The Effect of Individual Representation on the Performance of a Genetic Algorithm applied to a Supply Chain Network Design Problem

Krystel K. Castillo-Villar[1,*], José F. Herbert-Acero[2]

[1*] *Department of Mechanical Engineering, The University of Texas at San Antonio, One UTSA Circle, San Antonio, Texas 78249, USA.*
Krystel.Castillo@utsa.edu

[2] *Department of Physics, Tecnológico de Monterrey, Campus Monterrey. E. Garza Sada 2501 Sur, C.P. 64849, Monterrey, N.L., México.*
jf.herbert.phd.mty@itesm.mx

***Abstract*— This paper presents a comparison of a variety of individual representations in a procedure based on the Genetic Algorithm for a capacitated model for supply chain network design (SCND) that considers the cost of quality (COQ) as well as the traditional manufacturing and distribution costs. The model is known as the SCND-COQ and can be used at a strategic planning level to maximize profit subject to meeting an overall quality level. The SCND-COQ model internally computes quality costs for the whole supply chain considering the interdependencies among business entities, whereas previous works have assumed exogenously and independently given COQ functions (nonlinear functions). The 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 and the difficulty of solving non-convex nonlinear problems. The aim is to maximize the profit of the supply chain subject to: demand, capacity, flow balance, and overall quality level of the final product constraints. We provide a solution method based on the genetic algorithm (GA) for solving instances of practical and realistic size. We compare the performance of the GA with several individual representations and a greedy constructive heuristic procedure. Managerial insights for practitioners are provided and the results of computational testing are reported.**

Keywords*—supply chain management, supply chain network design, Cost of Quality, genetic algorithm, genotype.*

1. Introduction

This paper addresses the problem of supply chain network design (SCND). SCND involves selecting the business entities to include in the Supply Chain (SC). Cost of Quality (COQ) is a measurement system that translates poor quality into monetary terms.

Although COQ has been applied mostly within companies, COQ should be applied as an external measure to integrate these costs into SCND modeling. Several studies have provided models to ensure quality in multi-stage SC design [1]. Srivastava [2], 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. [3] also focus on integrating COQ in the SC. Their seminal study presents a mathematical formulation that integrates given 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. More recently, Alzaman et al. [4] propose a model with an *n* level bill of materials that incorporates a known COQ quadratic function based on a defect ratio at all SC nodes. The COQ function is known and based on Juran's original model [5]. Das [6] proposes a multi-stage global SC mathematical model for preventing recall risks.

In previous studies, functions for the total COQ based on percentage of defective units are assumed to be given. This paper proposes a model that computes the COQ for a whole SC based on interdependencies among business entities and internal decisions within the manufacturing plant such as fraction defective at the manufacturing process and error rate at inspection. The only previous works that have addressed how the COQ curves can be computed by taking internal operational decisions within the SC are Castillo et al. [7]-[8], the model was named SC-COQ model. Two solution procedures were developed to solve the SC-COQ, one based on a local search algorithm (simulated annealing) and the other based on a population algorithm (the genetic algorithm) [8]. The problem addressed here is 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, capacities at the business entities are considered in this model which is known as capacitated SCND-COQ model [9]. The capacitated SCND-COQ model is a more comprehensive model and a more challenging-to-solve problem. The number of constraints and decision variables increases exponentially as the problem size increases as well as the network possible configurations. Thus, the interrelationships among business entities become more

International Journal of Supply Chain Management
IJSCM, ISSN: 2050-7399 (Online), 2051-3771 (Print)
Copyright © ExcelingTech Pub, UK (http://excelingtech.co.uk/)

complex. The purpose of this paper is to compare and quantify the effect of different representations of a heuristic procedure based on the genetic algorithms and a global optimization solver.

2. The Capacitated SCND-COQ Model

The capacitated SCND-COQ model differs from the previous serial 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. This requires a pooled fraction defective from the selected suppliers to be computed. The main modeling assumptions are the following: 1) A consumer goods SC, consisting of three echelons: suppliers, manufacturers, and retailers, and a single product is modeled. 2) The overall quality level, *QL*, is sufficient to represent quality. 3) Suppliers and retailers are external to the plant and under separate management. 4) A 100% inspection is performed at the end of the manufacturing process to check product conformance. Inspection error is of type II. Type II error involves labeling a defective item as good and type I error involves classifying a good item as defective. Type I error is not considered in this model because is not detrimental to customer satisfaction. 5) All defective products are returned by customers and incur external failure costs. 6) Customer demand at each retailer (Dem_k) is known for the study period and retailers' capacity is not considered. 7) Suppliers and manufacturing plants have finite capacity.

The following sets are defined: *I*, set of suppliers ($i \in I$); *J*, set of manufacturing plants ($j \in J$); *K*, set of retailers ($k \in K$). The model constants are: 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; Ys_i, fraction defective at supplier $i \in I$; Yr_k, 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 $j \in J$, and $\overline{Ys}_j = \sum_i Ys_i w_{ij} / \sum_i w_{ij}$, pooled fraction defective of all suppliers shipping products to manufacturing plant $j \in J$. The model variables are: yI_j, inspection error rate at the output of manufacturing plant $j \in J$; yp_j, fraction defective at manufacturing plant $j \in J$; 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 opened, zero otherwise; w_{ij}^{sp}, number of components shipped from supplier $i \in I$ to manufacturing plant $j \in J$; w_{jk}^{pr}, number of components shipped from manufacturing plant $j \in J$ to retailer $k \in K$.

The problem is to maximize profit:

$$\sum_{j\in J}\sum_{k\in K} w_{jk}^{pr} p_{jk} P_j R_k - COQ(w_{ij}^{sp}, w_{jk}^{pr}, yI_j, yp_j, Z_i, P_j, R_k) - \sum_{i\in I}\sum_{j\in J} w_{ij}^{sp} Pc_{ij} Z_i P_j - \sum_{i\in I}\sum_{j\in J} w_{ij}^{sp} Po_{ij} Z_i P_j - \sum_{i\in I}\sum_{j\in J} w_{ij}^{sp} u_{ij} Z_i P_j - \sum_{j\in J}\sum_{k\in K} w_{jk}^{pr} l_{jk} P_j R_k - \sum_{j\in J} F_j P_j \tag{1}$$

subject to:

$$\sum_j w_{jk}^{pr} \le Dem_k R_k; \forall k \in K \tag{2}$$

$$\sum_i w_{ij}^{sp} = \sum_k w_{jk}^{pr}; \forall j \in J \tag{3}$$

$$\sum_i w_{ij}^{sp} \le Cap_j P_j; \forall j \in J \tag{4}$$

$$\sum_j w_{ij}^{sp} \le Cap_i Z_i; \forall i \in I \tag{5}$$

$$QL_k \ge l\ R_k; \forall k \in K \tag{6}$$

$$0 \le yI_j \le 1; \forall j \in J \tag{7}$$

$$0 \le yp_j \le 1; \forall j \in J \tag{8}$$

$$Z_i, P_j, R_k \in \{0,1\}; \forall i \in I, \forall j \in J, \forall k \in K \tag{9}$$

The first term of (1) is the sales revenue. The second term represents the total COQ for the network. A detailed explanation of the COQ term can be found in [8]. The parameters for the COQ function are shown in the Appendix. 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 processing cost for the components from selected supplier(s) 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) and the seventh term determines the fixed cost for opening plants. Constraints (2) enforce that demand at retailers is not exceeded. Constraints (3) ensure that the number of components shipped from suppliers to manufacturing plants equals the number of items shipped from manufacturing plants to retailers. Constraints (4) ensure that the plant capacity (in units) is not exceeded. Constraints (5) enforce that the exit capacity (in units) at the suppliers is not exceeded. Constraints (6) enforce the desired quality level. Constraints (7)-(9) 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. Mixed integer programming (MIP) and nonlinear problems (NLP) are known as NP-complete problems [10]; thus, solving MINLP problems can be a challenging task. Two heuristic procedures were developed: one based on the genetic algorithm with a global optimization solver and the second is a greedy constructive procedure with a global optimization solver.

3.1 Genetic Algorithm-based procedure

A genetic algorithm (GA) for solving the capacitated SCND-COQ model is described in this section. The GA procedure is based on the serial model [7]-[8]. 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 involving 5 suppliers, 3 plants, and 5 retailers is 75, which is considerably less than the number of possible network configurations. For instance, in a problem with 5 suppliers, 3 plants, and 5 retailers, the total number of possible configurations is 6,727.

The heuristic procedure based on the serial model can be divided into two stages and employs the idea that a network can be constructed by choosing a serial logistic route with the highest profit when sending the maximum possible flow of items through that route, adding that serial route to the network, updating the remaining capacities, and repeating the process. In order to find the best serial routes, at each iteration, the virtual gene Genetic Algorithm [11] was modified to address this problem. The heuristic procedure serves to construct a feasible network and to determine flows (Stage I). The GA finds the combination of business entities and a nonlinear solver, FMINCON available in MATLAB®, was used for finding the internal continuous variables (yI_j and yp_j) that minimize the COQ. In Stage II, we optimize the internal decision variables for the feasible network and flows found in Stage I by using the capacitated SCND-COQ model formulation (Stage I uses the serial model). The profit achieved by this network is taken as the best-found solution for the capacitated model. The procedure for the proposed solution method based on the genetic algorithm is illustrated as follows:

Stage I:

1.1 Create a list of all possible serial routes.

1.2 Compute the quality level attained by each serial route, eliminate the routes that do not meet the minimum level in Eq.(6), and save a result 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: $\min\{Cap_i, Cap_j, Dem_k\}$.

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 into account the fixed cost for opening a plant 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: non-positive 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 the GA-based solution procedure for the serial SC-COQ model [8]. The GA-based procedure decides the binary variables (supplier, plant and retailer) while 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®

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).

1.5.3. 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 re-optimized. It is worth noting that the network and flows found in Stage I are not modified.

2.1 Re-optimize the internal continuous variables associated with the opened manufacturing plants by using the GlobalSearch algorithm in MATLAB®.

The solution procedure presented here combines GA (Stage I) and the GlobalSearch algorithm (Stage II), it was named SGA and was implemented using MATLAB®. In previous experimentation, it was observed that for some problems relationships between entities (supplier-plant, plant-retailer, or supplier-retailer) produced a significant overall increment on the population fitness, leading the algorithm to find the best solutions. Therefore, five different representations (genotype and phenotype representations for individuals) were developed and compared: SGA_SPR, SGA_SP, SGA_SR, SGA_PR, and SGA_Ind. These representations were chosen in order to explore the non-linear relationships among entities and the impact of these representations on SGA performance.

In the SGA_SPR representation all entities are linked. First, a matrix of possible solutions (**PS** matrix) that satisfy the minimum quality level is obtained. The **PS** matrix is structured in rows containing feasible combinations of entities. The combinations are generated starting at suppliers. See Figure 1.

$$\begin{bmatrix} \textit{Supplier} & \textit{Plant} & \textit{Retailer} \\ 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ \ldots & \ldots & \ldots \end{bmatrix}$$

Figure 1. Mathematical Representation of the Search Space: Feasible Serial Routes.

This representation consists of linked entities, i.e., rows of the **PS** matrix or feasible serial routes. A chromosome (or genotype), represented as a vector containing binary digits (base 2), is constructed. The Minimum Amount of Bits (*MAB*) required to completely represent all the rows of the **PS** matrix is obtained using the relation shown in Eq. (10)

$$MAB = ceil\left(\frac{\log(Z)}{\log(2)}\right) \tag{10}$$

where the ceil function rounds the argument towards infinity and Z = SPR, the size of the **PS** matrix (number of rows). The genotype is interpreted as a real number (phenotype) which is rounded to obtain a feasible solution. For example, the chromosome [00000…] represents the first row of **PS** and [11111…] represents the last row of **PS**. The main issue with this representation resides on the genetic linkage problem, which refers to the issue of finding and creating linkages between important genes in the chromosome in order to find the best solutions. As the size of the problem increases, the number of bits used to describe the genotype increases, thus incrementing the linkage issue [12]. Therefore, it is of interest to find individual representations that helps the GA to find interdependences of the most important bits forming the genotype.

In a second representation, the SGA_SP representation, the supplier and plant are linked and the retailer is free. This representation considers a genotype containing two segments of binary digits. The first segment contains information about the connections between suppliers and plants and the last segment contains information about the retailers. Adding segments to the genotype representation provides the algorithm with an extra degree of freedom to explore the search space. Another benefit is the ability to set different crossover and mutation probabilities between segments. The first genotype segment is identical to the first representation (*MAB*) while the second segment contains MAB_R bits given by Eq. (11) with Z = #R, where #R is the number of retailers considered in the optimization problem. Note that MAB_R must be significantly lower than *MAB*. An example of the SGA_SP representation is: $Individual \rightarrow [MAB \quad MAB_R] = [11010 \quad 101]$

The base of the segments can be increased to reduce the number of bits used but mutation may have to be increased to reduce cardinality issues. In this paper, binary chromosomes are used. In the case that the individual is not on the **PS** matrix, a penalty is performed by giving a value of zero to the objective function. The computational cost used to evaluate penalized individuals is almost negligible.

In the SGA_SR representation the supplier and retailer are linked and the plant is free. The first segment contains information about the chosen supplier and retailer while the second segment contains information about the chosen plant. The number of bits required to represent the second segment MAB_P is given by Eq. (11) with Z = #P, where #P is the amount of plants considered in the optimization problem. Again MAB_P should be significantly lower than *MAB*.

In the SGA_PR representation the plant and retailer are linked and the supplier is free. Following the same reasoning as before, the MAB_S can be computed using Eq. (11) with Z = #S, where #S is the number of suppliers considered.

The SGA_Ind representation works with independent entities. Here, the genotype is represented by one vector containing three segments. Each segment contains the minimum amount of bits necessary to represent the amount of entities of each kind. The algorithm earns three degrees of freedom which result on a more flexible way to explore the solution space.

3.2 Greedy 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 profit.

The same two stages discussed before are applied for the greedy constructive procedure. The difference is that Stage I uses either the GA or the Greedy approach to select a serial route to be added to the network. Stage II remains the same. The whole procedure, that is, the greedy constructive approach for the serial routes at each iteration and the GlobalSearch for optimizing the internal decision variables of the constructed network is named serial greedy constructive procedure (SGreedy) and was implemented using MATLAB®.

4. Experimental Study

4.1 Test Problems

The data in test problems were generated randomly from a uniform distribution between the low and high levels documented in Table 1. The minimum required quality level (l) is fixed at 0.85 for all test instances. The interested reader can obtain the test problems from the authors.

Table 1. Ranges of the parameters used to generate realistic 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})	$1*Bv_j$	$5*Bv_j$
Variable cost for appraisal/inspection activities (Bv_j)	5	5
Price per 'sold as defective' items (P^*_{jk})	$1/4p_{jk}$	$3/4p_{jk}$
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

Moreover, three classes of instances were developed. In Class I, 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 1 and performing the following modifications. Once the parameter values are generated from uniform distributions with ranges as shown in Table 1, some of the parameters associated with the entities in the optimal serial route (Z* will denote the optimal supplier, P* the optimal plant, and R* the optimal retailer) are modified in order to force the optimal solution to be a specific serial route. A ratio (β) set to 0.6 will be used to adjust the parameters of Z*, P*, and R* so that the cost parameters in these entities are significantly lower than the rest of the values of the cost parameters of other entities in order 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 1 and multiplying by β. In this way, the costs are (1-β)% less than the rest of the cost parameters. The fraction defective at the supplier and retailer are also decreased by (1-β)% 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+β). 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 1. The resulting instances are verified by enumerating and evaluating all the possible serial routes to make sure that the serial route with Z*, P* and R* is the optimal solution; otherwise, the instances that do not have the Z*-P*-R* route as optimal solution are not used for testing.

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 1. However, the price for Class II problems ranges from 1.9 to 2. The sum of the randomly generated retailer capacities Dem_k is multiplied by 1.1 and divided between 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. For Class III problem instances, the parameter values are randomly generated from uniform distributions with ranges as shown in Table 1; thus, the optimal network is unknown.

4.2 Parameters' Tuning and Effect of Representations

A statistical design of experiment was conducted as a preliminary numerical study to explore the impact of the representation on the performance of the SGA for problems of considerably size (35 suppliers, 20 manufacturing plants and 35 retailers) and to provide guidance about the values of SGA's parameters. The SGA parameters are: initial population (*pop*), number of generations (*gen*), probability of mutation (*pm*), and probability of crossover (*pc*). Additional to the algorithm's parameters, the five different representations were considered in the design. The selected design is a general full factorial with two blocks (each block solved a different instance, that is, block one solved an instance from Class II and the second block solved an instance from Class III). The factors and levels are as follows. Representations (5 levels: SGA_SPR, SGA_SP, SGA_PR, SGA_SR, and SGA_Ind), *pop* as a percentage of the **PS** matrix (3 levels: 0.2%, 0.5%, and 0.8%), *gen* (3 levels: 5, 15, and 25), *pm* (3 levels: 0.02, 0.1, and 0.4), and *pc* (3 levels: 0.70, 0.85, and 0.98). In total 810 runs were generated, that is, a 35×20×35 size problem was repeatedly solved with different combination of factors' levels. Model adequacy checking on residuals did not show issues with the normality test and the constant variance assumption. The Analysis of Variance (ANOVA) shown in Table 2 indicates that the blocks, representations, initial population and number of generations are statistically significant while the probability of mutation and crossover are not statistically significant with α=0.1. Taking as a response the CPU time that takes to solve the instance, the representations are not significant but the representations are significant when taking the number of evaluations and the total COQ as responses.

Table 2. ANOVA for profit as response.

Analysis of Variance for Profit, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Blocks	1	9.86237E+19	9.86237E+19	9.86237E+19	4500355.17	0.000
Rep	4	6.34825E+14	6.34825E+14	1.58706E+14	7.24	0.000
Pop	2	1.95248E+15	1.95248E+15	9.76238E+14	44.55	0.000
Gen	2	2.60941E+15	2.60941E+15	1.30471E+15	59.54	0.000
Mut	2	2.63678E+13	2.63678E+13	1.31839E+13	0.60	0.548
Cross	2	9.62946E+13	9.62946E+13	4.81473E+13	2.20	0.112
Rep*Pop	8	2.05797E+14	2.05797E+14	2.57247E+13	1.17	0.312
Rep*Gen	8	1.24414E+14	1.24414E+14	1.55518E+13	0.71	0.683
Rep*Mut	8	2.52916E+14	2.52916E+14	3.16145E+13	1.44	0.175
Rep*Cross	8	2.83328E+14	2.83328E+14	3.54159E+13	1.62	0.116
Pop*Gen	4	1.13861E+15	1.13861E+15	2.84653E+14	12.99	0.000
Pop*Mut	4	8.65987E+13	8.65987E+13	2.16497E+13	0.99	0.413
Pop*Cross	4	1.08889E+14	1.08889E+14	2.72223E+13	1.24	0.292
Gen*Mut	4	5.82533E+13	5.82533E+13	1.45633E+13	0.66	0.617
Gen*Cross	4	8.52757E+13	8.52757E+13	2.13189E+13	0.97	0.422
Mut*Cross	4	2.56476E+14	2.56476E+14	6.41191E+13	2.93	0.020
Error	740	1.62168E+16	1.62168E+16	2.19147E+13		
Total	809	9.86479E+19				

S = 4681309   R-Sq = 99.98%   R-Sq(adj) = 99.98%

The main effect plots for the overall profit shown in Figure 2 shows that the representations that yield higher profit are SGA_SP, SGA_PR, and SGA_Ind. The SGA_SPR and SGA_SP representations have the worst performance.

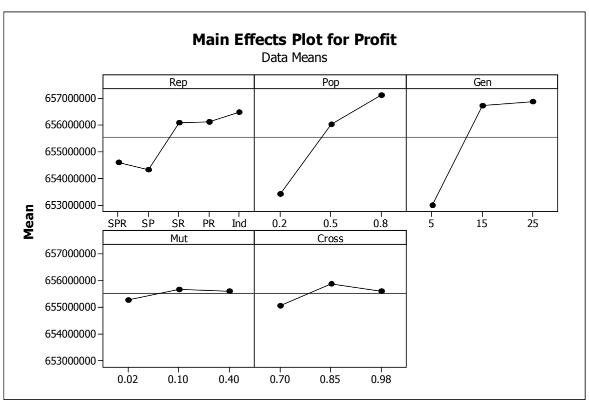

Figure 2. Main Effect Plot for Profit as response.

Since the different representations are statistically significant in the ANOVA, a computational study is conducted by solving a variety test instances from different classes. The algorithm's parameters were fixed based on the main effect plot. The effectiveness of the crossover probability dictates the rate of convergence, while the mutation probability helps the GA to avoid a premature convergence and helps it to escape from local solution. In general, the mutation probability should be low and the crossover high. Since there is no statistical evidence that the crossover and mutation probabilities increase the profit, the mutation was fixed in the lowest level and the crossover in 0.95. Regarding the number of generations and initial population, it can be observed in Figure 2 that the higher the values of the *pop* and *gen*, the greater the profit. However, a trade-off between the computation time and the values for *pop* and *gen* exists; thus, the *pop* was fixed in 0.5% (approximately 122 individuals for the size of the problem tested) and the *gen* was fixed in 15. It is recommended to adjust the *pop* and *gen* depending on the size of the problem while the crossover and mutation probabilities may remain the same regardless of the size of the problem.

In order to compare the performance of the various representations against optimal solutions, we randomly generated five instances from specially constructed instances (Class I described before) where the optimal solution is known in advance. Additionally, five instances from each class where the optimal solution is not known (Class II and III) were randomly generated. The pool of test instances consists of 45 problems for a 35×20×35 problem size. This problem size has 145 constraints and 1,530 decision variables.

5. Computational Results

A comparison of the SGA representations and SGreedy relative to each other was performed. 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 SGreedy, the serial route with the highest unit profit from all possible serial routes is the one added to the network, at each iteration. For SGA, at each iteration, the serial logistic route with the best-found profit in 3 runs is the one added to the network. The final network configuration is evaluated in the capacitated model and the re-optimization of the internal continuous variables is performed; the profit obtained by this network 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 SGreedy and SGA procedures, the number of evaluations is computed as the sum of the number of evaluations performed by the heuristic procedure when constructing the network and the number of evaluations performed by the GlobalSearch to re-optimize the internal continuous variables for the capacitated model.

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 procedure, i.e., Stage I and Stage II (including the multiple runs performed in the case of the SGA). The computer used for the computational experiments was a Sager NP8130 with Intel® i7^{TM} 2720QM operating at 3.3 GHz, with 16 GB of memory DDR3 on an Intel HM65 chipset motherboard. Table 3 shows the results.

As expected, the SGreedy method outperforms the GA-based procedures for Class I problems because the optimal solution is a single route. However, it is interesting to note that the maximum deviation is 6.44% which is reasonable due to the size and problem's complexity. The SGA_PR representation yields the best performance among the representations. The SGreedy performs approximately 68 times more evaluations than the SGAs and it requires 1.72 times more computational effort. For Class II problems, the SGA_PR has the smallest deviation from best found solutions. Noteworthy, for difficult problem classes with network solutions such as Class II and Class III, the maximum average deviation from the best found solution is 1% and 1.67%, respectively. For Class III instances, the SGreedy yields the best found profit, follow by SGA_PR and SGA_Ind.

6. Managerial Implications

The computational experience confirms the observations from the design of experiments. In section 4.2 was noticed that the best solutions where obtained with the SGA_PR, SGA_Ind and SGA_SR representations. Linking all the business entities does not help the algorithm in making an efficient search of the solution space and while linking the plants and retailers increases the profit achieved, linking the supplier and plant does not perform well. Based on the results form section 5, for all test problems, the algorithms can be ranked based on the average profit as follows: SGA_PR, SGreedy, SGA_Ind, SGA_SR, SGA_SP, SGA_SPR.

These results indicate (1) that increasing the flexibility of the representation improves the performance of the genetic algorithm for a capacitated supply chain network design with quality costs and constraints and (2) that relationships between contiguous entities have a measurable impact on the profit attained by the SGA procedure. The strongest relationship is the manufacturing plant and retailer (downstream of the supply chain); thus, a very good solution may depend on finding good combinations of plants and retailers. The practitioner can use the SCND-COQ model and the SGA_PR solution procedure to find a supply chain network design that maximizes overall profit while maintaining the highest possible quality of the final product at minimum Cost of Quality.

Table 3. Results for problem size 35×20×35.

		SGreedy	SGA_SPR	SGA_SP	SGA_ SR	SGA_PR	SGA_Ind
Class I	Avg_Profit	**59,724,004.15**	57,808,706.54	58,062,379.54	55,933,398.30	58,524,862.65	57,087,459.85
	Avg%dev	**0.16**	3.45	2.93	6.44	2.12	4.55
	Avg_Evals	**4,509,049.80**	52,121.60	60,285.00	62,068.80	64,705.80	66,082.00
	Avg_Time	**3,663.56**	849.38	1,162.19	2,128.04	1,262.12	1,667.85
Class II	Avg_Profit	981,487,545.86	978,690,227.10	980,378,709.44	984,544,155.94	**986,013,996.63**	983,533,295.26
	Avg%dev	0.73	1.00	0.84	0.42	**0.26**	0.51
	Avg_Evals	19,184,769.80	627,077.60	665,050.60	695,397.60	**680,551.00**	637,319.40
	Avg_Time	14,677.11	16,920.93	23,014.04	23,237.46	**23,096.04**	17,572.59
Class III	Avg_Profit	**309,497,478.07**	305,423,869.33	305,647,671.16	307,016,925.68	308,000,850.95	307,773,417.99
	Avg%dev	**0.11**	1.41	1.34	0.90	0.59	0.67
	Avg_Evals	**14,115,188.60**	380,910.20	405,316.80	405,361.20	395,740.00	376,995.40
	Avg_Time	**10,590.83**	10,372.47	14,684.33	14,917.53	9,907.69	10,860.13

7. Conclusions and Future Research

This paper presented an application of the genetic algorithm to a capacitated SCND problem. Several individual representations were tested and the top performers were identified. The test results indicate that the individual representation has a measurable effect on the performance of the SGA. The proposed solution procedure found good solutions for large problems. Using the SCND-COQ model and the solution procedure based on the genetic algorithm (SGA) allows the modeling of business entities with limited capacity and can assist organizations in improving their profitability and quality simultaneously when designing a supply chain network at a strategic level.

Future research involves the development of heuristic solution methods that are not based on adding serial supply chain lines to the network at each iteration but in optimizing the network flows directly. The heuristics could be based on metaheuristic procedures such as Genetic Algorithms, GRASP, Scatter Search, and Simulated Annealing, among others. Another future work includes extending the model to include more levels in the supply chain. This would involve extending the COQ modeling to include these additional echelons.

References

[1] Li, G. e. L., & Warfield, J. N., "*Perspectives on quality coordination and assurance in global supply chains"*, International Journal of Production Research, Vol 49, No. 1, pp. 1-4, 2011.

[2] Srivastava, S., "*Towards estimating Cost of Quality in supply chains*", Total Quality Management & Business Excellence, Vol 19, No. 3, pp. 193-208, 2008.

[3] Ramudhin, A., Alzaman, C., & Bulgak, A., "*Incorporating the cost of quality in supply chain design"*, Journal of Quality in Maintenance Engineering, Vol 14, No. 1, pp. 71-86, 2008.

[4] Alzaman, C., Bulgak, A., & Ramudhin, A., *"Quality in operational supply chain networks: an aerospace case study"*, International Journal of Operational Research, Vol 9, No. 4, pp. 426-442, 2010.

[5] Juran, *Quality Control Handbook* (1st ed. ed.). New York: McGraw-Hill, 1951.

[6] Das, K., "*A quality integrated strategic level global supply chain model*", International Journal of Production Research, Vol 49, No. 1, pp. 5-3, 2011.

[7] Castillo-Villar, K. K., Smith, N. R., & Simonton, J. L., *"A Model for Supply Chain Design Considering the Cost of Quality",* Applied Mathematical Modelling, Vol 36, No. 12, pp. 5920-5935, 2012a.

[8] Castillo-Villar, K. K., Smith, N. R., & Simonton, J. L., "*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, 2012b.

[9] Castillo-Villar, Krystel K., "*The Impact of the Cost of Quality on Supply Chain Network Design*", PhD Dissertation, Tecnológico de Monterrey, Monterrey, México, 2012.

[10] Bussieck, M. R., & Pruessner, A., *"Mixed-integer nonlinear programming"*, SIAG/OPT Newsletter: Views & News, Vol 14, No. 1, pp. 19-22, 2003.

[11] Valenzuela-Rendón, M., "*The virtual gene genetic algorithm*". In: E. Cantu-Paz, et al., eds. Genetic and evolutionary computation (GECCO 2003), Lecture Notes in Computer Sciences. Berlin: Springer, pp. 1457–1468, 2003.

[12] Goldberg, D. E., *"Genetic algorithms in search, optimization, and machine learning. Reading"*, MA: Addison-Wesley, 1989.

Appendix

Parameters for the COQ function:

Af_j: fixed cost for prevention activities at manufacturing plant $j \in J$.

Av_i: variable cost for prevention activities implemented by supplier $i \in I$.

Av_j: 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 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 $j \in J$.

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

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