Assessment of supply chain performance in an assembly company: evaluation of evolutionary algorithms.

Mario Pena
University of Cuenca

Juan Carlos Llivisaca
University of Cuenca

Josselin Orellana
University of Cuenca

Assessment of supply chain performance in an assembly company: evaluation of evolutionary algorithms

Josselin Orellana[1 [0000−0003−0164−1355]], Mario Peña[2,3 [0000−0002−3986−7707]], and Juan Llivisaca[2 [0000−0003−2154−3277]]

[1] University of Cuenca, Cuenca, Ecuador
{jimena.orellana96, mario.penao, juan.llivisaca}@ucuenca.edu.ec
[2] Department of Applied Chemistry and Systems of Production, Faculty of Chemical Sciences, University of Cuenca, Cuenca, Ecuador
[3] Research Department, University of Cuenca, Cuenca, Ecuador

Abstract. In current globalized markets, companies no longer compete with each other. They now compete with the supply chains (SC) to which they belong. SC optimization allows an efficient and effective management of resources. In many cases, optimization goals can conflict with one another. Therefore, the purpose of this work was to evaluate SC performance by comparing three optimization algorithms in a case study with multiple objectives. Two objectives are maximizing profit and maximizing the level of customer service. Also, the modeled problem considers multiple products and periods for two security inventory scenarios (maximum and minimum inventory level). Evolutionary algorithms were compared: NSGA-II, MOPSO, and MOMA. The NSGA-II algorithm obtained the best result. A minimal inventory, NSGA-II presented 97.87% service level and the best benefit. Results show the importance of SC management and its optimization as well as some relevant variables to be considered.

Keywords: Supply Chain · Multi-objective Optimization · Algorithms · NSGA-II · MOPSO · MPAES.

1 Introduction

The Supply Chain (SC) has been of great interest in recent years since it encompasses functions such as planning, provisioning, production, distribution, and return. According to [36], companies now compete not only with each other, but also with the supply chains to which they belong. Because of this, improving the operation of the SC is important for many companies. This is achieved by optimizating variables associated with the supply, production, and marketing of goods and services. However, the existence of numerous decision variables and their complex interrelationships, as well as the limitations specific to each system, make SC a highly complex system [21, 22]. SC research has grown exponentially and points to "optimization and mathematical modelling" as the main research topics [25].

On the other hand, evolutionary algorithms (EA), which are based on the social behavior and natural biological evolution of species, have been one of the most studied optimization methods in the last decade [8, 16]. Considering the complexity of SC and the search for the optimization of both costs and benefits, several metaheuristics have been developed in EA, such as genetic algorithms (GA), particle swarm (PSO), and memetic algorithms (MA). These have yielded solutions that come close to a global one. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is the GA most applied to SC optimization studies where the total cost of SC is evaluated considering objectives such as "lead time" [22], the level of customer service [8], distribution [1, 35], inventories [11, 28], and SC with a "just in time" (JIT) approach [26]. Optimization by PSO has been demonstrated in a few studies. Some of these studies, in addition to optimizing the cost of SC, have opted to optimize both conventional logistics and inverse logistics in closed SC [16], and in other cases, have dealt with problems of scheduling [34] and order delays [15]. MA optimization, which is also known as a hybrid algorithm because two or more algorithms are combined, is known to improve algorithms (GA, PSO) in terms of quality of results. In SC, MA have been used to optimize costs, increase chain response capacity [29], and decrease environmental effects [14]. These algorithms have answered various manufacturing, logistics, and assembly problems. However, most of the cases mentioned consider a single optimization objective [1, 14, 23], and few investigations have studied multiple objectives. Fewer still have researched the entire SC, due to the high complexity of testing multiple objectives in this dynamic environment [22, 8, 35, 29].

In SC, optimization of resources is essential, but many optimization objectives are considered in isolation or are in conflict. Additionally, there are no studies that compare at least three EA and determine which of the EA perform best. Therefore, this research proposes to combine relevant SC variables that together generate several objective functions that were optimized using three evolutionary optimization algorithms. These algorithms are evaluated using a case study from a television assembly company. Indeed, this work aims to answer the question, "Which algorithm performs better than others in optimizing a SC in terms of time and quality of response?" The remainder of the paper is divided as follows. In the second section a literature review related to optimization models and algorithms for SC is presented. The third section clarifies the problem statement for optimization, where the variables and criteria of the SC are identified that allow the formulation of two objective functions to be optimized. The fourth section shows the main results obtained, and the fifth section discusses them. The last section articulates the conclusions of the investigation.

2 Theoretical background

2.1 Multi-Objective Optimization Problem

A multi-objective problem (MOP) is one that answers several unknowns that are posed at the same time in a problem. The solution of a multi-objective optimization problem presents a set of optimal solution points known as Pareto Front or Pareto Optimal. This procedure is supported by a dominance analysis to find solutions to the problem. A dominant solution is one in which there is no other solution that improves an objective function without harming at least one of the rest of the objectives [35]. There may also be a solution that is the best, but indifferent (neither dominant nor dominated) with respect to the target values [4]. In general, a multi-objective optimization problem includes a set of n decision variables, a set of k objective functions, and a set of m constraints. Where, $x = [x_1, x_2, ..., x_n]^T$ is the decision vector, while the objective vector of the decision spaces is X and Y respectively, and gi are the constraints of the problem functions. Azzouz et al. [2] pose the multi-objective problem according to (1), (2) and (3):

$$Optimize : y = f\left(x\right) = \left(f_1\left(x\right), f_2\left(x\right), \ldots, f_k\left(x\right)\right) \tag{1}$$

$$Subject\ to : gi\left(x\right) = \left(g_1\left(x\right), g_2\left(x\right), \ldots, g_m\left(x\right)\right) \geq 0 \tag{2}$$

$$Where : x = \left(x_1, x_2, \ldots, x_n\right) \in X; y = \left(y_1, y_2, \ldots, y_n\right) \in Y \tag{3}$$

2.2 Optimization by Genetic Algorithms

Genetic algorithms (GA) are principally based on imitating the adaptation and evolution phenomena of species, carrying out transformation and selection processes which simulate natural genetics [20]. The algorithm begins by generating an initial population, which is constituted of individuals who compete for survival. Each individual represents a possible solution to the optimization problem. After an evaluation process, the strongest individuals (parents) are selected, who will have the opportunity to "procreate" new individuals (children), through crossover and/or mutation processes. New individuals will inherit characteristics from their parents or may even improve them; in this way, the initial population evolves, and the cycle is repeated over and over again. Each cycle or loop of the algorithm is known as a generation [30]. The evolution of the population occurs mostly by three processes that are known as genetic operators that are: selection, which is responsible for choosing from the population the best individuals to transmit its genetic code to future generations; crossing, which consists of combining or "mating" two individuals (parents) to mix their genetic information to generate new individuals (children); and mutation, which selects a random individual and changing one or more current genes for new alleles (the value that takes a gene in a certain position) [26, 13].

The evolution of multi-objective GA begins with the GA called Vector Evaluated Genetic Algorithm (VEGA); unfortunately, this technique was not efficient [6]. Then, Goldberg [10] proposed new concepts to treat MOEAs to improve the quality of optimization, and clarified pareto optimality, non-dominated classification, and selection methods for first time in this field. The early generation of GAs was characterized by the simplicity of their algorithms and the lack of methodology to validate them. The most important are multi-objective Genetic Algorithm (MOGA), Niched Pareto Genetic Algorithm (NPGA) and

Non-Dominated Classification Genetic Algorithm (NGSA). Fonseca Fleming [9] presented MOGA, which implements a Pareto hierarchy, where the hierarchy of an individual depends on the number of individuals dominating a given population. NPGA, which focuses on tournament selection based on the dominance of Pareto and "fitness sharing," which is a technique to conserve diversity in GAs [6]. Srinivas & Deb [31] developed NGSA. It has the same MOGA structure and differs in the assignment of the aptitude function. In NSGA, all individuals near the Pareto front have high fitness values. Also, a $\sigma share$ distance is defined to measure the degree to which individuals affect their fitness function [28, 6]. The next generation of GA searched the efficiency in optimization. Deb et al. [7] presented NSGA-II. It is based on a technique of "elitist selection" and "crowded tournaments". Elitism consists in ensuring that the individuals with the best value of the adaptation function continue in the following iterations in order to prevent a loss in the adjustment obtained. In the selection by crowded tournaments, the winner of a tournament is judged based on the level that its fitness function contributes to the tournament [28]. Algorithm 1 shows the pseudo-code for NSGA-II [7].

Algorithm 1: NSGA-II

```
Create initial random population P0 (N individuals) and evaluate;
Create a population of offspring Qt, by crossing and mutation;
Set Pt = ∅;
while the stopping criterion is not met do
    Join the Pt and Qt populations to create the Rt population;
    Perform a non-dominated classification to Rt, and identify better
      non-dominated F_i fronts;
    while the size of |Pt| + 1 < N, do
        Calculate the agglomeration distance in F_i;
        Sort in F_i in descending order;
        Update Pt, i.e, Pt = Pt ∪ F_i (the best in F_i elements are
          assigned);
        Apply genetic operators;
    end
end
```

2.3 Particle Swarm Optimization (PSO)

PSO is an evolutionary calculation technique presented by Kennedy & Eberhart [19]. It is a search algorithm based on the simulation of the interaction of social behavior and the grouping of birds and fish [6, 32]. PSO is different from GA because it does not seek the survival of the fittest, so it does not adopt an individual selection process. Instead, it works with a population that is evaluated by one or several adjustment functions. Then the population is updated and the optimal solution is found [17]. The algorithm starts with a group of random particles where each individual is treated as a particle without volume in a multi-dimensional search space and at a random speed. Then, a set of particles are "thrown" into a search space with an initial position and velocity. Each particle in that space knows and remembers its position, its best previous position, as well as of your neighbors, and the value of the objective function in that position. In this way, all particles constantly adjust their direction and search speed according to the two best positions [16].

In [32] mentions some main PSO terms such as velocity (v), a vector that controls the direction in which the particle must "fly" to improve its position. Inertia weight (W), is used to control the impact of the previous velocity history on the current velocity of a particle. Learning factor is composed of two constants: C1 is a cognitive learning factor that simulates the attraction that a particle has towards its success, and C2 is the social learning that simulates the attraction of a particle towards its neighbors. Algorithm 2 shows the pseudocode to solve a particle swarm optimization problem.

Algorithm 2: PSO

```
Initialize swarm (speeds and best positions);
Initialize external record (initially empty);
while the stop criterion is not satisfied (number of iterations or
  variation between the current and previous solution) do
    for each particle do
        Select a member of the external registry (if necessary);
        Update speed and position;
        Evaluate new position;
        Update of best position and external registration;
    end
end
```

2.4 Optimization by Memetic Algorithms

Memetic Algorithms (MA) have their origin in the eighties when evolutionary computing was booming, and metaheuristic techniques were used and studied to optimize difficult problems. MAs arise from the combination of concepts and strategies of different metaheuristics to improve their performance. The term "meme" is analogous to the term "gen" of GA and it is attributed to R. Dawkins [23]. The MAs are population nature since they have inherited characteristics of the evolutionary algorithms. Memetic Pareto Archived Evolution Strategy (M-PAES) is an algorithm that uses an external file, which generates a new child with a mutated Gaussian operator and selects the next generation based on an uncontrolled comparison of parents and children evaluated in good condition [33]. Algorithm 3 presents the M-PAES pseudo-code.

Algorithm 3: M-PAES

```
Generate an initial population P of n random solutions and evaluate;
Place each non-dominated member of P in a global G file;
while the stopping criteria are not satisfied do
    for each solution c in P do
        Create a local file H, initially at 0;
        Fill file H with solutions of G that do not dominate c;
        Copy solution c from P to H;
        Run the local search using the PAES process (c, G, H);
        Replace the improved solutions of c; in P;
        Create an intermediate population (P′) and a ni (initially at 0);
        while neither < n do
            Put the number of test recombinations r = 0;
            while ((c is dominated by G) ∨ (c is at the fullest location on
              the mesh)) ∧ (r <maximum recombination of tests) do
                Randomly select 2 pairs of P ∪ G, recombine to form
                  offspring c;
                Compare c with the solutions in G; Change G with c if
                  necessary;
                if c is dominated by G then
                    discard c and use binary tournament to select a new
                      solution c from G
                end
                Place c in the intermediate population P′;
                Update population P with intermediate population P′;
            end
        end
    end
end
```

3 Statement of the optimization problem

3.1 Objective functions and constraints used for SC optimization

The investigation analyzed three different optimization algorithms that were evaluated using the data from a case study of a television assembly company located in Cuenca – Ecuador. The company name is not mentioned for confidentiality terms. For the same reason, not all the input data for the optimization problem is presented. The SC of the case study is composed of three levels, namely, suppliers, assembly plant, and customers; and two stages, production and distribution. It is necessary to determinate the best production and distribution combination to meet the demand of customers under the capacity restrictions at each level. Furthermore, it is important to optimize the inventory levels that the company maintains at the end of each period, especially the security inventory. The SC is integrated by six suppliers (S1-S6) that supply more than

170 items. There is a television assembly plant (P1) which in the study period (2017) assembled 14 television models. The distribution of SC is limited to five distribution centers (DC) that represent 82% of annual sales (DC1-DC5). For the problem statement the following assumptions were considered: The number of suppliers, plants and distribution centers, the demand for each distribution center and the plant capacity are known. The supply capacity of the suppliers is considered infinite, i.e., that the suppliers can meet the demands of the plant completely. Furthermore, in the provisioning stage, there is no selection of suppliers, so this term can be simplified in the objective function, and customers receive products from a single distribution center. On the other hand, the indices used to identify products and the parts or levels of the problem posed are described in Table 1.

Table 1. Indices notation for the mathematical model.

Index	Meaning
b	Products
j	Plants
k	Markets
z	Period

The input data come from information gathered about the company, corresponding to the costs associated with SC and the system capacity constraints (see Table 2).

Table 2. Mathematical model notation.

Notation	Meaning
CM_{bjz}	Unit cost of raw material supplied to assemble b in plant j, in period z
CE_{bjz}	Assembly cost (conversion) of product b in plant j, in period z
CT_{bkjz}	Unit cost of transporting product b from plant j to market k, in period z
CI_{bjz}	Unit cost of maintaining inventory of product b in plant j, in period z
PV_{bk}	Sale price of product b in market k
D_{bkz}	Demand for product b by customer k, in period z
I_{b0}	Initial inventory level of product b, in period $z = 0$
pt_{bj}	Time required to produce a unit of b in plant j
tt_{jz}	Time available to produce at plant j per period z
SS_{bjz}	Safety stock of product b in plant j, in period z

Decision variables or output variables of the proposed optimization problem are depicted in Table 3.

The variables in Table 2 and Table 3 were determined through a systematic review. The objective functions for the optimization problem was generated with the previous information. Equations (4) and (6) allow to maximize the utility of SC and maximize the level of customer service respectively.

Table 3. Decision variables notation used in the mathematical model.

Variable	Meaning
P_{bjz}	Total quantity of product b assembled in plant j, in period z.
PD_{kbkz}	Quantity of product b shipped from plant j to market k, in period z.
I_{bjz}	Quantity of final inventory of product b in plant j, in period z.

Objective 1: Maximize profit (Max P)

$$Max\ P = \sum_z \sum_j \sum_k \sum_b (PD_{bkjz} \cdot PV_{bk}) - TC \tag{4}$$

$$\begin{aligned} TC = &\sum_z \sum_j \sum_b (PD_{bjz} \cdot (CM_{bjz} + CE_{bjz})) + \\ &\sum_z \sum_j \sum_k \sum_b (PD_{bkjz} \cdot CT_{bkjz}) + \sum_z \sum_j \sum_b (I_{bjz} \cdot CI_{bjz}) \end{aligned} \tag{5}$$

Objective 2: Maximize Customer Service (Max CS)

$$Max\ CS = \left(\frac{\sum_z \sum_j \sum_k \sum_b PD_{bkjz}}{\sum_z \sum_k \sum_b D_{bkz}} \right) \cdot 100 \tag{6}$$

Equation (4) constituted the difference between sales (first term) and total cost (TC), second term. In turn, the total cost (5) is made up of the total cost of raw material, plus the cost of assembly, the cost of distribution and transportation of the products, and the cost of maintaining inventory at the end of a period in the plant. Equation (6) represents the level of customer service as a ratio between the products distributed and their demand. The problem to be optimized requires customer service levels high or greater than 90%. On the other hand, the restrictions that were identified for the case study are presented below in equations (7)-(11).
Constraints:

$$\sum_b pt_{bj} \cdot P_{bjz} \leq tt_{jz}; \forall b, j, z \tag{7}$$

$$\sum_j PD_{jbkz} \leq D_{bkz}; \forall b, j, k, z \tag{8}$$

$$I_{bjz} = I_{bj(z-1)} + P_{bjz} - \sum_k PD_{bjkz}; \forall b, j, k, z \tag{9}$$

$$I_{bjz} \geq SS_{bjz} \tag{10}$$

$$P_{bjz}, PD_{bjkz}, I_{bjz} \geq 0 \tag{11}$$

Constraint (7) ensures that the total time required to produce the products b does not exceed the total available time of plant j at each period z. Constraint (8) limits the quantity of products b shipped to market k at each period z, i.e., is not greater than the demand. Constraint (9) is a final inventory balance equation of product b at the plant j at each period z. Constraint (10) indicates the minimum inventory of product b that must be kept in the plant j at the end of each period z. Finally, constraint (11) indicates the non-negativity of the decision variables, which must take a value greater than or equal to 0.

3.2 Input Data for the Optimization Model in SC

The information used for this study corresponds to the 2017 period is presented in Table 4. It was obtained from the work carried out by [27] and [12].

Table 4. Unit time and costs, by television model.

Model	Raw material cost CM_{bjz}	Assembly cost CE_{bjz}	Inventory holding cost CI_{bjz}	Transport cost CT_{bkjz}	Assembly time pt_{bj}(min)
1	$ 358.59	$ 14.77	$ 6.29	$ 4.00	12.058
2	$ 522.93	$ 15.20	$ 6.29	$ 5.00	12.435
3	$ 446.95	$ 12.82	$ 6.29	$ 5.00	12.435
4	$ 174.44	$ 10.83	$ 6.29	$ 3.00	8.272
5	$ 269.41	$ 17.08	$ 6.29	$ 3.00	8.272
6	$ 349.17	$ 15.51	$ 6.29	$ 3.00	12.167
7	$ 401.06	$ 15.13	$ 6.29	$ 5.00	12.167
8	$ 758.89	$ 25.23	$ 6.29	$ 7.00	18.426
9	$ 965.05	$ 32.38	$ 6.29	$ 7.00	24.517
10	$ 297.14	$ 20.05	$ 6.29	$ 3.00	18.426
11	$ 402.99	$ 13.51	$ 6.29	$ 4.00	12.435
12	$ 529.38	$ 15.58	$ 6.29	$ 4.00	18.519
13	$ 604.26	$ 17.23	$ 6.29	$ 5.00	18.519
14	$ 268.43	$ 8.33	$ 6.29	$ 3.00	9.728

Other input data for the optimization model are D_{bkz} and PV_{bk} as indicated in Table 2. However, they are not presented due to the terms of confidentiality with the company. In addition, maintaining a safety inventory is required SS_{bjz}, which is calculated using the standard deviation of the product (Z) and the number of standard deviations, equation (12). Generally, the aim is to obtain a service level of at least 95%, so for this value a number of standard deviations (Z) of 1,645 is obtained. The replacement time is considered one month, and the inventory at the beginning of each period I_{b0}, as zero (0) due to the limitation in the provisioning information.

$$SS = Z \cdot \sigma_{dLT} \tag{12}$$

Besides, both scenarios with a maximum and with a minimum security inventory are considered. For this, a “buffer” is used, which allows controlling inventories in zones or levels according to the Target Inventory Level (NOI). The first scenario is the part of the inventory level in which the products needed to meet demand can be consumed and must be greater than 2/3 of the NOI. The second scenario represents an alarm, which indicates that it must occur to fill the inventory in such a way that it is not completely consumed. This inventory is less than 1/3 of the NOI [5]. These scenarios allow analyzing the possible variations that may occur in the SC and estimates their impacts on the optimization objectives set. For the case study, the NOI value for each model is calculated with equation (12) and Table 5 presents the calculated values for both scenarios.

Table 5. Security inventory levels for the two analysis scenarios.

Model	NOI	Max 2/3 of NOI	Min 1/3 of NOI
1	815	543	272
2	0	0	0
3	878	585	293
4	1518	1012	506
5	0	0	0
6	274	183	91
7	344	229	115
8	68	45	23
9	44	30	15
10	313	209	104
11	506	337	169
12	36	24	12
13	45	30	15
14	268	179	89

3.3 Determination of the number of iterations and parameters

Determining the number of iterations or evaluations in the use of an algorithm is crucial since it allows reducing the computational cost without neglecting the normal performance of the algorithm. From 10,000 to 100,000 iterations have been tested for this study. For optimization using NSGA-II, and MOPSO algorithm, the parameters used are presented in Table 6.

Table 6. NSGA-II algorithm and MOPSO algorithm parameters.

NSGA-II algorithm parameters		MOPSO algorithm parameters	
Parameter	Value	Parameter	Value
Population size (N)	100	Initial Poblation	100
Crossover Probability (Pc)	0,9	Inertia weight, W	0,4
Mutation probability (Pm)	0,01	Local learning coefficient, C1	1
		Global learning coefficient, C2	1

Finally, for the memetic algorithm (M-PAES), with the increase in the number of iterations, the result as a function of the benefit and level of service improves considerably and the size of the population is reduced when $N = 100$ was initially indicated. We considered the parameters to implement M-PAES in Table 7.

Table 7. M-PAES algorithm parameters.

Parameter	Simbol	Value
Maximum number of consecutive failures of local movements	L_fails	5
Maximum number of local movements	L_opt	10
Number of crosses	C_trials	20

4 Results

4.1 Multi-objective Optimization Results

Based on the data and objective functions described in section 3, the results obtained by the three algorithms are presented with a minimum and maximum safety inventory, and 10k and 100k iterations (see Table 8 and Table 9).

Table 8. Results obtained from the three algorithms with minimum inventory.

Algorithm	F OBJ. 1	F OBJ. 2	TIME (s)
NSGA-II (10k)	$ 3,624,514.87	86.18	11.26
MOPSO (10k)	**$ 3,728,841.21**	**88.75**	**7.84**
M-PAES (10k)	$ 2,905,654.15	72.10	129.48
NSGA-II (100k)	**$ 4,272,669.41**	**97.87**	**89.94**
MOPSO (100k)	$ 3,881,112.75	90.33	55.61
M-PAES (100k)	$ 4,185,197.22	95.26	420

Table 9. Results obtained from the three algorithms with maximum inventory.

Algorithm	F OBJ. 1	F OBJ. 2	TIME (s)
NSGA-II (10k)	$ 2,856,451.52	85.26	9.55
MOPSO (10k)	**$ 3,074,152.51**	**87.47**	**8.19**
M-PAES (10k)	$ 2,135,499.08	70.31	48.68
NSGA-II (100k)	**$ 3,576,619.52**	**97.73**	**75.95**
MOPSO (100k)	$ 3,271,055.75	91.93	61.33
M-PAES (100k)	$ 3,475,744.91	95.49	385

Table 8 and Table 9 show as a general trend that the greater the number of iterations, the better the response level of the algorithm, but also the greater the computational processing time for all tested algorithms. Also, when the level of security inventory is the maximum, the costs associated with maintenance and conversion increase, reducing the profit of the company (see Fig. 1).

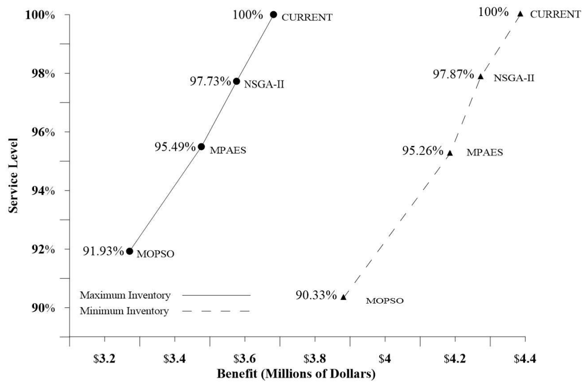

Fig. 1. Results of the tested algorithms.

The MOPSO algorithm obtains the lowest benefit, followed by M-PAES with a slight difference of an increase in its benefit, and finally, NSGA-II, which obtains the highest benefit among the three algorithms. On the other hand, with 100k evaluations and for the two inventory levels, MOPSO presents the lowest service level among the three cases, with 90.33% and 91.93% respectively, followed by M-PAES with 95.26% and 95.49% and finally with NSGA-II, which presents the best service level of 97.87% and 97.73%, respectively (see Fig. 2).

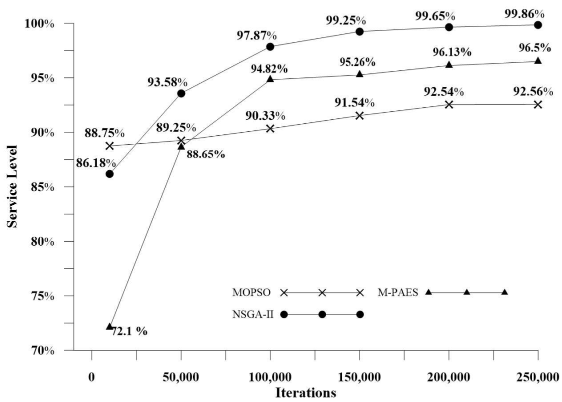

Fig. 2. Number of iterations per algorithm.

5 Discussion

In this research, two scenarios for optimization were testing. These two scenarios consider favorable and pessimistic options. The behavior of the service level concerning the benefit according to the different scenarios shows an improvement depending on the algorithm used (see Fig. 1). The NSGA-II performs best under the established conditions. This outcome coincides with [22] who mention in their work that the NSGA-II is the most widely used algorithm for multi-objective optimization. However, these authors compared their results with only one algorithm, while in the present research it was done with three algorithms, giving a more realistic comparison.

On the other hand, the three algorithms are compared, taking into account the relationship of the number of iterations with the service level (see Fig. 2). It can be seen that the behavior of the service level curve when it is optimized with NSGA-II and M-PAES is logarithmic, so a large increase in the number of iterations is needed to obtain a minimum increase in the service level. Although MOPSO, with a lower number of iterations, presents better results than the other two algorithms. While the number of iterations increases, the service level increases very slowly in the space of possible solutions, having a linear trend, so that the two other algorithms surpass it. M-PAES is known to be a complex hybrid algorithm that takes longer to present solutions, but it turned out to be better in terms of quality of results compared to MOPSO. Still, for M-PAES, these solutions improve by increasing the number of individuals in the population, differing from the other two algorithms where increasing the number of individuals affects the speed of convergence vs number of iterations. Therefore, 100k proposed iterations are enough to obtain favorable results and to be able to compare the three algorithms in a fair situation. Furthermore, the computational cost can be evaluated under the same conditions.

The SC optimization problem has been studied by several authors, resulting in different answers to questions posed or objectives raised. The results of this investigation are compared with others found in the bibliography, for example, with [16, 3, 15], who agree that optimization by GA gives better results than PSO in bi-objective optimization of the SC. All of them agree that the MOPSO algorithm presents solutions in less time than other algorithms, just like our optimization problem. While the memetic algorithm, M-PAES takes the longest time to present a quality answer, other SC problems as in [14, 29] and other similar studies [18, 24] have presented good results as a hybrid algorithm. In this case study, M-PAES, despite not being a widely used algorithm for optimizating the SC, presented better results than MOPSO, but worse than NSGA-II.

The SC studied gives more emphasis to assembly and distribution, which are two critical areas. While the suppliers are considered, they can provide the material necessary for the assembly line. Some works propose a SC of up to four stages with five levels (supplier, producer, distributor, wholesaler and retailer). However, research such as [3, 11, 16, 35], to name a few, work with few products and few study periods. In SC of the case study, the number of variables that the problem handles increase its complexity of resolution. Thus, a total of 1,176 variables were taken into account, since it considers 12 study periods, 1 assembly plant, 14 products and 5 distribution centers. This complexity reveals the relative efficiency of optimization algorithms, leading to find optimal results. On the other hand, the proximity of the solution sets obtained by the NSGA-II genetic algorithm is greater than the that of sets found by the MOPSO and M-PAES algorithms, for the two study scenarios, with a minimum and maximum security level inventory.

6 Conclusion

A multi-objective optimization problem of costs in the supply chain in an assembly company was studied. This problem was resolved by applying and comparing three evolutionary algorithms. The algorithm that had the best performance was NSGA-II. Furthermore, this research contributed to identifying the variables of the SC for assembly companies that are involved in an optimization. The findings clearly indicate that the number of iterations significantly affects some algorithms, and multi-objective optimization shows that NSGA-II is more effective than MOPSO and M-PAES in searching the feasible solutions space for a set of solutions that present very close objective vectors or contained in the real set of the Pareto optimum. Thus, a value of 97.87% of service level with a benefit of $4,272,669.41 was reached by this algorithm, this value represents an increase of $391,556.66 of the worst result given by MOPSO. The minimum level of security inventory was the best optimization scenario. One of the limitations in this investigation was assuming that the suppliers of the SC were capable of supplying all the requirements of the plant. However, in many SC the stage of raw material supply is considered strategic and presents several challenges, since it involves different lead times depending on the type of raw material and the location of the supplier. Future work will consider this limitation. This study can be expanded to consider the level of inventory in the assembly plant, because these inventories imply high costs in some sections of the SC.

Acknowledgments

This study is part of the research project "Modelo de optimización de costos en la cadena de suministro en empresas de ensamblaje" supported by the Research Department of the University of Cuenca (DIUC). The authors gratefully acknowledge the contributions and feedback provided by the IMAGINE Project team; similarly, to the management and operational staff of television assembly company for their willingness to being analyzed as a case study.

References

1. Altiparmak, F., Gen, M., Lin, L., Paksoy, T.: A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & industrial engineering **51**(1), 196–215 (2006)

2. Azzouz, R., Bechikh, S., Said, L.B.: Dynamic multi-objective optimization using evolutionary algorithms: a survey. In: Recent advances in evolutionary multi-objective optimization, pp. 31–70. Springer (2017)
3. Babaveisi, V., Paydar, M.M., Safaei, A.S.: Optimizing a multi-product closed-loop supply chain using nsga-ii, mosa, and mopso meta-heuristic algorithms. Journal of Industrial Engineering International **14**(2), 305–326 (2018)
4. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Computing **23**(9), 3137–3166 (2019)
5. Cuesta, S.: Propuesta de optimización de procesos basado en herramientas de manufactura esbelta en industrias de ensamblaje. Casos de estudio: ensambladoras de televisores y de tarjetas electrónicas. Ph.D. thesis, Universidad de Cuenca (Nov 2019), http://dspace.ucuenca.edu.ec/handle/123456789/33784
6. Deb, K.: Rise of evolutionary multi-criterion optimization: A brief history of time with key contributions. In: Evolution in Action: Past, Present and Future, pp. 351–368. Springer (2020)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
8. Farahani, R.Z., Elahipanah, M.: A genetic algorithm to optimize the total cost and service level for just-in-time distribution in a supply chain. International Journal of Production Economics **111**(2), 229–243 (2008)
9. Fonseca, C.M., Fleming, P.J., et al.: Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: Icga. vol. 93, pp. 416–423. Citeseer (1993)
10. Goldenberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading: MA (1989)
11. Guerrero, M., Gómez, D., Zapata, D., Cárdenas, M.V.: Comparación de tres metaheurísticas para la optimización de inventarios con estimación de demanda. Revista Ingeniería Industrial **15**(1), 51–68 (2016)
12. Guerrero, P.: Tiempos estándar y modelización de procesos de ensamblaje: Televisores y tarjetas electrónicas usando programación no lineal y BPMN (Tesis (Pregrado), Universidad de Cuenca). Ph.D. thesis (2018)
13. Hernández Romero, N.: Introduccion a Matlab para Resolver Problemas de IngenieraAplicando Algoritmos Geneticos (2012)
14. Jamshidi, R., Ghomi, S.F., Karimi, B.: Multi-objective green supply chain optimization with a new hybrid memetic algorithm using the taguchi method. Scientia Iranica **19**(6), 1876–1886 (2012)
15. Javanshir, H., Ebrahimnejad, S., Nouri, S.: Bi-objective supply chain problem using mopso and nsga-ii. International Journal of Industrial Engineering Computations **3**(4), 681–694 (2012)
16. Kannan, G., Noorul Haq, A., Devika, M.: Analysis of closed loop supply chain using genetic algorithm and particle swarm optimisation. International Journal of Production Research **47**(5), 1175–1200 (2009)
17. Kao, Y.T., Zahara, E.: A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied soft computing **8**(2), 849–857 (2008)
18. Karaoglan, I., Altiparmak, F.: A memetic algorithm for the capacitated location-routing problem with mixed backhauls. Computers & Operations Research **55**, 200–216 (2015)

19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE (1995)
20. Kramer, O.: Genetic algorithm essentials, vol. 679. Springer (2017)
21. Margolis, J.T., Sullivan, K.M., Mason, S.J., Magagnotti, M.: A multi-objective optimization model for designing resilient supply chain networks. International Journal of Production Economics **204**, 174–185 (2018)
22. Mendoza, A.A.M., Herrera, T.J.F., Cadavid, D.A.V.: Optimización multiobjetivo en una cadena de suministro. Revista ciencias estrategicas **22**(32), 295–308 (2014)
23. Moscato, P., Cotta, C.: An accelerated introduction to memetic algorithms. In: Handbook of Metaheuristics, pp. 275–309. Springer (2019)
24. Moshref-Javadi, M., Lee, S.: The latency location-routing problem. European Journal of Operational Research **255**(2), 604–619 (2016)
25. Movahedipour, M., Yang, M., Zeng, J., Wu, X., Salam, S.: Optimization in supply chain management, the current state and future directions: A systematic review and bibliometric analysis. Journal of Industrial Engineering and Management **9**(4), 933–963 (2016)
26. Nestor, V.B., NESTOR, V.B.: Algoritmo genético multiobjetivo autoevolutivo para resolver el problema de secuenciación de trabajos en una línea de ensamble con sistema de producción justo a tiempo. Ph.D. thesis
27. Ochoa, M.M.G., Arias, B.E.C., Siguenza-Guzman, L., Segarra, L.: Integración de información de costos para la toma de decisiones en industrias de ensamblaje. Revista Economía y Política pp. 100–117 (2020)
28. Pinto, E.G.: Supply chain optimization using multi-objective evolutionary algorithms. vol. 15, p. 2004 (2004)
29. Pishvaee, M.S., Farahani, R.Z., Dullaert, W.: A memetic algorithm for bi-objective integrated forward/reverse logistics network design. Computers & operations research **37**(6), 1100–1112 (2010)
30. Shi, J., Liu, Z., Tang, L., Xiong, J.: Multi-objective optimization for a closed-loop network design problem using an improved genetic algorithm. Applied Mathematical Modelling **45**, 14–30 (2017)
31. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation **2**(3), 221–248 (1994)
32. Trivedi, V., Varshney, P., Ramteke, M.: A simplified multi-objective particle swarm optimization algorithm. Swarm Intelligence pp. 1–34 (2019)
33. Vikhar, P.A.: Evolutionary algorithms: A critical review and its future prospects. In: 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC). pp. 261–265. IEEE (2016)
34. Xiong, F., Gong, P., Jin, P., Fan, J.: Supply chain scheduling optimization based on genetic particle swarm optimization algorithm. Cluster Computing **22**(6), 14767–14775 (2019)
35. Zapata Cortés, J.A.: Optimización de la distribución de mercancías utilizando un modelo genético multiobjetivo de inventario colaborativo de m proveedores con n clientes. Ph.D. thesis (2016)
36. Zhou, M., Dan, B., Ma, S., Zhang, X.: Supply chain coordination with information sharing: The informational advantage of gpos. European Journal of Operational Research **256**(3), 785–802 (2017)