CellSim:

Spreadsheet Factory Simulation

John O. McClain
Cornell University
24 Oct 2007

S. C. Johnson Graduate School of Management
Sage Hall, Cornell University
Ithaca NY 14850

Instructions from the CellSim Program File

This is a very simple factory simulator, intended for teaching purposes. You are welcome to use it in any manner, and change it as you see fit. This model comes without any guarantee whatsoever, and is distributed free of charge. It is based on ideas first developed at Cornell in the factory simulation program known as XCELL+, written by Richard Conway, William L. Maxwell and Steven L. Worona, with minor help from me. XCELL+ was the first simulation package with a graphical user interface, and appeared in the early 1980s when even the PC was a new idea.

Note: ***this program uses macros*** If you can't make it work, click on the link below for more directions.

Help with macro security

MODEL: A CellSim model is constructed on the worksheet named "Model" in this Workbook. You cannot make a model on another sheet. However, each time you run a simulation, the model's description is stored with the output. You may resurrect a previously saved model by using the Button "Open a Factory File".

SIMULATION RESULTS are stored on a series of worksheets in the output file:
The **"Inventory"** sheet contains average, ending, maximum and minimum values for the entire stock in each StorageArea.
Individual **Item** data is on a separate sheet with the name of the StorageArea.

The **"Machines"** sheet contains the fraction of time each machine was utilized (running), starved, blocked, or down for repair, the number of units scrapped and the number successfully completed, number of Batches and Lots completed, Setups done, Preemptions that occurred and the number of machine failures.
Individual **Process** data is on a separate sheet with the name of the Machine.

Use any of Microsoft Excel's methods to tabulate or display the results.

USING CellSim

The remaining instructions are in three sections:

CONTROLLING THE SIMULATION; what numbers you have to enter before running a simulation.

RUNNING A SIMULATION; what the buttons do.

MODELING; how to construct or change a model.

CONTROLLING THE SIMULATION

"Run-In Time": the amount of time you want to simulate during a Run-In period, often used to allow the system to move away from its initial state before collecting data.
Run-In Time can be set to zero if you do not want a run-in period.

"Run Length": the amount of time you want to simulate during each repetition of the simulation.

"Repetitions": how many times you want the simulation to repeat after the run-in.

"Random Number Seed": every New Simulation uses the same sequence of random numbers.
To get a different sequence, change the integer in this box. You might want to write down the numbers that you use, for future reference, in case you want to duplicate your runs.

"Animation": Check this box to see the factory change as the simulation runs.

"Step": Check this box to run the simulation one-step-at-a-time.

RUNNING A SIMULATION

"New Runs": sets all StorageAreas to their initial values and uses that information to set all machines to their first states. It then executes a run-in of the specified length, followed by a series of simulation runs using the Run Length and the number of Repetitions specified by the user.

"Continue Runs": adds another set of runs to the same data set, but **without** initialization and run-in. That is, the simulation picks up where it left off rather than starting over.

"Open a Model": Use this to re-open an output file so that a previous run may be continued. Opening the file also restores the model to the one that was in use at the time. After doing this, use "Continue Runs" to continue where you left off, or "New Runs" to do a new run (with a new output file) of that factory model.

"Merge a Model": Use this to expand your model by inserting a model from another output file. After doing this, use "New Runs" to do a new run (with a new output file) of the enalrged factory model.

"Show States" or "Show Model" toggles between a view of the current state of each Machine and StorageArea, and a view of the model that shows all arrows, Machine names and StorageArea names.

"ReDraw" and "FixMe" buttons have nothing to do with the simulation. Use ReDraw if the arrows do not line up. Use FixMe if you get an overflow message pertaining to "DrawingObjects".

MODELING

There are two kinds of elements: Machines and Storage. You can add an element to any empty cell by double-clicking on the cell and responding to the prompts.

MOVING: You may move an existing element by dragging it, Excel style, or by cutting and pasting to an **empty** cell on the same sheet.
After moving an item, press the "ReDraw" button to move the arrows.

COPYING: *YOU MAY NOT COPY* elements using the Excel menu. Instead, double-click or right-click on the machine or StorageArea; a button on the resulting screen is for copying.
Press that button, select an empty cell, and press OK.

MODELING: StorageAreas and Items

StorageAreas have 3 properties: Name, Capacity and a list of Items.
Double-click or right-click on a StorageArea to see or change the properties.

Items have 2 properties: Name and Initial Stock.
Double-click or right-click on a StorageArea, select an Item and click "Edit Item" to see or change the properties.

MODELING: Machines and Processes

Machines have 4 properties: Name, Failure Distribution, Repair Time Distribution, and a list of Processes. Double-click on a Machine to change its properties.

DISTRIBUTION OF TIMES BETWEEN FAILURES: Same instructions as Processing Times, below.

DISTRIBUTION OF REPAIR TIMES: Same instructions as Processing Times, below.

Processes have Many properties: Name, LotSize, BatchSize, Batch Percent Scrap, Unit Percent Scrap, Setup Time, Priority, Controls, 2 Inputs, Normal Output, Scrap Output, and Processing Time Distribution.

Double-click or right-click on a Machine select a Process and click "Edit Process" to see or change the properties.

LOTSIZE: Number of Batches run after each setup. (Integer, >= 0)
LotSize = 0 means the process will run until interrupted or starved.
LotSize > 0 means the process will run until interrupted or until "BatchSize*LotSize" non-defective units have been completed. Only Priority 3 Processes may preempt a Process with LotSize >0. If preempted, the process will restart (including setup) at a later time to finish the Lot.

BATCHSIZE: The number of units produced in each Processing Time. (Integer, >= 1)
All units in a batch are processed simultaneously. Preemption during a Batch is not allowed.
"Scrap" happens in two ways: Entire Batch, or Units Within a Batch. (See Below.)

BATCH PERCENT SCRAP (0 to 100): Example: if 15% are defective, enter 15 (not 0.15).
The computer randomly assigns THE ENTIRE BATCH to either scrap or normal output.

UNIT PERCENT SCRAP (0 to 100): Example: if 15% are defective, enter 15 (not 0.15).
The computer randomly assigns EACH UNIT to either scrap or normal output.

The fraction of units that will be defective depends on both of these parameters. For example, if the Batch Percent Defective is 30% and the Unit Percent Defective is 10%, then 30% will be scrapped through the Batch Scrap, and of the 70% that survive, 10% will be scrapped through the Unit Scrap, for a total of 0.3 + 0.7(0.1) = 0.37 or 37% scrap.

SETUP TIME: Time to prepare for production (Constant, >= 0)

PRIORITY: A Process's priority is either "Normal", "High", or "Preempt During a Lot". A Process may not be interrupted during production of a unit (or a batch, if BatchSize >1). However, when the unit is finished, priorities determine what happens next as follows.

If the current Process has LotSize>0, it will maintain use of the Machine until its Lot is completed, unless a Process that has the highest priority ("Preempt During a Lot") is ready to start. If it is interrupted in this manner, it will regain use of the Machine as soon as possible.

Once the Lot is completed, the Machine will switch to the highest-priority Process that is ready to begin. The search begins ***after*** the current Process. (The current Process moves to the "end of the list" of Processes in its priority class.) Thus, *at the end of a Lot, the machine will "change over" if* ***any*** *other Process of equal or higher Priority is ready to start.*

If no LotSize has been specified for the current Process (i.e. LotSize=0), it will retain use of the machine until it stops normally (runs out of material or is stopped by its Control mechanism, as described below) or until a higher-priority Process is ready to start.

INPUT: A Process may have 0, 1 or 2 inputs. An input may connect to a StorageArea, but not to another machine. Click either "Input:" button, position the cursor on an existing StorageArea, and click "OK". Then select an Item from that StorageArea.
If Input is unassigned, the machine operates with unlimited supply.
If both Inputs are assigned, the machine may not operate unless both Inputs are available.

NORMAL OUTPUT: Same instructions as for INPUT.
If Output is unassigned, the machine operates with unlimited demand; finished units are vaporized.
If the Output StorageArea is full when a process ends, the machine is Blocked and cannot operate.

SCRAP OUTPUT: Same instructions as for NORMAL OUTPUT.

CONTROLS: Click the "Change" button.
To select a StorageArea to host the control, click "Assign Control Point", position the cursor on an existing StorageArea, and click "OK". Then select an Item or just click "OK" again. If no item is selected, the control is based on the sum of all items in the StorageArea.

Put a "Control Level" in the appropriate box.
This determines the quantity at which the control will activate the process.

Select the desired control method.
This determines whether activation occurs when the inventory rises or when it falls.

Select the desired "characteristic" to monitor.
This determines whether activation is based on On-Hand Inventory, or Inventory Position. Inventory position takes into account stock that has been "promised" by a Process that has LotSize>0. Thus, once a Process has begun a setup to deliver that stock, the Position drops. This may deactivate the controls for other Processes that monitor the same stock. This can prevent several Processes from reacting to the same "Call for Inventory".

DISTRIBUTION OF PROCESSING TIMES: Click the "Change Processing Distrib." button.
This brings up another screen that displays the processing time distribution on a graph. You may enter the mean and standard deviation directly in the appropriate boxes. The shape of the distribution may be changed by choosing from several options (Uniform, Shifted Exponential, Shifted Erlang, and Normal) or by moving the "sliders" on the edge of the diagram.
To simulate constant processing time, set the standard deviation to zero.

Technical notes regarding the Distributions:

"Shifted" means adding a constant. Hence, the shifted exponential is just like the exponential, except that its minimum value is greater than zero. To return a "shifted" distribution to its "unshifted" position, either reduce the mean or increase the standard deviation. For the distributions listed in the box, you can just set the standard deviation equal to the mean, and then select the button next to the distribution name. The computer will automatically reduce the standard deviation to its "unshifted" value.

The "Uniform" distribution has fixed upper limit, b, and fixed lower limit, a. The mean is (a + b)/2 and the standard deviation is (b - a)/sqrt(12) = (b - a)/3.464102. Going the other way, the two limits are a = m - s sqrt(12) and b = m + s sqrt(12). For example, if processing time is uniformly distributed between 5 and 11 minutes, the mean is 8 and the standard deviation is 1.732. Select the "Uniform" button, type these values into the mean and standard deviation, and press the button "Enter Mean & StDev'.

All of the distributions except the uniform are approximations, using Ramburg and Schmeiser's generalization of Tukey's Lambda Distribution.

Labels on the graph give the mean, median, standard deviation, and the third and fourth standardized central moments. Colored lines on the graph show the mean, the median, and the ranges between common fractiles: 0.01 to 0.99, 0.05 to 0.95, and 0.25 to 0.75.

1 Getting Started

If you have not already done so, start CellSim by opening the file CellSim.xls. This file contains “Macros” that carry out the simulation, so Excel may warn you that the macros might contain bad stuff (like a virus) and ask whether you really want to enable those macros. Respond “yes” and wait for the program to load.

The first page of CellSim.xls is a brief set of Directions, which are reproduced in the previous section of this document for your convenience.

The actual simulation is carried out on the sheet named “Model”. Click on its tab at the bottom of the screen. The top 2 rows and the left-most 2 columns have information about the program, buttons to run the simulation and input cells to control it. If you pass the cursor over any of the shaded cells in column B (no need to click) an explanation of the cell will pop into view. (The explanation disappears when you move the cursor away from the cell.)

The area below and to the right of cell C3 is where the model is constructed and displayed. If you want to learn, right now, how to construct a new model “from scratch”, continue with this section. If you would rather start with a model that someone else has constructed, just skip to the “Demo Models” section.

1.1 Constructing a Model

This illustrates how to create new Machines and StorageAreas for your model.

1.1.1.1 __How To: Get Rid of Unwanted Model Elements__

Since your copy of CellSim.xls may already have a model, to “start from scratch” requires that you delete whatever Machines and StorageAreas already exist.

If there is anything in the area below and to the right of cell C3,

- “Click and Drag” to select all “occupied cells” below and right of C3, and click.
- Press the “Delete” key on your keyboard.
  - For each model element in the selected area, a message will appear asking whether you want to delete it. Respond “Yes”.
  - Other cells within your selection will also be cleared.

1.1.1.2 __How To: Create a StorageArea__

- Move the cursor to cell F4 and double-click.
- On the window that appears, select “Create New Storage Area” and click OK.

The next window shows the name assigned to this StorageArea, and its storage capacity. The area labeled “Items” is empty because no items have been assigned here yet. That will change automatically when we create a Machine that sends something here for storage.

- Click OK.

1.1.1.3 How To: Create a Machine

- Move the cursor to cell D and double-click.
- On the window that appears, select “Create New Machine” and click OK.

The next window shows the name assigned to this Machine. The area labeled “Processes” shows that “Proc1” as been created at this Machine. We will now modify that process in several ways.

1.1.1.4 How To: Assign a Destination for Output from a Process

First, let’s tell the Machine to send the resulting item to the StorageArea we just created.

- Move the cursor to the window where “Proc1” shows, and click on “Proc1”.
- Click the button labeled “Edit Process”. (This button does not appear until you select a process, as in the previous step.)
- In the new window, find the label “Normal Output: Unassigned” and click the button next to it, labeled “Assign”.
- Select the cell containing the StorageArea Inv1 (that is, click on that cell).
- Click OK. Notice the change: the label now says “Normal Output: Inv1”.

1.1.1.5 How To: Assign a Processing Time Distribution

Change the Processing Time distribution to have a mean of 10 and standard deviation of 2.

- Find the label “Process Time: Mean=1, StDev= 0” and click the button next to it, labeled “Change Proc. Time”.
- Click in the box labeled “Mean” and type 10.
- Click in the box labeled “StdDev” and type 2.
- Click the “Enter Mean & StdDev” button.

Notice that the diagram changes. The probability distribution is “uniform” between two values. This can be changed in several ways. For example,

- Click the “Shifted Erlang” button and use the arrows next to it until the label shows “Shifted Erlang 3”.

Notice that the shape changes as different versions of the Erlang distribution are selected. The higher the number, the more symmetrical the distribution becomes. You may play around with the scroll-bars in the margins of the diagram to see what different shapes are available. When you are finished playing,

- Click “Normal”. The distribution should now show the familiar “bell-shaped curve.”
- Click OK, and continue clicking OK until the windows have all closed and you are back to the “Model” spreadsheet.

If you have had any difficulty with the above instructions, but don’t want to go through this again, don’t worry! The model already exists as Demo1.xls, and the next section shows you how to get it.

2 Demo models

CellSim.xls comes packaged with several demonstration files. This section uses those files to show you many of the features of CellSim. Perhaps more importantly, it illustrates how you may use Machines and StorageAreas to represent entirely different aspects of a factory. For example, customers or orders arriving randomly may be represented by a Machine with random Processing Times.

2.1 Retrieving a Model from a File

This illustrates how to go back to a model that was constructed at an earlier time. Since the model is stored with the output, opening an output file previously stored by CellSim will bring up the model that was used to generate the output.

- Click the button labeled “Open a Model”.
- If necessary, navigate to the folder in which file Demo1.xls is contained.
- Double-click on Demo1.xls.
- The screen now shows the model that was stored in that file. The colors indicate the state of the model when the previous run was finished. The color codes are shown on the left side of the screen. The name “Proc1” is the Process that was active, and the number in the StorageArea is the inventory that was present at the end of that run.
- To see the names of the model elements, click “Show Model”. This changes the labels to “Mac1” (the name of the Machine) and “Inv1” (the name of the StorageArea).

2.2 Viewing Data from a Simulation

If you would like to see the data collected the last time that this model was run, use the Excel menu “Window” and select Demo1.xls. The tabs at the bottom of the page show the different data files.

- ***Machines*** is the sheet which gives an overall summary of all Machines. In addition, there is a separate sheet for each Machine, summarizing all of its Processes. In this case, there is only one Machine, Mac1.
- ***Mac1*** is the sheet that summarizes the Processes (in this case only Proc1) on that Machine.
- ***Inventory*** gives an overall summary of all StorageAreas. In addition, there is a separate sheet for each StorageArea, summarizing its Items.
- ***Inv1*** summarizes the Items (in this case, only Proc1) in that StorageArea.

The last two sheets are used by CellSim to restart the model.

- ***MachineList*** gives the descriptions of the Machines in the model.
- ***StorageList*** gives the descriptions of the StorageAreas in the model.

2.3 Demo 1: Random Arrivals

Before proceeding, you must have "opened" Demo1.xls using the instructions above. Make sure that the CellSim.xls window is showing. (If not, use the Excel menu "Window" and select CellSim.xls.)

The model contained in Demo1.xls shows how a Machine can be used to simulate the arrival of materials or orders. This is accomplished by a Process without inputs. The outputs are the arrivals. That is, in Demo1, arrivals occur at StorageArea "Inv1". Processing Time represents time between arrivals.

We will show you several ways to change the arrival Process. For example, if you want more than one unit to arrive at the same time, the Process's Batch Size causes "bulk arrivals". The "number that arrive" can also be made random (bulk arrivals, random quantity), by randomly "scrapping" some of the units.

2.3.1 Individual Arrivals:

Machine Mac1 has one Process, Proc1. That Process has no input, and its normal output goes to StorageArea Inv1. The Processing Time has mean 10 and standard deviation 2. Hence, units of Proc1 arrive at Inv1 at random intervals, averaging one per 10 time units.

2.3.1.1 How To: Change the Processing Time Distribution

Goal: Change the distribution of "time between arrivals" to Exponential with mean 10. To do this, you must have opened the Demo1.xls model. If you have not, follow the instructions at the beginning of this section to open it.

- Make sure the names "Mac1" and "Inv1" appear on the screen. If they do not, click the "Show Model" button.
- Double-click Machine Mac1.
- Select Proc1 and click "Edit Process".
- Click the "Change Proc. Time" button.
- Enter 10 in the "mean" box and 10 in the "StDev" box.
- Click "Enter Mean & StDev".
- Click "Shifted Exponential".
- Click OK buttons until the dialog is finished.

If you run this simulation, the arrivals will still occur one per 10 minutes, on average, but will be more variable than before because you increased the standard deviation. If you run the model under Animation, this difference will NOT be obvious.

2.3.2 Bulk Arrivals with Fixed Quantity per Arrival:

Goal: Generate arrivals that occur 5 units at-a-time.

2.3.2.1 How To: Copy a model element

Copy Machine Mac1 to a different cell as follows (you may wish to click "Show Model" first to reveal the names:

- Double-click Mac1,
- click the "Copy" button,
- select an empty cell, click "OK".
- Change the name to Mac2,

- click "OK" buttons until the dialog is finished.

2.3.2.2 How To: Change the Batch Size

- Double-click the new Machine, and make sure its name is Mac2.
- Select Process Proc1,
- click "Edit Process", and
- enter a Batch Size of 5.

(Note: do not use Lot Size. Units in a Lot are produced one-at-a-time, whereas all units in a Batch are produced during one Processing Time.)

- Click OK buttons until the dialog is finished.

2.3.3 Bulk Arrivals with Random Quantity per Arrival:

Goal: Cause Machine Mac2 to deliver batches with quantity governed by the Binomial Distribution with parameters n=5, p=0.7.

2.3.3.1 How To: Change the Percent Scrap

Use the instructions in the previous example to copy Machine Mac2, and name it Mac3.

- Double-click on Mac3, select Process Proc1, click "Edit Process", and
- enter a "Unit Percent Scrap" of 30. (Note: do not use the "Batch Percent Scrap" as that causes the entire batch to be scrapped.)
- Click OK buttons until the dialog is finished.

Since the "Scrap Output" is "unassigned" each unit of scrap disappears. Hence each unit in a batch "survives" with probability 0.7, so the quantity that leaves by the Normal Output averages 70% of a batch. That is, the "Normal Output" quantity at the conclusion of each Processing Time is Binomial with n = Batch Size and p = (1-PercentScrap/100).

2.3.4 Exercise: Adding StorageAreas and Reassigning Outputs

You now have three different kinds of arrivals, going into the same StorageArea. Change the model so that each Machine has its own output StorageArea. That is, Mac1 sends its stuff to Inv1, Mac2 to Inv2 and Mach 3 to Inv3. You should create 2 new StorageAreas, and then change the output links for Mac2 and Mac3.

- Double-click on an empty cell to the right of Machine Mac2.
- Select "Create new Storage Area".
- Click OK.
- Change the capacity to 10,000. (You need plenty of room for arrivals. If the StorageArea gets full, the Machine will be blocked and arrivals stop.
- Click OK until the dialog is finished.

You may create StorageArea Inv3 by the same Process, or as follows:

- Click StorageArea Inv2.
- Click "Copy",
- select an empty cell to the right of Mac3 and click OK.
- Change the name to Inv3.

Now you have to change the output links for Mac2 and Mac3.

- Double-click on Mac2, select Process Proc1 and click “Edit Process”.
- Click the “Change” button for Normal Output,
- select the cell containing Inv2 and then click OK until the dialog is finished.

Now edit Mac3 in a similar way so that its output goes to Inv3.

2.3.4.1 How To: Save a Factory Model

To store this marvelous creation for later use, the easiest way is to run a simulation, since the factory description is saved with the output.

- Click “New Runs”.
- Respond to the prompt, giving your model a file name.
- If something goes wrong, the file Demo1A has this model in it.

2.4 Demo 2: Controlling Multiple Processes on One Machine

The model in Demo2.xls shows one Machine with 2 Processes. Each Process has a Setup Time of 15 time units, following which output of 1 unit per minute continues until 10 ***non-defective*** units have been finished (Lot Size = 10). Then the Machine moves to the next Process. Since each Process has its own StorageArea for output, you may see the change-of-Process by selecting the “Animation” checkbox and running the simulation. The rest of this section illustrates different aspects of how CellSim chooses which Process to run.

2.4.1 Cyclic Production

Demo2.xls is an example of Cyclic Production since each Process runs one Lot, and surrenders the Machine to the next Process. When all Processes have “done their thing” the Machine returns to Proc1 and begins again.

2.4.1.1 How To: Add a Process

Goal: Add another Process to the cycle, with its own StorageArea for output.

- Click “Show Model” if the names are not showing on the Machine and StorageAreas.
- Double-click StorageArea Inv2, click “Copy” and select an empty cell. Change the name to Inv3. (Note the large capacity. This is necessary for a StorageArea to represent output of a factory, to avoid blocking the Machine.)
- Select Item “Proc2” and click “Delete Item”. (No point in keeping records on “Proc2” since it is not stored here.)
- Click “OK”.
- Double-click Machine Mac1.
- Select Proc2, and then click “Copy Process”.
- Click “Edit Process”.

Note the Setup Time, Processing Time, Lot Size and Batch Size. After 15 minutes of setup, each unit requires exactly 1 minute to Process, and the lot will be complete after 10 units. Each 1-minute Processing results in only 1 unit, since Batch Size is 1.

- Click the “Change” button next to “Normal Output, select the cell containing the new StorageArea, Inv3, and click OK.

- Click OK until the dialog is finished.

To see if it works properly, make sure that “Animation” is checked, and click “New Runs.” If you have trouble, the file Demo2A.xls has the model in it. You may view that model using “Open a Factory File”. (Don’t bother if your model is running correctly.)

2.4.1.2 How To: Use Process Controls

Goal: Make it so that Proc1 must produce 100 units before the Proc2 is allowed to begin.

- Click “Show Model” if the names are not showing on the Machine and StorageAreas.
- Double-click Mac1, Select Proc2 and click “Edit Process”.
- Find the area labeled “Controls” and click the “Change” button.
- Click “Assign Control” button,
- select the cell containing StorageArea Inv1,
- click OK.
- Select Item Proc1 and click OK.
- Change the control level to 100.
- Click the top “radio button” under “Activation Method”.
- Click the top “radio button” under “Inventory Characteristic Monitored”.

At this point, the label next to the “Assign Control Point” button should say, “Controlled by INVENTORY of Item ‘Proc1’ at StorageArea ‘Inv 1’.”

- Click OK

Look at the label in the “Controls” area. It should say “Run this Process when INVENTORY of Item ‘Proc1’ at StorageArea ‘Inv1’ is 100 or higher.”

- Click OK until the dialog is finished.

To see if it works properly, make sure that “Animation” is checked, and click “New Runs.” Proc2 should not run until fairly late in the simulation, at the point when 50 units are present in Inv1. If you have trouble, the file Demo2B.xls has the model in it. Open that file using “Open a Factory File” as explained earlier.

2.4.2 Cyclic Production with Multiple Setups of a Process

The previous example used a cycle of the type A-B-C, with no repetitions. Here is how to have a cycle of the type A-B-A-C, in which process A occurs twice per cycle. The key is to have two copies of that process, in different positions in the cycle.

2.4.2.1 How To: Change the Order of Processes

Goal: Make it so that Proc1 happens twice per cycle, in the first and third positions.

- Click “Show Model” if the names are not showing on the Machine and StorageAreas.
- Double-click Mac1

First, remove the controls from Proc 2

- Select Proc2 and click “Edit Process”
- In the “Controls” area, click “Delete”. Click OK.

Now make a copy of Proc1.

- Select Proc1
- Click “Copy Process”. Note that Proc4 appears. That is the copy of Proc1.
- Select “Proc4”.
- Locate the “Change Process Order” buttons, and click the Upward Pointing one. This should move Proc4 from its fourth position to the third position.
- Click “Edit Process”
- Change the name to Proc1 and click OK until the dialog is finished.

Select Animation, click “New Runs” and watch the cycle progress. Notice that the processes cycle as follows at Mac1: Proc1, Proc2, Proc1, Proc3, and the arrows to the output StorageAreas cycle accordingly. If you have trouble, the file Demo2C.xls has the model in it. Open that file using “Open a Factory File” as explained earlier.

2.5 Demo3: Kanban Controls

A Kanban is used to signal when materials are needed. A Machine controlled in this manner is supposed to sit idle if no Kanban is present. On the other hand, if a Kanban is present, only one Machine is supposed to respond to it. That is, if two Machines are capable of producing an item, but one Kanban is present requesting 20 units, then one of the Machines should produce the order and the other should do something else.

CellSim processes are kept idle when their control criteria are not met. This example demonstrates how this feature may be used to simulate Kanban. At the same time, it shows the difference between controls based on “inventory on-hand” versus “inventory position”. (The latter changes in response to a promise of a delivery, whereas the former changes only when the stuff actually arrives.)

2.5.1 A Two-Stage, Balanced Production Line

The simplest Kanban factory has one Machine that makes stuff for another. The following steps illustrate how to model such a factory.

2.5.1.1 How To: Merge Models

We will make this a 2-stage factory by importing a copy of the same factory.

- Click “Open a Model” and Select “Demo1.xls”
- Click “Merge a Model” and Select “Demo1.xls”

A second copy of the model appears. Click “Show Model” to display the names. Note that the names have been modified on the second copy. CellSim avoids duplicate names because the output is easier to read if each Machine and StorageArea has a unique name.

2.5.1.2 How To: Assign Input for a Process

To model this as an asynchronous production line, “hook up” the input for the second Machine to the “output StorageArea” for the first Machine.

- Double-click the second Machine, Mac1A
- Select Proc1 and click Edit Process
- Locate Input 1 and click “Assign” next to it.
- Select the cell in which the StorageArea Inv1 is located and click OK.

- Seclect Proc1 and click OK.

The “Input 1” box should now read “Input 1: Proc1 from Inv1”.

- Click OK until the dialog is finished.

An arrow should now appear from Inv1 to Mac1A. If you have trouble, the file Demo3.xls has the model in it. Open that file using “Open a Factory File” as explained earlier.

Make sure that Animation is checked and run the model. The inventory in Inv1 stays near zero because the two Machines are identical. That is, the line is perfectly balanced.

Now we will purposely unbalance the line by making Mac1 much faster. This simulates a situation in which the Machine that supplies parts has greater capacity than the Machine being supplied. Later we will make use of that extra capacity to supply a third Machine. Use the method explained earlier to change the Processing Time distribution, giving it a mean of 5 and standard deviation of 2. Just in case, here are some brief instructions:

- Double-click Mac1, select Proc1, click Edit Process,
- Click Change Proc. Time, change the mean to 5 and standard deviation to 1,
- Click OK until the dialog is finished.

Now run the simulation again, with Animation. Notice that the inventory builds up in the StorageArea between the Machines. This is to be expected because the first Machine is faster.

To keep that inventory low, we will institute a Kanban system that orders 10 units from Mac1 whenever inventory is 5 or less between the Machines.

- Double-click Mac1, select Proc1 and click Edit Process
- Change the Lot Size to 10
- In the Controls area, select Change
- Click “Assign Control Point”, select the cell containing the StorageArea between the two Machines and click OK
- Select Proc1 and click OK
- Change the Control Level to 5
- Select “Control Based on Inventory POSITION” (we will try the other setting later).
- Select “Activate when POSITION is ‘At or Below’ the control level”
- Click OK until the dialog is finished

Now run the simulation with Animation. Note that the first Machine remains idle until inventory is 5 or lower, then produces 10 units. Meanwhile the second Machine is busy all the time. If you have trouble, use the model in Demo3B.xls.

2.5.1.3 <u>How To: Use POSITION vs. ON-HAND Inventory in Controls</u>

Often there are several Machines capable of supplying a given part. To avoid overproduction, some method is needed to make sure that an order is “assigned” to only one Machine. That is the role played by Inventory Position. As soon as a process begins a setup for a Lot>0 of an item, its output buffer’s Inventory Position is “credited” by the amount of the Lot Size. Thus, any other Machine controlled by this

StorageArea’s inventory position will not be allowed to produce. To see this, create another copy of Mac1.

- Double-click Mac1, select Copy Machine, select an empty cell below Mac1 and click OK.

Run this simulation with Animation. You will see that the new Machine produces almost nothing. During the first part of the run, it gets an opportunity because there was no inventory at the start, but once the equilibrium has been reached, Mac1 does all of the production.

To see what happens if POSITION is not used, change the controls for both Machines.

- Double-click Mac1, select Proc1, click Edit Process
- In the Controls area, click Change
- Change the “Inventory Characteristic Monitored” to “Control Based on Inventory ON HAND”.
- Click OK until the dialog is finished.
- Repeat this process for Mac1B.

Now run a new simulation. Note that both Machines begin production at the same time, and the maximum inventory is much higher. That is because they both get permission to produce whenever inventory is 5 or less, and there is no signal that one of them has taken on the task. Inventory is higher because both Machines produce 10 units (the Lot Size) before shutting down.

2.6 Demo4: Rework of Rejected Units

When defective units are produced and detected immediately, in some cases they may be repaired. If repair takes place in a separate facility, the “Scrap Output” may be discharged to a StorageArea, and another Machine can be added to represent the repair shop. However, if repair takes place on the same Machine that produced the item, some other issues arise. First, does the repair take place immediately, or can the defective item join others in a StorageArea for later attention? Second, how much time does repair require compared to the original process? If repair is simply repeating the process, then they have the same distribution. However, repair may require additional operations (such as disassembly) or fewer operations (perhaps only the last step must be repeated).

Load the demonstration model by clicking “Open a Model” and selecting Demo4.xls. This file contains a 2 stage factory. Mac1 feeds a StorageArea named WIP; Mac2 takes those units, processes them and deposits them in a StorageArea named “Finished”. However, Mac2 (the second stage) sometimes produces items that are defective. Defects are determined randomly in the simulation using a “Unit Percent Scrap” of 20. The Scrap Output is assigned to a StorageArea named Rework, and a Machine named Repair fixes them and puts them in with the rest of the finished items.

Using Animation, run this model. You will see that inventory in the Rework area remains low. That is because the Repair Machine is (in this case) identical to the original, but only has to operate on 20% of the units. Therefore it is idle most of the time.

2.6.1 Rework by the Same Machine, after Pre-Processing by a Second Machine

Suppose now that the Repair Machine does not actually do the repair, but merely prepares the unit to be reprocessed by Mac2. Then its output needs to go to the WIP

StorageArea in front of Mac2. Assuming that these items are now indistinguishable from other input for Mac2, the following steps change the model to mix “new items” and “recycled items” in the WIP StorageArea.

- Double-click the Machine named “Repair.” (If you can’t see the names, click “Show Model” first.)
- Select Proc1 and click Edit Process.
- Find the Normal Output and click Change
- Select the cell containing StorageArea WIP
- Click OK until the dialog is finished.

Now run this model. (If you have difficulty, use “Open a Model” to get the model from file Demo4A.xls.) Notice that the inventory in the WIP buffer is building up. This is because Mac1 and Mac2 are identical, except that Mac2 has to repeat some of its work. In essence, Mac2’s workload is increased because of the rework.

2.6.2 Rework that Requires Different Time than the Original Process

Now suppose that the items requiring rework need only 2 minute of processing, with standard deviation 0. Then Mac2 has a new Process, one that takes much less time and has only “Rejects” as inputs. Therefore, we must label the rejected items so that the new Process will take the correct items from WIP.

In CellSim, the output of a Process takes the name of the Process, so we need to go to the Repair Machine and change the name of its Process.

- Double-click the Repair Machine, select Proc1 and click Edit Process.
- Change the name of the process to Recycled
- Click OK until the dialog is finished.

Next we will put the new process onto Mac 2 to fix the recycled item.

- Double-click Mac2,
- Click “Add Process”. A new process, named Proc2 is created.
- Select “Change Proc. Time”, change the Mean to 2 and click OK.
- Find Input 1 and click the Change button next to it,
  - Select the cell containing the WIP StorageArea and click OK
  - Select the item named “Recycled” and click OK.
- Finc Normal Output and click the Change button next to it, select the cell containing StorageArea “Finished” and click OK
- Click OK until the dialog is finished.

Before you run the simulation, be prepared to watch Mac2 to see how often Proc2 is carried out. Using Animation, run the model. (If you have trouble, click “Open a Model” and select Demo4B.xls.)

Now take a look at the results. Use the Excel menu “Window” and select the Excel file that contains the results (Demo4B.xls if you opened the demo file). The tabs at the bottom of the page show the different data files. Select the sheet named Mac2. You will see that “UnitsDone” is zero for Proc2. Somehow it never got started. Now select Machines, and see that Mac2 was busy all of the time after the runin. Somehow, it was so busy doing the regular items that it never got around to the rework.

Go back to the CellSim.xls window.

2.6.2.1 How to: Use Priorities

To force Mac2 to do the rework as soon as it is available, give Proc2 a higher priority.

- Double-click Mac2, select Proc2 and click Edit Process.
- Locate the "Priority" area and click "High".
- Click OK until the dialog is finished.

(This model is in the file Demo4C.xls.) Check the Animation box, and then select "New Runs". When the runs are finished, take a look at the results. (That is, use the Excel menu "Window" and select the file at the top of the list.) Select the sheet named Mac2. You will see that "UnitsDone" is no longer zero for Proc2. Select the sheet named WIP. The inventory at the "End" is entirely Proc1. The higher priority for "Recycled" caused Proc2 to clean out this material.

CellSim's priorities do not allow a unit that is "in-process" to be interrupted, but when a unit is finished, the Machine will switch to a higher priority process if its inputs are available. However, if Proc1 had a Lot Size above 0, the "high priority" of Proc2 would not be applied until Proc1 finished its entire Lot. (To interrupt during a Lot, use the next higher priority, labeled "Preempt during a Lot".)

2.6.3 Dummy Machines or Processes

A dummy process is a process with zero Processing Time. For example, in the previous demonstration, the Repair Machine could have zero Processing Time, signifying that its only function is to take the scrap items, re-label them, and move them into the WIP StorageArea.

Another example occurs for a machine that makes multiple parts of different types in the same batch. If output is to consist of 5 units of x and 3 units of y, this may be simulated as follows.

- First, create a machine with process P1, Batch Size 2, output to StorageArea B. Give this process the Processing Time of the real process.
- Then create a machine D with two dummy processes, x and y, each having Lot Size 1, Processing Time and Setup Time 0, and input of 1 unit of P1 from StorageArea B.
- Give process x a Batch Size of 5, and process y a Batch Size of 3. Direct their outputs to the appropriate StorageAreas.

In this example, Machine D and StorageArea B are both dummy elements, having no correspondence to a real part of the factory. This model is in file Demo5.xls. If you want to see it run, be sure to check "Animation".

3 Exercises

3.1 HighBounce Inc.

HighBounce Inc. has a 3-stage production line for its brand of sneakers. The first machine cuts the material, the second forms it into shape, and the third machine performs the sewing and gluing operations. There is a storage "buffer" between the stages to "uncouple" the processes. For example, the first buffer is supposed to prevent stage 2 from being starved in case stage 1 slows down for any reason. Each storage buffer has the capacity to store 4 sneakers.

Stage 1 has an average Processing Time of 2 minutes (standard deviation 0.5 minutes), during which it produces material for 4 sneakers. Stage 2 operates on one unit at-a-time, requiring 30 seconds per unit. After each hour of use, stage 2 must be stopped and cleaned, which requires 10 minutes. Stage 3 has a cycle time that averages 24 seconds with almost no variation, but 10% of the units must be scrapped.

(a) How many sneakers per hour may be produced, on average?
(b) What happens if the storage capacity is increased to 8? To 16? Explain why the output rate changes as storage capacity is increased.
(c) What happens if stage 3 were improved so that scrap no longer happened? Explain this result.
(d) What happens if stage 2's cleaning could be eliminated by a continuous, automatic cleaning device?
(e) What happens if variability at stage 1 could be eliminated?

Suggestions: Begin each analysis with the model from part (a), changing only one thing at a time. Also, do not worry about the exact "shape" of the Processing Time distribution at stage 1. Research has shown that the mean Processing Time has the greatest effect on system output, standard deviation has a much smaller effect, and other shape parameters have very little effect. Finally, be sure to adopt a single time unit for all Processing Times and for the repair cycle. The problem states some times in hours, some in minutes and some in seconds. I suggest converting them all to minutes. (If you do not use the same time units throughout, your results will differ wildly from reality.)

3.1.1 Solution

A model of this factory may be found in Ex1.xls. All times were converted to minutes as follows:

- The Processing Time at stage 2 is 0.5 minutes.
- The cleaning time of stage 2 may be represented as failures that occur every 60 minutes, with repairs that require 10 minutes.
- The average Processing Time for stage 3 is 24/60 = 0.8 minutes.

(a) After a runin of 1000 minutes, 5 runs of 1000 minutes were executed. For those 5 runs, the output of the line (that is, the output of stage 3) averaged 1054.6 units per 1000 minutes (varying from 1046 to 1064). Thus the average output rate is 63.3 sneakers per hour.

(b) Same run setup.
- For capacity 8, output averaged 1084 per 1000 minutes, or 65.0 sneakers per hour, 2.7% higher than part (a).
- For capacity 16, output averaged 1118.8 per 1000 minutes, or 67.1 sneakers per hour, 6.0% higher than part (a).

Larger storage capacity helped to avoid lost production. For example, the third machine was starved 6.4% of the time with capacity 4, 3.3% for capacity 8, and 0% for capacity 16.

(c) After a runin of 1000 minutes and 5 runs of 1000 minutes, the output of stage 3 averaged 1170.4 units per 1000 minutes, or 70.2 sneakers per hour, which is 10.9% higher than part (a). The reason is simple. The machine can process the same number of sneakers, but all of them go to the normal output rather than being divided between normal and scrap.

(d) After a runin of 1000 minutes and 5 runs of 1000 minutes, the output of stage 3 averaged 1126.4 units per 1000 minutes, or 67.6 sneakers per hour, which is 6.8% higher than part (a). Eliminating the "down time" of stage 2 made it possible to run stage 3 continuously (never starved). Before this change, stage 2 stopped producing for 10 minutes every hour, and the buffer of 4 units was not sufficient to keep stage 3 going during cleaning. Now, with cleaning eliminated, stage 3 never stops producing.

(e) After a runin of 1000 minutes and 5 runs of 1000 minutes, the output of stage 3 averaged 1059 units per 1000 minutes, or 63.5 sneakers per hour, which is only 0.4%% higher than part (a). For practical purposes there is no change. Stage 3 is still starved about 6.4% of the time. We saw that the principal reason for loss of output was the down-time at stage 2. Eliminating stage 1's variability may be an improvement, but it does not address the major reason for loss of output in this case.

3.2 MyPants, Inc.

MyPants manufactures hockey equipment. In one part of the factory, two machines are arranged in series, forming a 2-stage manufacturing cell. The industrial engineers worked hard to balance this cell so that the two stages have the same average Processing Time. However, there is substantial variation in the Processing Time. (The average Processing Time is 10 minutes and the standard deviation is also 10.) They are seeking your advice as to how large the storage area between the stages should be.

(a) Make a few simulation runs to demonstrate what the relationship between output rate and storage capacity looks like.

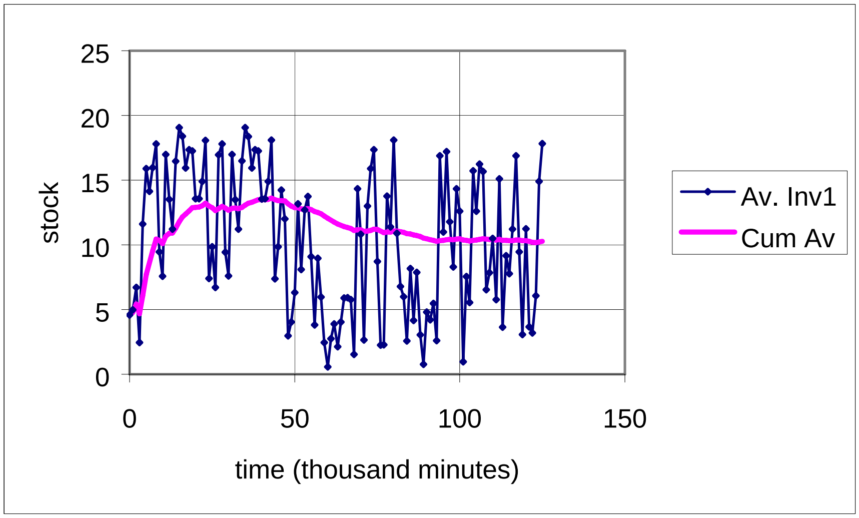
(b) How large should the storage area be, in your opinion?

(c) How does your answer change if they are able to cut the process standard deviation in half?

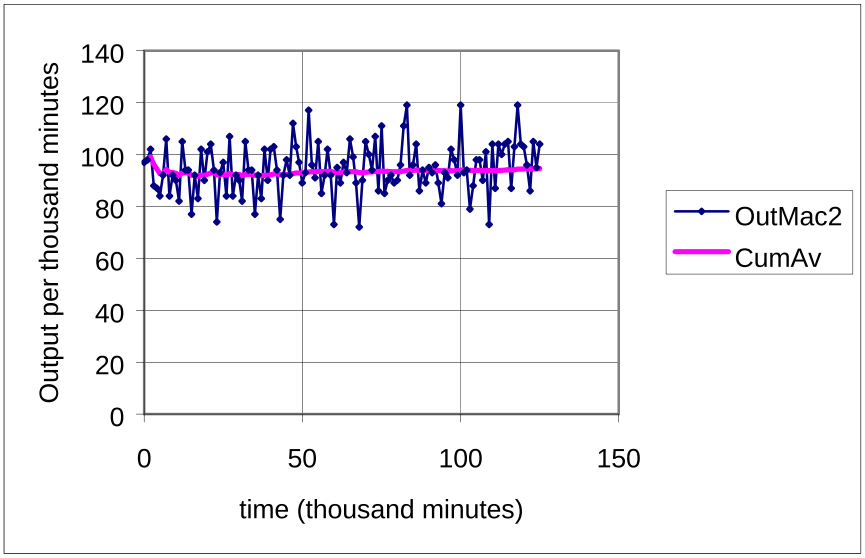
Suggestions: The exponential distribution has standard deviation equal to the mean. None of the other distributions available in CellSim can have standard deviation that large. Also, I suggest beginning with storage capacity 20 and trying a small number of other values. In each case, I suggest beginning with the storage area "half-full" (To do this, edit the item in the StorageArea, and set its “initial stock” equal to half of the storage capacity.) Make sure that your runs are long enough. It takes a lot of simulated events to be sure that the inventory has had plenty of opportunity to randomly grow and decline.

3.2.1 Solution

A model of this factory may be found in Ex1.xls. Some preliminary simulations were done with capacity 20, initializing at 10 units of stock. The diagram below shows the average level of inventory in each 1000 minute simulation as a broken line with dots. Note that 1000 time units is not sufficient to give us much accuracy, since it dances up and down a lot. The solid line is much smoother because it shows the cumulative average. For example, at time 50, the smooth line represents the average over the first 50,000 minutes. Note that this line eventually converges to a value of 10, which means that the storage area is half-full on average. This diagram shows how long it takes to get accurate inventory statistics. (Ex2.xls, sheet “Inventory”)

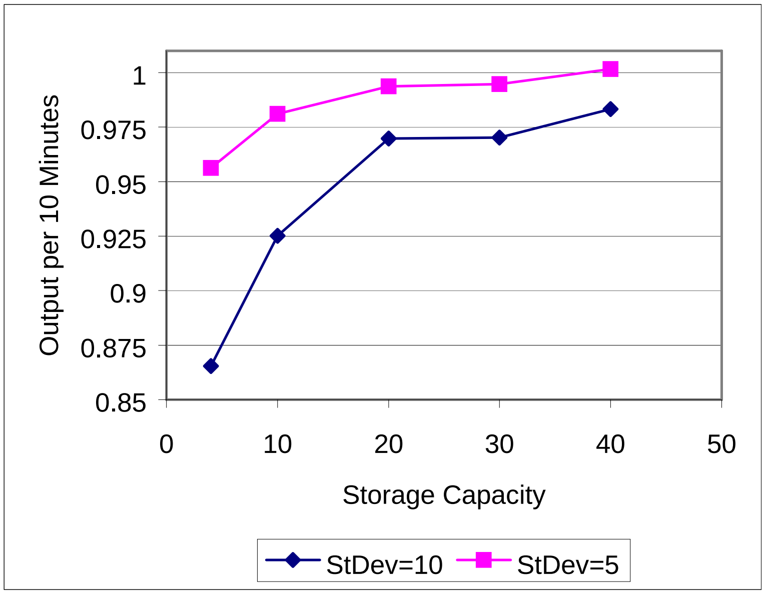


However, accurate statistics on the output (units finished per 1000 minutes) come much earlier, as the next diagram shows. (Ex2.xls, sheet “Machines”)



The cumulative average output quickly levels out at around 95.

(a) Based on this information, the runin period was set to 20,000 and run length to 100,000. Capacity of the StorageArea was then varied with the results shown below.



(b) You may draw your own conclusions about the necessary storage capacity. However, note that there are diminishing returns: as more storage capacity is added, the rate of improvement declines.

(c) If the standard deviation is cut in half, the storage capacity can be cut by more than half. For example, with standard deviation 5 and capacity 10, the output is higher than with standard deviation 10 and capacity 20.

3.3 SwitchIt Ltd.

SwitchIt Ltd. has recently purchased a forming machine. It is used to supply two different production lines. The first process on the forming machine makes the item used in one of the production lines, and the second process supplies the other line. Both have the same production time characteristics—the cycle time (i.e. Processing Time per unit) is 1 minute, and the Setup Time to change between processes is 15 minutes. Output from each process is immediately sent to a storage area in front of the corresponding production line. The storage capacity is essentially unlimited.

Both of the production lines have cycle times of 2.2 minutes, with standard deviation 1 minute. Since the forming machine is more than twice as fast as either production line, they expect that the forming machine will be able to keep both lines busy all of the time. However, initial tests found that both production lines were idle approximately half of the time. The operator of the forming machine pointed out that she had been instructed to produce lots of 10 for each product, and that this was causing 15-minute setups after each 10 minutes of run time.

(a) Make a few simulation runs to demonstrate what the relationship between output rate and Setup Time looks like. How large must the Lot Size be at the forming machine to avoid idle time in the production lines?

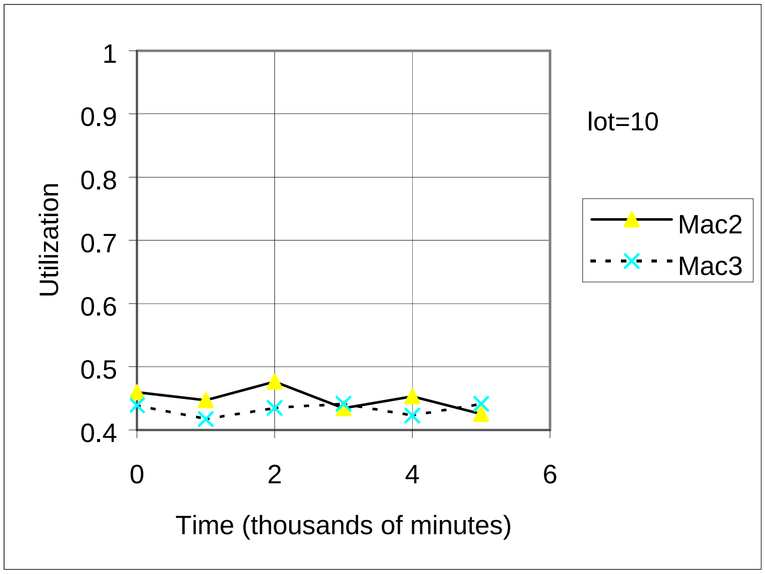
(b) What happens to inventory if this Lot Size is adopted?

(c) How does your answer change if they are able to cut the Setup Time to 5 minutes?
(d) Look carefully at the output to your simulation when Setup Time is 5 minutes. What problems do you predict for Switchit Ltd. if they implement this solution without additional restrictions? What do you suggest they do?

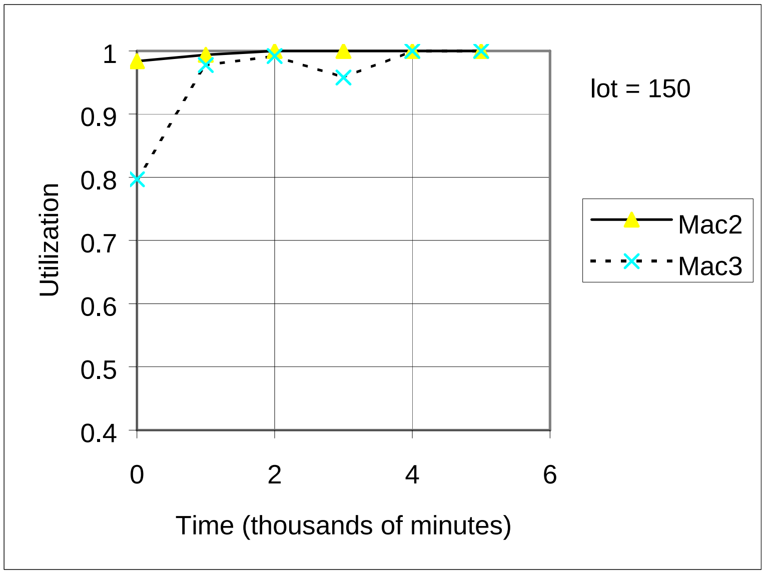
Suggestions: Do not confuse “Lot Size” with “Batch Size”. In your simulations, the Batch Size must remain at 1, because the forming machine makes one unit at-a-time. However, it produces 10 units between setups, so the Lot Size is 10, currently. Use very large storage capacity for each StorageArea (say 10,000) to represent “essentially unlimited” capacity. Also, particularly with large Lot Sizes, it may take some time before the system’s inventory builds up to a representative value, so be sure to use a long-enough runin time.

3.3.1 Solution

The first few simulations were done with Lot Size 10 (Ex3.xls). The diagram below shows the average utilization of the production lines in each 1000 minute simulation.

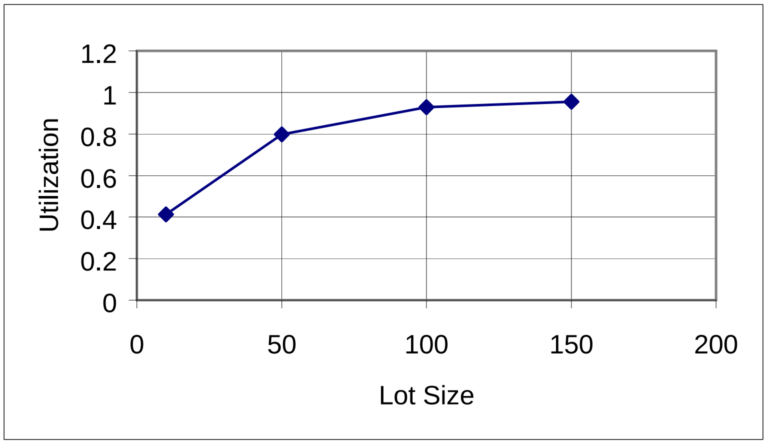


This was then repeated with Lot Size 50, 100 and 150. The diagram for the last run is shown below.

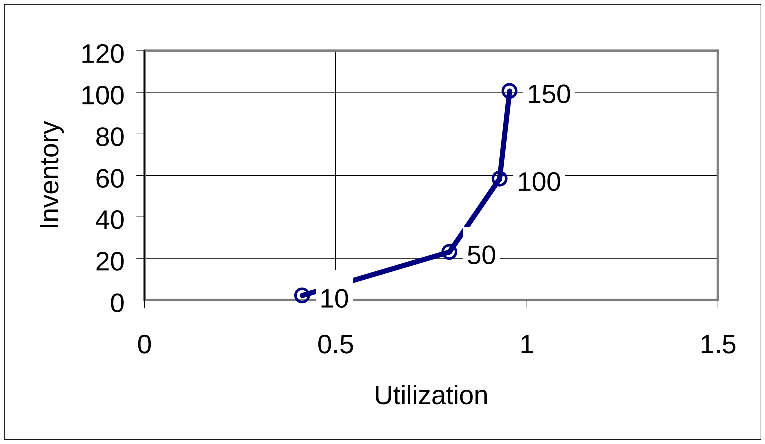


Since these show fairly consistent results after 4000 minutes, the data from the last 1000 minutes was used to address the question.

(a) The following graph shows Utilization (averaged for the two production lines, Mac2 and Mac3) as a function of Lot Size at the forming machine. It appears that Lots of more than 150 are needed to avoid idle time.



(b) The following graph shows Inventory (total in the two StorageAreas) as a function of Utilization as Lot Size is increased. Clearly a high price is paid in inventory storage to increase utilization by this method.



(c) Reducing the Setup Time makes everything work better. Two additional runs (not shown) revealed that utilization increases to 73% for Lot Size of 10, and to 100% for Lot Size of 50.

(d) The runs with Setup Time = 5 have inventory problems—it just keeps increasing. This occurs because the forming machine is now faster than the two lines it feeds. Hence they need some control to keep a cap on inventory. Two methods are explored in the next exercise.

3.4 SwitchIt Ltd. (2)

SwitchIt Ltd. has reduced the Setup Time of the forming machine to 5 minutes and are using a Lot Size of 50. However, inventory in front of the two production lines has begun to build, and shows no sign of stopping. The simple solution of putting a limited storage capacity has been suggested. An alternative is to direct the forming machine not to produce until stock is running low.

(a) Put a capacity limit of 25 on each StorageArea and see what happens.

(b) Put the capacities back to some very large number, and set the “Controls” for each process at the forming machine to disallow production unless inventory falls below 10, and see what happens.

(c) What do you recommend?

3.4.1 Solution

The factory with Setup Time 5 and Lot Size 50 is in (Ex4.xls).

(a) Changing the storage capacities to 25 gets rid of the inventory problem, but causes the forming machine to become “blocked”. This happens when a StorageArea gets completely full while the forming machine is in the middle of a Lot, which causes the forming machine to stop and wait until space becomes available to put the finished item. This is idle time because the forming machine cannot start anything else until the finished item is removed.

(b) Setting the controls to produce when the stock at the appropriate StorageArea is less than 10 also solves the inventory problem. The forming machine still has some idle time, but it is never blocked.