Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 218913, 17 pages
http://dx.doi.org/10.1155/2014/218913

Research Article

Design and Optimization of Capacitated Supply Chain Networks Including Quality Measures

Krystel K. Castillo-Villar,[1] Neale R. Smith,[2] and José F. Herbert-Acero[3]

[1] *Department of Mechanical Engineering, The University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA*
[2] *Quality and Manufacturing Center, Instituto Tecnológico y de Estudios Superiores de Monterrey, Eugenio Garza Sada 2501 Sur, 64849 Monterrey, NL, Mexico*
[3] *Department of Physics, Instituto Tecnológico y de Estudios Superiores de Monterrey, Eugenio Garza Sada 2501 Sur, 64849 Monterrey, NL, Mexico*

Correspondence should be addressed to Krystel K. Castillo-Villar; krystel.castillo@utsa.edu

Received 17 November 2013; Accepted 18 February 2014; Published 30 March 2014

Academic Editor: Vyacheslav Kalashnikov

This paper presents (1) a novel capacitated model for supply chain network design which considers manufacturing, distribution, and quality costs (named SCND-COQ model) and (2) five combinatorial optimization methods, based on nonlinear optimization, heuristic, and metaheuristic approaches, which are used to solve realistic instances of practical size. The SCND-COQ model is a mixed-integer nonlinear problem which can be used at a strategic planning level to design a supply chain network that maximizes the total profit subject to meeting an overall quality level of the final product at minimum costs. The SCND-COQ model computes the quality-related costs for the whole supply chain network considering the interdependencies among business entities. The effectiveness of the proposed solution approaches is shown using numerical experiments. These methods allow solving more realistic (capacitated) supply chain network design problems including quality-related costs (inspections, rework, opportunity costs, and others) within a reasonable computational time.

1. Introduction

The supply chain (SC) can be understood as the integration of all business entities that work together in order to ensure that the customer receives a product or service at the right time, with the right quality, and at low cost. To achieve this, it is necessary to coordinate all the business entities within a SC. This can be achieved through supply chain management (SCM). This paper addresses one of the problems included in SCM which is supply chain network design (SCND). The SCND aims at selecting the business entities that increase the overall performance of the SC. Cost of Quality (COQ) is a measurement system that translates the implications of poor quality into monetary terms. Although COQ has been applied mostly within enterprises, COQ can be applied as an external measure to integrate these costs into SCND modeling. Several studies have provided models to ensure quality in multistage SC [1]. Das [2] proposed a multistage global SC mathematical model for preventing recall risks. Srivastava [3], who initiates estimating COQ in a SC, measures COQ in monetary terms at selected third-party contract manufacturing sites of a pharmaceutical company. Ramudhin et al. [4] also focus on integrating COQ in the SC. Their seminal study presents a mathematical formulation that integrates known COQ functions into the modeling of a SC network for a single-product, three-echelon system and seeks to minimize the overall operational and quality costs. Ramudhin et al. [4] found that by adding a known and given quadratic COQ function that affects only the suppliers into the objective function results in a difference of approximately 16% in costs and changes the network selection. When COQ is not included, choices made solely on production costs could sacrifice quality and lead to additional quality nonconformance costs or corrective action costs in the subsequent stages of the SC. More recently, Alzaman et al. [5] established a mathematical model, considering an n level bill of materials, that incorporates a known COQ quadratic function based on a defect ratio at all SC nodes. As assumed in Ramudhin et al.'s work, the COQ function is known and is based on Juran's original model [6].

In previous studies, the COQ function, based on percentage of defective units, is assumed to be given. This paper deals with the development of an SCND-COQ model that computes the COQ for a whole SC based on internal decisions within the manufacturing plant, such as fraction defective at the manufacturing plant and error rate at inspection. No previous work has addressed how the COQ functions were obtained while taking internal operational decisions within the SC. Moreover, the proposed SCND-COQ model computes quality costs for the whole SC considering the interdependencies among business entities, whereas previous works have assumed independent COQ functions for each node of the SC.

This study aims to develop a strategic-level model for computing the COQ for a multistage, capacitated supply chain network design (SCND-COQ) problem. The proposed model is an extension of the serial supply chain model (SC-COQ model) developed by Castillo-Villar et al. [7] for which two solution procedures were developed [8]. The problem addressed here is significantly more difficult to solve due to the capacity constraints and the combinatorial nature of a MINLP (mixed-integer nonlinear programming) network problem. Moreover, it differs from the SC-COQ model in two main aspects: (1) several business entities can be selected at each echelon of the SC; (2) the components from all selected suppliers enter a plant and are mixed; thus, a shipment to a retailer contains products with components from different suppliers. Therefore, a pooled fraction defective from the selected suppliers is computed. Specialized solution procedures were developed to address this problem. The decision variables are to select the best combination of one or more suppliers, decide which plants of a given set to open, and select the best combination of one or more retailers in order to maximize the total profit and satisfy a minimum quality level for the final product as illustrated in Figure 1.

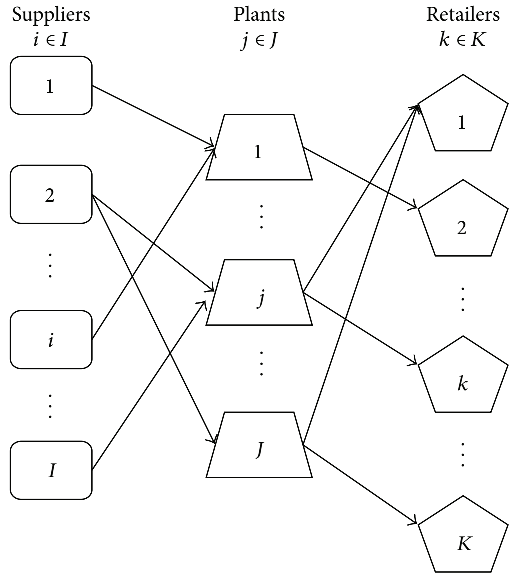


Figure 1: Illustration of a supply chain network.

2. The Capacitated SCND-COQ Model

The model assumptions are as follows.

(1) A consumer goods SC, consisting of three echelons: suppliers, manufacturers, and retailers, is modeled.

(2) A single product is modeled.

(3) The objective is to maximize profit.

(4) The overall quality level, QL, (see xvi) is sufficient to represent the quality of the final product.

(5) A new SC is being designed. Modifications to the model would be necessary if implemented for a manufacturing process which is currently working at a specific fraction defective at manufacturing and with an established inspection system.

(6) External failure costs can be tolerable for the firm. This implies direct applicability to products where external failure is not catastrophic for customers. However the overall quality level (QL) can be adjusted to address problems in which external failures are not desirable (e.g., aerospace and pharmaceutical industries).

(7) Suppliers and retailers are external to the manufacturing plant.

(8) It is assumed that a 100% inspection at the end of the manufacturing process is performed to check component conformance. Two types of errors may arise; Type I error involves classifying a good item as defective and Type II error involves labeling a defective item as good. Type I error is not considered in this model because it is not detrimental to customer satisfaction. Type I and Type II errors are defined in this paper as in [9], in terms of inspector fallibility.

(9) All defective products are returned by customers and incur external failure costs.

(10) The relevant operational costs are production, procurement, transportation from supplier to manufacturing plant, transportation from manufacturing plant to retailer, and a fixed cost for opening the plant.

(11) Customer demand at each retailer (Dem_k) is known for the study period; demand is determined by the retailers and communicated to the company. Therefore, retailers' capacity is not considered.

(12) Suppliers and manufacturing plants have finite capacity.

(13) At least one supplier, one plant, and one retailer must be selected (simplest supply chain network).

2.1. Computing the Quality Costs for the Capacitated Model. An example of the representation of the flow of items through

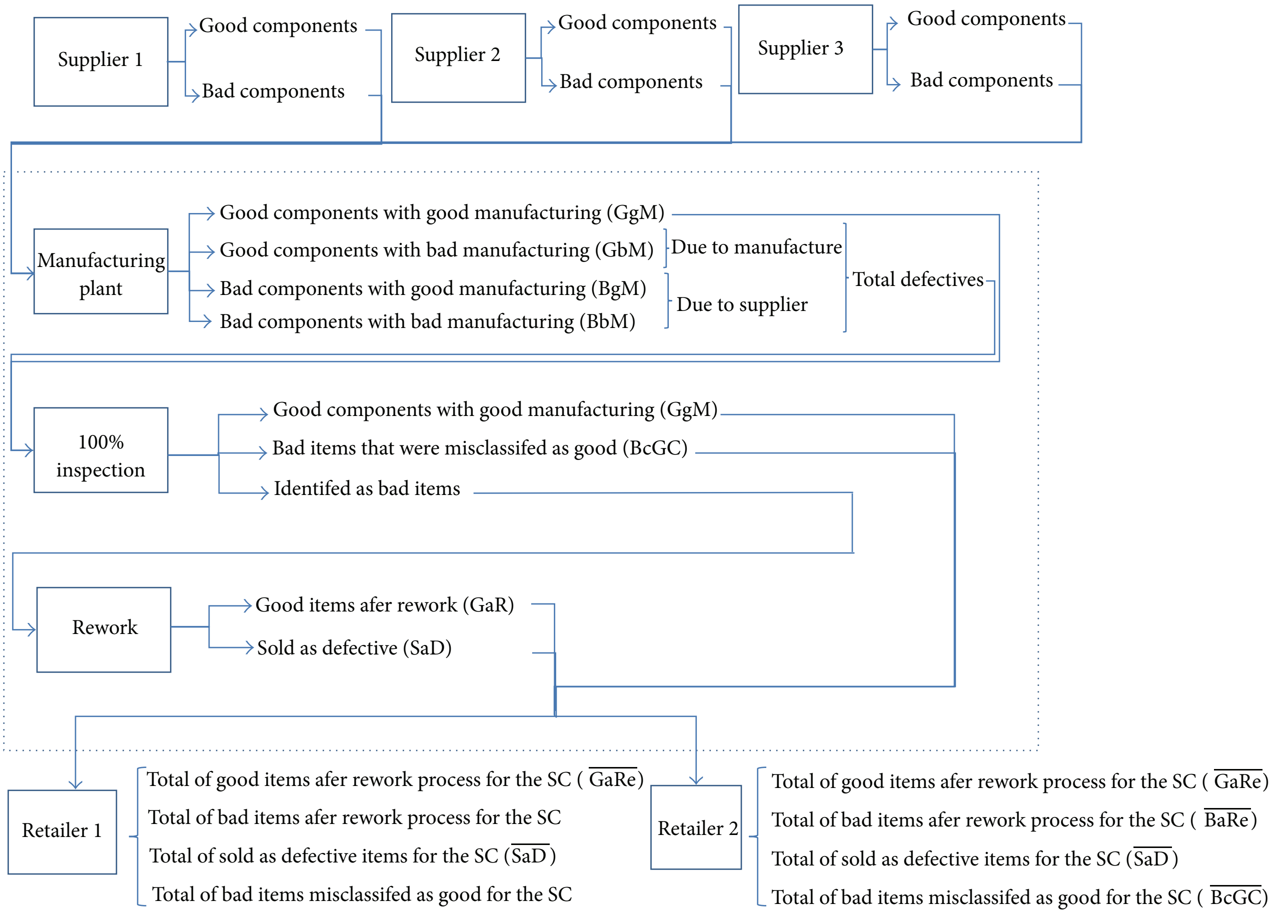

Supplier 1
Good components
Bad components
Supplier 2
Good components
Bad components
Supplier 3
Good components
Bad components
Manufacturing plant
Good components with good manufacturing (GgM)
Good components with bad manufacturing (GbM)
Bad components with good manufacturing (BgM)
Bad components with bad manufacturing (BbM)
Due to manufacture
Due to supplier
Total defectives
100% inspection
Good components with good manufacturing (GgM)
Bad items that were misclassifed as good (BcGC)
Identifed as bad items
Rework
Good items afer rework (GaR)
Sold as defective (SaD)
Retailer 1
Total of good items afer rework process for the SC (GaRe)
Total of bad items afer rework process for the SC
Total of sold as defective items for the SC (SaD)
Total of bad items misclassifed as good for the SC
Retailer 2
Total of good items afer rework process for the SC (GaRe)
Total of bad items afer rework process for the SC (BaRe)
Total of sold as defective items for the SC (SaD)
Total of bad items misclassifed as good for the SC (BcGC)

Figure 2: Representation of the flow of items through a supply chain network.

a SC network is shown in Figure 2 where a fixed network of three suppliers, one manufacturing plant, and two retailers is assumed.

The mathematical notation is presented below.

Sets

I: set of suppliers ($i \in I$).

J: set of manufacturing plants ($j \in J$).

K: set of retailers ($k \in K$).

Parameters

Dem_k: captured customer demand for retailer $k \in K$.

Cap_i: maximum capacity at supplier $i \in I$ for procuring components.

Cap_j: maximum capacity at manufacturing plant $j \in J$ for the production of items.

Y_{si}: fraction defective at supplier $i \in I$.

$\overline{Ys}_j = \sum_i Ys_i w_{ij} / \sum_i w_{ij}$: pooled fraction defective of all suppliers shipping products to manufacturing plant $j \in J$.

Y_{rk}: fraction defective at retailer $k \in K$.

p_{jk}: price per product sold by manufacturing plant $j \in J$ to retailer $k \in K$.

Pc_{ij}: direct cost of components shipped from supplier $i \in I$ to plant $j \in J$.

Po_{ij}: production cost (base cost) for component from supplier $i \in I$ transformed at manufacturing plant $j \in J$.

u_{ij}: cost of transporting one component from supplier $i \in I$ to plant $j \in J$.

l_{jk}: cost of transporting one item from plant $j \in J$ to retailer $k \in K$.

F_j: fixed cost for operating manufacturing plant $j \in J$.

Parameters for the COQ Function

A_{fj}: fixed cost for prevention activities at manufacturing plant $j \in J$.

A_{vi}: variable cost for prevention activities implemented by supplier $i \in I$.

A_{vj}: variable cost for prevention activities implemented by plant $j \in J$.

Av_{ij}: variable cost for combined prevention activities at supplier $i \in I$ and manufacturing plant $j \in J$.

Bf_j: fixed cost of inspection at manufacturing plant $j \in J$.

Bv_j: variable cost of inspection at manufacturing plant $j \in J$.

Cf_j: fixed cost for internal failure cost at manufacturing plant $j \in J$.

Cs_j: loss incurred due to failure of components procured from supplier to meet quality requirements (replacement costs and payroll costs incurred) at manufacturing plant $j \in J$.

Cr_j: rework cost per defective item at manufacturing plant $j \in J$.

$\overline{C}_j$: cost per defective item associated with repair or replacement of the product at manufacturing plant $j \in J$.

ϕ_j: rework rate at manufacturing plant $j \in J$.

$\bar{l} = (Cost/100)/(Ub - Lb)$: loss coefficient for the Taguchi loss function associated with the cost of working at the specification limit (for the whole network) and the width of the specification.

P^*_{jk}: price per "sold as defective" item sold by manufacturing plant j to retailer k.

Decision Variables

yI_j: inspection error rate at the output of manufacturing plant $j \in J$ (continuous variable between 0 and 1).

yp_j: fraction defective at manufacturing plant $j \in J$ (continuous variable between 0 and 1).

Z_i: binary variable which equals 1 if supplier $i \in I$ is selected, zero otherwise.

R_k: binary variable which equals 1 if retailer $k \in K$ is selected, zero otherwise.

P_j: binary variable which equals 1 if plant $j \in J$ is open, zero otherwise.

w^{sp}_{ij}: number of components shipped from supplier $i \in I$ to manufacturing plant $j \in J$.

w^{pr}_{jk}: number of components shipped from manufacturing plant $j \in J$ to retailer $k \in K$.

Expressions

(i) $\mathrm{GgM}(w^{sp}_{ij}, yp_j, Z_i, P_j) = \sum_i \sum_j (1 - Ys_i) w^{sp}_{ij} (1 - yp_j) Z_i P_j$ represents good components with successful manufacturing for the whole network.

(ii) $\mathrm{GbM}(w^{sp}_{ij}, yp_j, Z_i, P_j) = \sum_i \sum_j (1 - Ys_i) w^{sp}_{ij} yp_j Z_i P_j$ represents good components with defective manufacture for the whole network. These defective products are due to the manufacturing process.

(iii) $\mathrm{BgM}(w^{sp}_{ij}, yp_j, Z_i, P_j) = \sum_i \sum_j Ys_i w^{sp}_{ij} (1 - yp_j) Z_i P_j$ represents bad components with successful manufacture for the whole network. These defective products are due to the supplier.

(iv) $\mathrm{BbM}(w^{sp}_{ij}, yp_j, Z_i, P_j) = \sum_i \sum_j Ys_i w^{sp}_{ij} yp_j Z_i P_j$ represents bad components with defective manufacture for the whole network. These defective products are due to the supplier.

(v) $\mathrm{GaRe}(w^{sp}_{ij}, yp_j, yI_j, Z_i, P_j) = \sum_i \sum_j \phi_j (1 - yI_j) w^{sp}_{ij} [(1 - Ys_i) yp_j + Ys_i] Z_i P_j$ is a function that returns the number of good products after successful rework before leaving the plants.

(vi) $\mathrm{SaD}(w^{sp}_{ij}, yp_j, yI_j, Z_i, P_j) = \sum_i \sum_j (1 - \phi_j)(1 - yI_j) w^{sp}_{ij} [(1 - Ys_i) yp_j + Ys_i] Z_i P_j$ is a function that returns the number of defective products which will be sold at a reduced price before leaving the plants.

(vii) $\mathrm{BcGC}(w^{sp}_{ij}, yp_j, yI_j, Z_i, P_j) = \sum_i \sum_j yI_j w^{sp}_{ij} [(1 - Ys_i) yp_j + Ys_i] Z_i P_j$ is a function that returns the bad items that were misclassified as good at inspection before leaving the plants.

(viii) $\overline{\mathrm{SaD}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \sum_j \sum_k (1 - \phi_j)(1 - yI_j) w^{pr}_{jk} [(1 - \overline{Y}s_j) yp_j + \overline{Y}s_j] P_j R_k$ is a function that returns the number of defective products which will be sold at a reduced price at retailers. The computation utilizes the pooled fraction defective at suppliers because the items shipped to the retailer may come from different suppliers.

(ix) $\overline{\mathrm{SaD}}_j(w^{pr}_{jk}, yp_j, yI_j, R_k) = \sum_k (1 - \phi_j)(1 - yI_j) w^{pr}_{jk} [(1 - \overline{Y}s_j) yp_j + \overline{Y}s_j] R_k$ is a function that quantifies the number of defective products which will be sold at a reduced price at retailers *per manufacturing plant*.

(x) $\overline{\mathrm{BcGC}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \sum_j \sum_k yI_j w^{pr}_{jk} [(1 - \overline{Y}s_j) yp_j + \overline{Y}s_j] P_j R_k$ is a function that returns the number of bad products after the manufacturing process. The computation utilizes the pooled fraction defective at suppliers.

(xi) $\overline{\mathrm{BcGC}}_j(w^{pr}_{jk}, yp_j, yI_j, R_k) = \sum_k yI_j w^{pr}_{jk} [(1 - \overline{Y}s_j) yp_j + \overline{Y}s_j] R_k$ is a function that quantifies the number of bad products after the manufacturing process *per manufacturing plant*.

(xii) $\overline{\mathrm{GaRe}}\,(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \sum_j \sum_k (1 - Yr_k) [(1 - \overline{Y}s_j) w^{pr}_{jk} (1 - yp_j) + \phi_j (1 - yI_j) w^{pr}_{jk} ((1 - \overline{Y}s_j) yp_j + \overline{Y}s_j)]\ P_j R_k$ is a function that returns the number of good products delivered to the customers for the whole network.

(xiii) $\overline{\mathrm{BaRe}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \sum_j \sum_k Yr_k w^{pr}_{jk} [(1\text{-}\overline{Y}s_j)(1\text{-}yp_j) + \phi_j (1 - yI_j)((1 - \overline{Y}s_j) yp_j + \overline{Y}s_j)] P_j R_k$ is a function that returns the number of bad products delivered to the customers for the whole network.

(xiv) $\overline{\mathrm{BaRe}_j}(w^{pr}_{jk}, yp_j, yI_j, R_k) = \sum_k Yr_k w^{pr}_{jk} [(1 - \overline{Y}s_j)(1 - yp_j) + \phi_j (1 - yI_j)((1 - \overline{Y}s_j) yp_j + \overline{Y}s_j)] R_k$ is a function that quantifies the number of bad products delivered to the customers for the whole network *per manufacturing plant*.

(xv) $y(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \overline{\text{BaRe}} + \overline{\text{BcGC}} + \overline{\text{SaD}}/\sum_j \sum_k w^{pr}_{jk} P_j R_k$ is the overall fraction defective for the whole supply chain network.

(xvi) $QL_k(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \sum_j (1 - Yr_k)[(1 - \overline{Y}s_j)w^{pr}_{jk}(1 - yp_j) + \phi_j(1 - yI_j)w^{pr}_{jk}((1 - \overline{Y}s_j)yp_j + \overline{Y}s_j)]P_j/\sum_j w^{pr}_{jk} P_j R_k$ is the overall quality level achieved at retailer k.

(xvii) $\text{QL}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k) = \overline{\text{GaRe}}/\sum_j \sum_k w^{pr}_{jk} P_j R_k$ is the overall quality level achieved by the supply chain network.

2.1.1. Prevention Cost. Prevention cost is linked to the production of good products after the manufacturing process as given by

$$\begin{aligned} C_P&\left(w^{sp}_{ij}, yp_j, Z_i, P_j\right) \\ &= \sum_j Af_j P_j \\ &\quad + \sum_i \sum_j Av_i\left(Ys_i\right)\left[w^{sp}_{ij}\left(1 - Ys_i\right) Z_i P_j\right] \\ &\quad + \sum_i \sum_j Av_j\left(yp_j\right)\left[w^{sp}_{ij}\left(1 - Ys_i\right)\left(1 - yp_j\right)\right] Z_i P_j \\ &\quad + \sum_i \sum_j Av_{ij}\left(Ys_i, yp_j\right) \\ &\qquad \times \left[w^{sp}_{ij}\left(1 - Ys_i\right)\left(1 - yp_j\right)\right] Z_i P_j, \end{aligned} \tag{1}$$

where Af_j is a fixed cost. The variable cost for prevention activities is divided into three scenarios: (1) the prevention activity is carried out only at suppliers; (2) the prevention activity is implemented only at the manufacturing plants; and (3) the prevention activity is a coordinated action between a supplier and plant. In the first case, $Av_i(Ys_i)$ is a function of the fraction defective at a supplier that returns the cost per unit of good components for prevention activities. Even though this prevention activity is carried out at the supplier, the cost of this prevention activity is incurred by the manufacturing plant. For the second (resp., third) case, $Av_j(yp_j)(Av_{ij}(Ys_i, yp_j))$ is a function of the fraction defective at the selected plant (resp., supplier) that quantifies the cost per unit of good product after the manufacturing process, $\text{GgM}(w^{sp}_{ij}, yp_j, Z_i, P_j)$. These three functions typically increase as the fraction defective at supplier and/or plant decreases (i.e., when the quality level of the supplier and/or manufacturing process improves).

2.1.2. Appraisal Cost. A 100% inspection is performed at the end of the manufacturing process to verify conformance. The appraisal costs (C_A) are modeled by a fixed cost and a variable cost per item that is classified accurately. Thus, the appraisal cost increases when inspection is more accurate. The appraisal cost is given by

$$\begin{aligned} C_A\left(w^{sp}_{ij}, yI_j, Z_i, P_j\right) &= \sum_j Bf_j P_j \\ &\quad + \sum_i \sum_j Bv_j w^{sp}_{ij}\left(1 - yI_j\right) Z_i P_j, \end{aligned} \tag{2}$$

where Bf_j is a fixed cost and Bv_j is a variable cost. All items going through the system are inspected; thus, w^{sp}_{ij} represents the number of items to be inspected.

2.1.3. Internal Failure Cost. The internal failure cost for the capacitated model is given by (3). The first term is a fixed cost (Cf_j) for corrective activities. The second term is the internal failure cost due to unsuccessful manufacture computed as rework cost per item (Cr_j) times the identified good components with unsuccessful manufacture. It is assumed that these components can be recovered. The third term is the purchasing failure cost computed as the sum of losses incurred due to failure of purchased components to meet quality requirements (Cs_j) and rework cost per item (Cr_j), multiplied by the number of items identified as defective due to bad components, $\phi_j(1 - yI_j)$ BgM $(w^{sp}_{ij}, yp_j, Z_i, P_j)$, as well as the items identified as defective because of bad components and unsuccessful manufacture, $\phi_j(1 - yI_j)$ BbM $(w^{sp}_{ij}, yp_j, Z_i, P_j)$. The fourth term represents the difference between what would have been the income from sales of nondefective or good products (p_{jk}) and the actual income from sales of defective items (P^*_{jk}) [10], times the items sold as defective, $\overline{\text{SaD}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k)$:

$$\begin{aligned} C_{IF}&\left(w^{sp}_{ij}, w^{pr}_{jk}, yI_j, yp_j, Z_i, P_j, R_k\right) \\ &= \sum_j Cf_j P_j \\ &\quad + \sum_i \sum_j Cr_j \phi_j\left(1 - yI_j\right)\left(1 - Ys_i\right) w^{sp}_{ij} yp_j Z_i P_j \\ &\quad + \sum_i \sum_j \left(Cs_j + Cr_j\right)\phi_j\left(1 - yI_j\right) w^{sp}_{ij} Ys_i Z_i P_j \\ &\quad + \sum_j \sum_k \left(p_{jk} - P^*_{jk}\right)\left(1 - \phi_j\right) w^{pr}_{jk}\left(1 - yI_j\right) \\ &\qquad \times \left[\left(1 - \overline{Y}s_j\right) yp_j + \overline{Y}s_j\right] P_j R_k. \end{aligned} \tag{3}$$

2.1.4. External Failure Cost. The external failure costs and opportunity costs are given by

$$\begin{aligned} C_{EF}&\left(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k\right) \\ &= \sum_j \sum_k \overline{C}_j Yr_k w^{pr}_{jk}\left\{\left(1 - \overline{Y}s_j\right)\left(1 - yp_j\right)\right. \end{aligned}$$

$$\begin{aligned} &+ \phi_j \left(1 - yI_j\right) \\ &\quad \times \left[\left(1 - \overline{Y}s_j\right) yp_j + \overline{Y}s_j\right]\Big\} P_j R_k \\ &+ \sum_j \sum_k \overline{C}_j yI_j w^{pr}_{jk} \left[\left(1 - \overline{Y}s_j\right) yp_j + \overline{Y}s_j\right] P_j R_k \\ &+ \sum_j \overline{l}_j \left(y_j - Lb_j\right)^2 P_j, \end{aligned} \tag{4}$$

where the first and second terms represent the costs generated by defective items returned by customers, which is the product of $\overline{C}_j$ and the sum of the bad items due to the retailer, $\overline{\text{BaRe}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k)$, and the bad items that were classified as good, $\overline{\text{BcGC}}(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k)$. The third term is based on the Taguchi loss function. The loss constant coefficient, $\overline{l}$, depends on the cost for working at the specification limits and the width of the specification [11]. The cost for working at the specification limits (cp_j) is computed as a proportion (given by the parameter cost) of the income of the items sold by each manufacturing plant as given by

$$cp_j = Cost \sum_k w^{pr}_{jk} p_{jk} \quad \forall j \in J. \tag{5}$$

The width of the specification is defined by an upper limit (Ub_j) which is set to 100 to indicate the allowable deviation from target value. Notice that $y = 100\%$ is the worst case, that is, when the process has 100% defective products. Perfect inspection and manufacturing are assumed in (8) in order to obtain a lower bound or target value (Lb_j) for the Taguchi function as given by (6). Thus, the width of the specification is $Ub_j - Lb_j$:

$$Lb_j = \frac{\sum_k Yr_k w^{pr}_{jk} \left[\left(1 - \overline{Y}s_j\right) + \phi_j \overline{Y}s_j\right] R_k}{\sum_k w^{pr}_{jk} R_k} \times 100\% \quad \forall j \in J. \tag{6}$$

The loss coefficient is given by

$$\overline{l_j} = \frac{cp_j}{(100)\left(Ub_j - Lb_j\right)} \quad \forall j \in J. \tag{7}$$

The quality characteristic, y_j, is the overall percentage defective for plant j as shown in (8). The quality characteristic includes $\overline{\text{BaRe}}_j(w^{pr}_{jk}, yp_j, yI_j, R_k)$, $\overline{\text{BcGC}}_j(w^{pr}_{jk}, yp_j, yI_j, R_k)$, and $\overline{\text{SaD}}_j(w^{pr}_{jk}, yp_j, yI_j, R_k)$. Consider

$$\begin{aligned} y_j = &\Big(\overline{\text{BaRe}}_j \left(w^{pr}_{jk}, yp_j, yI_j, R_k\right) \\ &+ \overline{\text{BcGC}}_j \left(w^{pr}_{jk}, yp_j, yI_j, R_k\right) \\ &+ \overline{\text{SaD}}_j \left(w^{pr}_{jk}, yp_j, yI_j, R_k\right)\Big) \\ &\times \left(\sum_j \sum_k w^{pr}_{jk} P_j R_k\right)^{-1} \times 100\% \quad \forall j \in J. \end{aligned} \tag{8}$$

In order to compute the opportunity loss for the SC, a relative value of the quality characteristic is obtained by subtracting the target value (a lower bound) from the current overall percentage defective as shown in the last term of (4). In summary, the Taguchi loss is computed for each plant and these costs are summed to obtain the total opportunity loss for the network.

The total COQ is computed as the sum of the prevention, appraisal, and internal and external failure expressions as given by

$$\begin{aligned} &\text{COQ}\left(w^{sp}_{ij}, w^{pr}_{jk}, yI_j, yp_j, Z_i, P_j, R_k\right) \\ &\quad = C_P\left(w^{sp}_{ij}, yp_j, Z_i, P_j\right) + C_A\left(w^{sp}_{ij}, yI_j, P_j\right) \\ &\quad + C_{IF}\left(w^{sp}_{ij}, yI_j, yp_j, Z_i, P_j, R_k\right) \\ &\quad + C_{EF}\left(w^{pr}_{jk}, yp_j, yI_j, P_j, R_k\right). \end{aligned} \tag{9}$$

2.2. Mathematical Formulation of the SCND-COQ Model. The objective function is to maximize profit

$$\begin{aligned} &\sum_{j \in J} \sum_{k \in K} w^{pr}_{jk} p_{jk} P_j R_k - \text{COQ}\left(w^{sp}_{ij}, w^{pr}_{jk}, yI_j, yp_j, Z_i, P_j, R_k\right) \\ &\quad - \sum_{i \in I} \sum_{j \in J} w^{sp}_{ij} Pc_{ij} Z_i P_j - \sum_{i \in I} \sum_{j \in J} w^{sp}_{ij} Po_{ij} Z_i P_j \\ &\quad - \sum_{i \in I} \sum_{j \in J} w^{sp}_{ij} u_{ij} Z_i P_j - \sum_{j \in J} \sum_{k \in K} w^{pr}_{jk} l_{jk} P_j R_k - \sum_{j \in J} F_j P_j \end{aligned} \tag{10}$$

subject to

$$\sum_j w^{pr}_{jk} \le Dem_k R_k; \quad \forall k \in K, \tag{11}$$

$$\sum_i w^{sp}_{ij} = \sum_k w^{pr}_{jk}; \quad \forall j \in J, \tag{12}$$

$$\sum_i w^{sp}_{ij} \le Cap_j P_j; \quad \forall j \in J, \tag{13}$$

$$\sum_j w^{sp}_{ij} \le Cap_i Z_i; \quad \forall i \in I, \tag{14}$$

$$QL_k \ge lR_k; \quad \forall k \in K, \tag{15}$$

$$0 \le yI_j \le 1; \quad \forall j \in J, \tag{16}$$

$$0 \le yp_j \le 1; \quad \forall j \in J, \tag{17}$$

$$\begin{aligned} &Z_i \in \{0,1\}, \quad P_j \in \{0,1\}, \quad R_k \in \{0,1\}; \\ &\forall i \in I, \quad \forall j \in J, \quad \forall k \in K. \end{aligned} \tag{18}$$

The objective function given in (10) represents the profit; it is a nonconvex and nonlinear function and has seven components. The first term is the sales revenue. The second term represents the total COQ for the network. The third term represents the direct cost of acquiring components from the selected supplier(s) by the opened manufacturing plant(s). The fourth term represents the operational cost for the components from selected supplier(s) processed at the opened plant(s). The fifth term gives the transportation cost from the supplier(s) to opened plant(s). The sixth term represents the transportation costs from the opened plant(s) to the retailer(s). Lastly, the seventh component determines the fixed cost for opening plants.

Constraints in (11) enforce that demand at retailers is not exceeded. Constraints in (12) ensure that the number of components shipped from suppliers to manufacturing plantsequals the number of items shipped from manufacturing plants to retailers. Constraints in (13) ensure that the plant capacity (in units) is not exceeded. Since the same number of components received from the suppliers is transformed by the manufacturing plants and shipped to the retailers as either good items or sold as defective items, the capacity corresponds to an entry-exit capacity (maximum flow allowed within the plants). Constraints in (14) enforce that the exit capacity (in units) at the suppliers is not exceeded. The exit capacity is the amount of components shipped to manufacturing plants. Constraints in (15) are the quality level constraints; thus, the quality of the final product delivered at each retailer must meet the minimum required quality level; this set of constraints is nonlinear. Constraints shown in (16) and (17) define feasible ranges and binary requirements for the model variables.

3. Solution Procedures

The capacitated SCND-COQ model is a constrained mixed-integer nonlinear programming problem (MINLP) which is challenging to solve because it combines all the difficulties of both of its subcategories: the combinatorial nature of mixed integer programming (MIP) and the difficulty of solving non-convex nonlinear problems (NLP). These two subcategories are known as NPO-complete problems [12]; thus, solving MINLP problems can be a daunting task. Metaheuristics have proven to be computationally more efficient than gradient-based nonlinear programming methods for MINLP. Hence, developing an effective metaheuristic-based algorithm to deal with problems of practical and realistic size is preferred.

Five procedures for solving the capacitated SCND-COQ model are described and compared. Three heuristic procedures are based on the serial model and they can be divided into two stages. Stage I consists of finding serial logistic routes, a serial route being a combination of three entities (supplier-plant-retailer), with the highest profit per unit sold, which are to be added to the network at each iteration of the procedure. Once a feasible network is constructed, Stage II consists of evaluating the feasible network configuration using the capacitated SCND-COQ model. The difference among the serial-based procedures lies in how the search of the serial routes to be added to the network at each iteration is performed. The first heuristic procedure enumerates all the serial routes and applies a value-restricted selection for finding the serial logistic route to be added to the network at each iteration of the algorithm. The second procedure uses the local search metaheuristic simulated annealing (SA) for finding the serial logistic route to be added to the network at each iteration of the algorithm. The third procedure uses a population-based metaheuristic, the genetic algorithm (GA), for finding the serial logistic route to be added to the network at each iteration of the algorithm. In addition, two solution procedures which are not based on finding serial logistic routes were developed: (1) an exhaustive enumeration of all possible networks with calls to a GlobalSearch (GS) algorithm and (2) an exhaustive enumeration of all possible networks with calls to a MultiStart (MS) algorithm. A detailed description of each of the five solution procedures is presented in Sections 3.1 and 3.2.

3.1. Heuristic Procedures Based on Adding Serial Logistic Routes.

The number of possible serial logistic routes is computed as $|I| \times |J| \times |K|$. For instance, the number of serial routes for a problem with 5 suppliers, 3 plants, and 5 retailers is 75, which is considerably less than the number of possible network configurations, 6,727. The total number of possible network configurations is given by the following expression:

$$\sum_{i=1}^{|Z|} C(|Z|, i) \sum_{j=1}^{|P|} C(|P|, j) \sum_{k=1}^{|R|} C(|R|, k), \quad (19)$$

where $C(n, r)$ is the number of ways in which r items can be selected from among n items without replacement.

The heuristic procedures based on the serial model rely on the following idea: a network can be constructed by (1) choosing the serial logistic route with the highest profit per unit sold when sending the maximum possible amount of items through that route, (2) adding that serial route to the network, (3) updating the remaining capacities, and repeating the process.

The heuristic procedures serve to construct a feasible network and to determine the amount of items to be sent (Stage I). Despite the ease of implementation and speed of these heuristic procedures for constructing a network, some limitations exist. For each serial route, the internal decision variables error rate at inspection (yI_j) and fraction defective at manufacturing (yp_j) must be determined. This implies that, for example, a manufacturing plant included in multiple serial routes could have different values of the internal decision variables. Operating the real system in this way may not be feasible or desirable. To remedy this, a reoptimization of the internal decision variables (yI_j and yp_j) is performed for the constructed feasible SC network by using the capacitated SCND-COQ model (Stage II). The profit achieved by this network is taken as the best-found solution for the capacitated model. The general procedure for the heuristic serial-based solution methods is depicted as follows.

Stage I

(1.1) Create a list of all possible serial routes ($|I| \times |J| \times |K|$).

(1.2) Compute the quality level attained by each serial route, eliminate the routes that do not meet the minimum level in (15), and save the resultant matrix with all the feasible serial routes (**PS** matrix).

(1.3) Determine the maximum flow that can be sent through a route by evaluating the following: $\min\{\sum_{i\in I} Cap_i, \sum_{j\in J} Cap_j, \sum_{k\in K} Dem_k\}$.

(1.4) Prelocate the vector with not opened plants (NOP). Since the same plant can be selected in several serial routes (as long as the remaining plant's capacity is greater than zero), this vector avoids taking the fixed cost for opening a plant into account more than once.

(1.5) The search for additional serial routes to be added to the network continues until one of the five following cases occurs: nonpositive profit is obtained, the sum of the capacities of the suppliers is exhausted, the sum of the capacities of the plants is exhausted, the demand is satisfied, or there are no more feasible remaining routes to select from (the updated **PS** matrix is empty).

(1.5.1) The search is performed by using one of the following procedures: (1) evaluation of all the serial routes and a greedy or value-restricted constructive (VRC) procedure, (2) simulated annealing (SA), and (3) the genetic algorithm (GA). The details of the procedures are described in Sections 3.1.1, 3.1.2, and 3.1.3.

(1.5.2) Update the remaining capacities and demands.

(1.5.3) One or more of these three cases may occur: one supplier is saturated, one plant is saturated, or the demand at one retailer is fully satisfied. In each case, the business entities that were saturated are eliminated from the set of potential business entities and all the routes that include these business entities are eliminated from the matrix with possible serial routes (**PS** matrix).

(1.5.4) Update the NOP vector each time a plant is selected. For instance, if the selected route contains a plant that was already opened in a previous iteration, then the additional fixed cost is zero; otherwise, if the plant is in the NOP vector, then a fixed cost is incurred for opening that plant.

(1.6) Store results.

Stage II

The network with flows formed by adding serial routes is evaluated by using the capacitated SCND-COQ model and the internal continuous variables are reoptimized. It is worth noting that the network and flows found in Stage I are not modified.

(2.1) Reoptimize the internal continuous variables associated with the opened manufacturing plants by using the GlobalSearch algorithm in MATLAB.

3.1.1. Value-Restricted Constructive Procedure for Selecting Serial Routes. For each serial route (rows in the possible serial routes matrix, that is, the **PS** matrix), the internal decision variables (yI_j and yp_j) that minimize the total COQ are obtained by using a nonlinear solver, FMINCON with the interior-point algorithm of MATLAB. The total profit and the profit per unit sold are computed for each serial route. The profit per unit sold is used to select a serial route; this avoids selecting the route that generates the maximum profit based on volume. Ties are broken by selecting the route that yields a higher total profit.

Value-restricted selection (VRS) was used to choose the serial route to be added to the network at each iteration of the procedure. The selection of the logistic route at each iteration is determined by choosing a random route from a restricted candidate list (RCL). The RCL was used as a way to avoid a greedy choice as in the GRASP (greedy randomized adaptive search procedure) metaheuristic; refer to [13–15].

The RCL contains the routes with the higher values of profit per unit sold. Let δ be a real value such that $0 \le \delta \le 1$. The RCL consists of all the routes e such that the greedy function $c(e)$, total unit profit, is $c(e) \ge c^* - \delta(c^* - c_*)$. When $\delta = 0$, then the selection is greedy and when $\delta = 1$, the selection is completely randomized. The serial-based heuristic procedure using a VRS for selecting the serial routes at each iteration and the GlobalSearch method for optimizing the internal decision variables of the constructed network is named SVRC (serial value-restricted constructive procedure) and was implemented using MATLAB. The SVRC algorithm with $\delta = 0.1$ (δ was obtained from previous computational experiments) is named SVRC1 and with $\delta = 0$ (which is a greedy procedure) is named SVRC2).

3.1.2. SA for Selecting Serial Routes. Descriptions of the general SA procedure can be found in [16–18]. The SA-based solution procedure used to find the serial route that maximizes unit profit while satisfying a required quality level of the final product is described in [8]. The SA-based procedure chooses the serial route that yields the highest-found profit per unit sold at each iteration. In contrast to the SVRC procedures, SA search procedure does not enumerate all the serial routes at each iteration of the heuristic procedure. The SA serial-based heuristic procedure using the GlobalSearch method for optimizing the internal decision variables of the constructed network is named SSA1 (serial simulated annealing method version 1) and was implemented using MATLAB.

Two additional variations of this method were developed: SSA2 and SSA3. SSA2 has as decision variables the selection of serial routes while the internal continuous variables (yI_j and yp_j) are optimized using the nonlinear solver FMINCON with the interior-point algorithm of MATLAB. The difference between SSA1 and SSA2 is the use of FMINCON for the optimization of the continuous variables.

SSA3 also has as decision variables the selection of serial routes while the internal continuous variables (yI_j and yp_j) are optimized using the nonlinear solver FMINCON with the interior-point algorithm of MATLAB. The difference between

SSA1 and SSA2 versus SSA3 lies in the state representation. SSA1 and SSA2 determine the next neighborhood solution by indicating how many indexes apart from the current index of the **PS** matrix (matrix where rows represent feasible combinations of entities) this next solution could be. For SSA3, the state is represented by a vector containing three indexes, and each index represents an entity and its value ranges from 1 to the amount of entities of each type. When an unfeasible route is selected (that is not in **PS** matrix), a penalty is performed. In such cases, the value of the objective function is set to zero. This state representation for determining the next neighboring solution was implemented in SSA3 to gain insight about the impact of the state representation on the solution procedure's performance.

3.1.3. GA for Selecting Serial Routes. Genetic algorithms, introduced by Holland [19], refer to a class of adaptive search procedures based on the principles derived from natural evolution and genetics. A GA solution procedure to find the serial route that maximizes unit profit and satisfies a required quality level of the final product is described in [8]. The GA based procedure chooses the serial route that yields the highest profit per unit sold at each iteration. In contrast to the SVRC procedures, the GA search procedure does not enumerate all the serial routes at each iteration of the heuristic procedure. The serial-based heuristic procedure using GA and GlobalSearch is named SGA1 (serial genetic algorithm method version 1) and was implemented using MATLAB. Similar to SSA, SGA has three variations. SGA2 and SGA3 algorithms call the nonlinear solver FMINCON with the interior-point algorithm of MATLAB for the optimization of the internal continuous variables while SGA1 optimizes all decision variables (selection of entities and internal continuous variables). For SGA1 and SGA2, the representation of the individual genotype consists of binary numbers that in decimal base represent an index in the **PS** matrix. In SGA3, the individual genotype is segmented in three parts that define the serial route; each part consists of binary numbers that in decimal base represent an independent entity (supplier, plant, or retailer). Equivalently to SSA3, this representation was implemented to gain insight about the impact of the individual representation on the solution procedure's performance. During the optimization process (selection of serial routes), evolutionary operations (selection, crossover, and mutation) are performed over the population in order to improve the population fitness over generations. A binary tournament selection and a one-point crossover method were adopted in the present work for all GA-based solution procedures. When an unfeasible route is selected (that is not in **PS** matrix), a penalty is performed. In such cases, the value of the objective function is set to zero.

3.2. Exhaustive Enumeration Procedures. Exhaustive enumeration consists of listing all possible SC networks. For each SC network, the MATLAB GlobalSearch (GS) and the MultiStart (MS) algorithms are used to solve for the amount of items to be sent between suppliers and manufacturing plants and between plants and retailers (i.e., w_{ij}^{sp} and w_{jk}^{pr} matrices, resp.) as well as the variables representing the quality system within the manufacturing plant (i.e., yI_j and yp_j). Both algorithms have similar approaches; they run a nonlinear solver (FMINCON) from multiple starting points and try to maximize the total profit of the SC network while satisfying the SCND-COQ model constrains. The solution with the highest profit (related to one of the many SC network configurations with optimized variables) is reported as the best found.

It should be noted that the exhaustive enumeration of all possible SC networks does not necessarily mean that all the possible solutions in the search space are evaluated, as each possible SC network contains infinite possible solutions, which depend on the values of the decision variables (w_{ij}^{sp}, w_{jk}^{pr}, yI_j, and yp_j). Moreover, the nonlinear solvers will not always find optimal solution because the solvers may return a local maximum.

Since the number of networks grows exponentially as the number of business entities increases, as shown in (19) and as described in Section 3.1, exhaustive enumeration can only be performed for small problems and heuristic procedures are needed to deal with problems of practical and realistic size. For instance, a problem with 10 suppliers, 15 manufacturing plants, and 2 retailers has 3.5148×10^9 possible SC network configurations and 237 decision variables.

4. Experimental Study

To investigate the effectiveness of the developed solution procedures, a variety of network sizes (as shown in Table 1) were solved. Moreover, three classes of instances were defined and five instances from each class were randomly generated. The pool of test problems consists of 60 problems (15 for each problem size).

In order to compare the performance of the solution procedures against optimal solutions, specially constructed instances (problem class I described in Subsection 4.1.1) were constructed such that the optimal solution is known in advance. Additional class problems where the optimal solution is not known were also generated (classes II and III). A comparison of the solution procedures relative to each other was performed for classes II and III.

4.1. Test Problem Generation. The general approach used to generate instances is described next. The data used in test problems was generated randomly from a uniform distribution between the low and high levels documented in Table 2. The minimum required quality level (l) is fixed at 0.85 for all test instances. When the required quality level is too low, the solution space will be larger and finding the near-optimal solutions will be more difficult. When the required quality level is too high, the solution space will be smaller and highly constrained. The interested reader can obtain the test problems used in the development of this paper from the authors.

The function of the variable cost for prevention activities was considered as in scenario 3 (as described in Section 2.1.1),

Table 1: Test problems.

Number of suppliers (I)	Number of plants (P)	Number of retailers (R)	Number of constraints	Number of decision variables
3	2	3	13	24
5	4	6	25	67
8	8	10	44	186
35	20	35	145	1530

Table 2: Ranges of the parameters used to generate the instances.

Input parameter	Low level	High level
Fraction defective at supplier (Ys_i)	0.05	0.2
Fraction defective at retailer (Yr_k)	0.05	0.1
Extra percentage (extra) in price (p_{jk})	1.2	1.3
Procurement costs (Pc_{ij})	50	120
Production costs (Po_{ij})	70	130
Transportation costs: u_{ij} and l_{jk}	3	12
Fixed cost for opening manufacturing plants (F_j)	80,000	120,000
Fixed costs: Af_j, Bf_j, and Cf_j	5,000	15,000
Rework cost (Cr_j).	70	90
Loss incurred owing to failure of purchased components (Cs_j)	0.45 of average Pc_{ij}	0.55 of average Pc_{ij}
Variable cost for prevention activities ($Av_{ij}(Ys_i, yp_j)$)	Bv_j	5 Bv_j
Variable cost for appraisal/inspection activities (Bv_j)	5	5
Price per "sold as defective" items (P^*_{jk})	1/4 p_{jk}	3/4 p_{jk}
Cost for computing Taguchi loss function for the network ($Cost$)	1/10	1/3
Cost per defective item ($\overline{C}_j$)	1/4 of avg. price	1/2 of avg. price
Dem_k, Cap_i, and Cap_j	50,000	80,000

where $Av_{ij}(Ys_i, yp_j) = c$ (c is a constant) for the test problems. The rework rate (ϕ_j) is set to $Cr_j/100$. The price (p_{jk}) is calculated using (20). The price considers an extra percentage (extra) for revenue and for covering administrative costs as shown in Table 2. The vectors of average cost for all potential suppliers for plant j are differentiated from the matrices by using a bar ($\overline{Av}_j$ for prevention activities; $\overline{Pc}_j$ for procurement costs; and $\overline{u}_j$ and $\overline{l}_j$ for transportation costs) and used in

$$p_{jk} = \text{round}\left(\sum_k \left(\overline{Av}_j + Bv_j + Cr_j + Cs_j + \overline{Pc}_j + Po_j + \overline{u}_j + \overline{l}_j + \frac{\left(Af_j + Bf_j + Cf_j\right)}{Dem_k}\right)\right)\text{extra}. \tag{20}$$

The cost for computing the Taguchi loss function ($Cost$) is given by

$$Cost = \sum_{k\in K} w^{pr}_{jk} p_{jk} U(a,b) \quad \forall j \in J, \tag{21}$$

where p_{jk} is the average price and $U(a,b)$ is the continuous uniform distribution of the low and high levels in Table 2 for $Cost$.

4.1.1. Class I Test Instances. In class I problem instances, the optimal solution is a serial route that satisfies all the demand at the selected retailers. This was accomplished by generating instances as described above but setting the extra percentage (extra) to 0.5, instead of the interval shown in Table 2.

Once the parameter values are generated from uniform distributions with ranges as shown in Table 2, some of the parameters associated with the business entities in the optimal serial route (Z^* will denote the optimal supplier selection, P^* the optimal plants, and R^* the optimal retailers) are modified in order to force the optimal solution to be a specific serial route. A ratio $\beta = 0.6$ will be used to adjust the parameters of Z^*, P^*, and R^* so that the cost parameters of these entities are favorable (significantly lower than the rest of the values of the cost parameters of other business entities) to make this specific serial route the optimal solution. The optimal business entities' cost parameters are modified by taking the low level in the ranges in Table 2 and multiplying by β. In this way, the costs are $(1-\beta)\%$ less than the rest of the cost parameters. The fraction defective at the supplier and retailer are also decreased by $(1-\beta)\%$ for Z^* and R^*.

The rework rate (ϕ_j) is modified for P^* by considering the maximum rework rate among all the plants and then dividing by 100. The demand at R^* is set to the highest demand generated multiplied by $1+\beta$. The capacities at Z^* and P^* are set such that they exactly match the demand at R^*.

The sales price is set to three times the maximum generated price. The price of "sold as defective" items of P^* is computed as the sales price multiplied by the high level of P^*_{jk} in the range shown in Table 2.

The instances created by using the above procedure are verified by enumerating and evaluating all the possible serial routes. The maximum flow is sent through the route and the internal continuous variables are found by using a nonlinear solver to obtain the profit. The solution with the maximum profit should select Z^*, P^*, and R^*; otherwise, the instance is not used for testing.

4.1.2. Class II Test Instances. For class II problem instances, the opening of all the business entities to satisfy the demand at retailers is expected. This was accomplished by increasing the price of the final items and modifying capacities so that the retailers limit the flow.

The parameter values are randomly generated from uniform distributions with ranges as shown in Table 2. However, the price for class II problems ranges from 1.9 to 2. The higher prices are conducive to networks with positive profit (even when the quality costs are higher in some networks than others). Moreover, the capacities are modified so that retailers limit the amount of items to be sent. The sum of the randomly generated retailer capacities Dem_k is multiplied by 1.1 and divided by the number of suppliers to obtain the capacity at each supplier. This calculation is repeated for plants. Thus, the suppliers and plants have enough capacity to satisfy the demand at retailers.

4.1.3. Class III Test Instances. The data for class III problems was generated randomly from a uniform distribution between the low and high levels documented in Table 2 as described in the general approach presented at the beginning of Section 4.1.

4.2. Computational Results. Instances were solved using each of the five proposed solution procedures. Since the number of SC network configurations grows exponentially as the size of the problem increases, exhaustive enumeration procedures are used only for the 3 × 2 × 3 test problem size. For the rest of the problems, a comparison of SVRC, SSA, and SGA procedures (including its different versions) is performed.

4.2.1. Runs and Parameter Setting for the Serial-Based Solution Procedures. The SA algorithm is run 5 times in order to select one serial route with the best-found objective value at each iteration of the heuristic procedure. The GA algorithm is also run 5 times. The VRC procedure performs an enumeration of all the possible serial routes and selects one from the RCL at each iteration of the heuristic procedure. The whole procedure (steps (1.5) and (1.6) in Section 3.1) is run 5 times and the network configuration with the best accumulated profit was reported.

The heuristic parameter for SVRC1 and SVRC2 is δ. The heuristic parameters for SSA1, SSA2, and SSA3 (phase I) are the neighborhood size for the internal decision variables (*neigsize*), the initial system temperature (T_o), the rate of cooling (α), the number of accepted trials (*AccepTrials*), the maximum number of trials (*MaxTrials*), and the maximum number of Markov Chains (*MaxChains*). The heuristic parameters for SGA1, SGA2, and SGA3 (phase I) are the number of generations (*gen*), probability of mutation (*pm*), the probability of crossover (*pc*), and the initial population (*pop*) which is computed as a percentage of the **PS** matrix. All heuristic parameters were tuned using statistical designs of experiments. Table 3 shows the values of the heuristic parameters.

In previous studies, the COQ function, based on percentage of defective units, is assumed to be given. This paper deals with the development of an SCND-COQ model that computes the COQ for a whole SC based on internal decisions within the manufacturing plant, such as fraction defective at the manufacturing plant and error rate at inspection. No previous work has addressed how the COQ functions were obtained while taking internal operational decisions within the SC. Moreover, the proposed SCND-COQ model computes quality costs for the whole SC considering the interdependencies among business entities, whereas previous works have assumed independent COQ functions for each node of the SC.

4.2.2. Comparison among Solution Procedures. Tables 4–7 show the results for each problem size. Performance was measured by solution quality, number of evaluations of the objective function, and computational time in CPU seconds. Solution quality is characterized in two ways: (a) the average best-found profit (Avg_Profit) over 5 instances obtained by each solution procedure and (b) the average percentage deviation from the optimal solution for class I and from the best-found solution for classes II and III (Avg%dev) over the same 5 instances. The deviation at each instance is computed as [(optimal solution − Avg_Profit)/optimal solution)] × 100% for class I and as [(best-found solution − Avg_Profit)/best-found solution)] × 100% for classes II and III.

For the GS and MS procedures, the best-found profit, achieved from one of all possible SC network configurations, is the one reported. For SVRC1 and SVRC2 (greedy approach), only 1 run is performed in Stage 1 and the profit achieved by the network in Stage 2 is the one reported as the final objective value. For SSA/SGA, at each iteration, the serial logistic route with the best-found profit in 5 runs is the one added to the network in Stage 1; the profit obtained by the network in Stage 2 is the one reported.

The average number of evaluations of the objective function over 5 instances (Avg_Evals) is also considered as a performance measure. For the GS and MS procedures, the number of evaluations of the objective function is determined by the total number of times that the nonlinear solver is called while solving the problem instance. For SVRC, SSA, and SGA, the number of evaluations is computed as the sum of the number of evaluations performed by the heuristic procedures when constructing the network and the number of evaluations performed by the nonlinear solver while reoptimizing the internal continuous variables for the capacitated model.

Table 3: Heuristic parameters.

Solution procedure	Heuristic parameters				
SVRC1	$\delta = 0.1$				
SVRC2	$\delta = 0$				
SSA1	neigsize = [0.45 0.45 0.1 size (PS, 1)]	$T_o = 1 \times 10^5$	$\alpha = 0.8$	Accep/MaxTrials = 30	MaxChains = 3

SSA2	neigsize = [0.1 size (PS, 1)]
	neigsize == [0.3 size (PS, 1)]
	for 8 × 8 × 10, classes II and III
		$T_o = 1 \times 10^5$	$\alpha = 0.8$	Accep/MaxTrials = 30;
				50 for 8 × 8 × 10 class I;
				and 80 for classes II and III
					MaxChains = 2

SSA3	neigsize = [2 2 2]
	neigsize == [3 3 3]
	for 8 × 8 × 10, classes II and III
		$T_o = 1 \times 10^5$	$\alpha = 0.8$	Accep/MaxTrials = 30;
				50 for 8 × 8 × 10 class I;
				and 80 for classes II and III
					MaxChains = 2

SGA1	pm = 0.01	pc = 0.95	gen = 20	pop = 20% of **PS**	
SGA2	pm = 0.01	pc = 0.95	gen = 10	pop = 10% of **PS**	
SGA3	pm = 0.01	pc = 0.95	gen = 10	pop = 10% of **PS**	

Note: for the 35 × 20 × 35 size, SGA2 and SGA3 have gen = 25 and pop = 120 individuals. The number of runs was decreased from 5 to 3.

Finally, the average computational time (Avg_Time) is the average processing time duration in CPU seconds that is required for each solution procedure over 5 instances. The computational time considers the entire solution procedures, that is, phase I and phase II. The computer used for the computational experiments was a Sager NP8130 with an Intel i7 2720QM processor operating at 3.3 GHz, with 16 GB of memory DDR3 on an Intel HM65 chipset motherboard.

Table 4 gives computational results for the first problem size. For class I problems, it can be observed that most algorithms reached the optimal solutions in all five instances, with the exceptions being MS and SGA1. The GS (exhaustive enumeration procedure) was able to find the optimal solution for class I instances but was unable to find good solutions for classes II and III, thus, denoting a limitation of such procedure to tackle complex instances.

For class II problems, based on Avg_Profit, the SVRC2, SSA2, SSA3, SGA2, and the SGA3 reach the same average solution (best-found solution). For class III problems, the best-found solution was found by SVRC2, SSA1, SSA2, SSA3, SGA2, and SGA3. GS, MS, and SVRC1 present an Avg%dev greater than 6.6%.

Table 5 shows the results for the 5 × 4 × 6 problem size. As observed previously for class I problems, most algorithms reached the optimal solution, with the exception being SGA1 and SGA2. For class II problems, SSA2 found the best solutions on average, followed by SVRC2 and SGA2 (Avg%dev of 0.19% and 0.29%, resp.). Similarly, for class III, SSA2 found the best solutions on average, followed by SVRC2, SSA1, and SGA3. This shows that suboptimal selections of serial routes in SSA phase I may lead to a more profitable network configuration (as shown in SSA2) than the profit obtained by using a greedy approach (SVRC2).

Table 6 shows the results for the 8 × 8 × 10 problem size. For class I problems, most algorithms reached the optimal solution, with the exception being SSA3 and SGA1. For class II, SSA1 found the best solutions on average, closely followed by SVRC2, SSA2, and SGA3. For class III, SSA1 found the best solutions on average, followed by SGA1. It is noteworthy that for class III (considered difficult), the heuristic procedures (SSA1 and SGA1) outperform the enumeration-greedy approach (SVRC2) in average profit and computational time.

The results for the largest instance (35 × 20 × 35) are shown in Table 7. For class I, it was expected that SVRC2 outperformed the rest of the procedures because the optimal solution is a serial route; however, the SSA1 reached solutions with an Avg%dev of 1.64%. For class II, the best-found profits are obtained by SGA1 and SGA3 closely followed by SSA1 and SVRC2. For class III, SVRC2 provides the best found solutions followed by SGA1 and SGA2. In general, the algorithms that produce best solutions on average for large problems are SVRC2, the three GA-based algorithms, and SSA1. The computational burden required by the SVRC2 and the GA-based procedures is significantly greater than the time required by the SSA1 (2.68 hrs on average per instance for SVRC2 and 2.24 hrs on average for the SA-based procedures versus less than 4 minutes for SSA1). SSA1 is able to reach good solutions with a percentage deviation from the best-found average solution for all classes of 0.48%.

To conclude, GS and MS perform poorly, even for the smallest problem size, and require substantial computational effort. The computational time for SVRC2 is for only one run of the whole procedure and the CPU time for the remaining procedures includes the 5 runs in phase I. For realistically sized problems, the practitioner could use the SSA1 which requires less computational time than the remaining procedures to obtain good quality solutions or the GA-based procedures which require more computational effort but obtain the higher profits on average. Interestingly, the index-independent representation is not adequate for the SA-based procedure (SSA3) but improves the average profits reached by SGA3.

5. Conclusions and Future Research

The capacitated SCND-COQ model selects several business entities at each echelon of the SC and allows the modeling

Table 4: Results for problem size 3 × 2 × 3.

	GS	MS	SVRC1	SVRC2	SSA1	SSA2	SSA3	SGA1	SGA2	SGA3
Class I										
Avg_Profit	**41,806,816.26**	41,709,103.64	**41,806,816.26**	**41,806,816.26**	**41,806,816.26**	**41,806,816.26**	**41,806,816.26**	40,569,521.42	**41,806,816.26**	**41,806,816.26**
Avg%dev	**0.00**	0.26	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	2.57	**0.00**	**0.00**
Avg_Evals	**14,432,210.20**	16,986,412.80	**4,314.80**	**925.68**	**7,013.00**	**4,400.00**	**4,481.00**	4,404.00	**3,706.40**	**4,278.00**
Avg_Time	**12,266.16**	13,906.77	**5.40**	**1.09**	**3.20**	**16.89**	**31.83**	2.59	**12.59**	**20.67**
Class II										
Avg_Profit	43,977,372.68	40,721,911.06	44,986,134.32	**47,920,297.79**	47,390,930.58	**47,920,297.79**	**47,920,297.79**	47,576,092.96	**47,920,297.79**	**47,920,297.79**
Avg%dev	8.92	14.66	5.37	**0.00**	1.16	**0.00**	**0.00**	0.57	**0.00**	**0.00**
Avg_Evals	17,880,237.60	16,848,438.00	6,656.60	**1,711.56**	19,915.40	**12,811.20**	**12,864.00**	13,345.60	**9,601.00**	**8,850.00**
Avg_Time	12,414.88	11,740.01	8.13	**2.05**	9.48	**93.64**	**100.98**	8.02	**47.20**	**72.90**
Class III										
Avg_Profit	9,763,747.96	9,488,786.92	9,778,795.24	**10,475,602.91**	**10,475,602.91**	**10,475,602.91**	**10,475,602.91**	10,221,015.40	**10,475,602.91**	**10,475,602.91**
Avg%dev	6.63	6.62	8.02	**0.00**	**0.00**	**0.00**	**0.00**	3.74	**0.00**	**0.00**
Avg_Evals	9,420,031.20	19,270,506.00	5,155.60	**915.28**	**10,069.40**	**4,843.40**	**5,240.40**	7,119.80	**3,630.80**	**4,195.80**
Avg_Time	6,944.75	13,079.87	6.82	**1.42**	**3.75**	**44.34**	**59.39**	3.92	**21.77**	**36.30**

The bold font in the tables refers to the algorithms that reached best solutions.

Table 5: Results for problem size 5 × 4 × 6.

	SVRC1	SVRC2	SSA1	SSA2	SSA3	SGA1	SGA2	SGA3
Class I								
Avg_Profit	**52,844,809.95**	**52,844,809.95**	**52,844,809.95**	**52,844,809.95**	**52,844,809.95**	49,623,191.12	51,243,573.72	**52,844,809.95**
Avg%dev	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	5.58	3.08	**0.00**
Avg_Evals	**11,037.40**	**2,008.68**	**10,295.20**	**4,928.20**	**7,593.40**	8,846.60	6,951.40	**5,729.80**
Avg_Time	**26.95**	**5.61**	**5.65**	**46.68**	**44.73**	5.06	21.88	**17.27**
Class II								
Avg_Profit	118,455,380.46	125,070,837.06	124,353,125.54	**125,287,428.28**	123,051,779.22	123,204,151.47	124,955,144.86	123,927,909.74
Avg%dev	5.58	0.19	0.74	**0.00**	1.70	1.76	0.29	0.98
Avg_Evals	29,055.40	5,049.64	42,555.60	**24,460.00**	34,449.40	46,363.40	18,697.20	17,002.40
Avg_Time	74.04	17.07	14.70	**324.78**	317.41	24.52	133.09	146.33
Class III								
Avg_Profit	39,465,124.69	42,810,167.13	42,810,167.13	**42,894,361.03**	40,157,925.23	40,128,165.86	42,796,773.97	42,810,167.13
Avg%dev	8.16	0.20	0.20	**0.00**	6.95	6.23	0.23	0.20
Avg_Evals	22,111.00	5,210.44	38,800.00	**20,993.80**	17,804.60	37,709.60	13,984.20	13,701.60
Avg_Time	60.39	15.60	13.89	**220.63**	191.32	19.02	111.73	110.97

The bold font in the tables refers to the algorithms that reached best solutions.

Table 6: Results for problem size 8 × 8 × 10.

	SVRC1	SVRC2	SSA1	SSA2	SSA3	SGA1	SGA2	SGA3
Class I								
Avg_Profit	**55,003,836.49**	**55,003,836.49**	**55,003,836.49**	**55,003,836.49**	53,822,605.31	48,794,936.66	**55,003,836.49**	**55,003,836.49**
Avg%dev	**0.00**	**0.00**	**0.00**	**0.00**	2.02	11.21	**0.00**	**0.00**
Avg_Evals	**42,368.80**	**9,186.64**	**9,424.00**	**8,717.80**	7,919.80	22,529.20	**9,453.20**	**9,871.60**
Avg_Time	**149.10**	**29.28**	**3.82**	**49.93**	66.15	6.65	**83.40**	**76.31**
Class II								
Avg_Profit	210,780,559.55	234,647,930.81	**234,741,055.90**	234,647,930.81	201,791,597.82	234,518,007.14	234,524,750.57	234,647,930.81
Avg%dev	10.33	0.15	**0.11**	0.15	14.49	0.19	0.20	0.15
Avg_Evals	174,373.80	32,888.00	**98,987.40**	82,037.80	54,223.60	237,543.60	79,051.60	83,976.00
Avg_Time	791.55	115.83	**48.11**	1,487.61	1,027.24	107.95	1,239.64	1,575.92
Class III								
Avg_Profit	77,799,129.47	88,500,880.62	**89,252,162.29**	88,444,907.51	76,311,740.30	88,883,338.00	88,297,371.17	88,444,907.51
Avg%dev	13.47	0.96	**0.16**	1.03	15.29	0.55	1.18	1.03
Avg_Evals	166,321.40	35,431.92	**96,052.80**	88,296.60	63,744.00	245,791.60	87,427.60	87,802.00
Avg_Time	528.86	126.18	**35.45**	1,705.61	1,197.31	102.84	1,279.54	1,212.89

The bold font in the tables refers to the algorithms that reached best solutions.

Table 7: Results for problem size 35 × 20 × 35.

	SVRC1	SVRC2	SSA1	SSA2	SSA3	SGA1	SGA2	SGA3
Class I								
Avg_Profit	58,064,475.00	**59,724,004.15**	58,835,952.34	58,247,593.08	38,238,724.57	55,730,804.56	57,808,706.54	57,087,459.85
Avg%dev	2.92	**0.16**	1.64	2.61	36.44	6.92	3.45	4.55
Avg_Evals	1,949,047.20	**4,509,049.80**	48,971.00	24,684.00	19,717.20	1,563,423.40	52,121.60	66,082.00
Avg_Time	6,733.20	**3,663.56**	32.22	165.15	120.47	554.06	849.38	1,667.85
Class II								
Avg_Profit	867,540,166.20	981,487,545.86	981,963,721.23	974,060,070.59	565,928,537.54	**984,989,935.75**	978,690,227.10	983,533,295.26
Avg%dev	12.11	0.73	0.7	1.47	42.52	**0.38**	1.01	0.52
Avg_Evals	18,860,758.20	19,184,769.80	829,529.00	276,866.20	148,021.40	**35,715,505.40**	627,077.60	637,319.40
Avg_Time	68,691.96	14,677.11	337.97	5,037.87	2,320.08	**13,272.93**	16,920.93	17,572.59
Class III								
Avg_Profit	266,985,536.25	**309,497,478.07**	303,366,283.17	305,752,432.09	216,893,443.36	308,695,294.99	307,980,118.48	305,808,477.36
Avg%dev	13.75	**0.07**	2.07	1.28	29.74	0.34	0.56	1.26
Avg_Evals	14,534,583.40	**14,115,188.60**	464,191.00	190,929.40	158,399.80	19,569,407.20	434,557.00	212,050.20
Avg_Time	52,247.29	**10,590.83**	332.6	3,632.26	2,807.67	7,096.90	8,490.13	6,104.69

The bold font in the tables refers to the algorithms that reached best solutions.

of business entities with limited capacity. Using the SCND-COQ model can assist firms in improving their profitability and quality simultaneously.

Noteworthy, in the classes where SGA2, SGA3, SSA1, and SSA2 achieve better profits than SVRC2, the average percentage difference between the solutions was less than 0.73%. This may suggest that these methods could be close to the global solution for the classes where the optimal is unknown. As the problem size continues to increase, using SVRC2 will entail considerable computational burden. Thus, the use of GA-based procedures and SSA1 is more attractive for large problems and these methods find solutions close to the best-found solutions. The computational results demonstrate that the state and individual representation in the simulated annealing and the genetic algorithm, respectively, has a significant impact on the solution quality.

A possibility for future research would be to develop additional algorithms specifically designed for the capacitated model; metaheuristics such as SA, the GA, Tabu search, and scatter search could be explored. In addition, a multiproduct and multiobjective capacitated supply chain network design problem including COQ could be studied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was funded in-part by the University of Texas at San Antonio, Office of the Vice President for Research.

References

[1] L. Li and J. N. Warfield, "Perspectives on quality coordination and assurance in global supply chains," *International Journal of Production Research*, vol. 49, no. 1, pp. 1–4, 2011.

[2] K. Das, "A quality integrated strategic level global supply chain model," *International Journal of Production Research*, vol. 49, no. 1, pp. 5–31, 2011.

[3] S. K. Srivastava, "Towards estimating cost of quality in supply chains," *Total Quality Management and Business Excellence*, vol. 19, no. 3, pp. 193–208, 2008.

[4] A. Ramudhin, C. Alzaman, and A. A. Bulgak, "Incorporating the cost of quality in supply chain design," *Journal of Quality in Maintenance Engineering*, vol. 14, no. 1, pp. 71–86, 2008.

[5] C. Alzaman, A. A. Bulgak, and A. Ramudhin, "Quality in operational supply chain networks: an aerospace case study," *International Journal of Operational Research*, vol. 9, no. 4, pp. 426–442, 2010.

[6] Juran, *Quality Control Handbook*, McGraw-Hill, New York, NY, USA, 1951.

[7] K. K. Castillo-Villar, N. R. Smith, and J. L. Simonton, "A model for supply chain design considering the cost of quality," *Applied Mathematical Modelling*, vol. 36, no. 12, pp. 5920–5935, 2012.

[8] K. K. Castillo-Villar, N. R. Smith, and J. L. Simonton, "The impact of the cost of quality on serial supply chain network design," *International Journal of Production Research*, vol. 50, no. 19, pp. 5544–5566, 2012.

[9] D. P. Ballou and H. L. Pazer, "The impact of inspector fallibility on the inspection policy in serial production systems," *Management Science*, vol. 28, no. 4, pp. 387–399, 1982.

[10] J. T. Godfrey and W. R. Pasewark, "Controlling quality costs," *Management Accounting*, vol. 69, no. 9, pp. 48–51, 1988.

[11] W.-N. Pi and C. Low, "Supplier evaluation and selection using Taguchi loss functions," *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 1-2, pp. 155–160, 2005.

[12] M. R. Bussieck and A. Pruessner, "Mixed-integer nonlinear programming," *SIAG/OPT Newsletter: Views & News*, vol. 14, no. 1, pp. 19–22, 2003.

[13] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.

[14] M. Resende and C. Ribeiro, "Greedy randomized adaptive search procedures," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds., pp. 219–249, Springer, New York, NY, USA, 2003.

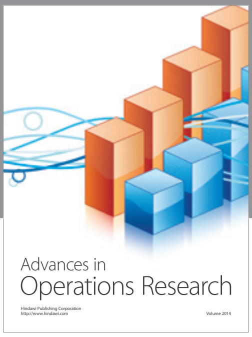
[15] P. Festa and M. G. C. Resende, "An annotated bibliography of GRASP-part I: algorithms," *International Transactions in Operational Research*, vol. 16, no. 1, pp. 1–24, 2009.

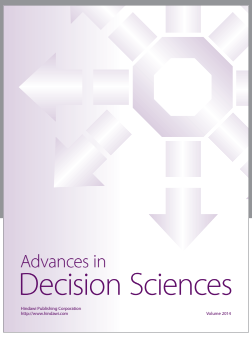
[16] R. W. Eglese, "Simulated annealing: a tool for operational research," *European Journal of Operational Research*, vol. 46, no. 3, pp. 271–281, 1990.

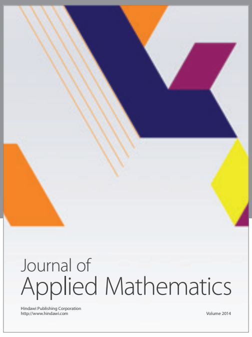
[17] E. Aarts, J. H. M. Korst, and P. J. M. van Laarhoven, "Local search in combinatorial optimization," in *Local Search in Combinatorial Optimization*, John Wiley & Sons, England, 1997.

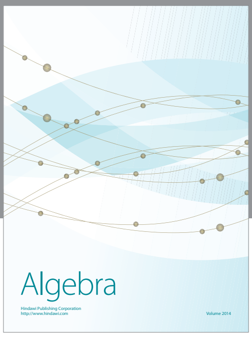
[18] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.

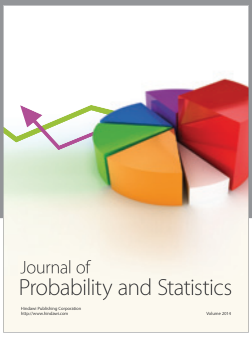
[19] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.


Advances in
Operations Research

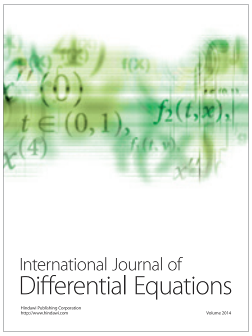

Advances in
Decision Sciences


Journal of
Applied Mathematics

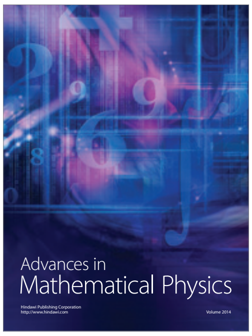

Algebra

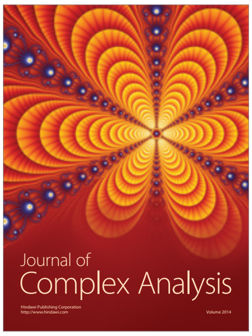

Journal of
Probability and Statistics

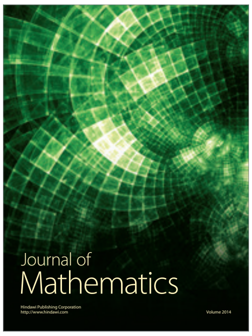
The Scientific
World Journal

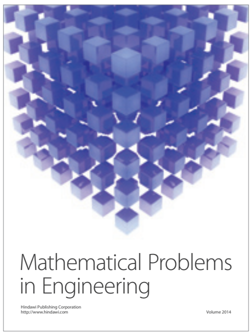

International Journal of
Differential Equations

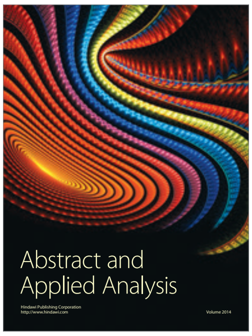
International Journal of
Combinatorics

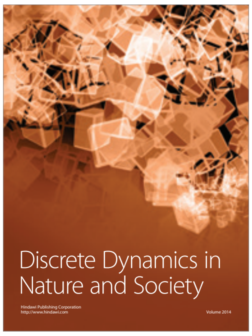

Advances in
Mathematical Physics

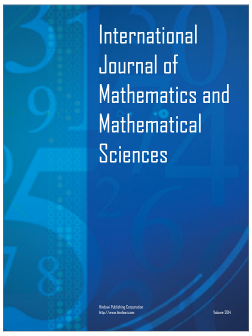

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

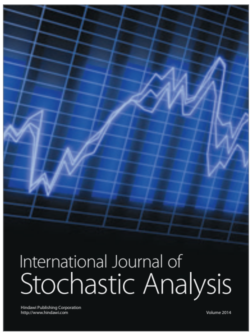

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization