International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

A rolling horizon simulation approach for managing demand with lead time variability

Francisco Campuzano-Bolarín , Josefa Mula , Manuel Díaz-Madroñero & Álvar-Ginés Legaz-Aparicio

To cite this article: Francisco Campuzano-Bolarín , Josefa Mula , Manuel Díaz-Madroñero & Álvar-Ginés Legaz-Aparicio (2020) A rolling horizon simulation approach for managing demand with lead time variability, International Journal of Production Research, 58:12, 3800-3820, DOI: 10.1080/00207543.2019.1634849

To link to this article: https://doi.org/10.1080/00207543.2019.1634849

Published online: 03 Jul 2019.

International Journal of Production Research, 2020
Vol. 58, No. 12, 3800–3820, https://doi.org/10.1080/00207543.2019.1634849

A rolling horizon simulation approach for managing demand with lead time variability

Francisco Campuzano-Bolarín [a], Josefa Mula[b*], Manuel Díaz-Madroñero [b] and Álvar-Ginés Legaz-Aparicio[a]

[a]*Department of Business Economics, Universidad Politécnica de Cartagena, Cartagena, Spain;* [b]*Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València, Escuela Politécnica Superior de Alcoy, Alicante, Spain*

(*Received 21 December 2018; accepted 12 June 2019*)

This paper proposes a rolling horizon (RH) approach to deal with management problems under dynamic demand in planning horizons with variable lead times using system dynamics (SD) simulation. Thus, the nature of dynamic RH solutions entails no inconveniences to contemplate planning horizons with unpredictable demands. This is mainly because information is periodically updated and replanning is done in time. Therefore, inventory and logistic costs may be lower. For the first time, an RH is applied for demand management with variable lead times along with SD simulation models, which allowed the use of lot-sizing techniques to be evaluated (Wagner-Whitin and Silver-Meal). The basic scenario is based on a real-world example from an automotive single-level SC composed of a first-tier supplier and a car assembler that contemplates uncertain demands while planning the RH and 216 subscenarios by modifying constant and variable lead times, holding costs and order costs, combined with lot-sizing techniques. Twenty-eight more replications comprising 504 new subscenarios with variable lead times are generated to represent a relative variation coefficient of the initial demand. We conclude that our RH simulation approach, along with lot-sizing techniques, can generate more sustainable planning results in total costs, fill rates and bullwhip effect terms.

Keywords: rolling horizon; demand management; simulation; supply chain dynamics

1. Introduction

The rolling horizon (RH) approach has been addressed mainly in the scientific literature in material requirement planning (MRP) (Whybark and Williams 1976), and in master production scheduling (MPS) (Sridharan, Berry, and Udayabhanu 1988) and supply chain (SC) production planning problems (Boulaksil, Fransoo, and van Halm 2009), among others. This approach is considered a flexible planning tool to help adapt to a planning horizon with uncertain information that contains unpredictable behaviour parameters by determining a schedule in which to work with as few errors as possible (Baker 1977; de Sampaio, Wollmann, and Vieira 2017; Sahin, Narayanan, and Robinson 2013). Nevertheless, Sahin and Robinson (2002, 2005) stress the need to conduct new research works based on the RH approach for SC management. It is important to highlight at this point that most RH approaches are based on mathematical programming or optimisation models (exact or heuristic algorithms) constrained for real situations under dynamic parameters. Thus, on the one hand, to deal with demand uncertainty in supply chains, Boulaksil, Fransoo, and van Halm (2009) determine safety stock levels while carrying out different experiments with mathematical programming formulations in an RH setting, whereas Rafiei et al. (2014) develop a decision platform to select the best production planning policy in a wood remanufacturing SC by with various mixed-integer programming models and a periodic replanning strategy based on an RH procedure. On the other hand, Simpson (1999) and van den Heuvel and Wagelmans (2005) compare the effectiveness of different lot-sizing and heuristic rules for production planning problems in an RH context. Vargas and Metters (2011) propose an optimisation algorithm for the MPS problem, which considers the stochastic character of demand levels based on the RH paradigm to achieve superior performance than traditional lot-sizing techniques. Lian, Liu, and Zhu (2010) address the replenishment problem with stochastic demand, and provide analytical formulas and algorithms to calculate the optimal order policy for the two-period RH problem as a dual-threshold type for updated information and a base-stock level. For a more extensive review on rolling horizon planning in SC, we refer readers to Sahin, Narayanan, and Robinson (2013) and to de Sampaio, Wollmann, and Vieira (2017).

Lot-sizing is one of the most important, and also one of the most difficult, problems in production systems. Indeed making the right decisions in lot-sizing will directly affect system performance and its productivity levels (Karimi, Fatemi Ghomi, and Wilson 2003). In order to help practitioners in the decision-making process of lot-sizing in production planning

*Corresponding author. Email: fmula@omp.upv.es

or replenishment scenarios, several exact and heuristic approaches have been developed in the literature, such as the Wagner and Whitin (1958) (WW) algorithm or the Silver and Meal (1973) (SM) method, the economic order quantity (EOQ) or order up to level (OUT) policies, which are widely used in industry, among others. The choice of one type of method or another for lot-sizing calculations or replenishment rules can imply variability existing in the placed orders, whose amplification generates the bullwhip effect (Bhattacharya and Bandyopadhyay 2011; Jakšič and Rusjan 2008; Lee et al. 1997a). According to Giard and Sali (2013), the main modelling approaches that can reflect the existence of the bullwhip effect in SCs are analytical formulations and simulation, specially system dynamics (SD) simulation, discrete-event (DE) simulation and multi-agent (MA) simulation. Simulation is a tool to be used in problem situations which are analytically intractable, or to confirm the results of analytic models (Wemmerlov and Whybark 1984). Using numerical simulations, Pujawan (2004) examined several properties of the SM and the least unit cost (LUC) rules on variability of orders created by an SC channel receiving uncertain demand from its downstream channel. Schmidt, Münzberg, and Nyhuis (2015) considered DE simulation techniques to compare different academic and practice-based lot-sizing approaches and their impact on real-world industrial systems. Pacheco et al. (2017) proposed a new replenishment policy based on the classical OUT and compared its performance in a multi-echelon SC with demand and lead times variabilities in bullwhip effect and service level terms by DE simulation. Dominguez, Cannella, and Framinan (2015) quantified the impact of different SC structures on their corresponding bullwhip effect with an MA simulation study for different business scenarios. Regarding SD simulation, Hussain and Drake (2011) studied the relationship between order batching and the bullwhip effect in a multi-echelon SC with information sharing by an SD approach. Li, Ghadge, and Tiwari (2016) compared the EOQ and target stock level replenishment strategies for small firms by focusing on e-business by developing an SD model that ran in different scenarios to reflect the overall behaviour of the SC performance indicators used for comparisons in bullwhip effect, service level and inventory cost terms.

Mathematical and analytical approaches can demand an academically advanced understanding of mathematics, something that most SC operations managers do not have (Agaran, Buchanan, and Yurtseven 2007). Using SD simulation methods can help practitioners to better understand SC indicators and to examine the effects of different parameters and their variability (Hussain and Drake 2011). For instance, in the order management context, Esteso et al. (2018) compared SD and mathematical programming models. This comparison revealed that the SD model better performs as the number of orders increases with near-optimum solutions in a shorter time. Our research focuses on demonstrating the usefulness and validity of the RH approach jointly with SD simulation to illustrate the advantages of both approaches, the RH to mitigate effects of uncertainty, and SD simulation to obtain competitive solutions in minimum computational times.

The SD methodology was herein used to deal with the problem of the impact that variable lead times have on demand management in the RH context. It allowed to model demand management throughout the SC during the considered planning horizon. For a review on SD models in the SC context, we refer readers to Campuzano-Bolarín, Mula, and Peidro (2013); Campuzano, Mula, and Peidro (2010); Mendoza, Mula, and Campuzano-Bolarin (2014) and Moreno, Mula, and Campuzano-Bolarin (2015), among others. Thus, it would be possible to analyse the impact of lead time variability on different key performance indicators (KPIs) of SC. In this way, the combination of both RH and SD approaches could act as a tool to manage demand in variable lead times contexts during the planning horizon by allowing distorted demand, or the bullwhip effect, to be analysed (Fransoo and Wouters 2000; Hosoda and Disney 2018; Lee, Padmanabhan, and Whang 1997a, 1997b) throughout the SC.

To the best of our knowledge, no papers address an SC demand management problem with an RH approach based on SD simulation to simulate different what-if scenarios in order to evaluate the bullwhip effect, lead time variability and different replenishment rules. This paper proposes a novel approach, dubbed as RH-SD-Java, based on an RH approach along with SD simulation to manage dyadic SC demands with lead time variability. Moreover, in order to process RH input and to calculate production orders, it proposes an external subroutine to carry out these tasks, which was done in the Java programming environment. Lot-sizing calculations are based on Silver and Meal (1973) and on Wagner and Whitin (1958). These techniques are applied to information about discretised planning horizon demands and provide the replanning of orders at a minimum cost. Moreover, which traditional SC parameters can help to cushion the bullwhip effect has also been studied (Lee, Padmanabhan, and Whang. 1997a, 1997b); i.e. reduce the distortion or variability of upstream customer demand, which is an important indicator in an uncertain demand management process with lead time variability.

This paper intended to answer the research question about whether the integrated RH and SD simulation proposed can provide new dynamic insight into classic static lot-sizing approaches based mainly on optimisation or heuristic rules. Hence the main contributions of this paper were to: (i) provide a simulation model using RH information to face demand and lead time variability with minimum inventory costs and at higher fill rates; (ii) reduce the bullwhip effect by using RH simulation techniques (SD plus Java programming) with unknown lead times; (iii) develop a new easy-to-use tool for SC planners to manage demand under uncertainty and with variable lead times by an RH approach and lot-sizing techniques;

(iv) validate the proposal through different what-if simulation scenarios based on the use of distinct parameters related to lot-sizing techniques.

The rest of the paper is organised as follows. Section 2 describes the proposal for demand management with lead time variability. Section 3 formulates the RH approach, the SD based-simulation model and the corresponding solution algorithms. Section 4 presents computational experiments and validates the proposed model. Finally, Section 5 provides the conclusions and further research lines.

2. The RH-SD-Java framework

A dyadic SC formed by one customer and one manufacturer was considered according to an e-shopping SC structure that refers to the scenario where the manufacturer receives orders directly from a customer and supplies products directly after the production lead time (Disney, Naim, and Potter 2004). The main point of this proposal lies in an SD model, which simulates demand management in various scenarios; that is, constant and variable lead times. This model, implemented with the Vensim® simulation software, receives input information from spreadsheets for the demand management process. Next, depending on the model's inventory state and on the time required to deliver production orders, it calls on a subroutine that is in charge of carrying out lot-sizing according to RH data (also from spreadsheets). Then the orders lot-sized by this subroutine are used by the simulation model to control demands. An SD environment is created to simulate and compare different manufacturing scenarios with the information saved from previous simulation results. Figure 1 describes the RH-SD-Java decision framework.

For example, if the demand data from a spreadsheet environment (a) comprise 500 units, the system dynamics environment (c) checks whether the inventory state is able to accomplish future demands. If it cannot because, e.g. the inventory state has 200 units, the Java environment (b) determines, with the RH data (forecasted demand) from the spreadsheet environment (a), the manufacturing orders lot-size by SM or WW techniques. The manufacturing orders that result from the Java environment (b) are sent to the spreadsheet environment (d) to be later compared with other orders created in constant or variable lead time environments with three different SD models (e) by SM, WW or OUT-S techniques in total costs, fill rates and bullwhip effect terms.

3. The RH-SD-Java approach

3.1. The RH approach

The RH approach has been addressed mainly in the scientific literature in material requirement planning (MRP) (Díaz-Madroñero, Mula, and Peidro 2017; Mula, Poler, and Garcia-Sabater 2007, 2008; Mula, Poler, and Garcia 2006; Whybark and Williams 1976), the master production scheduling (MPS) (Sridharan, Berry, and Udayabhanu 1988) and SC production planning problems (Boulaksil, Fransoo, and van Halm 2009; Mula, Peidro, and Poler 2010). This is chiefly because it is a proven flexible planning tool (de Sampaio, Wollmann, and Vieira 2017; Sahin, Narayanan, and Robinson 2013) for environments with little information and planning horizons containing variable performance parameters. The objective is to determine a dynamic functioning schedule with as few errors as possible along a static planning horizon. This method works with a set of planning horizons, PH_i, which result from the discretisation or segmentation of a longer planning horizon. In these PH_i which have shorter time periods than a total planning horizon, fewer planning errors can be made. Moreover, these errors diminish for all the PH_{i+1} as each one can put to good use the information generated by previous planning horizons, PH_{i-1}, in such a way that when each PH_{i+1} is replanned, it can enhance the certainty of input data and can, therefore, reduce output data errors.

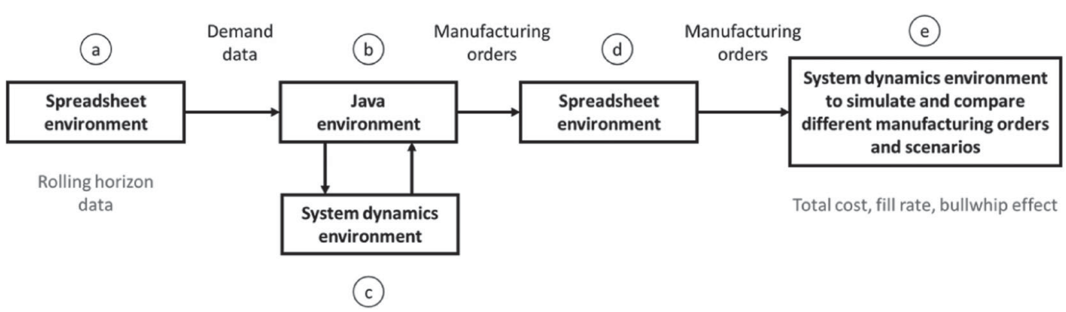

Figure 1. The RH-SD-Java framework.

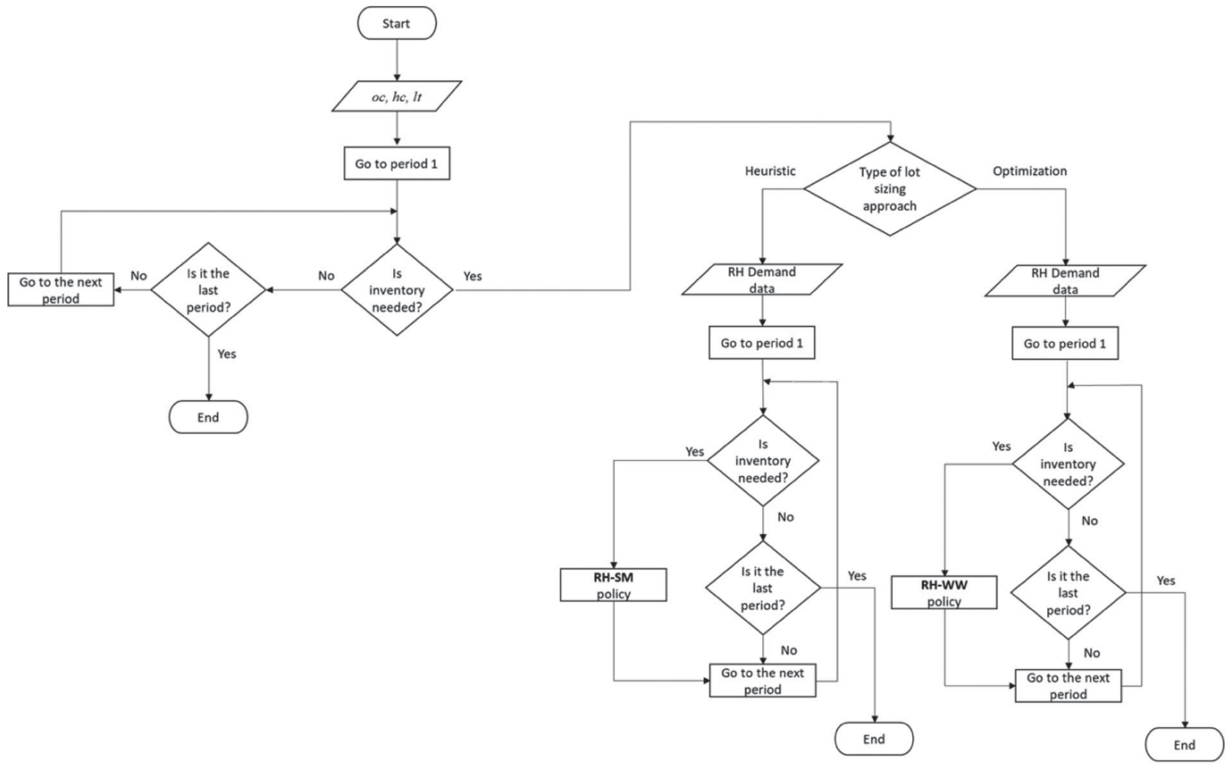

Figure 2. Flow chart of the demand management process simulation.

In the present work, the proposed RH was modelled and programmed in a Java programming environment. The choice of Java as a programming environment was due mainly to two factors: specific libraries exist that call on Vensim® from this programming environment; this programming language's simplicity and power. The RH functions as follows: when the Java model begins to run, it considers the first PH_i (PH_1) on which demand management is dealt with by generating manufacturing orders according to the forecast demand and the inventory state (Section 3.3). The RH approach works by updating real demands for each planning horizon time period instead of considering the forecasted demands for a given planning horizon. After solving this PH_i, the Java model acquires information about inventory levels so that when the RH updates the next PH_i (PH_{i+1}), it makes the most of all the information from the previous stage, and the replanning of this new PH_{i+1} includes the stockouts that might take place throughout the planning horizon during simulation. This procedure is repeated in each discretised PH_i with time. Figure 2 depicts the flow chart of the demand management process simulation according to an RH approach.

3.2. SD model formulation

Here we consider a dyadic single-product SC. Model formulation is based on the APIOBPCS (automatic pipeline, inventory and order-based production control system) (Campuzano, Mula, and Peidro 2010; John, Naim, and Towill 1994) and the previous works by Mula, Poler, and Garcia-Sabater (2008) and Díaz-Madroñero, Mula, and Jiménez (2014). The main contemplated characteristics are as follows:

- A one-stage SC consisting in a customer and a manufacturer, with customer orders to the manufacturer
- The manufacturer supplies products immediately upon receiving orders according to on-hand inventory levels. A pull planning strategy is considered
- Orders can be partially delivered and unfulfilled orders are backlogged
- Supplied goods arrive after a lead time
- Raw materials are considered infinite

Table 1 provides the nomenclature of the SD model. Figure 3 depicts the flow chart of the simulation model. Table 1 describes the nomenclature of the flow variables, level variables and auxiliary variables that the model employs (Forrester 1961).

Table 1. Nomenclature of the SD model.

Level variables	
MBO	Manufacturer's backordered orders (units)
MFR	Manufacturer's fill rate (%)
MOHI	Manufacturer's on-hand inventory (units)
MTC	Manufacturer's total costs (€)
MWIP	Manufacturer's work-in-process (units)
Flow variables	
BOD	Backordered orders delivered (units per time period)
MCC	Manufacturing capacity constraints (units per time period)
PDC	Products delivered to customer (units per time period)
PDM	Products delivered to manufacturer (units per time period)
Auxiliary variables	
be	Bullwhip effect (dmnl)
c	Manufacturer capacity (units)
d	End customer demand (units per time period)
ei	Expected inventory status (units per time period)
f	Manufacturer's forecast (units per time period)
fo	Firm orders (units per time period)
hc	Holding cost (€ per unit and time period)
lt	Manufacturing lead time (time periods)
mhc	Manufacturer's holding costs (€)
mis	Manufacturer inventory status (units per time period)
moc	Manufacturer's order costs (€)
mpc	Manufacturer's production costs (€)
msc	Manufacturer's stockout costs (€)
o	Orders to manufacturer (units per time period)
oc	Ordering cost (€ per order)
pc	Production cost (€ per unit)
per	Performance of the manufacturing capacity (%)
sc	Stockout cost (€ per unit)

Launching orders to be manufactured, *o* are determined by demand forecasts, *f*, backorders, *MBO*, on-hand inventory, *MOHI*, and orders being manufactured, *MWIP*.

Equation (1) determines the possible backorders, *MBO*.

$$MBO(t) = \int_{t_0}^{t} \left[-BOD(t) + \begin{cases} 0, & \text{if } MOHI(t) + PDM(t) - fo(t) \geq 0; \\ & \begin{cases} (d - PDC(t)), & \text{if } PDC(t) < d, \\ 0, & \text{in otherwise} \end{cases} \end{cases} \right] \mathrm{d}t. \tag{1}$$

Equation (2) determines the available inventory in the manufacturer's warehouse.

$$MOHI(t) = \int_{t_0}^{t} [PDM(t) - PDC(t)]\mathrm{d}t + MOHI(t_0). \tag{2}$$

Equation (3) defines the manufacturer's work-in-process, *MWIP*.

$$MWIP(t) = \int_{t_0}^{t} [o(t)] - PDM(t)]\mathrm{d}t + MWIP(t_0). \tag{3}$$

Equation (4) represents the total manufacturing cost, *MTC*, defined by the sum of the order costs, *moc*, storage costs, *mhc*, stockout costs, *msc*, and production costs, *mpc*.

$$MTC(t) = \int_{t_0}^{t} [moc(t) + mhc(t) + msc(t) + mpc(t)]\mathrm{d}t + MTC(t_0). \tag{4}$$

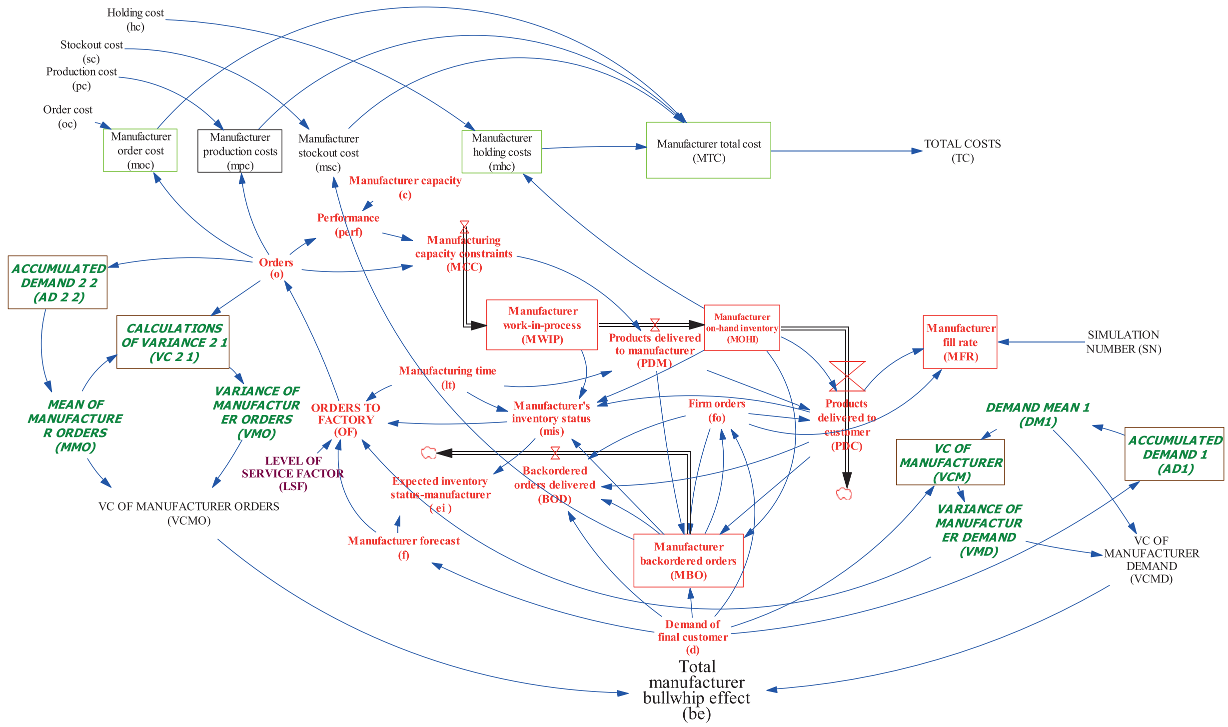

Figure 3. Part of the flow chart of the SD model from the demand management system.

Equation (5) determines the manufacturer fill rate, which is defined as the relationship between the products delivered to customers, $PDC(t)$, and the firm orders $fo(t)$ during the time periods.

$$MFR(t) = \begin{cases} \left(\dfrac{PDC(t)}{fo(t)}\right) \cdot \dfrac{1}{last(t)}, & \text{if } fo(t) > 0; \\ 1 \cdot \dfrac{1}{last(t)}, & \text{otherwise} \end{cases}. \tag{5}$$

Equation (6) calculates the products delivered to customers, PDC.

$$PDC(t) = \begin{cases} fo(t), & \text{if } (MOHI(t) + PDM(t) - fo(t)) \geq 0; \\ MOHI(t), & \text{otherwise} \end{cases}. \tag{6}$$

The backordered orders delivered, BOD, are stated in Equation (7).

$$BOD(t) = \begin{cases} MBO(t), & \text{if } PDC(t) = fo(t); \\ \begin{cases} (PDC(t) - d(t)), & \text{if } PDC(t) > d(t) \\ 0, & \text{in otherwise} \end{cases} \end{cases}. \tag{7}$$

The manufacturing capacity constraints, $MCC(t)$, are defined in Equation (8).

$$MCC(t) = \begin{cases} 0 * \text{perf}, & \text{if perf} < 1; \\ 0, & \text{in otherwise} \end{cases}. \tag{8}$$

Finally, the products delivered to the manufacturer, PDM, are defined in Equation (9).

$$PDM(t) = MCC(t + lt). \tag{9}$$

The seminal definition (going back to Forrester and 1961) of the bullwhip effect calculation in an e-shopping SC can be distorted. Although the bullwhip effect is mainly the result of lack of transparency across multiple echelons and with many parties in each echelon (Lee et al. 1997), Disney, Naim, and Potter (2004) proved that the bullwhip effect still exists in a single-level SC. Here the bullwhip effect indicator proved to be a useful tool to measure distortion in orders due to lead time variability and lot-sizing techniques in a single-echelon SC. Other authors have also proved this (Li, Ghadge, and Tiwari 2016). Thus in accordance with Fransoo and Wouters (2000), the bullwhip effect, *be*, is measured at a particular level in a multilevel SC as the quotient of the coefficient of variation in the demand generated by this level, C_{out}, and the coefficient of variation of the demand received at this level, C_{in}:

$$be = \frac{C_{\text{out}}}{C_{\text{in}}}, \tag{10}$$

where

$$C_{\text{out}} = \frac{\sigma(o(t, t+T))}{\mu(o(t, t+T))}, \tag{11}$$

and,

$$C_{\text{in}} = \frac{\sigma(d(t, t+T))}{\mu(d(t, t+T))}, \tag{12}$$

where $d(t, t+T)$ and $o(t, t+T)$ are the demands and orders during the time interval $(t, t+T)$, σ is the standard deviation and μ is the mean. This bullwhip effect measure (10), thought for a particular level in a multi-level SC, is also adequate for a single-echelon or e-shopping SC structure where, upstream and downstream bullwhip cyclicality (Geary, Disney, and Towill 2006), known as the boom-and-bust scenario in a multilevel SC, it does not need to be captured.

3.3. *Lot-sizing*

As previously mentioned, a subroutine programmed in Java is in charge of lot-sizing, which implements different lot-sizing methods. Below the pseudo codes developed for WW (Wagner and Whitin 1958) and SM (Silver and Meal 1973) lot-sizing algorithms, respectively, are presented. They send information to the SD model, as detailed in Section 2. The Java model formulation is included in the Annexes of this paper.

The pseudo code for WW is:

```
cost [0] = 0;
FOR n, from 1 to the maximum number of periods
        FOR p, from 0 to (n-1)
              current_cost =  cost[p] + cost of ordering during period p, all the demand data from p to n-1
              IF current_cost is the minimum:
                      cost[n] = current_cost
              END IF
        END FOR
END FOR
```

and, the pseudo code for SM is:

```
cost [0] = 0;
processed_index = 0
WHILE processes_index  <  number of demands
      FOR i from 1 to number of demands
            current_cost = calculate cost of ordering during period 'processed_index' all the demands between that period and 'processed index' + 1
            IF current_cost is higher than the last calculated
                  number of periods = i-1
                                                   // calculate accumulated cost
                  cost = cost + cost from process_index to (process_index + number of periods)
                  BREAK FOR
            END IF
      END FOR
      processed_index = processed_index  +  number of periods
END WHILE
RETURN cost
```

4. The RH-SD-Java application

In order to apply and validate the created and modelled RH-SD-Java approach, the results were compared and offered in costs, service level and bullwhip effect terms for different manufacturing orders. That is, the OUT-S (order up to level S) (Silver, Pyke, and Peterson 1998) (see the Annexe) orders simulated in an SD model (Campuzano, Mula, and Peidro 2010) and those which use lot-sizing according to the SM and WW techniques (Table 2), which are created by the RH-SD-Java approach during all the simulation periods according to the inventory requirements to cover demand.

The setup data for the experiments are based on a real-world example taken from an automotive SC composed of a first-tier supplier and a car assembler, which has been previously addressed in our research (Díaz-Madroñero, Mula, and Jiménez 2014; Mula, Poler, and Garcia-Sabater 2008). From these input data, four main scenarios were recreated and simulated according to different lead times ($lt = 2$ weeks, $lt = 3$ weeks, $lt = 4$ weeks and $lt = variable$; 2–4 weeks) (Table 3). In parallel, all four scenarios contemplated several subscenarios determined by different holding cost values, *hc*, and order cost values, *oc*; as shown on the right in Table 3. In order to measure the simulated SC performance, total costs (sum of order costs, *oc*, holding costs, *hc*, stockout costs, *sc* and production costs, *pc*), service level, *MFR*, and bullwhip effect, *be,* were used as the main parameters. Hence it was possible to determine which manufacturing order (OUT-S, RH-SM or RH-WW) implied improvements in the aforementioned parameters according to the different costs and lead times (constant or variable).

The characteristics of the simulated scenarios are as follows:

- Constant and variable lead times (Table 3, left). Specifically, four possible lead times values were considered, three of which contemplated constant lead times (*lt2, lt3, lt4*), while another scenario considered variable lead times (*ltv*).
- Different values were contemplated for the holding costs, *hc*, and order costs, *oc* (Table 3).
- Production costs, *pc*, and stockout costs, *sc*, remained constant for each simulated scenario (Table 3).

All the initial values for the level variables were defined as null, except the manufacturer on-hand inventory, *MOHI*, which was set up as 1458 units according to the used dataset taken from Mula, Poler, and Garcia-Sabater (2008) and Díaz-Madroñero, Mula, and Jiménez (2014).

Table 4 shows the considered actual demands (highlighted in bold) in relation to the forecasted demands. Thus Table 4 presents the demand information released by the customer (the car assembler) during each time period of the considered planning horizon to the first-tier supplier. The highlighted demand information is firm for each first releasing time period, while for the other time periods demand information represents forecasts, which the RH-SD-Java approach manages beforehand. We can find 21 demand releases that contemplate 30 time planning periods in an RH context.

4.1. Simulating scenarios

The model is simulated using Vensim DLL® (Ventana Systems Inc.), simulation software for SD, and worked in a Java environment. Each RH-SD-Java model run consists of 216 simulation scenarios by combining constant or variable lead times with different lot-sizing policies and holding and order costs. The computer used to carry out the experiments has an Intel® i5 2.50 GHz processor with 8 GB of RAM memory and a 64-bits operating system. The average computational time

Table 2. Manufacturing order type.

Manufacturing order type	Description
OUT-S	Order up to level S
RH-SM	Rolling horizon with SM
RH-WW	Rolling horizon with WW

Table 3. Values of lead times (*lt*) and costs setting.

Lead time setting		Cost setting			
lt	Values of *lt* in weeks	Holding costs (€/(period*unit))	Order costs (€/order)	Stockout cost (€/unit)	Production cost (€/unit)
lt2	2	2; 5.39; 8	500; 1000; 1500	269.598	6.34
lt3	3				
lt4	4		2000; 2500; 3000		
ltv	variable (2–4)				

Table 4. Demand (d) corresponding to the total PH_i length conformed by 21 demand releases and 30 time periods.

Demand releases – Planning horizon (PH_i)	Time periods																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	**677**	780	647	358	537	419	0	0	231	324	324	324	395	395	395	395	372	372	372	372	266	266	266	266	266	0	0	0	0	0
2		**780**	729	351	617	517	0	0	226	363	306	306	395	395	395	395	371	371	371	371	264	264	264	264	264	0	0	0	0	0
3			**753**	339	578	581	0	0	151	238	221	245	434	434	434	434	316	316	316	316	268	268	268	268	268	0	0	0	0	0
4				**339**	620	551	0	0	157	240	225	244	434	434	434	434	316	316	316	316	268	268	268	268	268	311	311	311	311	0
5					**618**	510	0	0	470	587	560	572	880	892	892	892	380	380	380	380	306	306	306	306	306	361	361	361	361	0
6						**524**	0	0	430	937	766	134	915	858	904	904	379	379	379	379	307	307	307	307	307	362	362	362	362	0
7							**0**	0	429	942	774	194	872	868	872	852	380	380	380	380	306	306	306	306	306	380	380	380	380	0
8								**0**	344	942	784	203	987	941	1014	994	429	429	429	429	309	309	309	309	309	380	380	380	380	303
9									**344**	942	784	203	930	983	1049	985	503	407	407	407	308	308	308	308	308	380	380	380	380	304
10										**940**	697	203	1000	965	997	985	526	483	353	353	310	310	310	310	310	380	380	380	380	304
11											**728**	688	1275	1176	1204	415	417	446	168	433	328	328	328	328	328	360	360	360	360	313
12												**660**	1275	1179	1209	464	421	399	179	459	327	327	327	327	327	360	360	360	360	314
13													**1275**	1179	1209	466	391	354	206	392	364	287	287	287	287	340	340	340	340	298
14														**1179**	1207	749	747	801	419	739	565	483	369	369	369	291	291	291	291	307
15															**1206**	748	1038	1283	635	658	471	504	395	352	352	291	291	291	291	307
16																**748**	1039	1282	628	659	512	466	404	429	312	291	291	291	291	310
17																	**1039**	1304	634	656	562	558	411	434	342	333	333	333	333	419
18																		**1301**	636	681	655	635	502	511	379	467	469	469	469	428
19																			**636**	654	350	840	661	641	243	456	458	480	480	428
20																				**652**	625	858	827	616	246	443	447	458	477	455
21																					**611**	850	595	558	475	489	468	480	488	544

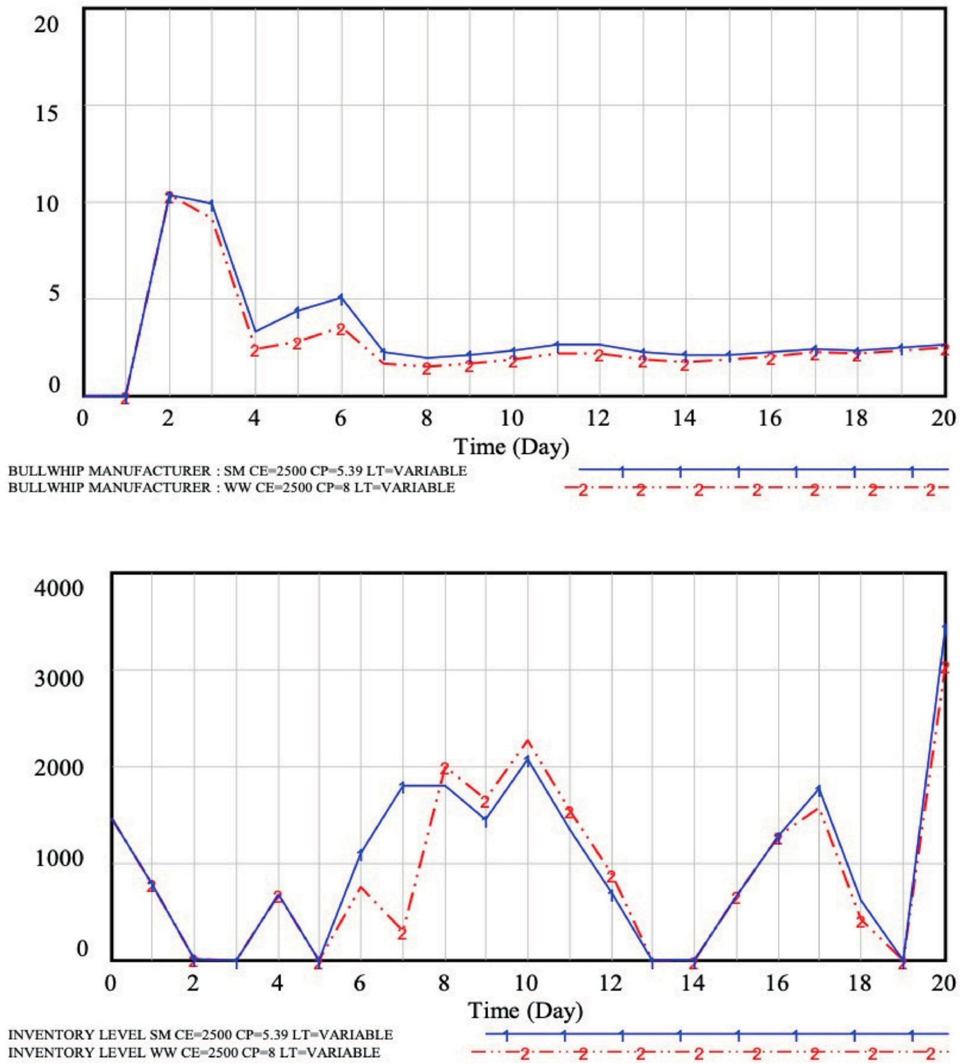

Figure 4. Steady state of the bullwhip effect KPI and the inventory level.

spent to solve the 216 simulation scenarios is about 3 min, which is not a critical point for practical use. By way of example, Figure 4 presents the bullwhip effect at the manufacturer level KPI, where we can see that the steady state is reached after approximately 10 time periods. Figure 4 also shows the warm up time of the inventory level and, as initial inventories are considered according to the dataset used, stockout initial periods can be avoided.

4.1.1. Constant lead times

Table 5 shows the results for the 54 simulated scenarios for a constant lead time of two unit times, *lt2*, and offers the results that reach the total costs, the service level, *MFR*, and the bullwhip effect, *be*, for the different *hc* and *oc* values. The results corresponding to scenarios *lt3* and *lt4* are also seen in Figures 5, 6 and 7.

The results obtained with constant lead times revealed how using manufacturing orders RH-SM and RH-WW provided better results in terms of costs and service levels than the OUT-S policy (Figures 5 and 6). Figure 5 shows how the RH-SD-Java approach performed better in terms of total costs for all the considered constant lead times, 2, 3 and 4, mainly because the demand updating from the RH data allowed the lot-sizing techniques to make more accurate calculations because the lot-size amounts were updated during each simulation time period, or demand releasing, which avoided higher costs due to demand uncertainty. Regarding the manufacturer fill rate, Figure 6 shows how the RH-SD-Java approach led to better performance with shorter lead times because the possible backorders were updated during each contemplated time period in such a way that they were added to the actual demand by creating more stability in the system.

Table 5. Total costs, fill rate, *MFR*, and bullwhip effect, *be*, for *lt*2.

Set up parameters	Manufacturing order type	$hc = 2$ Total costs (€)	$hc = 2$ *MFR* (%)	$hc = 2$ *be*	$hc = 5.39$ Total costs (€)	$hc = 5.39$ *MFR*(%)	$hc = 5.39$ *be*	$hc = 8$ Total costs (€)	$hc = 8$ MFR (%)	$hc = 8$ *be*
$oc = 500$	OUT S	1484.000	82.45	1.633	1543.000	82.45	1.633	1589.000	82.45	1.633
	RH-SM	1002.000	88.25	1.586	1069.000	88.32	1.511	1126.000	88.32	1.511
	RH-WW	994.500	88.32	1.511	1069.000	88.32	1.511	1126.000	88.32	1.511
$oc = 1000$	OUT S	1493.000	82.45	1.633	1552.000	82.45	1.633	1598.000	82.45	1.633
	RH-SM	994.700	88.43	1.886	1077.000	88.32	1.511	1135.000	88.32	1.511
	RH-WW	967.600	88.63	1.515	1077.000	88.32	1.511	1135.000	88.32	1.511
$oc = 1500$	OUT S	1502.000	82.45	1.633	1561.000	82.45	1.633	1607.000	82.45	1.633
	RH-SM	576.400	92.34	2.161	1094.000	88.25	1.586	1153.000	88.25	1.586
	RH-WW	780.800	89.87	1.768	1086.000	88.32	1.511	1143.000	88.32	1.511
$oc = 2000$	OUT S	1511.000	82.45	1.633	1570.000	82.45	1.633	1616.000	82.45	1.633
	RH-SM	643.300	94.21	2.007	1059.000	88.63	1.515	1161.000	88.25	1.586
	RH-WW	816.600	93.18	1.952	1059.000	88.63	1.515	1152.000	88.32	1.511
$oc = 2500$	OUT S	1520.000	82.45	1.633	1579.000	82.45	1.633	1625.000	82.45	1.633
	RH-SM	421.400	96.55	2.331	1103.000	88.43	1.886	1169.000	88.25	1.586
	RH-WW	429.300	96.43	2.089	1068.000	88.63	1.515	1160.000	88.32	1.511
$oc = 3000$	OUT S	1529.000	82.45	1.633	1588.000	82.45	1.633	1634.000	82.45	1.633
	RH-SM	27.000	99.04	2.950	674.300	92.38	1.714	1134.000	88.63	1.515
	RH-WW	264.900	98.97	2.012	1111.000	88.32	1.511	1134.000	88.63	1.515

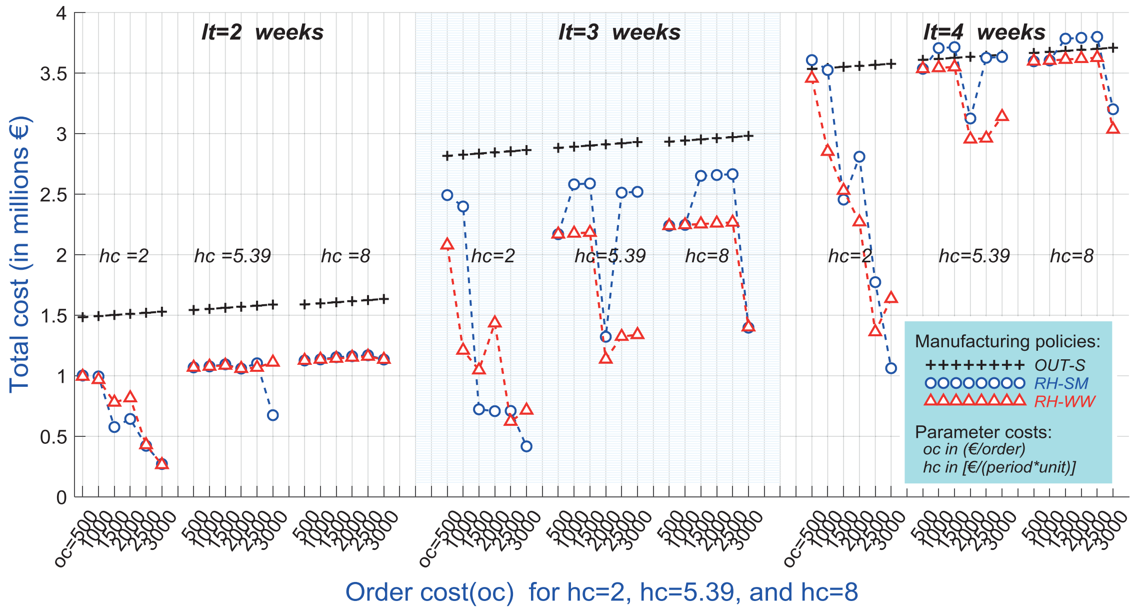

Figure 5. Total costs of manufacturing orders OUT-S, RH-SM and RH-WW.

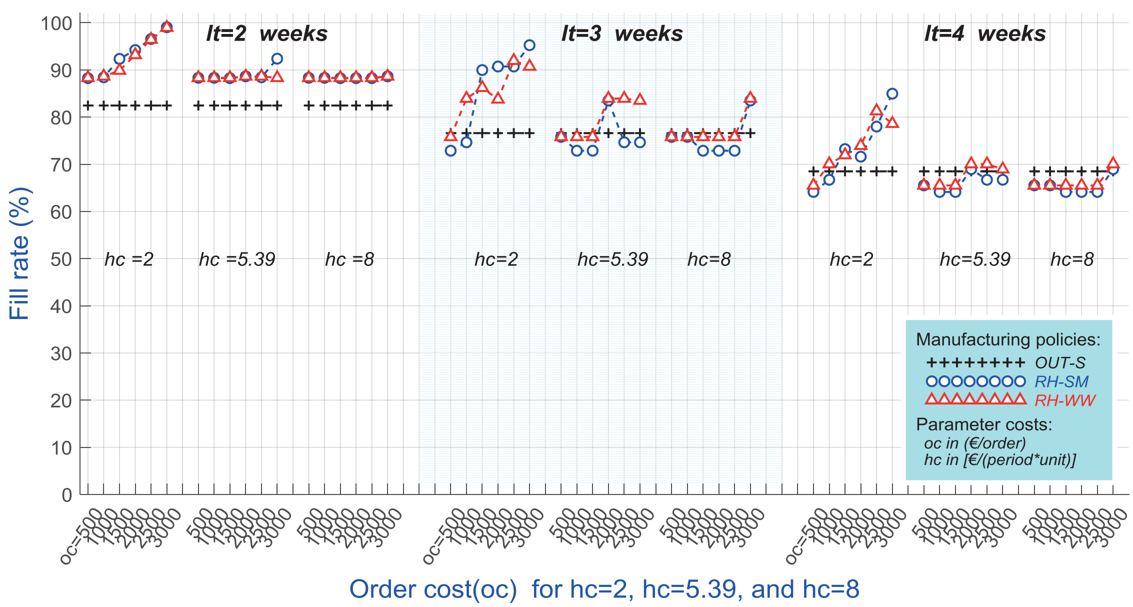

Figure 6. Fill rate of manufacturing orders OUT-S, RH-SM, RH-WW.

Figure 7 presents the bullwhip effect calculated to measure the distortion between generated orders and demand. We can see that as the holding cost goes up, the order distortion measured in bullwhip effect terms goes down. By using lot-sizing techniques, higher holding costs should yield smaller orders and reduce the order distortion. These results are in line with Lee, Padmanabhan, and Whang (1997a), who identified lot-sizing and non-null lead times as two of the main causes of the bullwhip effect. From this case study, it was concluded that the RH-SD-Java approach with the WW and SM lot-sizing techniques, and with demand management and constant lead times, offers better results in total costs and service levels terms than the OUT-S manufacturing orders generated by a static planning horizon approach.

4.1.2. *Variable lead times*

Table 6 corresponds to the variable lead times scenario, *ltv*. Here, lead time variability represents the existence of different lead times for each simulated time period. This is generated through a random uniform function between a minimum of two and a maximum of four unit times.

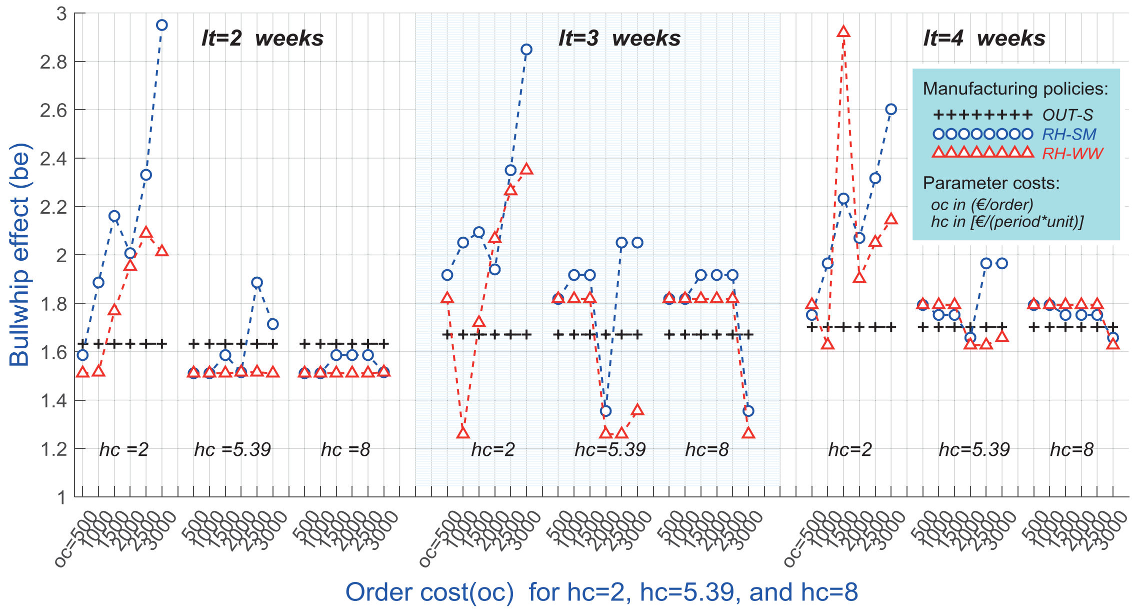

Figure 7. Bullwhip effect of manufacturing orders OUT-S, RH-SM, RH-WW.

Regarding the results corresponding to the scenarios with variable lead times (*ltv*), Figures 8, 9 and 10 reveal that the RH-WW and RH-SM policies are not always better than the OUT-S policy, which occurred in the scenarios with constant supply times. These policies might prove slightly better than others as far as the values for any previously defined parameter are concerned; i.e. order costs (*oc*) and/or holding costs (*hc*). For instance, with $hc = 2$, and after taking into account the values of the total costs and fill rates, the RH-WW and RH-SM policies proved better than OUT-S for an order with a cost over €1500 (Note A, Note D, respectively, in Figure 8 and Figure 9). However, when contemplating the bullwhip effect value (Figure 10) for the considered case of $hc = 2$, the best value of both policies was obtained with $oc = 2000$ (Note G in Figure 10). Similarly, to $hc = 5.39$, the RH-SM policy better performed than the other two in total costs and service level terms for an order cost (*oc*) value between 1000 and 1500 (Note B and Note E in Figures 8 and 9). Regarding the bullwhip effect value, RH-WW outperformed RH-SM for an order cost value between 1500 and 2000 (Note H in Figure 10). It is worth highlighting that for all cases, the bullwhip effect was lower when the RH-SD-Java approach was used with lot-sizing techniques than the OUT-S policy in a static simulation context. Finally, for $hc = 8$, the choice of a policy with an optimum point was more difficult because the RH-SM policy proved better for all the parameters employed for order cost values between 1500 and 2500 (Note C, Note F, and Note I, respectively, in Figures 8, 9 and 10). Nonetheless, RH-SM and RH-WW gave results that came close to those of OUT-S for $oc = 3000$ (Note J in Figure 10). Therefore, in a manufacturing environment with variable lead times, the replenishment order OUT-S (determined without an RH) should be carefully selected or, otherwise, an RH approach along with the WW or SM technique would be a better choice depending on the parameter we are interested in improving; i.e. total costs, service level or demand distortion. It is necessary to bear in mind that this choice would be sensitive to changes in order costs, *oc*, and holding costs, *hc*.

In order to provide a more exhaustive and general analysis, we experimented with 28 new numerical instances by considering variable lead times, several order and holding costs, and by using the same demand patterns provided in Table 4, but with different random number generators. Thus, the demand data for each time period were multiplied by the same random number between $\pm\alpha$, which represents a relative variation coefficient of the initial demand value provided in Table 4. This α value was between 10% and 50%. An easy-to-use interface for potential practitioners was also created to manage the RH-SD-Java model, which allows the number of different parameters to be selected to set the related holding costs, order costs and lead times (Figure 11).

From the generated output data (https://bit.ly/2Tp6nwv), we conclude that the RH-SD-Java approach, using SM and WW lot-sizing techniques, generally improves the total costs and fill rates for variable lead times compared to the OUT-S policy in a non-RH environment. Thus, from most of the simulations made, the SM technique behaved better than the WW and OUT-S procedures for lower holding costs and higher order costs, and behaved even better when order costs increased. This was due to the major distortion of the generated orders in relation to demands, which better absorbed the stockout periods generated mainly by the instability in the SC introduced by the variable lead times.

Table 6. Total costs, fill rate, *MFR*, and bullwhip effect, *be*, for $ltv = 2\text{–}4$ weeks.

		$hc = 2$			$hc = 5.39$			$hc = 8$		
Set up parameters	Manufacturing order type	Total cost (€)	*MFR* (%)	*be*	Total cost (€)	*MFR*(%)	*be*	Total cost (€)	*MFR*(%)	*be*
$oc = 500$	*OUT S*	3833.000	63.93	2.648	3892.000	63.93	2.648	3937.000	63.93	2.648
	RH-SM	4407.000	63.78	2.625	4426.000	63.15	2.48	4293.000	63.15	2.48
	RH-WW	4186.000	63.15	2.48	4246.000	63.15	2.48	4293.000	63.15	2.48
$oc = 1000$	*OUT S*	3839.000	63.93	2.648	3898.000	63.93	2.648	3943.000	63.93	2.648
	RH-SM	4412.000	63.78	2.625	3550.000	67.1	2.342	4299.000	63.15	2.48
	RH-WW	3921.000	63.1	2.461	4252.000	63.15	2.48	4299.000	63.15	2.48
$oc = 1500$	*OUT S*	3845.000	63.93	2.648	3904.000	63.93	2.648	3949.000	63.93	2.648
	RH-SM	4425.000	63.91	2.71	3556.000	67.1	2.342	3613.000	67.1	2.342
	RH-WW	3533.000	64.01	2.235	4258.000	63.15	2.48	4305.000	63.15	2.48
$oc = 2000$	*OUT S*	3851.000	63.93	2.648	3910.000	63.93	2.648	3955.000	63.93	2.648
	RH-SM	2664.000	75.93	2.321	3987.000	63.1	2.461	3619.000	67.1	2.342
	RH-WW	2789.000	74.61	2.182	3987.000	63.1	2.461	4311.000	63.15	2.48
$oc = 2500$	*OUT S*	3857.000	63.93	2.648	3916.000	63.93	2.648	3961.000	63.93	2.648
	RH-SM	2672.000	77.55	2.675	4493.000	63.78	2.625	3625.000	67.1	2.342
	RH-WW	1908.000	76.61	2.452	3993.000	63.1	2.461	4317.000	63.15	2.48
$oc = 3000$	*OUT S*	3863.000	63.93	2.648	3922.000	63.93	2.648	3967.000	63.93	2.648
	RH-SM	2285.000	77.79	3.133	4499.000	63.78	2.625	4039.000	63.1	2.461
	RH-WW	1815.000	77.98	2.502	3999.000	63.1	2.461	4039.000	63.1	2.461

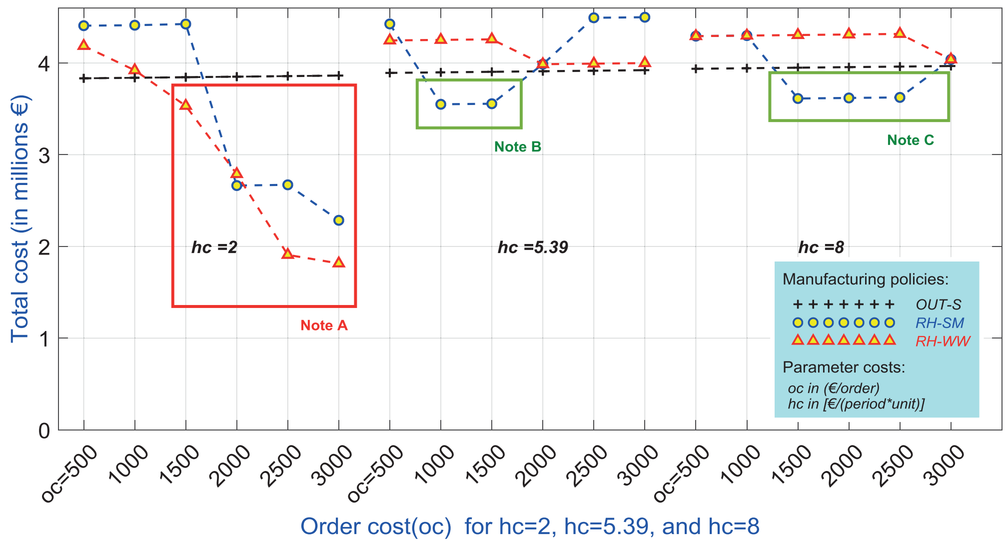

Figure 8. Total cost of manufacturing orders OUT-S, RH-SM, RH-WW in a scenario with *ltv* (2–4 weeks).

Figure 9. Fill rate for manufacturing orders OUT-S, RH-SM, RH-WW in a scenario with *ltv* (2–4 weeks).

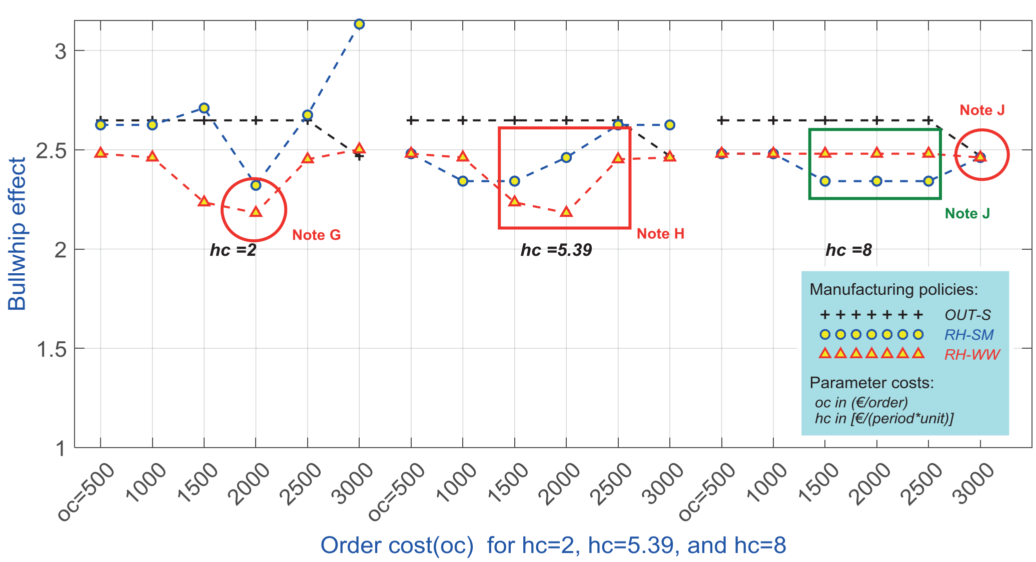

Figure 10. Bullwhip effect of manufacturing orders OUT-S, RH-SM, RH-WW in a scenario with *ltv* (2–4 weeks).

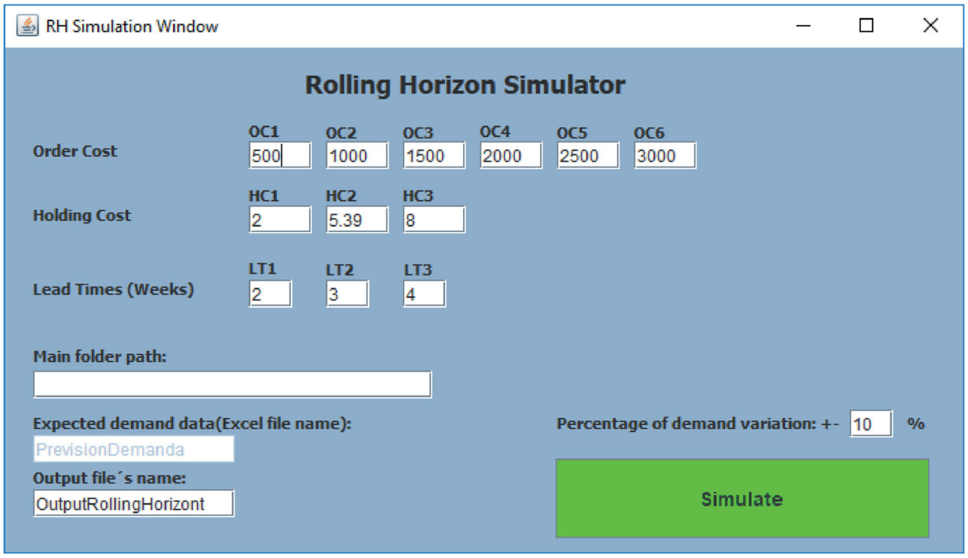

Figure 11. The RH-SD-Java simulator.

Moreover, with higher holding costs, the optimisation WW technique performed slightly better in total costs and fill rates terms than the SM and OUT-S techniques. It is highlighted that when lower total costs are generated, higher fill rates are also provided due mainly to the produced minimum stockout costs. Regarding the bullwhip effect in order distortions terms in relation to demand, as expected, it increased when using lot-sizing techniques, but mainly with lower holding costs, while order distortion can lower or remain unchanged compared to the OUT-S policy with higher holding and order costs. Here, WW performed better than SM because the generated lots were similar to the demand requirements. Additionally, the RH-SD-Java performed better than the OUT-S policy for a higher relative variation coefficient of the initial demand value $\pm\alpha$. From previous conclusions, we would recommend to the automotive SC under study proceeding as follows for a prescriptive way of making the order technique choice *a priori*:

- With variable lead times, low holding costs and high order costs, use the SM technique.
- With variable lead times and high holding costs, use the WW technique
- With variable lead times and minimum order distortions in relation to demands are required, use the WW technique

Therefore, we can positively answer the research question set out in this proposal about the validity and usefulness of the integrated use of RH and SD simulation approaches to manage demand under both uncertainty and lead time variability by evaluating different lot-sizing techniques. We also recommend planning practitioners to model and simulate their respective production planning systems by recreating different scenarios of holding and order costs, lead times and demand variabilities in a RH-SD-Java environment.

5. Conclusions

The paper proposes an RH approach to deal with management problems in a streamlined SC. This article presents a developed RH approach combined with SD in a Java environment, namely RH-SD-Java, to model and improve demand management processes in an environment with variable lead times and uncertain demand. It identifies replenishments with constant or variable lead times, and the modelled approach provides results that can help demand planners in their decision making. Nonetheless, to date all studies into RH have been applied mainly to different optimisation techniques based on mathematical programming. Hence the present study is pioneering insomuch as it is applied in SD-based simulation. To develop this new RH-SD-Java approach proposal, a Java programming environment was used along with Vensim® to calculate manufacturing orders using the optimum WW and SM lot-sizing techniques by an RH approach. This RH-SD-Java approach is proposed to face the dynamic nature that some SC demands can present, where the management of such demand must be solved immediately and in an unstable planning horizon lacking reliability that is not risk-free.

The main contribution of this paper aims to propose an integrated approach for production managers to provide new insight by combining RH planning for demand management with variable lead times, along with an SD simulation model. It also evaluates two different lot-sizing techniques, i.e. WW (optimisation) and SM (heuristic), integrated into an RH approach and compares them to an order up to inventory management policy (OUT-S). For a specific, but dynamic demand stream, this work evaluates costs, service levels and planning nervousness (bullwhip effect). A sensitivity analysis by varying lead times, holding costs and order costs is carried out. More numerical experiments with the same demand parameters, but with different random number generators, are analysed to acknowledge a more exhaustive analysis. Generally speaking, this work identifies how the application of an optimum lot-sizing technique in an RH setting with variable lead times can generate more sustainable planning results in total costs, fill rates and demand distortion (bullwhip effect) terms.

Managerial implications are referred to adopt an existent dynamic production plan (input/output data) and the replenishment order policy used by the company under study, and to compare them with other desired lot-sizing techniques (WW, SM, among others) by our RH-SD-Java approach. In the first place, technical requirements are related to an SD simulation modelling language (Vensim® or similar), spreadsheet and a Java environment. Some adaptations in the RH-SD-Java approach can be required as regards SC characteristics, i.e. inventory level variables, order and lead time auxiliary variables, among others. Moreover, end users will be provided with an easy-to-use interface, just as it was developed herein, to manage the RH-SD-Java model. In the context studied to manage demand with both uncertainty and lead time variability, a prescriptive way of making the order technique choice *a priori* was provided by comparing the WM and SM lot-sizing techniques and simulating several scenarios using different holding and order costs. From the managerial perspective, this integrated RH and SD simulation proposal could deduce more practical polices according to manager requirements and, subsequently, simulated scenarios (lot-sizing techniques, costs, among others). Lot size could also be constrained by the limited storage capacity limits in production processes, which should be especially considered in collaborative supply chain structures such as a vendor-managed inventory (VMI), where minimum and maximum inventories are usually predefined by a contract. In this case, the RH-SD-Java model may also include constraints to avoid an over-storage capacity in terms of reducing the selected order in accordance with updated on-hand inventory levels.

Finally, it is worth stressing that our forthcoming work intends to address the RH-SD-Java multilevel, multiproduct and supplier capacity constrained SC to test alternative lot-sizing techniques (least total cost, least unit cost, among others) and their application to different industrial or agri-food SC settings (Taylor and Fearne 2006). In this way, they will be able to better face the management of adverse demands which, for instance, this second sector presents suddenly or in the short or midterms, and result from external agents, like climate, natural disasters (Brown and Kshirsagar 2015; Ostberg et al. 2018), oil prices and international situations (Anderson and Nelgen 2012; Zhang and Qu 2015), among others, especially those that are uncontrolled.

Acknowledgement

This work was supported by the European Commission Horizon 2020 project Diverfarming [grant agreement 728003].

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the European Commission Horizon 2020 project Diverfarming [grant number 728003].

ORCID

Francisco Campuzano-Bolarín http://orcid.org/0000-0003-0527-4716
Manuel Díaz-Madroñero http://orcid.org/0000-0003-1693-2876

References

Agaran, B., W. W. Buchanan, and M. K. Yurtseven. 2007. "Regulating Bullwhip Effect in Supply Chains through Modern Control Theory." in *PICMET '07 – 2007 Portland International Conference on Management of Engineering & Technology*, 2391–2398. IEEE. http://doi.org/10.1109/PICMET.2007.4349573.

Anderson, K., and S. Nelgen. 2012. "Agricultural Trade Distortions during the Global Financial Crisis." *Oxford Review of Economic Policy* 28 (2): 235–260. doi:10.1093/oxrep/grs001.

Baker, K. R. 1977. "An Experimental Study of the Effectiveness of Rolling Schedules in Production Planning." *Decision Sciences* 8: 19–27. doi:10.1111/j.1540-5915.1977.tb01065.x.

Bhattacharya, R., and S. Bandyopadhyay. 2011. "A Review of the Causes of Bullwhip Effect in a Supply Chain." *The International Journal of Advanced Manufacturing Technology* 54 (9-12): 1245–1261. doi:10.1007/s00170-010-2987-6.

Boulaksil, Y., J. C. Fransoo, and E. N. G. van Halm. 2009. “Setting Safety Stocks in Multi-Stage Inventory Systems under Rolling Horizon Mathematical Programming Models.” *OR Spectrum* 31 (1): 121–140. doi:10.1007/s00291-007-0086-3.

Brown, M. E., and V. Kshirsagar. 2015. “Weather and International Price Shocks on Food Prices in the Developing World.” *Global Environmental Change* 35: 31–40. doi:10.1016/j.gloenvcha.2015.08.003.

Campuzano, F., J. Mula, and D. Peidro. 2010. “Fuzzy Estimations and System Dynamics for Improving Supply Chains.” *Fuzzy Sets and Systems* 161 (11): 1530–1542. doi:10.1016/J.FSS.2009.12.002.

Campuzano-Bolarín, F., J. Mula, and D. Peidro. 2013. “An Extension to Fuzzy Estimations and System Dynamics for Improving Supply Chains.” *International Journal of Production Research* 51 (10): 3156–3166. doi:10.1080/00207543.2012.760854.

de Sampaio, R. J. B., R. R. G. Wollmann, and P. F. G. Vieira. 2017. “A Flexible Production Planning for Rolling-Horizons.” *International Journal of Production Economics* 190: 31–36. doi:10.1016/j.ijpe.2017.01.003.

Díaz-Madroñero, M., J. Mula, and M. Jiménez. 2014. “Fuzzy Goal Programming for Material Requirements Planning under Uncertainty and Integrity Conditions.” *International Journal of Production Research* 52 (23): 6971–6988. doi:10.1080/00207543.2014.920115.

Díaz-Madroñero, M., J. Mula, and D. Peidro. 2017. “A Mathematical Programming Model for Integrating Production and Procurement Transport Decisions.” *Applied Mathematical Modelling* 52: 527–543. doi:10.1016/J.APM.2017.08.009.

Disney, S. M., M. M. Naim, and A. Potter. 2004. ï£¡ï£¡ï£¡Assessing the Impact of e-Business on Supply Chain Dynamics.” *International Journal of Production Economics* 89 (2): 109–118. doi:10.1016/S0925-5273(02)00464-4.

Dominguez, R., S. Cannella, and J. M. Framinan. 2015. “The Impact of the Supply Chain Structure on Bullwhip Effect.” *Applied Mathematical Modelling* 39 (23-24): 7309–7325. doi:10.1016/J.APM.2015.03.012.

Esteso, A., J. Mula, F. Campuzano-Bolarín, M. M. E. Alemany Diaz, and A. Ortiz. 2018. “Simulation to Reallocate Supply to Committed Orders under Shortage.” *International Journal of Production Research*, 1–19. doi:10.1080/00207543.2018.1493239.

Forrester, J. 1961. *Industrial Dynamics*. Waltham, MA: Pegasus Communications.

Fransoo, J. C., and M. J. F. Wouters. 2000. “Measuring the Bullwhip Effect in the Supply Chain.” *Supply Chain Management*. doi:10.1108/13598540010319993.

Geary, S., S. M. Disney, and D. R. Towill. 2006. “On Bullwhip in Supply Chains—Historical Review, Present Practice and Expected Future Impact.” *International Journal of Production Economics* 101 (1): 2–18.

Giard, V., and M. Sali. 2013. “The Bullwhip Effect in Supply Chains: A Study of Contingent and Incomplete Literature.” *International Journal of Production Research* 51 (13): 3880–3893. doi:10.1080/00207543.2012.754552.

Hosoda, T., and S. M. Disney. 2018. “A Unified Theory of the Dynamics of Closed-Loop Supply Chains.” *European Journal of Operational Research* 269 (1): 313–326. doi:10.1016/J.EJOR.2017.07.020.

Hussain, M., and P. R. Drake. 2011. “Analysis of the Bullwhip Effect with Order Batching in Multi-Echelon Supply Chains.” *International Journal of Physical Distribution & Logistics Management* 41 (10): 972–990. doi:10.1108/09600031111185248.

Jakšič, M., and B. Rusjan. 2008. “The Effect of Replenishment Policies on the Bullwhip Effect: A Transfer Function Approach.” *European Journal of Operational Research* 184 (3): 946–961. doi:10.1016/J.EJOR.2006.12.018.

John, S., M. M. M. Naim, and D. R. R. Towill. 1994. “Dynamic Analysis of a WIP Compensated Decision Support System.” *International Journal of Management Systems Design* 1 (4): 283–297. doi:10.2174/157488711796575577.

Karimi, B., S. M. T. Fatemi Ghomi, and J. M. Wilson. 2003. “The Capacitated Lot Sizing Problem: A Review of Models and Algorithms.” *Omega* 31 (5): 365–378. doi:10.1016/S0305-0483(03)00059-8.

Lee, H. L., V. Padmanabhan, and S. Whang. 1997a. “Information Distortion in a Supply Chain: The Bullwhip Effect.” *Management Science* 43 (4): 546–558. doi:10.1287/mnsc.43.4.546.

Lee, H. L., V. Padmanabhan, and S. Whang. 1997b. “The Bullwhip Effect in Supply Chains.” *Sloan Management Review* 38: 93–102. doi:10.1287/mnsc.43.4.546.

Li, J., A. Ghadge, and M. K. Tiwari. 2016. “Impact of Replenishment Strategies on Supply Chain Performance under e-Shopping Scenario.” *Computers & Industrial Engineering* 102: 78–87. doi:10.1016/J.CIE.2016.10.005.

Lian, Z., L. Liu, and S. X. Zhu. 2010. “Rolling-Horizon Replenishment: Policies and Performance Analysis.” *Naval Research Logistics (NRL)* 57 (6): 489–502. doi:10.1002/nav.20416.

Mendoza, J. D., J. Mula, and F. Campuzano-Bolarin. 2014. “Using Systems Dynamics to Evaluate the Tradeoff among Supply Chain Aggregate Production Planning Policies.” *International Journal of Operations & Production Management* 34 (8): 1055–1079. doi:10.1108/IJOPM-06-2012-0238.

Moreno, J. R., J. Mula, and F. Campuzano-Bolarin. 2015. “Increasing the Equity of a Flower Supply Chain by Improving Order Management and Supplier Selection.” *International Journal of Simulation Modelling* 14 (2): 201–214. doi:10.2507/IJSIMM14(2)2.284.

Mula, J., D. Peidro, and R. Poler. 2010. “The Effectiveness of a Fuzzy Mathematical Programming Approach for Supply Chain Production Planning with Fuzzy Demand.” *International Journal of Production Economics* 128 (1): 136–143. doi:10.1016/J.IJPE.2010.06.007.

Mula, J., R. Poler, and J. P. Garcia. 2006. “MRP with Flexible Constraints: A Fuzzy Mathematical Programming Approach.” *Fuzzy Sets and Systems* 157 (1): 74–97. doi:10.1016/J.FSS.2005.05.045.

Mula, J., R. Poler, and J. P. Garcia-Sabater. 2007. “Material Requirement Planning with Fuzzy Constraints and Fuzzy Coefficients.” *Fuzzy Sets and Systems* 158 (7): 783–793. doi:10.1016/J.FSS.2006.11.003.

Mula, J., R. Poler, and J. P. Garcia-Sabater. 2008. “Capacity and Material Requirement Planning Modelling by Comparing Deterministic and Fuzzy Models.” *International Journal of Production Research* 46 (20): 5589–5606. doi:10.1080/00207540701413912.

Ostberg, S., J. Schewe, K. Childers, and K. Frieler. 2018. “Changes in Crop Yields and Their Variability at Different Levels of Global Warming.” *Earth System Dynamics* 9 (2): 479–496. doi:10.5194/esd-9-479-2018.

Pacheco, E. O., S. Cannella, R. Lüders, and A. P. Barbosa-Povoa. 2017. "Order-Up-to-Level Policy Update Procedure for a Supply Chain Subject to Market Demand Uncertainty." *Computers & Industrial Engineering* 113: 347–355. doi:10.1016/J.CIE.2017.09.015.

Pujawan, I. N. 2004. "The Effect of Lot Sizing Rules on Order Variability." *European Journal of Operational Research* 159 (3): 617–635. doi:10.1016/S0377-2217(03)00419-3.

Rafiei, R., M. Nourelfath, J. Gaudreault, L. A. Santa-Eulalia, and M. Bouchard. 2014. "A Periodic Re-Planning Approach for Demand-Driven Wood Remanufacturing Industry: A Real-Scale Application." *International Journal of Production Research* 52 (14): 4198–4215. doi:10.1080/00207543.2013.869631.

Sahin, F., A. Narayanan, and E. P. Robinson. 2013. "Rolling Horizon Planning in Supply Chains: Review, Implications and Directions for Future Research." *International Journal of Production Research*. doi:10.1080/00207543.2013.775523.

Sahin, F., and E. P. Robinson. 2002. "Flow Coordination and Information Sharing in Supply Chains: Review, Implications, and Directions for Future Research." *Decision Sciences* 33 (4): 505–536. doi:10.1111/j.1540-5915.2002.tb01654.x.

Sahin, F., and E. P. Robinson. 2005. "Information Sharing and Coordination in Make-to-Order Supply Chains." *Journal of Operations Management* 23 (6): 579–598. doi:10.1109/CYBER.2015.7288126.

Schmidt, M., B. Münzberg, and P. Nyhuis. 2015. "Determining Lot Sizes in Production Areas – Exact Calculations versus Research Based Estimation." *Procedia CIRP* 28: 143–148. doi:10.1016/J.PROCIR.2015.04.024.

Silver, E. A., and H. Meal. 1973. "A Heuristic for Selecting Lot Size Quantities for the Case of a Deterministic Time-Varying Demand Rate and Discrete Opportunities for Replenishment." *Production and Inventory Management* 14: 64–74.

Silver, E. A., D. F. Pyke, and R. Peterson. 1998. *Inventory Management and Production Planning and Scheduling. Managerial and Decision Economics*. Vol. 11. John Wiley & Sons. https://doi.org/10.1002/mde.4090110504.

Simpson, N. C. 1999. "Multiple Level Production Planning in Rolling Horizon Assembly Environments." *European Journal of Operational Research* 114 (1): 15–28. doi:10.1016/S0377-2217(98)00005-8.

Sridharan, S. V., W. L. Berry, and V. Udayabhanu. 1988. "Measuring Master Production Schedule Stability under Rolling Planning Horizons." *Decision Sciences* 19 (1): 147–166.

Taylor, D. H., and A. Fearne. 2006. "Towards a Framework for Improvement in the Management of Demand in Agri-Food Supply Chains." *Supply Chain Management: An International Journal* 11 (5): 379–384. doi:10.1108/13598540610682381.

van den Heuvel, W., and A. P. M. Wagelmans. 2005. "A Comparison of Methods for Lot-Sizing in a Rolling Horizon Environment." *Operations Research Letters* 33 (5): 486–496. doi:10.1016/J.ORL.2004.10.001.

Vargas, V., and R. Metters. 2011. "A Master Production Scheduling Procedure for Stochastic Demand and Rolling Planning Horizons." *International Journal of Production Economics* 132 (2): 296–302. doi:10.1016/J.IJPE.2011.04.025.

Wagner, H. M., and T. M. Whitin. 1958. "Dynamic Version of the Economic Lot Size Model." *Management Science* 5 (1): 89–96. doi:10.1287/mnsc.5.1.89.

Wemmerlov, U., and D. C. Whybark. 1984. "Lot-Sizing under Uncertainty in a Rolling Schedule Environment." *International Journal of Production Research* 22 (3): 467–484. doi:10.1080/00207548408942467.

Whybark, D. C., and J. G. Williams. 1976. "Material Requirements Planning under Uncertainty." *Decision Sciences* 7 (4): 595–606. doi:10.1111/j.1540-5915.1976.tb00704.x.

Zhang, C., and X. Qu. 2015. "The Effect of Global Oil Price Shocks on China's Agricultural Commodities." *Energy Economics* 51: 354–364. doi:10.1016/J.ENECO.2015.07.012.

Annexe

Note: The nomenclature used in the following formulations belongs to the original authors.

OUT-S policy

The order-up-to level policy, OUT-S, is as follows (Silver, Pyke, and Peterson 1998):

$$O_t = S_t – inventoryposition.$$

The order quantity, O_t, equals S_t, reduced for the inventory position, where O_t is the ordering decision made at the end of time period t and S_t is the OUT-S used during time period t. OUT-S is updated each period according to:

$$S_t = \hat{D}_t^L + k\hat{\sigma}_t^L$$

where S_t equals the estimate mean of demand, $\hat{D}_t^L$, over L periods ($\hat{D}_t^L = \hat{D}_t \cdot L$) increased for the prescribed fill rate with buffer stocks. σ_t^L is an estimation of the standard deviation over L periods, and k is the fill rate factor (safety factor) that depends on the demand distribution.

$$inventory\ position = inventory\ on\ hand + orders\ placed\ but\ not\ yet\ received – backlogged\ orders.$$

Wagner-Whitin model

The mathematical model of Wagner-Whitin (WW) (1958) is based on dynamic programming, given by

$$F(t) = \min \left[\begin{array}{c} \min\limits_{1 \le j < t} \left[s_j + \sum\limits_{h=j}^{t-1} \sum\limits_{k=h+1}^{t} i_h d_k + F(j-1) \right] \\ s_t + F(t-1) \end{array} \right],$$

where, t is the current time period number; d_t = the amount demand during time period t; i_t = the interest charge per inventory unit carried forward to time period $t+1$; s_t = the ordering (or setup) cost during time period t; x_t = the amount ordered (or manufactured) during time period t; $F(t)$ = the minimal cost programme for periods 1 through to t.

Silver-Meal model

The heuristics technique of the SM (1977) model is based on dynamic programming, given by

$$\gamma_{i,j} = \frac{1}{j-i+1}\left[k + \sum_{k=1}^{j} h(k-i)d_k\right],$$

where, $\gamma_{i,j}$ is the cost per period if we buy from time period i through to j; h is the holding cost; d_k is the demand for time period k.

The Wagner-Whitin Java code

```
package wagnerwhitin;

// WagnerWithinCalculator is used to calculate the optimal lot-sizing using the
Wagner-Whitin algorithm that is based on dynamic programming

public class WagnerWhitinCalculator {

	private float [] originalDemands;		// demands
	private double orderCost;			// oc
	private double stockPerUnitPerPeriodCost;	// hc

	public WagnerWhitinCalculator(float [] demands, double orderCost, double
stockPerUnitPerPeriodCost) {
		this.originalDemands = demands;
		this.orderCost = orderCost;
		this.stockPerUnitPerPeriodCost = stockPerUnitPerPeriodCost;
	}

	public double getCost(int fromPeriod, int toPeriod) {
		double cost = orderCost;

		for (int period=fromPeriod + 1; period <= toPeriod; period++) {
			int step = period - fromPeriod;
			cost = cost + step * originalDemands[period] *
stockPerUnitPerPeriodCost;
		}
		return cost;
	}

	public float [] getOrders() {
		double [] costTable = new double[originalDemands.length + 1];
		int [] fromTable = new int[originalDemands.length + 1];
		costTable[0] = 0.0;

		for (int numberOfPeriods=1; numberOfPeriods < costTable.length;
numberOfPeriods++) {
			costTable[numberOfPeriods] = Double.POSITIVE_INFINITY;

			for (int fromNumberOfPeriods=0; fromNumberOfPeriods <
numberOfPeriods; fromNumberOfPeriods++) {

				double cost = costTable[fromNumberOfPeriods] +
getCost(fromNumberOfPeriods, numberOfPeriods - 1);
				if (cost <= costTable[numberOfPeriods])
					costTable[numberOfPeriods] = cost;
					fromTable[numberOfPeriods] =
fromNumberOfPeriods;
				}
			}
		}
		float [] orders = new float [originalDemands.length];
		int index = originalDemands.length;
		int ordersIndex = originalDemands.length - 1;

		while (index != 0) {
			int currentNumberOfPeriods = index - fromTable[index];
			int accumulatedOrders = 0;
			for (int demandsIndex=ordersIndex; demandsIndex >
ordersIndex - currentNumberOfPeriods; demandsIndex--)
				accumulatedOrders += originalDemands[demandsIndex];
			ordersIndex -= currentNumberOfPeriods;
			orders[ordersIndex + 1] = accumulatedOrders;
			index = fromTable[index];
		}
		return orders;	// optimum orders
	}

}
```

The Silver-Meal Java code

```
package silvermeal;

// SilvermealCalculator is used to calculate the lot-sizing using the heuristic
technique of Silver-Meal algorithm, that is based on dynamic programming

public class SilverMealCalculator {
	private Integer [] originalDemands;		// demands
	private double orderCost;			// oc
	private double stockPerUnitPerTimeCost;	// hc

	public SilverMealCalculator(Integer [] demands, double orderCost, double
stockPerUnitPerTimeCost) {
		this.originalDemands = demands;
		this.orderCost = orderCost;
		this.stockPerUnitPerTimeCost = stockPerUnitPerTimeCost;
	}

	private double calculateMeanCost(ArrayList<Integer> demands, int
numberOfPeriods) {
		double result = orderCost;
		for (int step = 1; step < numberOfPeriods; step++) {
			result = result + step * demands.get(step) *
this.stockPerUnitPerTimeCost;
		}
		result = result / numberOfPeriods;
		return result;
	}

	private int getNumberOfPeriods(ArrayList<Integer> demands) {
		double lastCost = Double.POSITIVE_INFINITY;
		double currentCost = 0;
		int result = 1;
		for (int period = 1; period <= demands.size(); period++)
		{
			currentCost = calculateMeanCost(demands, period);
			if (currentCost > lastCost)
				break;
			result = period;
			lastCost = currentCost;

		}
		return result;
	}

	public int [] getOrders() {
		ArrayList<Integer> demands = new
ArrayList<Integer>(Arrays.asList(originalDemands));
		int currentNumberOfPeriods;
		int [] orders = new int[originalDemands.length];
		int orderIndex = 0;
		while (demands.size() > 0) {
			currentNumberOfPeriods = getNumberOfPeriods(demands);
			int accumulatedDemand = 0;
			for (int index = 0 ; index < currentNumberOfPeriods; index++)
{
				accumulatedDemand = accumulatedDemand +
demands.get(0);
				demands.remove(0);
			}
			orders[orderIndex] = accumulatedDemand;
			orderIndex = orderIndex + currentNumberOfPeriods;

		}
		return orders;
	}

}
```