SIMULTECH 2023, Rome 12-14 July 2023

InventOpt

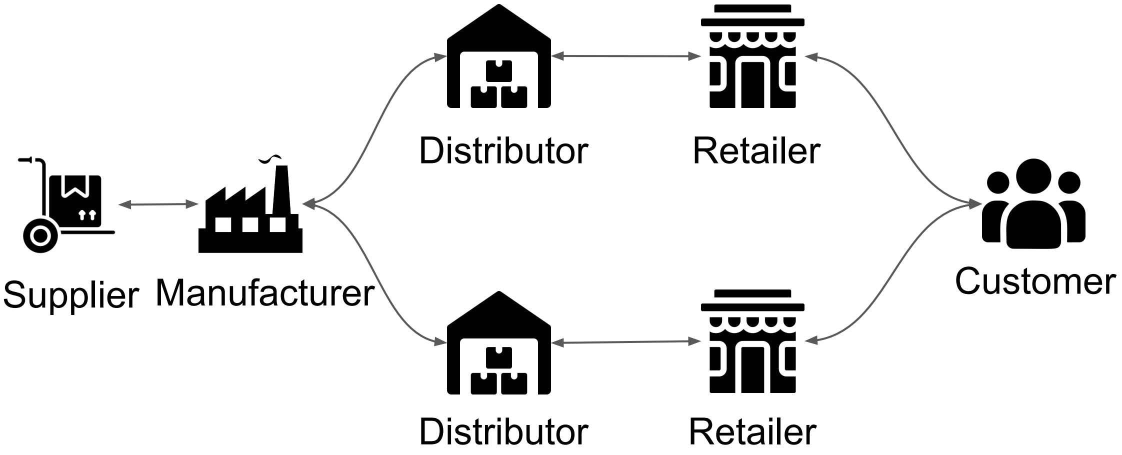
An Open Tool-set for Simulation, Design-space Exploration and Optimization of Supply Chains and Inventory Systems

Tushar Lone
(PhD student)

Co-authors: Lekshmi P (PhD student), Neha Karanjkar (Assistant Professor)

School of Mathematics and Computer Science
Indian Institute of Technology Goa (IIT Goa)

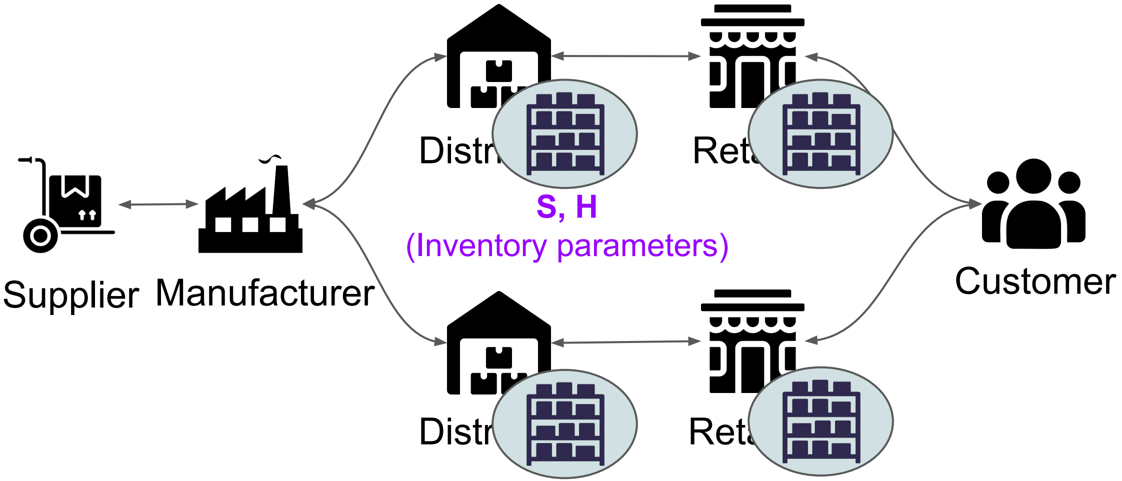
Introduction - Supply Chain Networks



Goals: Maximize profitability, minimize risk, improve customer satisfaction …

Icons downloaded from: *flaticon.com*, *vectorstock.com*

Introduction - Supply Chain Networks



Goal: Find optimum inventory parameter values to maximize SC profit

Icons downloaded from: *flaticon.com*, *vectorstock.com*

Simulation based optimization in Supply Chain

- Real SC networks have hundreds of nodes
- Huge design space
- Multiple objectives to optimize

Simulation-based Optimization (SBO)

- Discrete Event Simulation (DES)
- Agent based Simulation (ABS)
- Continuous
- Monte-Carlo

The simulation model is computationally expensive

Simulation based optimization in Supply Chain

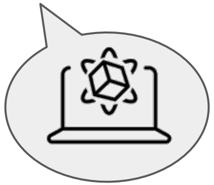
- Real SC networks have hundreds of nodes
- Huge design space
- Multiple objectives to optimize

Simulation-based Optimization (SBO)

- Discrete Event Simulation (DES)
- Agent based Simulation (ABS)
- Continuous
- Monte-Carlo

The simulation model is computationally expensive

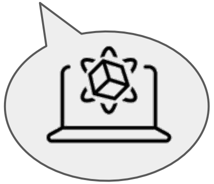
Objective function
F(X)



Simulation based optimization in Supply Chain

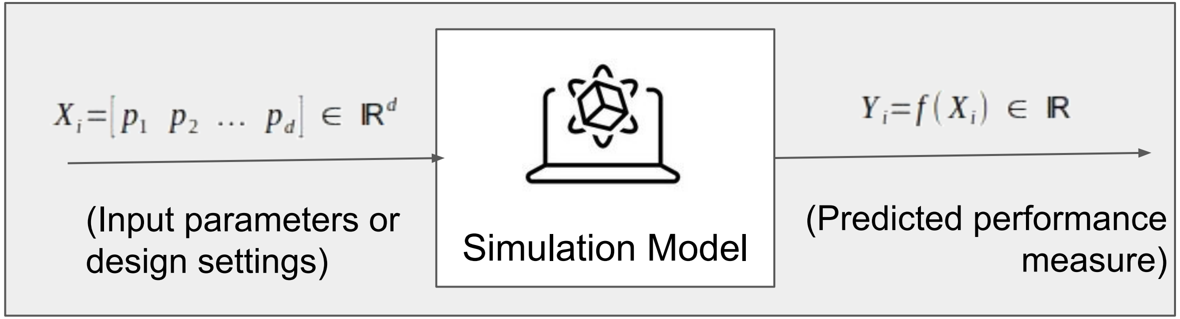
- Real SC networks have hundreds of nodes
- Huge design space
- Multiple objectives to optimize

Objective function
F(X)



Simulation-based Optimization (SBO)

- Discrete Event Simulation (DES)
- Agent based Simulation (ABS)
- Continuous
- Monte-Carlo



The simulation model is computationally expensive

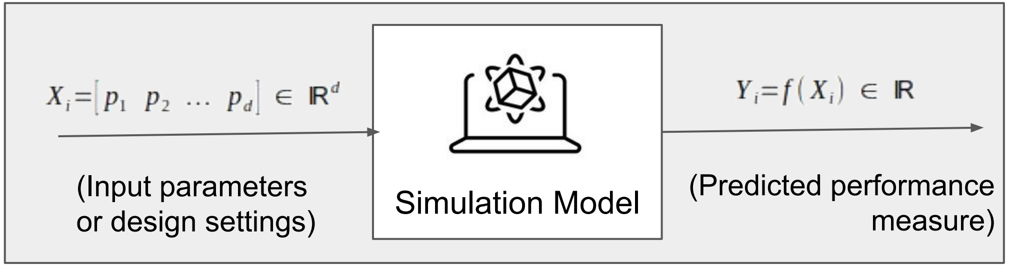
Meta-model based Optimization in Supply Chains

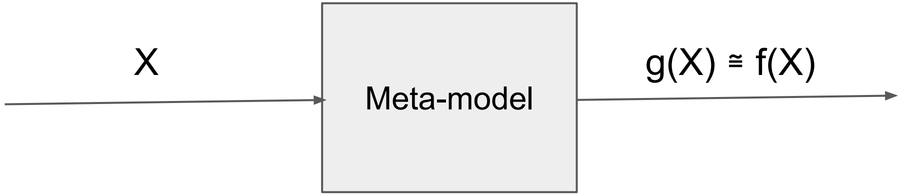
- Model of a model
- Generates approximation function g

  g(X) ≅ f(X)
- Less accurate, computationally inexpensive

Types

- Response Surface Meta-model (Linear, polynomial regression)
- Spatial Correlation meta-model (Kriging - Gaussian Process Regression)
- Neural Network Meta-model





(Lower Accuracy, computationally inexpensive)

Existing tools and platforms

- Commercial Tools
  - AnyLogic,
  - FlexSim,
  - Arena,
  - IBM Supply Chain Solutions
- Open Source
  - miniSCOT,
  - Supplychainpy

Commercial

Not tailored for
Inventory optimization

OptQuest
Optimization Engine

No support for Meta-model based
optimization

InventOpt

An open Python library and and tool-set for modeling, simulation, design exploration and optimization of Inventory problems in Supply Chain

- Open Source
- Variety of Inventory nodes
- Components Lib
- Support for parallelization
- General Framework
- GUI tool for visual aid
- Multiple Inventory policies
- Variety of transportation models
- Variety of Meta-models and Optimizers

InventOpt

Components:

- Modeling and simulation Library
  - Component based library
    (create/delete and connect supply chain nodes, and simulate)
- Cost vs Accuracy analysis (Py Scripts)
  - Access the execution time and accuracy of stochastic models
- Design Exploration Library
  - Experimental setting for meta-model based optimization
  - Running the model for N points in design space
- Meta-modeling
  - Variety of meta-models (parameter tuning, accessing goodness of fit)
- Optimization
  - Variety of Optimizers
  - Bayesian Optimization support
- GUI

DataVis (https://datavis.streamlit.app/)

InventOpt

- Python based
  - SimPy (DES)
  - scikit-learn
  - SciPy
  - Multiprocessing
  - Streamlit
  - Plotly

Modeling and Simulation

Meta-modeling, Optimization

Design exploration

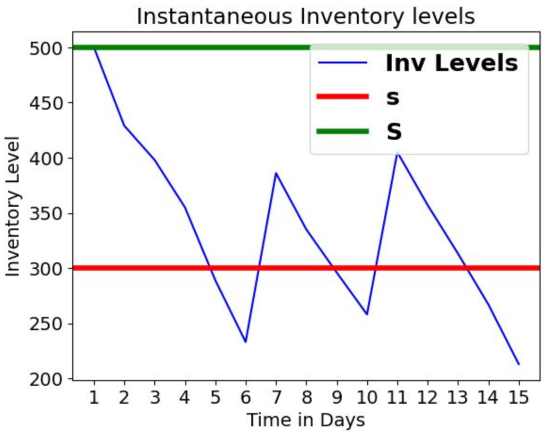
- In early phase of development
- Open source MIT licence

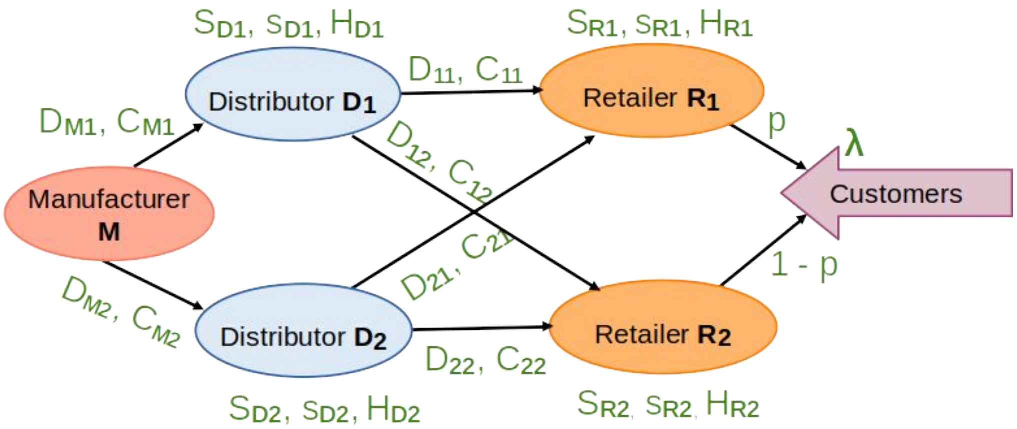
Case Study

- Meta-model based optimization process flow
- Address design decision made in the process

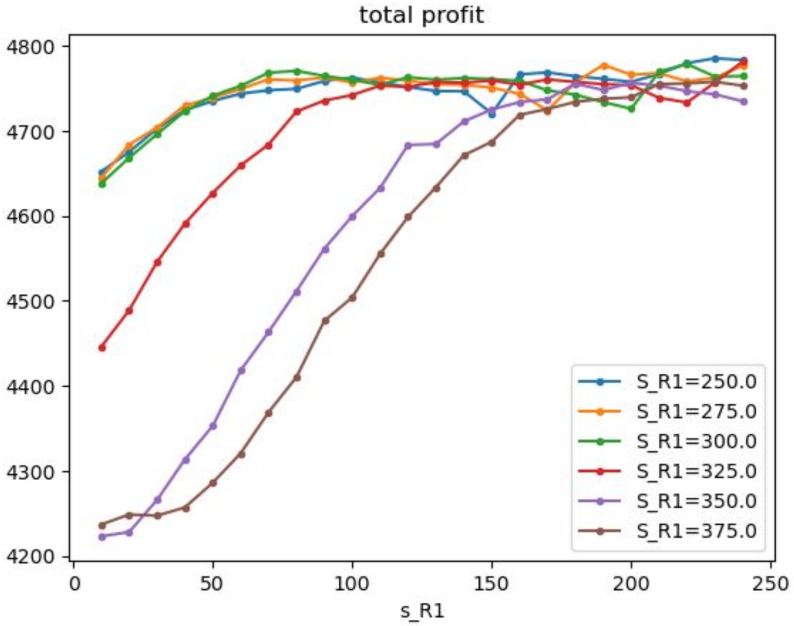
Case Study

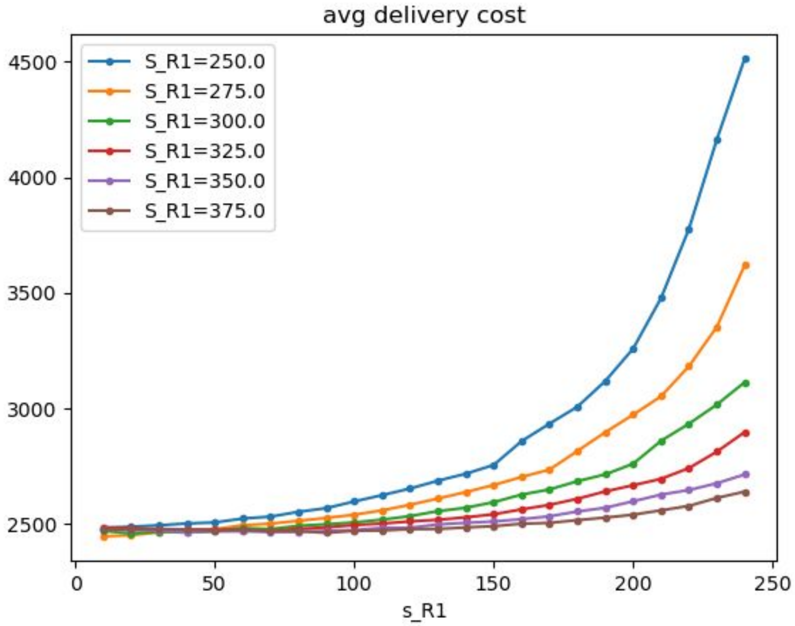
- To demonstrate the flow
- To figure out
    - Which meta-model to choose
    - How many training points are sufficient to produce a good fit
    - Which optimization method to choose





Case Study - Model validation





Case Study

Components

- Modeling (SimPy) (Line of Code = 331)
- Py scripts
  - access accuracy vs computational cost (LoC = 118)
  - Design exploration (LoC = 113)
  - Meta-model based optimization (LoC = 150)

Experiment Setting

- $4^8 = 65536$
- evaluating performance measures at single point in design space is ≈ 130 sec
- With parallel aid (speed up of 64x) ≈ 2 days

Case Study - Design Exploration

Select the components to visualize as a 3D slice:

x-axis: S_R1

y-axis: s_R1

function component: avg_net_profit

Select the range of values for the other axes

S_R2 range: 347.17 – 397.47 (300.00 – 450.00)

s_R2 range: 184.18 – 250.00 (100.00 – 250.00)

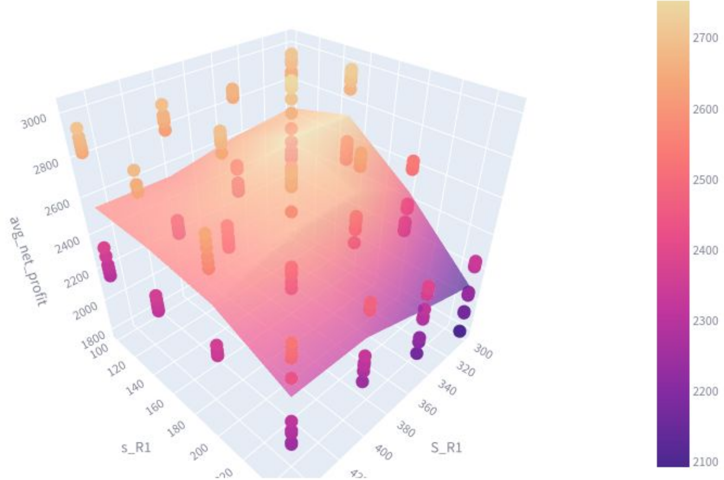
S_D1 range: 649.05 – 750.00 (600.00 – 750.00)

s_D1 range: 378.80 – 454.43 (350.00 – 500.00)

S_D2 range: 643.04 – 683.23 (600.00 – 750.00)

s_D2 range: 451.27 – 500.00 (350.00 – 500.00)

The selected slice contains 192 points



DataVis (https://datavis.streamlit.app/)

Case Study - Meta-model based optimization

(Meta-models are built with 25, 50 and 75% of the data)

Neural Network Meta-model

- Multilayered Feed Forward NN
- NN parameters - (1) number of hidden layers, (2) number of neurons in a layer

Kriging (Gaussian Process Regression (GPR))

- Spatial correlation meta-model
- Different kernels - RBF, Periodic, Gaussian Kernel

Optimizers

- SLSQP, Nelder-Mead, COBYLA, Powell

Case Study - Results

Observations and Results

- GPR stands out from NN
- 25% training data is sufficient for GPR to obtain good results
- Powell and Nelder-Mead performed better with GPR metamodel
- NN with Nelder-Mead failed as it is unconstrained optimization method
  It produced results outside the bounds

Conclusions

- The design decisions for optimization process are crucial
  - Meta-model type
    - How many points to sample
    - Which sampling method
  - Optimizer type
- Study inferes the need of an open source, easy to use, generic library and tool - InventOpt
- The case serves as a basis for implementation of the InventOpt tool
- InventOpt is work in progress and will released by July 2024, with open source MIT licence

Summary

Inventory optimization problems in supply chain

- Large number of input parameters, huge design space

Simulation based optimization methods are time consuming

- Does not ease use of gradient based optimization

Meta-model based optimization - making design decisions is crucial

- Which meta-model to choose
- How many points to sample, how to sample
- Which optimizer is suitable for selected meta-model
- **InventOpt**

Thank You!

InventOpt

It is in early phase of development.

It is open Python library and GUI tool for **modeling**, **simulation**, **design exploration** and **optimization** of inventory problems in Supply Chain.

Tushar Lone (PhD scholar) SMCS, IIT Goa, India.

Contact Mail: tushar20231102@iitgoa.ac.in

LinkedIn https://www.linkedin.com/in/tushar-lone-40097477/

GitHub: https://tusharlone.github.io/