The Capacitated Lot Sizing Problem with Batch Ordering: A MILP and Heuristic Approach

Gustavo Macedo-Barragán[1], Samuel Nucamendi-Guillén[1], Elías Olivares-Benitez[1] and Omar G. Rojas[2]

[1]*Universidad Panamericana, Facultad de Ingeniería, Prolongación Calzada Circunvalación Poniente 49, Zapopan, Jalisco, Mexico*

[2]*Universidad Panamericana, Escuela de Ciencias Económicas y Empresariales, Prolongación Calzada Circunvalación Poniente 49, Zapopan, Jalisco, 45010, Mexico*

Keywords: Batch Ordering, Heuristic, Mixed Integer Linear Programming, Lot Sizing, Multiple-product, Retail, Single Level.

Abstract: This article presents a mixed integer linear programming method and a heuristic algorithm to deal with the problem of multi-period, multiple-product batch purchases, with a finite time horizon, considering delivery times, order placement costs and independent batch size for each product. The objective of this problem is to minimize the costs of placing purchase orders and inventory. This problem is motivated by its application in a marketing company that handles the sale of fashion products (footwear and accessories) through catalogs and for which excess inventory represents a major problem given the short life cycle of its products. Experimental results show that the heuristic algorithm is able to obtain feasible solutions that improve in cost by up to 37% the best integer solutions reported by the model when it reaches the time limit. To validate the efficiency of the algorithm, a real scenario was solved for a trading company, obtaining results that improve by 28% compared to the current company's situation. These results show that the heuristic approach is promising in terms of the quality of the solution and the computational time required.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

At present, companies are immersed in a globalized market full of varied demand for products. The customers go in search of a quality product that meets their requirements and is also easy to access. Therefore, companies compete to adapt and satisfy each client's preferences, seeking new strategies such as optimal inventory management (Christopher et al., 2004).

Inventory control is a medullar part for the company's management since a large number of resources are used to buy merchandise that will later give profits to the company. But to be able to buy all such materials it is necessary to store them and place purchase orders that also have an economic cost.

This work was motivated by a real problem faced by a company that has over 28 years in the market selling clothing, footwear and accessories by catalog. The firm handles around 465 different SKUs per season. The company has a large amount of data that helps with the understanding of their products' demand. This information is used by employees to carry out the planning of the purchases and to satisfy such a demand. The problem with this practice is that it is carried out manually and supported by empirical *sui generis* methods of older, more experienced, workers. Coupled with this problem, each supplier has its own delivery time and, in addition, to be able to offer competitive prices, they establish that orders shall be in batches, which complicates the purchase and inventory management mechanism. That's why the company identified the need to incorporate efficient methods to determine the appropriate purchase program for each item so that the demand is always met and the weekly budget and inventory capacity are not exceeded.

In this work, a mixed integer linear programming (MILP) model for inventory management is presented in order to improve management's current situation. Alternatively, a heuristic algorithm that obtains high-quality solutions in reasonable computational times is presented, which allows the company to make decisions based on information ob-

Macedo-Barragán, G., Nucamendi-Guillén, S., Olivares-Benitez, E. and Rojas, O.
The Capacitated Lot Sizing Problem with Batch Ordering: A MILP and Heuristic Approach.
DOI: 10.5220/0007297402290236
In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES 2019)*, pages 229-236
ISBN: 978-989-758-352-0
Copyright © 2019 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

tained in "real-time".

2 LITERATURE REVIEW

According to DeHoratius (2008), inventory management pursues two fundamental objectives: 1) Minimize inventory levels and 2) Ensure stock availability. In particular, the fashion market shows peculiar characteristics such as short life cycles, volatility, poor predictability and shopping behavior by critical mass (Christopher et al., 2004). Liu et al. (2013) state that optimal management of inventory by retailers depends to a large extent on accuracy in predicting future demand.

For the economic problem of lot size, Wagner and Whitin (1958) designed an algorithm to solve it. Due to the large number of applications that this model has, several authors, such as Chowdhury et al. (2018), have developed an algorithm where lists are used and data structures are stacked to find the optimal solution. O'Neill and Sanni (2018) propose a model that seeks to manage efficiently the inventory of a single product in order to maximize the profit; since this problem seeks to increase profit, the price of the item and the replenishment time of the inventory must be found so that the demand can be met. Albrech (2017) presents a heuristic that uses item-based approximations, this can be very useful especially for retail companies. Similarly, Gruson, Cordeau, and Jans (2018) analyze how service level constraints affect lot size problems where there is a deterministic demand in a finite planning horizon. Torkul et al. (2016) propose a model of inventory in real-time where orders are made once delivery time to the customer and replenishment time of the supplier are the same; in this way, it is necessary to have only the items that are going to be needed and the amount of product to be stored is decreased.

In most cases, suppliers try to get customers to place their orders in large quantities by offering better purchase prices. Archetti, Bertazzi, & Speranza (2014) determined a balance point where the goods are bought at good prices but do not generate an increase in storage costs. Later, Alfares and Ghaithan (2016) developed a model that seeks to determine the size of the purchase orders and the sale price for the item and, upon modifying these two variables, what the purchase price of the items will be. Furthermore, Goisque and Rapine (2017) developed an algorithm that seeks to cover a deterministic demand in its entirety, in order to reduce the costs of storage and production at the two levels that it models: the manufacturer and the retailer.

In the case of some warehouses, the storage capacity will not always be the same throughout the planning horizon. That is why Tapia-Ubeda, Miranda and Macchi (2018) model this situation in a stochastic way and solve it using what they call *Generalized Bender Decomposition* with which they ensure optimality. Previously, Guan, and Li (2010) presented two models of lot size where the first only has a restriction on the storage capacity and the second adds a constraint on the quantity of orders. Fan and Wang (2018) model a lot size problem that seeks to reduce expenses; to this end, costs for ordering and storing are taken into account. It is also allowed to modify the size of the warehouse, incurring a cost for each change in capacity.

Yang et al. (2014) analyzed how to optimize inventory management of a single product that is purchased in batches. Similar to the case under study in the present document, the placement cost is taken into account and the warehouse capacity is limited. The case of a product acquired by lots can also be combined with a stochastic demand and achieved by means of a heuristic as developed by Zhu, Liu and Chen (2015). Akbalik et al. (2017) propose an algorithm to supply a company that buys its products in batches and where the storage cost varies in case of exceeding the determined capacity. Farhat et al. (2017) study the purchase in batches with the option of returning those items that were not sold to the supplier. In some cases, products purchased in batches are perishable; due to this, they must be treated with special care and the model must consider a series of constraints such as those proposed by Broekmeulen, and van Donselaar (2009).

To the best of the author’s knowledge, and after reviewing several articles related to the research presented here, an optimal approach for the problem of capacitated lot sizing considering variable batch sizes and variable delivery times applied to retail sales was not found in the literature. Although many of the articles consider discounts when buying in large volumes, in the model presented here, a fixed unit cost will be maintained because this is what the company handles. The aforementioned problem and the results obtained might be of interest to both academics and practitioners.

3 PROBLEM STATEMENT

3.1 Characteristics of the Problem

Formally, the problem under study has the following characteristics:

- A set of N items.
- A planning horizon of T weeks (periods).
- A weekly demand (D_{it}) for the item i in the week t .
- An ordering cost (S_i) for the purchase order of item i.
- Inventory cost (h_i) for item i (same for all periods).
- Unitary purchase cost (a_i) for item i (same for all periods).
- A weekly budget (for the period) P_t.
- Lead time (LT_i) determined for the item i.
- Batch size (BS_i) (in units) for item i.
- Maximum inventory size (R_i) in units for item i.
- A shifted time-horizon (ST).

An equal maximum inventory size is defined for all products since they have a similar volume. In addition, the warehouse constantly changes its availability due to it is also used to store other supplies.

3.2 Mathematical Formulation

The batch size indicates the number of products taken together for item i. On the other hand, the lot size refers to the total number of batches in the purchase order for all items. Thus, the quantities to order shall be merely an integer multiplier of the batch size of each product. The retailer also owns a capacitated warehouse. It is desirable to make decisions about "what products to order", "how much to order for each product" and "when to order".

Once the purchase order is placed, the arrival of the products will delay a certain number of periods according to their LT. To be able to represent the lead time of those purchases that are made to cover the first week of demand, a sufficient number of periods are added at the beginning of the planning horizon in order to place the purchases. The original planning horizon plus the weeks that were added are known as shifted time-horizon (ST).

Decision Variables:

X_{it} = Quantity of batches of item i to purchase in period t.

I_{it} = Quantity of units of item i to hold in inventory in period t.

$$B_{it} = \begin{cases} 1 & if\ purchase\ order\ for\ item\ i\ is\ placed\ in\ period\ t, \\ 0 & otherwise \end{cases}$$

Z = Total cost incurred by purchase and inventory.

Objective function:

$$\min Z = \sum_{i=1}^{N} \sum_{t=LT_{max}+1}^{ST} \left(S_i\, B_{i(t-LT_i)} + h_i\, I_{it} \right) \tag{1}$$

Subject to:

$$I_{i(t-1)} + X_{i(t-LT_i)}\, BS_i - D_{it} = I_{it} \quad \forall\ i \in N, t \in ST; t > LT_{max} \tag{2}$$

$$I_{it} \leq R_i \quad \forall\ i \in N, t \in ST; t > LT_{max} \tag{3}$$

$$I_{it} = 0 \quad \forall\ i \in N, t \in ST; t \leq LT_{max} \tag{4}$$

$$\sum_{i=1}^{N} X_{i(t-LT_i)}\, BS_i\, a_i + S_i\, B_{i(t-LT_i)} \leq P_t \quad \forall\ t \in ST; t > LT_{max} \tag{5}$$

$$X_{i(t-LT_i)} \leq \left(1 + \frac{\sum_{t=LT_{max}+1}^{ST} d_{it}}{BS_i} \right) B_{i(t-LT_i)} \quad \forall i \in N, t \in ST; t > LT_{max} \tag{6}$$

$$\sum_{t=1}^{ST} B_{it} \leq \frac{T}{LT_i} \quad \forall\ i \in N \tag{7}$$

$$I_{it} \geq 0 \quad \forall\ i \in N, t \in T$$

$$X_{it} \in \mathbb{N}^* \quad \forall\ i \in N, t \in T \tag{8}$$

$$B_{it} \in \{0,1\} \quad \forall\ i \in N, t \in T$$

In this model, the objective function (1) aims to minimize the total cost incurred by placing purchase orders and inventory. Equation (2) makes reference to the fact that the demand of item i in period t must be equal to the sum of the inventory of the previous period plus the products that arrive in the current period minus what remains of surplus in the current period. Equation (3) controls the maximum inventory allowed for each item in each period, while equation (5) controls the total order size based on the weekly budget. On the other hand, equation (4) ensures that during the lag period no inventory is generated. Equation (6) determines, based on the batch size for each item, the maximum quantity of lots allowed (in general) to buy for the item i. Equation (7) allows to determine, according to the lead time of each item, the maximum number of purchase orders allowed for the item i during the planning horizon. Finally, the group of equations (8) denotes the nature of the variables.

To provide an illustration of how the model works, the following example is presented: Consider the case of having 3 products (A, B and C) to purchase and a time-horizon of 6 periods. The lead times for the products are 2, 1 and 3 respectively. thus, the value for $LT_{max} = 3$. After solving the model, an example of the placement of purchases during the previous periods for the first demand period is shown in Table 1:

Table 1: Example of purchases with the shift.

Item / Period	1	2	3	4	5	6	7	8	9
1	0	2	0	0	1	2	0	0	0
2	0	0	1	2	0	0	2	0	0
3	3	0	0	3	0	0	0	0	0

In Table 1, it is shown that during the first 3 periods, purchases were made to satisfy the first period of demand set in the fourth stage.

As we can see, the model is in charge of deciding which purchasing policy favors the reduction of operating costs.

According to Akbalik, et al. (2017), the nature of the problem is NP-hard, even for the single item case, a reason for which the MILP model may not able to solve the problem for which it was developed in a reasonable computational time. As an alternative, a greedy heuristic to deal with large size instances is proposed.

3.3 Greedy Heuristic Procedure

For this problem, it must be contemplated that in order to solve the problem in a commercial optimizing engine it would be necessary to pay for the license and train people to use it, which increases the expenses, while a heuristic has the benefit of being coded in any language and compiled using free license Integrated Development Environments.

This heuristic algorithm is adapted from the algorithm presented by Maes (1986) and consists of three mechanisms: classification based in the time between orders (TBO), assignment and procurement of potential advanced purchases (consolidate the lot size) after calculating the purchase cost (PC_t) for the current period. These three mechanisms are described in Algorithm 1.

Algorithm 1: Pseudo-code for the proposed heuristic.

Input: Data Base
Output: Purchasing and Inventory Program

1. Mechanism classification based on TBO
2. For each period $t \in ST$
3. Apply search mechanism for purchasing
4. If (PC_t <P_t) Calculate savings for advancing future orders using the Silver-Meal criteria
5. Conformation of lot size for period t
6. End t
7. Report the program of orders and inventories

3.3.1 Mechanism of Classification based on TBO

For the case of heuristics, the ordering of the items considers the time between orders (*TBO*). This ordering is carried out from highest to lowest. This arrangement allows to first seek to advance purchases for those items that have a longer delivery time.

$$TBO_i = \sqrt{\frac{2\, S_i\, T}{H_i \sum_{t=LT_{max}+1}^{ST} D_{it}}} \quad \forall i \in N \tag{9}$$

3.3.2 Search Mechanism for Purchases by Period and Calculation of Savings

The next step in the heuristic procedure is to know and separate the amount of money needed to cover the demand of the period under analysis. Subsequently, and if there is still a budget available in the current week (CW), it is sought to advance purchases respecting the lexicographical order of appearance for the items. When making advances on an item, it is sought to cover as much as possible using the Silver-Meal (SM) criterion and checking not to exceed the budget or inventory capacity, as shown in eq. (10).

$$\frac{s_i + H_i \sum_{t=CW}^{CW+r}(t - CW)\left(d_{it} - I_{i(t-1)}\right)}{r+1} \le \frac{s_i + H_i \sum_{t=CW}^{CW+r-1}(t - CW)\left(d_{it} - I_{i(t-1)}\right)}{r} \tag{10}$$

The SM criterion indicates how many periods can be advanced by averaging the operating expenses. Index r is an integer that denotes the periods to be overtaken. To know how many periods would be advanced, the value of r is increased consecutively starting from 1 until finding the greatest value that satisfies the inequality (10), once the inequality is not

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

met, its value should not be increased further. Another stopping criterion, is that r can't be bigger than $ST - CW$. The appearance of the inventory aims to reduce the value of the demand, but consequently, it is necessary to update it from one period to another. This will cause it to decrease through the periods with the lower limit being the value of zero.

3.3.3 Lot Size Conformation Mechanism based on the Budget

The lot size order for the period in question will consist of the number of batches needed to cover the current period plus the overtaken ones, based on the criteria explained in the previous section. The lot size maintains the viability in terms of budget and maximum inventory. It is important to mention that the purchase order is scheduled with enough anticipation to cover the periods and to avoid shortage until the next order arrives.

In the next section, we report the experimental results obtained.

4 EXPERIMENTAL RESULTS

The objective of this section is to make a comparison between the results obtained with the model and those obtained with the heuristic.

4.1 Test Instances

To evaluate the performance of the formulation, small instances were generated by randomly choosing subsets of products, from the original problem, respecting their parameter values (demand, LT, BS, configuration, purchase and maintenance costs). As a result, 12 different instances of sizes 10, 20 and 30 were created (4 instances per size). All the experiments were executed in a PC with AMD E1-2500 processor @ 1.40 GHz with 4 GB of RAM memory with Windows 10 operating system. In the case of the formulation, this was solved using LINGO as an optimizer.

4.2 Comparison of the Results of the Model Vs the Heuristic Procedure

As a first approximation, it was decided to evaluate the resolution capacity of the model in terms of the size of the instance to be solved (only the quantity of items varied). The results of this experimentation are shown in section 4.2.1.

A second approach is to solve the case study. Due to the high complexity of the scenario for the model (in terms of the number of variables or items), it was only solved by the heuristic algorithm. The results obtained are reported in section 4.2.2.

4.2.1 Results for the Test Instances

The small instances were used to compare the results obtained through the model and the heuristic. The notation followed is "X-Y", where X denotes the number of items to analyze while Y indicates the number of scenarios. For all instances, a time horizon of 21 weeks is considered. The results obtained are reported in Table 2. The first column displays the name of the instance, while columns 2 and 4 report the best integer solution obtained by the model (*BSF*) and the heuristic (*BSH*), respectively. Finally, columns 3 and 5 show the Gap and percentage of improvement (%IMP), respectively.

Table 2: Comparison of results.

Instance name	BSF	% Gap	BSH	% IMP
10-1	$ 240,572.10	19.24	$ 247,843.22	-3
10-2	$ 289,740.69	38.35	$ 258,946.31	11
10-3	$ 230,528.85	19.31	$ 234,153.85	-2
10- 4	$ 184,613.20	10.05	$ 195,425.70	-6
20-1	$ 465,488.20	41.39	$ 473,609.95	-2
20-2	$ 581,353.59	91.00	$ 460,754.84	21
20-3	$ 621,747.58	**100**	$ 391,210.08	37
20-4	$ 643,663.88	**100**	$ 424,523.88	34
30-1	$ 736,259.16	53.02	$ 704,887.29	4
30- 2	$ 913,004.60	**100**	$ 680,633.35	25
30-3	$ 767,168.65	73.62	$ 691,104.90	10
30-4	$ 706,989.09	55.96	$ 677,200.38	4
Mean				**11.08**

As it can be observed, for the case of the formulation, no instance could be solved to optimality within the time allowed (2 hours). The *GAP* column is calculated as the percentage difference between the lower bound obtained (*LB*) against the best total solution obtained.

$$\%Gap = \frac{100\,(BSF - LB)}{LB} \tag{11}$$

We can see that generally, the *Gap* tends to be smaller in the instances of smaller size and to increase significantly for the largest instances. This may be because, by increasing the number of items, the model must analyze a greater amount of information and different possibilities for purchases along the planning horizon.

As a heuristic, the quality of the solution obtained *(BSH)* is measured by the percentage of improvement (IMP) respect to the best integer solution reported by the model using the following equation:

$$\%IMP = \frac{100\;(BSF - BSH)}{BSF} \tag{12}$$

Results obtained for each instance are reported in the %IMP column. In the case of negative values, this indicates that the algorithm obtained a worse solution than the one reported by the model.

In general, the heuristic presents an irregular behavior for the first 5 instances. Subsequently, its performance is stabilized by obtaining better results for larger instances (up to 37% improvement).

Another aspect to consider in the analysis of the heuristic performance is the execution time. The elapsed CPU time per instances is displayed in Fig.1.

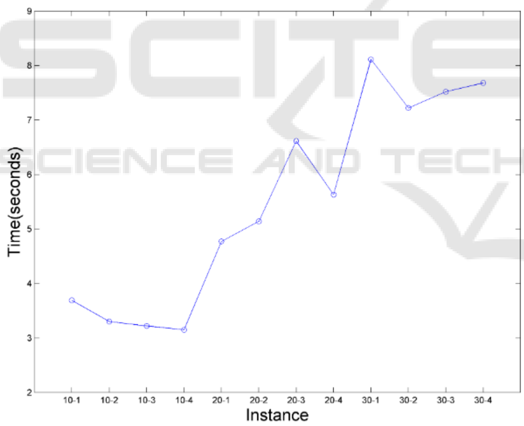

Figure 1: Elapsed CPU times per instance.

As it can be seen, the heuristic spent around 8 seconds to solve the largest instances. In summary, the heuristic outperforms the behavior of the model in both the quality of the solution and the time spent.

For instance, 10-1, comparative graphs are also presented to allow visualizing in detail the order schemes proposed both by the model and by the heuristic algorithm. Figures 2, 3 and 4 displays this comparison.

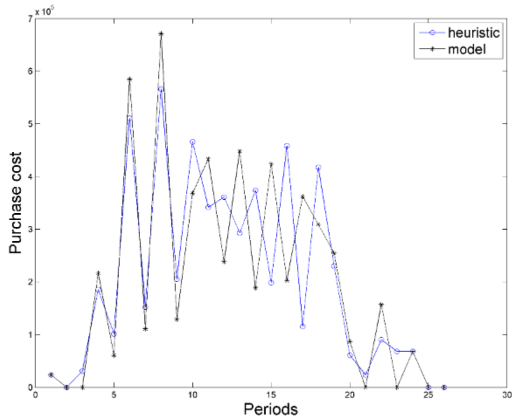

Figure 2: Apportionment of purchase costs for instance 10-1.

In Figure 2, it is observed that both the model and the heuristic propose large initial purchases at the beginning of the season and decrease the amount of money spent as the season progresses.

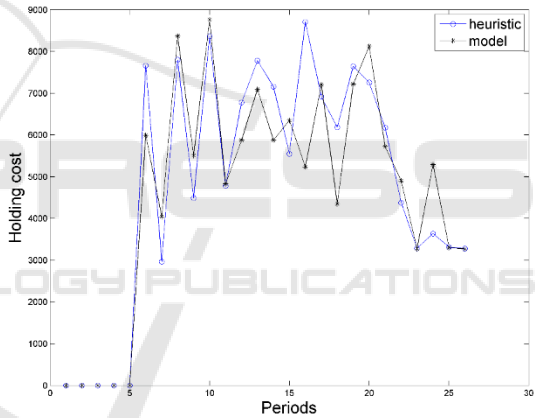

Figure 3: Apportionment of inventory costs for instance 10-1.

Figure 3 presents the total inventory cost for each week. It can be noted that the inventory at the end of the period represents 56% of the average inventory cost.

Figure 4 compares the amount of money spent every period and the budget average, the model reports periodical purchases than save around 52% of the total costs while the heuristic produces savings for 48%.

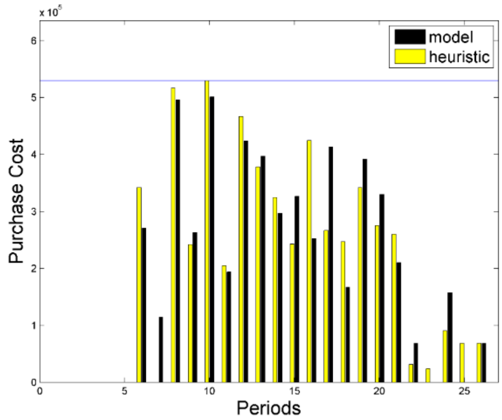

Figure 4: Apportionment of budget spent for instance 10-1.

4.2.2 Results for the Case Study

As explained previously, given that the effective performance of the heuristic is validated for larger instances, it is now implemented to solve the case study that motivated its development. The results obtained are shown in Figures 5, 6, 7.

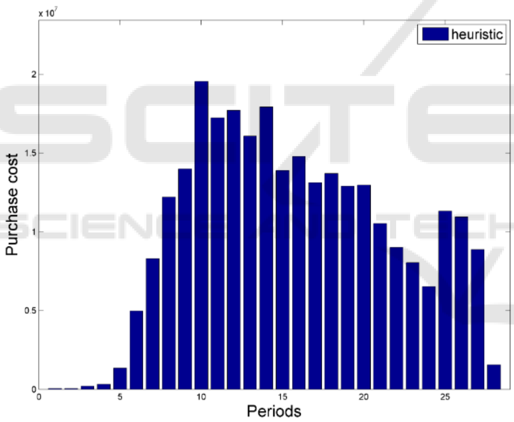

Figure 5: Apportionment of purchase costs for the case study.

As observed in Fig. 5, the purchases of the first weeks are very scarce because the items bought are those with large delivery times. The costs related to the inventory are broken down by periods (week) in Fig. 6.

In addition, the low cost in the last period is favorable for the company and indicates a good formulation of the algorithm. Finally, the amount of purchase in units for each period is presented in Fig. 7. Also, by minimizing the number of purchase orders placed for each item, this part of the cost can become negligible and the company can focus more on the configuration of the items to be purchased in each period.

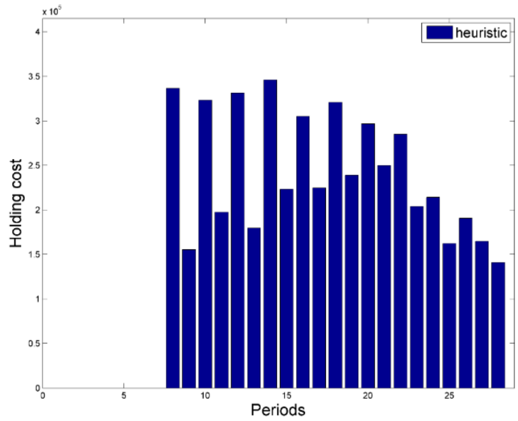

Figure 6: Apportionment of inventory costs for the case study.

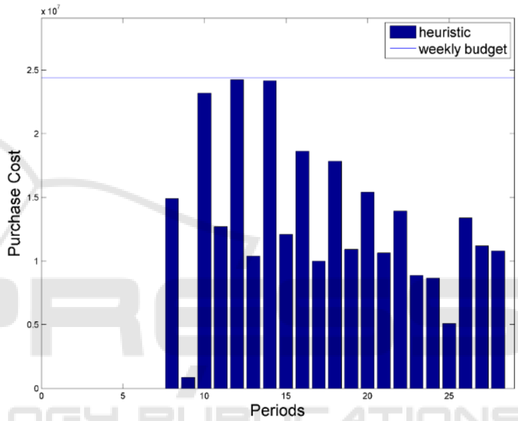

Figure 7: Apportionment of budget spent on the case study.

Finally, the inventories obtained by the company are compared against the inventory proposed by the heuristic at the end of the season. In the real situation, the inventory cost for the final week equals to $195,778.5 MXN, while the heuristic produced a final inventory cost of $140,866.75, According to this information, the heuristic improves the final cost of inventory by 28.04%.

5 CONCLUSIONS AND FUTURE WORK

In the case of the mathematical formulation, it was evident that none of the instances could be solved optimally within the time-limit, obtaining feasible solutions that deviated up to 100% of the lower bound reported.

As an alternative, a heuristic algorithm was developed. It was capable of generating high-quality solutions in a reasonable computational time. When

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

comparing it against the MILP model, it can be observed that in the small instances it obtains worst solutions than those achieved by the formulation, while in the case of the larger instances, the algorithm provides better solutions, reaching an improvement up to 37%. Another point of comparison is the elapsed CPU time to find the solution. It was shown that the time spent by the heuristic is quite fast and it is still more efficient for large size instances.

Regarding the case study, the algorithm was able to obtain a structure of purchases that improve the final inventory with respect to the real situation of the company by 28%. It is important to emphasize that, because items are acquired by batches, it would be very difficult to get rid of all the inventory at the end of the planning horizon.

Future works include modeling scenarios with stochastic demand and variable purchase cost. The development of metaheuristics is also proposed to improve the quality of the solution obtained without significantly worsening the execution time. Additionally, it is proposed to consider more realistic problems involving penalty costs for purchases of fractions of batches, over stock in the last period and allowing for non-compliance in some items.

REFERENCES

Akbalik, A., Hadj-Alouane, AB, Sauer, N., Ghribi, H. 2017. NP-hard and polynomial cases for the single-item lot sizing problem with batch ordering under capacity reservation contract. *European Journal of Operational Research* 257 (2017) 483–493.

Albrech, M. 2017. Optimization of safety stocks in models with an order service level objective or constraint. *European Journal of Operational Research* 263 (2017) 900–909.

Alfares, HK, Ghaithan, AM 2016. Inventory and pricing model with price-dependent demand, time-varying holding cost, and quantity discounts. *Computers and Industrial Engineering* 94 (2016) 170–177.

Archetti, C., Bertazzi, L., M. Grazia Speranza 2014. Polynomial cases of the economic lot sizing problem with cost discounts. *European Journal of Operational Research* 237 (2014) 519–527.

Broekmeulen, RACM, Van Donselaar, KH 2009. A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand. *Computers and Operations Research* 36 (2009) 3013 – 3018.

Chowdhury, NT, Baki, MF, Azab, A. 2018. Dynamic Economic Lot-Sizing Problem: A new O(T) Algorithm for the Wagner-Whitin Model. *Computers and Industrial Engineering* 117 (2018) 6–18.

Christopher, M., Lowson, R., and Peck, H. 2004. Creating agile supply chains in the fashion industry. *International Journal of Retail and Distribution Management*, 32(8), 367-376.

DeHoratius, N., Mersereau, A. J., and Schrage, L. (2008). Retail inventory management when records are inaccurate. *Manufacturing and Service Operations Management*, 10(2), 257-277.

Fan, J., Wang, G. 2018. Joint optimization of dynamic lot and warehouse sizing problems. *European Journal of Operational Research* 267 (2018) 849–854.

Farhat, M., Akbalik, A., Sauer, N., Hadj-Alouane, A. 2017. Procurement planning with batch ordering under periodic buyback contract. *IFAC PapersOnLine* 50-1 (2017) 13982–13986.

Goisque, G., Rapine, C. 2017. An efficient algorithm for the 2-level capacitated lot-sizing problem with identical capacities at both levels. *European Journal of Operational Research* 261 (2017) 918–928.

Gruson, M., Cordeau, JM., Jans, R. 2018. The impact of service level constraints in deterministic lot sizing with backlogging. *Omega* 79 (2018) 91-103.

Guan, Y., Li, T. 2010. Stochastic lot-sizing problem with inventory-bounds and constant order-capacities. *European Journal of Operational Research* 207 (2010) 1398–1409.

Liu, N., Ren, S., Choi, T.-M., Hui, C.-L. and Ng, S.-F. 2013 'Sales forecasting for fashion retailing service industry: a review', *Mathematical Problems in Engineering,* 2013.

Maes, J., and Van Wassenhove, LN 1986. A simple heuristic for the multi item single level capacitated lot sizing problem. *Operations research letters,* 4(6), 265-273.

O'Neill, B., and Sanni, S. (2018). Profit optimisation for deterministic inventory systems with linear cost. *Computers and Industrial Engineering* 122 (2018) 303-317.

Tapia-Ubedaa, FJ, Miranda, PA, Macchi M. 2018. A Generalized Benders Decomposition based algorithm for an inventory location problem with stochastic inventory capacity constraints. *European Journal of Operational Research 267 (2018) 806–817.*

Torkul, O., Yılmaz, R., Selvi, IH, Cesur, MR 2016. A real-time inventory model to manage variance of demand for decreasing inventory holding cost. *Computers and Industrial Engineering 102 (2016) 435–439.*

Yang, Y., Yuan, Q., Xue, W., Zhoud, Y., 2014. Analysis of batch ordering inventory models with setup cost and capacity constraint. *Int. J. Production Economics* 155 (2014) 340–350.

Wagner, HM, and Whitin, TM 1958. Dynamic version of the economic lot size model. *Management Science*, 5(1), 89–96. http://dx.doi.org/10.1287/mnsc.5.1.89.

Zhu, H., Liu, X., Chen, Y., 2015. Effective inventory control policies with a minimum order quantity and batch ordering. *Int. J. Production Economics* 168 (2015) 21–30.