International Journal for Quality Research 13(4) 901–914
ISSN 1800-6450

Bruno Cretú
Dalila B. M. M. Fontes[1]
Seyed Mahdi Homayouni

A GENETIC ALGORITHM FOR A MULTI-PRODUCT DISTRIBUTION PROBLEM

Article info:
Received 18.06.2019
Accepted 05.09.2019

UDC – 339.188
DOI – 10.24874/IJQR13.04-11

Abstract: *This paper addresses a distribution problem involving a set of different products that need to be distributed among a set of geographically disperse retailers and transported from the single warehouse to the aforementioned retailers. The distribution and transportation are made in order to satisfy retailers' demand while satisfying storage limits at both the warehouse and the retailers, transportation limits between the warehouse and the retailers, and other operational constraints. This problem is combinatorial in nature as it involves the assignment of a discrete finite set of objects, while satisfying a given set of conditions. Hence, we propose a genetic algorithm that is capable of finding good quality solutions. The genetic algorithm proposed is used to a real case study involving the distribution of eight products among 108 retailers from a single warehouse. The results obtained improve on those of company's current practice by achieving a cost reduction of about 13%.*

Keywords: *Genetic Algorithms; Multi-Product; Distribution Problem.*

1. Introduction

Physical distribution of products has been grown significantly over time, particularly in the last decades. This can be explained by the growing market of e-commerce, increasing customer service expectations, globalization, among other factors. Due to the highly competitive environment of physical distribution systems, several decision elements must be optimized in order to achieve efficiency and effectiveness, namely: transportation, inventory, warehousing, and order handling. These decision elements, typically, involve several issues from which many problems stem. For example, transportation problems may involve fleet management, fleet routing, transportation mode choices, to name but a few. Addressing simultaneously all the issues that may arise in each of the elements mentioned is often impossible due to the excessive complexity involve.

This work considers a supply chain, consisting of a single warehouse and multiple retailers facing deterministic demands, under a vendor managed inventory (VMI) policy. The vertical collaboration in the distribution network can enhance its efficiency. In this system, the vendor has dynamic information from the retailers about their inventory levels and is able to take decisions when and how much to deliver (Waller et al., 1999). Moreover, the problem being addressed falls into the inventory routing category. The basic inventory routing problem (IRP) is defined as "the repeated distribution of single product, from a single facility, to a set of customers over a given planning horizon" (Campbell et al., 1998). Several operational constraints related to inventory holding, transportation, and product supply capacities, as well as demand restrictions must be considered while planning the inventory and transportation operations.

[1] Corresponding author: Dalila B. M. M. Fontes
Email: dfontes@inesctec.pt

The inventory routing problem is not new, and it has been the subject of research for over 30 years, during which a multitude of versions arose. Many of the considered versions have their original, or at least are inspired, on real applications that require the consideration of aspects such as the planning period (single, multiple, infinite), the demand type (deterministic and stochastic), the route type (direct only, with transshipment), the inventory strategies (allowed/not allowed backlog and shortages), the timing of delivery (on time, early, late), among many others. A comprehensive discussion on inventory routing problems, including models and limitations can be found in (Moin & Salhi, 2007) and (Bertazzi & Speranza, 2012) and thorough reviews, based on a new classification of the problem, with respect the structural variants, and the availability of information on customer demand are provided by Coelho et al. (2013), Roldán et al. (2017), and Dong et al. (2017).

The basic IRP is NP-hard since it includes the vehicle routing problem, which is similar to a pick-up and delivery problem (Lenstra & Rinnooy Kan, 1981). Therefore, many researchers proposed heuristic or metaheuristic solution approaches for the problem. Li et al. (2014) considered the IRP in a gasoline distribution network, in which stockout avoidance in every station is of major importance. The problem has been solved using a tabu search algorithm. A solution is constructed by a sequence of the gas stations in non-decreasing order, and the stations are then inserted into the vehicle routes in a greedy manner. Coelho et al. (2014) defined an IRP that considers customer demands to be gradually revealed over the time. They propose heuristic methods based on proactive and on reactive policies over a rolling horizon. Under the reactive policy one tracks the state of the system, and an order to replenish the inventory to a certain level is triggered whenever the inventory reaches the reorder point. The reorder point is determined taking into account both the delivery lead time and the stockout risk associated due to the stochasticity of the demand. The proactive policy, in addition to tracking the state of the system also makes predictions regarding the following period demand. Recently, Cordeau et al. (2015) consider a similar version of the IRP, however, in it customers have a maximum inventory level for each product rather than a total shared capacity for all products. The authors propose a three-phase heuristic, which decomposes the decision process into i) replenishment plans using a Lagrangian-based method, ii) delivery sequences for the vehicles using a simple procedure, and iii) planning and routing decisions using a mixed-integer linear programming model. More recently, other approximate solution approaches have been designed for the IRP; for example, a genetic algorithm (GA) approach for the IRP of a single perishable product by Azadeh et al. (2017), a hybrid randomized variable neighborhood descent for the multi-vehicle, multi-product and multi-period IRP by Peres et al. (2017), a GA for location-inventory-routing model for perishable products by Hiassat et al. (2017), a discrete invasive weed optimization and a GA for the IRP for a single product with allowed backorders by Jahangir et al. (2019), to name but a few.

In recent years, environmental concerns have also made their way into the IRP and several works have been published. Rahimi et al. (2017) developed a multi-objective fuzzy mathematical model for the IRP considering the economic performance, the service level, and the carbon footprint. The authors also consider that perishable products have a specific expiration date. Further, they developed a Non-dominated Sorting GA II (NSGA-II), to find solutions for the problem. Soysal et al. (2018) consider the benefits of horizontal collaboration between multiple suppliers and customers of perishable products. They developed a mathematical formulation of the IRP which minimizes the expected inventory cost, the expected waste cost, fuel cost from transportation operations, and also driver cost. The interested reader is referred to (Malladi & Sowlati, 2018) for a recent review on IRPs considering conventional performance indicators and sustainability objectives.

In this work, we considered a particular case of the IRP, with simpler routing decisions, as will be discussed in Section 2. The motivation for addressing this problem is a specific application from a Portuguese fashion company, and it involves 108 retailers and eight products (merchandise categories). In the version studied here the routing part is ignored as the transportation is not done in-house, that is, a contract has been signed with a transportation company, in which the costs incurred are proportional to the boxes transported. Though, the transportation costs and its lower and upper limits are taken into account. Therefore, in here and regarding the transportation decisions we only determine the number of boxes to be transported from the warehouse to each retailer. These and other approximations have been considered in the literature; the more closely related one being the consideration of direct shipping only. It has been shown that if the quantity to deliver to each retailer is close a full truck load, then direct delivery is efficient (Gallego & Simchi-Levi, 1990). We use both direct and clustered shipping in order to minimize the overall inventory, transportation, and handling costs, while satisfying demand, storage capacity, and transportation capacity limits. All four above mentioned major elements are considered when deciding on each product quantities to transport from the single warehouse to the several retailers involved. Some elements are explicitly addressed; while others are considered through the relations among the elements, the limitations imposed, or the cost minimization. The remaining sections of this paper are organized as follows. Section 2 provides the detailed problem description and, after introducing the notation used, the mixed integer linear programming model developed. In Section 3, the proposed genetic algorithm is described in detail. Section 4 reports results corresponding to the parameter tuning for the proposed GA and comparative results for the case study of the Portuguese fashion company under consideration. Finally, conclusions are drawn in Section 5.

2. Problem Definition and Formulation

The problem being addressed consists of a single warehouse that supplies a set of retailers with known and deterministic demand. A fleet of vehicles with limited capacity is used to send the products from the warehouse to the retailers. The warehouse is supplied from outside sources and since these supplies are not decided within the scope of the problem, they are considered a problem parameter. However, warehouse incoming supplies have to be coordinated with outgoing ones as the warehouse has limited storage capacity. Warehouse outgoing shipments must also be coordinated with retailers incoming shipments (from the warehouse) as retailers, in addition to have minimum demand requirements also have limited storage capacity, and no shortage or backlog are allowed for them. Finally, the total cost, which is to be minimized, includes inventory holding costs and handling costs at both the warehouse and the retailers and shipping costs (from the warehouse to the retailers). Storage capacity at the retailers has a soft and a hard constraint. The soft constraint refers to the retailer's own storage space that can be used without incurring in additional costs since it is part of the retailer and thus, can be considered a sunk cost. The hard constraint refers to additional storage space, up to 10% of the original limit that can be used; however, to use this additional space a cost per square meter has to be paid.

Demand at each retailer is calculated based on the previous year sales as well as on an expected growth rate. Furthermore, due to business rules its value must cover for the expected demand of the following four weeks. An additional complexity that we need to deal with comes from the fact that product needs are specified in product units; however, product handling, transportation, and storage needs are all specified in boxes.

The company needs to make the product allocation decisions every week and usually considers a planning horizon of 52 weeks. In this work, this multi-period planning problem is addressed by a repeated single period approach, in which solutions are obtained for each week by the genetic algorithm described in Section 3. The notation used is provided in Table 1, followed by the mathematical programming model.

Table 1. Notation Used in the MILP Model

Indices and Sets	Description
$i \in I$	Set of products, $I = \{1, \ldots, n\}$.
$k \in K$	Set of retailers, including the warehouse, $K = \{1, \ldots, m\}$, where m is associated with the warehouse.
Variables	Description
x_{ik}	Quantity (in units) of product $i \in I$ sent to retailer $k \in K \backslash \{m\}$.
z_{ik}	Number of boxes required to send x_{ik} units of product $i \in I$ to retailer $k \in K \backslash \{m\}$.
S_{ik}^0	Initial stock (in units) of product $i \in I$ at facility (retailer or warehouse) $k \in K$.
S_{ik}^1	Final stock (in units) of product $i \in I$ at facility (retailer or warehouse) $k \in K$.
Q_{ik}	Number of boxes required to hold the stock product $i \in I$ in facility (retailer or warehouse) $k \in K$.
e_k	Excessive overall stock, in boxes, at facility (retailer or warehouse) $k \in K$.
Parameters	Description
D_{ik}	Demand (in units) of product $i \in I$ by retailer $k \in K \backslash \{m\}$.
P_{ik}	Expected sales (in units) of product $i \in I$ by retailer $k \in K \backslash \{m\}$.
A_{ik}	Real sales (in units) of product $i \in I$ by retailer $k \in K \backslash \{m\}$.
US_k	Total storage capacity (in boxes) of facility (retailer or warehouse) $k \in K$.
OO_i	On order quantity (in boxes) of product $i \in I$.
CH	Handling cost per box.
CR_k	Rent cost per square meter for facility (retailer or warehouse) $k \in K$.
TC	Transportation cost per box.
UM	Maximum number of boxes that can be transported to a retailer.
UN_{ik}	Maximum quantity of product $i \in I$ that can be transported to retailer $k \in K \backslash \{m\}$.
LN_{ik}	Minimum quantity of product $i \in I$ that needs to be transported to retailer $k \in K \backslash \{m\}$.
C_i	Box capacity (in units) for product $i \in I$.

$$Minimize \quad C = TC \sum_{i \in I} \sum_{k \in K \backslash \{m\}} z_{ik} + CH \sum_{i \in I} \sum_{k \in K} z_{ik} + \sum_{k \in K} \frac{CR_k \times e_k}{2}. \tag{1}$$

Subject to:

$$z_{ik} \geq \frac{x_{ik}}{C_i}. \qquad \forall i \in I. k \in K \backslash \{m\}, \tag{2}$$

$$z_{im} = \sum_{k \in K \backslash \{m\},} z_{ik}. \qquad \forall i \in I, \tag{3}$$

$$Q_{ik} \geq \frac{s_{ik}^0 + x_{ik}}{C_i}. \qquad \forall i \in I, k \in K \backslash \{m\}, \tag{4}$$

$$Q_{im} \geq \frac{s_{im}^{0} - \sum_{k \in K \setminus \{m\},} x_{ik}}{C_i}. \qquad \forall i \in I, \qquad (5)$$

$$e_k \geq \sum_{i=1}^{n} Q_{ik} - US_k. \qquad \forall k \in K \setminus \{m\}, \qquad (6)$$

$$e_m \geq \sum_{i \in I,} (Q_{im} + OO_i) - US_m. \qquad (7)$$

$$s_{ik}^{0} + x_{ik} \geq D_{ik}, \qquad \forall\, i \,\epsilon\, I, k \,\epsilon\, K \setminus \{m\}, \qquad (8)$$

$$e_k \leq 0.1\, US_k, \qquad \forall k \in K, \qquad (9)$$

$$x_{ik} \leq UN_{ik}, \qquad \forall i \in I, k \in K \setminus \{m\}, \qquad (10)$$

$$x_{ik} \geq LN_{ik}, \qquad \forall i \in I, k \in K \setminus \{m\}, \qquad (11)$$

$$\sum_{i \in I,} x_{ik} \leq UM. \qquad \forall k \in K \setminus \{m\}, \qquad (12)$$

$$s_{ik}^{1} = s_{ik}^{0} + x_{ik} - P_{ik}, \qquad \forall i \in I, k \in K \setminus \{m\}, \qquad (13)$$

$$s_{im}^{1} = s_{im}^{0} - \sum_{k \in K \setminus \{m\},} x_{ik},. \qquad \forall\, i \,\epsilon\, I, \qquad (14)$$

$$x_{ik}, z_{ik}, Q_{ik} \geq 0 \text{ and integer}, \qquad \forall i \in I. k \in K, \qquad (15)$$

$$s_{im}^{0}, s_{im}^{1}, e_k \geq 0, \qquad \forall i \in I. k \in K. \qquad (16)$$

The cost minimization nature of the problem is described by Equation (1). There are three types of costs, namely transportation costs, handling costs, and storage costs. Both the transportation costs and the handling costs are proportional to the total number of boxes; while the storage costs are only incurred with excessive storage, i.e., when the total number of boxes in inventory is over the pre-defined storage capacity but within the 10% allowed.

Constraints (2) to (5) ensure the required number of boxes are used to send the products from the warehouse to the retailers, the number of boxes leaving the warehouse is the same as the total number of boxes arriving at the retailers, the correct number of boxes is used to hold the inventory at both the warehouse and the retailers. Constraints (6) and (7) determine the number of boxes required to hold the inventory over the capacity limits for the warehouse and the retailers, respectively. Constraints (8) ensure that each retailer has enough of each product to satisfy the demand calculated using the business rule, while constraints (9) ensure that the overall excessive stock in each retailer does not exceed the 10% limit over the pre-defined storage capacity. Retailer incoming shipments, per product, must be within retailer upper and lower limits as given by inequalities (10) and (11). In addition, an upper limit on the total amount of product each retailer can handle is imposed by inequalities (12). Equations (14) and (15) are the usual balance equations for the retailers and the warehouse, respectively. Finally, the nature of the variables is stated in constraints (16) and (17). It should be noticed that, the variables associated with stock (s and e variables) need only to be defined as non-negative, since they are obtained by adding and subtracting integer values.

Recall that the problem involves the planning for 52 weeks, thus the model introduced is to be solved every week. Therefore, the inventory held at the retailers and warehouse at the beginning of the week is given as any other input, while the one at the end of the week is an output to be used in the following week. Thus, at the beginning of each week the inventory needs to be corrected by using the real sales rather than the expected sales: $s_{ik}^1 = s_{ik}^0 + P_{ik} - A_{ik}$. Regarding the warehouse, at the end of the current week the on-order true value becomes known and thus the inventory is updated as $s_{im}^1 = s_{im}^0 + OO_i$.

3. The Genetic Algorithm

As mentioned before, the problem being addressed is NP-hard; therefore, a metaheuristic approach, more specifically a genetic algorithm, is proposed. This section describes the GA in detail.

GA is a stochastic search metaheuristic based in the principles of natural selection and evolution of the species, first proposed by Holland and his colleagues (Holland, 1975). A population (set of individuals representing solutions) is evolved (improved) over several generations (iterations) through the use of genetic operators (selection, crossover, and mutation). The basic idea is to select the fittest (best) individuals (solutions) from the population to produce offspring (new solutions) that inherit the genetic material (solution characteristics) of the parents. Fitter (better) parents (solutions) are more likely to produce fitter (better) offspring (new solutions), which in turn will have a better chance of surviving and being chosen for future reproduction. This process is repeated many times with the purposes of finding a population with the fittest individuals, i.e. a set of very good quality solutions that hopefully contains an optimal or at least one very close to it.

Solutions that consist of a set of variables represent an individual that consist of a set of genes: The set of genes forms a chromosome, which in a GA is, usually, encoded as a string. The GA works with these strings, which then need to be decoded, i.e., converted into a solution in order to evaluate its fitness (solution quality). This conversion needs to be fast as it is carried out for the whole population at every generation (iteration). The structure of the GA proposed in this work is depicted in Figure 1.

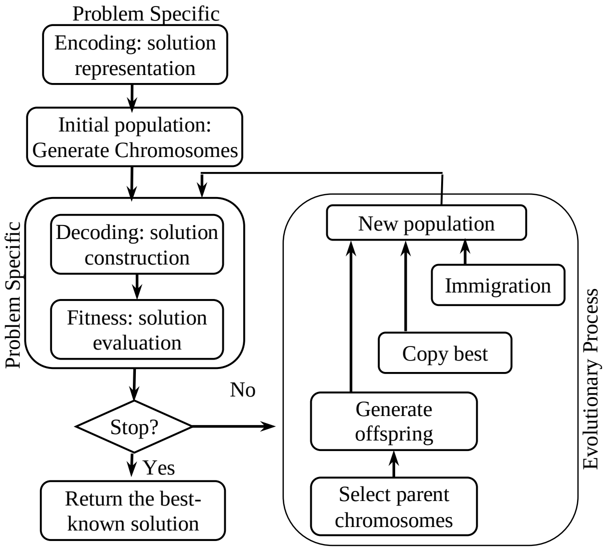

Figure 1. Structure of the proposed genetic algorithm

3.1. Solution Representation

Encoding of chromosomes is one of the problems, when you are starting to solve problem with GA. Encoding very depends on the problem.

Solution representation, also known as encoding is very much problem dependent. Several encoding schemes have been proposed over the years; in here a solution is represented by a matrix of integers. Rows are associated with retailers and thus, there are as many rows as retailers to be supplied and columns are associated with products, i.e., the number of columns is given by the number of products. Therefore, the genes are the quantities of each product to be sent to each retailer. This encoding is easy to understand and provides a straightforward decoding; although feasibility issues may arise. Figure 2 provides an example of a possible solution for an instance involving eight products and four retailers to which the products are to be delivered to. The first column shows the quantities of product 1 to be sent to the retailers, in this example: none to retailers 1 and 2, 72 units to retailer 3, and 147 units to retailer 4. If we look at rows, we can read the amount of each product that is sent to each retailer; for example, the first rows, specifies that retailer 1 receives 33 units of product 2, 530 units of product 4, and three units of product 7, but it does not receive any units of products 1, 3, 5, 6, and 8.

	Products							
Retailers	1	2	3	4	5	6	7	8
1	0	33	0	530	0	0	3	0
2	0	68	77	139	83	434	37	25
3	72	56	115	225	113	429	126	15
4	147	0	28	235	241	105	8	34

Figure 2. Example of a chromosome with the chosen representation.

To create an initial population, for each gene we generate a random number respecting the corresponding transport bounds. This is done not only due to feasibility issues, but also to prevent the generation of very large numbers that would compromise the individuals at the start of the execution. However, we allow the generation of non-feasible solutions and deal with the infeasibilities in the fitness function. Nevertheless, in the final generation, a repair mechanism is used to ensure feasibility of at least the five most promising solutions. The repair mechanism addresses one constraint type at the time and in the following order using the lexicographic rule: warehouse capacity, demand, retailers' capacity, and transportation limits. Nonetheless, transportation limits are an exception.

The repair mechanism starts by checking if the warehouse capacity is exceeded and if needed applies a coefficient greater than 1 all quantities being sent to the retailers. The reason behind this choice is that sending the over the limit stock to the retailers is more efficient, in terms of costs. The retailers' storage capacity is then checked and the quantities sent are rectified whenever needed. Next, the availability of each product in each retailer is checked in order to ensure that retailers do not run out of stock. If needed additional quantities are sent. In case this condition is not met, the quantities to be received by these retailers needs to be reduced. Lastly, it is ensured that the transport limits are met. If the value is above the upper limit, then some or all quantities to be sent need to be decreased; while if it is below the lower limit the reverse is true. Changes to satisfy these limits may override previous corrections.

3.2. The Fitness Function

The fitness function is used to evaluate the quality of solutions. In this work, this is done by calculating a score value for each solution which is made of two components: one based on the solution cost and another based on the solution infeasibilities. This way we are able to classify the individual’s admissibility and alignment with the objective.

Cost related points are awarded to each solution having a cost below the population average cost; that is, a solution gets 50×m points, where m is the number of facilities (retailers+1).

Feasibility points are awarded per satisfied constraint. Solutions get 10 points for each satisfied constraint, except for the ones regarding demand satisfaction and excessive stock. Regarding the former, 10 points are awarded to each retailer for each product, whenever demand can be fully satisfied; 5 points whenever 90% of the expected demand is covered; and 0 otherwise. Recall that by demand here we mean the 4-weeks expected demand imposed by the business rule. Finally, regarding the excessive stock, if it remains within the 10% allowed, although undesirable, 5 points are awarded. Additionally, transportation limits are treated as an all or nothing, that is, a solution gets 20 points whenever the quantity of a product sent to a retailer falls within both its specified upper and lower limits; or 0 otherwise. Note that there are (3n+2)×(m-1)+1 constraints.

3.3. Generation of New Population

Once all the chromosomes have been evaluated by using the fitness function, the current population is sorted in decreasing order of such value.

To generate the next population, first a percentage of the top individuals, i.e., solutions with the best fitness scores (elites), are copied directly onto the next population. By preserving the best individuals of each generation, we can speed up the performance of the GA and ensure that the best solution found is kept. The rest of the new population is generated using the crossover operator and mutation.

The selection strategy creates a pool with the best scoring solutions and then chooses randomly two chromosomes from this pool. After selecting two chromosomes a two-point crossover strategy (see Figure 3) is used to generate new chromosomes. This is repeated until a predefined number of chromosomes is obtained.

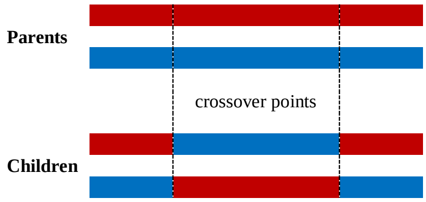

Figure 3. A two-point crossover example

To avoid excessive convergence, we introduce new randomly generated chromosomes, rather than the usual mutation. These new chromosomes are generated in the same manner as the initial population, mimicking the immigration.

Finally, the new generation is obtained by joining together the chromosomes obtained through crossover, the newly generated ones, and the very best chromosomes of the current generation.

4. Computational Experiments

This section reports on the performance of the proposed GA, which is evaluated by comparing its results to those of the current decision strategy of the company. As mentioned before, the specific problem under consideration is faced a Portuguese major player in the fashion industry. The instance solved here involves deciding on a weekly the quantities of eight products to be sent from a single warehouse to 108 retailers, over a 52-week planning horizon. Products are handled and fitted in the boxes prior to shipping. Each box can carry up to 6 units of each product, except for two of them. In the latter case, 12 units can be fitted into a box. The boxes are transported by the trucks, directly, from the warehouse to the retailers. There are limits on transported quantities and on both retailers and warehouse holding capacities that cannot not be exceeded. In addition, holding capacities at both the retailers and the warehouse above a pre-specified limit, which lower than capacity, are allowed but undesirable. The computational tests were carried out on a PC with 64 bits @ 2.2GHz Intel® Core™ i5-5200U CPUs with 6 GB RAM. The algorithm was implemented in Python 3.6.0.

Once a decision has been made regarding the quantities of each product to be sent to each retailer, the products need to be boxed at the warehouse and then loaded on to trucks. Recall that no mixed product boxes are allowed. Handling costs are incurred at the warehouse. Then, the trucks transport the boxes to the respective retailers and transportation costs are incurred. Both these costs are proportional to the number of boxes involved. Upon arrival at the retailers, the boxes are verified, and the products need to be unboxed and properly stored – retailer handling costs. The retailer handling costs are also proportional to the number of boxes received.

Regarding inventory, as said before there are two types of upper limit. Up until the lower one, no additional inventory holding costs are incurred as they are considered sunk costs. However, if the inventory is between this value and the maximum admissible one, then a holding cost is incurred. This holding cost is based on rent cost. More specifically, holding costs are paid for each additional square meter of space required to retailer the boxes and in each square meter up to two boxes can be fitted.

Handling costs are based on the minimum wage ($3.16\ €/h$) and on the fact that on average each box takes about 10 and 20 minutes to be handled at the warehouse and at the retailers, respectively; thus, the values used are 0.53€ and 1.05€, respectively. Storage costs use the zone regulated cost per square meter, which given the retailers location is either 679.35€ or 602.92€. Finally, the transportation costs, as mentioned before, are fixed and per box and were set to 1 euro. Demand varies along the 52-weekss period, as well as with retailer and product.

4.1. Parameter Selection for the GA

Parameter values have been empirically determined by running the algorithm for 25 weeks for 27 different testing scenarios, corresponding to the combinations of three possibilities of crossover, elite, and mutant rates, three population sizes, and three number of generations as follows: crossover, elite, and mutant rates of (0.75, 0.15, 0.1), (0.79, 0.2, 0.01), and (0.8, 0.15, 0.05); population sizes of 30, 40, and 60 solutions; and 200, 800, and 2000 generations.

The best performing four test settings out of the 27 considered are shown in Table 2. These l tests have shown the algorithm to converge quickly, since three out of the four run only for 200 generations.

Figure 4 compares two of the best four test settings (4 and 10) with two other settings (5 and 12) regarding total costs over the 25-weeks period considered in the initial test settings.

As it can be seen, in the first weeks the total costs as well as their increase are very similar for all test settings. However, from week 17 and until the end costs of test 5 become much larger and increase at a faster pace. The same happens with test 12 although the magnitude of the increase is slightly smaller.

Table 2. Top Four Test Out of the 27 Considered

Test setting	Generations	Population	Crossover	Elite	Mutants
2	200	30	0.79	0.20	0.01
4	200	40	0.75	0.15	0.10
8	200	60	0.79	0.20	0.01
10	800	30	0.75	0.15	0.10

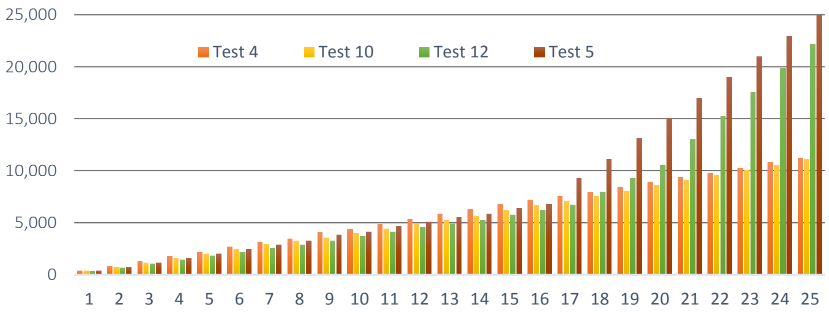

Figure 4. Comparison of two good performing combinations with two underperforming combinations

Another feature that can be observed in this graph is that the costs are always increasing. This, as will be discussed later, is due to the fact that the company is accumulating stock over time, i.e., the quantities purchased are larger than needed and, as a consequence, the costs, mainly the holding costs, are always increasing. However, noting can be done as purchasing decisions are out of the problem being considered; they are indeed an input parameter.

4.2. General Results

As said before, we address the problem by solving each week in the isolation. However, since the decisions of one week impact the following week inventories, there is still a connection between the decisions. Nevertheless, the optimization done is local rather than global, in terms of the time span.

The quality of the solutions obtained, and in particular of the best one, are compared with those of current practice in the company. In particular, comparisons are made regarding weekly costs and accumulated costs over the 52-weeks time horizon, see the graphs in Figures 5 and 6. All results reported refer to averages over the ten runs executed for the four combinations of parameter values corresponding to four test settings described in Table 2.

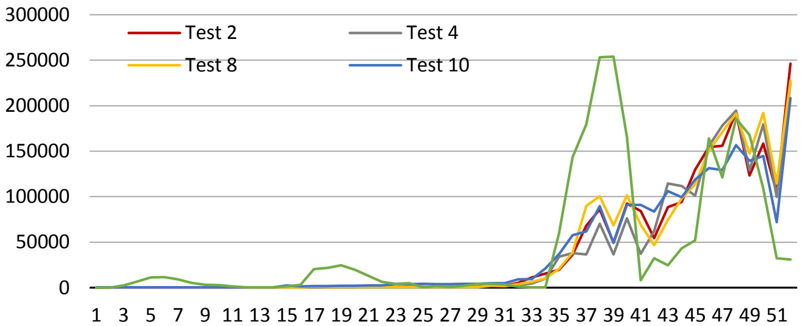

Figure 5. Weekly total costs

As it can be seen, all four combinations outperform current practice. The latter one having higher weekly costs up until week 23 and then in line with the ones of our approach until about week 32. Current practice weekly costs are then lower for a couple of weeks and in week 35 they spike up, increasing by about 14 000%. Finally, from week 41 onwards current practice leads to smaller weekly costs in most weeks. The cumulative weekly costs in our approach, regardless the parameter value settings considered, are always lower. The overall reduction is at least 6.6% (test 8) and can be as large as 12.76% (test 4).

As said before, and due to the stochastic nature of GAs, each test was run 10 times. Table 3 reports the average, minimum, and maximum total cost and its standard deviation, as well as the average computational time required to obtain the solutions reported. Test setting 4 provides an average improvement of 12.76% over current practices, ranging from an astonishing 22.84% to a loss of 12.90%. Therefore, this combination of parameter values seems to be the less robust. Test setting 2, on the other hand, seems to be the most robust one. Although it does not always find a better solution than the current practice, on average it improves on it by 8.46% and in the worst case the solution it finds is only 1.28% worse. Nevertheless, test setting 10 seems to be the most interesting one. In addition to be the one that finds the best solution (32.77% better that of the current practice), it provides an average cost savings of 9.65% with the worst solution found being only 2.38% worse than that of current practice. Finally, test setting 8 seems to be the least interesting one, since it has the smallest average improvement (6.59%), without being the more robust or the one finding the best solution.

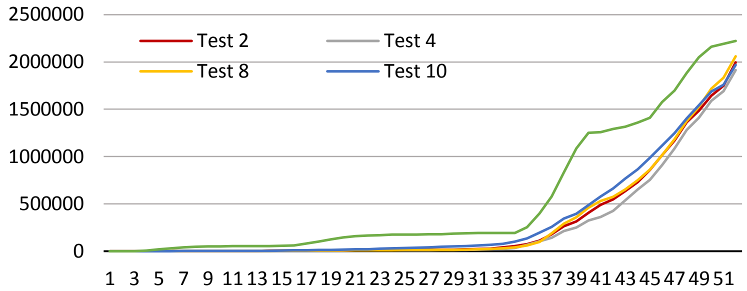

Figure 6. Accumulated total costs.

Table 3. Summary of the Results Obtained with the Best Four GA Parameter Settings

Test setting	Average cost	Minimum total cost	Maximum total cost	Standard Deviation	Average total time (Sec)
2	2,034,499	1,810,830	2,250,966	160,799	3600
4	1,938,824	1,714,728	2,509,217	301,809	9478
8	2,075,908	1,801,760	2,322,256	163,918	7920
10	2,007,966	1,494,040	2,275,324	266,491	14460

To determine the company's current practice policy, a downstream manager spends 7.5 to 10 hours a week analyzing the quantities, of each product, to send to each retailer. Although, the number of products being analyzed by the manager is greater than the number that we used in our algorithm. A downstream manager looks at, at least, 30 products per season. It was not possible to access more data than what was used (i.e., 108 retailers, 8 products, 52 weeks). Furthermore, the results can be used to compare with future work and to further convince the managers that the tool is a good option to what is used today. Table 4 reports the runtime of for each test settings used. To conclude the analysis of the results, we believe that taking into account the computational time requirements, test settings 2 is the most reliable. With the tests that were executed, despite not yielding the best results, it has the lowest variance and runs faster than the remaining ones, providing a solution in about one hour.

Table 4. Computation Runtime

Test setting	Iterations per Sec	Generations	Time (Sec) per iteration	Total Time (Sec)
2	2.85	200	70	3600
4	1.10	200	182	9478
8	1.31	200	153	7920
10	2.87	800	279	14460

5. Conclusions

This work addresses a multiproduct distribution problem that is closely related to the direct shipping inventory routing problem. The problem being solved is a real application of the apparel industry. Due to the combinatorial nature of the problem a genetic algorithm approach was proposed. In the GA, although the population is generated at random, the random values are limited by the transportation quotas as lower and upper bounds. By seeding this information, the individuals achieve good fitness scores at the early stages. Further, a specific score system to evaluate the chromosomes is proposed. The individuals are given points not only for fulfillment of the objective function, but also for not violating constraints of the problem. Thus, admissibility of a solution can also be assessed throughout the execution of the algorithm. Although, at the end of each run (a week), a repair function is used to rectify any deviations from the constraints of the problem and thus, enforce feasibility.

The solutions obtained have been compared to those of the company's current practice. In addition, to find them quicker and autonomously, we were able to obtain total costs improvements that on average are 6.59% to 12.76% lower, depending on the GA parameter values. Although in about 20% of all solutions obtained by the GA, the total cost incurred is 5.27% higher, in the remaining 80% the total cost was improved by an average of 12.93%, ranging from 0.67% up to 32.78%.

Future work already undergoing aims at solving the mixed integer linear programming model developed, either exactly or approximately, perhaps via the relaxation of some integer variables.

Acknowledgments: In this research, we acknowledge the support of ERDF - European Regional Development Fund through the Operational Programme Competitiveness and Internationalization - COMPETE 2020 and by national funds through the Portuguese agency, FCT – Fundação para a Ciência e Tecnologia within projects NORTE-01-0145-FEDER-000020, POCI-01-0145-FEDER- 031821 and PTDC/EEI-AUT/2933/2014-POCI-01-0145-FEDER-016858.

References:

Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, *104*, 124-133.

Bertazzi, L., & Speranza, M.G. (2012). Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, *1*, 307.

Campbell A., Clarke L., Kleywegt A., & Savelsbergh M. (1998). The Inventory Routing Problem. In: Crainic T.G., Laporte G. (Eds.), *Fleet Management and Logistics.* Centre for Research on Transportation. Springer, Boston, MA

Coelho, L. C., Cordeau, J. F., & Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, *48*(1), 1-19.

Coelho, L. C., Cordeau, J. F., & Laporte, G. (2014). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, *52*, 55-67.

Cordeau, J. F., Laganà, D., Musmanno, R., & Vocaturo, F. (2015). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, *55*, 153-166.

Dong, Y., Maravelias, C. T., Pinto, J. M., & Sundaramoorthy, A. (2017). Solution methods for vehicle-based inventory routing problems. *Computers & Chemical Engineering*, *101*, 259-278.

Gallego, G., & Simchi-Levi, D. (1990). On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer R-systems. *Management Science*, *36*, 240-243.

Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of manufacturing systems*, *42*, 93-103.

Holland, K., & John, H. (1975). *Adaptation in Natural and Artificial Systems.* MIT Press.

Jahangir, H., Mohammadi, M., Pasandideh, S. H. R., & Nobari, N. Z. (2019). Comparing performance of genetic and discrete invasive weed optimization algorithms for solving the inventory routing problem with an incremental delivery. *Journal of Intelligent Manufacturing*, *30*(6), 2327-2353.

Lenstra, J. K., & Rinnooy Kan, A. H. G. (1981). Complexity of Vehicle Routing and Scheduling Problems. *Networks*, *11*(2), 221-227.

Li, K., Chen, B., Sivakumar, A. I., & Wu, Y. (2014). An inventory–routing problem with the objective of travel time minimization. *European Journal of Operational Research*, *236*(3), 936-945.

Malladi, K. T., & Sowlati, T. (2018). Sustainability aspects in Inventory Routing Problem: A review of new trends in the literature. *Journal of Cleaner Production*, *197,* 804-814.

Moin, N. H., & Salhi, S. (2007). Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, *58*(9), 1185-1194.

Peres, I. T., Repolho, H. M., Martinelli, R., & Monteiro, N. J. (2017). Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. *International Journal of Production Economics*, *193,* 748-756.

Rahimi, M., Baboli, A., & Rekik, Y. (2017). Multi-objective inventory routing problem: A stochastic model to consider profit, service level and green criteria. *Transportation Research Part E: Logistics and Transportation Review*, *101*, 59-83.

Roldán, R. F., Basagoiti, R., & Coelho, L. C. (2017). A survey on the inventory-routing problem with stochastic lead times and demands. Journal of Applied Logic.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, *89*, 168-182.

Waller, M., Johnson, M. E., & Davis, T. (1999). Vendor-managed inventory in the retail supply chain. *Journal of business logistics*, *20*, 183-204.

Bruno Cretú
Faculdade de Economia da Universidade do Porto, Porto, Portugal
brunocretu.pte@gmail.com

Dalila B. M. M. Fontes
Faculdade de Economia da Universidade do Porto, Porto, Portugal;
LIAAD- INESC TEC, Porto, Portugal
dfontes@inesctec.pt

Seyed Mahdi Homayouni
LIAAD- INESC TEC, Porto, Portugal;
Department of Industrial Engineering, Lenjan Branch, Islamic Azad University, Esfahan, Iran
homayouni@iauln.ac.ir