The use of Hybrid Models to Solve Manufacturers Resource Planning

Jorge Ivan Romero-Gelvez , Edisson Alexander Delgado-Sierra, Jorge Aurelio
Herrera-Cuartas, and Olmer Garcia-Bedoya
Universidad de Bogotá Jorge Tadeo Lozano, Bogotá, Colombia
{jorgei.romerog@utadeo.edu.co}

ABSTRACT: *The purpose of this work is to contribute to the extended use of hybrid models to solve MRP issues dealing with stochastic demand over main stock-keeping units. The methodology development first apply SARIMA (Seasonal Autoregressive Integrated Moving Average Model), Long short term memory networks and Fb-prophet as forecasting methods to predict the demand for the master production schedule, next applies an integer programming model with JuMP (Julia Mathematical programming) for solving the MRP using the Lot for lot approach (L4L). The main contribution of this work is to show a way to solve dynamic demand problems over the Forecasting-MIP approach.*

Keywords: SARIMA, LSTMN, Fb-prophet, Forecasting, MIP, JuMP

Received: 24 September 2019, Revised 12 December 2019, Accepted 20 December 2019

DOI: 10.6025/jitr/2020/11/2/55-66

1. Introduction

The heart of a Material Requirements planning system is the demand forecasting for the main stock-keeping-unit. However, it is a complicated, time spending and inaccurate task. The randomness in the forecast has led to improve them using a variety of tools, such as the Box-Jenkins methodology [3], machine learning techniques [2], and even to change the paradigm of their use in Material Requirements planning [23]. In forecasting we notice the lot-sizing techniques and their performance [15] and Box-jenkins implementations for MRP had been used for MRP by several authors, [5], but the case study had no values for cost maintain inventory or cost of order. Lee work over eoq lot-sizing calculation errors [16] and there are several applications in Order-Up-To (OUT) [17],machine learning [22].

The incremental use of computers in the 1960's industry allows them to improve complex scheduling and inventory control [12], which leads to Orlicki and others to develop a manufacture planning approach based on an independent demand over main items and a dependent demand over the material [21].

The development and adoption of Material requirements planning were slow but in 1972 the American production and inventory control society APICS starts to implement this model and initiates the mrp crusade that continues in our time [18]. The early development of MRP (today also know as “little-mrp” or MRP-I) refers to a model that comes from a production plan and focuses over get the optimal requirements for every Stock keeping unit involved in the bill of manufacture (also called BOM), [29]. The MRP improvement years later is called MRP II, but the letters MRP in MRP II stand for Manufacturing Resources Planning to make it clear that resources are considered in addition to materials as in mrp. The word “resource” is used to emphasize that any type of productive capability can be considered, not just machines. This work uses only the MRP I model, focus the effort in the forecast of demands and manage large sets of data for every SKU in the company portfolio of products.

2. Literature Review and Basic Background

This section describes the basic concepts of Material requirement planning, Demand Driven Material requirement planning and Forecasting techniques needed to solve the dynamic demand problem presented in the case study.

2.1. Fundamentals of Material Requirement Planning

This item first describes every part in a production system and later focuses on MRP implementation. A production plan describes in detail the quantity of principal sku (final product to sell) and produced in subsets of sku, the exact time of production and lot sizing. The production plan can be divided in master production schedule (MPS), Material requirement planning (MRP) and the detailed plan of jobs in the production floor. See Figure 1.

The APICS dictionary defines MRP as: “*A set of techniques that uses a bill* of material data, inventory data, and the master production schedule to calculate *requirements for materials*” [4]. MRP requires three basic inputs. First the master roduction schedule, second a bill of material (BOM) for each sku (part number) what other sku are required as direct components, and third the actual level of inventory for every sku. See Figure 2. According to [13] an MRP system serves a central role in material planning

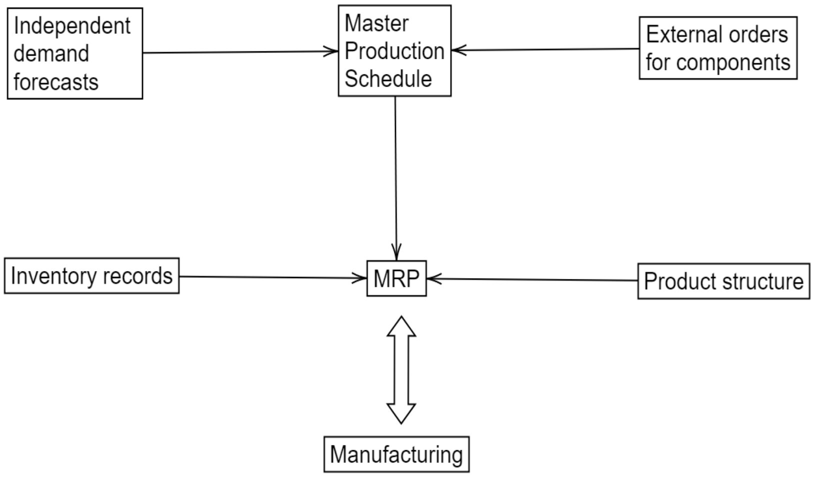
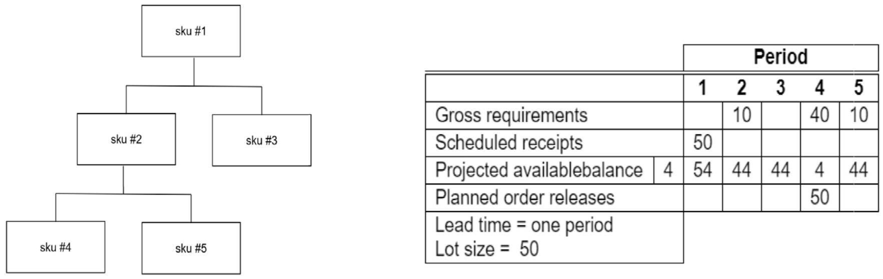

Independent demand forecasts
Master Production Schedule
External orders for components
Inventory records
MRP
Product structure
Manufacturing

Figure 1. The conventional planning schema. Based on [13]



		Period				
		1	2	3	4	5
Gross requirements			10		40	10
Scheduled receipts		50				
Projected availablebalance	4	54	44	44	4	44
Planned order releases					50	
Lead time = one period Lot size = 50						

(a) BOM for a Simple Example. (b) The basic MRP record form: [13].

Figure 2. Example of a bill of manufacture and basic records for feeding the MRP system

and control. It translates the overall plans for production into the detailed individual steps necessary to accomplish those plans. It provides information for developing capacity plans, and it links to the systems that actually get the production accomplished. [20]

Lot for lot MRP Optimization Model The formulation for MRP is based in the model proposed by [29] and follow the form:

$$\underset{x}{\text{minimize}} \sum_{i=1}^{P} \sum_{t=1}^{T} (T-t) X_{i,t}$$

$$\begin{aligned}
&\text{subject to} \\
&\quad \sum_{\tau=1}^{t-LT(i)} X_{i,\tau} + I(i,0) \geq \sum_{\tau=1}^{t} \left(D(i,\tau) + \sum_{j=1}^{P} R(i,j) X_{i,j} \right) \\
&\quad X_{i,t} - \delta_{i,t} LS(i) \geq 0 \\
&\quad \delta_{i,t} - \frac{X_{i,t}}{M} \geq 0 \\
&\quad \delta_{i,t} \in 0, 1 \\
&\quad \forall i \in 1, ..., P, \forall t \in 1, ..., T.
\end{aligned}$$

P Number of SKUs

T Number of time buckets (i.e., planing horizon)

LT_i Lead time for each SKU *i*

$R_{i,j}$ Number *i* necesarios para hacer un *j*

$D_{i,t}$ External demand for *i* over *t* period

$I_{i,0}$ Initial inventory for every SKU *i*

LS_i Minimum lot size for every SKU *i*

M A large number

The planning horizon *i* calculated by [25] and the overall costs are as well found to significantly increase with forecast error [1]

2.2. Forecasting Traditional Methods

The core of MPS is the forecasting job to predict future demands. This work uses the Box-Jenkins methodology to applying the seasonal ARIMA model in order to obtain the future demand of principal sku.

Time Series Analysis According to [13] the time series methods are called common methods because they no longer require other information of past values. Time series is a term to refer to the collection of observations of economic or physical phenomena drawn at discrete points in time. The idea is that past information can be used to forecast future values of the series. In time series analysis we try to isolate the patterns that arise most frequently, these include the following:

– **Trend:** Refers to the trend of a series of time that exhibits a stable pattern of growth or decrease.

– **Seasonality:** A seasonality pattern are those that are repeated at fixed intervals.

– **Cycles:** The variation of cycles is similar to seasonality, except that the duration and the magnitude of the cycle varies. One associates cycles with economic variations that are also present in seasonal fluctuations.

– Randomness: A random series is where you do not have a recognized pattern of data. One can generate a random series of data that have a specific structure . The data that seems to have apparently a randomness , actually have a specific structure. Actually the random data fluctuate around a fixed average.

Box-Jenkins Methodology This method was given thanks to two known statisticians [3]. The models proposed are based on exploiting the self-correlations structure of a time series. Box jenkins models are also known by the name of ARIMA models, which is an acronym for integrated auto-regressive mobile average. The self-correlation function plays a central role in the development of these models, this is the characteristic that distinguishes the ARIMA model from the other methods mentioned above. As all these forecasts are handled through models, we denote the time series, denoting the time series of interest as $D_1,D_2, ...D_n$. We are going to assume initially that the series is stationary. In this way $E = D_i = \mu$ and $var = D_i = \sigma^2 \forall i \in 1, 2, ...n$. Practically speaking, in the seasonality there is no growth or decrease in the series, and the variation remains relatively constant. This is important to denote that seasonality does not imply independence. Therefore it is possible to evaluate D_i and D_j they are dependent random variables when $i \neq j$ although their marginal density functions are the same.

The assumption of seasonality implies that the marginal distribution of two observations separated by the same time interval are the same. This means that D_t and D_{t+1} they have the same distribution as D_{t+m} y D_{t+m+1} for any $m \geq 1$. This implies that the covariance of D_t and D_{t+1} is exactly the same covariance of D_{t+m} and D_{t+m+1}. Therefore , the covariance of these two observations depends only on the number of periods that separate them. In this context, covariance is also known as self-covariance , we are comparing two values of the same series separated by a fixed delay. Leave $Cov(D_{t+m},D_{t+m+k})$ be the covariance of D_{t+m} and D_{t+m+k} given by $Cov(D_{t+m},D_{t+m+k}) = E(D_{t+m}D_{t+m+k}) - E(D_{t+m})E(D_{t+m+k}) \forall k, int \geq 1$.

There are four main steps needed to build BOX-JENKINS forecast models.

1. Data Transformations: The BOX-JENKINS method is based on the start of a series of stationary time. To be sure that the time series is stationary, several steps are needed preliminarily. We know that differentiation eliminates trend and seasonality. However, if the average of the series is relatively fixed, this may be the case in which the variance is not constant, so a transformation of the data is going to be required.

2. Identification of the Model: This step refers exactly to which is the most appropriate ARIMA model. The identification of the type of model is both art and science, it is difficult if not impossible, to identify the model only the series is examined. It is much more effective to study the self-correlations of samples and partial self-correlations for the identification of patterns that coincide with those of the known processes. In some cases, the self-correlation structure will point to a simple AR or MA process, but it is more common to mix these two terms to have a better fit.

3. Parameter Estimation: Once the appropriate model has been identified, the optimal values of the parameters of the model (*i.e.,* $a_0, a_1, ..., a_p$ *y* $b_0, b_1, ..., b_q$) must be determined, usually this step is carried out by means of least squares adjustment methods or by the maximum likelihood method, this step is performed by a computer program.

4. Forecasts: Once the model has been identified and the values of the optimal parameters determined, the model provides forecasts of future values of the series. The BOX-JENKINS models are more effective in providing one-step forecasts, but can also provide multi- step forecasts.

5. Evaluation: The waste pattern (forecast errors) can give useful information to see the quality of the model. The residuals must form a white noise (ie, random) process with zero means. When there are patterns in the waste, it is suggested that the model can improve.

2.3. Long short-term Memory Networks

The bright idea of introducing self-loops to produce paths where the gradient can flow for long durations is a core contribution of the first long short-term memory (LSTM) model [11]. In this case, we mean that even for an LSTM with fixed parameters, the time scale of integration can change based on the input sequence, because of the time constants.

The LSTM has been found hugely successful in many applications, such as unconstrained handwriting recognition [8], speech

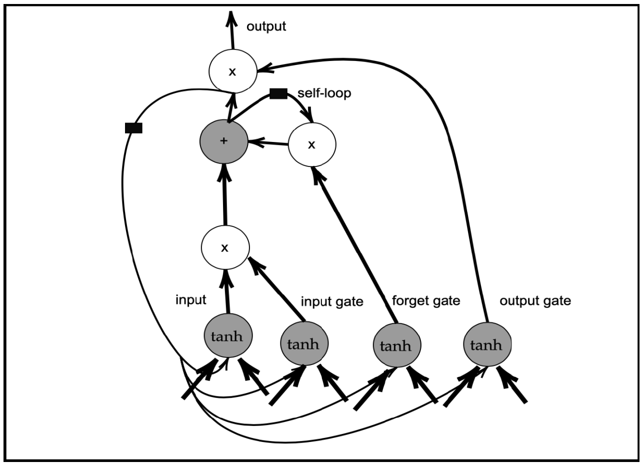


Figure 3. Block diagram of the LSTM recurrent network “cell”

recognition [7,9] handwriting generation [6], machine translation (Sutskever et al., 2014), image captioning [14, 28, 30], and parsing [27] .

The LSTM block diagram is illustrated in Figure 3. The corresponding forward propagation equations are given below, for a shallow recurrent network architecture.

Cells are connected recurrently to each other, replacing the usual hidden units of ordinary recurrent networks. An input feature is computed with a regular artificial neuron unit. Its value can be accumulated into the state if the sigmoidal input gate allows it. The state unit has a linear self-loop whose weight is controlled by the forget gate. The output of the cell can be shut off by the output gate. All the gating units have a sigmoid nonlinearity, while the input unit can have any squashing nonlinearity. The state unit can also be used as an extra input to the gating units. The black square indicates a delay of a single time step.

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right), \tag{1}$$

$$s_i^{(t)} = f_i^{(t)}\ s_i^{(t-1)} + g_i^{(t-1)}\sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right), \tag{2}$$

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right). \tag{3}$$

$$h_i^{(t)} = tanh\left(s_i^t\right) q_i^{(t)}, \tag{4}$$

$$q_i^{(t)} = \sigma\left(b_i^0 + \sum_j U_{i,j}^0 x_j^{(t)} + \sum_j W_{i,j}^0 h_j^{(t-1)}\right), \tag{5}$$

2.4. The Prophet Forecasting Model

We use Prophet open-source software in Python [26] based in a decomposable time series model [10] components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + t \qquad (6)$$

Were $g(t)$ is the trend function which models non periodic changes in the value of the time series, $s(t)$ represents periodic changes, and $h(t)$ represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term t represents any idiosyncratic changes that are not accommodated by the model; later we will make the parametric assumption that t is normally distributed.

3. Solution Method

Open source tools in optimization are increasing popularity over the last years and can be observed in works like [24] for the optimization process in supply chain management and [19] in machine learning applications. We describe our approach for using open source tools in MRP with dynamic demand as follows.

– **Forecasting Model Selection:** In order to use the method that generates less error t. First, we apply the SARIMA model through the ASTSA library[1], from the *R* programming language. In addition, Fb-prophet[2] and Lstm- Keras[3] are also used with the Python programming language. Next, we select the method with the least error of them.

– **JuMP-Julia for Mathematical Programming[4]:** JuMP it is a package for mathematical optimization and operations research.

– **Cbc Solver[5]:** Cbc it is a free solver, that let us solve linear and integer optimization problems.

-

– **IDE:** IPython/IR/IJulia/Jupyter notebooks and Google-Colab.

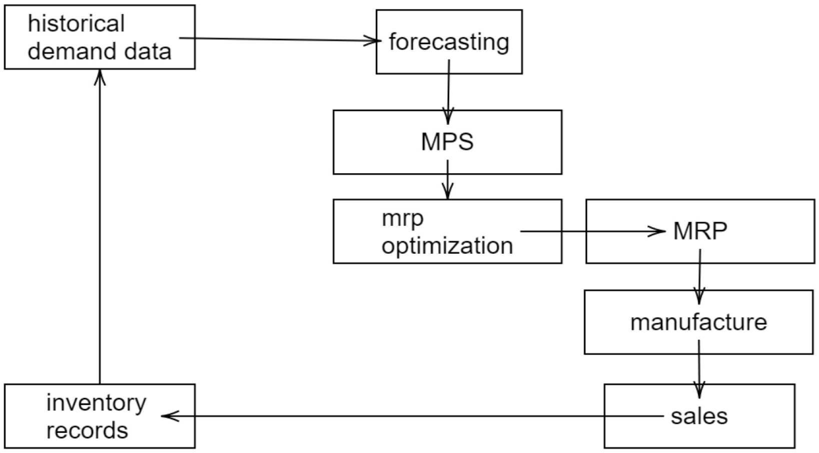


Figure 4. Methodological approach

In this paper, we propose a methodology based on a combination of Forecasting and MIP techniques to help the managers of the fragrance industry to decide the best inventory and supply policies. The steps followed in the methodology are shown in the Figure 4.

[1] https://cran.r-project.org/web/packages/astsa/index.html

[2] https://facebook.github.io/prophet/docs/quick_start.html

[3] https://keras.io/

[4] http://www.juliaopt.org/JuMP.jl/v0.13/

[5] https://github.com/coin-or/Cbc

The methodology is applied to a particular manufacturing case study: to help the managers of the fragrance industry to achieve their goals related to minimizing inventory and minimizing significant deviations in their forecasts. First, the forecasting job is performed by using the box-Jenkins methodology although ASTSA package in R, the Fb-Prophet package and Lstmn with KERAS in Python, next, the Master Production Schedule, is created based on the previous forecasts, and the MRP optimization is performed to get the final MRP plan. The results of the MRP model are evaluated comparing their results with the real demand and improved by adjusting the historical data demand records.

4. Results

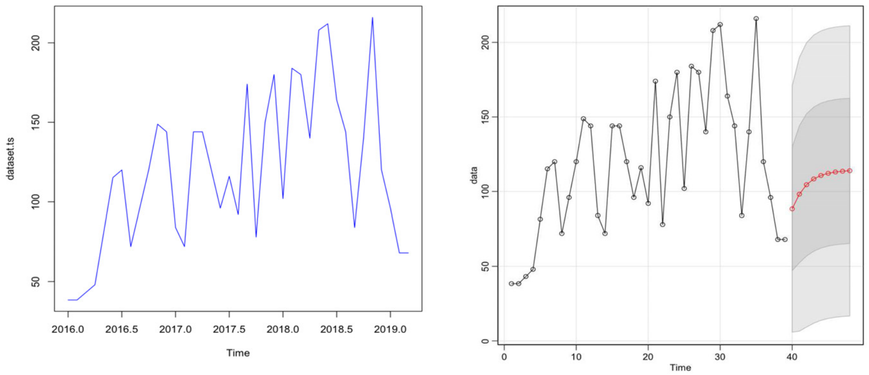
We apply the solution method to a company dedicated to the creation development, manufacture, and marketing of flavors and fragrances for the Colombian industry. A step by step application is described as follows:

4.1. Forecasting the Principal SKU

First, the historical demand data is plotted in order to isolate the data patterns. Later we apply the SARIMA model by using the R language Fig. 5, also the Fb-Prophet package and Lstmn using KERAS and Google Colab Figure 6.

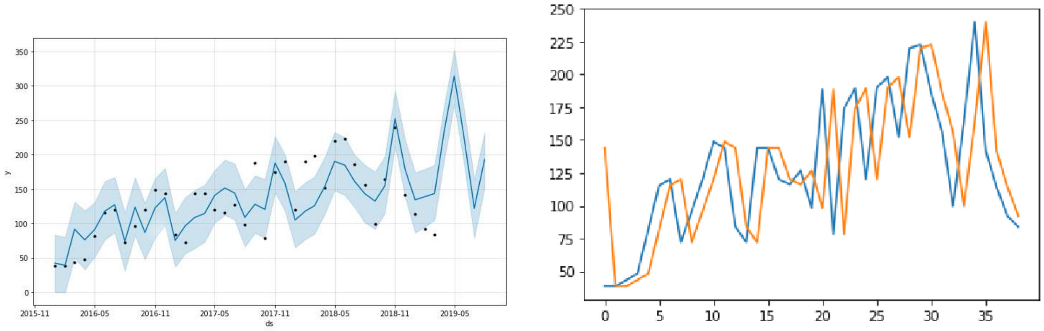
Table 1 shows the difference in forecasts generated and their associated errors in the last row. The techniques with the lowest associated error are SARIMA and LSTMN, so the predicted demand generated by them will be used, which can be seen in Table 1.

As it can be seen in Table 2 the future demand for 9 months is the principal input for the MRP formulation, and it will be represented on the subset D_j .



(a) Historical demand data plot (b) Forecasting for 2019 (monthly).

Figure 5. Forecast using SARIMA in *R* language



(a) Fb-Forecast data plot (b) Lstm-Keras

Figure 6. Forecast using SARIMA in *R* language

Technique	SARIMA	FB-PROPHET	LSTMN
m1	122,1863	220,064847	222,8
m2	144,89	293,060167	185,6
m3	114,5067	197,096387	156
m4	108,1722	71,020883	99,6
m5	108,3905	182,760315	164
m6	112,6212	20,875617	240
m7	119,4588	186,352169	141,6
m8	121,4177	419,670517	114
m9	123,3766	132,994213	92
RMSE	**39,95761**	**127,764023**	**53,53296**

Table 1. Comparison of techniques and RMSE

Apr	May	Jun	Jul	Aug
88.34232	98.20327	104.55308	108.39207	110.74301
Sept	**Oct**	**Nov**	**Dec**	
112.17886	113.05630	113.59244 1	13.92004	

Table 2. Forecasted Demand D_j

4.2. Using the Optimization Model for MRP

In this item, we use the Julia programing language and the package JuMP for mathematical optimization, the code for the problem it can be seen as follows.

```
using JuMP,Cbc,NamedArrays,DataFrames
filas=size(D,1)
col=size(D,2)
mrp=Model(solver=CbcSolver())
@variables mrp begin
x[1:filas,1:col]>=0
d[1:filas,1:col]>=0
end
T=col
@objective(mrp,Min,sum( x[i,j]*((T-j)) for i=1:filas, j=1:col))
for i=1:filas,t=1:col
```

```
@constraint(mrp, sum(x[i,s] for s=1:t-LT[i,1])+I[i,1]>=sum(D[i,s]
+sum(R[i,j]*x[j,s] for j=1:filas) for s=1:t))
end
@constraint(mrp,x-d.*LS.>=0)
@constraint(mrp,d-x/1000000000000000.>=0)
status=solve(mrp)
println(getobjectivevalue(mrp))
println(DataFrame(getvalue(x)))
println(DataFrame(getvalue(d)))
```

We use the $R_{i,j}$ bill of manufacture that shows in the Table 3 as follows:

	j1	**j2**	**j3**	**j4**	**j5 . . .**	**j34**
i1	0	0	0	0	0 . . .	0
i2	253.816	0	0	0	0 . . .	0
i3	0.4	0	0	0	0 . . .	0
i4	20	0	0	0	0 . . .	0
i5	20	0	0	0	0 . . .	0
i6	105.784	0	0	0	0 . . .	0
i7	0.1	0	0	0	0. . .	0
i8	0	0	0	0	6.44 . . .	0
i9	0	0	0	0	0.04 . . .	0
i10	0	0	0	0	0.24 . . .	0
i11	0	0	0	0	0.2 . . .	0
i12	0	0	0	0	2.312 . . .	0
i13	0	0	0	0	1.4 . . .	0
i14	0	0	0	0	1.16 . . .	0
i15	0	0	0	0	1.04 . . .	0
i16	0	0	0	0	0.88 . . .	0
i17	0	0	0	0	0.85 . . .	0
i18	0	0	0	0	0.7 . . .	0
i19	0	0	0	0	0.48 . . .	0
i20	0	0	0	0	0.456 . . .	0
i21	0	0	0	0	0.4 . . .	0
i22	0	0	0	0	0.376 . . .	0
i23	0	0	0	0	0.296 . . .	0
i24	0	0	0	0	0.2 . . .	0
i25	0	0	0	0	0.124 . . .	0

i26	0	0	0	0	0.08 . . .	0
i27	0	0	0	0	0.098 . . .	0
i28	0	0	0	0	0.044 . . .	0
i29	0	0	0	0	0.04 . . .	0
i30	0	0	0	0	0.04 . . .	0
i31	0	0	0	0	0.04 . . .	0
i32	0	0	0	0	0.04 . . .	0
i33	0	0	0	0	0.024 . . .	0
i34	0	0	0	0	2 . . .	0

Table 3. Bill of manufacture $R_{i,j}$

4.3. Final Results

	t1	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**t8**
sku1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
sku2	0.0	0.0	0.0	0.0	22422.7	24925.6	26537.2	27511.6
sku3	0.0	0.0	0.0	0.0	35.3	39.3	41.8	43.4
sku4	0.0	0.0	0.0	0.0	1766.8	1964.1	2091.1	2167.8
sku5	0.0	0.0	0.0	0.0	1766.8	1964.1	2091.1	2167.8
sku6	0.0	0.0	0.0	0.0	9345.2	10388.3	11060.0	11466.1
sku7	0.0	0.0	0.0	0.0	8.8	9.8	10.5	10.8
sku8	11378.5	12648.6	13466.4	13960.9	14263.7	14448.6	14561.7	14630.7
sku9	70.7	78.6	83.6	86.7	88.6	89.7	90.4	90.9
sku10	424.0	471.4	501.9	520.3	531.6	538.5	542.7	545.2
sku11	353.4	392.8	418.2	433.6	443.0	448.7	452.2	454.4
sku12	4084.9	4540.9	4834.5	5012.0	5120.8	5187.2	5227.7	5252.5
sku13	2473.6	2749.7	2927.5	3035.0	3100.8	3141.0	3165.6	3180.6
sku14	2049.5	2278.3	2425.6	2514.7	2569.2	2602.5	2622.9	2635.3
sku15	1837.5	2042.6	2174.7	2254.6	2303.5	2333.3	2351.6	2362.7
sku16	1554.8	1728.4	1840.1	1907.7	1949.1	1974.3	1989.8	1999.2
sku17	1501.8	1669.5	1777.4	1842.7	1882.6	1907.0	1922.0	1931.1
sku18	1236.8	1374.8	1463.7	1517.5	1550.4	1570.5	1582.8	1590.3
sku19	848.1	942.8	1003.7	1040.6	1063.1	1076.9	1085.3	1090.5
sku20	805.7	895.6	953.5	988.5	1010.0	1023.1	1031.1	1036.0
sku21	706.7	785.6	836.4	867.1	885.9	897.4	904.5	908.7
sku22	664.3	738.5	786.2	815.1	832.8	843.6	850.2	854.2
sku23	523.0	581.4	619.0	641.7	655.6	664.1	669.3	672.5

sku24	353.4	392.8	418.2	433.6	443.0	448.7	452.2	454.4
sku25	219.1	243.5	259.3	268.8	274.6	278.2	280.4	281.7
sku26	141.3	157.1	167.3	173.4	177.2	179.5	180.9	181.7
sku27	173.2	192.5	204.9	212.4	217.1	219.9	221.6	222.6
sku28	77.7	86.4	92.0	95.4	97.5	98.7	99.5	100.0
sku29	70.7	78.6	83.6	86.7	88.6	89.7	90.4	90.9
sku30	70.7	78.6	83.6	86.7	88.6	89.7	90.4	90.9
sku31	70.7	78.6	83.6	86.7	88.6	89.7	90.4	90.9
sku32	70.7	78.6	83.6	86.7	88.6	89.7	90.4	90.9
sku33	42.4	47.1	50.2	52.0	53.2	53.8	54.3	54.5
sku34	3533.7	3928.1	4182.1	4335.7	4429.7	4487.2	4522.3	4543.7

Table 4. Final results for every sku *i* and their needs over time in kgs

5. Discussion and Future Investigation

Future investigation must be related to DDMRP demand driven MRP and machine learning techniques related to forecasting methods. The first affirmations is related with methods for maintain a minimum and maximum intervals for inventory management over every sku. The second affirmation is related to method that helps us to obtain better and more accurates forecasts for future demand.

6. Conclusions

The use of Open source optimization and forecasting tools allows to solve optimization and predictive problems efficiently.

The forecasting problem is the principal problem in an MRP based company and can be solved at scale using open source tools.

The biggest contribution of this work is to show how to solve a real problem over forecasting and optimization using three programming languages in order to reduce error and improve the time scheduling for every stock keeping unit.

The company uses this implementation in order to plan his manufacturing process at a larger scale including every product of their portfolio.

References

[1] Altendorfer, K., Felberbauer, T., Jodlbauer, H. (2016). Effects of forecast errors on optimal utilisation in aggregate production planning with stochastic customer demand. *International Journal of Production Research,* 54(12), 3718–3735 (2016)

[2] Bousqaoui, H., Achchab, S., Tikito, K. (2017). Machine learning applications in supply chains: long short-term memory for demand forecasting. In: International Conference of Cloud Computing Technologies and Applications. pp. 301–317. Springer (2017)

[3] Box, G. E., Jenkins, G.M., Reinsel, G. C., Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

[4] Cox, J. F., Blackstone, J. H. (2002). APICS dictionary. Amer Production & Inventory (2002)

[5] Disney, S. M., Farasyn, I., Lambrecht, M., Towill, D. R., Van de Velde, W. (2006). Taming the bullwhip effect whilst watching customer service in a single supply chain echelon. *European Journal of Operational Research* 173(1), 151–172 (2006)

[6] Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)

[7] Graves, A., Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. *In: International conference on machine learning*. p 1764–1772 (2014)

[8] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence,* 31(5), 855–868 (2008)

[9] Graves, A., Mohamed, A.r., Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, *speech and signal processing*. pp. 6645–6649. IEEE (2013)

[10] Harvey, A.C., Peters, S. (1990). Estimation procedures for structural time series models. *Journal of Forecasting,* 9(2), 89–108.

[11] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural computation* 9(8), 1735–1780 (1997)

[12] Hopp, W. J., Spearman, M. L. (2011). Factory physics. Waveland Press (2011)

[13] Jacobs, F. R. (2011). Manufacturing planning and control for supply chain management. McGraw-Hill (2011)

[14] Kiros, R., Salakhutdinov, R., Zemel, R.S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)

[15] Lee, T., Adam Jr, E.E. (1986). *Forecasting error evaluation in material requirements planning (mrp) production-inventory systems. Management Science,* 32(9), 1186–1205 (1986)

[16] Lee, T., Shih, W. (1989). Optimal forecast biasing in theoretical inventory models. *The International Journal Of Production Research,* 27(5), 809–830 (1989)

[17] Li, Q., Disney, S. M. (2017). Revisiting rescheduling: Mrp nervousness and the bullwhip effect. *International Journal of Production Research*, 55(7), 1992–2012 (2017)

[18] Mabert, V.A. (2007). The early road to material requirements planning. Journal of Operations Management 25(2), 346–356.

[19] Moreno, R. H., Garcia, O., et al. (2018). Model of neural networks for fertilizer recommendation and amendments in pasture crops. In: 2018 ICAI Workshops (ICAIW). p 1–5. IEEE (2018)

[20] Nahmias, S., Cheng, Y. (2005). Production and operations analysis, vol. 6. McGraw-hill New York (2005)

[21] Orlicki, J.A. (1975). Material requirements planning: the new way of life in production and inventory management. McGraw-Hill (1975)

[22] Priore, P., Ponte, B., Rosillo, R., de la Fuente, D. (2018). Applying machine learning to the dynamic selection of replenishment policies in fast-changing supply chain environments. *International Journal of Production Research,* p 1–15 (2018)

[23] Ptak, C. A., Smith, C. (2016). Demand Driven Material Requirements Planning (DDMRP). Industrial Press, Incorporated (2016)

[24] Romero-Gelvez, J. I., Gonzales-Cogua, W. C., Herrera-Cuartas, J. A. (). Cvrptw model for cargo collection with heterogeneous capacity-fleet. In: International Conference on Applied Informatics. Springer (2019)

[25] Sridharan, V., BERRY, W.L. Master production scheduling make-to-stock products: a framework for analysis. *The International Journal of Production Research*, 28(3), 541–558.

[26] Taylor, S. J., Letham, B. Forecasting at scale. The American Statistician 72(1), 37–45.

[27] Vinyals, O., Kaiser, A., Koo, T., Petrov, S., Sutskever, I., Hinton, G. Grammar as a foreign language. In: *Advances in neural information processing systems*. p 2773–2781.

[28] Vinyals, O., Toshev, A., Bengio, S., Erhan, D. Show and tell: A neural image caption generator. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 3156–3164.

[29] Voß, S., Woodruff, D. L. Introduction to computational optimization models for production planning in a supply chain, vol. 240. Springer Science & Business Media.

[30] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044.