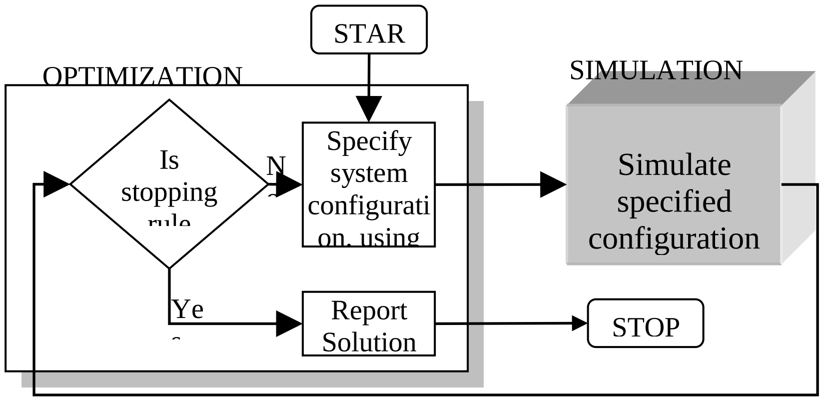
Simulation-based Optimization

- Need to interface an optimum-seeking or *optimization* package with simulation
  - Simulation model replaces the analytical objective function and the constraints



OptQuest: Uses implementations of Scatter Search, Tabu Search and Neural Networks. Their specifics are beyond this course. We shall learn how to use it.

Terminologies in OptQuest

Control Parameters: Are *variables* or *resources* that can be meaningfully manipulated to affect the performance of a simulated system. For example: the quantity of products to makes, the number of workers per activity etc. OptQuest will change the values of these controls with each simulation until we get the best objective.

Responses: Are the outputs from simulation model that is of interest.

Constraints: Used to define the relationships among controls and/or responses.

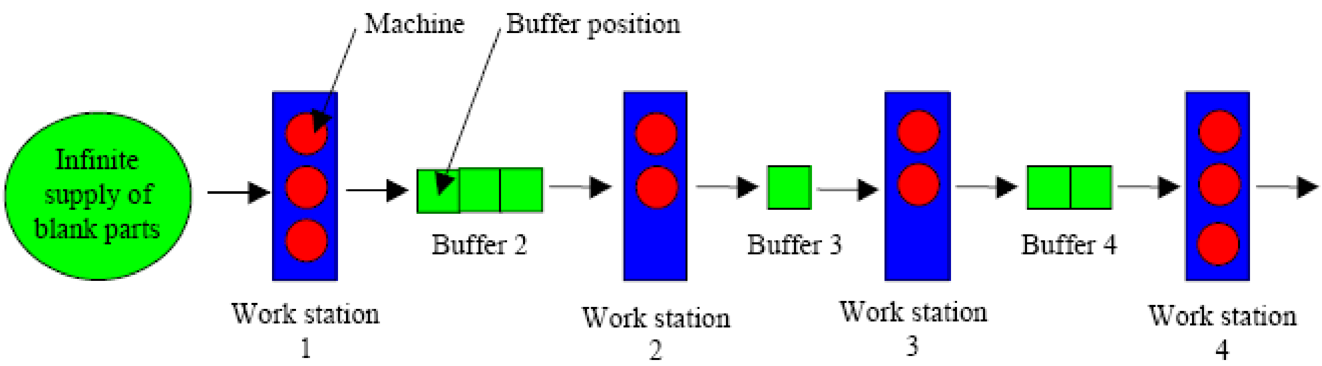
> *In the case of control parameter constraints*: For example, a constraint might ensure that the total amount of money allocated among various investments cannot exceed a specified amount, or at most one machine from a certain group can be selected. Constraints on the controls will be satisfied BEFORE each simulation run!
>
> *In the case of response constraints (requirements)*: For example, a constraint might be that the percentage of tardy jobs should be 90%. We can get these data only after the simulation run. So, constraints on the responses are checked for feasibility AFTER each simulation run!

Objective: Is a response or an expression used to represent the model's objective, such as minimizing queues or maximizing profits.

Using OptQuest (Eg. from Law and McComas, 2000, Winter Simulation Conference)

System Description

- Consider the manufacturing system consisting of 4 work stations and 3 buffers (queues) as shown in Figure. Whenever a machine in work station 1 is idle, it will pull a blank (new) part in from an infinite supply. A machine cannot discharge a part to downstream machine if the succeeding buffer is full. Processing times of machines (in hrs) have exponential distribution with mean time as follows: 0.3333, 0.5, 0.2 and 0.25 hrs for machines in workstation 1, 2, 3 and 4, respectively.



- **Task**: Determine optimal number of machines in each workstation and the optimal number of slots (capacity) in each queue, to **maximize** the **profit** over a 30-day production period.
- **Constraints:**
  - Number of machines in each workstation can be varied between 1 and 3.
  - Number of slots in each queue can be varied between 1 and 10.
  - Total machines in system should not exceed 10.
- **Profit** = ($200*throughput) – ($25000*number of machines) – ($1000*number of slots in queue)
  - where, Throughput = total parts produced in that 30-day period

- The simulation run length for is 720 hours (30 days) with an *additional* warmup period of 240 hours (10 days). Thus, the throughput & profit is computed from the final 720 hours of each 960-hour replication.
- If we do not use OptQuest, how many simulation runs do we need, i.e. how many combinations needs to be checked?

- Download and open 'SimOptBasic.alp' from course website. Get familiar with the model.
    - Each Workstation has a generic format of *Seize* → *Delay* → *Queue* → *Hold* → *Release.*
        - When an entity come to the "*Seize* - *Delay* - *Release*" modules, it will 'capture' 1 machine from the defined workstation ResourcePool, get delayed for specified time (in delay module) and then will release the resource to proceed downstream.
        - The HOLD blocks checks if the downstream buffer has enough space. If not, it blocks current machine by not releasing machine after use.
            - See the 'on Enter' and 'on Exit' properties of SEIZE module subsequent to a HOLD module.
    - Seven parameters have been defined using PARAMETERS module from Agent Palette: "Buffer2", "Buffer3", "Buffer4" defines the capacity of queues between Workstation 1&2, Workstation 2&3, Workstation 3&4 respectively. Parameters WS1, WS2, WS3, WS4 represents the capacity of the four workstations.
    - Two variables are defined using VAIRABLES module from Agent Palette, "throughput" and "profit" are defined. "throughput" counts the entities that have gone through the system; "profit" computes the total profit as per expression given in previous page.
    - 'Throughput' are computed in the last 'Sink' module. 'Profit' is computed in the *EventProfit* module
    - Simulation Run Length = 960 hrs of which 240 hrs are warm-up (defined by resetting statistics in *EventWarmup* module. So, effective run length is?

Explore the model. Run the model. Observe the results.

BUILD OPTIMIZATION

1. In Projects view panel, right-click on 'Main', and select New>Experiments. This opens a new dialog box.
    a. Select Optimization from the 'Experiment Type' list.
    b. Change 'Name' to *Optimize*
    c. Click FINISH.

2. Click *Optimize: Main* from the Projects view panel, and navigate to its properties. In Properties Panel,
    a. Enter objective as *root.profit* (you need to **maximize** this!)
    b. Under 'Parameters', you will see all the queue and WS parameters we had defined in the model.
        1. Select the 'Type' as ***design*** for all 7 parameters
        2. For queue parameters, set 'Min' as *1*, 'Max' as *10*, 'Step' as *1*
        3. For WS parameters, set 'Min' as *1*, 'Max' as *3*, 'Step' as *1*
    c. In Constraints tab
        i. In 'Constraints on Simulation Parameters', write under 'expression': *WS1+WS2+WS3+WS4*, under 'type': <=, under 'bound': *10*
        ii. Check the enabled box next to the constraint.
    d. In Randomness
        i. Check 'Random Seed'
    e. Now, click 'Create default UI'
3. RIGHT-Click *Optimize: Main* from the Projects view panel, and click. Observe the results.
    a. Note down the best solution you get (single rep case)

We can also do multiple replications with Optimisation. To do that:

In *Optimize: Main*, Properties panel,

a. In Replications tab, check 'Use Replications'
    i. Check 'Use replications'. Set 'Replications per Iteration' to 5.
b. Again, click 'Create default UI'

Run the Optimization to observe the results. Note down the best solution you get (multi-replication case)

TO DOs

1. In the *Optimize:Main* Properties Panel,
    a. Randomness: Use ‘Fixed Seed’ with your birth day or birth month as the seed number.
    b. Replications: Check ‘Use replications’. Set ‘Replications per Iteration’ to 5.
    c. Optimise your model and note down the solution and objective value.
        - Compare the results obtained with that presented in Law and McComas, 2000, Winter Simulation Conference (paper available in class site; see last paragraph of Section 3 in the paper)
    d. (Optional) Directly enter the Best Solution you got (multiple replications scenario) as the corresponding Parameter values in the Simulation model. Run the model for 10 replications and build a 95% CIs of the mean profit and average hourly throughput.

2. Add the following constraints to the above model and perform simulation-optimization to determine the new optimum configuration and the associated profit:
    a. Total number of machines in the system ≤ 10
    b. Total number of buffer spaces in the system ≤ 20.
    c. Optimise the model with these additional constraints (and with same Fixed Seed and Replications as for (1)). Report the final solution & objective value obtained.