 50 mpm
161
162 Surgical_Team_7  1  By Unit  None  Oldest
163      Empty: 50 mpm
164      Full: 50 mpm
165
166 Surgical_Team_8  1  By Unit  None  Oldest
167      Empty: 50 mpm
168      Full: 50 mpm
169
170 Surgical_Team_9  1  By Unit  None  Oldest
171      Empty: 50 mpm
172      Full: 50 mpm
173
174 Surgical_Team_10 1  By Unit  None  Oldest
175      Empty: 50 mpm
176      Full: 50 mpm
177
178 Surgical_Team_11 1  By Unit  None  Oldest
179      Empty: 50 mpm
180      Full: 50 mpm
181
182 Surgical_Team_12 1  By Unit  None  Oldest
183      Empty: 50 mpm
184      Full: 50 mpm
185
186 Surgical_Team_13 1  By Unit  None  Oldest
187      Empty: 50 mpm
188      Full: 50 mpm
189
190 Surgical_Team_14 1  By Unit  None  Oldest
191      Empty: 50 mpm
192      Full: 50 mpm
193
194
195
196
197
198
199

```



```

241 OperatingRoom_5 1 By Unit None Oldest Empty: 50 mpm
242 Full: 50 mpm
243
244 OperatingRoom_6 1 By Unit None Oldest Empty: 50 mpm
245 Full: 50 mpm
246
247 OperatingRoom_7 1 By Unit None Oldest Empty: 50 mpm
248 Full: 50 mpm
249
250 OperatingRoom_8 1 By Unit None Oldest Empty: 50 mpm
251 Full: 50 mpm
252
253 OperatingRoom_9 1 By Unit None Oldest Empty: 50 mpm
254 Full: 50 mpm
255
256
257 *****
258 * Processing
259 *****
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301

```

Entity	Location	Operation	Blk	Output	Destination	Rule	Move Logic
PatientBatch	ArrivalBuffer	Split maximumNumberOfPatients			(PatientBatch_SurgicalTeam, PatientBatch_Scenario]	As Patient	
Patient	ArrivalBuffer	Patient_SurgicalTeam = PatientBatch_SurgicalTeam	1	PatientBatch EXIT		FIRST 1	
Patient	Arrival	Patient_OperatingRoom = PatientBatch_OperatingRoom					
Patient	Arrival	Patient_DayofSurgery = PatientBatch_DayofSurgery					
Patient	Arrival	Patient_Scenario = PatientBatch_Scenario					
Patient	Arrival	Wait N(60,3) min	1	Patient	Arrival	FIRST 1	
Patient	PreSurgeryBuffer	Graphic 2	1	Patient	PreSurgeryBuffer	FIRST 1	
Patient	PreSurgeryBuffer		1	Patient	OperatingRoom1	IF Patient_OperatingRoom = 1, 1	
Patient	PreSurgeryBuffer			Patient	OperatingRoom2	IF Patient_OperatingRoom = 2	
Patient	PreSurgeryBuffer			Patient	OperatingRoom3	IF Patient_OperatingRoom = 3	
Patient	PreSurgeryBuffer			Patient	OperatingRoom4	IF Patient_OperatingRoom = 4	
Patient	PreSurgeryBuffer			Patient	OperatingRoom5	IF Patient_OperatingRoom = 5	
Patient	PreSurgeryBuffer			Patient	OperatingRoom6	IF Patient_OperatingRoom = 6	
Patient	PreSurgeryBuffer			Patient	OperatingRoom7	IF Patient_OperatingRoom = 7	
Patient	PreSurgeryBuffer			Patient	OperatingRoom8	IF Patient_OperatingRoom = 8	
Patient	PreSurgeryBuffer			Patient	OperatingRoom9	IF Patient_OperatingRoom = 9	
Patient	OperatingRoom1	IF Patient_SurgicalTeam = 1 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_1 For A[1,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[1,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 2 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_2 For A[2,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[2,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 3 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_3 For A[3,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[3,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 4 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_4 For A[4,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[4,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 5 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_5 For A[5,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[5,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 6 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_6 For A[6,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[6,Patient_Scenario]	hr
Patient	OperatingRoom1	Else If Patient_SurgicalTeam = 7 Then					
Patient	OperatingRoom1	Use 1 Surgical_Team_7 For A[7,Patient_Scenario]	hr	And 1	OperatingRoom_1	For A[7,Patient_Scenario]	hr

302 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_1 For A[7, Patient_Scenario] hr
303
304 Else If Patient_SurgicalTeam = 8 Then
305 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_1 For A[8, Patient_Scenario] hr
306
307 Else If Patient_SurgicalTeam = 9 Then
308 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_1 For A[9, Patient_Scenario] hr
309
310 Else If Patient_SurgicalTeam = 10 Then
311 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_1 For A[10, Patient_Scenario] hr
312
313 Else If Patient_SurgicalTeam = 11 Then
314 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_1 For A[11, Patient_Scenario] hr
315
316 Else If Patient_SurgicalTeam = 12 Then
317 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_1 For A[12, Patient_Scenario] hr
318
319 Else If Patient_SurgicalTeam = 13 Then
320 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_1 For A[13, Patient_Scenario] hr
321
322 Else If Patient_SurgicalTeam = 14 Then
323 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_1 For A[14, Patient_Scenario] hr
324
325 Else If Patient_SurgicalTeam = 15 Then
326 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_1 For A[15, Patient_Scenario] hr
327
328 Else If Patient_SurgicalTeam = 16 Then
329 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_1 For A[16, Patient_Scenario] hr
330
331 Else If Patient_SurgicalTeam = 17 Then
332 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_1 For A[17, Patient_Scenario] hr
333
334 Else If Patient_SurgicalTeam = 18 Then
335 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_1 For A[18, Patient_Scenario] hr
336
337 Else If Patient_SurgicalTeam = 19 Then
338 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_1 For A[19, Patient_Scenario] hr
339
340 Else If Patient_SurgicalTeam = 20 Then
341 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_1 For A[20, Patient_Scenario] hr
342
343 Else If Patient_SurgicalTeam = 21 Then
344 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_1 For A[21, Patient_Scenario] hr
345
346 Else If Patient_SurgicalTeam = 22 Then
347 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_1 For A[22, Patient_Scenario] hr
348
349 Else If Patient_SurgicalTeam = 23 Then
350 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_1 For A[23, Patient_Scenario] hr
351
352 Else If Patient_SurgicalTeam = 24 Then
353 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_1 For A[24, Patient_Scenario] hr
354
355 Else If Patient_SurgicalTeam = 25 Then
356 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_1 For A[25, Patient_Scenario] hr
357
358 Else If Patient_SurgicalTeam = 26 Then
359 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_1 For A[26, Patient_Scenario] hr
360
361 Else If Patient_SurgicalTeam = 27 Then
362 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_1 For A[27, Patient_Scenario] hr


```

363 Else If Patient_SurgicalTeam = 28 Then
364 Use 1 Surgical_Team28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_1 For A[28, Patient_Scenario] hr
365
366 Else If Patient_SurgicalTeam = 29 Then
367 Use 1 Surgical_Team29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_1 For A[29, Patient_Scenario] hr
368
369 Else If Patient_SurgicalTeam = 30 Then
370 Use 1 Surgical_Team30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_1 For A[30, Patient_Scenario] hr
371
372 Use 1 Patient
373   PreRecoveryBuffer FIRST 1
374 Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_2 For A[1, Patient_Scenario] hr
375
376 Else If Patient_SurgicalTeam = 2 Then
377 Use 1 Surgical_Team2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_2 For A[2, Patient_Scenario] hr
378
379 Else If Patient_SurgicalTeam = 3 Then
380 Use 1 Surgical_Team3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_2 For A[3, Patient_Scenario] hr
381
382 Else If Patient_SurgicalTeam = 4 Then
383 Use 1 Surgical_Team4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_2 For A[4, Patient_Scenario] hr
384
385 Else If Patient_SurgicalTeam = 5 Then
386 Use 1 Surgical_Team5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_2 For A[5, Patient_Scenario] hr
387
388 Else If Patient_SurgicalTeam = 6 Then
389 Use 1 Surgical_Team6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_2 For A[6, Patient_Scenario] hr
390
391 Else If Patient_SurgicalTeam = 7 Then
392 Use 1 Surgical_Team7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_2 For A[7, Patient_Scenario] hr
393
394 Else If Patient_SurgicalTeam = 8 Then
395 Use 1 Surgical_Team8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_2 For A[8, Patient_Scenario] hr
396
397 Else If Patient_SurgicalTeam = 9 Then
398 Use 1 Surgical_Team9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_2 For A[9, Patient_Scenario] hr
399
400 Else If Patient_SurgicalTeam = 10 Then
401 Use 1 Surgical_Team10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_2 For A[10, Patient_Scenario] hr
402
403 Else If Patient_SurgicalTeam = 11 Then
404 Use 1 Surgical_Team11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_2 For A[11, Patient_Scenario] hr
405
406 Else If Patient_SurgicalTeam = 12 Then
407 Use 1 Surgical_Team12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_2 For A[12, Patient_Scenario] hr
408
409 Else If Patient_SurgicalTeam = 13 Then
410 Use 1 Surgical_Team13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_2 For A[13, Patient_Scenario] hr
411
412 Else If Patient_SurgicalTeam = 14 Then
413 Use 1 Surgical_Team14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_2 For A[14, Patient_Scenario] hr
414
415 Else If Patient_SurgicalTeam = 15 Then
416 Use 1 Surgical_Team15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_2 For A[15, Patient_Scenario] hr
417
418 Else If Patient_SurgicalTeam = 16 Then
419 Use 1 Surgical_Team16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_2 For A[16, Patient_Scenario] hr
420
421 Else If Patient_SurgicalTeam = 17 Then
422 Use 1 Surgical_Team17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_2 For A[17, Patient_Scenario] hr
423

```

```

424 Else If Patient_SurgicalTeam = 18 Then
425 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_2 For A[18, Patient_Scenario] hr
426
427 Else If Patient_SurgicalTeam = 19 Then
428 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_2 For A[19, Patient_Scenario] hr
429
430 Else If Patient_SurgicalTeam = 20 Then
431 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_2 For A[20, Patient_Scenario] hr
432
433 Else If Patient_SurgicalTeam = 21 Then
434 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_2 For A[21, Patient_Scenario] hr
435
436 Else If Patient_SurgicalTeam = 22 Then
437 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_2 For A[22, Patient_Scenario] hr
438
439 Else If Patient_SurgicalTeam = 23 Then
440 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_2 For A[23, Patient_Scenario] hr
441
442 Else If Patient_SurgicalTeam = 24 Then
443 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_2 For A[24, Patient_Scenario] hr
444
445 Else If Patient_SurgicalTeam = 25 Then
446 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_2 For A[25, Patient_Scenario] hr
447
448 Else If Patient_SurgicalTeam = 26 Then
449 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_2 For A[26, Patient_Scenario] hr
450
451 Else If Patient_SurgicalTeam = 27 Then
452 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_2 For A[27, Patient_Scenario] hr
453
454 Else If Patient_SurgicalTeam = 28 Then
455 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_2 For A[28, Patient_Scenario] hr
456
457 Else If Patient_SurgicalTeam = 29 Then
458 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_2 For A[29, Patient_Scenario] hr
459
460 Else If Patient_SurgicalTeam = 30 Then
461 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_2 For A[30, Patient_Scenario] hr
462
463 If Patient_SurgicalTeam = 1 Then
464     Patient
465     OperatingRoom3
466     Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_3 For A[1, Patient_Scenario] hr
467
468 Else If Patient_SurgicalTeam = 2 Then
469     Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_3 For A[2, Patient_Scenario] hr
470
471 Else If Patient_SurgicalTeam = 3 Then
472     Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_3 For A[3, Patient_Scenario] hr
473
474 Else If Patient_SurgicalTeam = 4 Then
475     Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_3 For A[4, Patient_Scenario] hr
476
477 Else If Patient_SurgicalTeam = 5 Then
478     Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_3 For A[5, Patient_Scenario] hr
479
480 Else If Patient_SurgicalTeam = 6 Then
481     Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_3 For A[6, Patient_Scenario] hr
482
483 Else If Patient_SurgicalTeam = 7 Then
484     Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_3 For A[7, Patient_Scenario] hr
485
486 Else If Patient_SurgicalTeam = 8 Then

```

485 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_3 For A[8, Patient_Scenario] hr
486
487 Else If Patient_SurgicalTeam = 9 Then
488 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_3 For A[9, Patient_Scenario] hr
489
490 Else If Patient_SurgicalTeam = 10 Then
491 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_3 For A[10, Patient_Scenario] hr
492
493 Else If Patient_SurgicalTeam = 11 Then
494 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_3 For A[11, Patient_Scenario] hr
495
496 Else If Patient_SurgicalTeam = 12 Then
497 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_3 For A[12, Patient_Scenario] hr
498
499 Else If Patient_SurgicalTeam = 13 Then
500 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_3 For A[13, Patient_Scenario] hr
501
502 Else If Patient_SurgicalTeam = 14 Then
503 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_3 For A[14, Patient_Scenario] hr
504
505 Else If Patient_SurgicalTeam = 15 Then
506 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_3 For A[15, Patient_Scenario] hr
507
508 Else If Patient_SurgicalTeam = 16 Then
509 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_3 For A[16, Patient_Scenario] hr
510
511 Else If Patient_SurgicalTeam = 17 Then
512 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_3 For A[17, Patient_Scenario] hr
513
514 Else If Patient_SurgicalTeam = 18 Then
515 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_3 For A[18, Patient_Scenario] hr
516
517 Else If Patient_SurgicalTeam = 19 Then
518 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_3 For A[19, Patient_Scenario] hr
519
520 Else If Patient_SurgicalTeam = 20 Then
521 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_3 For A[20, Patient_Scenario] hr
522
523 Else If Patient_SurgicalTeam = 21 Then
524 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_3 For A[21, Patient_Scenario] hr
525
526 Else If Patient_SurgicalTeam = 22 Then
527 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_3 For A[22, Patient_Scenario] hr
528
529 Else If Patient_SurgicalTeam = 23 Then
530 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_3 For A[23, Patient_Scenario] hr
531
532 Else If Patient_SurgicalTeam = 24 Then
533 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_3 For A[24, Patient_Scenario] hr
534
535 Else If Patient_SurgicalTeam = 25 Then
536 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_3 For A[25, Patient_Scenario] hr
537
538 Else If Patient_SurgicalTeam = 26 Then
539 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_3 For A[26, Patient_Scenario] hr
540
541 Else If Patient_SurgicalTeam = 27 Then
542 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_3 For A[27, Patient_Scenario] hr
543
544 Else If Patient_SurgicalTeam = 28 Then
545 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_3 For A[28, Patient_Scenario] hr

```

546 Else If Patient_SurgicalTeam = 29 Then
547 Use 1 Surgical_Team29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_3 For A[29, Patient_Scenario] hr
548
549
550 Else If Patient_SurgicalTeam = 30 Then
551 Use 1 Surgical_Team30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_3 For A[30, Patient_Scenario] hr
552 PreRecoveryBuffer FIRST 1
553 If Patient_SurgicalTeam = 1 Then
554 Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_4 For A[1, Patient_Scenario] hr
555
556 Else If Patient_SurgicalTeam = 2 Then
557 Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_4 For A[2, Patient_Scenario] hr
558
559 Else If Patient_SurgicalTeam = 3 Then
560 Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_4 For A[3, Patient_Scenario] hr
561
562 Else If Patient_SurgicalTeam = 4 Then
563 Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_4 For A[4, Patient_Scenario] hr
564
565 Else If Patient_SurgicalTeam = 5 Then
566 Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_4 For A[5, Patient_Scenario] hr
567
568 Else If Patient_SurgicalTeam = 6 Then
569 Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_4 For A[6, Patient_Scenario] hr
570
571 Else If Patient_SurgicalTeam = 7 Then
572 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_4 For A[7, Patient_Scenario] hr
573
574 Else If Patient_SurgicalTeam = 8 Then
575 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_4 For A[8, Patient_Scenario] hr
576
577 Else If Patient_SurgicalTeam = 9 Then
578 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_4 For A[9, Patient_Scenario] hr
579
580 Else If Patient_SurgicalTeam = 10 Then
581 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_4 For A[10, Patient_Scenario] hr
582
583 Else If Patient_SurgicalTeam = 11 Then
584 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_4 For A[11, Patient_Scenario] hr
585
586 Else If Patient_SurgicalTeam = 12 Then
587 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_4 For A[12, Patient_Scenario] hr
588
589 Else If Patient_SurgicalTeam = 13 Then
590 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_4 For A[13, Patient_Scenario] hr
591
592 Else If Patient_SurgicalTeam = 14 Then
593 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_4 For A[14, Patient_Scenario] hr
594
595 Else If Patient_SurgicalTeam = 15 Then
596 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_4 For A[15, Patient_Scenario] hr
597
598 Else If Patient_SurgicalTeam = 16 Then
599 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_4 For A[16, Patient_Scenario] hr
600
601 Else If Patient_SurgicalTeam = 17 Then
602 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_4 For A[17, Patient_Scenario] hr
603
604 Else If Patient_SurgicalTeam = 18 Then
605 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_4 For A[18, Patient_Scenario] hr
606

```

```

607 Else If Patient_SurgicalTeam = 19 Then
608 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_4 For A[19, Patient_Scenario] hr
609
610 Else If Patient_SurgicalTeam = 20 Then
611 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_4 For A[20, Patient_Scenario] hr
612
613 Else If Patient_SurgicalTeam = 21 Then
614 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_4 For A[21, Patient_Scenario] hr
615
616 Else If Patient_SurgicalTeam = 22 Then
617 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_4 For A[22, Patient_Scenario] hr
618
619 Else If Patient_SurgicalTeam = 23 Then
620 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_4 For A[23, Patient_Scenario] hr
621
622 Else If Patient_SurgicalTeam = 24 Then
623 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_4 For A[24, Patient_Scenario] hr
624
625 Else If Patient_SurgicalTeam = 25 Then
626 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_4 For A[25, Patient_Scenario] hr
627
628 Else If Patient_SurgicalTeam = 26 Then
629 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_4 For A[26, Patient_Scenario] hr
630
631 Else If Patient_SurgicalTeam = 27 Then
632 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_4 For A[27, Patient_Scenario] hr
633
634 Else If Patient_SurgicalTeam = 28 Then
635 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_4 For A[28, Patient_Scenario] hr
636
637 Else If Patient_SurgicalTeam = 29 Then
638 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_4 For A[29, Patient_Scenario] hr
639
640 Else If Patient_SurgicalTeam = 30 Then
641 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_4 For A[30, Patient_Scenario] hr
642
643 If Patient_SurgicalTeam = 1 Then
644     Patient
645     Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_5 For A[1, Patient_Scenario] hr
646
647 Else If Patient_SurgicalTeam = 2 Then
648     Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_5 For A[2, Patient_Scenario] hr
649
650 Else If Patient_SurgicalTeam = 3 Then
651     Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_5 For A[3, Patient_Scenario] hr
652
653 Else If Patient_SurgicalTeam = 4 Then
654     Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_5 For A[4, Patient_Scenario] hr
655
656 Else If Patient_SurgicalTeam = 5 Then
657     Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_5 For A[5, Patient_Scenario] hr
658
659 Else If Patient_SurgicalTeam = 6 Then
660     Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_5 For A[6, Patient_Scenario] hr
661
662 Else If Patient_SurgicalTeam = 7 Then
663     Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_5 For A[7, Patient_Scenario] hr
664
665 Else If Patient_SurgicalTeam = 8 Then
666     Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_5 For A[8, Patient_Scenario] hr
667
668 Else If Patient_SurgicalTeam = 9 Then

```

668 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_5 For A[9, Patient_Scenario] hr
669
670 Else If Patient_SurgicalTeam = 10 Then
671 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_5 For A[10, Patient_Scenario] hr
672
673 Else If Patient_SurgicalTeam = 11 Then
674 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_5 For A[11, Patient_Scenario] hr
675
676 Else If Patient_SurgicalTeam = 12 Then
677 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_5 For A[12, Patient_Scenario] hr
678
679 Else If Patient_SurgicalTeam = 13 Then
680 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_5 For A[13, Patient_Scenario] hr
681
682 Else If Patient_SurgicalTeam = 14 Then
683 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_5 For A[14, Patient_Scenario] hr
684
685 Else If Patient_SurgicalTeam = 15 Then
686 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_5 For A[15, Patient_Scenario] hr
687
688 Else If Patient_SurgicalTeam = 16 Then
689 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_5 For A[16, Patient_Scenario] hr
690
691 Else If Patient_SurgicalTeam = 17 Then
692 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_5 For A[17, Patient_Scenario] hr
693
694 Else If Patient_SurgicalTeam = 18 Then
695 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_5 For A[18, Patient_Scenario] hr
696
697 Else If Patient_SurgicalTeam = 19 Then
698 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_5 For A[19, Patient_Scenario] hr
699
700 Else If Patient_SurgicalTeam = 20 Then
701 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_5 For A[20, Patient_Scenario] hr
702
703 Else If Patient_SurgicalTeam = 21 Then
704 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_5 For A[21, Patient_Scenario] hr
705
706 Else If Patient_SurgicalTeam = 22 Then
707 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_5 For A[22, Patient_Scenario] hr
708
709 Else If Patient_SurgicalTeam = 23 Then
710 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_5 For A[23, Patient_Scenario] hr
711
712 Else If Patient_SurgicalTeam = 24 Then
713 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_5 For A[24, Patient_Scenario] hr
714
715 Else If Patient_SurgicalTeam = 25 Then
716 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_5 For A[25, Patient_Scenario] hr
717
718 Else If Patient_SurgicalTeam = 26 Then
719 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_5 For A[26, Patient_Scenario] hr
720
721 Else If Patient_SurgicalTeam = 27 Then
722 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_5 For A[27, Patient_Scenario] hr
723
724 Else If Patient_SurgicalTeam = 28 Then
725 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_5 For A[28, Patient_Scenario] hr
726
727 Else If Patient_SurgicalTeam = 29 Then
728 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_5 For A[29, Patient_Scenario] hr

```

729 Else If Patient_SurgicalTeam = 30 Then
730 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_5 For A[30, Patient_Scenario] hr
731 PreRecoveryBuffer FIRST 1
732 1 Patient
733 If Patient_SurgicalTeam = 1 Then
734 Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_6 For A[1, Patient_Scenario] hr
735
736 Else If Patient_SurgicalTeam = 2 Then
737 Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_6 For A[2, Patient_Scenario] hr
738
739 Else If Patient_SurgicalTeam = 3 Then
740 Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_6 For A[3, Patient_Scenario] hr
741
742 Else If Patient_SurgicalTeam = 4 Then
743 Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_6 For A[4, Patient_Scenario] hr
744
745 Else If Patient_SurgicalTeam = 5 Then
746 Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_6 For A[5, Patient_Scenario] hr
747
748 Else If Patient_SurgicalTeam = 6 Then
749 Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_6 For A[6, Patient_Scenario] hr
750
751 Else If Patient_SurgicalTeam = 7 Then
752 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_6 For A[7, Patient_Scenario] hr
753
754 Else If Patient_SurgicalTeam = 8 Then
755 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_6 For A[8, Patient_Scenario] hr
756
757 Else If Patient_SurgicalTeam = 9 Then
758 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_6 For A[9, Patient_Scenario] hr
759
760 Else If Patient_SurgicalTeam = 10 Then
761 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_6 For A[10, Patient_Scenario] hr
762
763 Else If Patient_SurgicalTeam = 11 Then
764 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_6 For A[11, Patient_Scenario] hr
765
766 Else If Patient_SurgicalTeam = 12 Then
767 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_6 For A[12, Patient_Scenario] hr
768
769 Else If Patient_SurgicalTeam = 13 Then
770 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_6 For A[13, Patient_Scenario] hr
771
772 Else If Patient_SurgicalTeam = 14 Then
773 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_6 For A[14, Patient_Scenario] hr
774
775 Else If Patient_SurgicalTeam = 15 Then
776 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_6 For A[15, Patient_Scenario] hr
777
778 Else If Patient_SurgicalTeam = 16 Then
779 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_6 For A[16, Patient_Scenario] hr
780
781 Else If Patient_SurgicalTeam = 17 Then
782 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_6 For A[17, Patient_Scenario] hr
783
784 Else If Patient_SurgicalTeam = 18 Then
785 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_6 For A[18, Patient_Scenario] hr
786
787 Else If Patient_SurgicalTeam = 19 Then
788 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_6 For A[19, Patient_Scenario] hr
789

```

```

790 Else If Patient_SurgicalTeam = 20 Then
791 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_6 For A[20, Patient_Scenario] hr
792
793 Else If Patient_SurgicalTeam = 21 Then
794 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_6 For A[21, Patient_Scenario] hr
795
796 Else If Patient_SurgicalTeam = 22 Then
797 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_6 For A[22, Patient_Scenario] hr
798
799 Else If Patient_SurgicalTeam = 23 Then
800 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_6 For A[23, Patient_Scenario] hr
801
802 Else If Patient_SurgicalTeam = 24 Then
803 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_6 For A[24, Patient_Scenario] hr
804
805 Else If Patient_SurgicalTeam = 25 Then
806 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_6 For A[25, Patient_Scenario] hr
807
808 Else If Patient_SurgicalTeam = 26 Then
809 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_6 For A[26, Patient_Scenario] hr
810
811 Else If Patient_SurgicalTeam = 27 Then
812 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_6 For A[27, Patient_Scenario] hr
813
814 Else If Patient_SurgicalTeam = 28 Then
815 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_6 For A[28, Patient_Scenario] hr
816
817 Else If Patient_SurgicalTeam = 29 Then
818 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_6 For A[29, Patient_Scenario] hr
819
820 Else If Patient_SurgicalTeam = 30 Then
821 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_6 For A[30, Patient_Scenario] hr
822
823 Patient
      OperatingRoom7
      If Patient_SurgicalTeam = 1 Then
824 Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_7 For A[1, Patient_Scenario] hr
825
826 Else If Patient_SurgicalTeam = 2 Then
827 Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_7 For A[2, Patient_Scenario] hr
828
829 Else If Patient_SurgicalTeam = 3 Then
830 Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_7 For A[3, Patient_Scenario] hr
831
832 Else If Patient_SurgicalTeam = 4 Then
833 Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_7 For A[4, Patient_Scenario] hr
834
835 Else If Patient_SurgicalTeam = 5 Then
836 Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_7 For A[5, Patient_Scenario] hr
837
838 Else If Patient_SurgicalTeam = 6 Then
839 Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_7 For A[6, Patient_Scenario] hr
840
841 Else If Patient_SurgicalTeam = 7 Then
842 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_7 For A[7, Patient_Scenario] hr
843
844 Else If Patient_SurgicalTeam = 8 Then
845 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_7 For A[8, Patient_Scenario] hr
846
847 Else If Patient_SurgicalTeam = 9 Then
848 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_7 For A[9, Patient_Scenario] hr
849
850 Else If Patient_SurgicalTeam = 10 Then

```


851 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_7 For A[10, Patient_Scenario] hr
852
853 Else If Patient_SurgicalTeam = 11 Then
854 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_7 For A[11, Patient_Scenario] hr
855
856 Else If Patient_SurgicalTeam = 12 Then
857 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_7 For A[12, Patient_Scenario] hr
858
859 Else If Patient_SurgicalTeam = 13 Then
860 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_7 For A[13, Patient_Scenario] hr
861
862 Else If Patient_SurgicalTeam = 14 Then
863 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_7 For A[14, Patient_Scenario] hr
864
865 Else If Patient_SurgicalTeam = 15 Then
866 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_7 For A[15, Patient_Scenario] hr
867
868 Else If Patient_SurgicalTeam = 16 Then
869 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_7 For A[16, Patient_Scenario] hr
870
871 Else If Patient_SurgicalTeam = 17 Then
872 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_7 For A[17, Patient_Scenario] hr
873
874 Else If Patient_SurgicalTeam = 18 Then
875 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_7 For A[18, Patient_Scenario] hr
876
877 Else If Patient_SurgicalTeam = 19 Then
878 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_7 For A[19, Patient_Scenario] hr
879
880 Else If Patient_SurgicalTeam = 20 Then
881 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_7 For A[20, Patient_Scenario] hr
882
883 Else If Patient_SurgicalTeam = 21 Then
884 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_7 For A[21, Patient_Scenario] hr
885
886 Else If Patient_SurgicalTeam = 22 Then
887 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_7 For A[22, Patient_Scenario] hr
888
889 Else If Patient_SurgicalTeam = 23 Then
890 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_7 For A[23, Patient_Scenario] hr
891
892 Else If Patient_SurgicalTeam = 24 Then
893 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_7 For A[24, Patient_Scenario] hr
894
895 Else If Patient_SurgicalTeam = 25 Then
896 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_7 For A[25, Patient_Scenario] hr
897
898 Else If Patient_SurgicalTeam = 26 Then
899 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_7 For A[26, Patient_Scenario] hr
900
901 Else If Patient_SurgicalTeam = 27 Then
902 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_7 For A[27, Patient_Scenario] hr
903
904 Else If Patient_SurgicalTeam = 28 Then
905 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_7 For A[28, Patient_Scenario] hr
906
907 Else If Patient_SurgicalTeam = 29 Then
908 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_7 For A[29, Patient_Scenario] hr
909
910 Else If Patient_SurgicalTeam = 30 Then
911 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_7 For A[30, Patient_Scenario] hr

```

912 Patient
913 OperatingRoom8
914 Patient
915 PreRecoveryBuffer FIRST 1
916 If Patient_SurgicalTeam = 1 Then
917 Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_8 For A[1, Patient_Scenario] hr
918 Else If Patient_SurgicalTeam = 2 Then
919 Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_8 For A[2, Patient_Scenario] hr
920 Else If Patient_SurgicalTeam = 3 Then
921 Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_8 For A[3, Patient_Scenario] hr
922 Else If Patient_SurgicalTeam = 4 Then
923 Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_8 For A[4, Patient_Scenario] hr
924 Else If Patient_SurgicalTeam = 5 Then
925 Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_8 For A[5, Patient_Scenario] hr
926 Else If Patient_SurgicalTeam = 6 Then
927 Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_8 For A[6, Patient_Scenario] hr
928 Else If Patient_SurgicalTeam = 7 Then
929 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_8 For A[7, Patient_Scenario] hr
930 Else If Patient_SurgicalTeam = 8 Then
931 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_8 For A[8, Patient_Scenario] hr
932 Else If Patient_SurgicalTeam = 9 Then
933 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_8 For A[9, Patient_Scenario] hr
934 Else If Patient_SurgicalTeam = 10 Then
935 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_8 For A[10, Patient_Scenario] hr
936 Else If Patient_SurgicalTeam = 11 Then
937 Use 1 Surgical_Team_11 For A[11, Patient_Scenario] hr And 1 OperatingRoom_8 For A[11, Patient_Scenario] hr
938 Else If Patient_SurgicalTeam = 12 Then
939 Use 1 Surgical_Team_12 For A[12, Patient_Scenario] hr And 1 OperatingRoom_8 For A[12, Patient_Scenario] hr
940 Else If Patient_SurgicalTeam = 13 Then
941 Use 1 Surgical_Team_13 For A[13, Patient_Scenario] hr And 1 OperatingRoom_8 For A[13, Patient_Scenario] hr
942 Else If Patient_SurgicalTeam = 14 Then
943 Use 1 Surgical_Team_14 For A[14, Patient_Scenario] hr And 1 OperatingRoom_8 For A[14, Patient_Scenario] hr
944 Else If Patient_SurgicalTeam = 15 Then
945 Use 1 Surgical_Team_15 For A[15, Patient_Scenario] hr And 1 OperatingRoom_8 For A[15, Patient_Scenario] hr
946 Else If Patient_SurgicalTeam = 16 Then
947 Use 1 Surgical_Team_16 For A[16, Patient_Scenario] hr And 1 OperatingRoom_8 For A[16, Patient_Scenario] hr
948 Else If Patient_SurgicalTeam = 17 Then
949 Use 1 Surgical_Team_17 For A[17, Patient_Scenario] hr And 1 OperatingRoom_8 For A[17, Patient_Scenario] hr
950 Else If Patient_SurgicalTeam = 18 Then
951 Use 1 Surgical_Team_18 For A[18, Patient_Scenario] hr And 1 OperatingRoom_8 For A[18, Patient_Scenario] hr
952 Else If Patient_SurgicalTeam = 19 Then
953 Use 1 Surgical_Team_19 For A[19, Patient_Scenario] hr And 1 OperatingRoom_8 For A[19, Patient_Scenario] hr
954 Else If Patient_SurgicalTeam = 20 Then
955 Use 1 Surgical_Team_20 For A[20, Patient_Scenario] hr And 1 OperatingRoom_8 For A[20, Patient_Scenario] hr
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972

```

```

973 Else If Patient_SurgicalTeam = 21 Then
974 Use 1 Surgical_Team_21 For A[21, Patient_Scenario] hr And 1 OperatingRoom_8 For A[21, Patient_Scenario] hr
975
976 Else If Patient_SurgicalTeam = 22 Then
977 Use 1 Surgical_Team_22 For A[22, Patient_Scenario] hr And 1 OperatingRoom_8 For A[22, Patient_Scenario] hr
978
979 Else If Patient_SurgicalTeam = 23 Then
980 Use 1 Surgical_Team_23 For A[23, Patient_Scenario] hr And 1 OperatingRoom_8 For A[23, Patient_Scenario] hr
981
982 Else If Patient_SurgicalTeam = 24 Then
983 Use 1 Surgical_Team_24 For A[24, Patient_Scenario] hr And 1 OperatingRoom_8 For A[24, Patient_Scenario] hr
984
985 Else If Patient_SurgicalTeam = 25 Then
986 Use 1 Surgical_Team_25 For A[25, Patient_Scenario] hr And 1 OperatingRoom_8 For A[25, Patient_Scenario] hr
987
988 Else If Patient_SurgicalTeam = 26 Then
989 Use 1 Surgical_Team_26 For A[26, Patient_Scenario] hr And 1 OperatingRoom_8 For A[26, Patient_Scenario] hr
990
991 Else If Patient_SurgicalTeam = 27 Then
992 Use 1 Surgical_Team_27 For A[27, Patient_Scenario] hr And 1 OperatingRoom_8 For A[27, Patient_Scenario] hr
993
994 Else If Patient_SurgicalTeam = 28 Then
995 Use 1 Surgical_Team_28 For A[28, Patient_Scenario] hr And 1 OperatingRoom_8 For A[28, Patient_Scenario] hr
996
997 Else If Patient_SurgicalTeam = 29 Then
998 Use 1 Surgical_Team_29 For A[29, Patient_Scenario] hr And 1 OperatingRoom_8 For A[29, Patient_Scenario] hr
999
1000 Else If Patient_SurgicalTeam = 30 Then
1001 Use 1 Surgical_Team_30 For A[30, Patient_Scenario] hr And 1 OperatingRoom_8 For A[30, Patient_Scenario] hr
1002
1003 If Patient_SurgicalTeam = 1 Then
1004     Patient
1005     PreRecoveryBuffer FIRST 1
1006     Use 1 Surgical_Team_1 For A[1, Patient_Scenario] hr And 1 OperatingRoom_9 For A[1, Patient_Scenario] hr
1007
1008 Else If Patient_SurgicalTeam = 2 Then
1009 Use 1 Surgical_Team_2 For A[2, Patient_Scenario] hr And 1 OperatingRoom_9 For A[2, Patient_Scenario] hr
1010
1011 Else If Patient_SurgicalTeam = 3 Then
1012 Use 1 Surgical_Team_3 For A[3, Patient_Scenario] hr And 1 OperatingRoom_9 For A[3, Patient_Scenario] hr
1013
1014 Else If Patient_SurgicalTeam = 4 Then
1015 Use 1 Surgical_Team_4 For A[4, Patient_Scenario] hr And 1 OperatingRoom_9 For A[4, Patient_Scenario] hr
1016
1017 Else If Patient_SurgicalTeam = 5 Then
1018 Use 1 Surgical_Team_5 For A[5, Patient_Scenario] hr And 1 OperatingRoom_9 For A[5, Patient_Scenario] hr
1019
1020 Else If Patient_SurgicalTeam = 6 Then
1021 Use 1 Surgical_Team_6 For A[6, Patient_Scenario] hr And 1 OperatingRoom_9 For A[6, Patient_Scenario] hr
1022
1023 Else If Patient_SurgicalTeam = 7 Then
1024 Use 1 Surgical_Team_7 For A[7, Patient_Scenario] hr And 1 OperatingRoom_9 For A[7, Patient_Scenario] hr
1025
1026 Else If Patient_SurgicalTeam = 8 Then
1027 Use 1 Surgical_Team_8 For A[8, Patient_Scenario] hr And 1 OperatingRoom_9 For A[8, Patient_Scenario] hr
1028
1029 Else If Patient_SurgicalTeam = 9 Then
1030 Use 1 Surgical_Team_9 For A[9, Patient_Scenario] hr And 1 OperatingRoom_9 For A[9, Patient_Scenario] hr
1031
1032 Else If Patient_SurgicalTeam = 10 Then
1033 Use 1 Surgical_Team_10 For A[10, Patient_Scenario] hr And 1 OperatingRoom_9 For A[10, Patient_Scenario] hr
1034
1035 Else If Patient_SurgicalTeam = 11 Then

```



```

1095 Patient RecoveryWardBuffer 1 Patient RecoveryWard FIRST 1
1096 Patient RecoveryWard If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 1) Then
1097 Patient RecoveryWard Wait sli1() day
1098
1099 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 2) Then
1100 Patient RecoveryWard Wait sli2() day
1101
1102 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 3) Then
1103 Patient RecoveryWard Wait sli3() day
1104
1105 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 4) Then
1106 Patient RecoveryWard Wait sli4() day
1107
1108 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 5) Then
1109 Patient RecoveryWard Wait sli5() day
1110
1111 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 6) Then
1112 Patient RecoveryWard Wait sli6() day
1113
1114 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 7) Then
1115 Patient RecoveryWard Wait sli7() day
1116
1117 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 8) Then
1118 Patient RecoveryWard Wait sli8() day
1119
1120 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 9) Then
1121 Patient RecoveryWard Wait sli9() day
1122
1123 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 10) Then
1124 Patient RecoveryWard Wait sli10() day
1125
1126 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 11) Then
1127 Patient RecoveryWard Wait sli11() day
1128
1129 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 12) Then
1130 Patient RecoveryWard Wait sli12() day
1131
1132 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 13) Then
1133 Patient RecoveryWard Wait sli13() day
1134
1135 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 14) Then
1136 Patient RecoveryWard Wait sli14() day
1137
1138 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 15) Then
1139 Patient RecoveryWard Wait sli15() day
1140
1141 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 16) Then
1142 Patient RecoveryWard Wait sli16() day
1143
1144 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 17) Then
1145 Patient RecoveryWard Wait sli17() day
1146
1147 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 18) Then
1148 Patient RecoveryWard Wait sli18() day
1149
1150 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 19) Then
1151 Patient RecoveryWard Wait sli19() day
1152
1153 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 20) Then
1154 Patient RecoveryWard Wait sli20() day
1155

```

```

1156 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 21) Then
1157     Wait s1i21() day
1158
1159 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 22) Then
1160     Wait s1i22() day
1161
1162 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 23) Then
1163     Wait s1i23() day
1164
1165 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 24) Then
1166     Wait s1i24() day
1167
1168 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 25) Then
1169     Wait s1i25() day
1170
1171 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 26) Then
1172     Wait s1i26() day
1173
1174 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 27) Then
1175     Wait s1i27() day
1176
1177 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 28) Then
1178     Wait s1i28() day
1179
1180 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 29) Then
1181     Wait s1i29() day
1182
1183 If (Patient_Scenario = 1) And (Patient_SurgicalTeam = 30) Then
1184     Wait s1i30() day
1185
1186 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 1) Then
1187     Wait s2i1() day
1188
1189 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 2) Then
1190     Wait s2i2() day
1191
1192 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 3) Then
1193     Wait s2i3() day
1194
1195 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 4) Then
1196     Wait s2i4() day
1197
1198 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 5) Then
1199     Wait s2i5() day
1200
1201 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 6) Then
1202     Wait s2i6() day
1203
1204 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 7) Then
1205     Wait s2i7() day
1206
1207 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 8) Then
1208     Wait s2i8() day
1209
1210 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 9) Then
1211     Wait s2i9() day
1212
1213 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 10) Then
1214     Wait s2i10() day
1215
1216 If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 11) Then

```

```

1217     Wait s2i11() day
1218
1219     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 12) Then
1220         Wait s2i12() day
1221
1222     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 13) Then
1223         Wait s2i13() day
1224
1225     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 14) Then
1226         Wait s2i14() day
1227
1228     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 15) Then
1229         Wait s2i15() day
1230
1231     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 16) Then
1232         Wait s2i16() day
1233
1234     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 17) Then
1235         Wait s2i17() day
1236
1237     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 18) Then
1238         Wait s2i18() day
1239
1240     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 19) Then
1241         Wait s2i19() day
1242
1243     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 20) Then
1244         Wait s2i20() day
1245
1246     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 21) Then
1247         Wait s2i21() day
1248
1249     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 22) Then
1250         Wait s2i22() day
1251
1252     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 23) Then
1253         Wait s2i23() day
1254
1255     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 24) Then
1256         Wait s2i24() day
1257
1258     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 25) Then
1259         Wait s2i25() day
1260
1261     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 26) Then
1262         Wait s2i26() day
1263
1264     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 27) Then
1265         Wait s2i27() day
1266
1267     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 28) Then
1268         Wait s2i28() day
1269
1270     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 29) Then
1271         Wait s2i29() day
1272
1273     If (Patient_Scenario = 2) And (Patient_SurgicalTeam = 30) Then
1274         Wait s2i30() day
1275
1276     If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 1) Then
1277         Wait s3i1() day

```

```

1278 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 2) Then
1279     Wait s3i2() day
1280
1281
1282 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 3) Then
1283     Wait s3i3() day
1284
1285 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 4) Then
1286     Wait s3i4() day
1287
1288 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 5) Then
1289     Wait s3i5() day
1290
1291 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 6) Then
1292     Wait s3i6() day
1293
1294 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 7) Then
1295     Wait s3i7() day
1296
1297 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 8) Then
1298     Wait s3i8() day
1299
1300 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 9) Then
1301     Wait s3i9() day
1302
1303 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 10) Then
1304     Wait s3i10() day
1305
1306 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 11) Then
1307     Wait s3i11() day
1308
1309 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 12) Then
1310     Wait s3i12() day
1311
1312 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 13) Then
1313     Wait s3i13() day
1314
1315 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 14) Then
1316     Wait s3i14() day
1317
1318 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 15) Then
1319     Wait s3i15() day
1320
1321 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 16) Then
1322     Wait s3i16() day
1323
1324 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 17) Then
1325     Wait s3i17() day
1326
1327 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 18) Then
1328     Wait s3i18() day
1329
1330 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 19) Then
1331     Wait s3i19() day
1332
1333 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 20) Then
1334     Wait s3i20() day
1335
1336 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 21) Then
1337     Wait s3i21() day
1338

```



```

1339 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 22) Then
1340     Wait s3i22() day
1341
1342 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 23) Then
1343     Wait s3i23() day
1344
1345 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 24) Then
1346     Wait s3i24() day
1347
1348 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 25) Then
1349     Wait s3i25() day
1350
1351 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 26) Then
1352     Wait s3i26() day
1353
1354 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 27) Then
1355     Wait s3i27() day
1356
1357 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 28) Then
1358     Wait s3i28() day
1359
1360 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 29) Then
1361     Wait s3i29() day
1362
1363 If (Patient_Scenario = 3) And (Patient_SurgicalTeam = 30) Then
1364     Wait s3i30() day
1365
1366 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 1) Then
1367     Wait s4i1() day
1368
1369 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 2) Then
1370     Wait s4i2() day
1371
1372 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 3) Then
1373     Wait s4i3() day
1374
1375 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 4) Then
1376     Wait s4i4() day
1377
1378 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 5) Then
1379     Wait s4i5() day
1380
1381 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 6) Then
1382     Wait s4i6() day
1383
1384 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 7) Then
1385     Wait s4i7() day
1386
1387 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 8) Then
1388     Wait s4i8() day
1389
1390 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 9) Then
1391     Wait s4i9() day
1392
1393 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 10) Then
1394     Wait s4i10() day
1395
1396 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 11) Then
1397     Wait s4i11() day
1398
1399 If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 12) Then

```

```

1400      Wait s4i12() day
1401
1402      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 13) Then
1403          Wait s4i13() day
1404
1405      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 14) Then
1406          Wait s4i14() day
1407
1408      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 15) Then
1409          Wait s4i15() day
1410
1411      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 16) Then
1412          Wait s4i16() day
1413
1414      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 17) Then
1415          Wait s4i17() day
1416
1417      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 18) Then
1418          Wait s4i18() day
1419
1420      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 19) Then
1421          Wait s4i19() day
1422
1423      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 20) Then
1424          Wait s4i20() day
1425
1426      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 21) Then
1427          Wait s4i21() day
1428
1429      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 22) Then
1430          Wait s4i22() day
1431
1432      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 23) Then
1433          Wait s4i23() day
1434
1435      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 24) Then
1436          Wait s4i24() day
1437
1438      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 25) Then
1439          Wait s4i25() day
1440
1441      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 26) Then
1442          Wait s4i26() day
1443
1444      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 27) Then
1445          Wait s4i27() day
1446
1447      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 28) Then
1448          Wait s4i28() day
1449
1450      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 29) Then
1451          Wait s4i29() day
1452
1453      If (Patient_Scenario = 4) And (Patient_SurgicalTeam = 30) Then
1454          Wait s4i30() day
1455
1456      If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 1) Then
1457          Wait s5i1() day
1458
1459      If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 2) Then
1460          Wait s5i2() day

```

1461 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 3) Then
1462 Wait s5i3() day
1463
1464 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 4) Then
1465 Wait s5i4() day
1466
1467 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 5) Then
1468 Wait s5i5() day
1469
1470 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 6) Then
1471 Wait s5i6() day
1472
1473 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 7) Then
1474 Wait s5i7() day
1475
1476 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 8) Then
1477 Wait s5i8() day
1478
1479 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 9) Then
1480 Wait s5i9() day
1481
1482 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 10) Then
1483 Wait s5i10() day
1484
1485 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 11) Then
1486 Wait s5i11() day
1487
1488 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 12) Then
1489 Wait s5i12() day
1490
1491 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 13) Then
1492 Wait s5i13() day
1493
1494 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 14) Then
1495 Wait s5i14() day
1496
1497 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 15) Then
1498 Wait s5i15() day
1499
1500 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 16) Then
1501 Wait s5i16() day
1502
1503 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 17) Then
1504 Wait s5i17() day
1505
1506 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 18) Then
1507 Wait s5i18() day
1508
1509 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 19) Then
1510 Wait s5i19() day
1511
1512 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 20) Then
1513 Wait s5i20() day
1514
1515 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 21) Then
1516 Wait s5i21() day
1517
1518 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 22) Then
1519 Wait s5i22() day
1520
1521

```

1522 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 23) Then
1523     Wait s5i23() day
1524
1525 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 24) Then
1526     Wait s5i24() day
1527
1528 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 25) Then
1529     Wait s5i25() day
1530
1531 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 26) Then
1532     Wait s5i26() day
1533
1534 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 27) Then
1535     Wait s5i27() day
1536
1537 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 28) Then
1538     Wait s5i28() day
1539
1540 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 29) Then
1541     Wait s5i29() day
1542
1543 If (Patient_Scenario = 5) And (Patient_SurgicalTeam = 30) Then
1544     Wait s5i30() day
1545
1546 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 1) Then
1547     Wait s6i1() day
1548
1549 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 2) Then
1550     Wait s6i2() day
1551
1552 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 3) Then
1553     Wait s6i3() day
1554
1555 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 4) Then
1556     Wait s6i4() day
1557
1558 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 5) Then
1559     Wait s6i5() day
1560
1561 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 6) Then
1562     Wait s6i6() day
1563
1564 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 7) Then
1565     Wait s6i7() day
1566
1567 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 8) Then
1568     Wait s6i8() day
1569
1570 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 9) Then
1571     Wait s6i9() day
1572
1573 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 10) Then
1574     Wait s6i10() day
1575
1576 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 11) Then
1577     Wait s6i11() day
1578
1579 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 12) Then
1580     Wait s6i12() day
1581
1582 If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 13) Then

```

```

1583     Wait s6i13() day
1584
1585     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 14) Then
1586         Wait s6i14() day
1587
1588     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 15) Then
1589         Wait s6i15() day
1590
1591     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 16) Then
1592         Wait s6i16() day
1593
1594     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 17) Then
1595         Wait s6i17() day
1596
1597     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 18) Then
1598         Wait s6i18() day
1599
1600     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 19) Then
1601         Wait s6i19() day
1602
1603     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 20) Then
1604         Wait s6i20() day
1605
1606     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 21) Then
1607         Wait s6i21() day
1608
1609     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 22) Then
1610         Wait s6i22() day
1611
1612     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 23) Then
1613         Wait s6i23() day
1614
1615     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 24) Then
1616         Wait s6i24() day
1617
1618     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 25) Then
1619         Wait s6i25() day
1620
1621     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 26) Then
1622         Wait s6i26() day
1623
1624     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 27) Then
1625         Wait s6i27() day
1626
1627     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 28) Then
1628         Wait s6i28() day
1629
1630     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 29) Then
1631         Wait s6i29() day
1632
1633     If (Patient_Scenario = 6) And (Patient_SurgicalTeam = 30) Then
1634         Wait s6i30() day
1635
1636     If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 1) Then
1637         Wait s7i1() day
1638
1639     If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 2) Then
1640         Wait s7i2() day
1641
1642     If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 3) Then
1643         Wait s7i3() day

```

1644 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 4) Then
1645 Wait s7i4() day
1646
1647 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 5) Then
1648 Wait s7i5() day
1649
1650 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 6) Then
1651 Wait s7i6() day
1652
1653 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 7) Then
1654 Wait s7i7() day
1655
1656 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 8) Then
1657 Wait s7i8() day
1658
1659 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 9) Then
1660 Wait s7i9() day
1661
1662 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 10) Then
1663 Wait s7i10() day
1664
1665 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 11) Then
1666 Wait s7i11() day
1667
1668 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 12) Then
1669 Wait s7i12() day
1670
1671 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 13) Then
1672 Wait s7i13() day
1673
1674 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 14) Then
1675 Wait s7i14() day
1676
1677 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 15) Then
1678 Wait s7i15() day
1679
1680 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 16) Then
1681 Wait s7i16() day
1682
1683 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 17) Then
1684 Wait s7i17() day
1685
1686 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 18) Then
1687 Wait s7i18() day
1688
1689 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 19) Then
1690 Wait s7i19() day
1691
1692 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 20) Then
1693 Wait s7i20() day
1694
1695 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 21) Then
1696 Wait s7i21() day
1697
1698 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 22) Then
1699 Wait s7i22() day
1700
1701 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 23) Then
1702 Wait s7i23() day
1703
1704

```

1705 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 24) Then
1706     Wait s7124() day
1707
1708 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 25) Then
1709     Wait s7125() day
1710
1711 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 26) Then
1712     Wait s7126() day
1713
1714 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 27) Then
1715     Wait s7127() day
1716
1717 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 28) Then
1718     Wait s7128() day
1719
1720 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 29) Then
1721     Wait s7129() day
1722
1723 If (Patient_Scenario = 7) And (Patient_SurgicalTeam = 30) Then
1724     Wait s7i30() day
1725
1726 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 1) Then
1727     Wait s811() day
1728
1729 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 2) Then
1730     Wait s812() day
1731
1732 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 3) Then
1733     Wait s813() day
1734
1735 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 4) Then
1736     Wait s814() day
1737
1738 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 5) Then
1739     Wait s815() day
1740
1741 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 6) Then
1742     Wait s816() day
1743
1744 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 7) Then
1745     Wait s817() day
1746
1747 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 8) Then
1748     Wait s818() day
1749
1750 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 9) Then
1751     Wait s819() day
1752
1753 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 10) Then
1754     Wait s8i10() day
1755
1756 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 11) Then
1757     Wait s811() day
1758
1759 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 12) Then
1760     Wait s8i12() day
1761
1762 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 13) Then
1763     Wait s8i13() day
1764
1765 If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 14) Then

```

```

1766      Wait s8i14() day
1767
1768      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 15) Then
1769          Wait s8i15() day
1770
1771      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 16) Then
1772          Wait s8i16() day
1773
1774      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 17) Then
1775          Wait s8i17() day
1776
1777      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 18) Then
1778          Wait s8i18() day
1779
1780      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 19) Then
1781          Wait s8i19() day
1782
1783      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 20) Then
1784          Wait s8i20() day
1785
1786      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 21) Then
1787          Wait s8i21() day
1788
1789      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 22) Then
1790          Wait s8i22() day
1791
1792      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 23) Then
1793          Wait s8i23() day
1794
1795      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 24) Then
1796          Wait s8i24() day
1797
1798      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 25) Then
1799          Wait s8i25() day
1800
1801      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 26) Then
1802          Wait s8i26() day
1803
1804      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 27) Then
1805          Wait s8i27() day
1806
1807      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 28) Then
1808          Wait s8i28() day
1809
1810      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 29) Then
1811          Wait s8i29() day
1812
1813      If (Patient_Scenario = 8) And (Patient_SurgicalTeam = 30) Then
1814          Wait s8i30() day
1815
1816      If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 1) Then
1817          Wait s9i1() day
1818
1819      If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 2) Then
1820          Wait s9i2() day
1821
1822      If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 3) Then
1823          Wait s9i3() day
1824
1825      If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 4) Then
1826          Wait s9i4() day

```


1827 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 5) Then
1828 Wait s9i5() day
1829
1830 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 6) Then
1831 Wait s9i6() day
1832
1833 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 7) Then
1834 Wait s9i7() day
1835
1836 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 8) Then
1837 Wait s9i8() day
1838
1839 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 9) Then
1840 Wait s9i9() day
1841
1842 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 10) Then
1843 Wait s9i10() day
1844
1845 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 11) Then
1846 Wait s9i11() day
1847
1848 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 12) Then
1849 Wait s9i12() day
1850
1851 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 13) Then
1852 Wait s9i13() day
1853
1854 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 14) Then
1855 Wait s9i14() day
1856
1857 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 15) Then
1858 Wait s9i15() day
1859
1860 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 16) Then
1861 Wait s9i16() day
1862
1863 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 17) Then
1864 Wait s9i17() day
1865
1866 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 18) Then
1867 Wait s9i18() day
1868
1869 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 19) Then
1870 Wait s9i19() day
1871
1872 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 20) Then
1873 Wait s9i20() day
1874
1875 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 21) Then
1876 Wait s9i21() day
1877
1878 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 22) Then
1879 Wait s9i22() day
1880
1881 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 23) Then
1882 Wait s9i23() day
1883
1884 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 24) Then
1885 Wait s9i24() day
1886
1887

1888 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 25) Then
1889 Wait s9i25() day
1890
1891 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 26) Then
1892 Wait s9i26() day
1893
1894 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 27) Then
1895 Wait s9i27() day
1896
1897 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 28) Then
1898 Wait s9i28() day
1899
1900 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 29) Then
1901 Wait s9i29() day
1902
1903 If (Patient_Scenario = 9) And (Patient_SurgicalTeam = 30) Then
1904 Wait s9i30() day
1905
1906 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 1) Then
1907 Wait s10i1() day
1908
1909 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 2) Then
1910 Wait s10i2() day
1911
1912 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 3) Then
1913 Wait s10i3() day
1914
1915 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 4) Then
1916 Wait s10i4() day
1917
1918 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 5) Then
1919 Wait s10i5() day
1920
1921 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 6) Then
1922 Wait s10i6() day
1923
1924 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 7) Then
1925 Wait s10i7() day
1926
1927 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 8) Then
1928 Wait s10i8() day
1929
1930 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 9) Then
1931 Wait s10i9() day
1932
1933 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 10) Then
1934 Wait s10i10() day
1935
1936 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 11) Then
1937 Wait s10i11() day
1938
1939 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 12) Then
1940 Wait s10i12() day
1941
1942 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 13) Then
1943 Wait s10i13() day
1944
1945 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 14) Then
1946 Wait s10i14() day
1947
1948 If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 15) Then

```

1949      Wait s10i15() day
1950
1951      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 16) Then
1952          Wait s10i16() day
1953
1954      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 17) Then
1955          Wait s10i17() day
1956
1957      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 18) Then
1958          Wait s10i18() day
1959
1960      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 19) Then
1961          Wait s10i19() day
1962
1963      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 20) Then
1964          Wait s10i20() day
1965
1966      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 21) Then
1967          Wait s10i21() day
1968
1969      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 22) Then
1970          Wait s10i22() day
1971
1972      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 23) Then
1973          Wait s10i23() day
1974
1975      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 24) Then
1976          Wait s10i24() day
1977
1978      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 25) Then
1979          Wait s10i25() day
1980
1981      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 26) Then
1982          Wait s10i26() day
1983
1984      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 27) Then
1985          Wait s10i27() day
1986
1987      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 28) Then
1988          Wait s10i28() day
1989
1990      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 29) Then
1991          Wait s10i29() day
1992
1993      If (Patient_Scenario = 10) And (Patient_SurgicalTeam = 30) Then
1994          Wait s10i30() day
1995
1996      If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 1) Then
1997          Wait s1i1() day
1998
1999      If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 2) Then
2000          Wait s1i2() day
2001
2002      If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 3) Then
2003          Wait s1i3() day
2004
2005      If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 4) Then
2006          Wait s1i4() day
2007
2008      If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 5) Then
2009          Wait s1i5() day

```

2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070

```

If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 6) Then
    Wait s1116() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 7) Then
    Wait s1117() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 8) Then
    Wait s1118() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 9) Then
    Wait s1119() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 10) Then
    Wait s1110() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 11) Then
    Wait s1111() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 12) Then
    Wait s1112() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 13) Then
    Wait s1113() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 14) Then
    Wait s1114() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 15) Then
    Wait s1115() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 16) Then
    Wait s1116() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 17) Then
    Wait s1117() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 18) Then
    Wait s1118() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 19) Then
    Wait s1119() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 20) Then
    Wait s1120() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 21) Then
    Wait s1121() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 22) Then
    Wait s1122() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 23) Then
    Wait s1123() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 24) Then
    Wait s1124() day
If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 25) Then
    Wait s1125() day

```

2071 If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 26) Then
 2072 Wait s11i26() day
 2073
 2074 If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 27) Then
 2075 Wait s11i27() day
 2076
 2077 If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 28) Then
 2078 Wait s11i28() day
 2079
 2080 If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 29) Then
 2081 Wait s11i29() day
 2082
 2083 If (Patient_Scenario = 11) And (Patient_SurgicalTeam = 30) Then
 2084 Wait s11i30() day
 2085
 2086 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 1) Then
 2087 Wait s12i1() day
 2088
 2089 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 2) Then
 2090 Wait s12i2() day
 2091
 2092 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 3) Then
 2093 Wait s12i3() day
 2094
 2095 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 4) Then
 2096 Wait s12i4() day
 2097
 2098 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 5) Then
 2099 Wait s12i5() day
 2100
 2101 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 6) Then
 2102 Wait s12i6() day
 2103
 2104 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 7) Then
 2105 Wait s12i7() day
 2106
 2107 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 8) Then
 2108 Wait s12i8() day
 2109
 2110 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 9) Then
 2111 Wait s12i9() day
 2112
 2113 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 10) Then
 2114 Wait s12i10() day
 2115
 2116 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 11) Then
 2117 Wait s12i11() day
 2118
 2119 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 12) Then
 2120 Wait s12i12() day
 2121
 2122 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 13) Then
 2123 Wait s12i13() day
 2124
 2125 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 14) Then
 2126 Wait s12i14() day
 2127
 2128 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 15) Then
 2129 Wait s12i15() day
 2130
 2131 If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 16) Then

```

2132      Wait s12i16() day
2133
2134      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 17) Then
2135          Wait s12i17() day
2136
2137      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 18) Then
2138          Wait s12i18() day
2139
2140      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 19) Then
2141          Wait s12i19() day
2142
2143      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 20) Then
2144          Wait s12i20() day
2145
2146      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 21) Then
2147          Wait s12i21() day
2148
2149      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 22) Then
2150          Wait s12i22() day
2151
2152      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 23) Then
2153          Wait s12i23() day
2154
2155      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 24) Then
2156          Wait s12i24() day
2157
2158      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 25) Then
2159          Wait s12i25() day
2160
2161      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 26) Then
2162          Wait s12i26() day
2163
2164      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 27) Then
2165          Wait s12i27() day
2166
2167      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 28) Then
2168          Wait s12i28() day
2169
2170      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 29) Then
2171          Wait s12i29() day
2172
2173      If (Patient_Scenario = 12) And (Patient_SurgicalTeam = 30) Then
2174          Wait s12i30() day
2175
2176      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 1) Then
2177          Wait s13i1() day
2178
2179      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 2) Then
2180          Wait s13i2() day
2181
2182      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 3) Then
2183          Wait s13i3() day
2184
2185      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 4) Then
2186          Wait s13i4() day
2187
2188      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 5) Then
2189          Wait s13i5() day
2190
2191      If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 6) Then
2192          Wait s13i6() day

```

2193 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 7) Then
2194 Wait sl317() day
2195
2196 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 8) Then
2197 Wait sl318() day
2198
2199 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 9) Then
2200 Wait sl319() day
2201
2202 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 10) Then
2203 Wait sl3110() day
2204
2205 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 11) Then
2206 Wait sl3111() day
2207
2208 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 12) Then
2209 Wait sl3112() day
2210
2211 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 13) Then
2212 Wait sl3113() day
2213
2214 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 14) Then
2215 Wait sl3114() day
2216
2217 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 15) Then
2218 Wait sl3115() day
2219
2220 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 16) Then
2221 Wait sl3116() day
2222
2223 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 17) Then
2224 Wait sl3117() day
2225
2226 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 18) Then
2227 Wait sl3118() day
2228
2229 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 19) Then
2230 Wait sl3119() day
2231
2232 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 20) Then
2233 Wait sl3120() day
2234
2235 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 21) Then
2236 Wait sl3121() day
2237
2238 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 22) Then
2239 Wait sl3122() day
2240
2241 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 23) Then
2242 Wait sl3123() day
2243
2244 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 24) Then
2245 Wait sl3124() day
2246
2247 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 25) Then
2248 Wait sl3125() day
2249
2250 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 26) Then
2251 Wait sl3126() day
2252
2253

2254 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 27) Then
 2255 Wait s13i27() day
 2256
 2257 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 28) Then
 2258 Wait s13i28() day
 2259
 2260 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 29) Then
 2261 Wait s13i29() day
 2262
 2263 If (Patient_Scenario = 13) And (Patient_SurgicalTeam = 30) Then
 2264 Wait s13i30() day
 2265
 2266 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 1) Then
 2267 Wait s14i1() day
 2268
 2269 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 2) Then
 2270 Wait s14i2() day
 2271
 2272 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 3) Then
 2273 Wait s14i3() day
 2274
 2275 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 4) Then
 2276 Wait s14i4() day
 2277
 2278 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 5) Then
 2279 Wait s14i5() day
 2280
 2281 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 6) Then
 2282 Wait s14i6() day
 2283
 2284 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 7) Then
 2285 Wait s14i7() day
 2286
 2287 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 8) Then
 2288 Wait s14i8() day
 2289
 2290 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 9) Then
 2291 Wait s14i9() day
 2292
 2293 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 10) Then
 2294 Wait s14i10() day
 2295
 2296 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 11) Then
 2297 Wait s14i11() day
 2298
 2299 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 12) Then
 2300 Wait s14i12() day
 2301
 2302 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 13) Then
 2303 Wait s14i13() day
 2304
 2305 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 14) Then
 2306 Wait s14i14() day
 2307
 2308 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 15) Then
 2309 Wait s14i15() day
 2310
 2311 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 16) Then
 2312 Wait s14i16() day
 2313
 2314 If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 17) Then


```

2315      Wait s14i17() day
2316
2317      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 18) Then
2318          Wait s14i18() day
2319
2320      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 19) Then
2321          Wait s14i19() day
2322
2323      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 20) Then
2324          Wait s14i20() day
2325
2326      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 21) Then
2327          Wait s14i21() day
2328
2329      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 22) Then
2330          Wait s14i22() day
2331
2332      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 23) Then
2333          Wait s14i23() day
2334
2335      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 24) Then
2336          Wait s14i24() day
2337
2338      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 25) Then
2339          Wait s14i25() day
2340
2341      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 26) Then
2342          Wait s14i26() day
2343
2344      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 27) Then
2345          Wait s14i27() day
2346
2347      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 28) Then
2348          Wait s14i28() day
2349
2350      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 29) Then
2351          Wait s14i29() day
2352
2353      If (Patient_Scenario = 14) And (Patient_SurgicalTeam = 30) Then
2354          Wait s14i30() day
2355
2356      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 1) Then
2357          Wait s15i1() day
2358
2359      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 2) Then
2360          Wait s15i2() day
2361
2362      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 3) Then
2363          Wait s15i3() day
2364
2365      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 4) Then
2366          Wait s15i4() day
2367
2368      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 5) Then
2369          Wait s15i5() day
2370
2371      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 6) Then
2372          Wait s15i6() day
2373
2374      If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 7) Then
2375          Wait s15i7() day

```

2376 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 8) Then
 2377 Wait s1518() day
 2378
 2379 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 9) Then
 2380 Wait s1519() day
 2381
 2382 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 10) Then
 2383 Wait s1510() day
 2384
 2385 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 11) Then
 2386 Wait s1511() day
 2387
 2388 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 12) Then
 2389 Wait s1512() day
 2390
 2391 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 13) Then
 2392 Wait s1513() day
 2393
 2394 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 14) Then
 2395 Wait s1514() day
 2396
 2397 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 15) Then
 2398 Wait s1515() day
 2399
 2400 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 16) Then
 2401 Wait s1516() day
 2402
 2403 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 17) Then
 2404 Wait s1517() day
 2405
 2406 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 18) Then
 2407 Wait s1518() day
 2408
 2409 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 19) Then
 2410 Wait s1519() day
 2411
 2412 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 20) Then
 2413 Wait s1520() day
 2414
 2415 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 21) Then
 2416 Wait s1521() day
 2417
 2418 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 22) Then
 2419 Wait s1522() day
 2420
 2421 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 23) Then
 2422 Wait s1523() day
 2423
 2424 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 24) Then
 2425 Wait s1524() day
 2426
 2427 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 25) Then
 2428 Wait s1525() day
 2429
 2430 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 26) Then
 2431 Wait s1526() day
 2432
 2433 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 27) Then
 2434 Wait s1527() day
 2435
 2436

2437 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 28) Then
 2438 Wait sl5i28() day
 2439
 2440 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 29) Then
 2441 Wait sl5i29() day
 2442
 2443 If (Patient_Scenario = 15) And (Patient_SurgicalTeam = 30) Then
 2444 Wait sl5i30() day
 2445
 2446 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 1) Then
 2447 Wait sl6i1() day
 2448
 2449 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 2) Then
 2450 Wait sl6i2() day
 2451
 2452 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 3) Then
 2453 Wait sl6i3() day
 2454
 2455 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 4) Then
 2456 Wait sl6i4() day
 2457
 2458 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 5) Then
 2459 Wait sl6i5() day
 2460
 2461 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 6) Then
 2462 Wait sl6i6() day
 2463
 2464 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 7) Then
 2465 Wait sl6i7() day
 2466
 2467 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 8) Then
 2468 Wait sl6i8() day
 2469
 2470 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 9) Then
 2471 Wait sl6i9() day
 2472
 2473 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 10) Then
 2474 Wait sl6i10() day
 2475
 2476 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 11) Then
 2477 Wait sl6i11() day
 2478
 2479 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 12) Then
 2480 Wait sl6i12() day
 2481
 2482 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 13) Then
 2483 Wait sl6i13() day
 2484
 2485 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 14) Then
 2486 Wait sl6i14() day
 2487
 2488 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 15) Then
 2489 Wait sl6i15() day
 2490
 2491 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 16) Then
 2492 Wait sl6i16() day
 2493
 2494 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 17) Then
 2495 Wait sl6i17() day
 2496
 2497 If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 18) Then

```

2498      Wait s16i118( ) day
2499
2500      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 19) Then
2501          Wait s16i119( ) day
2502
2503      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 20) Then
2504          Wait s16i20( ) day
2505
2506      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 21) Then
2507          Wait s16i21( ) day
2508
2509      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 22) Then
2510          Wait s16i22( ) day
2511
2512      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 23) Then
2513          Wait s16i23( ) day
2514
2515      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 24) Then
2516          Wait s16i24( ) day
2517
2518      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 25) Then
2519          Wait s16i25( ) day
2520
2521      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 26) Then
2522          Wait s16i26( ) day
2523
2524      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 27) Then
2525          Wait s16i27( ) day
2526
2527      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 28) Then
2528          Wait s16i28( ) day
2529
2530      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 29) Then
2531          Wait s16i29( ) day
2532
2533      If (Patient_Scenario = 16) And (Patient_SurgicalTeam = 30) Then
2534          Wait s16i30( ) day
2535
2536      Patient      Discharge      1      Patient      Discharge      FIRST 1
2537      Patient      Wait N(30,1) min 1      Patient      EXIT      FIRST 1
2538
2539      *****
2540      * Arrivals
2541      *
2542      *****
2543
2544      Entity      Location      Qty Each      First Time      Occurrences Frequency      Logic
2545      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_SurgicalTeam = 15
2546      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_OperatingRoom = 1
2547      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_DayofSurgery = 1
2548      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_Scenario = 1 + rand(15)
2549      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_SurgicalTeam = 5
2550      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_OperatingRoom = 2
2551      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_DayofSurgery = 1
2552      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_Scenario = 1 + rand(15)
2553      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_SurgicalTeam = 17
2554      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_OperatingRoom = 3
2555      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_DayofSurgery = 1
2556      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_Scenario = 1 + rand(15)
2557      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_SurgicalTeam = 26
2558      PatientBatch ArrivalBuffer 1      Mon, Feb 01 2010 @ 06:00 AM 1      PatientBatch_OperatingRoom = 4

```

Graphic 3

2559						PatientBatch_DayofSurgery = 1
2560						PatientBatch_Scenario = 1 + rand(15)
2561	PatientBatch_ArrivalBuffer 1	Mon, Feb 01 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 6
2562						PatientBatch_OperatingRoom = 5
2563						PatientBatch_DayofSurgery = 1
2564						PatientBatch_Scenario = 1 + rand(15)
2565	PatientBatch_ArrivalBuffer 1	Mon, Feb 01 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 28
2566						PatientBatch_OperatingRoom = 6
2567						PatientBatch_DayofSurgery = 1
2568						PatientBatch_Scenario = 1 + rand(15)
2569	PatientBatch_ArrivalBuffer 1	Mon, Feb 01 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 10
2570						PatientBatch_OperatingRoom = 7
2571						PatientBatch_DayofSurgery = 1
2572						PatientBatch_Scenario = 1 + rand(15)
2573	PatientBatch_ArrivalBuffer 1	Mon, Feb 01 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 12
2574						PatientBatch_OperatingRoom = 8
2575						PatientBatch_DayofSurgery = 1
2576						PatientBatch_Scenario = 1 + rand(15)
2577	PatientBatch_ArrivalBuffer 1	Mon, Feb 01 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 21
2578						PatientBatch_OperatingRoom = 9
2579						PatientBatch_DayofSurgery = 1
2580						PatientBatch_Scenario = 1 + rand(15)
2581	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 28
2582						PatientBatch_OperatingRoom = 1
2583						PatientBatch_DayofSurgery = 2
2584						PatientBatch_Scenario = 1 + rand(15)
2585	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 30
2586						PatientBatch_OperatingRoom = 2
2587						PatientBatch_DayofSurgery = 2
2588						PatientBatch_Scenario = 1 + rand(15)
2589	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 18
2590						PatientBatch_OperatingRoom = 3
2591						PatientBatch_DayofSurgery = 2
2592						PatientBatch_Scenario = 1 + rand(15)
2593	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 17
2594						PatientBatch_OperatingRoom = 4
2595						PatientBatch_DayofSurgery = 2
2596						PatientBatch_Scenario = 1 + rand(15)
2597	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 14
2598						PatientBatch_OperatingRoom = 5
2599						PatientBatch_DayofSurgery = 2
2600						PatientBatch_Scenario = 1 + rand(15)
2601	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 10
2602						PatientBatch_OperatingRoom = 6
2603						PatientBatch_DayofSurgery = 2
2604						PatientBatch_Scenario = 1 + rand(15)
2605	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 16
2606						PatientBatch_OperatingRoom = 7
2607						PatientBatch_DayofSurgery = 2
2608						PatientBatch_Scenario = 1 + rand(15)
2609	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 21
2610						PatientBatch_OperatingRoom = 8
2611						PatientBatch_DayofSurgery = 2
2612						PatientBatch_Scenario = 1 + rand(15)
2613	PatientBatch_ArrivalBuffer 1	Tue, Feb 02 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 22
2614						PatientBatch_OperatingRoom = 9
2615						PatientBatch_DayofSurgery = 2
2616						PatientBatch_Scenario = 1 + rand(15)
2617	PatientBatch_ArrivalBuffer 1	Wed, Feb 03 2010 @ 06:00 AM	1			PatientBatch_SurgicalTeam = 2
2618						PatientBatch_OperatingRoom = 1
2619						PatientBatch_DayofSurgery = 3

2620	PatientBatch ArrivalBuffer 1	Wed, Feb 03 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
2621	PatientBatch SurgicalTeam = 4			PatientBatch_SurgicalTeam = 4
2622	PatientBatch OperatingRoom = 2			PatientBatch_OperatingRoom = 2
2623	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2624	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2625	PatientBatch SurgicalTeam = 14			PatientBatch_SurgicalTeam = 14
2626	PatientBatch OperatingRoom = 3			PatientBatch_OperatingRoom = 3
2627	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2628	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2629	PatientBatch ArrivalBuffer 1	Wed, Feb 03 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
2630	PatientBatch OperatingRoom = 21			PatientBatch_OperatingRoom = 21
2631	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2632	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2633	PatientBatch SurgicalTeam = 24			PatientBatch_SurgicalTeam = 24
2634	PatientBatch OperatingRoom = 5			PatientBatch_OperatingRoom = 5
2635	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2636	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2637	PatientBatch SurgicalTeam = 11			PatientBatch_SurgicalTeam = 11
2638	PatientBatch OperatingRoom = 6			PatientBatch_OperatingRoom = 6
2639	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2640	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2641	PatientBatch ArrivalBuffer 1	Wed, Feb 03 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
2642	PatientBatch OperatingRoom = 7			PatientBatch_OperatingRoom = 7
2643	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2644	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2645	PatientBatch SurgicalTeam = 13			PatientBatch_SurgicalTeam = 13
2646	PatientBatch OperatingRoom = 8			PatientBatch_OperatingRoom = 8
2647	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2648	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2649	PatientBatch SurgicalTeam = 19			PatientBatch_SurgicalTeam = 19
2650	PatientBatch OperatingRoom = 9			PatientBatch_OperatingRoom = 9
2651	PatientBatch DayofSurgery = 3			PatientBatch_DayofSurgery = 3
2652	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2653	PatientBatch SurgicalTeam = 26			PatientBatch_SurgicalTeam = 26
2654	PatientBatch OperatingRoom = 1			PatientBatch_OperatingRoom = 1
2655	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2656	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2657	PatientBatch SurgicalTeam = 5			PatientBatch_SurgicalTeam = 5
2658	PatientBatch OperatingRoom = 2			PatientBatch_OperatingRoom = 2
2659	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2660	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2661	PatientBatch SurgicalTeam = 23			PatientBatch_SurgicalTeam = 23
2662	PatientBatch OperatingRoom = 3			PatientBatch_OperatingRoom = 3
2663	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2664	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2665	PatientBatch SurgicalTeam = 1			PatientBatch_SurgicalTeam = 1
2666	PatientBatch OperatingRoom = 4			PatientBatch_OperatingRoom = 4
2667	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2668	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2669	PatientBatch SurgicalTeam = 7			PatientBatch_SurgicalTeam = 7
2670	PatientBatch OperatingRoom = 5			PatientBatch_OperatingRoom = 5
2671	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2672	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2673	PatientBatch SurgicalTeam = 6			PatientBatch_SurgicalTeam = 6
2674	PatientBatch OperatingRoom = 6			PatientBatch_OperatingRoom = 6
2675	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2676	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2677	PatientBatch SurgicalTeam = 19			PatientBatch_SurgicalTeam = 19
2678	PatientBatch OperatingRoom = 7			PatientBatch_OperatingRoom = 7
2679	PatientBatch DayofSurgery = 4			PatientBatch_DayofSurgery = 4
2680	PatientBatch_Scenario = 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)

2681	PatientBatch ArrivalBuffer 1	Thu, Feb 04 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 12
2682				PatientBatch_OperatingRoom = 8
2683				PatientBatch_DayofSurgery = 4
2684				PatientBatch_Scenario = 1 + rand(15)
2685	PatientBatch ArrivalBuffer 1	Thu, Feb 04 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 29
2686				PatientBatch_OperatingRoom = 9
2687				PatientBatch_DayofSurgery = 4
2688				PatientBatch_Scenario = 1 + rand(15)
2689	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 14
2690				PatientBatch_OperatingRoom = 1
2691				PatientBatch_DayofSurgery = 5
2692				PatientBatch_Scenario = 1 + rand(15)
2693	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 1
2694				PatientBatch_OperatingRoom = 2
2695				PatientBatch_DayofSurgery = 5
2696				PatientBatch_Scenario = 1 + rand(15)
2697	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 12
2698				PatientBatch_OperatingRoom = 3
2699				PatientBatch_DayofSurgery = 5
2700				PatientBatch_Scenario = 1 + rand(15)
2701	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 15
2702				PatientBatch_OperatingRoom = 4
2703				PatientBatch_DayofSurgery = 5
2704				PatientBatch_Scenario = 1 + rand(15)
2705	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 18
2706				PatientBatch_OperatingRoom = 5
2707				PatientBatch_DayofSurgery = 5
2708				PatientBatch_Scenario = 1 + rand(15)
2709	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 9
2710				PatientBatch_OperatingRoom = 6
2711				PatientBatch_DayofSurgery = 5
2712				PatientBatch_Scenario = 1 + rand(15)
2713	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 6
2714				PatientBatch_OperatingRoom = 7
2715				PatientBatch_DayofSurgery = 5
2716				PatientBatch_Scenario = 1 + rand(15)
2717	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 25
2718				PatientBatch_OperatingRoom = 8
2719				PatientBatch_DayofSurgery = 5
2720				PatientBatch_Scenario = 1 + rand(15)
2721	PatientBatch ArrivalBuffer 1	Fri, Feb 05 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 26
2722				PatientBatch_OperatingRoom = 9
2723				PatientBatch_DayofSurgery = 5
2724				PatientBatch_Scenario = 1 + rand(15)
2725	PatientBatch ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 9
2726				PatientBatch_OperatingRoom = 1
2727				PatientBatch_DayofSurgery = 8
2728				PatientBatch_Scenario = 1 + rand(15)
2729	PatientBatch ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 5
2730				PatientBatch_OperatingRoom = 2
2731				PatientBatch_DayofSurgery = 8
2732				PatientBatch_Scenario = 1 + rand(15)
2733	PatientBatch ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 7
2734				PatientBatch_OperatingRoom = 3
2735				PatientBatch_DayofSurgery = 8
2736				PatientBatch_Scenario = 1 + rand(15)
2737	PatientBatch ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 25
2738				PatientBatch_OperatingRoom = 4
2739				PatientBatch_DayofSurgery = 8
2740				PatientBatch_Scenario = 1 + rand(15)
2741	PatientBatch ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 17

2742						PatientBatch_OperatingRoom = 5
2743						PatientBatch_DayofSurgery = 8
2744						PatientBatch_Scenario = 1 + rand(15)
2745	PatientBatch_ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 10
2746						PatientBatch_OperatingRoom = 6
2747						PatientBatch_DayofSurgery = 8
2748	PatientBatch_ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM 1	1			PatientBatch_Scenario = 1 + rand(15)
2749						PatientBatch_SurgicalTeam = 15
2750						PatientBatch_OperatingRoom = 7
2751						PatientBatch_DayofSurgery = 8
2752						PatientBatch_Scenario = 1 + rand(15)
2753	PatientBatch_ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 8
2754						PatientBatch_OperatingRoom = 8
2755						PatientBatch_DayofSurgery = 8
2756						PatientBatch_Scenario = 1 + rand(15)
2757	PatientBatch_ArrivalBuffer 1	Mon, Feb 08 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 21
2758						PatientBatch_OperatingRoom = 9
2759						PatientBatch_DayofSurgery = 8
2760						PatientBatch_Scenario = 1 + rand(15)
2761	PatientBatch_ArrivalBuffer	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 10
2762						PatientBatch_OperatingRoom = 1
2763						PatientBatch_DayofSurgery = 9
2764						PatientBatch_Scenario = 1 + rand(15)
2765	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 30
2766						PatientBatch_OperatingRoom = 2
2767						PatientBatch_DayofSurgery = 9
2768						PatientBatch_Scenario = 1 + rand(15)
2769	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 21
2770						PatientBatch_OperatingRoom = 3
2771						PatientBatch_DayofSurgery = 9
2772						PatientBatch_Scenario = 1 + rand(15)
2773	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 16
2774						PatientBatch_OperatingRoom = 4
2775						PatientBatch_DayofSurgery = 9
2776						PatientBatch_Scenario = 1 + rand(15)
2777	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 17
2778						PatientBatch_OperatingRoom = 5
2779						PatientBatch_DayofSurgery = 9
2780						PatientBatch_Scenario = 1 + rand(15)
2781	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 14
2782						PatientBatch_OperatingRoom = 6
2783						PatientBatch_DayofSurgery = 9
2784						PatientBatch_Scenario = 1 + rand(15)
2785	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 4
2786						PatientBatch_OperatingRoom = 7
2787						PatientBatch_DayofSurgery = 9
2788						PatientBatch_Scenario = 1 + rand(15)
2789	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 19
2790						PatientBatch_OperatingRoom = 8
2791						PatientBatch_DayofSurgery = 9
2792						PatientBatch_Scenario = 1 + rand(15)
2793	PatientBatch_ArrivalBuffer 1	Tue, Feb 09 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 28
2794						PatientBatch_OperatingRoom = 9
2795						PatientBatch_DayofSurgery = 9
2796						PatientBatch_Scenario = 1 + rand(15)
2797	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 30
2798						PatientBatch_OperatingRoom = 1
2799						PatientBatch_DayofSurgery = 10
2800						PatientBatch_Scenario = 1 + rand(15)
2801	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM 1	1			PatientBatch_SurgicalTeam = 4
2802						PatientBatch_OperatingRoom = 2

2803							PatientBatch_DayofSurgery = 10
2804							PatientBatch_Scenario = 1 + rand(15)
2805	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 11
2806							PatientBatch_OperatingRoom = 3
2807							PatientBatch_DayofSurgery = 10
2808							PatientBatch_Scenario = 1 + rand(15)
2809	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 21
2810							PatientBatch_OperatingRoom = 4
2811							PatientBatch_DayofSurgery = 10
2812							PatientBatch_Scenario = 1 + rand(15)
2813	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 23
2814							PatientBatch_OperatingRoom = 5
2815							PatientBatch_DayofSurgery = 10
2816							PatientBatch_Scenario = 1 + rand(15)
2817	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 2
2818							PatientBatch_OperatingRoom = 6
2819							PatientBatch_DayofSurgery = 10
2820							PatientBatch_Scenario = 1 + rand(15)
2821	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 5
2822							PatientBatch_OperatingRoom = 7
2823							PatientBatch_DayofSurgery = 10
2824							PatientBatch_Scenario = 1 + rand(15)
2825	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 25
2826							PatientBatch_OperatingRoom = 8
2827							PatientBatch_DayofSurgery = 10
2828							PatientBatch_Scenario = 1 + rand(15)
2829	PatientBatch_ArrivalBuffer 1	Wed, Feb 10 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 24
2830							PatientBatch_OperatingRoom = 9
2831							PatientBatch_DayofSurgery = 10
2832							PatientBatch_Scenario = 1 + rand(15)
2833	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 25
2834							PatientBatch_OperatingRoom = 1
2835							PatientBatch_DayofSurgery = 11
2836							PatientBatch_Scenario = 1 + rand(15)
2837	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 5
2838							PatientBatch_OperatingRoom = 2
2839							PatientBatch_DayofSurgery = 11
2840							PatientBatch_Scenario = 1 + rand(15)
2841	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 7
2842							PatientBatch_OperatingRoom = 3
2843							PatientBatch_DayofSurgery = 11
2844							PatientBatch_Scenario = 1 + rand(15)
2845	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 29
2846							PatientBatch_OperatingRoom = 4
2847							PatientBatch_DayofSurgery = 11
2848							PatientBatch_Scenario = 1 + rand(15)
2849	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 1
2850							PatientBatch_OperatingRoom = 5
2851							PatientBatch_DayofSurgery = 11
2852							PatientBatch_Scenario = 1 + rand(15)
2853	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 26
2854							PatientBatch_OperatingRoom = 6
2855							PatientBatch_DayofSurgery = 11
2856							PatientBatch_Scenario = 1 + rand(15)
2857	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 19
2858							PatientBatch_OperatingRoom = 7
2859							PatientBatch_DayofSurgery = 11
2860							PatientBatch_Scenario = 1 + rand(15)
2861	PatientBatch_ArrivalBuffer 1	Thu, Feb 11 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 12
2862							PatientBatch_OperatingRoom = 8
2863							PatientBatch_DayofSurgery = 11

2864	PatientBatch	ArrivalBuffer	1	Thu, Feb 11 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
2865	PatientBatch	SurgicalTeam	= 23			PatientBatch_SurgicalTeam = 23
2866	PatientBatch	DayofSurgery	= 9			PatientBatch_DayofSurgery = 9
2867	PatientBatch	DayofSurgery	= 11			PatientBatch_DayofSurgery = 11
2868	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2869	PatientBatch	SurgicalTeam	= 23			PatientBatch_SurgicalTeam = 23
2870	PatientBatch	OperatingRoom	= 1			PatientBatch_OperatingRoom = 1
2871	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2872	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2873	PatientBatch	SurgicalTeam	= 3			PatientBatch_SurgicalTeam = 3
2874	PatientBatch	OperatingRoom	= 2			PatientBatch_OperatingRoom = 2
2875	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2876	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2877	PatientBatch	SurgicalTeam	= 15			PatientBatch_SurgicalTeam = 15
2878	PatientBatch	OperatingRoom	= 3			PatientBatch_OperatingRoom = 3
2879	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2880	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2881	PatientBatch	SurgicalTeam	= 12			PatientBatch_SurgicalTeam = 12
2882	PatientBatch	OperatingRoom	= 4			PatientBatch_OperatingRoom = 4
2883	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2884	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2885	PatientBatch	SurgicalTeam	= 18			PatientBatch_SurgicalTeam = 18
2886	PatientBatch	OperatingRoom	= 5			PatientBatch_OperatingRoom = 5
2887	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2888	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2889	PatientBatch	SurgicalTeam	= 9			PatientBatch_SurgicalTeam = 9
2890	PatientBatch	OperatingRoom	= 6			PatientBatch_OperatingRoom = 6
2891	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2892	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2893	PatientBatch	SurgicalTeam	= 14			PatientBatch_SurgicalTeam = 14
2894	PatientBatch	OperatingRoom	= 7			PatientBatch_OperatingRoom = 7
2895	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2896	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2897	PatientBatch	SurgicalTeam	= 26			PatientBatch_SurgicalTeam = 26
2898	PatientBatch	OperatingRoom	= 8			PatientBatch_OperatingRoom = 8
2899	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2900	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2901	PatientBatch	SurgicalTeam	= 1			PatientBatch_SurgicalTeam = 1
2902	PatientBatch	OperatingRoom	= 9			PatientBatch_OperatingRoom = 9
2903	PatientBatch	DayofSurgery	= 12			PatientBatch_DayofSurgery = 12
2904	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2905	PatientBatch	SurgicalTeam	= 7			PatientBatch_SurgicalTeam = 7
2906	PatientBatch	OperatingRoom	= 1			PatientBatch_OperatingRoom = 1
2907	PatientBatch	DayofSurgery	= 15			PatientBatch_DayofSurgery = 15
2908	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2909	PatientBatch	SurgicalTeam	= 5			PatientBatch_SurgicalTeam = 5
2910	PatientBatch	OperatingRoom	= 2			PatientBatch_OperatingRoom = 2
2911	PatientBatch	DayofSurgery	= 15			PatientBatch_DayofSurgery = 15
2912	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2913	PatientBatch	SurgicalTeam	= 28			PatientBatch_SurgicalTeam = 28
2914	PatientBatch	OperatingRoom	= 3			PatientBatch_OperatingRoom = 3
2915	PatientBatch	DayofSurgery	= 15			PatientBatch_DayofSurgery = 15
2916	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2917	PatientBatch	SurgicalTeam	= 11			PatientBatch_SurgicalTeam = 11
2918	PatientBatch	OperatingRoom	= 4			PatientBatch_OperatingRoom = 4
2919	PatientBatch	DayofSurgery	= 15			PatientBatch_DayofSurgery = 15
2920	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)
2921	PatientBatch	SurgicalTeam	= 15			PatientBatch_SurgicalTeam = 15
2922	PatientBatch	OperatingRoom	= 5			PatientBatch_OperatingRoom = 5
2923	PatientBatch	DayofSurgery	= 15			PatientBatch_DayofSurgery = 15
2924	PatientBatch	Scenario	= 1 + rand(15)			PatientBatch_Scenario = 1 + rand(15)

2925	PatientBatch ArrivalBuffer 1	Mon, Feb 15 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 9
2926				PatientBatch_OperatingRoom = 6
2927				PatientBatch_DayofSurgery = 15
2928	PatientBatch ArrivalBuffer 1	Mon, Feb 15 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
2929				PatientBatch_SurgicalTeam = 10
2930				PatientBatch_OperatingRoom = 7
2931				PatientBatch_DayofSurgery = 15
2932				PatientBatch_Scenario = 1 + rand(15)
2933	PatientBatch ArrivalBuffer 1	Mon, Feb 15 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 17
2934				PatientBatch_OperatingRoom = 8
2935				PatientBatch_DayofSurgery = 15
2936				PatientBatch_Scenario = 1 + rand(15)
2937	PatientBatch ArrivalBuffer 1	Mon, Feb 15 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 25
2938				PatientBatch_OperatingRoom = 9
2939				PatientBatch_DayofSurgery = 15
2940				PatientBatch_Scenario = 1 + rand(15)
2941				PatientBatch_OperatingRoom = 21
2942	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_OperatingRoom = 1
2943				PatientBatch_DayofSurgery = 16
2944				PatientBatch_Scenario = 1 + rand(15)
2945	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 30
2946				PatientBatch_OperatingRoom = 2
2947				PatientBatch_DayofSurgery = 16
2948				PatientBatch_Scenario = 1 + rand(15)
2949	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 16
2950				PatientBatch_OperatingRoom = 3
2951				PatientBatch_DayofSurgery = 16
2952				PatientBatch_Scenario = 1 + rand(15)
2953	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 26
2954				PatientBatch_OperatingRoom = 4
2955				PatientBatch_DayofSurgery = 16
2956				PatientBatch_Scenario = 1 + rand(15)
2957	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 28
2958				PatientBatch_OperatingRoom = 5
2959				PatientBatch_DayofSurgery = 16
2960				PatientBatch_Scenario = 1 + rand(15)
2961	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 14
2962				PatientBatch_OperatingRoom = 6
2963				PatientBatch_DayofSurgery = 16
2964				PatientBatch_Scenario = 1 + rand(15)
2965	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 10
2966				PatientBatch_OperatingRoom = 7
2967				PatientBatch_DayofSurgery = 16
2968				PatientBatch_Scenario = 1 + rand(15)
2969	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 19
2970				PatientBatch_OperatingRoom = 9
2971				PatientBatch_DayofSurgery = 16
2972				PatientBatch_Scenario = 1 + rand(15)
2973	PatientBatch ArrivalBuffer 1	Tue, Feb 16 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 5
2974				PatientBatch_OperatingRoom = 8
2975				PatientBatch_DayofSurgery = 16
2976				PatientBatch_Scenario = 1 + rand(15)
2977	PatientBatch ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 5
2978				PatientBatch_OperatingRoom = 1
2979				PatientBatch_DayofSurgery = 17
2980				PatientBatch_Scenario = 1 + rand(15)
2981	PatientBatch ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 2
2982				PatientBatch_OperatingRoom = 2
2983				PatientBatch_DayofSurgery = 17
2984				PatientBatch_Scenario = 1 + rand(15)
2985	PatientBatch ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 18

2986					PatientBatch_OperatingRoom = 3
2987					PatientBatch_DayofSurgery = 17
2988					PatientBatch_Scenario = 1 + rand(15)
2989	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 7
2990					PatientBatch_OperatingRoom = 4
2991					PatientBatch_DayofSurgery = 17
2992	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_Scenario = 1 + rand(15)
2993					PatientBatch_SurgicalTeam = 21
2994					PatientBatch_OperatingRoom = 5
2995					PatientBatch_DayofSurgery = 17
2996					PatientBatch_Scenario = 1 + rand(15)
2997	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 1
2998					PatientBatch_OperatingRoom = 6
2999					PatientBatch_DayofSurgery = 17
3000					PatientBatch_Scenario = 1 + rand(15)
3001	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 28
3002					PatientBatch_OperatingRoom = 7
3003					PatientBatch_DayofSurgery = 17
3004					PatientBatch_Scenario = 1 + rand(15)
3005	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 11
3006					PatientBatch_OperatingRoom = 8
3007					PatientBatch_DayofSurgery = 17
3008					PatientBatch_Scenario = 1 + rand(15)
3009	PatientBatch_ArrivalBuffer 1	Wed, Feb 17 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 4
3010					PatientBatch_OperatingRoom = 9
3011					PatientBatch_DayofSurgery = 17
3012					PatientBatch_Scenario = 1 + rand(15)
3013	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 28
3014					PatientBatch_OperatingRoom = 1
3015					PatientBatch_DayofSurgery = 18
3016					PatientBatch_Scenario = 1 + rand(15)
3017	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 1
3018					PatientBatch_OperatingRoom = 2
3019					PatientBatch_DayofSurgery = 18
3020					PatientBatch_Scenario = 1 + rand(15)
3021	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 19
3022					PatientBatch_OperatingRoom = 3
3023					PatientBatch_DayofSurgery = 18
3024					PatientBatch_Scenario = 1 + rand(15)
3025	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 15
3026					PatientBatch_OperatingRoom = 4
3027					PatientBatch_DayofSurgery = 18
3028					PatientBatch_Scenario = 1 + rand(15)
3029	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 6
3030					PatientBatch_OperatingRoom = 5
3031					PatientBatch_DayofSurgery = 18
3032					PatientBatch_Scenario = 1 + rand(15)
3033	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 5
3034					PatientBatch_OperatingRoom = 6
3035					PatientBatch_DayofSurgery = 18
3036					PatientBatch_Scenario = 1 + rand(15)
3037	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 2
3038					PatientBatch_OperatingRoom = 7
3039					PatientBatch_DayofSurgery = 18
3040					PatientBatch_Scenario = 1 + rand(15)
3041	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 20
3042					PatientBatch_OperatingRoom = 8
3043					PatientBatch_DayofSurgery = 18
3044					PatientBatch_Scenario = 1 + rand(15)
3045	PatientBatch_ArrivalBuffer 1	Thu, Feb 18 2010 @ 06:00 AM 1	1		PatientBatch_SurgicalTeam = 29
3046					PatientBatch_OperatingRoom = 9

3047							PatientBatch_DayofSurgery = 18
3048							PatientBatch_Scenario = 1 + rand(15)
3049							PatientBatch_SurgicalTeam = 14
3050	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_OperatingRoom = 1
3051							PatientBatch_DayofSurgery = 19
3052							PatientBatch_Scenario = 1 + rand(15)
3053	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 1
3054							PatientBatch_OperatingRoom = 2
3055							PatientBatch_DayofSurgery = 19
3056							PatientBatch_Scenario = 1 + rand(15)
3057	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 9
3058							PatientBatch_OperatingRoom = 3
3059							PatientBatch_DayofSurgery = 19
3060							PatientBatch_Scenario = 1 + rand(15)
3061	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 18
3062							PatientBatch_OperatingRoom = 4
3063							PatientBatch_DayofSurgery = 19
3064							PatientBatch_Scenario = 1 + rand(15)
3065	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 26
3066							PatientBatch_OperatingRoom = 5
3067							PatientBatch_DayofSurgery = 19
3068							PatientBatch_Scenario = 1 + rand(15)
3069	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 16
3070							PatientBatch_OperatingRoom = 6
3071							PatientBatch_DayofSurgery = 19
3072							PatientBatch_Scenario = 1 + rand(15)
3073	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 27
3074							PatientBatch_OperatingRoom = 7
3075							PatientBatch_DayofSurgery = 19
3076							PatientBatch_Scenario = 1 + rand(15)
3077	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 12
3078							PatientBatch_OperatingRoom = 8
3079							PatientBatch_DayofSurgery = 19
3080							PatientBatch_Scenario = 1 + rand(15)
3081	PatientBatch ArrivalBuffer 1	Fri, Feb 19 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 25
3082							PatientBatch_OperatingRoom = 9
3083							PatientBatch_DayofSurgery = 19
3084							PatientBatch_Scenario = 1 + rand(15)
3085	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 17
3086							PatientBatch_OperatingRoom = 1
3087							PatientBatch_DayofSurgery = 22
3088							PatientBatch_Scenario = 1 + rand(15)
3089	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 5
3090							PatientBatch_OperatingRoom = 2
3091							PatientBatch_DayofSurgery = 22
3092							PatientBatch_Scenario = 1 + rand(15)
3093	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 10
3094							PatientBatch_OperatingRoom = 3
3095							PatientBatch_DayofSurgery = 22
3096							PatientBatch_Scenario = 1 + rand(15)
3097	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 11
3098							PatientBatch_OperatingRoom = 4
3099							PatientBatch_DayofSurgery = 22
3100							PatientBatch_Scenario = 1 + rand(15)
3101	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 25
3102							PatientBatch_OperatingRoom = 5
3103							PatientBatch_DayofSurgery = 22
3104							PatientBatch_Scenario = 1 + rand(15)
3105	PatientBatch ArrivalBuffer 1	Mon, Feb 22 2010 @ 06:00 AM	1				PatientBatch_SurgicalTeam = 15
3106							PatientBatch_OperatingRoom = 6
3107							PatientBatch_DayofSurgery = 22

3108	PatientBatch	ArrivalBuffer	1	Mon, Feb 22 2010 @ 06:00 AM	1	PatientBatch_Scenario = 1 + rand(15)
3109	PatientBatch	ArrivalBuffer	1	Mon, Feb 22 2010 @ 06:00 AM	1	PatientBatch_Scenario = 7
3110	PatientBatch	OperatingRoom	7			PatientBatch_DayofSurgery = 7
3111	PatientBatch	OperatingRoom	7			PatientBatch_DayofSurgery = 22
3112	PatientBatch	OperatingRoom	7			PatientBatch_Scenario = 1 + rand(15)
3113	PatientBatch	OperatingRoom	7			PatientBatch_Scenario = 9
3114	PatientBatch	OperatingRoom	8			PatientBatch_DayofSurgery = 8
3115	PatientBatch	OperatingRoom	8			PatientBatch_DayofSurgery = 22
3116	PatientBatch	OperatingRoom	8			PatientBatch_Scenario = 1 + rand(15)
3117	PatientBatch	OperatingRoom	9			PatientBatch_Scenario = 29
3118	PatientBatch	OperatingRoom	9			PatientBatch_DayofSurgery = 9
3119	PatientBatch	OperatingRoom	9			PatientBatch_DayofSurgery = 22
3120	PatientBatch	OperatingRoom	9			PatientBatch_Scenario = 1 + rand(15)
3121	PatientBatch	OperatingRoom	17			PatientBatch_Scenario = 17
3122	PatientBatch	OperatingRoom	17			PatientBatch_DayofSurgery = 1
3123	PatientBatch	OperatingRoom	17			PatientBatch_DayofSurgery = 23
3124	PatientBatch	OperatingRoom	17			PatientBatch_Scenario = 1 + rand(15)
3125	PatientBatch	OperatingRoom	30			PatientBatch_Scenario = 30
3126	PatientBatch	OperatingRoom	2			PatientBatch_DayofSurgery = 2
3127	PatientBatch	OperatingRoom	2			PatientBatch_DayofSurgery = 23
3128	PatientBatch	OperatingRoom	2			PatientBatch_Scenario = 1 + rand(15)
3129	PatientBatch	OperatingRoom	26			PatientBatch_Scenario = 26
3130	PatientBatch	OperatingRoom	3			PatientBatch_DayofSurgery = 3
3131	PatientBatch	OperatingRoom	3			PatientBatch_DayofSurgery = 23
3132	PatientBatch	OperatingRoom	30			PatientBatch_Scenario = 1 + rand(15)
3133	PatientBatch	OperatingRoom	21			PatientBatch_Scenario = 21
3134	PatientBatch	OperatingRoom	4			PatientBatch_DayofSurgery = 4
3135	PatientBatch	OperatingRoom	4			PatientBatch_DayofSurgery = 23
3136	PatientBatch	OperatingRoom	12			PatientBatch_Scenario = 1 + rand(15)
3137	PatientBatch	OperatingRoom	12			PatientBatch_Scenario = 12
3138	PatientBatch	OperatingRoom	5			PatientBatch_DayofSurgery = 5
3139	PatientBatch	OperatingRoom	23			PatientBatch_DayofSurgery = 23
3140	PatientBatch	OperatingRoom	23			PatientBatch_Scenario = 1 + rand(15)
3141	PatientBatch	OperatingRoom	10			PatientBatch_Scenario = 10
3142	PatientBatch	OperatingRoom	6			PatientBatch_DayofSurgery = 6
3143	PatientBatch	OperatingRoom	23			PatientBatch_DayofSurgery = 23
3144	PatientBatch	OperatingRoom	4			PatientBatch_Scenario = 1 + rand(15)
3145	PatientBatch	OperatingRoom	4			PatientBatch_Scenario = 4
3146	PatientBatch	OperatingRoom	7			PatientBatch_DayofSurgery = 7
3147	PatientBatch	OperatingRoom	23			PatientBatch_DayofSurgery = 23
3148	PatientBatch	OperatingRoom	19			PatientBatch_Scenario = 1 + rand(15)
3149	PatientBatch	OperatingRoom	19			PatientBatch_Scenario = 19
3150	PatientBatch	OperatingRoom	8			PatientBatch_DayofSurgery = 8
3151	PatientBatch	OperatingRoom	23			PatientBatch_DayofSurgery = 23
3152	PatientBatch	OperatingRoom	28			PatientBatch_Scenario = 1 + rand(15)
3153	PatientBatch	OperatingRoom	9			PatientBatch_Scenario = 9
3154	PatientBatch	OperatingRoom	23			PatientBatch_DayofSurgery = 23
3155	PatientBatch	OperatingRoom	23			PatientBatch_Scenario = 1 + rand(15)
3156	PatientBatch	OperatingRoom	21			PatientBatch_Scenario = 21
3157	PatientBatch	OperatingRoom	24			PatientBatch_DayofSurgery = 24
3158	PatientBatch	OperatingRoom	24			PatientBatch_DayofSurgery = 24
3159	PatientBatch	OperatingRoom	30			PatientBatch_Scenario = 1 + rand(15)
3160	PatientBatch	OperatingRoom	30			PatientBatch_Scenario = 30
3161	PatientBatch	OperatingRoom	2			PatientBatch_DayofSurgery = 2
3162	PatientBatch	OperatingRoom	24			PatientBatch_DayofSurgery = 24
3163	PatientBatch	OperatingRoom	2			PatientBatch_Scenario = 1 + rand(15)
3164	PatientBatch	OperatingRoom	2			PatientBatch_Scenario = 2
3165	PatientBatch	OperatingRoom	3			PatientBatch_DayofSurgery = 3
3166	PatientBatch	OperatingRoom	24			PatientBatch_DayofSurgery = 24
3167	PatientBatch	OperatingRoom	1			PatientBatch_Scenario = 1 + rand(15)
3168	PatientBatch	OperatingRoom	1			PatientBatch_Scenario = 1 + rand(15)

3169	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 6
3170				PatientBatch_OperatingRoom = 4
3171				PatientBatch_DayofSurgery = 24
3172				PatientBatch_Scenario = 1 + rand(15)
3173	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 25
3174				PatientBatch_OperatingRoom = 5
3175				PatientBatch_DayofSurgery = 24
3176				PatientBatch_Scenario = 1 + rand(15)
3177	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 11
3178				PatientBatch_OperatingRoom = 6
3179				PatientBatch_DayofSurgery = 24
3180				PatientBatch_Scenario = 1 + rand(15)
3181	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 17
3182				PatientBatch_OperatingRoom = 7
3183				PatientBatch_DayofSurgery = 24
3184				PatientBatch_Scenario = 1 + rand(15)
3185	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 28
3186				PatientBatch_OperatingRoom = 8
3187				PatientBatch_DayofSurgery = 24
3188				PatientBatch_Scenario = 1 + rand(15)
3189	PatientBatch ArrivalBuffer 1	Wed, Feb 24 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 24
3190				PatientBatch_OperatingRoom = 9
3191				PatientBatch_DayofSurgery = 24
3192				PatientBatch_Scenario = 1 + rand(15)
3193	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 5
3194				PatientBatch_OperatingRoom = 1
3195				PatientBatch_DayofSurgery = 25
3196				PatientBatch_Scenario = 1 + rand(15)
3197	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 1
3198				PatientBatch_OperatingRoom = 2
3199				PatientBatch_DayofSurgery = 25
3200				PatientBatch_Scenario = 1 + rand(15)
3201	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 3
3202				PatientBatch_OperatingRoom = 3
3203				PatientBatch_DayofSurgery = 25
3204				PatientBatch_Scenario = 1 + rand(15)
3205	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 26
3206				PatientBatch_OperatingRoom = 4
3207				PatientBatch_DayofSurgery = 25
3208				PatientBatch_Scenario = 1 + rand(15)
3209	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 20
3210				PatientBatch_OperatingRoom = 5
3211				PatientBatch_DayofSurgery = 25
3212				PatientBatch_Scenario = 1 + rand(15)
3213	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 7
3214				PatientBatch_OperatingRoom = 6
3215				PatientBatch_DayofSurgery = 25
3216				PatientBatch_Scenario = 1 + rand(15)
3217	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 19
3218				PatientBatch_OperatingRoom = 7
3219				PatientBatch_DayofSurgery = 25
3220				PatientBatch_Scenario = 1 + rand(15)
3221	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 12
3222				PatientBatch_OperatingRoom = 8
3223				PatientBatch_DayofSurgery = 25
3224				PatientBatch_Scenario = 1 + rand(15)
3225	PatientBatch ArrivalBuffer 1	Thu, Feb 25 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 23
3226				PatientBatch_OperatingRoom = 9
3227				PatientBatch_DayofSurgery = 25
3228				PatientBatch_Scenario = 1 + rand(15)
3229	PatientBatch ArrivalBuffer 1	Fri, Feb 26 2010 @ 06:00 AM	1	PatientBatch_SurgicalTeam = 14

```

3230 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 1
3231 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3232 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3233 PatientBatch_ArrivalBuffer 1 Fri, Feb 26 2010 @ 06:00 AM 1 PatientBatch_SurgicalTeam = 1
3234 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 2
3235 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3236 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3237 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 26
3238 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 3
3239 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3240 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3241 PatientBatch_ArrivalBuffer 1 Fri, Feb 26 2010 @ 06:00 AM 1 PatientBatch_SurgicalTeam = 9
3242 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 4
3243 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3244 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3245 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 6
3246 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 5
3247 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3248 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3249 PatientBatch_ArrivalBuffer 1 Fri, Feb 26 2010 @ 06:00 AM 1 PatientBatch_SurgicalTeam = 12
3250 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 6
3251 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3252 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3253 PatientBatch_ArrivalBuffer 1 Fri, Feb 26 2010 @ 06:00 AM 1 PatientBatch_SurgicalTeam = 18
3254 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 7
3255 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3256 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3257 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 15
3258 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 8
3259 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3260 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3261 PatientBatch_ArrivalBuffer 1 Fri, Feb 26 2010 @ 06:00 AM 1 PatientBatch_SurgicalTeam = 25
3262 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 9
3263 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3264 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3265 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 6
3266 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 2
3267 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3268 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3269 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 12
3270 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 3
3271 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3272 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3273 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 9
3274 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 4
3275 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3276 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3277 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 25
3278 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 9
3279 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3280 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3281 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 6
3282 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 2
3283 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3284 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3285 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 12
3286 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 3
3287 PatientBatch_ArrivalBuffer 1 PatientBatch_DayofSurgery = 26
3288 PatientBatch_ArrivalBuffer 1 PatientBatch_Scenario = 1 + rand(15)
3289 PatientBatch_ArrivalBuffer 1 PatientBatch_SurgicalTeam = 9
3290 PatientBatch_ArrivalBuffer 1 PatientBatch_OperatingRoom = 4

```

```

*****
* Shift Assignments
*****
Locations... Resources... Calendar File... Priorities... Disable Logic...
-----
ORcalendar.pmcsl 99,99,99,99 No
OperatingRoom_1
OperatingRoom_2
OperatingRoom_3
OperatingRoom_4
OperatingRoom_5
OperatingRoom_6
OperatingRoom_7
OperatingRoom_8
OperatingRoom_9
Surgical_Team_1
Surgical_Team_10
Surgical_Team_11
Surgical_Team_12
Surgical_Team_13
Surgical_Team_14
Surgical_Team_15
Surgical_Team_16
Surgical_Team_17

```



```

3291 Surgical_Team_18
3292 Surgical_Team_19
3293 Surgical_Team_2
3294 Surgical_Team_20
3295 Surgical_Team_21
3296 Surgical_Team_22
3297 Surgical_Team_23
3298 Surgical_Team_24
3299 Surgical_Team_25
3300 Surgical_Team_26
3301 Surgical_Team_27
3302 Surgical_Team_28
3303 Surgical_Team_29
3304 Surgical_Team_3
3305 Surgical_Team_30
3306 Surgical_Team_4
3307 Surgical_Team_5
3308 Surgical_Team_6
3309 Surgical_Team_7
3310 Surgical_Team_8
3311 Surgical_Team_9
3312
3313
3314

```

```

3315 *****Attributes*****
3316 *****
3317 *****
3318 *****

```

```

ID Type Classification
-----
PatientBatch_Scenario Integer Entity
Patient_Scenario Integer Entity
PatientBatch_SurgicalTeam Integer Entity
Patient_SurgicalTeam Integer Entity
PatientBatch_OperatingRoom Integer Entity
Patient_OperatingRoom Integer Entity
PatientBatch_DayofSurgery Integer Entity
Patient_DayofSurgery Integer Entity

```

```

3331 *****Arrays*****
3332 *****
3333 *****
3334 *****

```

```

ID Dimensions Type Import File Export File Disable Persist
-----
A 30,16 Real simulationdata.xlsx None No
maximumNumberOfPatients 30,16 Integer simulationdata.xlsx None No

```

```

3340 *****User Distributions*****
3341 *****
3342 *****
3343 *****
3344 *****

```

```

ID Type Cumulative Percentage Value
-----
s111 Discrete No 33.1051745879108 0day
33.318534832404 1day
33.5762905796852 2day
s112 Discrete No 16.7641467972294 0day
16.6902703421826 1day

```

3352				16.2785635181429	2day
3353				16.5680763289729	3day
3354				16.9982580820921	4day
3355				16.7006849313802	5day
3356	sl13	Discrete	No	31.4014218354058000	0day
3357				33.7276495147772	1day
3358				34.870928649817	2day
3359	sl14	Discrete	No	24.8006347452137	0day
3360				25.6774441182768	1day
3361				24.3448533757185	2day
3362				25.177067760791	3day
3363	sl15	Discrete	No	24.607972385488	0day
3364				24.6469413456395	1day
3365				25.3700760179004	2day
3366	sl16	Discrete	No	25.3750102509721	3day
3367				33.9644776326125	0day
3368				32.7238725245386	1day
3369				33.311649842849	2day
3370	sl17	Discrete	No	25.2279398295615	0day
3371				25.123037297746	1day
3372				25.2213813697123	2day
3373				24.4276415029602	3day
3374	sl18	Discrete	No	52.564528993973	0day
3375				47.435471006027	1day
3376	sl19	Discrete	No	25.42245173625	0day
3377				25.3193125437274	1day
3378				24.6591309302351	2day
3379				24.5991047897875	3day
3380	sl110	Discrete	No	20.0833199908548	0day
3381				19.7964282434059	1day
3382				20.1980748883227	2day
3383				20.2283490137011	3day
3384				19.6938278637154	4day
3385	sl111	Discrete	No	34.3321511152545	0day
3386				32.7307921024526	1day
3387				32.9370567822929	2day
3388	sl112	Discrete	No	100	0day
3389	sl113	Discrete	No	33.0939793631048	0day
3390				34.479396285432	1day
3391				32.4266243514632	2day
3392	sl114	Discrete	No	50.5165817246744	0day
3393				49.4834182753256	1day
3394	sl115	Discrete	No	19.8445662491549	0day
3395				19.4261618783574	1day
3396				20.3421057487247	2day
3397				20.504163343608	3day
3398				19.8830027894021	4day
3399	sl116	Discrete	No	16.1591757265093	0day
3400				17.3031248493839	1day
3401				16.3021898446645	2day
3402				16.8143446137032	3day
3403				16.6034591191249	4day
3404				16.8177058466141	5day
3405	sl117	Discrete	No	24.6931455414591	0day
3406				24.6205587714937	1day
3407				24.6416301761388	2day
3408				26.0446655109084	3day
3409	sl118	Discrete	No	25.1070844783193	0day
3410				24.3766303273594	1day
3411				24.7857666707122	2day
3412				25.730518523609	3day

3413	s1i19	Discrete	No	33.2380527327141	0day
3414				33.6599720853014	1day
3415				33.1019751819844	2day
3416	s1i20	Discrete	No	25.0893141782455	0day
3417				24.0107233637528	1day
3418				25.091975785319	2day
3419				25.8079864626827	3day
3420	s1i21	Discrete	No	19.778239351934	0day
3421				19.2459025773186	1day
3422				20.25096589772	2day
3423				19.9938022657153	3day
3424				20.7269592152602	4day
3425	s1i22	Discrete	No	25.1109982433898	0day
3426				25.6351157127634	1day
3427				24.6969819831012	2day
3428				24.5569040607456	3day
3429	s1i23	Discrete	No	16.4925605511723	0day
3430				16.5668591913936	1day
3431				16.8710763079242	2day
3432				16.89099488227076	3day
3433				16.3513845011373	4day
3434				16.827124625665	5day
3435	s1i24	Discrete	No	16.32096221145789	0day
3436				17.3929073444032	1day
3437				16.50542032646	2day
3438				15.5826859536725	3day
3439				17.4485444398455	4day
3440				16.74947983104	5day
3441	s1i25	Discrete	No	49.6619346019887	0day
3442				50.3380653980113	1day
3443	s1i26	Discrete	No	25.719968863813	0day
3444				24.4148690004872	1day
3445				25.1767111676629	2day
3446				24.6884509860369	3day
3447	s1i27	Discrete	No	25.4932787894904	0day
3448				25.0098983592752	1day
3449				24.4313095110717	2day
3450				25.0655133601627	3day
3451	s1i28	Discrete	No	49.2114108192687	0day
3452				50.7865891807313	1day
3453	s1i29	Discrete	No	16.0227674405706	0day
3454				16.7315345779971	1day
3455				16.6804140310763	2day
3456				17.4178930712871	3day
3457				16.6116658049711	4day
3458				16.5357250740978	5day
3459	s1i30	Discrete	No	51.7514004277033	0day
3460				48.2485995722967	1day
3461	s2i1	Discrete	No	100	0day
3462	s2i2	Discrete	No	16.9360501107808	0day
3463				16.6542892504429	1day
3464				16.6612338786615	2day
3465				16.0784898663391	3day
3466				16.9872768956176	4day
3467				16.6826602009062	5day
3468	s2i3	Discrete	No	20.0220577075669	0day
3469				20.750604524976	1day
3470				19.6187817634775	2day
3471				20.2019852502052	3day
3472				19.4065707537744	4day
3473	s2i4	Discrete	No	33.3228862701776	0day

3474				33.7964761185095	1day
3475				32.880637611313	2day
3476		Discrete	No	25.482475620314	0day
3477	s2i5			24.7175306630821	1day
3478				24.9786656278845	2day
3479				24.8213280887194	3day
3480	s2i6	Discrete	No	100	0day
3481	s2i7	Discrete	No	49.3771844704967	0day
3482		Discrete	No	50.6228155295033	1day
3483	s2i8	Discrete	No	50.0835047819671	0day
3484		Discrete	No	49.9164952180329	1day
3485	s2i9	Discrete	No	49.8994111110809	0day
3486		Discrete	No	50.1005888889191	1day
3487	s2i10	Discrete	No	32.8053466175867	0day
3488		Discrete	No	34.1221198051951	1day
3489		Discrete	No	33.0725353772182	2day
3490	s2i11	Discrete	No	16.63345288523380000000%	0day
3491		Discrete	No	16.883287538840000000%	1day
3492		Discrete	No	16.65004457376560000000%	2day
3493		Discrete	No	16.41562035857300000000%	3day
3494		Discrete	No	16.98586469372630000000%	4day
3495		Discrete	No	16.43172934817300000000%	5day
3496	s2i12	Discrete	No	24.9966215134944	0day
3497		Discrete	No	25.0690591767834	1day
3498		Discrete	No	24.5933319885242	2day
3499		Discrete	No	25.3409873111979	3day
3500	s2i13	Discrete	No	19.9076145993101	0day
3501		Discrete	No	19.9842934943149	1day
3502		Discrete	No	19.9901660519269	2day
3503		Discrete	No	19.7725026594794	3day
3504		Discrete	No	20.3454231949687	4day
3505	s2i14	Discrete	No	20.0623131095582	0day
3506		Discrete	No	19.94471667497	1day
3507		Discrete	No	19.7997893251763	2day
3508		Discrete	No	19.7734392677011	3day
3509		Discrete	No	20.4197416227675	4day
3510	s2i15	Discrete	No	16.7041867840328	0day
3511		Discrete	No	16.4531234188641	1day
3512		Discrete	No	16.5692905495551	2day
3513		Discrete	No	17.094069213444	3day
3514		Discrete	No	16.1346057567711	4day
3515		Discrete	No	17.044724277333	5day
3516	s2i16	Discrete	No	50.7292521649273	0day
3517		Discrete	No	49.27074783507270000000%	1day
3518	s2i17	Discrete	No	16.7806379394092	0day
3519		Discrete	No	16.5677522047729	1day
3520		Discrete	No	16.3688743579573	2day
3521		Discrete	No	16.8092723900563	3day
3522		Discrete	No	16.770637912499	4day
3523		Discrete	No	16.7028251953051	5day
3524	s2i18	Discrete	No	50.0818357492525	0day
3525		Discrete	No	49.9181642507475	1day
3526	s2i19	Discrete	No	19.845174362239	0day
3527		Discrete	No	20.5396621780198	1day
3528		Discrete	No	20.2023389321986	2day
3529		Discrete	No	19.5814368692158	3day
3530		Discrete	No	19.8313876583269	4day
3531	s2i20	Discrete	No	19.7475738514698	0day
3532		Discrete	No	20.046538520653	1day
3533		Discrete	No	19.7184893107365	2day
3534		Discrete	No	20.398964065784	3day

3535				20.08843425135568	4day
3536	s2i21	Discrete	No	50.9764397830425	0day
3537				49.0235602169575	1day
3538	s2i22	Discrete	No	25.0418315115393	0day
3539				25.2665405650978	1day
3540				24.658242024079	2day
3541	s2i23	Discrete	No	25.0333858992838	3day
3543				33.5991050090216	0day
3544				33.1159910211189	1day
3545	s2i24	Discrete	No	33.7765422560726	2day
3546				32.7664893078052	1day
3547				33.4569684361222	2day
3548	s2i25	Discrete	No	25.1680357263732	0day
3549				24.6307915873609	1day
3550				23.2132082195884	2day
3551				24.9879644666775	3day
3552	s2i26	Discrete	No	16.6771954338683	0day
3553				16.4838160414925	1day
3554				16.8601810035471	2day
3555				16.5667894001283	3day
3556				16.8415023137189	4day
3557				16.5705158072449	5day
3558	s2i27	Discrete	No	100	0day
3559	s2i28	Discrete	No	25.0624255375633	0day
3560				24.6004251580916	1day
3561				25.17046050802	2day
3562	s2i29	Discrete	No	25.1666887835431	3day
3563				25.2579753309248	0day
3564				24.8204905001769	1day
3565				25.4258723657238	2day
3566				24.4956618031745	3day
3567	s2i30	Discrete	No	32.9752202667347	0day
3568				32.8212984138348	1day
3569				34.2034813194305	2day
3570	s3i1	Discrete	No	20.0361595971713	0day
3571				19.9473474143028	1day
3572				20.1642684598568	2day
3573				20.0222462132993	3day
3574				19.8299783333698	4day
3575	s3i2	Discrete	No	32.8769402883526	0day
3576				33.3796869673693	1day
3577				33.7433727442781	2day
3578	s3i3	Discrete	No	16.7124585838174	0day
3579				16.5585172798918	1day
3580				16.6572363054957	2day
3581				16.7954635214517	3day
3582				16.7582514148649	4day
3583				16.5180728944785	5day
3584	s3i4	Discrete	No	100	0day
3585	s3i5	Discrete	No	49.640110851808	0day
3586				50.359889146192	1day
3587	s3i6	Discrete	No	100	0day
3588	s3i7	Discrete	No	20.1182021215628	0day
3589				19.9475862132679	1day
3590				19.9169366645972	2day
3591				20.0202871159806	3day
3592				19.9969878845914	4day
3593	s3i8	Discrete	No	16.8288844738073	0day
3594				16.5015513058006	1day
3595				16.6290785879071	2day

3596			16.8295146014181	3day
3597			16.7549279162411	4day
3598			16.4560431148259	5day
3599	s3i9	Discrete	16.6049599138972	0day
3600		No	16.9779210702807	1day
3601			16.6651480356827	2day
3602			16.6978830129269	3day
3603			16.5225946957736	4day
3604			16.5314932714389	5day
3605	s3i10	Discrete	24.6824009117885	0day
3606		No	25.0249988630138	1day
3607			25.0376130621925	2day
3608			25.2549871630052	3day
3609	s3i11	Discrete	49.8188198335622	0day
3610		No	50.1811801664378	1day
3611	s3i12	Discrete	20.1207628986008	0day
3612		No	19.7084493051291	1day
3613			19.8042760530303	2day
3614			20.2895714358606	3day
3615			20.0769403073792	4day
3616	s3i13	Discrete	20.1765198382883	0day
3617		No	19.7451824704407	1day
3618			20.0307316916887	2day
3619			20.3007100795353	3day
3620			19.7468559200469	4day
3621	s3i14	Discrete	16.572943504001	0day
3622		No	16.657876564444	1day
3623			16.6535454729556	2day
3624			16.5641095573691	3day
3625			16.7609236699893	4day
3626			16.7888012312405	5day
3627	s3i15	Discrete	20.2021155873719	0day
3628		No	19.7945708841236	1day
3629			19.648903638525	2day
3630			20.4046155771237	3day
3631			19.9497943128558	4day
3632			25.185303684506	0day
3633	s3i16	Discrete	24.8700387738893	1day
3634			25.0389290201063	2day
3635	s3i17	Discrete	24.9056818475539	3day
3636		No	16.5916586898253	0day
3637			16.8865685126072	1day
3638			16.4584941219854	2day
3639			16.5638130664422	3day
3640			16.7355407297774	4day
3641			16.7639248811626	5day
3642	s3i18	Discrete	100	0day
3643	s3i19	Discrete	50.3270039040248	0day
3644		No	49.6729960959752	1day
3645	s3i20	Discrete	49.7645187453722	0day
3646		No	50.2354812546278	1day
3647	s3i21	Discrete	33.1311262058833	0day
3648		No	33.3545686782274	1day
3649			33.5143051158893	2day
3650	s3i22	Discrete	24.7710500212556	0day
3651		No	25.0095003195112	1day
3652			25.2505488358609	2day
3653			24.9689008233723	3day
3654	s3i23	Discrete	100	0day
3655	s3i24	Discrete	25.2529652518391	0day
3656		No	25.1635617221058	1day

3657				24.8555946590902	2day
3658				24.7278783669649	3day
3659			No	20.336768069417	0day
3660	s3i25	Discrete		19.9858635862269	1day
3661				19.7550957528429	2day
3662				19.8792683722656	3day
3663				20.0430042192476	4day
3664	s3i26	Discrete	No	50.0669204696941	0day
3665				49.9330795303059	1day
3666	s3i27	Discrete	No	16.6166141484961	0day
3667				16.7801585104305	1day
3668				16.6486450541957	2day
3669				16.7587422188617	3day
3670				16.68679072002	4day
3671				16.509049347994	5day
3672	s3i28	Discrete	No	100	0day
3673	s3i29	Discrete	No	16.9172335043898	0day
3674				16.3248642589308	1day
3675				16.9325609180201	2day
3676				16.5894305248259	3day
3677				16.6414591800625	4day
3678				16.5944516137708	5day
3679	s3i30	Discrete	No	33.232096602461	0day
3680				33.4405460234199	1day
3681				33.3273573741191	2day
3682	s4i1	Discrete	No	100	0day
3683	s4i2	Discrete	No	100	0day
3684	s4i3	Discrete	No	20.0428837359469	0day
3685				19.9536313640158	1day
3686				19.9873895104057	2day
3687				19.9210470003142	3day
3688				20.0950483893173	4day
3689	s4i4	Discrete	No	49.7207649682348	0day
3690				50.2792350317651	1day
3691	s4i5	Discrete	No	20.282647296455	0day
3692				19.9421411328695	1day
3693				20.0906186591904	2day
3694				19.7152263897463	3day
3695				20.0437490885482	4day
3696	s4i6	Discrete	No	24.643419335589	0day
3697				24.8838703103428	1day
3698				25.222527148112	2day
3699				25.250457639257	3day
3700	s4i7	Discrete	No	32.9905753218997	0day
3701				33.3742775243334	1day
3702				33.6351471557669	2day
3703	s4i8	Discrete	No	100	0day
3704	s4i9	Discrete	No	20.1587043604761	0day
3705				19.8048943581986	1day
3706				20.1434627185065	2day
3707				20.0073886411689	3day
3708				19.8855499216498	4day
3709	s4i10	Discrete	No	33.6067683922517	0day
3710				33.2669347884912	1day
3711				33.1262968192572	2day
3712	s4i11	Discrete	No	33.1253013614914	0day
3713				33.3882249970731	1day
3714				33.4864736414355	2day
3715	s4i12	Discrete	No	33.5405994492777000000000%	0day
3716				33.07133653291930000000%	1day
3717				33.38806401780300000000%	2day

3718	s4i13	Discrete	No	16.5516860231895	0day
3719				16.6663953487243	1day
3720				16.7758834354989	2day
3721				16.6217580111856	3day
3722				16.6657953688108	4day
3723				16.718481812591	5day
3724	s4i14	Discrete	No	50.1858919151164	0day
3725				49.8141080848837	1day
3726	s4i15	Discrete	No	33.5676052983551	0day
3727				33.160778473837	1day
3728				33.2716162278079	2day
3729	s4i16	Discrete	No	33.2090453664538	0day
3730				33.4992741762794	1day
3731				33.2916804572667	2day
3732	s4i17	Discrete	No	100	0day
3733	s4i18	Discrete	No	50.5875373921802	0day
3734				49.4124626078198	1day
3735	s4i19	Discrete	No	49.602126266325	0day
3736				50.397873733675	1day
3737	s4i20	Discrete	No	19.7438595015983	0day
3738				19.725460963345	1day
3739				20.103551899963	2day
3740				20.1805536544481	3day
3741				20.2465739806455	4day
3742	s4i21	Discrete	No	24.9894140084953	0day
3743				24.9127717261418	1day
3744				25.0483443689541	2day
3745				25.0494698964088	3day
3746	s4i22	Discrete	No	19.9372822314145	0day
3747				19.9735664228189	1day
3748				20.1810572580314	2day
3749				19.686228951015	3day
3750				20.2218651367202	4day
3751	s4i23	Discrete	No	20.0999659258178	0day
3752				20.0751187717707	1day
3753				19.8035601509742	2day
3754				20.1062015757445	3day
3755				19.9151535756928	4day
3756	s4i24	Discrete	No	16.5610514418245	0day
3757				16.7304755632546	1day
3758				16.6904987343275	2day
3759				16.6716351200764	3day
3760				16.5137668743403	4day
3761				16.8325722661767	5day
3762	s4i25	Discrete	No	100	0day
3763	s4i26	Discrete	No	33.743971124479	0day
3764				33.3857839295663	1day
3765				32.8702449459547	2day
3766	s4i27	Discrete	No	19.8885587791393	0day
3767				20.03191201608	1day
3768				20.0743252129411	2day
3769				20.0924033456444	3day
3770				19.9127733461951	4day
3771	s4i28	Discrete	No	20.0507478004898	0day
3772				20.1325038292319	1day
3773				19.8893530529226	2day
3774				19.8536421336653	3day
3775				20.073753186904	4day
3776	s4i29	Discrete	No	16.792246793386	0day
3777				16.8586055074479	1day
3778				16.6003036537884	2day

3779				16.3836707800523	3day
3780				16.726139292925533	4day
3781				16.6390339727721	5day
3782	s4i30	Discrete	No	25.2376114125706	0day
3783				24.9328287079091	1day
3784				25.0288910328122	2day
3785				24.8006688467081	3day
3786	s5i1	Discrete	No	19.4996966666419	0day
3787				20.2646025935285	1day
3788				20.5780481568236	2day
3789				20.1452347229963	3day
3790				19.5124178602326	4day
3791				33.65214186688	0day
3792	s5i2	Discrete	No	33.4423780556473	1day
3793				32.9054800774726	2day
3794	s5i3	Discrete	No	50.7330788578074	0day
3795				49.2669211421926	1day
3796	s5i4	Discrete	No	24.3088814921641	0day
3797				24.508507474094	1day
3798				25.4911134632166	2day
3799				25.6914975705253	3day
3800	s5i5	Discrete	No	16.7649070964711	0day
3801				16.2499134571583	1day
3802				16.4257907272434	2day
3803				16.8331132871649	3day
3804				16.6867819536563	4day
3805				17.039493476306	5day
3806	s5i6	Discrete	No	32.8895200364461	0day
3807				33.6987193088206	1day
3808				33.4117606527333	2day
3809	s5i7	Discrete	No	25.380322708935	0day
3810				24.896800304805	1day
3811				24.7396701840422	2day
3812				24.9832068022178	3day
3813	s5i8	Discrete	No	16.3115372903951	0day
3814				17.0334270268998	1day
3815				17.0788920666944	2day
3816				16.9211261053029	3day
3817				16.6340933099056	4day
3818				16.0209242008022	5day
3819	s5i9	Discrete	No	50.0323071282285	0day
3820				49.9676928717715	1day
3821	s5i10	Discrete	No	16.5121258538309	0day
3822				17.4501339803553	1day
3823				16.1254255109936	2day
3824				17.4934842228819	3day
3825				16.2924958663738	4day
3826				16.1263345656466	5day
3827	s5i11	Discrete	No	50.2968495081418	0day
3828				49.7031504918582	1day
3829				20.5161786957748	0day
3830	s5i12	Discrete	No	19.1685505816273	1day
3831				20.2761546453089	2day
3832				19.6195312370033	3day
3833				20.4195848402857	4day
3834	s5i13	Discrete	No	16.944546391639	0day
3835				17.1293287890542	1day
3836				16.4437448921225	2day
3837				16.4239608754781	3day
3838				16.8108671925933	4day
3839				16.247551859113	5day

3840	s5i14	Discrete	No	100	0day
3841	s5i15	Discrete	No	20.2562400062132	0day
3842				19.736069099345	1day
3843				20.2795116554116	2day
3844				19.7571322444631	3day
3845				19.9710469945671	4day
3846	s5i16	Discrete	No	50.5884364861227	0day
3847				49.4115635138773	1day
3848	s5i17	Discrete	No	100	0day
3849	s5i18	Discrete	No	25.1359876161658	0day
3850				24.777007024045	1day
3851				25.4370485557466	2day
3852				24.6499568040426	3day
3853	s5i19	Discrete	No	19.9811955608679	0day
3854				19.9336559160283	1day
3855				19.1736223967736	2day
3856				20.5520493333952	3day
3857				20.3594767803782	4day
3858	s5i20	Discrete	No	50.9044857661329	0day
3859				49.0955142338671	1day
3860	s5i21	Discrete	No	24.3427423511048	0day
3861				24.557391474553	1day
3862				25.6256186869865	2day
3863				25.4742474873556	3day
3864	s5i22	Discrete	No	31.9347286274385	0day
3865				34.634253517163	1day
3866				33.4310178553984	2day
3867	s5i23	Discrete	No	100	0day
3868	s5i24	Discrete	No	49.3814179803124	0day
3869				50.6185820196876	1day
3870	s5i25	Discrete	No	33.157374562165	0day
3871				33.95378286126	1day
3872				32.8888425825226	2day
3873	s5i26	Discrete	No	16.3383937806154	0day
3874				16.9222059317387	1day
3875				16.4044396238984	2day
3876				16.9867834512496	3day
3877				16.4610427580284	4day
3878				16.8871344454694	5day
3879	s5i27	Discrete	No	19.897884396838	0day
3880				19.5265520933295	1day
3881				20.2444241644561	2day
3882				20.2598105643059	3day
3883				20.0713288282249	4day
3884	s5i28	Discrete	No	100	0day
3885	s5i29	Discrete	No	100	0day
3886	s5i30	Discrete	No	33.8905751960537	0day
3887				32.993530009963	1day
3888				33.1158718029499	2day
3889	s6i1	Discrete	No	32.8922089880054	0day
3890				33.481874764393	1day
3891				33.6259162468016	2day
3892	s6i2	Discrete	No	49.1543066285828	0day
3893				50.8456933714172	1day
3894	s6i3	Discrete	No	20.4437097886591	0day
3895				20.2988362897534	1day
3896				19.7339102806029	2day
3897				19.310517047011	3day
3898				20.2130265939735	4day
3899	s6i4	Discrete	No	16.8495379401498	0day
3900				16.4410276131959	1day

3901			16.3042336985395	2day
3902			16.7515559427288	3day
3903			16.7521767042182	4day
3904			16.9014681011677	5day
3905	s6i5	No	19.7655916694986	0day
3906			19.8442381743964	1day
3907			19.8953425869867	2day
3908			20.36172557531	3day
3909			20.1331019938083	4day
3910	s6i6	No	25.003411630044	0day
3911			24.5187639351471	1day
3912			24.928028555883	2day
3913			25.5497958792206	3day
3914	s6i7	No	100	0day
3915	s6i8	No	33.469979565007	0day
3916			32.952924277863	1day
3917			33.5770961572067	2day
3918			33.8905011897976	0day
3919	s6i9	No	33.0008524584375	1day
3920			33.1086463517649	2day
3921	s6i10	No	19.7077665374839	0day
3922			19.9223949608578	1day
3923			20.2778411705867	2day
3924			19.5443387321759	3day
3925			20.5476585988957	4day
3926	s6i11	No	32.8292446812282	0day
3927			34.049937085929	1day
3928			33.1208182328428	2day
3929	s6i12	No	33.919381808661	0day
3930			32.9144258443797	1day
3931			33.1661923469593	2day
3932	s6i13	No	100	0day
3933	s6i14	No	100	0day
3934	s6i15	No	50.1528649625935	0day
3935			49.8471350374065	1day
3936	s6i16	No	33.3282946149912	0day
3937			33.498042676227	1day
3938			33.1736627087818	2day
3939	s6i17	No	33.0902358642641	0day
3940			33.7961482145418	1day
3941			33.1136159211941	2day
3942	s6i18	No	20.3982250587597	0day
3943			20.2780943610506	1day
3944			19.9586879875124	2day
3945			19.6377584988047	3day
3946	s6i19	No	19.7272341538727	4day
3947			16.8195450490001	0day
3948			16.9208419429296	1day
3949			16.6216138695107	2day
3950			16.2472637210403	3day
3951			16.6893537849191	4day
3952	s6i20	No	16.7013816326002	5day
3953			16.2545489700512	0day
3954			16.9318283720317	1day
3955			16.7156079356643	2day
3956			16.7901533990991	3day
3957			16.9086274878288	4day
3958			16.3992338353249	5day
3959	s6i21	No	19.9926413145809	0day
3960			19.2934678096897	1day
3961			20.0869930470806	2day

3962				20.4129374106407	3day
3963				20.2139604180081	4day
3964	s6i22	Discrete	No	20.1000634126205	0day
3965				20.2602630662804	1day
3966				20.235232128362	2day
3967				19.1849675094978	3day
3968				20.2194738832394	4day
3969	s6i23	Discrete	No	16.724176335182	0day
3970				16.569387639649	1day
3971				16.6438584057739	2day
3972				16.6469898590059	3day
3973				16.7771595248964	4day
3974				16.6384282354928	5day
3975	s6i24	Discrete	No	25.6143762788842	0day
3976				24.3750038947827	1day
3977				24.947463495698	2day
3978				25.063156330635	3day
3979	s6i25	Discrete	No	25.6318099152757	0day
3980				24.863988770594	1day
3981				24.8136730549965	2day
3982				24.6905282591338	3day
3983	s6i26	Discrete	No	24.7831137608997	0day
3984				24.8964358688789	1day
3985				25.2377998916735	2day
3986				25.0826504585479	3day
3987	s6i27	Discrete	No	34.7224891991855	0day
3988				32.3659367282501	1day
3989				32.9115740725644	2day
3990	s6i28	Discrete	No	19.2983172329394	0day
3991				20.4102635078504	1day
3992				20.3394009386673	2day
3993				19.9921910421104	3day
3994				19.9598272232325	4day
3995	s6i29	Discrete	No	16.4521947794454	0day
3996				16.8350971771807	1day
3997				16.780329998716	2day
3998				16.512429200241	3day
3999				16.6494003459987	4day
4000				16.7705484878794	5day
4001	s6i30	Discrete	No	50.0763192999608	0day
4002				49.9216807000392	1day
4003	s7i1	Discrete	No	49.9327608197561	0day
4004				50.0672391802439	1day
4005	s7i2	Discrete	No	20.3979990351597	0day
4006				20.0661949835623	1day
4007				19.9378365991274	2day
4008				19.7522325956114	3day
4009				19.8457367865392	4day
4010	s7i3	Discrete	No	50.2127340081055	0day
4011				49.7872659918945	1day
4012	s7i4	Discrete	No	25.259265986541	0day
4013				24.8437869693256	1day
4014				24.8720027454667	2day
4015				25.0249442986667	3day
4016	s7i5	Discrete	No	16.5166189734752	0day
4017				16.5350067755943	1day
4018				16.6896536561318	2day
4019				16.792706316344	3day
4020				16.6382656455388	4day
4021				16.8276843176255	5day
4022	s7i6	Discrete	No	16.4073998960352	0day

4023		16.9734079534502	1day
4024		16.3834488830451	2day
4025		16.7760561560748	3day
4026		16.9725349476372	4day
4027		16.4871521637576	5day
4028	No	33.6894113885965	0day
4029		33.1848109918281	1day
4030		33.1257476095755	2day
4031	No	24.8172642641467	0day
4032		25.0068918738321	1day
4033		25.2813136843529	2day
4034		24.8945301776683	3day
4035	No	100	0day
4036	No	25.1183427969461	0day
4037		24.7367143147896	1day
4038		25.2560178058681	2day
4039		24.8889250823763	3day
4040	No	16.8090490597037	0day
4041		16.7089889610863	1day
4042		16.7199758037744	2day
4043		16.5578825083325	3day
4044		16.8203686630133	4day
4045		16.3837350089899	5day
4046	No	19.9316675928154	0day
4047		20.2697082559535	1day
4048		20.1176296382992	2day
4049		19.9945130522848	3day
4050		19.664814606472	4day
4051	No	25.0367082119973	0day
4052		24.7700656278524	1day
4053		25.2178099794725	2day
4054		24.9754161806778	3day
4055	No	25.3361269455401	0day
4056		24.8060386967384	1day
4057		25.0519033729176	2day
4058		24.8059309848039	3day
4059	No	49.9280636276675	0day
4060		50.0719363723325	1day
4061	No	24.9266768250787	0day
4062		25.043503147006	1day
4063		25.1771738454433	2day
4064		24.852646182472	3day
4065	No	49.7522454523404	0day
4066		50.2477545476596	1day
4067	No	25.18315634341	0day
4068		25.2590229079817	1day
4069		24.8393164508745	2day
4070		24.7185042977338	3day
4071	No	100	0day
4072	No	24.702507058359	0day
4073		25.129900269249	1day
4074		25.2277428493116	2day
4075		24.9398498224045	3day
4076	No	50.050392855491	0day
4077		49.949607144509	1day
4078	No	25.1120393690771	0day
4079		24.8838867821597	1day
4080		24.9270663891834	2day
4081		25.07700749595797	3day
4082	No	19.8909185285775	0day
4083		20.1141813744062	1day

4084				19.96999956066636	2day
4085				20.0495510932872	3day
4086				19.9753494430655	4day
4087	s7i24	Discrete	No	50.3759669646473	0day
4088				49.6240330353527	1day
4089	s7i25	Discrete	No	16.626773427246	1day
4090				16.5488973803129	1day
4091				16.5709493089096	2day
4092				16.7587489436088	3day
4093				16.7466888218075	4day
4094				16.7479421181152	5day
4095	s7i26	Discrete	No	100	0day
4096	s7i27	Discrete	No	16.6588661139001	0day
4097				16.7611534062003	1day
4098				16.6008160988441	2day
4099				16.8825637331482	3day
4100				16.5825271585182	4day
4101				16.5140734883891	5day
4102	s7i28	Discrete	No	49.3646444420002	0day
4103				50.6353555579998	1day
4104	s7i29	Discrete	No	19.7186683164162	0day
4105				20.1209135966392	1day
4106				19.6781614383782	2day
4107				20.4025804813827	3day
4108				20.0796761651837	4day
4109	s7i30	Discrete	No	24.9592475394239	0day
4110				24.8366981150049	1day
4111				25.38979103566	2day
4112				24.8142564352144	3day
4113	s8i1	Discrete	No	100	0day
4114	s8i2	Discrete	No	100	0day
4115	s8i3	Discrete	No	49.752022201815	0day
4116				50.247977798185	1day
4117	s8i4	Discrete	No	100	0day
4118	s8i5	Discrete	No	16.8503598270544	0day
4119				16.888053863674	1day
4120				16.5732097741479	2day
4121				16.636986670564	3day
4122				16.4331372322828	4day
4123				16.6182526322768	5day
4124	s8i6	Discrete	No	100	0day
4125	s8i7	Discrete	No	100	0day
4126	s8i8	Discrete	No	100	0day
4127	s8i9	Discrete	No	33.5880732991412	0day
4128				33.4218712651331	1day
4129				32.9900554357257	2day
4130	s8i10	Discrete	No	33.1650659035775	0day
4131				33.036797125212	1day
4132				33.79813669712104	2day
4133	s8i11	Discrete	No	19.8668269807034	0day
4134				20.0437384263391	1day
4135				20.0888426889128	2day
4136				20.0348275390368	3day
4137				19.9657643650079	4day
4138	s8i12	Discrete	No	16.63516229469642	0day
4139				16.6546952470977	1day
4140				16.8271153139443	2day
4141				16.6597091103301	3day
4142				16.8024544854993	4day
4143				16.4208628961644	5day
4144	s8i13	Discrete	No	100	0day

4145	s8i14	Discrete	No	25.1330995282951	0day
4146				24.9421417018827	1day
4147				25.0386576388169	2day
4148				24.8861011310053	3day
4149	s8i15	Discrete	No	100	0day
4150	s8i16	Discrete	No	49.4354347678684	0day
4151	s8i17	Discrete	No	100	0day
4152				50.5645652321316	1day
4153	s8i18	Discrete	No	16.9089790736503	0day
4154				16.6197353009027	1day
4155				16.5465544613294	2day
4156				16.8072666500669	3day
4157				16.6922499694986	4day
4158				16.4252145445521	5day
4159	s8i19	Discrete	No	33.2422487505029	0day
4160				33.1600793459264	1day
4161				33.5976719035707	2day
4162	s8i20	Discrete	No	50.077985328395	0day
4163				49.922014671605	1day
4164	s8i21	Discrete	No	24.921752734514	0day
4165				25.2536966848745	1day
4166				24.500484849256	2day
4167				25.3240431867484	3day
4168	s8i22	Discrete	No	100	0day
4169	s8i23	Discrete	No	49.5562661789265	0day
4170				50.4437338210735	1day
4171	s8i24	Discrete	No	50.3290505146704	0day
4172				49.6709494853296	1day
4173	s8i25	Discrete	No	50.3781882273741	0day
4174				49.6218117726259	1day
4175	s8i26	Discrete	No	49.7319793033579	0day
4176				50.2680206966421	1day
4177	s8i27	Discrete	No	33.333649486831	0day
4178				33.4864437567694	1day
4179				33.1799067563995	2day
4180	s8i28	Discrete	No	100	0day
4181	s8i29	Discrete	No	33.2003385990707	0day
4182				33.2472763929957	1day
4183				33.5523850079336	2day
4184	s8i30	Discrete	No	33.030295737574	0day
4185				33.1031921831516	1day
4186				33.8665120792745	2day
4187	s9i1	Discrete	No	34.5134547630157	0day
4188				32.3393118115585	1day
4189				33.1472334254258	2day
4190	s9i2	Discrete	No	20.1989430465325	0day
4191				19.8940466190728	1day
4192				19.8625921400662	2day
4193				19.7896304677867	3day
4194				20.2547877265418	4day
4195	s9i3	Discrete	No	17.6313125757873	0day
4196				16.2881329219221	1day
4197				16.2581087336021	2day
4198				16.769699664572	3day
4199				16.9893501052914	4day
4200				16.0634256969399	5day
4201	s9i4	Discrete	No	33.7050567351202	0day
4202				33.098483095487	1day
4203				33.1964949553312	2day
4204	s9i5	Discrete	No	19.7027283835171	0day
4205				19.6773181735419	1day

4206				19.93776916364447	2day
4207				20.2234354805867	3day
4208				20.4588263259095	4day
4209	s9i6	Discrete	No	24.7096003371658	0day
4210				24.3926403218231	1day
4211				25.3098192100552	2day
4212				25.5879401309559	3day
4213	s9i7	Discrete	No	25.0991415209587	0day
4214				24.9504984254468	1day
4215				24.4974646315633	2day
4216				25.4528954220312	3day
4217	s9i8	Discrete	No	50.0339108467507	0day
4218				49.9660891532493	1day
4219	s9i9	Discrete	No	20.6229338390309	0day
4220				19.9313129238464	1day
4221				19.6093503544909	2day
4222				19.4920800151331	3day
4223				20.3443228674987	4day
4224				20.3467042219887	0day
4225	s9i10	Discrete	No	20.1742409178966	1day
4226				20.352938898051	2day
4227				18.6315161902255	3day
4228				20.4945997718382	4day
4229	s9i11	Discrete	No	49.7495707141584	0day
4230				50.2504292858416	1day
4231	s9i12	Discrete	No	33.6510150415852	0day
4232				33.7852664013333	1day
4233				32.5637185570615	2day
4234	s9i13	Discrete	No	48.978229827026	0day
4235				51.021770172974	1day
4236	s9i14	Discrete	No	17.0401659609663	0day
4237				16.8599781886652	1day
4238				16.5602914985955	2day
4239				16.4439767125204	3day
4240				16.5201291777779	4day
4241				16.5754584614747	5day
4242	s9i15	Discrete	No	25.5946243928773	0day
4243				25.0206813568169	1day
4244				24.4670500742127	2day
4245				24.917644176093	3day
4246	s9i16	Discrete	No	20.43418615172050	0day
4247				20.0073588944848	1day
4248				19.0822862341381	2day
4249				19.383850120131	3day
4250				21.0923185995256	4day
4251	s9i17	Discrete	No	33.7111582611122	0day
4252				33.2365920965876	1day
4253				33.0522496423002	2day
4254	s9i18	Discrete	No	100	0day
4255	s9i19	Discrete	No	20.4037963450583	0day
4256				19.2038632782016	1day
4257				19.6922062295146	2day
4258				19.893742593461	3day
4259				20.8064098248794	4day
4260	s9i20	Discrete	No	19.703605056562	0day
4261				19.9811051289389	1day
4262				20.711216080901	2day
4263				19.0682373186649	3day
4264				20.535836415839	4day
4265	s9i21	Discrete	No	16.6717762645892	0day
4266				16.6002639249206	1day

4267			16.7804272591996	2day
4268			16.7510667527953	3day
4269			16.6674778697554	4day
4270			16.5289969287399	5day
4271	s9i22	No	24.7397275724249	0day
4272			25.9110075450255	1day
4273			25.1051414223426	2day
4274			24.244123460207	3day
4275	s9i23	No	33.3068233133297	0day
4276			33.2607293852405	1day
4277			33.4324473014298	2day
4278	s9i24	No	16.88822456758	0day
4279			16.9421456827849	1day
4280			16.7340027913694	2day
4281			16.6977396085297	3day
4282			15.9217188390289	4day
4283			16.8161685215293	5day
4284	s9i25	No	100	0day
4285	s9i26	No	25.4564205500269	0day
4286			23.6478139101136	1day
4287			25.684989919221	2day
4288			25.2107756206385	3day
4289	s9i27	No	25.1093660806173	0day
4290			24.63782233148548	1day
4291			25.3847789148481	2day
4292			24.8680316896798	3day
4293	s9i28	No	24.9288550437679	0day
4294			25.0126001785513	1day
4295			25.854530076163	2day
4296			24.2039917700846	3day
4297	s9i29	No	100	0day
4298	s9i30	No	24.7332697952217	0day
4299			25.0027841587126	1day
4300			24.9801954829072	2day
4301			25.2837505631585	3day
4302	s10i11	No	100	0day
4303	s10i12	No	24.709330574889	0day
4304	s10i13	No	24.7884085005745	1day
4305			24.7229596017177	2day
4306			25.7793261488179	3day
4307	s10i14	No	32.1731500627775	0day
4308			33.4958518454227	1day
4309			34.3309980917997	2day
4310			25.9287049873675	0day
4311	s10i15	No	24.1632761224835	1day
4312			24.3192481772446	2day
4313			25.58877029045	3day
4314			24.7902604133879	0day
4315	s10i16	No	25.3789236765835	1day
4316			25.2421802441541	2day
4317			24.5886356658745	3day
4318			51.463825906826	0day
4319	s10i17	No	48.536174093174	1day
4320			19.7478745328728	0day
4321	s10i18	No	19.8322376265655	1day
4322			19.9622586646108	2day
4323			20.0575020933521	3day
4324			20.4001270825988	4day
4325			19.4198322592465	0day
4326	s10i19	No	20.0301245087946	1day
4327				

4328				20.1796245168088	2day
4329				19.99709065271169	3day
4330				20.3732828024332	4day
4331	s10i10	Discrete	No	19.9978011847643	0day
4332				19.6091185188266	1day
4333				20.521917963376	2day
4334				19.6347291874498	3day
4335				20.2364331555832	4day
4336	s10i11	Discrete	No	24.330615058586	0day
4337				25.3263487720109	1day
4338				25.1171721306419	2day
4339				25.2258640314912	3day
4340	s10i12	Discrete	No	19.8786509845785	0day
4341				19.5561316553695	1day
4342				20.7510669117093	2day
4343				20.0797588921188	3day
4344				19.734394586224	4day
4345	s10i13	Discrete	No	33.2259185202684	0day
4346				32.8491317336461	1day
4347				33.9249497460855	2day
4348	s10i14	Discrete	No	25.451911888936	0day
4349				25.5066925861314	1day
4350				24.7234217297129	2day
4351				24.3179738282621	3day
4352	s10i15	Discrete	No	19.7907116330839	0day
4353				19.7318270696075	1day
4354				19.8272944189517	2day
4355				20.3947546730297	3day
4356				20.2554122053272	4day
4357	s10i16	Discrete	No	16.5193024440254	0day
4358				16.9912943911429	1day
4359				16.8007367610242	2day
4360				16.7062346752512	3day
4361				16.419915355857	4day
4362				16.5625163726995	5day
4363	s10i17	Discrete	No	49.0052135156286	0day
4364				50.99478664843714	1day
4365	s10i18	Discrete	No	100	0day
4366	s10i19	Discrete	No	100	0day
4367	s10i20	Discrete	No	100	0day
4368				19.8798792355209	0day
4369				19.4089334832743	1day
4370				19.9454314244097	2day
4371				20.126489374343	3day
4372	s10i21	Discrete	No	20.639266482452	4day
4373				49.444268655373	0day
4374	s10i22	Discrete	No	50.5557313444627	1day
4375	s10i23	Discrete	No	100	0day
4376				20.1765302162208	0day
4377				20.072499092579	1day
4378				19.9612723052878	2day
4379				19.9715022704084	3day
4380	s10i24	Discrete	No	19.818196108251	4day
4381	s10i25	Discrete	No	100	0day
4382				34.2251826723757	0day
4383				31.7962507621862	1day
4384	s10i26	Discrete	No	33.978566565438	2day
4385	s10i27	Discrete	No	100	0day
4386				33.2939088075148	0day
4387				33.2391642458409	1day
4388	s10i28	Discrete	No	33.4669269466442	2day
				24.7252911710428	0day

4389				25.2697981795845	1day
4390				25.031239604005	2day
4391				24.9736710453677	3day
4392				25.2657221336552	0day
4393				24.6998130889439	1day
4394				25.0451368202495	2day
4395				24.9893279571513	3day
4396				100	0day
4397				20.016987679386	0day
4398				20.3899748923051	1day
4399				19.8791652262361	2day
4400				20.0286655794848	3day
4401				19.6852066240355	4day
4402				24.5848098822648	0day
4403				25.1613036881335	1day
4404				25.0319654522713	2day
4405				25.2219209773304	3day
4406				25.1131996647443	0day
4407				25.0825185364128	1day
4408				24.8734997763367	2day
4409				24.930782025061	3day
4410				100	0day
4411				16.4030077170352	0day
4412				16.4421663559571	1day
4413				16.8626028591885	2day
4414				16.5981200437473	3day
4415				16.7840186996083	4day
4416				16.9100843244636	5day
4417				50.0547782784406	0day
4418				49.9452217215594	1day
4419				50.1581075500211	0day
4420				49.841892449789	1day
4421				100	0day
4422				24.9985107258792	0day
4423				24.8223063136121	1day
4424				24.9022031146945	2day
4425				25.2769798458142	3day
4426				19.6070677672831	0day
4427				20.0536030246925	1day
4428				20.2820699417966	2day
4429				20.0028680619268	3day
4430				20.054391204301	4day
4431				100	0day
4432				16.7928659184015	0day
4433				16.666268647838	1day
4434				16.4425368425659	2day
4435				16.8364341177655	3day
4436				16.5772208983446	4day
4437				16.6846735750845	5day
4438				19.9644846232274	0day
4439				19.8910296939151	1day
4440				20.002208208087	2day
4441				19.911904730073	3day
4442				20.2303601319758	4day
4443				24.7531027019525	0day
4444				25.3462176315957	1day
4445				25.3351338857131	2day
4446				24.5655457807387	3day
4447				25.1666414557872	0day
4448				24.9630819004603	1day
4449				25.0646466031064	2day

4450					24.8056300406461	3day
4451	s11i16	Discrete	No	19.8282005262178	0day	
4452				19.7339661231049	1day	
4453				20.28102165451710	2day	
4454				19.8957161064174	3day	
4455				20.2610955897429	4day	
4456	s11i17	Discrete	No	100	0day	
4457	s11i18	Discrete	No	33.5848393450229	0day	
4458	s11i19	Discrete	No	33.5799444521086	0day	
4459				32.8352161428685	1day	
4460				49.6527026029597	2day	
4461	s11i20	Discrete	No	50.3472973970403	0day	
4462				25.0116693742031	1day	
4463	s11i21	Discrete	No	25.2944348315142	0day	
4464				25.0379341991021	1day	
4465				24.6559615951806	2day	
4466				50.0713113550401	3day	
4467	s11i22	Discrete	No	49.9286886449599	0day	
4468				33.3679749879232	1day	
4469	s11i23	Discrete	No	33.1669618568497	0day	
4470				33.4650631552272	1day	
4471				16.5217731201147	2day	
4472	s11i24	Discrete	No	16.6455299181372	0day	
4473				16.805482181418	1day	
4474				16.805482181418	2day	
4475				16.6026063965306	3day	
4476				16.8127982068072	4day	
4477				16.611741402685	5day	
4478				19.8248400529943	0day	
4479	s11i25	Discrete	No	20.0792831899234	1day	
4480				19.9628407984467	2day	
4481				20.082421510548	3day	
4482				20.0505938075807	4day	
4483	s11i26	Discrete	No	16.7230897129474	0day	
4484				16.7522514531548	1day	
4485				16.5635399108351	2day	
4486				16.5256619178553	3day	
4487				16.6801907562777	4day	
4488				16.7552662489298	5day	
4489				24.5712572243823	0day	
4490	s11i27	Discrete	No	25.4187370078823	1day	
4491				25.0017873930824	2day	
4492				25.008218374653	3day	
4493	s11i28	Discrete	No	25.048490049451	0day	
4494				25.308988352811	1day	
4495				24.8430896422977	2day	
4496				24.7994319554402	3day	
4497	s11i29	Discrete	No	50.4930275754955	0day	
4498				49.5069724245045	1day	
4499	s11i30	Discrete	No	33.6709470732915	0day	
4500				33.3566472718812	1day	
4501				32.9724056548273	2day	
4502	s12i1	Discrete	No	16.6608460608833	0day	
4503				16.7026791512264	1day	
4504				16.5811520787769	2day	
4505				16.8242640130793	3day	
4506				16.455081518026	4day	
4507				16.7759751780081	5day	
4508	s12i2	Discrete	No	49.9503746749197	0day	
4509				50.049625325250803	1day	
4510	s12i3	Discrete	No	100	0day	

4511	s12i4	Discrete	No	20.3401563222602	0day
4512				20.0075908376642	1day
4513				19.83282521345	2day
4514				20.0234762798249	3day
4515				19.7959513468007	4day
4516	s12i5	Discrete	No	100	0day
4517	s12i6	Discrete	No	50.1366686441137	0day
4518				49.8633313558863	1day
4519	s12i7	Discrete	No	50.2877245338588	0day
4520				49.7122754661412	1day
4521	s12i8	Discrete	No	16.6101476645924	0day
4522				16.8164400454276	1day
4523				16.5454118693362	2day
4524				16.7860805256946	3day
4525				16.6429034842272	4day
4526				16.599016410722	5day
4527	s12i9	Discrete	No	100	0day
4528	s12i10	Discrete	No	16.8917331137539	0day
4529				16.6934557095906	1day
4530				16.5759661861542	2day
4531				16.671591699756	3day
4532				16.5683384654136	4day
4533				16.5989148253317	5day
4534				16.53560987060580	0day
4535	s12i11	Discrete	No	16.5990739198495	1day
4536				16.9917193518245	2day
4537				16.5119661003428	3day
4538				16.7114265044039	4day
4539				16.6502042529736	5day
4540	s12i12	Discrete	No	20.0021740729324	0day
4541				20.0136057341812	1day
4542				19.8663562428541	2day
4543				19.8751545046924	3day
4544				20.2427094453399	4day
4545	s12i13	Discrete	No	25.1363548718177	0day
4546				24.800989060814	1day
4547				25.073778905065	2day
4548				24.9888771623033	3day
4549	s12i14	Discrete	No	33.0018754523819	0day
4550				33.2518081890978	1day
4551				33.7463163885203	2day
4552	s12i15	Discrete	No	100	0day
4553	s12i16	Discrete	No	16.567420753222	0day
4554	s12i17	Discrete	No	16.5767666410015	1day
4555				16.3724602271099	2day
4556				16.7102715458245	3day
4557				16.7862656404575	4day
4558				16.9868151923846	5day
4559				16.7454314629296	0day
4560	s12i18	Discrete	No	16.818716160596	1day
4561				16.5516927395349	2day
4562				16.413198494959	3day
4563				16.8756931167246	4day
4564				16.595268025625	5day
4565				49.8039436469171	0day
4566	s12i19	Discrete	No	50.1960563530829	1day
4567				100	0day
4568	s12i20	Discrete	No	49.8913073758424	0day
4569	s12i21	Discrete	No	50.1086926241576	1day
4570				33.4472380591033	0day
4571	s12i22	Discrete	No		

4572					33.110080939668	1day
4573					33.4519525442167	2day
4574			No	Discrete	16.7222645769602	0day
4575	s12i23				16.6793260922932	1day
4576					16.5849033178338	2day
4577					16.7472320052241	3day
4578					16.5974301371014	4day
4579					16.6688438705874	5day
4580	s12i24	No		Discrete	100	0day
4581	s12i25	No		Discrete	100	0day
4582	s12i26	No		Discrete	16.7597191636488	0day
4583					16.6708013533577	1day
4584					16.8302179222486	2day
4585					16.693499782146	3day
4586					16.5836200323525	4day
4587					16.4621417462464	5day
4588	s12i27	No		Discrete	25.0203942023604	0day
4589					25.0046104231577	1day
4590					25.0824822235315	2day
4591					24.8925131509504	3day
4592	s12i28	No		Discrete	100	0day
4593	s12i29	No		Discrete	20.0662279624492	0day
4594					19.8842227344589	1day
4595					19.9704141543266	2day
4596					19.9476929725591	3day
4597					20.1314421762062	4day
4598	s12i30	No		Discrete	25.0665333062598	0day
4599					25.14378931247	1day
4600					24.7444617954056	2day
4601					25.0452155858646	3day
4602	s13i11	No		Discrete	33.7843601787583	0day
4603					33.1392896635546	1day
4604					33.0763501576871	2day
4605	s13i12	No		Discrete	19.2567244088338	0day
4606					20.2274587291466	1day
4607					19.9594952048417	2day
4608					20.9772196195892	3day
4609					19.5791020375887	4day
4610	s13i13	No		Discrete	25.5636109607486	0day
4611					24.769849864681	1day
4612					24.7209103479679	2day
4613					24.9456288266025	3day
4614	s13i14	No		Discrete	15.4864737656771	0day
4615					16.4028959991805	1day
4616					17.2961270112138	2day
4617					17.2009774318754	3day
4618					16.7371101407013	4day
4619					16.8764156513519	5day
4620	s13i15	No		Discrete	34.4947068120996	0day
4621					32.3563887468078	1day
4622					33.1148904410926	2day
4623	s13i16	No		Discrete	100	0day
4624	s13i17	No		Discrete	24.5682527637128	0day
4625					24.9427197176576	1day
4626					25.1100670184176	2day
4627					25.3789605002119	3day
4628	s13i18	No		Discrete	17.3254359466664	0day
4629					16.3114979930938	1day
4630					16.8406369977488	2day
4631					16.2577884041576	3day
4632					17.16043508885692	4day

4633				16.10420555697641	5day
4634	s13i19	Discrete	No	19.5773522795748	0day
4635				20.0284173066634	1day
4636				20.2648655402661	2day
4637				20.4174660045144	3day
4638				19.7118988689813	4day
4639	s13i10	Discrete	No	100	0day
4640	s13i11	Discrete	No	32.5160314226718	0day
4641				34.7645120396877	1day
4642				32.7194565376405	2day
4643	s13i12	Discrete	No	100	0day
4644	s13i13	Discrete	No	33.2727136672847	0day
4645				31.9625100202862	1day
4646				34.764763124291	2day
4647	s13i14	Discrete	No	19.4101420852409	0day
4648				19.6130173736257	1day
4649				20.3771959244832	2day
4650				20.4029522247491	3day
4651				20.196692391901	4day
4652	s13i15	Discrete	No	33.3541383652236	0day
4653				33.0639727845698	1day
4654				33.5818888502066	2day
4655	s13i16	Discrete	No	100	0day
4656	s13i17	Discrete	No	20.1599215374563	0day
4657				20.2302699663168	1day
4658				20.5951104751183	2day
4659				19.2907957130637	3day
4660	s13i18	Discrete	No	19.723902306045	4day
4661				20.3859643519045	0day
4662				20.0550968162745	1day
4663				19.9457735782193	2day
4664				19.7487423954827	3day
4665				19.864422858119	4day
4666	s13i19	Discrete	No	24.6284227217456	0day
4667				25.3740026169431	1day
4668				24.7166407416449	2day
4669	s13i20	Discrete	No	25.2809339196624	3day
4670				49.8764517951451	0day
4671	s13i21	Discrete	No	50.1235482048549	1day
4672				50.5103305891974	0day
4673	s13i22	Discrete	No	49.4896994108026	1day
4674				24.3495915086827	0day
4675				25.1631426169768	1day
4676				25.3443925259489	2day
4677	s13i23	Discrete	No	25.1428733483916	3day
4678				50.8778278514871	0day
4679				49.1221721485129	1day
4680	s13i24	Discrete	No	33.3581051489168	0day
4681				34.1206393441405	1day
4682	s13i25	Discrete	No	32.5212555069427	2day
4683				24.29078266515	0day
4684				25.443032261742	1day
4685				24.8454764743281	2day
4686	s13i26	Discrete	No	25.4207076343478	3day
4687				16.9853976339351	0day
4688				16.9911394381202	1day
4689				16.4216043672498	2day
4690				16.9373189394868	3day
4691				15.9514456045149	4day
4692				16.7130940166932	5day
4693	s13i27	Discrete	No	16.6096072563402	0day

4694			16.7268349713865	1day
4695			16.7330925535356	2day
4696			17.0890928861813	3day
4697			16.2324506508731	4day
4698			16.6089217116832	5day
4699		No	25.1701442450244	0day
4700	s13i28	Discrete	24.1112110728458	1day
4701			25.4229552182498	2day
4702			25.29568946388	3day
4703		No	25.4146466849593	0day
4704	s13i29	Discrete	23.26200282286	1day
4705			25.6809098535669	2day
4706			25.6424406386139	3day
4707	s13i30	Discrete	25.3657228523684	0day
4708			25.5825405826895	1day
4709			24.7618101260084	2day
4710			24.2899264389337	3day
4711	s14i1	Discrete	25.172676387861	0day
4712		No	25.4442283753494	1day
4713			24.4279856352212	2day
4714			24.9551096506433	3day
4715	s14i2	Discrete	100	0day
4716	s14i3	Discrete	19.8057215884931	0day
4717		No	20.3655987857305	1day
4718			19.4744255480393	2day
4719			20.2503227585013	3day
4720			20.1039313092358	4day
4721	s14i4	Discrete	20.0643156184251	0day
4722		No	19.8270607863493	1day
4723			20.1311711851774	2day
4724			19.8946032175081	3day
4725			20.0828492125401	4day
4726	s14i5	Discrete	49.7515901296345	0day
4727		No	50.2484098703655	1day
4728	s14i6	Discrete	25.6674300738416	0day
4729		No	24.892322401378	1day
4730			25.3904112477293	2day
4731			24.049836277051	3day
4732	s14i7	Discrete	100	0day
4733	s14i8	Discrete	16.8485399729926	0day
4734		No	16.5730899045306	1day
4735			16.5977980221132	2day
4736			17.1306792640667	3day
4737			16.0349274971118	4day
4738			16.8149653391851	5day
4739	s14i9	Discrete	25.0282554406215	0day
4740		No	24.910936015759	1day
4741			25.0448072172452	2day
4742			25.0159437405575	3day
4743		No	25.1073643476651	0day
4744	s14i10	Discrete	25.3114120493059	1day
4745			25.3712881701324	2day
4746			24.2099354328966	3day
4747	s14i11	Discrete	20.2416589111749	0day
4748		No	19.3709421742428	1day
4749			20.2677602480076	2day
4750			19.9464112241951	3day
4751			20.1732274423796	4day
4752	s14i12	Discrete	100	0day
4753	s14i13	Discrete	33.8839980832046	0day
4754		No	33.6848841088499	1day

4816	s15i3	Discrete	No	25.0815428494156	0day
4817				24.3119813526735	1day
4818				25.0534289864396	2day
4819				25.5530468294713	3day
4820	s15i4	Discrete	No	16.6608576135675	0day
4821				16.6716770365951	1day
4822				16.5772843325797	2day
4823				16.4826111925709	3day
4824				16.7183456641195	4day
4825				16.8892241605674	5day
4826	s15i5	Discrete	No	24.98277427796	0day
4827				24.9370498651636	1day
4828				24.7806070106176	2day
4829				25.2995688764228	3day
4830	s15i6	Discrete	No	24.825731170875	0day
4831				24.9416647519405	1day
4832				25.1416492081006	2day
4833				25.0909548690839	3day
4834	s15i7	Discrete	No	19.8889614562727	0day
4835				20.0378788668778	1day
4836				19.9351655798317	2day
4837				19.7839255297561	3day
4838				20.3540684672617	4day
4839	s15i8	Discrete	No	16.756665376735	0day
4840				16.5982929183591	1day
4841				16.631929187373	2day
4842				16.6845661383502	3day
4843				16.6928490850745	4day
4844				16.6356335627439	5day
4845	s15i9	Discrete	No	20.3926243998156	0day
4846				19.8818359023961	1day
4847				19.7732495981887	2day
4848				20.0241980925751	3day
4849				19.9280920070244	4day
4850	s15i10	Discrete	No	24.9577754057937	0day
4851				25.3806174605297	1day
4852				24.9195121445653	2day
4853				24.7420949891113	3day
4854	s15i11	Discrete	No	100	0day
4855	s15i12	Discrete	No	24.885632965322	0day
4856				25.3609367580239	1day
4857				24.6637464354979	2day
4858				25.0896775099459	3day
4859	s15i13	Discrete	No	24.9718232046183	0day
4860				24.9439971348239	1day
4861				25.1332004803869	2day
4862				24.9509791801708	3day
4863	s15i14	Discrete	No	33.2548091375485	0day
4864				33.2257440064867	1day
4865				33.519446859648	2day
4866	s15i15	Discrete	No	49.3360985010754	0day
4867				50.6639014989246	1day
4868	s15i16	Discrete	No	50.3224683252402	0day
4869				49.6775316747598	1day
4870	s15i17	Discrete	No	20.0104849498329	0day
4871				20.2260934921838	1day
4872				19.7868801411156	2day
4873				19.9599437753887	3day
4874				20.016597641479	4day
4875	s15i18	Discrete	No	19.8556577575736	0day
4876				19.8525836003155	1day

4877				19.8747409006715	2day
4878				20.2235656179812	3day
4879				20.1934612054581	4day
4880	s15i19	Discrete	No	100	0day
4881	s15i20	Discrete	No	100	0day
4882	s15i21	Discrete	No	49.6865316894953	0day
4883				50.314683105047	1day
4884	s15i22	Discrete	No	25.1200539645509	0day
4885				24.8445094209671	1day
4886				24.733283707337	2day
4887				25.302152907145	3day
4888	s15i23	Discrete	No	100	0day
4889	s15i24	Discrete	No	20.3563137589761	0day
4890				20.0304443854075	1day
4891				19.8641880007752	2day
4892				19.7407295947947	3day
4893				20.0083242600465	4day
4894				25.3384090729936	0day
4895	s15i25	Discrete	No	24.8439812490102	1day
4896				25.0575343709012	2day
4897				24.7600753070951	3day
4898	s15i26	Discrete	No	49.6123852394527	0day
4899				50.3876147605474	1day
4900				25.0158436513236	0day
4901	s15i27	Discrete	No	25.1732133591958	1day
4902				24.9329871756836	2day
4903				24.8779558137971	3day
4904	s15i28	Discrete	No	19.7675580760376	0day
4905				20.2278062237779	1day
4906				20.0001522425419	2day
4907				19.9475246551608	3day
4908				20.0569588024818	4day
4909	s15i29	Discrete	No	33.1772651446532	0day
4910				32.949929402175	1day
4911				33.8728054531718	2day
4912	s15i30	Discrete	No	100	0day
4913	s16i1	Discrete	No	16.5936324344704	0day
4914				16.6106385501188	1day
4915				16.5877404753719	2day
4916				16.6156717579398	3day
4917				16.8930265743158	4day
4918				16.6992902077832	5day
4919				33.0756619367428	0day
4920	s16i2	Discrete	No	33.0704012172725	1day
4921				33.8539368459847	2day
4922	s16i3	Discrete	No	100	0day
4923	s16i4	Discrete	No	25.1139265071439	0day
4924				25.3561382718484	1day
4925				24.8113464942399	2day
4926				24.7185887267679	3day
4927	s16i5	Discrete	No	33.2922070952943	0day
4928				33.4531862288454	1day
4929				33.2546066758603	2day
4930	s16i6	Discrete	No	16.6461994536727	0day
4931				16.7476112776863	1day
4932				16.8098644424134	2day
4933				16.4943905829309	3day
4934				16.5231950110724	4day
4935				16.778732322243	5day
4936	s16i7	Discrete	No	16.8295554851048	0day
4937				16.6423517276496	1day

4938				16.6562872931065	2day
4939				16.7132527812413	3day
4940				16.5236838276435	4day
4941				16.6348688852542	5day
4942		No	Discrete	25.3359364198866	0day
4943	s16i18			25.0019669411516	1day
4944				25.0713679297414	2day
4945				24.5907287092204	3day
4946	s16i19	No	Discrete	20.031412718053	0day
4947				20.0895267318168	1day
4948				19.8187588003112	2day
4949				20.1684374125762	3day
4950				19.8918643334906	4day
4951	s16i10	No	Discrete	100	0day
4952	s16i11	No	Discrete	33.256269096229	0day
4953				33.2370454611706	1day
4954				33.5066854392064	2day
4955	s16i12	No	Discrete	33.4260563078564	0day
4956				33.3416995623719	1day
4957				33.2322441297717	2day
4958	s16i13	No	Discrete	50.4734863890924	0day
4959				49.5265136109076	1day
4960	s16i14	No	Discrete	25.2492480288552	0day
4961				25.0802420282711	1day
4962				24.7325659500059	2day
4963				24.9379439928678	3day
4964	s16i15	No	Discrete	24.9800431348046	0day
4965				25.064121649764	1day
4966				24.8510901586629	2day
4967				25.1047450568023	3day
4968	s16i16	No	Discrete	25.0862565466538	0day
4969				25.1738641109171	1day
4970				24.7585534055466	2day
4971				24.9813259368825	3day
4972	s16i17	No	Discrete	20.1161913699956	0day
4973				19.9344362876642	1day
4974				20.1277467542339	2day
4975				19.9528498101542	3day
4976				19.8687575779521	4day
4977	s16i18	No	Discrete	16.6214788376704	0day
4978				16.5495790886637	1day
4979				16.7520542921397	2day
4980				16.5038398120083	3day
4981				16.9406399874884	4day
4982				16.6324079820561	5day
4983	s16i19	No	Discrete	19.9301280479214	0day
4984				20.0078201325536	1day
4985				19.9653160107671	2day
4986				20.0917130181185	3day
4987				20.0050227906569	4day
4988	s16i20	No	Discrete	33.1418791735746	0day
4989				33.2909451094853	1day
4990				33.56717571694	2day
4991	s16i21	No	Discrete	33.1183034995397	0day
4992				33.3288227927181	1day
4993				33.5528737077422	2day
4994	s16i22	No	Discrete	19.8109845794123	0day
4995				19.7130928392669	1day
4996				20.4154461239189	2day
4997				20.0922564831867	3day
4998				19.9682199742152	4day

4999						16.7001485886027				0day
5000						16.4842473589228				1day
5001						16.5431666777957				2day
5002						16.7118214121692				3day
5003						16.8723228091504				4day
5004						16.6882931533592				5day
5005	s16i24	Discrete	No			33.3762569303611				0day
5006						33.668814688766				1day
5007						22.9549283808729				2day
5008	s16i25	Discrete	No			33.4037903461296				0day
5009						33.52311437762378				1day
5010						33.0730658776325				2day
5011	s16i26	Discrete	No			33.2210342813215				0day
5012						33.1082500657847				1day
5013						33.6707156528937				2day
5014	s16i27	Discrete	No			100				0day
5015	s16i28	Discrete	No			16.8040212330651				0day
5016						16.8457169123369				1day
5017						16.5315906700007				2day
5018						16.6893989896376				3day
5019						16.5106749128474				4day
5020						16.6185972821123				5day
5021	s16i29	Discrete	No			49.6971089269698				0day
5022						50.3028910730302				1day
5023	s16i30	Discrete	No			33.5494542513609				0day
5024						33.288890443251				1day
5025						33.1616553053881				2day
5026										
5027										
5028										
5029						External Files				*
5030										*
5031										*
5032	ID	Type			File Name				Prompt	
5033	(null)	Shift			simulationdata.xlsx					
5034	(null)	Shift			ORcalendar.pmcal					
5035										

VITA AUCTORIS

NAME: Justin Britt
PLACE OF BIRTH: Ottawa, Ontario, Canada
YEAR OF BIRTH: 1989
EDUCATION: BSc (General) in Physics, University of Winnipeg, 2010
BS in Industrial Engineering, Louisiana Tech University, 2014