Notes of Week Jan 09, 2017

Jianqiu Wang

January 11, 2017

1 Study of PySOT

1.1 Brief Introduction to PySOT

PySOT is a tool box designed for solving both continuous and discontinuous surrogated optimization problems.

2 Apply PySOT to Two-node Network

Based on this paper??, we simply consider a network with one supplier and receiver. We use the case in part 7, only consider node 1 and node 4.

2.1 Important Concepts

- On-hand inventory The inventory in stock
- On-order inventory The ordered inventory but not yet shipped
- Backorder Unsatisfied order
- Inventory position Amount of order we ́have: on-hand + on-order - backorder

2.1.1 Model Assumption

- A demand-driven inventory system under base-stock policy and order rationing policy
- General network structure for the inventory system (all the pri- mary supplier, secondary supplier and direct customer node(s) of each node are designated, if any) instead: one supplier and one director customer node
- Length of planning horizon: 200 days with 100 days of warm-up simulation
- Length of review cycle for each inventory
- Probability distributions of demands at sales regions node 2: Normal(150,30)
- Probability distributions of the delivery preparation times (include but not limited to time for reprocessing, transportation, sub-packaging, etc.) at each inventory node node 1: Uniform(2,4)
- Lower bounds for service levels at each node: node 1: 0.70 node 2: 0.95
- unit holding cost, unit backordering cost: 1 m.u./unit/day

2.1.2 Objective

2.1.3 Source of Uncertainties

3 Simulation Process

3.1 Discrete-Event System

When running stochastic simulation of discrete-event system, we treat them as as generalized semi-Markov processe (Stochastic Simulation, page 65). We define two sets: set S: states of node, and set E: set of possible events that can trigger state transitions. We will determine state transitions by competing clocks: when a event $e \in E$ is scheduled, the clock runs down at predetermined rate, and when it counts to 0, the event happens and state changes. Then usually we need to reschedule new event.

3.2 Simulation Algorithm for a general model

- 1. Initialization: Set the simulation clock T to 0. Choose the initial system state X and event clock readings $\{C_i\}$.
- 2. Let $T = \min_i C_i$ be advanced to the time of the next event and let I be the index of the clock reading that achieves this minimum.
- 3. Execute the logic associated with event I, including updating the system state X and event clock $\{C_i\}$.
- 4. Go to Step 2.