The anticipative concept in warehouse optimization using simulation in an uncertain environment

Davorin Kofjač a,*, Miroljub Kljajić a, Valter Rejec b

a University of Maribor, Faculty of Organizational Sciences, Cybernetics and DSS Laboratory, Kidričeva cesta 55a, SI-4000 Kranj, Slovenia
b Iskra Actoelektrika d.d., Polje 15, SI-5290 Sempeter pri Gorici, Slovenia

Received 25 September 2006; accepted 18 June 2007
Available online 22 October 2007

Abstract

The modern business environment is highly unpredictable. An anticipation approach in a real case study is presented to cope with such instability and minimize the total inventory cost without stock-outs occurring and inventory capacity being exceeded. The anticipation concept is performed using simulation models supported by inventory control algorithms on a selected sample of representative items. The inventory control algorithms include Silver–Meal, Part period balancing, Least-unit cost, and Fuzzy inventory control algorithm based on fuzzy stock-outs, highest inventory level and total cost. Transportation cost is explicitly defined as a discrete function of shipment size. The algorithms are tested on historic data. Simulation results are presented and the risk of accepting them as reliable is discussed. The process of simulation model implementation is briefly discussed to further validate the model and train order managers to use the simulation model in their order placement process.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Inventory; Optimization; Anticipatory systems; Fuzzy sets; Decision support systems

1. Introduction

This paper presents a real case study of inventory optimization in an automotive company. The most important issue with optimization is the cost reduction of the inventory control processes to a minimum without stock-outs occurring and inventory capacity being exceeded. Several different principles of inventory optimization are described by Silver et al. [16] and Tompkins and Smith [17]. One method of inventory optimization is to find the right order placement strategy. Therefore, the order manager deals with the problem of deciding when and how many items to order. This decision problem is vast, especially when considering that the right replenishment strategy for more than 10,000 items stored in a warehouse must be found. Here the assistance of a decision support system (DSS) is crucial as the order managers mainly use only their experience and intuition.

The DSS presented in this paper is based on an anticipatory simulation model, inventory optimization algorithms, e.g. Least-unit cost, and fuzzy sets. The simulation model is capable of performing several scenarios based on inventory control algorithms in a short period of time, thus providing the order manager with alternative replenishment data on a specific item. It is the order manager’s choice whether to accept this data as valuable information to help him with the decision problem he is facing. In comparison with other methods, a dynamic analysis of the considered system behaviour is the main advantage of testing the strategy with the aid of simulation scenarios [7,9].

Anticipation of future inventory dynamics is crucial to cost minimization and is performed by simulating different strategies that yield an insight into a probable future based on stochastic demand (production plan) and stochastic lead times.
There have been several studies that applied fuzzy sets in inventory control [3,8,11–13,15,18]. These papers presented studies of various fuzzy inference systems, where fuzzy demand, inventory level, order quantity, lead times, and various fuzzy costs, e.g. shortage cost, were studied and analysed. Rotshtein and Rakityanskaya [15] provided an interesting way of improving the fuzzy inference system by fine tuning membership function parameters using genetic algorithms and neural networks. One shortcoming of this study is the exclusion of cost function and lead times, which are very important for our case. Kofjač et al. [8] presented the study of inventory control using fuzzy number of stock-outs, highest inventory level, and stochastic lead times and demand.

The potential risk of accepting the simulation model as reliable is too high to start with an ad hoc implementation. Moreover, the implementation of new algorithms is usually a very delicate task as the workforce is used to its own routine and not open to new methodologies. Barlas and Öze-...

...are several possibilities for improving the inventory control process. Order managers solve these complex problems mostly by utilising their experience.

As the production process and delivery are stochastic, future inventory dynamics cannot be determined with certainty and have to be anticipated. “An anticipatory system is a system containing a predictive model of itself and/or of its environment which allows it to change state at an instant in accord with the model’s predictions pertaining to a latter instant” [14]. In other words, anticipatory models will act according to the past states of the system and according to the desirable and possible future states as defined by function η:

\[ x_{t+1} = η(x_1, x_2, \ldots, x_t, x'_s), \quad (1) \]

where \( x'_s, \ t + 1 \leq s \leq t + k(t) \), is a prediction \( x_s \) done at time \( t \), and \( k(t) \) is the size of the forecasting interval at time \( t \). If \( x \) is the inventory variable, then Eq. (1) describes its past, present and future dynamics. Anticipation is done by running simulation scenarios where inventory control algorithms predict future inventory dynamics based on production plans.

2.1. Mathematical formulation

From the control aspect, the inventory optimization problem can be described with the following equation:

\[ x(k + 1) = x(k) + d(k) - p(k), \quad k = 0, 1, 2, \ldots, \]
\[ x(0) = x_0, \quad (2) \]

where \( x(k) \) represents inventory level variable, \( d(k) \) stochastic item delivery, \( p(k) \) stochastic production process, and \( x_0 \) the initial inventory at \( k = 0 \). The demand of production is stochastic and discontinuous; its quantity keeps changing, and time varying gaps occur between two successive demands. Due to stochastic demand and lead times, the expected value \( EV(x(k + 1)) \) is estimated by the Monte Carlo simulation using random generators as described in the following text.

The delivery function \( d(k) \) is delayed for a lead time \( τ_d \) of an order \( o(k) \), where \( o(k) \) has been divided into a number of batches \( b \):

\[ d_j(k) = \frac{o(k - φ(τ_d))}{λ(b)}, \quad j = 1, 2, \ldots, λ(b), \quad (3) \]

where \( φ(τ_d) \) represents a discrete uniform probability density function of lead time at interval \([τ_{d, min}, τ_{d, max}]\) and \( λ(b) \) discrete uniform probability density function of a number of batches at interval \([b_{min}, b_{max}]\). Lead times \( τ_d \) tend to be very long for some items (e.g. 2–3 months), thus representing a problem because they are usually much longer than the production plan horizon \( τ_p \) in which the production plan can be predicted with certainty.

There are two types of items stored in the inventory regarding delivery:
(a) The order quantity \( o(k) \) can be of any value. These items are delivered in several batches.

(b) The order quantity \( o(k) \) must be rounded up to a multiplier of the quantity contained in one package:

\[
o(k) = \left\lceil \frac{o(k)}{q_p} \right\rceil q_p,
\]

where \( o(k) \) is a suggested order quantity and \( q_p \) the predefined quantity in one package. These items are delivered in one batch.

For example, if a suggested order quantity \( o(k) \) is 155 pieces, then the new order quantity for items under (a) would be the same as suggested (155 pieces), while for items under (b) it would be 180 pieces (6 \( \times \) 30 = 180), if we assume that \( q_p \) is 30 pieces.

To compensate for the stochastic delivery delay, the order placement policy \( o(k) \) has to be defined as

\[
o(k) = \gamma(x(k), d(k - \tau_u), p(k + \tau_p), x_{\text{min}}),
\]

where \( \tau_p \) represents a production plan horizon and \( x_{\text{min}} \) safety stock.

The total cost function \( c \) is defined by

\[
c = \sum_{k=0}^{n} (c_c x(k) + c_p x(k) + c_1 (w) d(k) + c_o o(k) + c_v d(k)),
\]

where \( n \) represents the number of simulation steps, \( w \) transport weight, \( c_c \) cost of capital, \( c_p \) cost of physical storage, \( c_1 \) transportation cost, \( c_o \) fixed ordering costs, and \( c_v \) cost of taking over products.

It is necessary to find a replenishment policy \( o(k) \) that minimizes the total cost function \( c \), where the following restrictions have to be considered: (a) maximum inventory level \( x_{\text{max}} \) for a specific product must not be exceeded, and (b) no stock-outs may occur \( x_{\text{min}} \). Formally:

\[
\min_c \quad x_{\text{min}} \leq x(k) \leq x_{\text{max}}
\]

In most papers, it has been implicitly assumed that transportation cost is a part of the ordering cost and is therefore assumed to be independent of the size of shipment. As such, the effect of transportation cost is not adequately reflected in final planning decisions. There is a need for models explicitly involving transportation cost for better decision-making [4]. Rather than assuming it to be a part of the fixed ordering cost or to be insignificant, we will take transportation cost as a function of shipment lot size. In this case transportation cost \( c_1 \) can be described by the following function:

\[
c_1(w) = \begin{cases} 0, & w = 0, \\ C_{l1}, & w \in [0, W_1], \\ C_{l2}, & w \in [W_1, W_2], \\ \vdots & \vdots, \\ C_{ln}, & w \in [W_{n-1}, W_n], \end{cases}
\]

where \( C_{l1}, C_{l2}, \ldots, C_{ln} \) are transportation cost constants, dependent on transport weight \( w \), and \( W_1, W_2, \ldots, W_n \) are weight constants (see Fig. 1).

The results of the simulation are \( m \times n \) matrices of inventory \( X \), orders \( O \), and delivery \( D \), where \( m \) represents the number of simulation steps and \( n \) the number of inventory control algorithms. Production (demand) vector \( P \), of a dimension \( m \), is imported into the simulation model at the beginning of the simulation. Vectors \( S \) (highest inventory level) and \( \bar{z} \) (number of stock-outs), of dimension \( n \), are calculated from inventory matrix \( X \), where each vector component represents the result of an individual inventory control algorithm. The simulation also provides the \( n \)-dimensional total cost vector \( \bar{c} \) containing the total cost of each inventory control algorithm.

### 2.2. The inventory control model

Fig. 2 presents the simplified inventory control model. It is a model with an added decision support simulation system (DSSS) supported by inventory control algorithms. The DSSS gathers information on the production plan, stock-on-hand, and delivery, and runs different simulation scenarios according to the algorithms to anticipate future inventory dynamics. A fuzzy strategy assessor (FSA) is used to assess the results of simulation runs. Inventory control algorithms and the FSA are described in the following sections. Based on the FSA assessment, the DSSS provides ordering process information on how much and when to order.

The detailed inventory control model is represented by the causal loop diagram shown in Fig. 3, from which the influences of the elements can be observed. The arrow...
represents the direction of the influence and the + or − sign its polarity. The + sign is used between two elements if the causal element has the same influence on the consequential element. The − sign is used if the causal element has an opposite influence on the consequential element.

There are two negative feedback loops in the causal loop diagram in Fig. 3. The first interconnects Inventory, Ordering, and Delivery, and represents the fact that less is ordered if the inventory level is high. The second interconnects Delivery and Ordering, and represents the concept that less is ordered if more was ordered before. This loop considers orders which have not been delivered yet and will have an impact on the inventory level later.

2.3. Fuzzy strategy assessment

In general, imprecise and incomplete data information could be introduced in decision-making problems such as order placement. Furthermore, the decisions made by experts rely on their individual competence and are subjective. Usually, experts express their opinions by means of numerical values. When they are unable to give exact numerical values to express their opinions, a more realistic alternative option is to use linguistic assessments instead of numerical values [5]. For example, the value of stock-out or shortage cost is difficult to assess in our case since items held in the inventory are often a part of several finished products, unlike the study of Katagiri and Ishii [6] where fuzzy shortage cost was used to solve inventory problems. The number of stock-outs is the only information available in our case and difficult to assess regarding costs. To state that the number of stock-outs is “high” or “low” is more realistic than using numerical (crisp) values, and fuzzy sets [19] are an appropriate option to deal with such a problem because fuzzy systems allow the processing of information in linguistic terms. Information is expressed in the form of IF–THEN rules and is built on the analogy with human reasoning. Besides the linguistic layer, information processing in fuzzy systems also takes place at the numerical level using the special inference algorithm. The inference algorithm is a procedure with a large degree of flexibility (there is a large family of inference operators and fuzzification and defuzzification methods) that creates unique input–output mapping between the system (base) variables. This unique architecture of fuzzy systems makes them useful for man–
machine interaction problems and permits the use of human experience and knowledge that is usually expressed in vague terms and is otherwise difficult to implement.

The fuzzy strategy assessor (FSA) is based on fuzzy logic and used to assess the simulation results of \( n \) replenishment strategies (Fig. 4). The inputs of FSA are \( n \)-dimensional crisp vectors for number of stock-outs \( z \), total cost \( c \), and highest inventory level \( S \), where the vector component represents the simulation result of an individual replenishment strategy. The output of the FSA is the \( n \)-dimensional crisp strategy estimation vector \( \vec{E} \).

Vector components for \( z \), \( c \), and \( S \) are normalized at interval \([0,1]\) using the following equation:

\[
g'_i = \frac{g_i}{\max(g)}, \quad i = 1, 2, \ldots, n, \tag{9}\]

where \( g \) is either vector \( z \), \( c \) or \( S \), \( g_i \) vector component, \( g'_i \) normalized vector component, \( \max(g) \) is the maximum value of vector components, and \( n \) vector size. Vectors are then classified (fuzzified) using equally spaced Gaussian membership functions (MFs) as shown in Table 1.

The FSA inference system contains the expert’s rule base consisting of 125 \( (=5^3) \) rules. Estimation of the \( i \)th strategy is calculated according to the following rule:

\[
\text{If } z_i \text{ and } c_i \text{ and } S_i \text{ then } E_i, \quad i = 1, 2, \ldots, n, \tag{10}\]

where \( n \) is the number of simulation strategies. A few examples of the rules are given below:

- If \( z_i \) is none and \( c_i \) is very low and \( S_i \) is very low then \( E_i \) is excellent
- If \( z_i \) is many and \( c_i \) is very high and \( S_i \) is low then \( E_i \) is poor

The rules are used to obtain the final output \( \vec{E} \), which is then defuzzified using a Som (smallest value of minimum) function at interval \([0,1]\). The strategy with the lowest grade is selected as the one yielding the best simulation results. If several strategies achieve the same lowest grade, they are assessed once more. Vectors \( \vec{z}', \vec{c}', \vec{S}' \) are formed from vectors \( \vec{z}, \vec{c}, \vec{S} \); they contain only strategies with the lowest grade and are again normalized using Eq. (9) and fuzzified. A new estimation vector \( \vec{E} \) is calculated. This process is repeated until only one strategy has achieved the lowest grade.

### 3. Inventory control algorithms

The DSSS is based on the simulation model described previously and the following algorithms:

- Fixed review period (FRP),
- Least-unit cost (LUC),
- Part period balancing (PPB),
- Silver–Meal algorithm (SM),
- Fuzzy control algorithm (FZA).

The Least-unit cost, Part period balancing, and Silver–Meal algorithms are described in detail by Silver et al. [16]. The following is a description of the FRP and FZA algorithm.

#### 3.1. Fixed review period algorithm

The Fixed review period algorithm (FRP) is similar to the \((R,S)\) system [16] where every \( R \) units of time an order is made to adjust the inventory level to the order-up-to-level \( S \). In contrast to the \((R,S)\) system, \( S \) is not a fixed value in the FRP algorithm. The FRP is based on making a sum of demand over a specific period of time. The quantity of this sum is used in order quantity calculations together with past orders and stock-on-hand. This model is appropriate for products with large warehouse capacity and is presented in Fig. 5.

Let us assume the specific period is fixed, e.g., five weeks, and it does not vary. This means that an order is placed every five weeks \( (R = 5) \). If we assume the lead time is two weeks, and a production plan can be predicted with certainty for three weeks \( (r_s = 3) \), then we have a reliable prediction period of one week for the quantity to be ordered \( (3 - 2 = 1) \). For the rest of the review period (four weeks; \( 5 - 1 = 4 \)), the production plan is unreliable, and a production plan uncertainty factor must be considered.

Table 2 presents the numerical example of the FRP algorithm. The order quantity at Week 1 is the sum of the production plan from Weeks 3 to 7. The order quantity at Week 6 is the sum of the production plan from Weeks 8 to 12 and so on. The order quantity at Week 16 is symbolically represented with the letter \( q \) since the scope of the Table is not wide enough to make an exact sum. Stock-on-hand and on-order stock are omitted from Table 2 in order to explain how the algorithm works and are considered when running simulation scenarios.

#### 3.2. Fuzzy control algorithm

This algorithm is based on all the inventory control algorithms mentioned before. All algorithms are run in a
simulation, thus yielding \( m \times n \) matrix \( O \) which contains simulation order quantities for each algorithm, where \( m \) represents the number of simulation steps and \( n \) the number of algorithms. Following is an example of the matrix \( O \) (see Fig. 6).

Simulation results are assessed using a fuzzy strategy assessor (FSA), thus providing the estimation vector \( \hat{E} \).

The principle of the fuzzy control algorithm (FZY) is shown in Fig. 7. The inputs are the replenishment strategy estimation vector \( \hat{E} \), order matrix \( O \), and time step \( k \), while the output is an order \( o \) at time step \( k \) as defined by function \( \psi \).

Vector component \( E_j \) with the lowest value is selected as the strategy that provided the best results. The \( j \)th column in matrix \( O \) contains the orders of the selected strategy. The first order that is found in that column is the one that is transmitted to the output, thus forming order \( o(k) \).

The numerical example of this fuzzy control algorithm is provided in Table 3. The algorithm with the lowest assessment is indicated in bold; in this case it is FRP 3 week.

### Table 2
Numerical example of the FRP algorithm

<table>
<thead>
<tr>
<th>Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prod. plan (P)</td>
<td>10</td>
<td>6</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>13</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Order quantity (O)</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>q</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Delivery (D)</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>q</td>
</tr>
</tbody>
</table>

Fig. 5. Fixed review period algorithm as a function of inventory level and production plan.

Fig. 6. Example of the ordering matrix \( O \).

Fig. 7. The fuzzy control algorithm.
The simulation was run for a period of seven years in RP and VP, but also with subperiods of 24 weeks in the VP, depending on the algorithm, at each time step (continuous review) or at each review period (periodic review) the simulation was run for 24 weeks since this is the production plan horizon \( \tau_p \), thus providing an order placement schedule until the next review.

A Monte Carlo simulation was used for variation of production plan unreliability; the production plan variability was simulated by perturbations of its quantity every two weeks for \( \pm 5\% \) by the uniform random generator. Variable lead times were simulated by the discrete uniform random generator at interval \([\tau_{d_{\text{min}}}, \tau_{d_{\text{max}}}]\). If stock-outs occurred during the simulation, the missing quantity was transferred into the next period.

Simulation results are shown in Table 5, where the Real column represents simulation results of the RP, while the other columns show the results of the VP. Results for stock-outs \( z \), maximum inventory level \( S \), and total cost \( c \) are represented for each item. The best strategy among VP algorithms for each item is represented in bold, and it was assessed by the FSA described earlier. The research team had supported the FSA assessments. Results in the column Real are omitted from the assessment and are used in the following table providing a summary of simulation results (Table 6).

The results in Table 6 show a comparison of RP and the strategy chosen by the FSA among VP strategies. Significant savings were achieved for almost all items, except for Item 8 marked with superscript a. For this item, optimization algorithms could not achieve any savings; on the contrary, they produced higher costs than the RP. Evidently, the order manager has found a better order placement policy than the ones proposed by the algorithms.

The results presented in Table 6 also indicate that the FZY algorithm was chosen for all items, except one, as the one yielding the best results. LUC, PPB and S–M algorithms were never chosen. As the FRP algorithm was chosen for Item 4, it is clear that it needs a strict order placement policy than the ones proposed by the algorithms.

In-depth cost analysis is presented in Tables 7 and 8. Table 7 shows the share of fixed ordering costs \( c_0 \), costs of taking over products \( c_r \), transportation cost \( c_t \), cost of physical storage \( c_p \), and cost of capital \( c_c \) with regard to total cost \( c \) for each item. One can notice that transportation cost, cost of physical storage, and cost of capital have a predominant share for most items. Items 2, 6 and 9 are exceptions and also have a dominant ordering cost.

Table 8 presents each cost category’s increase or decrease, signed by “+” or “−” respectively, of the selected VP strategy (as indicated in Table 6) regarding RP. Savings for Items 1 and 7 are mainly due to the major decrease in the physical storage cost and capital cost, whereas transportation cost reduction was crucial for Items 2, 3, 5 and 6. The simulation for Item 4 has mainly reduced

(j = 2) and the order quantity \( o(k) \) will be 137 pieces, if we consider the example matrix \( O \) provided earlier.

4. Results

The simulation model has been built using Matlab. The experiment was performed with the historic data provided by the observed company. Altogether, nine items were examined in this study and their details are described in Table 4. Each item has a variable lead time and the batch size for items 2, 3, 5, and 8 is rounded up to the quantity of the packaging unit.

Items rounded to a packaging unit are delivered in one batch, therefore \( b_{\text{min}} = b_{\text{max}} = 1 \), while other items are delivered in no more than three batches, therefore \( b_{\text{min}} = 1 \) and \( b_{\text{max}} = 3 \). Safety stock \( x_{\text{min}} \) was equal to the average weekly demand of a specific item.

The simulation of the actual warehouse process (RP – real process) was performed using real data of delivery, orders, and demand, while the simulation with inventory control algorithms (VP – virtual process) was performed using only real demand data; the ordering and delivery process in VP were controlled by inventory control algorithms.

The RP simulation was run only once, whereas 10 simulation runs were executed for every VP algorithm. Based on these simulation runs, average costs and average stock-outs were calculated. With several simulation runs and a calculation of average values we have tried to minimize the influences of the random generator, which represent the stochastic environment. Out of all simulation runs the maximum inventory level was considered as the capacity limit; therefore, the strategy with minimal highest inventory level is favoured.

Table 4

| Item class, lead times, and rounding up to packaging unit |
|-----------------|---|---|---|---|---|---|---|---|---|
| Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Class | AX | AY | AY | AZ | AZ | BX | BY | BY | BZ |
| \( \tau_{d_{\text{min}}} \) | 5 | 5 | 6 | 5 | 14 | 6 | 5 | 6 | 6 |
| \( \tau_{d_{\text{min}}} \) | 6 | 6 | 7 | 6 | 16 | 7 | 6 | 8 | 7 |
| Round | No | Yes | Yes | No | Yes | No | No | Yes | No |

The simulation was run for a period of seven years in RP and VP, but also with subperiods of 24 weeks in the VP, depending on the algorithm, at each time step (continuous review) or at each review period (periodic review) the simulation was run for 24 weeks since this is the production plan horizon \( \tau_p \), thus providing an order placement schedule until the next review.

A Monte Carlo simulation was used for variation of production plan unreliability; the production plan variability was simulated by perturbations of its quantity every two weeks for \( \pm 5\% \) by the uniform random generator. Variable lead times were simulated by the discrete uniform random generator at interval \([\tau_{d_{\text{min}}}, \tau_{d_{\text{max}}}]\). If stock-outs occurred during the simulation, the missing quantity was transferred into the next period.

Simulation results are shown in Table 5, where the Real column represents simulation results of the RP, while the other columns show the results of the VP. Results for stock-outs \( z \), maximum inventory level \( S \), and total cost \( c \) are represented for each item. The best strategy among VP algorithms for each item is represented in bold, and it was assessed by the FSA described earlier. The research team had supported the FSA assessments. Results in the column Real are omitted from the assessment and are used in the following table providing a summary of simulation results (Table 6).

The results in Table 6 show a comparison of RP and the strategy chosen by the FSA among VP strategies. Significant savings were achieved for almost all items, except for Item 8 marked with superscript a. For this item, optimization algorithms could not achieve any savings; on the contrary, they produced higher costs than the RP. Evidently, the order manager has found a better order placement policy than the ones proposed by the algorithms.

The results presented in Table 6 also indicate that the FZY algorithm was chosen for all items, except one, as the one yielding the best results. LUC, PPB and S–M algorithms were never chosen. As the FRP algorithm was chosen for Item 4, it is clear that it needs a strict order placement policy than the ones proposed by the algorithms.

In-depth cost analysis is presented in Tables 7 and 8. Table 7 shows the share of fixed ordering costs \( c_0 \), costs of taking over products \( c_r \), transportation cost \( c_t \), cost of physical storage \( c_p \), and cost of capital \( c_c \) with regard to total cost \( c \) for each item. One can notice that transportation cost, cost of physical storage, and cost of capital have a predominant share for most items. Items 2, 6 and 9 are exceptions and also have a dominant ordering cost.

Table 8 presents each cost category’s increase or decrease, signed by “+” or “−” respectively, of the selected VP strategy (as indicated in Table 6) regarding RP. Savings for Items 1 and 7 are mainly due to the major decrease in the physical storage cost and capital cost, whereas transportation cost reduction was crucial for Items 2, 3, 5 and 6. The simulation for Item 4 has mainly reduced
transportation cost, cost of physical storage, and cost of capital, which are predominant for this item. The total cost increased for Item 8, because the cost of physical storage and cost of capital have increased. The ordering cost and transportation cost needed to be reduced to reduce the total cost for Item 9.

Although physical storage cost and capital cost increased largely for Items 2, 5 and 9, it can be concluded that it is essential to reduce the transportation cost for these items; it is preferable to have more in stock rather than order more frequently, thus producing a lower transportation cost. However, Items 1 and 7 need frequent

Table 5
Simulation results of inventory control algorithms for all items

<table>
<thead>
<tr>
<th>Item</th>
<th>Criteria</th>
<th>Real</th>
<th>FRP</th>
<th>PPB</th>
<th>LUC</th>
<th>S–M</th>
<th>FZY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( z \times 10^1 )</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>( S \times 10^4 )</td>
<td>4.32</td>
<td>1.88</td>
<td>2.95</td>
<td>1.88</td>
<td>3.15</td>
<td>3.41</td>
</tr>
<tr>
<td>3</td>
<td>( c \times 10^6 )</td>
<td>2.50</td>
<td>1.52</td>
<td>1.47</td>
<td>1.39</td>
<td>1.41</td>
<td>1.46</td>
</tr>
<tr>
<td>4</td>
<td>( z \times 10^1 )</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>( S \times 10^5 )</td>
<td>6.25</td>
<td>5.08</td>
<td>6.21</td>
<td>7.41</td>
<td>6.42</td>
<td>9.79</td>
</tr>
<tr>
<td>6</td>
<td>( c \times 10^6 )</td>
<td>8.59</td>
<td>9.56</td>
<td>9.73</td>
<td>8.53</td>
<td>7.76</td>
<td>8.41</td>
</tr>
<tr>
<td>7</td>
<td>( z \times 10^1 )</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>( S \times 10^5 )</td>
<td>1.04</td>
<td>2.11</td>
<td>2.12</td>
<td>2.20</td>
<td>2.18</td>
<td>2.25</td>
</tr>
<tr>
<td>9</td>
<td>( c \times 10^6 )</td>
<td>2.75</td>
<td>1.88</td>
<td>1.90</td>
<td>1.93</td>
<td>1.93</td>
<td>1.95</td>
</tr>
</tbody>
</table>

Table 6
Summary of simulation results

<table>
<thead>
<tr>
<th>Item</th>
<th>RP cost</th>
<th>VP cost</th>
<th>Savings (%)</th>
<th>The best VP strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 2.50 \times 10^6 )</td>
<td>( 1.33 \times 10^6 )</td>
<td>46.8</td>
<td>FZY</td>
</tr>
<tr>
<td>2</td>
<td>( 2.75 \times 10^6 )</td>
<td>( 1.68 \times 10^6 )</td>
<td>38.9</td>
<td>FZY</td>
</tr>
<tr>
<td>3</td>
<td>( 8.59 \times 10^6 )</td>
<td>( 6.05 \times 10^6 )</td>
<td>29.6</td>
<td>FZY</td>
</tr>
<tr>
<td>4</td>
<td>( 2.32 \times 10^6 )</td>
<td>( 1.76 \times 10^6 )</td>
<td>24.1</td>
<td>FZY</td>
</tr>
<tr>
<td>5</td>
<td>( 6.77 \times 10^6 )</td>
<td>( 3.82 \times 10^6 )</td>
<td>43.6</td>
<td>FZY</td>
</tr>
<tr>
<td>6</td>
<td>( 1.67 \times 10^6 )</td>
<td>( 1.37 \times 10^6 )</td>
<td>25.9</td>
<td>FZY</td>
</tr>
<tr>
<td>7</td>
<td>( 7.21 \times 10^6 )</td>
<td>( 1.11 \times 10^6 )</td>
<td>65.4</td>
<td>FZY</td>
</tr>
<tr>
<td>8</td>
<td>( 1.03 \times 10^6 )</td>
<td>( 1.32 \times 10^6 )</td>
<td>28a</td>
<td>FZY</td>
</tr>
<tr>
<td>9</td>
<td>( 2.65 \times 10^5 )</td>
<td>( 2.51 \times 10^5 )</td>
<td>5.3</td>
<td>FZY</td>
</tr>
</tbody>
</table>

a VP cost is higher than RP cost.

descriptive text...

Table 7
The share (percentage) of fixed ordering costs \( c_o \), costs of taking over products \( c_b \), transportation cost \( c_t \), cost of physical storage \( c_p \), and cost of capital \( c_c \) with regard to total cost for each item

<table>
<thead>
<tr>
<th>Cost category</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>( c_o )</td>
<td>6.72</td>
</tr>
<tr>
<td>( c_b )</td>
<td>8.05</td>
</tr>
<tr>
<td>( c_t )</td>
<td>13.61</td>
</tr>
<tr>
<td>( c_p )</td>
<td>34.03</td>
</tr>
<tr>
<td>( c_c )</td>
<td>37.59</td>
</tr>
</tbody>
</table>
order and less in stock since capital and physical storage costs have the biggest impact on the total cost.

5. Inventory control simulation model validation and implementation

With the promising results discussed in the previous section, the goal is to implement this simulation system in the company’s information system, thus offering a decision support system to the order managers. Since the savings mentioned above are much higher than expected, they cannot be accepted as reliable; therefore, a time frame of one year has been chosen to validate the model thoroughly and train the order managers. The inventory control model will be implemented with the concept of virtual reality – the Virtual Warehouse (VW).

For the process of validation and implementation, several significant items held in the warehouse were selected. Information about costs, the production plan, orders, and delivery is exported from the SAP environment to the VW. The information is processed in real time. During the one-year period, the validation and implementation process will adhere to the following principles:

1. The order manager will use his inventory control policy in SAP without any changes.
2. The VW will be running simultaneously with the first principle. The simulator will be placing orders at the same time as the order manager. It will, however, be using optimization algorithms to determine order quantity.
3. The second VW will also be running simultaneously with the first two principles. This simulator will be placing orders autonomously, using optimization algorithms to determine order quantity and order schedule.

During this period, the order managers will actively participate in VW results analysis and optimization algorithms implementation.

6. Conclusion

Simulation results yielded significant cost savings. The FZY algorithm surpassed the results of other algorithms, including conventional ones like Least-unit cost, Part period balancing, and Silver–Meal, for all items except one, where the FRP algorithm had yielded the best results. The savings exceeded expectations by far as usually a saving of a few percentage points is considered a significant success. Although the results are repeatable and logical validation was accepted from the team, the simulation model will be validated in the future by using the concept of virtual reality before its actual implementation in the inventory control process.

The simulator’s transparency and visual presentation can play a significant role in the learning process of using new optimization algorithms as people tend to resist using methodologies that they are unfamiliar with, especially if they are used to controlling the process using only their experience and intuition. Considering the simulator’s characteristics, we can conclude that the order manager could quickly understand and adapt to a new methodology used in the inventory control process.

Persuading people in management about the profitability of using modern optimization methodologies is another advantage of the simulator as modern technology provides a way to efficiently analyse the current state of the process and its future behaviour. Equipped with a visual presentation, the simulator can give a more presentable and understandable insight into sophisticated techniques which can offer tremendous support in the decision process, thus providing assistance to a company in cutting costs.

Results in this study are based on historical demand; however, some preliminary research has been made on the actual production plan. It has been confirmed that the actual production plan is unreliable and not sufficient enough to make a good order placement policy, but is still conveying some information. Future work would mean improving the order placement process by also including historical data and by setting appropriate weights to emphasize the historic or future component in the process of anticipation. Our goal is to implement the simulation model into the company’s information system as a supplement to the expert’s replenishment strategy, thereby increasing man–machine interaction in order to improve the order placement process.

Acknowledgement

This research was supported by the Ministry of Higher Education, Science and Technology of the Republic of Slovenia (Programme No. UNI-MB-0586-P5-0018).
References


