Assigning State

Properties of an object are the input parameters associated with that object. A Property holds a specific data type. It is a static input parameter (e.g. processing time, batch size) for the object and does not change during the running of the simulation. Properties are used to send information between objects (inputs and outputs) and they are also used for experimentation when a user wants to experiment with using different input parameters for their model.

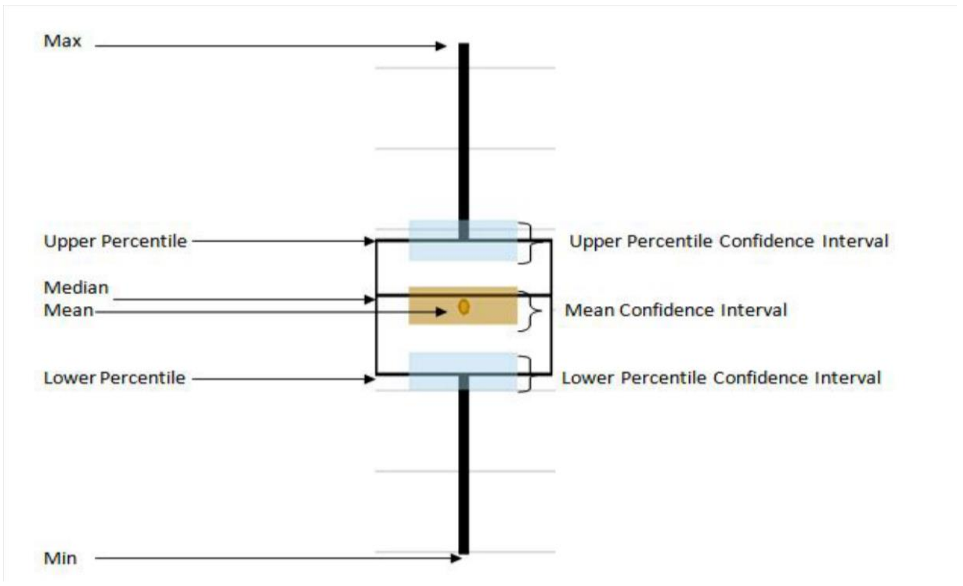
A State represents a numeric or string value that may change during the run. It is sometimes referred to as a State Variable because its value can vary during the run. A State can be added to any object in your Project Library. When a State is added to the main model object, it can be seen and accessed by all the objects within that main model. Therefore, it is similar to what is often referred to as a "global variable" in other software packages. When a State is added to another object, such as the ModelEntity object, it is part of that object's definition. Therefore, a state on a ModelEntity is similar to what is often referred to as an "attribute" of each entity that is going through the system. For example, if you add a Discrete State to the default ModelEntity object, called "MyCustomState", then each entity that enters the system will have MyCustomState on it and therefore can have a unique value assigned to that State. For example, a State on an entity might contain a value that specifies this entity's unique bar code, this entity's due date or number that will be used to identify which shipment this entity should go into. And the value of the state can be updated or referenced during the run with the syntax ModelEntity.MyCustomState, in an expression.

There are two categories of states: Discrete and Continuous. A discrete State may change via an assignment only at discrete points in time. A continuous State will change continuously and automatically when its rate or acceleration is non-zero.

State variable-Picture(default): A utility state to hold a picture index. This state can then be referenced in an expression for Current Symbol Index.

Add-on Processes are a special case of processes. Add-on processes are typically defined by a user to help customize object behavior or add specific behavior into a model. The objects in the Standard Library contain Add On Process triggers. When an Add On Process is defined in the Add On Process trigger property of an object, when the object fires a particular event, the Add On Process is executed.

Risk: Percentiles indicate where the system is likely to actually operate
Error: confidence intervals indicate how well the mean and percentiles are estimated


Max
Upper Percentile
Median
Mean
Lower Percentile
Min
Upper Percentile Confidence Interval
Mean Confidence Interval
Lower Percentile Confidence Interval

Statistical difference: depends on how much sampling variability there is in the estimate.(the difference between two or more scenarios)

Subset Selection

In general, the purpose of creating a simulation model and, subsequently, an experiment for that model is to create multiple scenarios for that system and pick the best one to implement in the future. Therefore, it is extremely important to accurately choose the best alternative for the system being studied.
A common comparison technique is to display a mean, and often, a confidence around the mean and allow for sorting based on comparison of the means. A user is inclined to pick the scenario with the highest/lowest mean with little to no information regarding how the scenario compares to other alternatives.

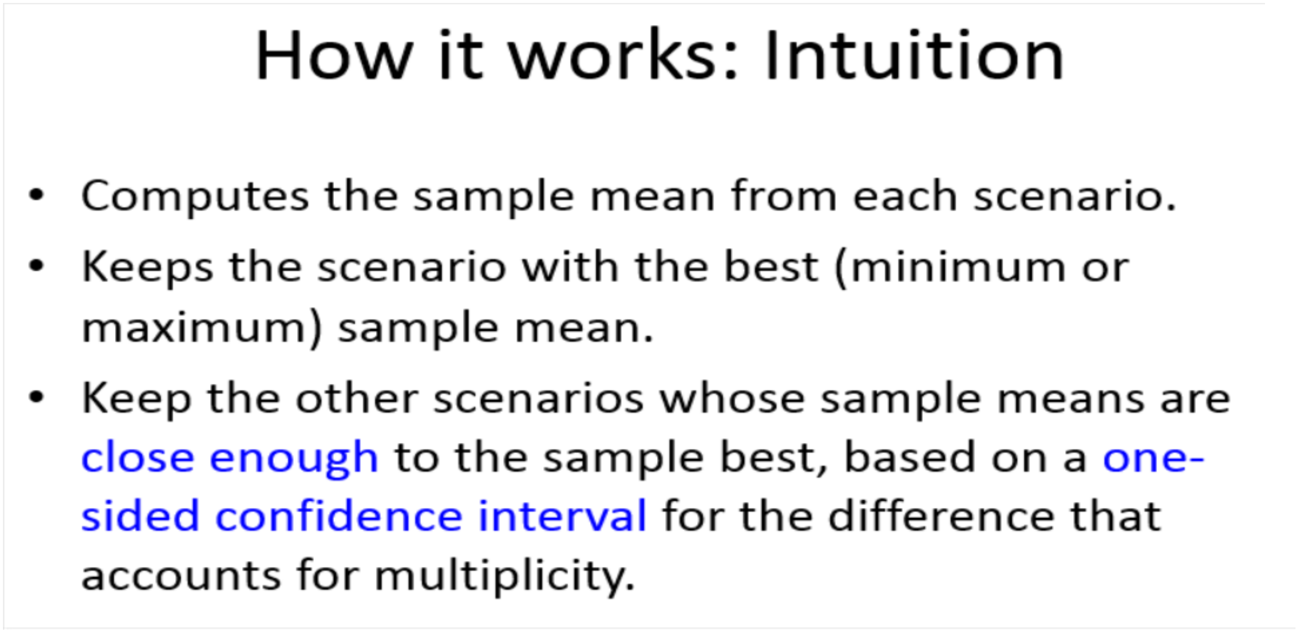
He has come up with a series of algorithms that group Response values into the subgroups "Possible Best" and "Rejects" within each Response. The Possible Best group will consist of scenarios that cannot be proven statistically different from each other, but can be proven to be statistically better than all of the scenarios in the Rejects group.

To indicate what constitutes "Best" the user will define a Response's Objective in the Response Properties window, to be either Maximize or Minimize. Maximize Objective will group the values that are statistically greater than the group of rejects, and conversely the Minimize Objective will group together the smallest response values.

In the Experiment Window's Design window, the cells in a Response column will be highlighted brown if they are considered to be "Rejects" leaving the "Possible Best" scenarios with their intended coloring. A

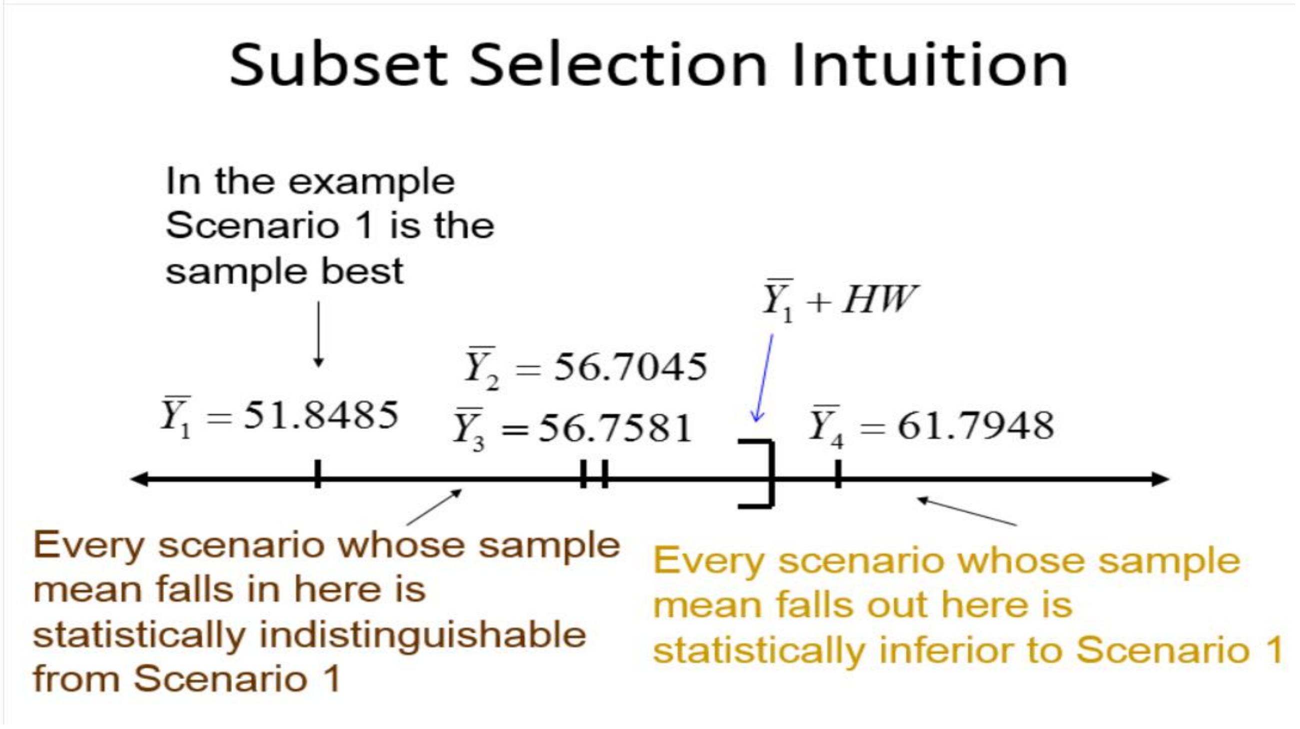
scenario can have a cell in one column considered “Possible Best” and at the same time “Rejects” for another response.

How subset selection works?



How it works: Intuition

- Computes the sample mean from each scenario.
- Keeps the scenario with the best (minimum or maximum) sample mean.
- Keep the other scenarios whose sample means are close enough to the sample best, based on a one-sided confidence interval for the difference that accounts for multiplicity.

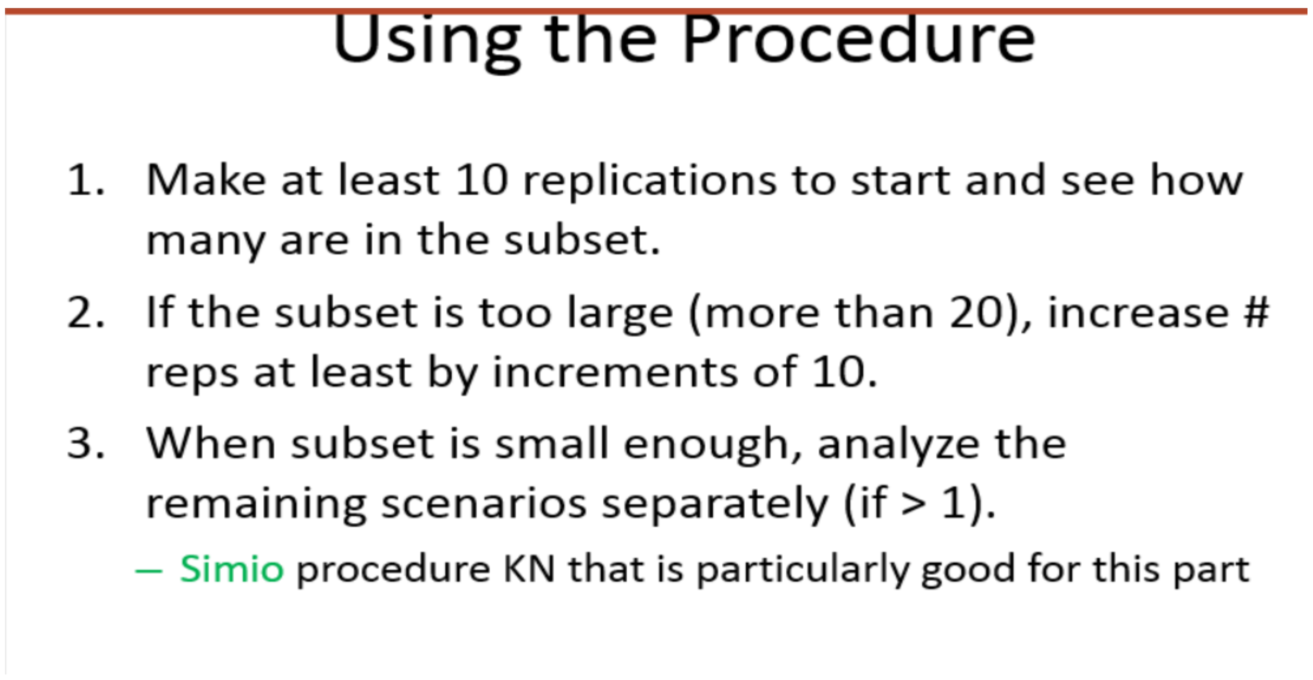


Subset Selection Intuition

In the example Scenario 1 is the sample best

Every scenario whose sample mean falls in here is statistically indistinguishable from Scenario 1

Every scenario whose sample mean falls out here is statistically inferior to Scenario 1



Using the Procedure

1. Make at least 10 replications to start and see how many are in the subset.
2. If the subset is too large (more than 20), increase # reps at least by increments of 10.
3. When subset is small enough, analyze the remaining scenarios separately (if > 1).
   – Simio procedure KN that is particularly good for this part

KN(Add-IN)

What does this add-in do?

It takes an initial set of scenarios and runs additional replications until it is confident that a particular scenario is the best, or within a specified range of the best (called the indifference zone). It will uncheck all scenarios except for the best. As it is executing it will "kick out" scenarios that are not a candidate for the best, and no more replications will be performed on those scenarios.

When to use this add-in?

This Add-In is to be used with a set of "interesting" scenarios. An interesting scenario is one that after running an initial number of replications still may possibly be the best. To achieve this, simply run your entire set of scenarios and make use of Simio's Subset Selection feature. This feature will present to you a set of Possible Bests; sometimes this set will be just one scenario while other times it may be several. This Add-In is designed to take it one step further to attempt to narrow it down to just one best as opposed to a set of possible bests.

How does it work?

The first thing that the algorithm does is force the scenarios with fewer replications to run until every scenario has the same number of replications completed. The Add-In will then run a fixed number of additional replications for each scenario and analyze the results to determine if any of the candidate scenarios can be deemed as rejects. If a scenario is rejected, it gets unchecked and no longer included in the selection process. These scenarios will be removed from the Response Chart. It repeats this process, running a fixed number of replications for each remaining candidate until one candidate can be selected as the best.

How to use the Add-In

As stated earlier, you must first run an initial number of replications in order to determine a set of interesting scenarios. Then you must select the *Select the Best Scenario using KN* Add-In in the Add-Ins section of the Design Ribbon.

After the Add-In is engaged, the Experiment Properties will now show a "Select the Best Scenario using KN – Parameters" section with three new Parameters:

1. Confidence Level – The statistical confidence in which the algorithm is to compare scenarios
2. Indifference Zone – This is number is the smallest meaningful distance between the best response so far and the other candidate responses, and is required for the Add-In to operate. If two response values are separated by less than this number (i.e. they are in "The Indifference Zone") they are considered to be the same and either can be considered to be the best.
    - *For example, you are buying a car and have an indifference zone of $300 with an objective to minimize cost. If one car costs $25,600 and the same model costs $25,800, you would consider them to be the same price. You would keep looking at other versions of the same model until you found one listed at $25,400. Now, the $25,800 is rejected and you continue to look for cars until you have exhausted your search and are down to one for $24,850 and $24,900. You have two cars that are considered to have the same price and now you make your decision based on color instead of price.*
3. Replication Limit – The maximum number of replications you are willing to wait for the algorithm to find the definite best. If it reaches this limit, and no definite best can be determined, it will just leave the possible bests left checked while un-checking the rejects.

The Add-In can only analyze one response at a time, so you must define a Primary Response in the Analysis window. This is the Response column that you are going to base your decision upon and represents the quality of your system, similar to an Objective Function. You must also define and Objective (Maximize or Minimize) so that the Add-In can determine what is considered "Best" – lower values or higher values.

Once you have all your parameters defined correctly, press the Run button and the Add-In will do all of the work. After, the add-in is done running there will be only one scenario left checked in the Design Tab and only one scenario in the Response Chart.