Computers and Chemical Engineering 87 (2016) 164–179

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A computationally efficient simulation-based optimization method with region-wise surrogate modeling for stochastic inventory management of supply chains with general network structures

Wenhe Ye, Fengqi You*

Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL 60208 USA

ARTICLE INFO

Article history:
Received 7 May 2015
Received in revised form 22 January 2016
Accepted 22 January 2016
Available online 1 February 2016

Keywords:
Simulation-based optimization
Inventory management
Trust-region algorithm
Surrogate modeling
Kriging

ABSTRACT

Simulation-based optimization is widely used to improve the performance of an inventory system under uncertainty. However, the black-box function between the input and output, along with the expensive simulation to reproduce a real inventory system, introduces a huge challenge in optimizing these performances. We propose an efficient framework for reducing the total operation cost while satisfying the service level constraints. The performances of each inventory in the system are estimated by kriging models in a region-wise manner which greatly reduces the computational time during both sampling and optimization. The aggregated surrogate models are optimized by a trust-region framework where a model recalibration process is used to ensure the solution's validity. The proposed framework is able to solve general supply chain problems with the multi-sourcing capability, asynchronous ordering, uncertain demand and stochastic lead time. This framework is demonstrated by two case studies with up to 18 nodes with inventory holding capability in the network.

1. Introduction

A supply chain is a complicated system containing networks of information flows and material flows (Chopra, 2010; Garcia and You, 2015; Simchi-Levi, 2008). A typical supply chain usually contains different but interdependent facility nodes for different purposes, ranging from the procurement of raw materials, product processing, delivery of finished products, etc. The efficiency of a supply chain can be evaluated from different aspects, including the economics, rapidness of service, environmental sustainability, etc. Therefore, the supply chain management is focused on improving the above metrics and it has become more and more important for manufacturing companies to achieve growth in profits (Grossmann, 2005; Relvas et al., 2006; Varma et al., 2007; Wassick et al., 2012). Among the supply chain system, the inventory plays a critical role for connecting different functional units into a highly integrated system (Zipkin, 2000). The inventory management concerns demand forecasting, physical inventory carrying and quality control. It is dedicated to coordinating the logistics with production planning, and it can also help the supply chain system to respond to various kinds of uncertainties (Michalski, 2009).

The objective of inventory management optimization under demand uncertainty is to maximize the performance of the inventory system in a stochastic environment by determining an optimal set of inventory control parameters with regard to a certain inventory control policy. This is a challenging problem since there are two key performance measures: maintaining a low operation cost and not violating the service level constraints. Combined with its stochastic nature, the problems are usually intractable (Cheng et al., 2003; Jung et al., 2004). Also, the idealized assumptions necessary for mathematical models make the solutions inapplicable for real world cases (Kochel and Nielander, 2005). In order to reproduce details such as the multi-stage nature of the network, fluctuation of demands, uncertain lead times, and the integration of versatile "intelligent" control strategies, the simulation method has become mainstream for the modeling of a real-world supply chain and has been adopted by an increasing number of companies to evaluate their performance (Shah, 2005).

However, there are still a few research challenges in simulation-based optimization approaches for stochastic inventory management. The simulation is usually computationally expensive since multiple replications are required to overcome the noise in the returned result. Also, in contrast with mathematical models, simulation provides a "what-if" response to the system inputs in which there is no accessible gradient information. Though there are general approaches for solving simulation-based optimization

* Corresponding author. Tel.: +1 847 467 2943; fax: +1 847 491 3728.
E-mail address: you@northwestern.edu (F. You).

http://dx.doi.org/10.1016/j.compchemeng.2016.01.015
0098-1354/

Nomenclature

Experiment design

N_d	number of days in a planning horizon
N_{inv}	number of inventory stocking nodes in the network
N_{rep}	number of simulation replications
N_s	number of sampling points

Index

i,	j inventory node
q	design point
t	planning period

Sets

I	inventories in the entire inventory network
D_i	customer nodes immediately linked to node i

Simulation parameters

c_i^b	unit backorder cost for node i
c_i^h	unit inventory holding cost for node i
$sl_i^{\min}$	service level lower bound for node i

Simulation variables

ABO_i	average backorder of node i
AOC_i	average daily operation cost of node i
AOH_i	average on-hand inventory of node i
BO_i	real time backorder of node i
BS_i	base-stock level at node i
FOD_{it}	sum of orders filled by node i without delay on day t
$IMFO_{ijt}$	immediately fulfilled order from node j to node i on day t
$INVB_{it}$	backorder amount of node i at the beginning of day t
$INVH_{it}$	on-hand inventory of node i at the beginning of day t
$INVO_{it}$	on-order inventory of node i at the beginning of day t
$INVP_{it}$	inventory position of node i at the beginning of day t
OD_i	order amount of node i at the beginning of day t
OH_i	real time on-hand inventory of node i
ROD_{ijt}	order received by node i from node j on day t
SL_i	service level of node i
SOD_{it}	sum of orders received by node i on day t
TLC	total daily operation cost of the supply chain
$TLOD_i$	total amount of orders received by node i over the entire horizon
$TLSO_i$	total amount of orders satisfied by node i without delay over the entire horizon

Parameters in optimization algorithm

Δ_k	trust-region size at iteration k
α	positive multiplier for model recalibration
δ	modified trust-region size
λ_i	error bound for the region-wise surrogate at node i
ε_i	error term for the region-wise surrogate at node i
υ_m	component m in vector υ
ρ	indicator of the similarity in the trust-region algorithm
η	indicator of the step-size in the trust-region algorithm

Vector/matrix

$\mathbf{X}_s$	design matrix for sampling
$\mathbf{X}_{s,q}$	the q^{th} design point in the design matrix
$\mathbf{x}$	vector of the base-stock levels
$\mathbf{x}_k$	vector $\mathbf{x}$ at the kth iteration
$\mathbf{y}$	result vector for the sample input vector

problems such as genetic algorithm (GA) and simulated annealing (SA), they are all metaheuristic approaches which cannot guarantee the solution's quality (Mansouri, 2006; Mele et al., 2006). An important branch of simulation-based optimization resorts to the use of surrogate models, where the black-box functions are sampled and predicted by analytical approximations ranging from linear regression to adaptive nonlinear models (Cozad et al., 2014; Wang and Shan, 2007). However, inventory control optimization has unique and unfavorable features. First, a supply chain network is usually composed of a number of facility nodes and the problem's dimension can be high if there are many control parameters to be determined. Second, both the objective and the constraints for the inventory control problem are black-boxes; hence, it is a significant burden to construct the surrogate function for each equation. In addition, many existing surrogate-based methods lack generality and are based on some premises such as normally distributed demand, divergent networks, convexity assumptions, etc. Thus, an efficient algorithm for solving more general inventory control optimization problems under uncertainty is the next step for research (Chu et al., 2015b; Jung et al., 2008).

Instead of delving into supply chains with idealized presumptions, we focus our solutions to inventory system simulators featured with more realistic considerations. The definition of "general" allows the inventory system to adopt more flexible inventory control policies and principles including multi-sourcing and asynchronous decision making. The change of inventory control parameters will not only affect the performance of each node in the system but also the expected material flow rates within the entire network. A concomitant challenge is that the simulation becomes a more unpredictable black-box response, losing benign characteristics such as convexity and the possibility to be simplified as multiple single-stage models. In this work, we propose a novel region-wise surrogate modeling method for solving the inventory management optimization problem for a general supply chain network under uncertainty. By integrating the design of a simulation experiment, simulation by objective-oriented programing, and kriging metamodeling, a trust-region framework with iterative model recalibration algorithm is introduced to tackle the general inventory control problem with multi-sourcing options and asynchronous reordering. We reduce the entire supply chain network into region-wise approximations and further extend the surrogate-based optimization framework's capability to solve a large inventory system under both demand uncertainty and lead time uncertainty.

The novelties of this work are summarized as follows:

- A novel region-wise surrogate modeling of an inventory system with complex network structure
- Trust-region based algorithm with iterative model recalibration for surrogate-based optimization
- Application of simulation-based optimization to general supply chain networks with multi-sourcing capability

The remainder of the paper is organized as follows. In Section 2, a literature review for recent studies on simulation-based optimization for inventory systems is presented. Our assumptions for

the general network structure and the base-stock control policy are introduced in Section 3. Section 4 presents the problem statement of inventory management optimization in general network under uncertainty. In Section 5, the simulation method and modeling procedures are explained. The efficient region-wise surrogate modeling optimization framework is stated in Section 6, where we also briefly introduce the surrogate-based optimization method to the black-box system. During Section 7, the proposed algorithm is demonstrated by two representative case studies and compared with using GA. Section 8 is the conclusion.

2. Literature review

The theory of stochastic inventory management with different modeling methods can be referred to in a great variety of literatures (Graves et al., 2000; Porteus, 2002; Zipkin, 2000). Based on these foundational works, extensive efforts have been directed towards the management of safety stocks under demand or lead time uncertainty if the inventory system follows the base-stock control policy (Ettl et al., 2000; Inderfurth and Minner, 1998; H. L. Lee and Billington, 1993; You and Grossmann, 2008, 2011a; Yue and You, 2013). Though successfully capturing the main features of stochastic inventory management, mathematical models omit a significant amount of details in order to make the model solvable. Therefore, the solution of the model may not be the true optimum for the real system. In order to make a clone of the real-world system, computer simulation methods developed rapidly during the past decade and simulation has become a tool for optimization purposes (Fu, 2002). The simulation modeling of inventory systems in a decision by decision procedure can be referred to as discrete-event simulation or the agent-based method (Chu et al., 2015b; Law and Kelton, 2000; J. H. Lee and Kimz, 2008). The simulation has neither analytical form nor gradient information, therefore the optimization usually relies on derivative-free optimizations (DFOs), statistical approaches like cross the entropy (CE) method, or metaheuristic algorithms mentioned in Section 1 (Conn et al., 2009; De Boer et al., 2005). A thorough comparison of the recently developed DFO solvers is available (Rios and Sahinidis, 2013).

Apart from the DFOs and the metaheuristic algorithms directly optimize the inventory simulation model, the use of hybrid models has been proved to be efficient. Hybrid models consist of both a simulation model and a mathematical model. The versatile combination of these two models attracts a multitude of research interests and can be applied to supply chains with different features. If a mixed integer linear programing (MILP) planning model is embedded in the simulation in a rolling horizon scheme, then the production and planning decisions under demand uncertainty can be made by accessing both the demand forecast and the real-time status (Jung et al., 2004; Subramanian et al., 2001). The use of these two models can also be carried out in a sequential approach, where the performance of the upstream supplier and downstream customer are captured by simulation and further linearized as an off-line linear programing (LP) model. Solving the resulting LP can lead to an improved solution regardless of the scale of the network (Jung et al., 2008). The hybrid use of mathematical and simulation models is also applied to solve supply chain problems with centralized or decentralized structure and is extended to tackle supply chain systems with synchronous and asynchronous decision making strategies by iteratively updating the solution between the two models (Chen et al., 2012; Mele et al., 2006; Sahay and Ierapetritou, 2014; Subramanian et al., 2000). Another important method to facilitate the use of mathematical models in solving simulation-based inventory and supply chain problems is to use surrogate-based optimization (Wan et al., 2005). The surrogate-based approach contains the design of simulation experiments and an iterative model updating strategy. If a multi-echelon supply chain operates in the "tight" region that a high service level constraint has imposed, the convexity of the simulation response allows the appended use of cutting-planes to the estimated surrogate model, allowing for convergence to a local optimum (Chu et al., 2015b).

Although the emergence of high-performance computer processors and software may facilitate both the simulation modeling process and computational process, major challenges still exist in the aforementioned methods. DFOs and metaheuristic methods can be regarded as genial approaches that purely use the inputs and outputs without using the knowledge embedded in the simulator. However, the time for optimization will grow enormously as the problem's input space expands. On the other hand, the incorporation of mathematical models into simulation models may greatly reduce the required computation time by utilizing the major correlations in the model. However, this method usually requires some certain restrictions and assumptions, and the compatibility and even the feasibility can be compromised when applied to more realistic cases. A general inventory system may contain features which cannot be generalized by the mathematical formulations. Nevertheless, the supply chain is a highly organized group of nodes, and the performance of each facility deeply relies on its position in the network. A possible solution to enhance the computational efficiency can be accessing the structural information of its network, which is studied in our work.

3. Background

In this paper, we propose a computational framework for solving the inventory optimization problems for general supply chain networks with given structure. The inventory system is a demand-driven network, and each inventory is ruled by the base-stock policy. The customers are grouped into several sales regions and external orders can be directed to any facility in the network.

3.1. Base-stock policy

The inventory system operates in a discrete-time manner and adopts the base-stock policy. The base-stock policy is one of the most basic inventory policies widely used, and it only contains one control parameter, namely the base-stock level for each inventory. We consider the system's behavior over a certain length of planning horizon, a total of N_d periods, and all the actions taken by the inventories are counted in days. The daily control of the inventory at node i on day t is governed by Eqs. (1) and (2).

$$INVP_{it} = INVH_{it} + INVO_{it} - INVB_{it} \tag{1}$$

$$OD_{it} = BS_i - INVP_{it} \tag{2}$$

The amount of order is equal to the gap between the base-stock level and the previous inventory position, which is the sum of the on-hand inventory with the total amount of orders not yet arrived, less the amount of backorders.

The dynamic illustration for the base-stock policy is shown in Fig. 1.

3.2. General inventory network

Unlike a typical tree structure network where one node can only be assigned to one predecessor, we extend the rule and stipulate that each inventory stocking node in the network may have up to two suppliers. Though the change from one to two may not seem to be an ambitious one, the network we are interested in can be fundamentally different from the others. Fig. 2 illustrates the general network, and each node from 1 to 4 contains an inventory storing

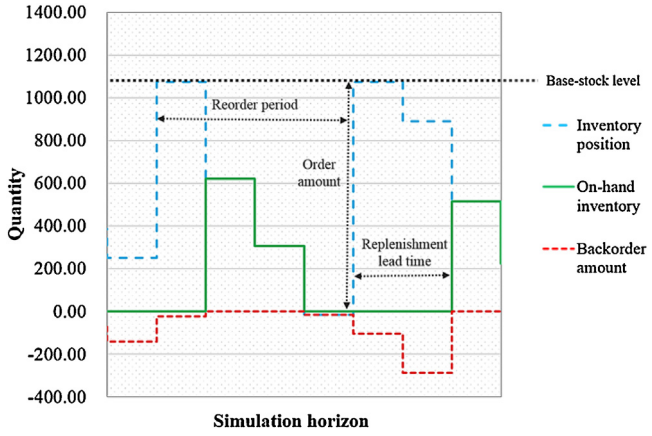

Fig. 1. Diagram of the base-stock policy. The backorder amount is presented in negative value for clarity.

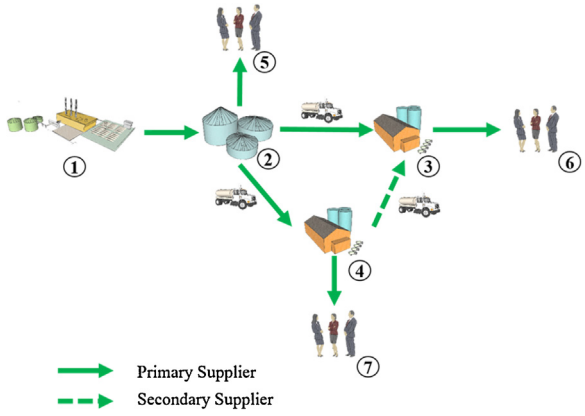

Fig. 2. Illustration of the general network structure for an inventory system.

site. The first difference compared to a purely divergent network is that the multi-sourcing capability which provides the network with additional flexibility to overcome the demand surges when one of the two entities experiences stockouts. This rule allows the inventory at node 3 to be served not only by the primary supplier, the warehouse at node 2, but also by the secondary supplier at node 4. The second difference is that the general network allows multiple manufacturing sites in a single network while the tree-like divergent network can only have one root. The above deviations in network structure allow many realistic features in the inventory system especially when a nation-wide or even an intercontinental supply chain is being studied.

4. Problem statement

The supply chain is dedicated to serve all demands from the sales regions, which occur during each planning period and follow some probability distributions. Each inventory manager monitors the inventory position and places orders to its supplier nodes to maintain a certain level of base-stock once per given cycle of reorder period. Orders are processed by the suppliers with their on-hand inventory and can be either fulfilled immediately or backlogged. If the order is to be fulfilled by the manufacturing plant, it can always be satisfied without delay. In the case when the node has two suppliers, the order from which can be satisfied is determined by a coordination procedure. If the orders from a direct customer can be satisfied by the on-hand inventory, a stochastic length of time is needed to prepare the deliveries. Therefore, the replenishment lead time will become uncertain as a comprehensive outcome by the above factors. Details of the operation procedures can be referred to in the simulation details stated in Section 5.1.

The carrying of products in the inventory will incur inventory holding costs, and the backlogged orders will incur backordering costs as a penalty of the delay. The quality of service is also quantified by the service level for each inventory, defined as the expected fraction of orders that is satisfied immediately by the on-hand stock over the entire planning horizon. The objective is to minimize the inventory system's daily total operation cost which is the sum of the total inventory holding cost and the total backordering cost per day for the entire system. We also impose a set of service level lower bounds to all the inventories as a need for quality control. Overall, we state the inventory management optimization problem below.

Given:

- A demand-driven inventory system under base-stock policy and order rationing policy
- General network structure for the inventory system (all the primary supplier, secondary supplier and direct customer node(s) of each node are designated, if any)
- Length of planning horizon
- Length of review cycle for each inventory
- Probability distributions of demands at sales regions
- Probability distributions of the delivery preparation times (include but not limited to time for reprocessing, transportation, sub-packaging, etc.) at each inventory node
- Unit inventory holding cost
- Unit backordering cost
- Lower bounds for service levels at each node

Source of uncertainties:

- Stochastic demands from sales regions
- Stochastic delivery preparation times
- Stochastic production times

To be determined:

- The optimal base-stock levels setting for the entire system

Objective:

- To minimize the daily total operation cost (the sum of the inventory holding cost and the backordering cost)
- To keep service levels above lower bounds for each node

It is worth noting that our model only considers the most essential parts of the cost breakdown, namely the variable cost and omits a universe of fixed cost for simplicity. The above statement for the inventory management under general assumptions allows greater freedom than a typical tree-like divergent network does. In a single-sourcing inventory system, the average material flow on each branch is determined by the mean value of the aggregated demand in the end. However, this rule no longer holds while introducing the multi-sourcing capability. A partially satisfied order can be diverted to the secondary supplier at any time; namely the change of base-stock levels will not only affect the service levels but also change the average material flows on each possible branch. As a result, the input variables to the simulator have complicated influences on the outputs.

5. Simulation details and model formulation

To solve the simulation-based optimization, we need to design a computer experiment to simulate the operation of an inventory

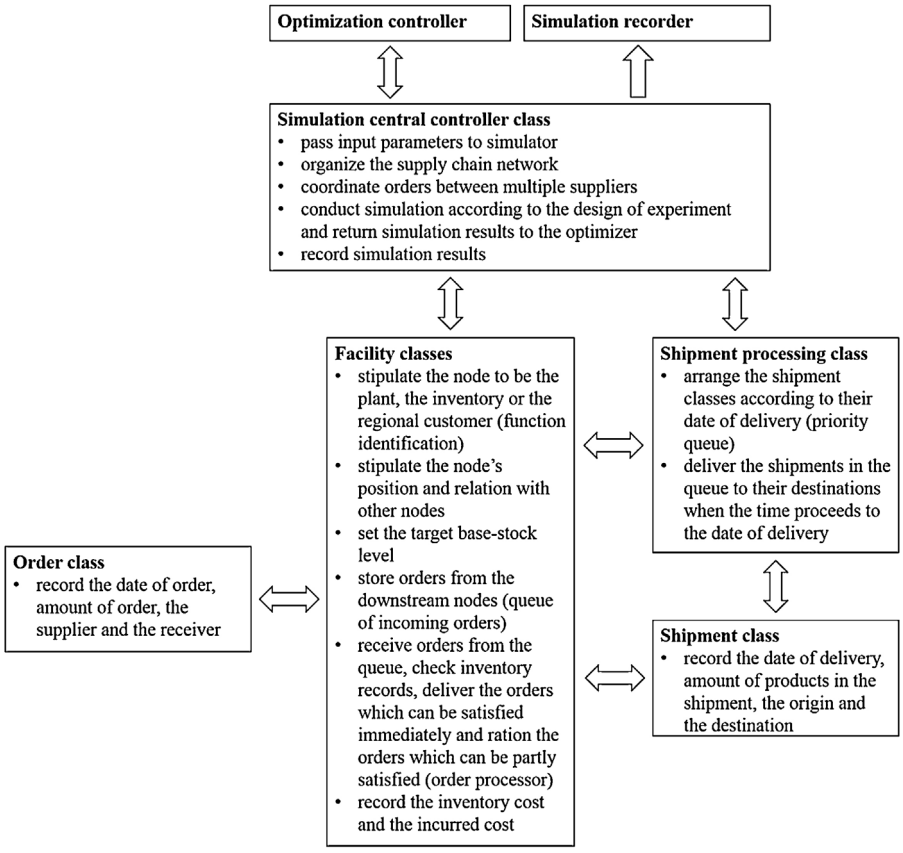

Fig. 3. A few important elements and their relationships that comprise the inventory system simulator while using objective oriented programing.

system. Section 5.1 illustrates the process of inventory simulation with a general network structure. In Section 5.2, we formulate the corresponding optimization problem with the simulation model.

5.1. Simulation model for the inventory system with general network

The inventory system is a complex network containing different functional supply chain nodes. It can usually be identified as a network of material flows in which the manufacturing plants are the sources of the flows and the customer nodes are the sinks. In the determinant cases, these flows can occur either constantly or periodically without variations. However, under general network assumptions where each inventory node may have up to two potential suppliers combined with demand uncertainty, the product flows occur in the network can become transient and unpredictable. In order to capture the performance of each node and to evaluate the total operation cost of the entire network, we use Monte Carlo simulation to estimate the expected total cost by averaging over a sufficiently large but finite number of simulation replications.

In order to simulate the supply chain system to a realistic level while harmonizing computation efficiency with the complexity of operating the inventory and material flows, we utilize the objective oriented programing (OOP) approach to reproduce the whole inventory system by abstracting the main functional units into classes, fields, and methods. We select a few important classes (subclasses) and list them in Fig. 3. Some of the important fields and methods embedded are also described in the boxes. An illustrative figure of the simulation model structure can be found in Fig. 4.

Fig. 5 illustrates the entire simulation process though a flowchart. During each simulation replication, the simulator is first

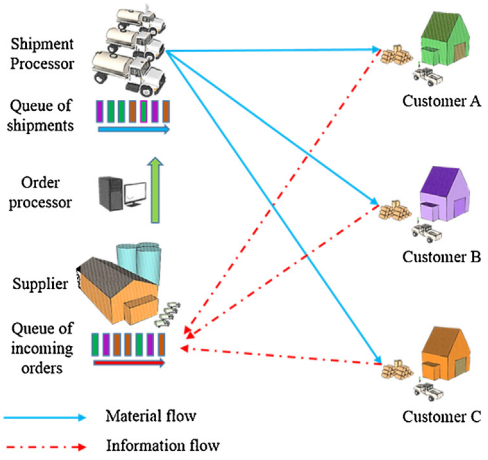

Fig. 4. Illustration of the simulation model for the inventory system.

initialized and the simulation clock is set to zero. Each planning period is one day, and each replication has a planning horizon of N_d days. All the inventory nodes manage their inventories according to the base-stock policy. That is, at the beginning of the planning period t, each inventory node $i \in I$ checks its inventory position, denoted by $INVP_{it}$, if the current period is the starting period of a reviewing cycle. An order OD_{it} is then generated to replenish the inventory position to the desired base-stock level BS_i. The customer nodes orders are generated by a random number generator following a known distribution to simulate the stochastic demand. When the generated orders are received by the primary suppliers, the shipment processor then clears all the delivery tasks due during the current period, and all the inventory records are updated

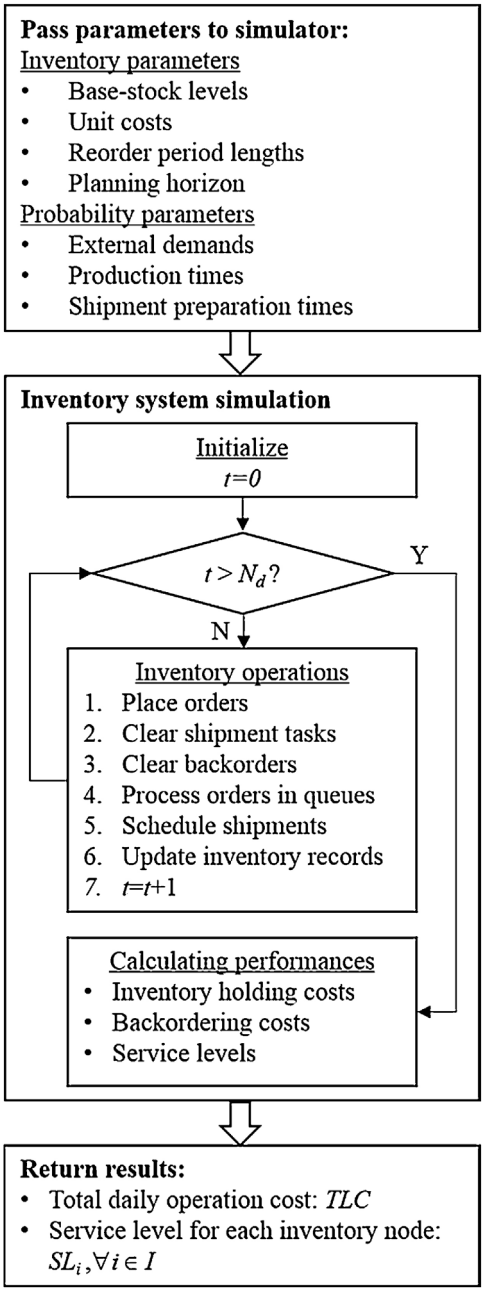

Fig. 5. The flowchart of the procedures for simulating the operations in the inventory system.

subsequently. After all the shipments are delivered, the inventory manager should start to prepare to deliver the products by checking their on-hand inventory level OH_i, which is the sum of the on-hand inventory level they already have at the beginning of the period and all the shipments arrive during the current period. The order processor in each inventory node checks the orders in the entire queue, and all the backlogged orders will be cleared first. Due to the introduction of a queue, the backorders are automatically handled in a first-in-first-out (FIFO) principle and the computation efficiency is significantly promoted without labeling each order with a timestamp. If the total order amount waiting in the queue can be satisfied by the remaining on-hand inventory, then all the orders are fulfilled immediately, and shipments are scheduled and pushed to the shipment processing queue. However, if the primary supplier does not have sufficient on-hand inventory to satisfy the remaining orders in the queue, then the on-hand stock are rationed according to Eq.

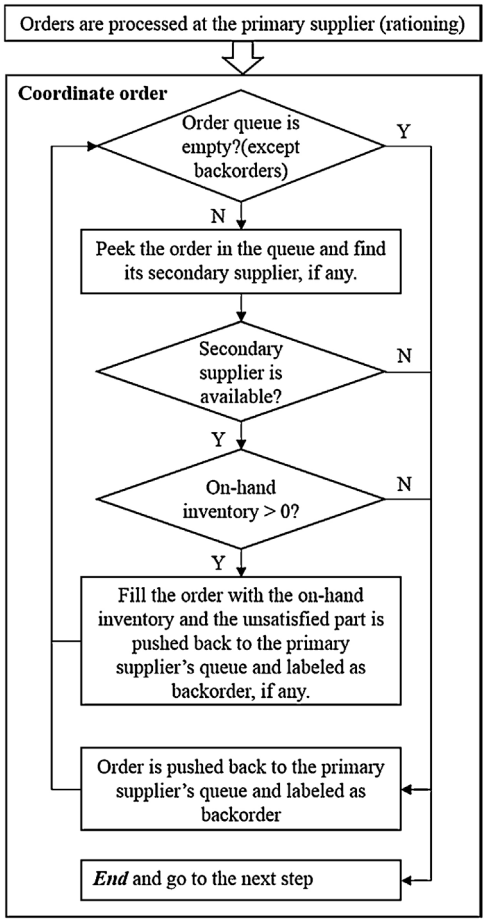

Fig. 6. Illustration for the order coordination procedure.

(3), the orders in queue are partially satisfied, and their remaining parts of the order will be filled by the secondary supplier, if possible. (A detailed procedure can be found in Fig. 6.)

$$IMFO_{ijt} = OH_i \times \frac{ROD_{ijt}}{SOD_{it}} \quad \forall j \in D_i \tag{3}$$

Nevertheless, in the worst case scenario, if unsatisfied parts of orders still remain, all of these unsatisfied orders will be backlogged at the primary supplier until there are sufficient on-hand stocks in the coming periods. The inventory record is updated again at the end of each planning period, and both the backorder amount BO_i and on-hand inventory levels OH_i are recorded for calculating the incurred operation cost.

The inventory performances are evaluated over the entire planning horizon after each simulation replication.

The average daily on-hand inventory of node i is

$$AOH_i = \frac{1}{N_d}\sum_{t=1}^{N_d} INVH_{it}, \quad \forall i \in I \tag{4}$$

The average daily backorder amount of facility i is

$$ABO_i = \frac{1}{N_d}\sum_{t=1}^{N_d} INVB_{it}, \quad \forall i \in I \tag{5}$$

The average daily operation cost of the node i is

$$AOC_i = c_i^h AOH_i + c_i^b ABO_i \tag{6}$$

Hence the average total daily cost of the entire supply chain network is

$$TLC = \sum_{i \in I} c_i^h AOH_i + \sum_{i \in I} c_i^b ABO_i = \sum_{i \in I} AOC_i \tag{7}$$

where c_i^h and c_i^b are the unit holding cost and unit backorder cost, respectively for node i.

The service level of a facility can be calculated by the records stored in the node. In this work the type II service level is used which is the expected proportion of demands satisfied by the on-hand inventory without delay (Chan, 2003).

The sum of order demands received by facility i during the entire time horizon is

$$TLOD_i = \sum_{t=1}^{N_d} SOD_{it} \quad \forall i \in I \tag{8}$$

and the total amount of orders satisfied immediately without delay is

$$TLSO_i = \sum_{t=1}^{N_d} FOD_{it} \quad \forall i \in I \tag{9}$$

Therefore, the service levels can be calculated:

$$SL_i = \frac{TLSO_i}{TLOD_i}, \quad \forall i \in I \tag{10}$$

Hence, we have the succinct function-form formulation for the simulation experiment

$$TLC = f(\mathbf{x}, \boldsymbol{\theta}) \tag{11}$$

$$SL_i = g_i(\mathbf{x}, \boldsymbol{\theta}), \quad \forall i \in I \tag{12}$$

In which the input variables are grouped as a vector of the base-stock level settings

$$\mathbf{x} = [x_1, \ldots, x_{N_{inv}}] = [BS_1, \ldots, BS_{N_{inv}}] \tag{13}$$

and the vector $\boldsymbol{\theta}$ represents a realization of all the uncertain parameters such as the fluctuating demands, stochastic preparation time, etc. involved over the entire planning horizon.

5.2. Formulation of the simulation-based optimization problem

Since our problem is coupled with several kinds of uncertainties, the result calculated from each simulation can only reflect the system's performance under a certain realization $\boldsymbol{\theta}$. We are more interested in the expected performance of the inventory system under all possible scenarios. However, there are infinitely many possible outcomes since most of the uncertainty parameters are continuous and have combinatory features. A straightforward approach is to use the Monte Carlo method to estimate the expected performance with a sufficiently large number of simulations for each set of base-stock levels. Each simulation is called a replication. Using the Monte Carlo method, the expected performances can be described by Eqs. (14) and (15).

$$\varphi(\mathbf{x}) = E_\theta[f(\mathbf{x}, \boldsymbol{\theta})] = \frac{1}{N_{rep}} \times \sum_{r=1}^{N_{rep}} f(\mathbf{x}, \boldsymbol{\theta}_r) \tag{14}$$

$$\phi_i(\mathbf{x}) = E_\theta[g_i(\mathbf{x}, \boldsymbol{\theta})] = \frac{1}{N_{rep}} \times \sum_{r=1}^{N_{rep}} g_i(\mathbf{x}, \boldsymbol{\theta}_r), \quad \forall i \in I \tag{15}$$

We use $\varphi(\mathbf{x})$ and $\phi_i(\mathbf{x})$ to denote the expected daily operation cost and the expected service level at node i. Since there is no explicit mathematical formulation for both $\varphi(\mathbf{x})$ and $\phi_i(\mathbf{x})$, they are black-box functions whose values must be evaluated by Monte Carlo simulation. Based on the problem statement in Section 4, the simulation-based inventory control optimization problem can be formulated as the following compact form in Eqs. (16) and (17) with the non-negativity constraint Eq. (18)

Sim-Opt Compact

$$\min_{\mathbf{x}} \varphi(\mathbf{x}) \tag{16}$$

s.t

$$\phi_i(\mathbf{x}) \geq sl_i^{\min}, \quad \forall i \in I \tag{17}$$

$$\mathbf{x} \geq 0 \tag{18}$$

in which $sl_i^{\min}$ is the service level lower bound for node i.

6. Surrogate-based optimization framework

In this section, we propose our computationally efficient algorithm for solving the simulation-based optimization problem for the inventory system. In Section 6.1, we introduce the kriging metamodeling method as the background for our surrogate-based approach to black-box systems. In Section 6.2, the region-wise surrogate modeling method is presented, and the simulation-based problem is rephrased by its reduced order formulation. The overall iterative procedures are detailed in Section 6.3. The computational framework is based on a trust-region framework for the management of approximation models and includes design of experiment (DOE) techniques, the kriging metamodeling method, and result validation with an iterative model correction process.

6.1. Surrogate-based optimization

Though a simulation-based approach can precisely reproduce the highly customizable inventory operations of real-world applications and incorporate impacts from different sources of uncertainties by Pseudo-Random Numbers (PRNs) generated by a computer, it also possesses drawbacks, such as a lack of a closed-form formulation between the input variables and output responses, longer computing times for conducting the simulations, unpredictable noise in the results, etc. All of these properties will amplify if more details are to be implemented during the simulation. In order to achieve a more satisfactory result from the "what if" realizations, the direct use of commercially available gradient-based solvers should be avoided. Due to the complicated interactions inside the simulator along with the uncertainties, its response to input variables can be viewed as a black-box function as described Eqs. (14) and (15).

In order to make the best use of off-the-shelf gradient-based solvers, we must find an explicit mathematical formulation to replace the original implicit representation. Linear regression is one of the most important techniques to fit a linear model from a few sampled data to make predictions.

We can present the output vector as a linear response to the input data $\mathbf{X}_s$ given a data set $\{\mathbf{y}|\mathbf{y} = h(\mathbf{X}_s)\}$, where the vector $\mathbf{y}$ is the output of unknown function $h(\mathbf{X})$ with sample input matrix $\mathbf{X}_s$. This is shown in as Eq. (19).

$$\mathbf{y} = \mathbf{X}_s\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{19}$$

where $\boldsymbol{\beta}$ is called the parameter vector and contains a set of linear regression coefficients, and $\boldsymbol{\varepsilon}$ is the vector for the error terms which quantify the deviation from the real response $h(\mathbf{X}_s)$. In linear regression, the least-squares estimator for the unknown function $h(\mathbf{X})$ is

$$\hat{h}(\mathbf{X}) = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{20}$$

where the conditional ordinary least-squares (OLS) estimator $\hat{\boldsymbol{\beta}}$ for $\mathbf{X}_s$ is calculated by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}_s^{\mathrm{T}}\mathbf{X}_s)^{-1}\mathbf{X}_s^{\mathrm{T}}\mathbf{y} \tag{21}$$

and we add a hat to the original function or vector to indicate that it is an estimation.

However, real-world cases are usually highly nonlinear; a more natural approach could be to fit the input and output data to nonlinear models. In this work we use the kriging model to estimate the nonlinear response of the black-box function. The kriging method is an efficient and flexible metamodeling technique based on spatial interpolation which was initiated by the South African engineer, Krige. It has since been improved and developed to build the input–output (I/O) relation for deterministic cases as well as stochastic cases (Ankenman et al., 2010; Kleijnen, 2009).

An expensive black-box function $h(\mathbf{X})$ can be fitted to the kriging estimator $\hat{h}(\mathbf{x})$ as

$$\hat{h}(\mathbf{x}) = R(\mathbf{X}) \cdot \hat{\boldsymbol{\beta}} + K(\mathbf{X}) \tag{22}$$

The first term is the classical regression model and it can be a constant model, a linear regression model, or even a quadratic model, which is used to approximate the trend of the response in question. The second term is the spatial correlation basis function and maps the vector $\mathbf{X}$ and corrects the predictions according to the design points. The second term $K(\mathbf{X})$ in our work has the form:

$$K(\mathbf{X}) = \exp\left[-\sum_{q=1}^{N_s}\sigma_q d_q^r\right] = \prod_{q=1}^{N_s}\exp[-\sigma_j d_q^r] \tag{23}$$

where $\mathbf{X}_{s,q}$ is in the design matrix $\mathbf{X}_s$ and d_q is the distance between the prediction point $\mathbf{X}$ to the qth design point:

$$d_q = \left|\mathbf{X} - \mathbf{X}_{s,q}\right| \tag{24}$$

Therefore, the prediction tends to be closer to the sampling response when it is in the neighborhood of a design point. Also, the smoothness of the function can be adjusted by the exponent r. In our work, we use the Gaussian correlation function in which r is equal to 2.

Hence, we can use the kriging estimator to model the simulation-based stochastic inventory optimization problem in **Sim-Opt Compact** using the explicit formulation shown below:

Sim-Opt Krig

$$\min_{\mathbf{x}} \hat{\varphi}(\mathbf{x}) \tag{25}$$

$$\text{s.t. } \hat{\phi}_i(\mathbf{x}) \geq sl_i^{\min}, \quad \forall i \in I \tag{26}$$

$$\mathbf{x} \geq 0 \tag{27}$$

where all the black-box functions are replaced with the corresponding estimators with a hat.

However, a closer examination into the surrogate-based formulation **Sim-Opt Krig** will raise a few questions and concerns. How should the design points be placed, both in terms of pattern and density? How should the solution returned by solving **Sim-Opt Krig** be returned? In addition, not only the objective function but also all the constraints in Eqs. (25) and (26) are the responses of all the components in vector $\mathbf{x}$. This will inevitably introduce an enormous amount of complexity to the model and drastically slow down the computation, especially since a certain degree of precision is required for a high dimensional problem. These concerns are addressed in the next subsection.

6.2. Region-wise surrogate-based optimization problem formulation

A typical method for striking a balance between the computational efficiency and the model's accuracy is to build a reduced order model (ROM) (Antoulas, 2005). Reduced order models can be rephrased as "rough" approximations to the precise models. In a dynamic system which comprises partial differential equations (PDEs), reduced order modeling can significantly reduce the computation load by principal component analysis or proper orthogonal decomposition (POD) (Agarwal and Biegler, 2013). If a "rough" model is used for optimization, then the optimization framework is a multi-fidelity optimization (Forrester et al., 2007; Kennedy and O'Hagan, 2000). However, the inventory simulation only provides black-box functions for estimating the performances, and there is no technique that can be immediately applied to the system.

In this section, we introduce region-wise surrogate modeling and the trust-region method based optimization framework for solving the black-box optimization problem for inventory management. The inventory network can usually be regarded as a cloud of nodes with complicated interactions between the nodes. Different kinds of service levels are usually evaluated for bridging the performances of two interconnected levels in the simulation-based models (Chu et al., 2015a; You and Grossmann, 2011b; You et al., 2011). The impact from the upstream node can also be quantified as a displacement to both the expected on-hand inventory and the service level, since a higher base-stock level kept in the predecessor will result in a higher service level if the remaining parameters are unchanged (Jung et al., 2008). This displacement in performance is nonlinearly correlated to the change in base-stock level at the upstream node and also depends on the variance of the uncertainty, the network structure, the inventory control policy, etc. However, the impact of an upstream inventory diminishes rapidly as it propagates through the chain of various downstream inventories as the setup of inventory itself is for hedging against these uncertainties. It is reasonable to assume that the performance of a certain facility is mostly affected by its immediate supplier nodes in a general network. Therefore, the full-space model can be disintegrated into reduced order surrogates by capturing the regional information of the nodes. Since the operation cost and the service level of each inventory node in the system is estimated from their regional surrogate model, we refer to this procedure as region-wise surrogate modeling. By constructing the surrogate models in the reduced dimension, the total number of the design points may not grow exponentially as the problem's dimension becomes large.

Therefore, we can use the regional information to formulate the reduced order surrogate models for the inventory control problem.

Rgn Sgt Opt

$$\min_{\mathbf{x}} \sum \hat{\varphi}_i(\mathbf{x}_i) \tag{28}$$

$$\text{s.t. } \hat{\phi}_i(\mathbf{x}_i) \geq sl_i^{\min}, \quad \forall i \in I \tag{29}$$

$$\mathbf{x}_i \geq 0, \quad \forall i \in I \tag{30}$$

This formulation is very similar to the full space surrogate model **Sim-Opt Krig;** however, there are two differences. First, the input vector for the estimated functions are now replaced by the vector $[\mathbf{x}_i]$, which only contains the input base-stock levels of the most direct supplier nodes and the base-stock level of the interested node. If we use the example network in Fig. 2 for illustration, then

$$\mathbf{x}_2 = [x_2, x_1] \tag{31}$$

Namely, the inventory cost and the service level performance of node 2 are principally affected by the base-stock levels at node 2 and node 1. The same holds for node 4 and node 3.

$$\mathbf{x}_4 = [x_4, x_2] \tag{32}$$

$$\mathbf{x}_3 = [x_3, x_2, x_4] \tag{33}$$

since both node 2 and node 4 are node 3's direct suppliers, all of these 3 base-stock levels are of greater importance to node 3. But the input vector for node 1 only has one component.

$$\mathbf{x}_1 = [x_1] \tag{34}$$

Node 1 is the manufacturing plant, it does not have any upstream suppliers so the vector only contains a single component.

The second difference compared to the full space surrogate is that in the reduced order model, both the estimation of the operation cost and the service levels are constructed in the neighborhood of a fixed point **x**. This is done by perturbing the components in the vector $\mathbf{x}_i$ in order to reduce the model from full space. Although this reduction in the model is less precise than the full space, it results in a computationally efficient model

In addition, it is unnecessary to introduce additional constraints to stipulate the network structure in the region-wise surrogate formulation since they are implicitly included when the components for each input vector $\mathbf{x}_i$ are decided. For example, $\mathbf{x}_3$ and $\mathbf{x}_4$ share the component x_2 but $\mathbf{x}_3$ includes the component x_4. Therefore we can infer the triangular structure among node 2, node 3 and node 4 in Fig. 2 where both node 2 and node 4 are suppliers for node 3.

Thus, the region-wise surrogate formulation **Rgn Sgt Opt** decomposes the entire supply chain into low-dimension (up to 3 in this work) surrogate models which are implicitly linked together by the input vectors. In order to capture the nonlinear response to the input variables, we construct a kriging model with Gaussian correlations to replace the costly black-box simulation experiments. The regional surrogate models are built after the design of experiment (DOE) techniques are employed. We limit the perturbation to the key input variables in a constrained space and use the Latin hypercube sampling method to estimate the kriging parameters. The explicit formulation of the kriging estimator can greatly facilitate the resulting optimization process. That is, we can solve the region-wise surrogate formulation with a gradient-based NLP solver.

It is worth mentioning that the proposed region-wise surrogate modeling strategy is finalized after a thorough comparison with less parameterized models, for instance, the linear regression model and localized surrogates. Both the nonlinear kriging model and the incorporation of region-wise rather than localized information (one-dimension model) enable the surrogates to capture sufficient nonlinearity along with the complicated interactions between different nodes and avoid the optimization process to terminate too early. Nevertheless, the accuracy of the reduced space surrogate models can hardly guarantee an exact prediction since we only take major correlations into account and the concomitant issues will be

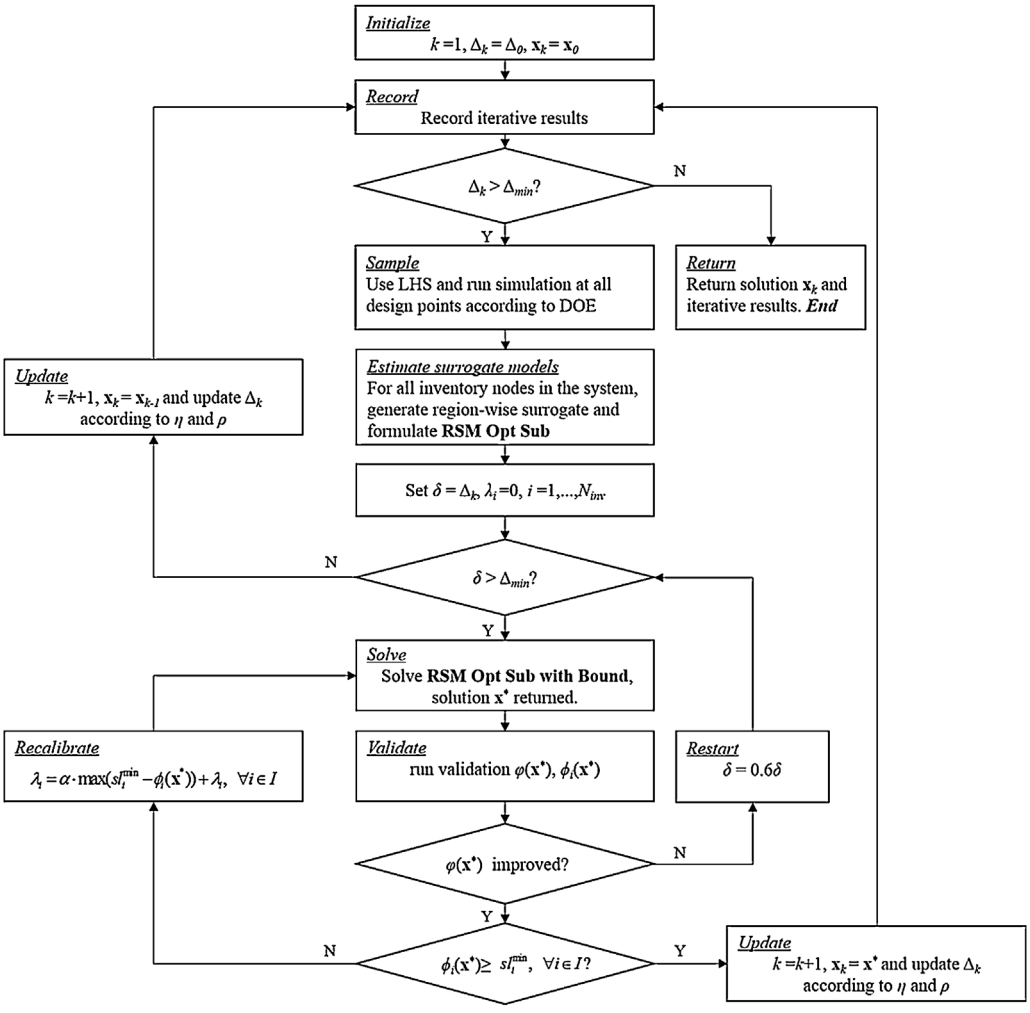

Fig. 7. Iterative procedure of the trust-region framework with model recalibration. α is a positive coefficient for offsetting the constraint violation during the recalibration.

addressed in Section 6.3. Therefore, the trust-region method can work as proposed.

6.3. Trust-region method with iterative model recalibration

In order to converge to a valid and improved solution, we proposed a trust-region framework with model recalibration procedure to manage the use of the regional surrogate models. Since the framework is highly heuristic and the use of the trust-region framework has already greatly enhanced the convergence property in practice, we do not add sophisticated model calibration techniques. Some of these model calibration techniques are the fraction of Cauchy decrease (FCD) and the consistency constraints in the theory of Approximation Management Framework (AMF) (Alexandrov et al., 1998; Alexandrov and Lewis, 2003; Gunnerud et al., 2013).

The iterative procedure of the trust-region framework is shown in Fig. 7. First, similar to the conventional trust-region method, the algorithm is initialized with an initial trust-region size Δ_0 and the iteration number is set to 1. The algorithm starts from a feasible solution $\mathbf{x}_0$ which keeps a high service level at each node. Since a higher base-stock level will result in a high service level, we can find the initial feasible solution without difficulty.

During the kth iteration with the current iterate $\mathbf{x}_k$, the region-wise kriging-based surrogate models $\hat{\varphi}_i(\mathbf{x}_i)$ and $\hat{\phi}_i(\mathbf{x}_i)$ for each node i are estimated with the previously mentioned method within the subspaces constrained by the trust-region Δ_k. Here, the trust-region is defined as the first-order norm which favors the LHS to place the design points.

$$\left|\mathbf{x}-\mathbf{x}_k\right| \le \Delta_k \tag{35}$$

A constrained trust-region problem based on the aggregated regional surrogate models is to be solved to get an improved result.

RSM Opt Sub

$$\min_{\mathbf{x}} \sum \hat{\varphi}_i(\mathbf{x}_i) \tag{36}$$

$$\text{s.t. } \hat{\phi}_i(\mathbf{x}_i) \ge sl_i^{\min}, \quad \forall i \in I \tag{37}$$

$$\mathbf{x}_i \ge 0, \quad \forall i \in I \tag{38}$$

$$\left|\mathbf{x}-\mathbf{x}_k\right| \le \Delta_x \tag{39}$$

However, the solution returned by solving **RSM Opt Sub** should be validated before claiming it is a successful iteration. Since the surrogate models are built by neglecting the impacts from more distant nodes, it is highly possible that the solution does not achieve any improvement or even become infeasible when we validate it by running the simulation at the returned solution $\mathbf{x}^*$. The situation may become even worse when considering the uncertain nature of the stochastic simulation.

In the case when the validation run violates one or more of the constraints, a recalibration process is introduced below to improve the quality of the solution (Fig. 7). In order to offset the deviation caused by the use of the low-fidelity region-wise surrogate models, we first need to identify the sources of such discrepancies. Specifically for the stochastic inventory optimization problem under general network assumptions, there are a few reasons to make the prediction imprecise. The first and most straightforward reason is the regional surrogate model neglects the impact from the distant nodes, which is also the intrinsic reason for causing bias in the prediction. Second, while the base-stock vector changes, the network will redistribute the flows of the orders. A customer inventory node in the network may redirect its orders from the primary supplier to the secondary supplier or vice versa. Also, since we use a limited number of design points to build the surrogate models, modeling error is inevitable. Furthermore, even if the solver takes a reasonable step without compromising the surrogate's validity, the uncertainty underlying the stochastic simulation will still contain some noise in the response. This phenomenon always coexists with the optimization process as we can only use finitely many replications to conduct the Monte Carlo simulation.

To overcome the above difficulties, we use an iterative recalibration process to correct the error. We first combine all the concerns into a single error term ε_i.

$$\hat{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}) + \varepsilon_i, \quad \forall i \in I \tag{40}$$

When estimating the service level constraints, the surrogate model can either overestimate or underestimate the simulation response, so the term ε_i can be either positive or negative. However, we only need to avoid the case in which the surrogate model $\hat{\phi}(\mathbf{x}_i)$ is the over-estimator of the true response $\phi(\mathbf{x})$ since the validation run will not pass the feasibility test if the prediction is overly optimistic.

We can then further assume the absolute value of the error term is bounded by a finite positive value λ_i within the trust-region Δ_k.

$$\left|\varepsilon_i\right| \le \lambda_i, \quad \forall i \in I \tag{41}$$

if

$$\left|\mathbf{x}-\mathbf{x}_k\right| \le \Delta_k \tag{42}$$

Combining with Eq. (40), the surrogate model is bounded aside the simulation response.

$$\phi(\mathbf{x}) + \lambda_i \ge \hat{\phi}(\mathbf{x}_i) \ge \phi(\mathbf{x}) - \lambda_i, \quad \forall i \in I \tag{43}$$

Thus, in order to eliminate the overestimation, we need to find the value of the bound λ_i, then plug it into the right hand side of the constraint Eq. (37) to find a feasible solution, and the trust-region subproblem is changed to solve

RSM Opt Sub with Bound

$$\min_{\mathbf{x}} \sum \hat{\varphi}_i(\mathbf{x}_i) \tag{44}$$

$$\text{s.t } \hat{\phi}_i(\mathbf{x}_i) \ge sl_i^{\min} + \lambda_i, \quad \forall i \in I \tag{45}$$

$$\mathbf{x}_i \ge 0, \quad \forall i \in I \tag{46}$$

$$\left|\mathbf{x}-\mathbf{x}_k\right| \le \Delta_k \tag{47}$$

In practice, the bound λ_i is updated iteratively by adding a positive multiplier of the constraint violation, $\max(sl_i^{\min} - \phi_i(\mathbf{x}^*))$ during the validation process in Fig. 7. If a feasible and improved solution is found, then the algorithm claims a successful step and proceeds to the next iteration with an updated trust-region size. Otherwise, the bound will keep increasing and each update confines the problem **RSM Opt Sub with Bound** into a smaller feasible region until either the objective value exceeds the previous iteration or a successful iteration is claimed.

On the other hand, it is interesting to note that the major components of the error term also evolve when the trust-region size changes as in Fig. 8. At the initial stage of the optimization, the algorithm builds relatively rough surrogate models with a large trust-region size which will result in some imprecise but ambitious improving steps. However, when the algorithm approaches the final solution, the trust-region will shrink, but the surrogate model provides more accurate predictions. When the trust-region diminishes to the minimum size, the error term will almost exactly contain the simulation noise but with negligible bias in the prediction. Since a successful improving step is only claimed when the validation run satisfies both the service level constraints and minimum descent criteria, the recalibration process will automatically establish a certain amount of "safety distance" to reduce the probability for the service levels to fall below their tolerances.

The change of the trust-region size follows the traditional trust-region management rules. Therefore, the algorithm terminates

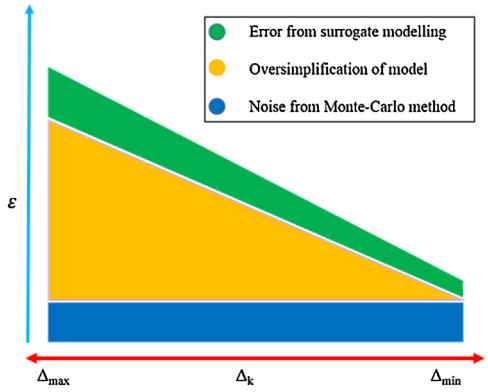

Fig. 8. Illustration for the evolution of the error term.

while the trust-region size goes to the minimum. Two indicators η and ρ, representing the step size and the similarity respectively, are used to determine the trust-region size for the next step. We also add two features to increase the computational efficiency. First, the number of simulation replications is dynamic with regard to the trust-region size. The replication number is cut to 1/10 of the maximum amount when building the "rough" surrogate models while, it gradually increases as the trust-region shrinks. Another modification is that the procedure for solving the **RSM Opt Sub with Bound** can be restarted several times in a diminishing range δ per iteration, this would avoid being trapped in some infeasible solutions and it may save time by not having to start over the metamodeling process. Thus, the constraint Eq. (47) in **RSM Opt Sub with Bound** is replaced with

$$\left|\mathbf{x} - \mathbf{x}_k\right| \leq \delta_k \tag{48}$$

7. Case study

We apply the presented region-wise surrogate modeling and optimization framework to two case study problems for inventory management optimization. The Monte Carlo simulation for the inventory system is programed in Java using Netbeans. The network structures, operation parameters and initial values for the decision variables are loaded from an Excel document. The Java package can be called by MATLAB and we use the DACE toolbox for estimating the nonlinear surrogate functions for input–output relations (Myers, 2002; Santner et al., 2003). The trust-region method with recalibration is implemented with MATLAB and the constrained nonlinear minimization subproblems are solved by using the FMINCON function embedded in MATLAB. All problems are coded with a desktop with a 3.00 GHz CPU, 4.00 GB RAM with Windows 7 Professional (64 bit).

7.1. General inventory network with 8 stocking nodes

Fig. 9 is the network structure for the first case study. The supply chain network consists of one manufacturing plant (Facility 1 as indexed in Fig. 9), three warehouses for distribution purposes (Facility 2, 3, 6) and four regional customer nodes (Facility 4, 5, 7, 8). Each facility in the network contains an inventory to store the product. The type of the arrows which link two different facilities indicates the priority when sourcing for each node, and each node may have up to two potential suppliers. In contrast to a classical multi-echelon supply chain structure, the possibility for multiple-sourcing allows the network to be general since there is no multi-echelon structure.

Each inventory stocking node operates according to the base-stock policy. For a specific node in the supply chain, orders are

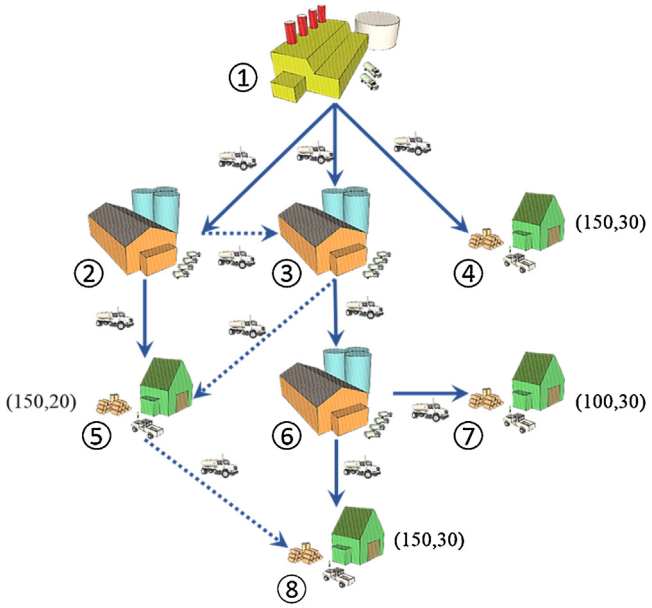

Fig. 9. The structure of the 8-node inventory system. The demands occur at the sales regions every day and follow some normal distributions. For clarity, all inventory nodes are numbered. The probability distribution parameters are labeled next to the sales region nodes in the form of *norm*(mean, standard deviation).

generated once the simulation clock moves through a certain length of time. The demands of the customer regions occur every day, and they are simulated as some normally distributed random variables with known means and standard deviations given in Fig. 9. The unit holding cost and the unit backordering cost of the inventories are both 1 m.u./unit/day (m.u. is referred as monetary unit). The required lower bounds for the service levels are 70%, 70%, and 95% for the inventories at the manufacturing plant, warehouses, and customer regions, respectively. Since enterprises usually prefer to satisfy the external customer demands at a higher level compared with their internal demands, lower tolerance levels are applied at plants and warehouses. The detailed operation parameters can be found in Table 1.

In order to obtain a reasonable estimation, the inventory system simulation is conducted with a planning horizon of 200 days, and a 100 day long warm-up simulation is also implemented beforehand to initialize the inventory status. Hence, the total planning horizon length in each simulation replication is 300 days, and we use the average results for up to 100 replications to estimate the expected value of interest, including the expected total cost and expected service level achieved at each facility node.

The proposed computational framework in Section 6.2 is applied to solve the inventory control optimization problem for the general supply chain network, and the procedure is illustrated in Fig. 7. The number of simulation replications at the beginning of the algorithm is set to 10 since a relatively rough set of sampling points during the initial stage is sufficient to estimate the surrogate models. As the radius of the trust-region shrinks iteration by iteration, we place up to 100 simulation replications on each design point, which is large enough to guarantee an accurate estimation. **RSM Opt Sub with Bound** is solved by the FMINCON function in MATLAB. The trust-region algorithm proceeds to the next iteration if either a minimum trust-region size is not reached or the solution to the trust-region subproblem is more than 0.05% larger than that of the previous step.

The region-wise surrogate modeling and optimization framework stops after 10 iterations, and the total computation time for solving the 8-dimension problem is 353.5 s. We also count the total

Table 1
The parameter settings and service level constraints for the 8-node inventory system.

Node	1	2	3	4	5	6	7	8
Reorder period (day)	1	1	2	1	1	1	1	1
Order preparation time (day)	(2, 4)	(1, 2)	1	1	1	1	1	1
Service level lower bound	70%	70%	70%	95%	95%	70%	95%	95%

The order preparation time is a random number with uniform distribution (discrete) *unif* (minimum value, maximum value) and rounded to integer. Otherwise, single values indicate they are deterministic.

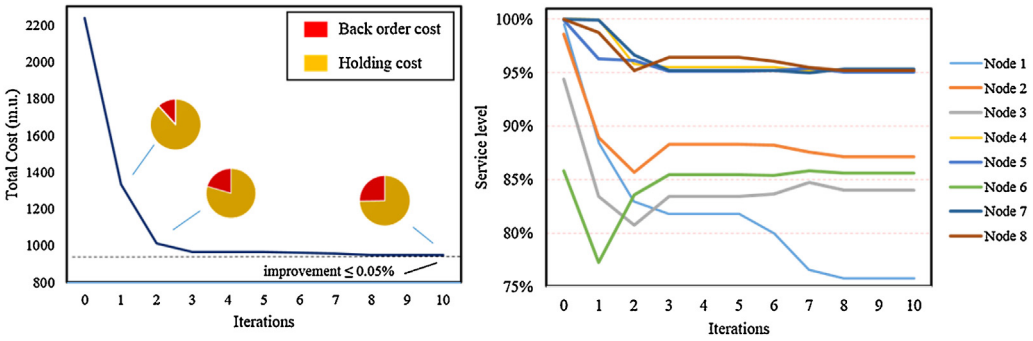

Fig. 10. Iterative results of the 8-node inventory system.

Table 2
Initial and optimal solutions with service levels at optimal solution for the 8-node inventory system.

Node	1	2	3	4	5	6	7	8
Initial solution	3000.0	600.0	800.0	800.0	600.0	400.0	500.0	500.0
Optimal solution	2169.0	519.5	823.1	518.8	452.2	456.0	282.3	374.7
Service levels at optimum	75.76%	87.12%	84.02%	95.18%	95.04%	85.59%	95.34%	95.20%

number of sampling points during the entire computation process, which is 1833 and includes simulation efforts invested on building the kriging models, model recalibration, and results validation. The iterative procedure is shown in Fig. 10, and both the initial and optimal solutions are presented in Table 2.

It is worth noting that the service levels realized at the optimal solution are all above the required lower bounds. For sales region nodes which immediately serve the external customers, the service levels are pushed to the lowest tolerant level (95%), and thanks to the iterative recalibration procedure mentioned in Section 6.2, a certain amount of "safety distances" are left to prevent the service levels from violating the constraints by chance (Table 2).

In order to evaluate the quality of the optimal base-stock levels returned by our proposed optimization, we further compare it with the solution obtained by using a genetic algorithm (GA). The latter is frequently regarded as a general and efficient solution method to black-box problems given sufficiently many computation budgets. To solve such a problem with both a black-box objective function and implicit constraints, the constrained minimization problem is converted to an unconstrained one in Eq. (49) by adding a penalty for constraint violations.

Sim-Opt GA

$$\min G(\mathbf{x}) = \varphi(\mathbf{x}) + \sum_{i \in I} \upsilon_i \cdot \max(sl_i^{\min} - \phi_{sl,i}(\mathbf{x}), 0) \tag{49}$$

The GA method is conducted in MATLAB by calling the MATLAB Genetic Algorithm Toolbox. We use the "10k" rule to set the GA's population to 80 for this 8-dimension problem, and the penalty coefficient υ_i is set to 1,000,000, which is large enough to guarantee the service levels to be above the imposed lower bounds. Starting near the same initial solution, the GA returns a solution with an objective value of 951.1 m.u./day in 2104.2 s with 14,562 function calls to the simulation. It is very close to the result of 948.2 m.u./day (within 0.3%) given by the region-wise surrogate

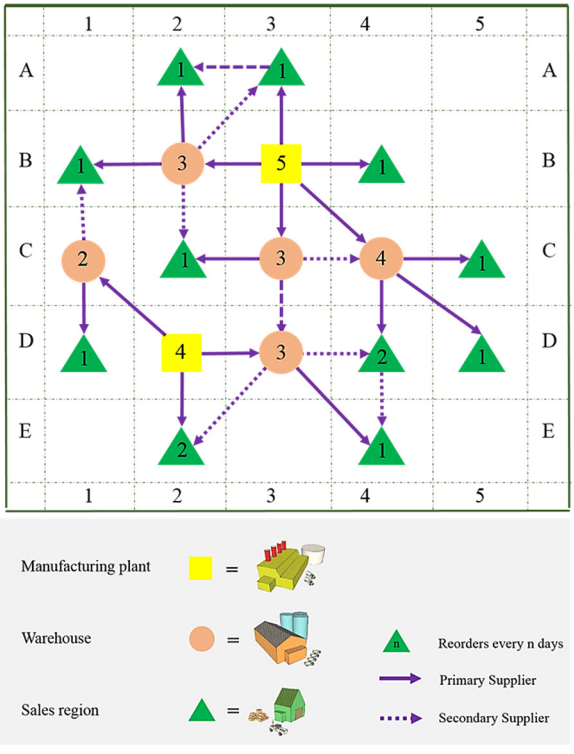

Fig. 11. Illustration for the 18-node inventory network. For simplicity, we use the shorthand icons to indicate different facility nodes in the network. The network is presented in a grid, and we use the coordinates to label the nodes. The numbers on the icons indicate the length of review period for each inventory. Details are in the legend located below the grid map.

Table 3
The comparation between the solutions returned by our proposed algorithm and MATLAB GA toolbox.

Method	RSM							
Node	1	2	3	4	5	6	7	8
Initial solution	3000	600	800	800	600	400	500	500
Optimal solution	2169	520	823	519	452	456	282	375
Service levels at optimum	75.8%	87.1%	84.0%	95.2%	95.0%	85.6%	95.3%	95.2%
Optimal value	948.2							
Method	GA							
Node	1	2	3	4	5	6	7	8
Initial solution	3000	600	800	800	600	400	500	500
Optimal solution	2205	516	920	515	402	476	266	356
Service levels at optimum	77.9%	89.5%	91.9%	95.4%	95.7%	90.5%	95.1%	95.2%
Optimal value	951.1							

RSM: region-wise surrogate modeling method.

Table 4
The parameter settings and service level constraints for the 18-node inventory system.

Node	A2	A3	B1	B2	B3	B4
Reorder period (day)	1	1	1	3	5	1
Demand	(150, 30)	(200, 20)	(100, 30)	N/A	N/A	(150, 30)
Preparation time (day)	1	1	1	(1, 2)	(1, 3)	1
Service level lower bound	95%	95%	95%	70%	70%	70%
Node	C1	C2	C3	C4	C5	D1
Reorder period (day)	2	1	3	4	1	1
Demand	N/A	(150, 30)	N/A	N/A	(150, 20)	(200, 30)
Preparation time (day)	1	1	(1, 2)	(1, 2)	(1, 2)	1
Service level lower bound	70%	95%	70%	70%	95%	95%
Node	D2	D3	D4	D5	E2	E4
Reorder period (day)	4	3	2	1	2	1
Demand	N/A	N/A	(150, 30)	(150, 20)	(100, 30)	(150, 30)
Preparation time (day)	(2, 3)	(1, 2)	1	1	1	1
Service level lower bound	70%	70%	95%	95%	95%	95%

The order preparation time is a random number with uniform distribution (discrete) *unif* (minimum value, maximum value) and rounded to integer. Otherwise, single values indicate they are deterministic.

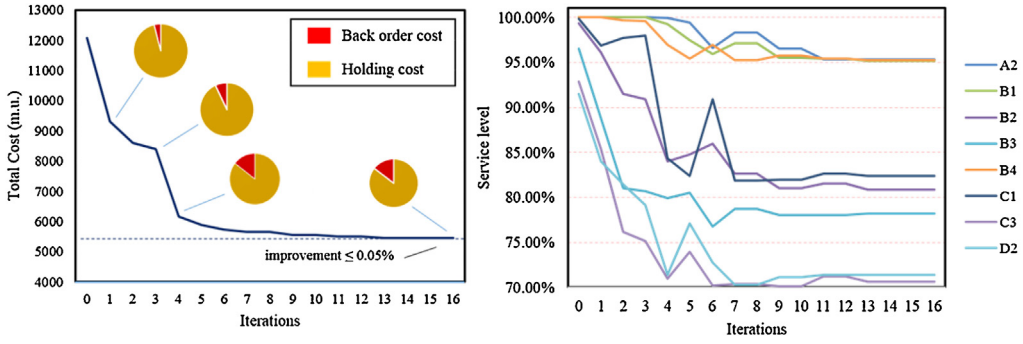

Fig. 12. Iterative results for the 18-node inventory system.

modeling framework (Table 3). A few variations may occur in the detailed base-stock level settings to each node after a close examination of the solutions returned by these two methods. This is probably due to the non-convexity of the problem, and there might be a considerable number of local optimal solutions scattered across the response surface. Theoretically, either solution will work because they have very similar expected total costs. However in practice, additional constraints may help the decision maker to break the tie. For instance, we can introduce capacity constraints to each inventory or take the geographical distance between nodes into consideration which is beyond the scope of our study.

We may expect a more economical solution if larger computation budgets are allowed to be spent on GA (increased simulation replications, harsher stopping criteria, etc.). However, we limit the resources only to demonstrate that our proposed algorithm can return a solution that is better than that returned by a GA with a lower computation cost.

7.2. General inventory network with 18 stocking nodes

In the second case study, we apply the proposed algorithm to a larger supply chain network which contains 18 inventory stocking nodes. For simplicity, the network is presented by Fig. 11 where

Table 5
Initial and optimal solutions with service levels at optimal solution for the 18-node inventory system.

Node	A2	A3	B1	B2	B3	B4
Initial solution	1000	1500	600	1200	10,000	1000
Optimal solution	477.1	1171.0	361.4	821.0	8032.0	710.0
Service levels at optimum	95.37%	95.30%	95.19%	80.83%	78.22%	95.23%
Node	C1	C2	C3	C4	C5	D1
Initial solution	1500	1000	1300	3000	800	800
Optimal solution	1210.3	525.0	1027.4	2744.5	527.6	605.3
Service levels at optimum	82.39%	95.32%	70.64%	82.03%	95.33%	95.28%
Node	D2	D3	D4	D5	E2	E4
Initial solution	3000	1000	1000	1000	800	800
Optimal solution	2246.6	563.3	899.8	530.4	566.0	466.6
Service levels at optimum	71.43%	81.29%	95.48%	95.60%	95.16%	95.34%

Table 6
The result returned by the MATLB GA toolbox after 29,279 s.

Node	A2	A3	B1	B2	B3	B4
GA solution	630.6	1343.1	506.4	345.6	8570.6	672.9
Service levels at optimum	99.83%	96.58%	99.20%	72.74%	84.24%	97.15%
Node	C1	C2	C3	C4	C5	D1
GA solution	1214.1	599.6	858.9	2656.0	651.8	859.3
Service levels at optimum	74.35%	97.92%	71.11%	80.26%	99.18%	99.20%
Node	D2	D3	D4	D5	E2	E4
GA solution	2374.3	622.5	873.5	568.1	581.8	571.5
Service levels at optimum	90.93%	81.12%	95.46%	96.44%	96.52%	99.73%

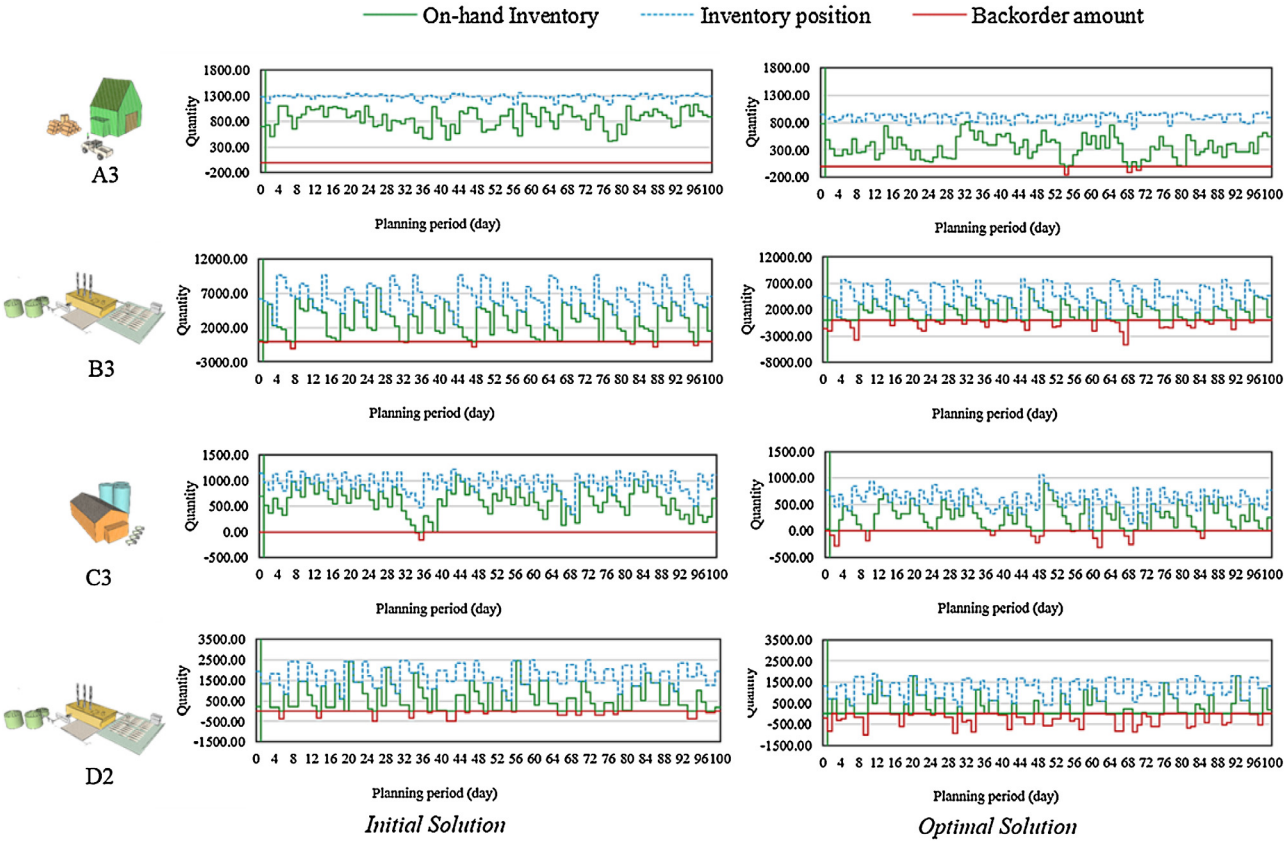

Fig. 13. Inventory records during the first 100 days in the planning horizon at selected nodes. The left column shows the results at the initial solution and the right column shows the results at the optimal solution. The backorders exist when the red line go below zero. For a better comparison, diagrams are in the same scale for each node.

the facility nodes are displayed by symbols in the grid. The network contains 2 sources (manufacturing plants located in B3 and D2) and 11 sinks (the sales regions scattered among the network). Both of the plants can fill the orders from their customers without delay and the production times follow the uniform distribution (discrete) *unif*(2, 4) and *unif*(1, 4) for B3 and D2, respectively. Coordination is also possible in the supply chain since each inventory is allowed to have up to 2 suppliers. Overall, there are 18 base-stock levels to be determined in order to minimize the total operation cost while satisfying the service level constraints at different nodes. The detailed simulation parameters are listed in Table 4.The demand at each sales region node is updated every day and follows a normal distribution with known mean and standard deviation. Each replication contains a planning horizon of 200 days after a 100 day warm-up period. The unit holding cost and the unit backordering cost of the inventories are both 1 m.u./unit/day. Again, we impose relatively higher service level lower bound constraints to the nodes that immediately serve the external customers than to the internal inventory nodes.

The proposed optimization framework returns the optimal solution of a total operation cost of 5462.3 m.u./day after 16 iterations, and the computation time for solving the 18-dimension problem is 2499.1 s. The total number of sampling points during the entire computation process for metamodeling and validation is 8057. For simplicity, the iterative procedure for some selected nodes is shown in Fig. 12, and both the initial and optimal solutions are shown in Table 5. The service levels of all the sales region nodes are pushed to their lower bound of 95%. Interestingly, only the warehouses can reach their lowest tolerable service level of 70%, while the service levels of the intermediate inventory stocking nodes are not pushed to their lowest possible value. This is mainly due to the tradeoff between a lower total operation cost and a higher satisfactory level for the direct customer nodes.

The dynamic trajectory of the inventory records according to the initial solution and the optimized solution are shown in Fig. 13. After being optimized by the proposed algorithm, the inventories in the system allow a moderately higher but reasonable probability for backordering; thus, less carrying cost will be incurred to hold the on-hand physical stocks (Table 6).

We also use the GA toolbox to solve the high-dimension black-box problem. According to the '10k' rule, the population is set to 180, however, the search only returns a feasible but not optimal solution after 29,279 s with a total number of 24,222 calls to the sampling points. The solution of 6385.3 m.u./day is 16.9% higher than that of the region-wise surrogate modeling method. The solution is barely satisfactory in fact since most service levels are still maintained at high values which will result in some unnecessary operation cost. We expect the GA to return a solution of higher quality if we increase the number of replications and impose more strict stopping criteria, however, the computation time will be extended enormously and optimality still cannot be guaranteed.

8. Conclusion

A simulation-based optimization for optimizing inventory control of general inventory systems was proposed. We introduced the multi-sourcing capability into the inventory network where each inventory can have up to 2 supplier nodes; thus, the network can have more desirable features such as order coordination and multiple manufacturing sites. The inventory simulation was programmed by the objective oriented programing method, and we used the Monte Carlo method to estimate both the expectations of the objective function and the constraints. By disintegrating the entire network into sub-regions, the original problem's dimension was largely reduced after kriging surrogate models were used to approximate the regional input–output correlations. The aggregated region-wise surrogates provided explicitly formulated reduced order predictors for the costly Monte Carlo simulation estimator of the objective and constraints. The converted NLP was optimized in a trust-region framework, and the infeasible solutions during the iterative procedure were corrected by the zero-order model, which is a positive multiple of the constraint deviation. When the trust-region shrank to a minimum, the optimal solution was obtained.

The proposed region-wise surrogate modeling and optimization framework was demonstrated by two case studies. The first case study was a general inventory system with 8 nodes and the second one contained as many as 18 inventories. The trust-region framework with model recalibration iteratively solved both cases in 353.5 s and 2499.1 s, respectively, and returned solutions with reduced total operation cost with a tolerable chance to sustain stockouts. The advantages of the proposed solution method relative to using a GA method allowed for the identification of better solutions in a shorter time frame.

Moreover, our proposed computational framework could be improved in a number of directions. First, a sophisticated transportation cost model can be added to better aid the decision-making process in a real-world case. In this case, both the distances between different nodes and the unit price per unit distance should be defined.

Additionally, the reuse of sampling data in the trust-region algorithm might be another promising way to further increase the overall performance. Though the region-wise surrogate models are only valid within one iteration in our proposed computational framework, there is still a chance to reduce the computational burden by reusing sampled data if the previous iteration achieves only insignificant improvement.

References

Agarwal A, Biegler LT. A trust-region framework for constrained optimization using reduced order modeling. Optim Eng 2013;14:3–35.

Alexandrov NM, Dennis JE, Lewis RM, Torczon V. A trust-region framework for managing the use of approximation models in optimization. Struct Optim 1998;15:16–23.

Alexandrov NM, Lewis RM. First-order approximation and model management in optimization. Large-Scale PDE-Constrain Optim 2003;30:63–79.

Ankenman B, Nelson BL, Staum J. Stochastic kriging for simulation metamodeling. Oper Res 2010;58:371–82.

Antoulas AC. An overview of approximation methods for large-scale dynamical systems. Annu Rev Control 2005;29:181–90.

Chan FTS. Performance measurement in a supply chain. Int J Adv Manuf Technol 2003;21:534–48.

Chen Y, Mockus L, Orcun S, Reklaitis GV. Simulation-optimization approach to clinical trial supply chain management with demand scenario forecast. Comp Chem Eng 2012;40:82–96.

Cheng L, Subrahmanian E, Westerberg AW. Design and planning under uncertainty: issues on problem formulation and solution. Comp Chem Eng 2003;27:781–801.

Chopra S. Supply chain management: strategy, planning, and operation. 4th ed. Boston: Prentice Hall; 2010.

Chu YF, You F, Wassick JM, Agarwal A. Integrated planning and scheduling under production uncertainties: bi-level model formulation and hybrid solution method. Comp Chem Eng 2015a;72:255–72.

Chu Y, You F, Wassick JM, Agarwal A. Simulation-based optimization framework for multi-echelon inventory systems under uncertainty. Comp Chem Eng 2015b;73:1–16.

Conn AR, Scheinberg K, Vicente LN. Introduction to derivative-free optimization. Philadelphia: Society for Industrial and Applied Mathematics/Mathematical Programming Society; 2009.

Cozad A, Sahinidis NV, Miller DC. Learning surrogate models for simulation-based optimization. AIChE J 2014;60:2211–27.

De Boer PT, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. Ann Oper Res 2005;134:19–67.

Ettl M, Feigin GE, Lin GY, Yao DD. A supply network model with base-stock control and service requirements. Oper Res 2000;48:216–32.

Forrester AIJ, Sobester A, Keane AJ. Multi-fidelity optimization via surrogate modelling. Proc R Soc A: Math Phys Eng Sci 2007;463:3251–69.

Fu MC. Optimization for simulation: theory vs. practice. Informs J Comp 2002;14:192–215.

Garcia DJ, You F. Supply chain design and optimization: challenges and opportunities. Comp Chem Eng 2015;81:153–70.
Graves SC, Willems SP, Zipkin P. Optimizing strategic safety stock placement in supply chains. Manuf Serv Oper Manage 2000;2:16.
Grossmann I. Enterprise-wide optimization: a new frontier in process systems engineering. AIChE J 2005;51:1846–57.
Gunnerud V, Conn A, Foss B. Embedding structural information in simulation-based optimization. Comp Chem Eng 2013;53:35–43.
Inderfurth K, Minner S. Safety stocks in multi-stage inventory systems under different service measures. Eur J Oper Res 1998;106:57–73.
Jung JY, Blau G, Pekny JF, Reklaitis G, Eversdyk D. Integrated safety stock management for multi-stage supply chains under production capacity constraints. Comp Chem Eng 2008;32:2570–81.
Jung JY, Blau G, Pekny JF, Reklaitis GV, Eversdyk D. A simulation based optimization approach to supply chain management under demand uncertainty. Comp Chem Eng 2004;28:2087–106.
Kennedy MC, O'Hagan A. Predicting the output from a complex computer code when fast approximations are available. Biometrika 2000;87:1–13.
Kleijnen JPC. Kriging metamodeling in simulation: a review. Eur J Oper Res 2009;192:707–16.
Kochel P, Nielander U. Simulation-based optimisation of multi-echelon inventory systems. Int J Prod Econ 2005;93-4:505–13.
Law AM, Kelton WD. Simulation modeling and analysis. 3rd ed. Boston: McGraw-Hill; 2000.
Lee HL, Billington C. Material management in decentralized supply chains. Oper Res 1993;41:835–47.
Lee JH, Kimz CO. Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. Int J Prod Res 2008;46:233–65.
Mansouri SA. A simulated annealing approach to a bi-criteria sequencing problem in a two-stage supply chain. Comp Chem Eng 2006;50:105–19.
Mele FD, Guillen G, Espuna A, Puigjaner L. A simulation-based optimization framework for parameter optimization of supply-chain networks. Ind Eng Chem Res 2006;45:3133–48.
Michalski G. Inventory management optimization as part of operational risk management. Econ Comp Econ Cybern Stud Res 2009;43:213–22.
Myers DE. Interpolation of spatial data: some theory for kriging. Int J Geogr Inf Sci 2002;16:205–7.
Porteus EL. Foundations of stochastic inventory theory. Stanford, CA: Stanford Business Books, an imprint of Stanford University Press; 2002.
Relvas S, Matos HA, Barbosa-póvoa APFD, Fialho J, Pinheiro AS. Pipeline scheduling and inventory management of a multiproduct distribution oil system. Ind Eng Chem Res 2006;45:7841–55.
Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. J Glob Optim 2013;56:1247–93.
Sahay N, Ierapetritou M. Hybrid simulation based optimization framework for centralized and decentralized supply chains. Ind Eng Chem Res 2014;53:3996–4007.
Santner TJ, Williams BJ, Notz W. The design and analysis of computer experiments. New York: Springer; 2003.
Shah N. Process industry supply chains: advances and challenges. Comp Chem Eng 2005;29:1225–35.
Simchi-Levi D. Designing and managing the supply chain: concepts, strategies, and case studies. 3rd ed. Boston: McGraw-Hill/Irwin; 2008.
Subramanian D, Pekny JF, Reklaitis GV. A simulation-optimization framework for addressing combinatorial and stochastic aspects of an R&D pipeline management problem. Comp Chem Eng 2000;24:1005–11.
Subramanian D, Pekny JF, Reklaitis GV. A simulation-optimization framework for Research and Development Pipeline management. AIChE J 2001;47:2226–42.
Varma VA, Reklaitis GV, Blau GE, Pekny JF. Enterprise-wide modeling & optimization: an overview of emerging research challenges and opportunities. Comp Chem Eng 2007;31:692–711.
Wan XT, Pekny JF, Reklaitis GV. Simulation-based optimization with surrogate models – application to supply chain management. Comp Chem Eng 2005;29:1317–28.
Wang GG, Shan S. Review of metamodeling techniques in support of engineering design optimization. J Mech Design 2007;129:370–80.
Wassick JM, Agarwal A, Akiya N, Ferrio J, Bury S, You F. Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products. Comp Chem Eng 2012;47:157–69.
You F, Grossmann IE. Design of responsive supply chains under demand uncertainty. Comp Chem Eng 2008;32:3090–111.
You F, Grossmann IE. Balancing responsiveness and economics in process supply chain design with multi-echelon stochastic inventory. AIChE J 2011a;57:178–92.
You F, Grossmann IE. Stochastic inventory management for tactical process planning under uncertainties: MINLP models and algorithms. AIChE J 2011b;57:1250–77.
You F, Pinto JM, Grossmann IE, Megan L. Optimal distribution-inventory planning of industrial gases: II. MINLP models and algorithms for stochastic cases. Ind Eng Chem Res 2011;50:2928–45.
Yue D, You F. Planning and scheduling of flexible process networks under uncertainty with stochastic inventory: MINLP models and algorithm. AIChE J 2013;59:1511–32.
Zipkin PH. Foundations of inventory management, vol. 2. New York: McGraw-Hill; 2000.