# Table of Contents

**Introduction** .................................................................................................................. 3

**Modeling Assumptions** .................................................................................................. 3
   Assumptions ....................................................................................................................... 3

**Event Graph** .................................................................................................................. 4

**Verbal Event Graph Descriptions** .................................................................................. 7
   Arrivals and Departures ................................................................................................. 7
   Trail and Lift Selection ..................................................................................................... 8
   Ski Lifts ............................................................................................................................ 8
   Initialization ..................................................................................................................... 8
   Data Used ......................................................................................................................... 9

**Model Description** ....................................................................................................... 9
   Starting the Run ................................................................................................................ 9
   Lift Section Breakdown .................................................................................................. 9
      Variables Used ............................................................................................................ 9
      Start Lifts .................................................................................................................... 9
      Load Lifts .................................................................................................................... 10
     Unload Lifts ............................................................................................................... 10
      Queue .......................................................................................................................... 10
      Decision Section Breakdown..................................................................................... 10
      PRCALCULATOR ........................................................................................................ 10

**Running the Simulation** ............................................................................................... 11
   Needed Data .................................................................................................................... 11
   PROBS.DAT ................................................................................................................... 11
   DELAY.DAT .................................................................................................................... 11
   LIFTS.DAT ..................................................................................................................... 11
   Arguments to the simulation run .................................................................................. 11

**Data Collection & Costs of Obtaining Information** .................................................... 12

**Run Control Strategy** ................................................................................................ 13
   Initializing the simulation run ....................................................................................... 13
Addressing Warm Up ........................................................................................................................................... 13

Experimental Design & Sample Sensitivity Analysis ....................................................................................... 13

Arrivals and Departures ......................................................................................................................................... 14

Trailheads, Lifts, and Delay Probabilities ............................................................................................................... 16

Study Results .......................................................................................................................................................... 17

Project Management ............................................................................................................................................... 18
Introduction
Currently, many ski resorts around the world do not operate at maximum efficiency. One way we can help optimize their operation is by modeling their ski lift systems, where large costs are frequently incurred by inefficient operation practices. These practices, which result in wasted operations time, include, but are not limited to:

- Selecting which lifts to open
- Selecting which lifts to close down during the off-season
- Choosing when to build new paths or upgrade certain ski lifts.

If the network of lifts and slopes are configured for the most efficient transport of customers, ski resort owners will experience dramatic cost savings in operations, as well as rising customer satisfaction.

We present a simulation solution to help with these kinds of optimization tasks. Users of our system can configure a simulated ski resort and see how it reacts to varying arrival rates, lift statistics, and more. This document will explain how to run a simulation through an Excel interface, and provide some sample output analysis.

In the simulation, skiers will be travelling to and from various nodes on the resort. These nodes can correspond with the resort base (from where departures occur), a trailhead (the top of a ski run), or a lift base (where queues wait). In our system, the nodes are identified by number, as explained later. We consider skiers to be traveling from one node to another at any given time in the system, and allow the simulation user to tweak the precise delay times of slopes and lift qualities to their needs.

One simulation run equates to a full day at the resort. The simulated day is 10 hours long, ending when the simulation clock reaches 600 minutes or when no further events are scheduled.

Modeling Assumptions

Assumptions

- **Ski lifts accurately portrayed as always running, regardless of queue status** – This is the way resorts work; modeling lifts as service queues is faulty.
- **No arrivals within two hours of resort closure** – We assume the number of people who would arrive at the end of the day is negligible for all practical purposes. In addition, this is an added precaution to make sure the simulation terminates sanely (we want to prevent results that show skiers leaving the resort long after it has closed)
- **Lifts do not break down** – Lifts tend to “break down” or stop often for short periods of time; if necessary, this time can be worked into the ski lift’s travel time. In addition, any serious breakdowns shouldn’t be expected (for obvious safety reasons) so modeling them as such would be faulty.
- **All skiers are homogeneous in their choices** – We concluded modeling different kinds of skiers in the system as unnecessary. Data gathering processes for this kind of differentiation is almost impossible, as there is no way for data collectors to easily different between different skiers
(beginner, casual, advanced). Furthermore, skier differentiation is already implicit data that can be observed from the use of ski lifts and trails of differing difficulty.

Event Graph
Figure 1 is a complete image of the event graph that underlies our simulation model. It provides a clear picture of everything that happens when our model is run. The following tables detail code-level state changes and edge conditions. Information corresponding to a “one-sentence-per-edge” description (i.e. a verbal event graph) can be found in a later section.
Figure 1: Conceptual Event Graph
<table>
<thead>
<tr>
<th>Node</th>
<th>Parameters</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>ARRIVE[1-4], DEPART[1-4]</td>
<td>NUMLIFTS=10, LID=0, CLOAD=0, ARRCOUNT=0, DEP=0</td>
</tr>
<tr>
<td>Collect Statistics</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Load Data</td>
<td>I, J</td>
<td>TPROBS[i;j]=DISK(PROB.DAT;0), TDELAY[i;j]=DISK(DELAY.DAT;0)</td>
</tr>
<tr>
<td>Start Lifts</td>
<td>LID</td>
<td>TIA[LID]=DISK(LIFTS.DAT;0), CAPAC[LID]=DISK(LIFTS.DAT;0), LDEST[LID]=DISK(LIFTS.DAT;0), QUEUE[LID]=0</td>
</tr>
<tr>
<td>Skier Arrives</td>
<td>-</td>
<td>ARRCOUNT = ARRCOUNT + (CLK&lt;480), ARRIV[0] = (ARRIV[1]<em>ERL{1}</em>(CLK&lt;120)) + (ARRIV[2]<em>ERL{1}</em>(CLK&lt;240&amp;CLK&gt;=120)) + (ARRIV[3]<em>ERL{1}</em>(CLK&lt;360&amp;CLK&gt;=240)) + (ARRIV[4]<em>ERL{1}</em>(CLK&lt;480&amp;CLK&gt;=360))</td>
</tr>
<tr>
<td>Base</td>
<td>-</td>
<td>R=RND, DEPRATE[0] = DEPRATE[1]<em>(CLK&lt;120) + DEPRATE[2]</em>(CLK&lt;240&amp;CLK&gt;=120) + DEPRATE[3]<em>(CLK&lt;360&amp;CLK&gt;=240) + DEPRATE[4]</em>(CLK&lt;480&amp;CLK&gt;=360)</td>
</tr>
<tr>
<td>Depart</td>
<td>-</td>
<td>DEP=DEP+1</td>
</tr>
<tr>
<td>Queue</td>
<td>LID</td>
<td>QUEUE[LID]=QUEUE[LID]+1</td>
</tr>
<tr>
<td>Load Lift</td>
<td>LID</td>
<td>CLOAD=MIN{QUEUE[LID];CAPAC[LID]}, QUEUE[LID]=QUEUE[LID]-CLOAD</td>
</tr>
<tr>
<td>Unload Lift</td>
<td>CLOAD, LID</td>
<td>CLOAD=CLOAD-1</td>
</tr>
<tr>
<td>Probability Calc</td>
<td>LID, CHOICE, R, TOTAL</td>
<td>TOTAL=TOTAL+TPROBS[LID;CHOICE]</td>
</tr>
<tr>
<td>Trailhead</td>
<td>LID, CHOICE</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delay on Edge</th>
<th>Node From</th>
<th>Node To</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>Collect Statistics</td>
<td>Collect Statistics</td>
<td>.5</td>
</tr>
<tr>
<td>t₂</td>
<td>Skier Arrives</td>
<td>Skier Arrives</td>
<td>MIN{ARRIV[0];20}</td>
</tr>
<tr>
<td>t₃</td>
<td>Trailhead</td>
<td>Probability Calc</td>
<td>TDELAY[LID;CHOICE]</td>
</tr>
<tr>
<td>t₄</td>
<td>Trailhead</td>
<td>Queue</td>
<td>TDELAY[LID;CHOICE]</td>
</tr>
<tr>
<td>t₅</td>
<td>Trailhead</td>
<td>Base</td>
<td>TDELAY[LID;CHOICE]</td>
</tr>
<tr>
<td>t₆</td>
<td>Load Lift</td>
<td>Unload Lift</td>
<td>TDELAY[LID+1;LDEST[LID]]</td>
</tr>
<tr>
<td>t₇</td>
<td>Load Lift</td>
<td>Load Lift</td>
<td>TIA[LID]</td>
</tr>
<tr>
<td>Condition on Edge</td>
<td>Node From</td>
<td>Node To</td>
<td>Formula</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------</td>
<td>------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Rows Remaining</td>
<td>Load Data</td>
<td>Load Data</td>
<td>(I \leq 19 &amp; J &lt; 19)</td>
</tr>
<tr>
<td>All rows loaded and</td>
<td>Load Data</td>
<td>Load Data</td>
<td>(I &lt; 19 &amp; J \geq 19)</td>
</tr>
<tr>
<td>columns remaining</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All data loaded</td>
<td>Load Data</td>
<td>Start Lifts</td>
<td>(I \geq 19 &amp; J = I)</td>
</tr>
<tr>
<td>All data loaded</td>
<td>Load Data</td>
<td>Skier Arrives</td>
<td>(I \geq 19 &amp; J = I)</td>
</tr>
<tr>
<td>All lifts haven’t started</td>
<td>Start Lifts</td>
<td>Start Lifts</td>
<td>(LID &lt; NUMLIFTS)</td>
</tr>
<tr>
<td>More than 2 hours</td>
<td>Skier Arrives</td>
<td>Skier Arrives</td>
<td>CLK &lt; 480</td>
</tr>
<tr>
<td>before closing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skier departs</td>
<td>Base</td>
<td>Depart</td>
<td>((R &lt; DEPRATE[0]) \mid (CLK &gt; 480))</td>
</tr>
<tr>
<td>Skier doesn’t depart</td>
<td>Base</td>
<td>Probability Calc</td>
<td>((R \geq DEPRATE[0]) &amp; (CLK &lt; 480))</td>
</tr>
<tr>
<td>Skiers on lift</td>
<td>Load Lift</td>
<td>Unload Lift</td>
<td>CLOAD &gt; 0</td>
</tr>
<tr>
<td>Skiers to unload</td>
<td>Unload Lift</td>
<td>Unload Lift</td>
<td>CLOAD &gt; 0</td>
</tr>
<tr>
<td>TOTAL is less than or</td>
<td>Probability Calc</td>
<td>Probability Calc</td>
<td>TOTAL \leq R</td>
</tr>
<tr>
<td>equal to R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL is greater than</td>
<td>Probability Calc</td>
<td>Trailhead</td>
<td>TOTAL &gt; R</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trail leads to trailhead</td>
<td>Trailhead</td>
<td>Probability Calc</td>
<td>CHOICE &gt; 10</td>
</tr>
<tr>
<td>Trail leads to base</td>
<td>Trailhead</td>
<td>Base</td>
<td>CHOICE == 0</td>
</tr>
<tr>
<td>Trail leads to lift</td>
<td>Trailhead</td>
<td>Queue</td>
<td>((CHOICE = 0) &amp; (CHOICE \leq 10))</td>
</tr>
</tbody>
</table>

**Verbal Event Graph Descriptions**

Our event graph is broken down into 5 sections: Arrivals and Departures, Trail and Lift Selection, Ski Lifts, Initialization, and Data Used. The following represents a “one-sentence-per-edge” English description of the event graph outlined in the above section.

**Arrivals and Departures**

The Arrivals and Departures section of the event graph has people arrive at a random rate whose mean depends upon the time of day and has percentage of the skiers who reach the base leave the resort. This percentage is dependent upon the time of the day. The first Skier Arrives event is called by arrow at the bottom which comes from the Load Data event. The Skier Arrives event will call itself so long as there are 2 or more hours before the resort closes. The time between arrivals of skiers is an exponential distribution with an upper bound and a time dependent mean. When a skier arrives, they are sent to the ski resort Base. A skier may reach the Base by arriving at the resort or taking a ski trail which ends at the Base; this is represented by the arrow coming into Base from the right that is called by the Trailhead event. Anytime a skier reaches the Base, a time dependent probability calculation determines if they will depart or continue skiing. If the skier departs they go to the Depart event and leave the resort. If the skier doesn’t depart, the arrow leaving Base to the right sends them to the Probability Calc event to determine which lift or run they take from the base.
Trail and Lift Selection
The Trail and Lift Selection section of the event graph is the real engine behind the model; skiers enter this section having arrived at a trailhead and are routed, based on the architecture of the resort, to their next trail or lift. Skiers are sent to Probability Calc from lifts (the arrow entering Probability Calc from the bottom comes from the Unload Lift event), the ski resort Base (the arrow entering Probability Calc from the left comes from the Base event, or from other trails (the arrow from Trailhead to Probability Calc). The Probability Calc event checks the probability that the skier will go from the current trailhead to every other trailhead and recursively calls itself until it finds where the skier will go next. Once the destination trailhead is found, Probability Calc calls the Trailhead event with the origin and destination trailhead. If the destination trailhead is the base, the skier is sent by the arrow leaving the Trailhead event to the left to the Base event after the amount of time that it takes to get from the origin trailhead to the base. If the destination trailhead is a lift, the skier is sent by the arrow leaving downwards from Trailhead event to the Queue event after the amount of time that it takes to get from the origin trailhead to the selected lift. If the destination trailhead is top of another trail, the skier is sent from Trailhead event back to the Probability Calc event after the amount of time that it takes to get from the origin trailhead to the destination trailhead.

Ski Lifts
The Ski Lifts section of the event graph comprises the ski lifts and queues for the lifts. If a skier takes a trail that leads to a lift, then the Trailhead event calls the Queue event with the arrow that enters Queue from above. The skier is placed into the queue for the lift as specified by the parameter CHOICE. The lifts are started by the Start Lifts event using the arrow that enters Load Lift from the bottom. Load Lift simulates a chair arriving for skiers to load; the event reoccurs with the same frequency that chairs arrive on the lift it models. The Load Lift event sends the minimum of the capacity of a chair on the lift and the number of people in the queue up to the Unload Lift event after the amount of time that it takes to ride the lift to the top. Unload Lift sends one skier to the Probability Calc event to determine which ski run or lift the skier will take next. If there are still skiers to unload, Unload Lift will call itself until all skiers are unloaded.

Initialization
The initialization section of the Event Graph is responsible for loading data from DAT files, starting the arrivals process, starting the ski lifts, and collecting statistics on the run. The Run event has the Load Data event begin inputting data from the zeroeth row and zeroeth column. This data comes from two dat files that are loaded into two separate 2-D matrices (see the Data Used section for details on what data is loaded). If there are still rows of data left to be inputed, the Load Data event is called again with the row to be inputed incremented by 1. If all rows have been inputed and there are still columns to be inputed, then the Load Data event is called again with the column to be inputed incremented by one and the row set to zero. Once all rows and columns of data have been inputed, two events occur. First, the arrow leaving Load Data on the left schedules a Skier Arrives event. Second, the Start Lifts event is called with parameter 1. This starts lift 1 by having the arrow leaving the Start Lifts event call a Load Lift event for lift 1. Start Lifts will call itself to start the next lift until all lifts have been started. The run event
also calls the Collect Statistics event to record the number of arrivals, departures, and length of queues for lifts. This event calls itself every half minute of simulation time to continue recording statistics.

Data Used
The mean arrival rate is time dependent. There is a different arrival rate for each 2 hour block of the day. These rates are inputed as parameters when the simulation is run.

The percentage of skiers who leave the resort when they reach the base is time dependent. There is a different percentage of skiers departing for each 2 hour block of the day. These percentages are inputed as parameters when the simulation is run.

The probability that a skier will go from one lift or trail to a different lift or trail is encoded in a matrix of probabilities. This is originally a dat file and is read into a 2-d matrix by the Load Data event. By having probabilities of zero to go from certain trails to other trails, the layout of the lifts and runs can be encoded into this data.

If there is a chance that a skier can go from one trail to another trail, then the time that it takes must be included. The time between any two trails or lifts is listed in a dat file and is read into a 2-d matrix by the Load Data event.

We will follow by describing each of the sections in detail.

Model Description

Starting the Run
RUN reads in the arrivals data to ARR, departure data into DEP, probabilities between trails into TPROBS, the capacity of each lift into CAPAC, and the time to go up a lift into TS. These variables will be defined as we encounter them in the separate subsections.

Lift Section Breakdown

Variables Used
- TS: the array of normal times of service for the lifts
- TIA: the time between chairs arriving for the lifts
- CAPAC: the capacity of a chair for the lifts
- LIFT: the number of the lift that is passed
- C: the actual number of people on the chair
- QUEUE: the number of people in the lines for the lifts

Start Lifts
Start Lifts goes through each lift and starts picking up people. It initially receives an input with the parameter 1. Every time it is called, it sends an edge with the value received to Load Lifts. An edge then loops back to itself with the value passed in plus one, until it is equal to the number of lifts.
Load Lifts
This node occurs every time that a chair is ready to load passengers. It is parameterized to signify the lift that is being used. Every time the node is called, it determines how many people will go on the chair, and subtracts that many people from the queue.

\[
C = \min(\text{QUEUE[LIFT]}, \text{CAPAC[LIFT]}),
\]
\[
\text{QUEUE[LIFT]} = \text{QUEUE[LIFT]} - C
\]

The node calls itself with a delay of the time between chairs. The delay of \( t_8 \) is:

\[
\text{TIA[LIFT]}
\]

If it calculates that people are on the chair, it schedules them to unload after the time it takes to get to the top of the lift. The delay of \( t_7 \) is:

\[
\text{TS[LIFT]}
\]

Unload Lifts
This node occurs when a chair reaches the top of the lift. It is parameterized to receive the lift that the people are unloading from and the number of people on the chair. It unloads one person and sends that person to the trailhead. It calls itself until everyone has been unloaded off the chair.

Queue
This node puts one person into the queue for the lift that is specified by the value passed to it. It does not schedule any other nodes because lifts operate at set time intervals.

Decision Section Breakdown

PRCALCULATOR
From the BASE (LODGE equivalent) or any ski lift (or any ski lift loading area), a user gets his parameters initialized in preparation for a decision.

\[
R = \text{RND}
\]
\[
\text{TOTAL} = 0
\]
\[
A = 0
\]

Starting at 0, the system iterates through all the lift choices (variable A keeps track of this) and PRCALCulates which lift number this user might take by taking a random number from a flat distribution (R) and seeing where it lies on the 0-1 range.

\[
\text{TOTAL} = \text{TOTAL} + \text{TPROBS[A]} \quad \text{until} \quad \text{TOTAL} > R
\]

This decision is based on the set probabilities of each lift (controllable but should be set in data files and populated by RUN into the TPROBS matrix). Setting the probability of a lift to be zero effectively disables it (or hints that that lift is not accessible from this position).

PRCALC passes the lift ID of the lift that was chosen. (This is what A keeps track of during PRCALC.)
Running the Simulation

Needed Data
The simulation needs 3 external .DAT files to run. By using the Excel interface, these files will be created for you at run time. However, you will still need to populate the data in the Excel spreadsheets and understand the necessary variables.

PROBS.DAT
This file contains a 20-by-20 matrix of probabilities. The matrix is organized in an (I,J) format – any given entry in the matrix represents the “path” from a start node (I) to a destination node (J). The value of each entry corresponds to the probability of any given skier traveling to node J when situated at node I. The system recognizes certain ranges of indices to mean specific types of nodes on the mountain. This is the current scheme that the system expects probabilities to be specified for:

<table>
<thead>
<tr>
<th>Node Identifier (row in DAT file)</th>
<th>Type of node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Resort base</td>
</tr>
<tr>
<td>1 through 10</td>
<td>Lifts (namely, the lift bases)</td>
</tr>
<tr>
<td>11 through 19</td>
<td>Actual trailheads</td>
</tr>
</tbody>
</table>

Example: if we wanted to set 15% as the probability a skier would travel from the trailhead at node 11 to the lift base at node 2, we would enter “0.15” in entry (11,2) in the matrix.

DELAY.DAT
This file contains a 20-by-20 matrix of delay (travel) times between trailheads, also indexed by (I,J) pairs. However, instead of probabilities as entries, DELAY.DAT records delay (in minutes) between nodes.

LIFTS.DAT
This table contains data pertaining to the lifts. Each line is laid out as follows:

<table>
<thead>
<tr>
<th>Lift interarrival time</th>
<th>Capacity (number of seats)</th>
<th>Destination of this lift</th>
</tr>
</thead>
</table>

The model can support a maximum of 10 lifts, which means this file can only be ten lines long. The non-operational lifts must be represented by the set (0,0,0) in their data row. We may increase and decrease the number of operational lifts by modifying or resetting data from this file.

Arguments to the simulation run
The model takes in 8 parameters, in order as follows:

ARRIV[1],ARRIV[2],ARRIV[3],ARRIV[4],DEPRATE[1],DEPRATE[2],DEPRATE[3],DEPRATE[4]

The simulated day is divided into five distinct time periods – each lasting two hours. For the first 8 hours, we leave the arrival and departure rates controllable by the user via the above parameters, explained below.
Arrivals are controlled by a random interarrival time for incoming customers. The interarrival time is based on an exponential distribution with parameter $1/\text{ARRIV}[X]$ where $X$ is an integer from 1-4 signifying the time period of the day. $\text{ARRIV}[1-4]$ are the first four parameters to our simulation, which represent the desired mean time between arrivals. To prevent unusually long interarrival times, we impose an arbitrary maximum of 20 minutes between arriving customers.

Departures are dictated in a different manner. We simply request that the parameters $\text{DEPRATE}[X]$ be the probabilities of leaving the resort every time a skier arrives at the ski resort base during time period $X$ in the day.

**Data Collection & Costs of Obtaining Information**

In a real system, the necessary data for our simulation to run would be collected empirically, relying heavily on averages of data over a certain time. The nature of our data – arrivals and departures of skiers, as well as their choices of lifts and trails – lends itself to a high degree of randomness. A useful tool for analyzing the data would be to compare sets of data from the same period of time (i.e. morning, midday, or afternoon) but from different days, and take averages from them.

To estimate the cost for collecting this kind of data, one primarily needs to take into account time value of money, since very little of the data can be electronically stored, perhaps with the exception of skiers’ initial arrival into the resort (by recording times of ticket purchases). A suggested method of data collection and costs follows:

**Arrivals:** In order to be user-friendly, the simulation will give the customer the ability to determine when customers are most likely to arrive. We assume that all arrivals happen within the first three hours of the resort being open. The customer will be able to input how many people arrive throughout a given day. Gathering this information could be done by simply keeping track of ticket purchases.

**Trailhead, lifts, and delay probabilities:** For each of these, fluctuations in data will be, for the most part, random and affected by the whim of individual skiers. Because of this, we propose that direct observation will provide the most reliable data. Observation for periods of one hour each during the morning, midday, and afternoon over a couple days would be sufficient.

**Base (departures):** Our model assumes that if a skier enters the base within an hour of closing time, he or she will depart. It might be plausible to consider not gathering this data directly, by observation. Instead, it could be calculated indirectly by observing decreases in average lift queues and waiting times.

Data collection required to run this model is simply observational data of skier choice behavior on where to travel to. Ideally we would have a low-skill worker with a counting device placed at each node in the resort, monitoring traffic over a few days (in order to get stable data). All costs will essentially be man-hours, which should not exceed a few thousand dollars.
Run Control Strategy

Initializing the simulation run
The model takes in 8 parameters, in order as follows:

ARRIV[1],ARRIV[2],ARRIV[3],ARRIV[4],DEPRATE[1],DEPRATE[2],DEPRATE[3],DEPRATE[4]

The simulated day is divided into five distinct time periods – each lasting two hours. For the first eight hours, we leave the arrival and departure rates controllable by the user via the above parameters, explained below. We assume that during the last two hours of the day, customer arrivals cease and every skier chooses to depart upon reaching the base.

Arrivals are controlled by having a random interarrival time for incoming customers. The interarrival time is based on an exponential distribution with parameter 1/ARRIV[X] where X is an integer from 1-4 signifying the time period of the day. ARRIV[1-4] are the first four parameters to our simulation, which represent the desired mean time between arrivals. To prevent unusually long interarrival times, we impose an arbitrary maximum of 20 minutes between arriving customers.

Departures are dictated in a different manner. We simply request that the parameters DEPRATE[X] be the probabilities of leaving the resort every time a skier arrives at the ski resort base during time period X in the day.

Addressing Warm Up
We do not need to “warm up” our simulation because our simulation runs from the start to the end of a full day – at the beginning of the day, there should not be any customers at the resort. Furthermore, any customers that arrive begin at the base, and not elsewhere in the system.

Experimental Design & Sample Sensitivity Analysis
Using the intuitive Excel spreadsheet interface, you will be able to perform analyses of your current model at a glance. Our system graphs these tracked variables for you throughout the run:

- **ARRCOUNT**: total number of people who have arrived in the system
- **DEP**: total number of people who have departed the system (through the BASE)
- **QUEUE[N]**, where N = [1:10]: queue length of lift(N) at any given time

We present some sample output analysis for you here.
Arrivals and Departures

Plot 1: Departure rate progressively increasing throughout the day

Plot 2: Uniform arrival rate throughout the morning
Plot 3: Departure rate highly concentrated towards the last part of the day

**Plot 1** represents the “default” day, in the sense that it is the graph that our simulation will generate when opened and run without altering any of the delays or probabilities. It is fairly representative of a typical day, and though it does not account for many things, it nevertheless illustrates the increasing and decreasing number of skiers towards the middle and the end of the day, respectively. More generally, if the arrival count (dark blue) and the departure rate (pink) are ever really far apart in the graph, more skiers can be expected to be present at the ski resort, and naturally, queues will be longer.

**Plot 2** accounts for the consideration of differing arrival count rates in the morning hours of a given business day, and shows the results of a uniform arrival rate. (Variables $a_1$ and $a_2$ both set to 0.30.) Without considerable increases in any specific queue length, in turns out that the time of day with the longest queue lengths is delayed a bit, with slightly lower queue times. This is important because in plot 2, the shape of the arrival counts (dark blue) more closely matches the shape of the departure rate curve. Either way, it gives the customer a clearer idea of when the ski resort might need extra employees or extra lifts.

**Plot 3**, on the other hand, demonstrates the effect of a delayed departure rate. A very long queue can be expected during midday to about three-quarters of the way through the day. The queue lengths are very lopsided throughout, with a sharp increase in the morning, and a sharp decrease towards the late afternoon. Although not an ideal situation, it nevertheless reflects a real resort, at least in part, since many skiers generally want to stay until the resort closes so as to maximize the value of their lift tickets.
Even though it is difficult to control the departure rate of customers directly, it is important to keep in mind that values of $d_1$, $d_2$, and $d_3$ less than 0.15 will significantly increase queue lengths.

**Trailheads, Lifts, and Delay Probabilities**  
*(using arrival and departure input data of Plot 1)*

![Graph showing queue lengths](image)

**Plot 4:** Lift capacity reduced to three passengers per lift
Plot 5: Delay times from one trailhead to another reduced by 25%

Running experiments with a variety of probabilities for delay times is a critical way of seeing how and when a queue in a given area might become busy. Plot 4 allows for the possibility that a resort might not be fully equipped with lifts for four passengers. Needless to say, the average queue lengths are definitely higher, and even more so with two-passenger lifts (not pictured). There are many valid considerations regarding where to upgrade a lift’s capacity, and the customer will be able to see where upgrading lift capacity might make the most impact.

Plot 5 takes into consideration the fact that a route going from one trailhead to another trailhead will generally attract more advanced skiers, in addition to being nonstop (i.e., the skier does not have to stop and walk on over to the next lift queue). As such, the delay times can be expected to be lower. Generally, a reduced delay time will only affect the queue lengths associated with that trailhead; queues 1, 2, and 4 are noticeably higher than in other plots.

**Study Results**

The fastest and most effective way to “validate” our model would be to test simulated changes. Say we simulate the closing or a certain trail or lift. The resort could easily test this under the guise of “maintenance” for a day, in order to observe the changes in traffic at ski lifts and see if they match those expected from the simulation.
Project Management

Our project team adhered to a schedule of weekly meetings. By using an online poll to determine the availability of each of our team members, we were able to easily accommodate each of our busy schedules. Through the use of a collaborative wiki at [http://ieor131ski.pbwiki.com/](http://ieor131ski.pbwiki.com/) (similar to Wikipedia), we were able to upload files for the rest of us to see and effectively keep each other informed about the progress of our individual parts.

As a team, we made it a point to determine what we would expect to do at our meetings, as well as what each of us would be expected to do later on. More specifically, before each meeting, we would e-mail each other with suggestions on how to improve our model. As a group, we could then more easily and more objectively determine whether or not it would be beneficial for our product.