Author(s): Kania, Adhe; Sipilä, Juha; Misitano, Giovanni; Miettinen, Kaisa; Lehtmäki, Jussi

Title: Integration of lot sizing and safety strategy placement using interactive multiobjective optimization

Year: 2022

Version: Published version

Computers & Industrial Engineering 173 (2022) 108731

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Integration of lot sizing and safety strategy placement using interactive multiobjective optimization

Adhe Kania [a,c,*], Juha Sipilä [b], Giovanni Misitano [a], Kaisa Miettinen [a], Jussi Lehtimäki [b]

[a] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
[b] JAMK University of Applied Sciences, School of Technology, Jyväskylä, Finland
[c] Institut Teknologi Bandung, Faculty of Mathematics and Natural Sciences, Jl Ganesha 10, Bandung 40132, Indonesia

ARTICLE INFO

Keywords:
Inventory management
Safety stock
Uncertain demand
Multiple objective optimization
Interactive method
E-NAUTILUS

ABSTRACT

We address challenges of unpredicted demand and propose a multiobjective optimization model to integrate a lot sizing problem with safety strategy placement and optimize conflicting objectives simultaneously. The novel model is devoted to a single-item multi-period problem in periodic review policy. As a safety strategy, we use the traditional safety stock concept and a novel concept of safety order time, which uses a time period to determine the additional stock to handle demand uncertainty. The proposed model has four objective functions: purchasing and ordering cost, holding cost, cycle service level and inventory turnover. We bridge the gap between theory and a real industrial problem and solve the formulated problem by using an interactive trade-off-free multiobjective optimization method called E-NAUTILUS. It is well suited for computationally expensive problems. We also propose a novel user interface for the method. As a proof of concept for the model and the method, we use real data from a manufacturing company with the manager as the decision maker. We consider two types of items and demonstrate how a decision maker can find a most preferred solution with the best balance among the conflicting objectives and gain valuable insight.

1. Introduction

To achieve a competitive advantage, many companies strive to reduce their inventory values. Their main goal is to store a proper quantity of items in order to satisfy demand but concurrently avoid shortages and excess inventory. This problem, known as a lot sizing problem, has been considered in the literature for decades using economic order quantity (EOQ) (Harris, 1913; Wagner & Whitin, 1958). Recently, researchers have shown an increased interest in this area by considering more complex situations, see e.g. Andriolo, Battini, Grubbström, Persona, and Sgarbossa (2014), Bahl, Ritzman, and Gupta (1987), Glock, Grosse, and Ries (2014).

A lot sizing problem becomes more challenging when uncertainty is considered in the model. The uncertainty mostly comes from demand which can be affected by many conditions, such as weather, economy and market competition (Zipkin, 2000), as well as supplier reliability. A safety stock (SS) has been widely used to protect against demand uncertainty (Graves, 1988; Guide & Srivastava, 2000; New, 1975). A SS is described as a level of item, which is usually called a stock keeping unit (SKU), that is kept in inventory in order to manage the unpredicted demand. A SKU is defined as an individually identifiable item stored in inventory (Sawaya & Giauque, 1986). The problem of determining the amount of a SS to hold is called safety stock placement. Even though lot sizing and safety stock placement have been investigated in many research studies, they are typically managed separately. A SS is usually calculated by defining a desired service level and the lot sizing problem is then solved using some optimization methods (Zipkin, 2000). The integration of a lot sizing problem and safety stock placement was proposed in Kumar and Aouam (2018). The authors formulated a single objective optimization model to minimize system-wide production and inventory costs with a service level requirement constraint, and proposed an extension of an existing safety stock replacement algorithm to solve it.

SS plays an important role in industrial management and has been used for half a century to handle demand uncertainty (New, 1975). However, as a static method, SS is not suitable when demand fluctuates a lot (Açıkgöz, Çağıl, & Uyaroğlu, 2020). Some researchers use dynamic SS that can be dynamically changed from period to period (Inderfurth & Vogelgesang, 2013; Rafiei, Nourelfath, Gaudreault, De Santa-Eulalia, & Bouchard, 2015). However, when a lot sizing problem has large sizes of decision variables and various types of practical production constraints, it is difficult to solve the problem by using a dynamic SS (Tavaghof-Gigloo & Minner, 2021).

* Correspondence to: University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland.
E-mail address: adhe.a.kania@student.jyu.fi (A. Kania).

https://doi.org/10.1016/j.cie.2022.108731
Received 11 May 2021; Received in revised form 9 September 2022; Accepted 2 October 2022
Available online 5 October 2022
0360-8352/

The basic problem in lot sizing is to determine an order quantity to minimize costs but satisfy demand and prevent shortages, which are naturally conflicting with each other. Therefore, multiobjective optimization (Miettinen, 1999) is needed to solve this problem. Multiobjective optimization has been studied to solve different topics in lot sizing problems (Aslam & Amos, 2010), such as supplier selection (Rezaei & Davoodi, 2011; Ustun & Demirtas, 2008), perishability issues (Amorim, Antunes, & Almada-Lobo, 2011) and sustainability issues (Azadnia, Saman, & Wong, 2015). Integrating a lot sizing problem with safety stock placement gives additional conflicting objectives, because keeping a high amount of safety stock introduces a trade-off between costs and service level (Chan & Chan, 2006). By using multiobjective optimization, a decision maker can clearly see the trade-offs between objectives before he/she selects the final solution that best represents his/her preferences.

Multiobjective optimization problems usually have many solutions, called Pareto optimal solutions, which reflect trade-offs among the conflicting objectives. Pareto optimal solutions are incomparable from a mathematical point of view, and the final solution is the one that best represents a decision maker's preferences, who is an expert in the problem domain. Interactive methods (Miettinen, Hakanen, & Podkopaev, 2016), which iteratively incorporate the decision maker's preferences, are viable methods to find a solution that satisfies the decision maker's preferences. In interactive methods, the decision maker can learn about the trade-offs and adapt one's preferences while learning. This increases confidence and satisfaction with the final solution. So far, however, there have only been few articles proposing or applying interactive methods to solve their lot sizing problems (Agrell, 1995; Bouchery, Ghaffari, Jemai, & Dallery, 2012; Heikkinen, Sipilä, Ojalehto, & Miettinen, 2021; Ustun & Demirtas, 2008), and none of them were designed for computationally expensive problems.

This paper is an instance of data-driven decision support, where multiobjective optimization is applied. Starting with real data, we propose a multiobjective optimization model inspired by real challenges on a lot sizing problem in a manufacturing company. To bridge the gap between theory and practice, we verify the model with the supply chain manager of the said company to ensure the model is applicable.

We consider a single-item lot sizing problem in multiple time periods. By considering stochastic demands, we propose an additional way to handle the uncertainty of demand, which is called a safety order time (SOT), in addition to the SS. The idea of SOT is to keep additional stock based on time. For example, by setting SOT as one week, additional SKUs to cover one week's worth of demand are always kept in the storage and can be used to accommodate demand uncertainty. The proposed SOT fills the need of having dynamic stock to handle unpredicted demand efficiently. We combine SS and SOT in the model in order to manage the stochasticity of demand. The problem of determining the amount of SS and SOT is defined as a safety strategy placement. Integrating a lot sizing problem and a safety strategy placement to decide the optimal order quantity of SKUs for each period, as well as the best combination of the SS and SOT, are our aims in this research. Therefore, we propose a novel model that integrates a lot sizing problem not only with a SS placement but also with a SOT placement.

Compared to other relevant studies on lot sizing, contributions of this paper are summarized in Table 1. In this table, SOP stands for optimization problems with a single objective function and MOP for multiobjective optimization problems. The second row is not an exhaustive list but provides examples of studies. There are many multiobjective lot sizing studies which do not utilize interactive methods (Aslam & Amos, 2010). The table shows that this paper, for the first time, uses multiobjective optimization considering an integration of a lot sizing problem with both SS and SOT, and applies an interactive method to solve it.

To solve the defined lot sizing problem, we propose a multiobjective optimization model with four objective functions to characterize different perspectives of lot sizing decision. We adapt the cost objectives from the dynamic EOQ model (Wagner & Whitin, 1958) as the first and the second objectives. However, we separate the purchasing and ordering cost in the first objective and the holding cost in the second objective, because they show different behavior of inventory system (Rashid, Bozorgi-Amiri, & Seyedhoseini, 2015). The holding cost has a positive gradient and the other costs have negative gradients when the order quantity is increased. Thus, we enable studying this trade-off. Furthermore, we consider cycle service level as the third objective to measure the capability of the proposed safety strategy to deal with the stochasticity of demand. And lastly, we have the inventory turnover in the fourth objective as the primary performance measurement in inventory management (Silver, Pyke, & Thomas, 2017) to measure the effectiveness of this model in managing inventory. These four objectives can maximize the effectiveness of inventory with minimal costs and sufficient safety strategy to maximally handle demand uncertainty.

We apply the trade-off-free interactive method E-NAUTILUS (Ruiz, Sindhya, Miettinen, Ruiz, & Luque, 2015), for the first time in this field, to solve the proposed problem. The strength of this method is that it starts from the worst possible objective function values and iteratively improves all objectives, allowing the decision maker to find his/her most preferred solution without having to trade-off among the objectives. Sometimes, decision makers tend to anchor around the starting point because of trading-off (Buchanan & Corner, 1997) and, thus, fail to find preferred solutions. Thanks to the structure of the method, this is avoided. Lot sizing problems have been identified as computationally challenging problems in many articles (Alem, Curcio, Amorim, & Almada-Lobo, 2018; Bitran & Yanasse, 1982), and demand uncertainty increases the complexity of the problem (Efthymiou, Mourtzis, Pagoropoulos, Papakostas, & Chryssolouris, 2016). The E-NAUTILUS method is designed for solving computationally expensive problems, which makes it an adequate choice to solve the lot sizing problem defined in this research. Furthermore, we develop a novel web-based user interface for E-NAUTILUS, which can be freely accessed and is made available as open-source software.

As said, as a proof of concept, we consider a real case study and the supply chain manager who acted as the decision maker found the model and the results useful. We demonstrate that the E-NAUTILUS method can be successfully applied to solve our integrated computationally expensive lot sizing problem for the real case study of two SKUs. From the managerial perspective, the parallel exploitation of SS and SOT is a welcomed addition to traditional inventory management models. The decision maker appreciated the benefit of SOT to manage additional stocks dynamically in an efficient way. He was satisfied with the results and willing to adopt the model more widely for inventory planning and control, especially for critical SKUs.

To sum up, the main contributions of this paper can be written as follows:

(1) Proposing a novel concept of safety order time (SOT) to handle demand uncertainty.
(2) Introducing a multiobjective optimization model which integrates a lot sizing problem and the safety strategy placement.
(3) Applying an interactive trade-off-free method E-NAUTILUS that is appropriate for computationally expensive lot sizing problems.
(4) Developing a new web-based user interface for E-NAUTILUS (as a free and open-source software).
(5) Solving the problem successfully and finding a final solution that best represents the decision maker's preferences by using the E-NAUTILUS method.

The rest of the paper is organized as follows. Section 2 describes the main concepts of multiobjective optimization and the E-NAUTILUS method. Section 3 presents the assumptions, notations, objective functions and constraints of the proposed multiobjective optimization model, while details of the developed web-interface implementation are discussed in Section 4. In Section 5, a real case study with data from a manufacturing company is considered with results and analysis of the decision making process using the E-NAUTILUS method. Finally, we conclude our work and discuss future directions in the last section.

Table 1
Comparison with other relevant studies on lot sizing.

Source	SOP/MOP	SS	SOT	Interactive
Kumar and Aouam (2018), Tavaghof-Gigloo and Minner (2021)	SOP	Yes	No	No
Rezaei and Davoodi (2011), Amorim et al. (2011), Azadnia et al. (2015), survey Aslam and Amos (2010), and more	MOP	No	No	No
Agrell (1995), Ustun and Demirtas (2008), Bouchery et al. (2012), Heikkinen et al. (2021)	MOP	No	No	Yes
This paper	MOP	Yes	Yes	Yes

2. Background in multiobjective optimization

2.1. Basic concepts

We consider multiobjective optimization problems of the following form:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_k(\boldsymbol{x}))^T \\ \text{subject to} \quad & \boldsymbol{x} \in S, \end{aligned} \tag{1}$$

where $f_i : S \rightarrow \mathbb{R}$ for $1 \leq i \leq k$ and $k \geq 2$ are the objective functions which are to be optimized simultaneously. The vector of decision variables $\boldsymbol{x} = (x_1, \dots, x_n)^T$ is bounded by the feasible region S, which is a subset of the decision space $\mathbb{R}^n$. The feasible region is formed by constraints, which can be lower and upper bounds for $\boldsymbol{x}$ and/or equality and inequality constraints. The image of the feasible region $Z = \boldsymbol{f}(S)$ is called a feasible objective region, which is a subset of the objective space $\mathbb{R}^k$. A vector $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_k(\boldsymbol{x}))^T$, $\boldsymbol{z} \in Z$, which is called an objective vector, consists of objective values calculated at $\boldsymbol{x} \in S$.

Objective functions are usually conflicting with each other. Therefore, it is impossible to find one solution where each objective achieves its individual optimum. A multiobjective optimization problem (1) usually has several solutions which are called Pareto optimal solutions. For two objective vectors $\boldsymbol{z}^1, \boldsymbol{z}^2 \in Z$, $\boldsymbol{z}^1$ is said to dominate $\boldsymbol{z}^2$ if $z_i^1 \leq z_i^2$ for all $i = 1, \dots, k$ and $z_j^1 < z_j^2$ for at least one $j = 1, \dots, k$. Otherwise, $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ are nondominated. A decision vector $\boldsymbol{x}'$ and its corresponding objective vector $\boldsymbol{z}'$ are Pareto optimal if there does not exist another decision vector $\boldsymbol{x} \in S$ such that $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ dominates $\boldsymbol{z}'$. The set of Pareto optimal solutions in the decision space is called a Pareto optimal set, and its image in the objective space is known as a Pareto optimal front.

The ranges of the objective function values in the Pareto optimal front may provide useful information for the decision maker. Lower and upper bounds of the Pareto optimal front are represented in an ideal point $\boldsymbol{z}^*$ and a nadir point $\boldsymbol{z}^{nad}$, respectively. They represent the best and the worst values that can be achieved by each objective function in the Pareto optimal front. The ideal point can be calculated by minimizing each of the objective functions individually, while the nadir point is more difficult to obtain because it depends on the whole Pareto optimal front which is usually not fully known. There is no reliable procedure for calculating the nadir point with more than two objectives (Miettinen, 1999), but it can be approximated for example by using a payoff table (Benayoun, de Montgolfier, Tergny, & Laritchev, 1971).

Pareto optimal solutions are incomparable mathematically, thus we need some additional information from a decision maker to determine the most preferred solution as the final one. A decision maker is an expert who is responsible for making a strategic decision in the problem domain. In lot sizing, he/she is usually a supply chain manager in a manufacturing company. Besides the decision maker, solving a multiobjective optimization problem involves an analyst, who supports the decision maker in mathematical aspects. The analyst is assumed to know multiobjective optimization methods and is responsible for the mathematical model and making preparations before the decision maker is involved.

Based on the role of the decision maker during the solution process, methods to solve multiobjective optimization problems can be divided into four classes (Miettinen, 1999). The first class is no-preference methods. These methods do not use any preference from the decision maker. Then, in the second class, called a priori methods, preference information from the decision maker is first required and a Pareto optimal solution reflecting this information is then found. In contrast, several Pareto optimal solutions are first generated and presented to the decision maker in the third class, which is called a posteriori methods, and he/she then has to select the most preferred one. The last class is interactive methods, where the decision maker is actively involved to give his/her preferences iteratively.

Interactive methods are regarded as promising methods to get a final solution that best satisfies the decision maker (Miettinen & Hakanen, 2009; Miettinen, Ruiz, & Wierzbicki, 2008). In interactive methods, the decision maker does not need any global preference structure about the problem, but he/she is able to learn about the interrelationships among the objectives during the solution process. In each iteration, some information is presented and the decision maker is asked to express his/her preferences by answering some relevant questions. Then, the preferences are accounted for to improve the solutions in the following iteration. There are many ways to inquire preference information from the decision maker (Miettinen et al., 2016).

In this paper, we use the E-NAUTILUS method developed by Ruiz et al. (2015), where the decision maker iteratively approaches the Pareto optimal front and can avoid trading-off by improving in all objectives simultaneously. The reason for using this method is its ability of handling computationally expensive problems, which is appropriate for lot sizing problems, and the possibility to avoid anchoring and find the most preferred solution without trading-off.

2.2. E-NAUTILUS method

The E-NAUTILUS method (Ruiz et al., 2015) is a variant of NAUTILUS methods (Miettinen & Ruiz, 2016). These methods are motivated by the prospect theory (Kahneman & Tversky, 1979), saying that people do not react similarly to gains and losses, but they fear losses more than they desire gains. Based on this philosophy, instead of starting with some Pareto optimal solution as most other interactive methods do, NAUTILUS methods choose the worst objective function values as the starting point, that is, the nadir point. Thereafter, new candidates are generated where objective function values are improved iteratively, and the preferred Pareto optimal solution will be the final solution. In this way, the decision maker can have a free search without requiring any trade-offs, and he/she always experiences an improvement in all of the objective values at every iteration until the Pareto optimal front is reached.

An important concept in NAUTILUS methods is reachable values of objective functions referring to values of each objective function that still can be reached from the current candidate without sacrifices in other objectives. The decision maker is given information on the lower bounds of reachable values. Upper bounds of reachable values are given by the candidate. Naturally, the range of reachable values gets smaller during the iterations that is, when the candidates get closer to the Pareto optimal front. In E-NAUTILUS, several candidates are shown to the decision maker at each iteration. Each candidate represents different directions to move towards the Pareto optimal front. The decision maker selects the candidate, that is, the direction, one likes as preference information. Information of reachable values from each candidate can help the decision maker in order to not loose sight of the Pareto optimal front at any iteration during the solution process. In the E-NAUTILUS method, three kinds of information are provided to the decision maker: several candidates, lower bounds of corresponding reachable values, to be referred to as their best reachable values and closeness of the candidates to the Pareto optimal front.

The E-NAUTILUS method is particularly developed to handle computationally expensive problems. This method consists of three stages: pre-processing, interactive decision making and post-processing stages. Solving the original multiobjective optimization problem, which can be computationally expensive, is done without involvement of the decision maker in the pre-processing stage. In this stage, a set of Pareto optimal solutions P is generated using any a posteriori method. Therefore, an analyst who has knowledge on an appropriate (a posteriori type) method is needed here to generate a sufficient number of Pareto optimal solutions. In addition, to know the ranges of the Pareto optimal front, the nadir point and the ideal point are estimated based on P.

The second stage is the main part of the E-NAUTILUS method. This is the only part that needs the involvement of the decision maker. The candidates which are presented to the decision maker in each iteration, are calculated based on the data generated in the previous stage. The original computationally expensive problem is not solved in this stage, which reduces the waiting time of the decision maker in each iteration. This interactive stage can be described in the following steps:

(1) The ranges of the objective functions are shown to the decision maker by showing the estimated ideal point $\boldsymbol{z}^*$ and nadir point $\boldsymbol{z}^{nad}$.
(2) The decision maker is asked to provide the number of iterations N_I and the number of candidates N_S that he/she wants to see at each iteration.
(3) Set the starting point $\boldsymbol{z}(0) = \boldsymbol{z}^{nad}$, current iteration $h = 1$ and current set of Pareto optimal solutions $P(h) = P$.
(4) Select N_S solutions that well represent solutions in $P(h)$ by dividing $P(h)$ into N_S subsets and determine a representative solution of each, denoted by $\bar{\boldsymbol{z}}(h,i)$, $i = 1,\dots,N_S$.
(5) Calculate N_S candidates, denoted by $\boldsymbol{z}(h,i)$, $i = 1,\dots,N_S$, which lie on the line segment joining the previous preferred candidate $\boldsymbol{z}(h-1)$ and each representative solution $\bar{\boldsymbol{z}}(h,i)$ with the following formula:

$$\boldsymbol{z}(h,i) = \frac{it(h)-1}{it(h)}\boldsymbol{z}(h-1) + \frac{1}{it(h)}\bar{\boldsymbol{z}}(h,i), \tag{2}$$

where $it(h) = N_I - h + 1$ is the number of iterations left (including the current iteration).
(6) Calculate the best reachable values for each candidate as by solving the following ϵ-constraint problem (Haimes, Lasdon, & Wismer, 1971) for $r = 1,\dots,k$:

$$\begin{aligned} &\text{minimize} && f_r(\boldsymbol{x}) \\ &\text{subject to} && f_j(\boldsymbol{x}) \le z_j(h,i),\ j = 1,\dots,k, j \ne r \\ &&& \boldsymbol{x} \in P(h). \end{aligned} \tag{3}$$

(7) Calculate the closeness of each candidate to the Pareto optimal front, which is shown as a percentage, as follows:

$$d(h,i) = \frac{\|\boldsymbol{z}(h,i) - \boldsymbol{z}^{nad}\|}{\|\bar{\boldsymbol{z}}(h,i) - \boldsymbol{z}^{nad}\|} \times 100\%,\ i = 1,\dots,N_S. \tag{4}$$

(8) Show the N_S candidates together with their best reachable values and closeness information to the decision maker. Ask him/her to select his/her most preferred solution among the candidates as the current preferred candidate, denoted by $\boldsymbol{z}(h)$.
(9) Set $h = h + 1$, and update $P(h)$ by deleting the Pareto optimal solutions which cannot be reached without trade-offs from $\boldsymbol{z}(h)$.
(10) Repeat step 4–9 until $h = N_I + 1$.

From the interactive decision making stage, we have $\boldsymbol{z}(N_I)$ as the most preferred candidate selected by the decision maker. The Pareto optimality of this candidate depends on the a posteriori method used in the first stage. Some a posteriori methods, for example evolutionary methods, cannot theoretically prove the Pareto optimality of the solutions. Thus, to ensure the Pareto optimality of the final solution $\boldsymbol{z}^{final}$, the post-processing stage can be needed. In this stage, we project $\boldsymbol{z}(N_I)$ onto the Pareto optimal front by minimizing an achievement scalarizing function (Wierzbicki, 1980) with $\boldsymbol{z}(N_I)$ as a reference point. For further details of the methods, see Ruiz et al. (2015).

3. Multiobjective optimization model

As mentioned in the introduction, we consider a lot sizing problem with a safety strategy to handle uncertainty on demand. Traditionally, a SS is used to reserve a certain amount of stock to prepare for unpredicted surges of demand. By assuming a constant lead time, a SS only depends on the standard deviation of demand and the desired service level (Talluri, Cetin, & Gardner, 2004). For instance, high and low demand SKUs could have the same amount of SS, if they have the same demand deviation and service level. Therefore, in real life, supply chain managers need to think about a certain time period that can be covered with a SS. For example, they sometimes convert a SS into days by dividing it with the daily demand.

In this paper, we propose a SOT as an additional safety strategy, which keeps additional stock in the inventory based on time. When an order is placed, instead of considering demand along lead time as a typical way to solve a lot sizing problem, with this strategy, additional SOT days/weeks are also considered. For example, by setting a SOT as one week and having lead time as two weeks, demand for three weeks is considered for each period, but an order will arrive after two weeks. Therefore, the additional SKUs to cover demand for one following week are always kept in the inventory and can be used to accommodate demand uncertainty.

With SS, we keep the same amount of stock along the period considered, while demand can fluctuate a lot. This may increase the risk of running out of stock in case of high demand. On the other hand, SOT keeps stock based on demand in the following period, which can be higher for high demand and lower for low demand. Thus, instead of a constant amount of stock, SOT adapts to the demand of the following period and handles cases of high peak of demand better than SS. Because SS has an advantage in handling deviation of demand, the combination of SS and SOT increases the preparedness for demand uncertainty. For this reason, we use both SS and SOT in our proposed model.

SS and SOT are both usable indicators for inventory management when managing unpredictable fluctuation in demand. SS is a static method and, thus, reacts with a delay to changes in demand. Because of that, if demand increases, the SS coverage in days on hand decreases. This may result in stock out situations as the SS adequacy is less satisfactory. Thus, more certainty is required and, therefore, we propose SOT in our model. Unlike SS, SOT is more dynamic and, thus, serves the needs of management for stock planning purposes. This becomes clear in the context of our case study, as the decision maker states. The novelty value of SOT is essential because, as said, SOT is a dynamic factor and does not require as frequent updates as SS. Typically, a manufacturing company has a considerable number of SKUs to manage, and it is time-consuming to recalculate SKU stock control data, such as SS, continuously. SOT does not need to be updated that often and, thus, it supports management in an efficient way.

To solve the defined lot sizing problem, we formulate a multiobjective optimization problem with four objective functions and four constraints. The assumptions and notations which are used throughout the paper are defined before the multiobjective optimization formulation is introduced in this section.

3.1. Assumptions

We consider a single-item multi-period lot sizing problem with stochastic demand. We work in discrete time, so we review the lot size over m time periods $t = 1, \dots, m$ and the replenishment process follows a periodic review policy. The decision maker reviews the ordered quantity $Q(t)$ at the beginning of each period, and the order will arrive after a constant lead time L.

The idea of a SOT is shown in Fig. 1. For each order, we do not only consider the demand needed until the order arrives, but also an additional SOT time unit is considered. Hence, the order is actually needed after $L+SOT$ time units, but it comes earlier after L time units. With this strategy, we always have excess SKUs in the amount of the predicted demand during a SOT time unit, besides a SS. The excess can be used if unpredicted demand occurs.

We make the following assumptions.

(1) All of the data is ready to use (which means checking correctness and reliability of the data).
(2) Demand is normally distributed with a mean μ and a standard deviation σ. We define $D(t)$ as the total of predicted demand from the beginning of period t until the end of this period. Demands in different time periods are independent of each other.
(3) There is no capacity limit in ordering SKU, which means that the cost for one order is c, regardless of the quantity of SKUs in the order.
(4) There is no backorder cost.
(5) Every order can be placed with a minimum order quantity moq and it rounds up by a rounding value r. The multiplication of r is increased after moq. It means that the order can only be placed by following the formula $moq + a\,r$ for any integer $a \geq 0$.

3.2. Notation

The following notations are used in this paper.

Index

$\{t \mid t = 1, \dots, m\}$	index of time period

Data

p	price to purchase one SKU
c	cost to place one order
h	cost to hold one SKU for one period
T	length of one period
L	lead time
$D(t)$	predicted demand during period t
σ	standard deviation of demand for one period
μ	average demand
moq	minimum order quantity (for lot size)
r	rounding value (for lot size)

Decision variables

$Q(t)$	lot size at period t
SS	safety stock
SOT	safety order time

Dependent variables

$Y(t)$	order indicator, $Y(t) = 1$ if the order is placed ($Q(t) > 0$), otherwise $Y(t) = 0$
$I(t)$	inventory position at the end of period t (sum of inventory position at the end of the previous period and incoming order at period t decreased by the demand during period t), $I(t) = I(t-1) + Q(t - \lfloor L \rfloor) - D(t)$

Other Notations

$\lfloor u \rfloor$	the greatest integer less than or equal to u
$\lceil u \rceil$	the least integer greater than or equal to u

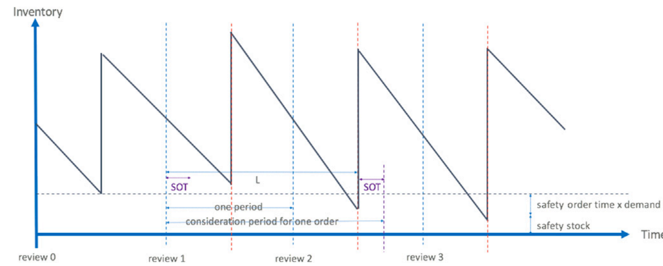

Fig. 1. Illustration of SOT in periodic review policy.

3.3. Objective functions

As mentioned, we have four objectives to consider simultaneously. Cost functions are as the first and the second objectives. According to the literature, in a lot-sizing problem, a purchasing manager must consider three types of cost (Chopra & Meindl, 2016): purchasing cost, ordering cost and holding cost. Most of the research considers total cost as one objective function. However, in this paper, we propose to separate it as two different cost functions. It is interesting to see holding cost individually, because it may show different behavior from the other costs (Rashid et al., 2015). Therefore, we minimize purchasing and ordering cost as the first objective and minimize holding cost as the second objective. Then, the adequacy of the safety strategy in handling unpredicted demand is measured in the third objective function. We maximize the cycle service level for this purpose. Lastly, maximizing inventory turnover, which is an important measurement in lot-sizing (Grant, Lambert, Stock, & Ellram, 2006), is considered in the last objective.

Purchasing cost is the expense of buying SKUs from a supplier. The price p is assumed to be fixed and no discount rate is applied. Ordering cost is the cost of placing one order, regardless of the number of SKUs in the order. It is fixed based on our assumption. In the first objective, we minimize the purchasing and ordering cost (POC) that can be written as follows:

$$POC = \sum_t Q(t)\, p + \sum_t Y(t)\, c. \tag{5}$$

A holding cost (HC) is the expense for holding SKUs, which can be calculated using several formulas (Alfares & Ghaithan, 2019). In this research, we calculate holding cost at one period by multiplying quantity of SKUs at this period and the cost for holding one SKU for one period h. For simplicity, the quantity of SKUs in one period is calculated as the average amount of inventory in this period. The formula of HC, which is treated as the second objective to be minimized, can be written as follows:

$$HC = \sum_t \frac{I(t-1) + I(t)}{2}\, h. \tag{6}$$

A cycle service level (CSL) is the probability of not having a stockout in a replenishment cycle (Chopra & Meindl, 2016). It measures how the safety strategy deals with the unpredicted demand during one replenishment cycle. One replenishment cycle is defined as one cycle that needs to be covered by one order, which is one period in our case. With the proposed safety strategy, we have a SS and demand for SOT time units to cover unpredicted demand in one period. Thus, we propose the CSL formula as follows to be maximized:

$$CSL = F\left(\frac{SS + \mu\, SOT}{\sigma}\right), \tag{7}$$

where F is the standard normal distribution function.

An inventory turnover (ITO) is a measurement for inventory performance that is quite important from a practical point of view. It means the number of times inventory turns over annually, which indicates how fast a company is selling the SKU or using it in the production. The ITO can be measured as the ratio between SKU usage and average inventory. We do not exactly know the demand in future periods, but we define the SKU usage as the addition of the predicted demand and the demand deviation which represents the SKU usage from the unpredicted demand. Hence, we propose to maximize the ITO formula as follows:

$$ITO = \sum_{t} \frac{D(t) + \sigma}{(I(t-1) + I(t))/2}. \tag{8}$$

3.4. Constraints

We propose four kinds of constraints to be considered in the multiobjective optimization model. To guarantee the availability of SKUs to cover the predicted demand, we set the fill rate as the first constraint. The second constraint aims to impose the order quantity policy, while the third constraint enforces the availability of safety inventory to cover the unpredicted demand. Finally, in the last constraint, we set the lower bounds of SS and SOT.

A fill rate (FR) is the fraction of demand which is satisfied from the inventory (Chopra & Meindl, 2016; Teunter, Syntetos, & Babai, 2017). This constraint is defined to ensure that the inventory in each period (excluding SS) can cover the predicted demand. As previously described, the consideration period for one order is $P = L + SOT$. Hence, a FR constraint for each period $t = 1, \dots, m$ can be written as:

$$FR(t) = \frac{I(t-1) + \sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{D_P} \geq 1, \tag{9}$$

where D_P is demand during P, which can be defined as:

$$D_P = \sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P - \lfloor P \rfloor) D(\lceil P \rceil). \tag{10}$$

Based on the order policy, an order can be placed with a certain minimum order quantity moq and multiplication of a rounding value r. It is common in practice and typically based on an agreement between a supplier and a company (Zhu, Liu, & Chen, 2015). Hence, for each period $t = 1, \dots, m$, the following constraint must be fulfilled:

$$Q(t) = Y(t)\,(moq + a\,r), \tag{11}$$

for any integer $a \geq 0$.

To ensure the availability of the safety strategy in the inventory, for each period $t = 1, \dots, m$, the following constraint must be fulfilled:

$$I(t) \geq SS + SOT\, D(t). \tag{12}$$

Finally, to eliminate negative values, lower bounds of SS and SOT must be defined as follows:

$$SS \geq 0 \text{ and } SOT \geq 0. \tag{13}$$

In conclusion, the proposed multiobjective optimization model can be written as:

$$\begin{aligned} &\text{minimize} && (POC, HC, -CSL, -ITO)^T \\ &\text{subject to} && (9), (11), (12), (13) \end{aligned} \tag{14}$$

4. Interactive E-NAUTILUS graphical user interface

As part of this paper, we developed a web-based graphical user interface (for short, interface) to ease the interaction between the decision maker and the interactive stage of E-NAUTILUS. The E-NAUTILUS interface was built on top of a computational *back-end* implementing the numerical steps of the interactive stage of E-NAUTILUS described in Section 2.2. The back-end was implemented as part of the latest iteration of the open-source DESDEO software framework (Ojalehto & Miettinen, 2019). Both the back-end and the interface were implemented using Python.

The E-NAUTILUS interface was developed for visualizing information related to a multiobjective optimization problem to a decision maker. We used the *Dash* platform (https://dash.plotly.com/) to build the interface. The reasons to use Dash were manifold:

(1) Dash is implemented in Python, which means that utilizing DESDEO in conjunction with Dash is seamless.
(2) Dash can be utilized with *plotly*, which is another Python library for building visualizations. Usage of plotly is desirable because it offers a wide variety of different interactive visualizations types.
(3) Applications using Dash can be used in any modern web-browser by having the application running either locally or on a remote web-server. This makes the application very accessible.
(4) Dash comes with an open-source variant, which allows for the free and unconstrained distribution of applications build using the said variant.

In the developed E-NAUTILUS interface (see Fig. 2), the decision maker is shown three distinct visualizations (Miettinen, 2014) to present the different candidates computed by E-NAUTILUS. These are: (i) a *spider plot* (Figs. 2 and 3), (ii) a *value path plot* (Fig. 2), and (iii) *tabulated objective values* (Fig. 2). In the spider plot and the tabulated objective values views, the candidates of each iteration are visualized alongside the candidate best reachable values, which is named as candidate best in the interface for simplicity. However, the value paths plot shows only the objective values of the current candidates because visualizing the reachable values in the value paths plot can result in excess visual clutter. The currently selected candidate is always highlighted in red in the value paths. Furthermore, in the spider plot, the decision maker is also able to select which of the candidates he/she wishes to simultaneously view. This can facilitate the comparisons of different candidates.

Each of the three described views is also linked. This means that by selecting one of the candidates shown in an iteration using the radio button seen in Fig. 2, the same candidates are then highlighted in each of the views. Having different visualizations of the same candidates, and linking the visualizations allows the decision maker to easily explore the available information which can aid him/her to learn about the problem (Roberts, 2007). Linking is evident in Fig. 2, where the third candidate has been selected. The same candidate is then automatically shown in the spider plot view, highlighted as a red line in the value path view, and highlighted as the blue rows in the tabulated values view. As the decision maker changes the currently selected candidate, each of the views is updated accordingly in real-time.

Moreover, the candidate chosen in the previous iteration is also shown in the spider plot view. This is not part of the original description of E-NAUTILUS. This feature was the result of a wish presented by the decision maker in the case study discussed in Section 5. By visualizing the previously selected candidate, the decision maker is able to compare the newly computed candidates to the previous candidate and see how each of the objectives has improved. This may also aid the decision maker in exploring and learning about the problem.

As described in Section 2.2, the E-NAUTILUS method also shows closeness information of the candidates to the Pareto optimal front. However, this option was not used in this paper. Instead, the information about the number of iteration left is provided to the decision maker to give an estimation about the closeness of the candidates to the Pareto optimal front.

The source code of the web-based graphical user interface developed for E-NAUTILUS is available as open-source code on GitHub https://github.com/industrial-optimization-group/desdeo-dash. Furthermore, the interface discussed in this section is also available online in https://desdeo.it.jyu.fi/dash.

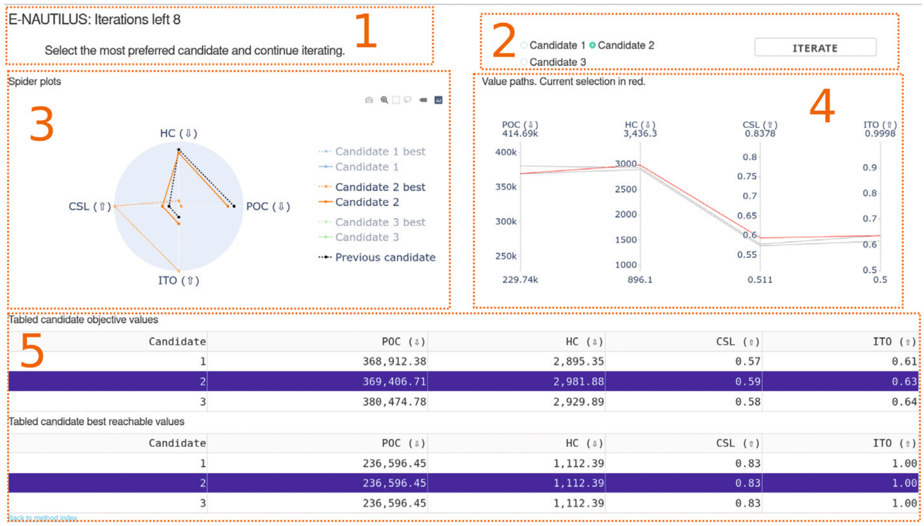

Fig. 2. The main dashboard shown to the decision maker in the E-NAUTILUS interactive. (1) Number of iterations left and short instructions to guide the decision maker. (2) Radio-buttons for selecting a candidate and the ITERATE-button to proceed to the next iteration with the selected candidate. (3) Spider plot view. See Fig. 3 for a more detailed description. (4) Value path view of candidates. (5) Tabulated values view. Top table: the candidates' individual objective values. Bottom table: the best reachable values from each candidate. The highlighted rows show the candidate selected with the radio-buttons shown in (1). The arrows shown next to the objective names (POC, HC, CSL and ITO) across the dashboard indicate whether an objective is to be minimized (down arrow) or maximized (up arrow).

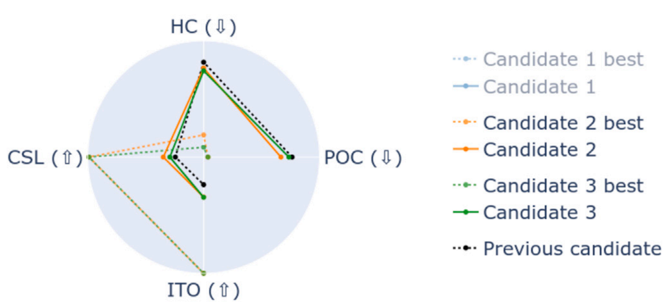

Fig. 3. The spider plot view in the E-NAUTILUS interface. The decision maker is able to select (by clicking on the legend on the right of the plot) one or multiple candidates to be shown simultaneously for comparison. Candidates 2 and 3 have been selected for comparisons in the figure. The best reachable values of each candidate (written as candidate best) are also shown by the dashed line. Also, the candidate selected in the previous iteration is shown (as the black dashed lines in the figure). The names of the objective functions are shown on the outer radius of the plot, where an arrow shows if the objective is to be either minimized (down arrow) or maximized (up arrow). Each of the candidates and their best reachable values can be moused over, which will display detailed numerical information.

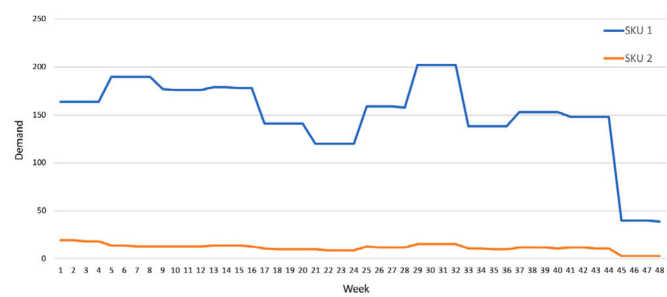

Fig. 4. Demands for SKU 1 (top line) and SKU 2 (bottom line).

5. Computational results

As a proof of concept, in this section, we present the results of solving the proposed model using real data from a manufacturing company. As mentioned in the introduction, our model is particularly suited for problems with various types of constraints and many decision variables and, this case study demonstrates the need of having a dynamic stock to handle demand uncertainty in a better way. After introducing the case study, we demonstrate how a supply chain manager from the company, acting as the decision maker, found the most preferred solution for him using the developed E-NAUTILUS interface.

5.1. Case study

Real data of two different types of SKUs are analyzed: one with high demand (called SKU 1) and another with low demand (called SKU 2). The time period for inventory planning is one week, and we consider lot sizes for 48 weeks. Therefore, the multiobjective optimization model involves 50 decision variables. The data was received from the ERP system of the company.

Based on the data, the price of SKU 1 is €134 which is almost four times less than that of SKU 2 with a price of €483.85, but the demand is on average more than ten times higher than SKU 2 (see Fig. 4). A high volume order must be placed for SKU 1 with a minimum of 70 units and rounding by 14 units for one order, while SKU 2 can be ordered with a minimum of 3 units and the same rounding value.

The case company utilizes a pre-order method with these SKUs. A scheduled order for the supplier is placed one year ahead for separate weekly deliveries. The method consists of a frozen zone and a liquid zone planning times. During the frozen zone, no changes can be done in the pre-ordered amounts, but changes can be made during the liquid zone. Based on this fact, we set the lead time as the frozen zone, which is six weeks for both SKU 1 and SKU 2. The historical data shows that during this six week period, the company has made previous orders (420, 70, 140, 210, 140, 140) for SKU 1 and (6, 9, 9, 9, 12, 6) for SKU 2, with the opening inventory 596 and 75 for SKU 1 and SKU 2, respectively.

After introducing the idea to the decision maker, an additional constraint was defined as a request from him. With this additional constraint, the proposed multiobjective optimization model has five constraints. In this case, the SS and the SOT as the safety strategy must be limited. Without this limitation, the stock level can be significantly high to make a near-perfect CSL, but it makes the holding cost significantly high and the ITO significantly low, which is not reasonable for the decision maker. The decision maker is only interested in a combination of the safety strategy under the following constraint:

$$SOT + \frac{SS}{\mu} \leq MS, \tag{15}$$

where MS is maximum number of periods that can be covered by the safety strategy. We set $MS = 1$ week for SKU 1 and $MS = 1.4$ weeks for SKU 2.

5.2. Pre-processing stage

As previously described, the E-NAUTILUS method starts by generating a large number of Pareto optimal solutions using any a posteriori method. We applied an evolutionary method called NSGA-III (Deb & Jain, 2014) by using the pymoo framework (Blank & Deb, 2020). It has been developed for problems with four or more objectives. We selected an evolutionary method since they do not set requirements on the type of functions involved and can handle integer variables. However, as mentioned in Section 2.2, they cannot guarantee the Pareto optimality of solutions. All we know is that the solutions are nondominated, that is, not dominated by each other. Thus, also the third stage of E-NAUTILUS was needed in the solution process.

Because of the computational cost, it is a challenge to generate a large number of nondominated solutions for the defined lot sizing problem. In addition, integer decision variables and five constraints limit the number of nondominated solutions. Therefore, a single run of NSGA-III could not generate enough nondominated solutions even though the size of the initial population was increased to get more solutions. Naturally, increasing the number of solutions increases the computation time exponentially. To overcome this issue, we generated the solutions iteratively using different sizes for the initial populations and combined the generated solutions by deleting the recurring and dominated solutions. More detailed information can be seen in B. As a result, we obtained 651 nondominated solutions for SKU 1 and 518 nondominated solutions for SKU 2 to be used in the next stage of E-NAUTILUS.

From these nondominated solutions, the ideal and nadir points were calculated to approximate the ranges of the Pareto optimal front. The best-found objective function values were set as the ideal point and the worst values found were set as the nadir point.

5.3. Interactive decision making stage

The novel E-NAUTILUS interface was applied to support the decision maker in solving the two problems involving the two SKUs. As discussed in Section 2.2, the decision maker was shown solution candidates to compare with some additional information and was asked to provide preference information at each iteration. The goal of this stage is to find a nondominated solution that best represents the decision maker's preferences. The step-by-step decision making process for both SKUs is described in detail below.

5.3.1. SKU 1

First of all, the estimated ideal and nadir vectors, as shown in Table 2, were presented to the decision maker. Then, he was asked to provide the number of iterations to be carried, and the number of candidates to be shown in each iteration. He noticed that the Pareto optimal front has a wide range. If he chose the number of iterations too low, the candidates would approach too fast to the final solution and he might lose some of the potentially interesting candidates during the decision making process. Therefore, the decision maker ultimately decided to select ten iterations and four candidates to consider in each iteration.

In each iteration, the decision maker was provided with four candidates and their best reachable values. Using the E-NAUTILUS interface with three types of visualizations, the decision maker could easily compare the candidates before selecting one of the available candidates. In what follows, each iteration is reported, while more detailed information on the candidates, the corresponding reachable values, and the selected candidate for each iteration can be seen in Table A.1 of Appendix A.

Table 2
Ideal and nadir points of SKU 1.

	POC	HC	CSL	ITO
Ideal point	747 820	2 717.24	1.0	252.96
Nadir point	1 046 028	9 133.52	0.5	13.66

Iteration 1. In the first four candidates shown, their reachable values were basically still the whole Pareto optimal front and, thus, taking a step from the estimated nadir point to any of the candidates would not limit the objective values much. The decision maker initially paid more attention to ITO than the other objectives. He decided to select the candidate $\boldsymbol{z}(1) = \boldsymbol{z}(1,1) =$ (1 025 459.60, 8 590.38, 0.50, 22.7) to get the best values of ITO and had a chance to improve on the other objectives.

Iteration 2. The second iteration showed a variation of the reachable values, especially in POC and ITO. The decision maker chose the candidate $\boldsymbol{z}(2) = \boldsymbol{z}(2,2) =$ (1 004 891.20, 8 047.24, 0.51, 31.88). He noticed that it had the worst CSL value, but it was pretty close with the others and he had the best ITO with this choice.

Iteration 3. In this iteration, the decision maker was still interested in pursuing the best ITO value, hence he chose the candidate $\boldsymbol{z}(3) = \boldsymbol{z}(3,3) =$ (987 940.30, 7 518.98, 0.51, 47.2). He realized that his choice had the worst CSL, but in his opinion, the reachable values for this candidate were quite good.

Iteration 4. The decision maker changed the direction to get the better CSL value in this iteration. He decided to select the candidate $\boldsymbol{z}(4) = \boldsymbol{z}(4,2) =$ (967 533.97, 6 920.64, 0.56, 48.57) which had the best CSL. Even though this candidate had the worst ITO, he needed to take care of the CSL.

Iteration 5. The CSL was still the main focus of the decision maker in this iteration. He preferred the candidate $\boldsymbol{z}(5) = \boldsymbol{z}(5,3) =$ (947 127.64, 6 322.29, 0.61, 49.9) to achieve the best value in CSL. He noticed that this candidate had the worst ITO but he was satisfied enough with the ITO values of all candidates.

Iteration 6. In this iteration, the decision maker still paid more attention to the CSL value, because he was satisfied with the current ITO value. The candidate he liked most in this iteration was $\boldsymbol{z}(6) = \boldsymbol{z}(6,3) =$ (923 009.31, 5 709.51, 0.67, 50.29) which had the best CSL value.

Iteration 7. With the same considerations as in the previous iteration, in this iteration the decision maker's selected candidate was $\boldsymbol{z}(7) = \boldsymbol{z}(7,3) =$ (898 890.99, 5 096.73, 0.73, 50.6) which had the best CSL value.

Iteration 8. This iteration became more interesting to the decision maker because the reachable values of CSL and ITO were exactly the same for all candidates. After considering the candidates, he preferred to select the candidate $\boldsymbol{z}(8) = \boldsymbol{z}(8,2) =$ (871 379.32, 4 486.96, 0.78, 52.1) due to the best CSL and pretty good ITO values.

Iteration 9. Among the candidates shown in this iteration, the decision maker liked most the candidate $\boldsymbol{z}(9,2)$ which had the best CSL value. Then, we set $\boldsymbol{z}(9) = \boldsymbol{z}(9,2) =$ (843 867.66, 3 877.18, 0.83, 53.6) as the selected candidate of this iteration.

Iteration 10. Finally, in the last iteration, the decision maker considered both the cost values in his choice, because he was satisfied with CSL and ITO values. He selected the candidate $\boldsymbol{z}(10) = \boldsymbol{z}(10,4) =$ (810 528, 3 355.80, 0.90, 54.48) to get the best POC. The HC value was the worst in this candidate but it was pretty close to the other candidates.

Table 3
Ideal and nadir points of SKU 2.

	POC	HC	CSL	ITO
Ideal point	220 829.40	770.44	1.0	107.02
Nadir point	428 949.50	4 284.86	0.5	11.40

5.3.2. SKU 2

As mentioned, the interactive decision making process for SKU 2 was started by presenting the ideal and nadir vectors, shown in Table 3, to the decision maker. He observed that the ideal and nadir points for the SKU 2 were generally lower than for SKU 1, except for the CSL.

The decision maker made the same choice of four candidates and ten iterations for this SKU for the same reason as for SKU 1. The details of the decision making process are described below, and all of the information provided to the decision maker for this SKU is presented in Table A.2 of Appendix A.

The decision maker applied a different strategy for SKU 2. He considered both CSL and ITO values and selected the best balance between these values from the beginning until the third iteration. In the first iteration, he selected $z(1) = z(1,4) =$ (410 144.51, 3 959.10, 0.54, 14.15) which did not have the best CSL and ITO values but was sufficiently good compared to the others. In the second iteration, out of the four candidates, he compared $z(2,2)$ and $z(2,4)$ which had the best CSL and chose $z(2) = z(2,4) =$ (393 180.29, 3 661.26, 0.59, 15.93) to obtain the better ITO. For the next iteration, he was interested in $z(3,1)$ and $z(3,4)$ and he preferred $z(3) = z(3,4) =$ (374 301.64, 3 333.05, 0.63, 19.39) which had pretty good ITO and CSL values in his point of view.

The CSL was the main consideration for the decision maker in the fourth and fifth iteration. He liked most the candidate $z(4,4) =$ (362 466.02, 3 065.26, 0.69, 20.15) due to the best CSL among all of the candidates. He realized that this candidate had the worst ITO value, but the same objective values for ITO can be reached from all candidates. Next, the candidates $z(5,2)$ and $z(5,4)$ attracted his attention due to the best CSL values. He then decided to select $z(5) = z(5,4) =$ (342 446.86, 2 766.30, 0.73, 22.10) to get the better ITO value.

The decision maker changed the direction by considering ITO values in the sixth iteration. He chose the candidate $z(6) = z(6,4) =$ (326 281.74, 2 416.54, 0.76, 27.01) in order to achieve the best ITO value. He realized that the CSL value of this candidate was not the best, but the difference was not significant. After this iteration, the ITO value seemed acceptable for the decision maker in all of the candidates, and he was more interested in directing the search towards solutions that require the highest CSL in the next three iterations. Hence, he decided to continue with the candidate $z(7) = z(7,3) =$ (314 984.16, 2 117.72, 0.82, 27.79).

In the next iteration, the candidates $z(8,1)$ and $z(8,3)$ had the highest CSL, therefore he selected the candidate $z(8) = z(8,3) =$ (300 516.80, 1 826.81, 0.87, 29.53) to get a better ITO. Then, he selected the candidate $z(9) = z(9,4) =$ (277 440.15, 1 466.81, 0.94, 31.11) in the ninth iteration. Besides the CSL, this candidate had a reasonable value for holding cost and ITO for him. Finally, in the last iteration, the decision maker was very happy for the improvement of all the candidates. He looked at all of the solutions, which had good values, especially in CSL and ITO. Then, he decided to select the candidate $z(10) = z(10,3) =$ (276 936.75, 1 074.71, 0.99, 34.33).

5.4. Post-processing stage

As described in Section 2.2, the post-processing stage is needed to assure the Pareto optimality of the final solution if an evolutionary algorithm is used in the first stage. In this stage, we used the preferred candidate of the interactive decision making stage $z(10)$ as a reference point and project it onto the Pareto optimal front to get the final solution z_{final}. The corresponding optimization problem was solved by using a branch and bound method (Land & Doig, 1960), which is commonly used for solving optimization problems with integer variables.

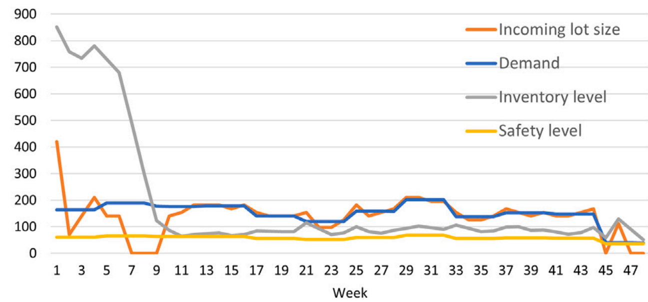

Fig. 5. Result for SKU 1.

We had $z(10) =$ (810 528, 3 355.80, 0.90, 54.48) as the reference point for SKU 1, and the final solution improved to $z_{final} =$ (753 848, 2 329.41, 0.924, 89.18). The lot sizes corresponding to z_{final} can be seen in Fig. 5. The other decision variables were $SS = 28$, $SOT = 1$ day. In the figure, the orange line represents incoming lot sizes for each week, which is $q(t - L)$ for $t = 7, \dots, 48$ and the previous order data for $t = 1, \dots, 6$. The demand data is illustrated by the blue line for comparison. We also provide the inventory level and the safety level in the gray and the yellow lines, respectively, to show that the inventory level is larger than the safety level for every week. It indicates that, by using the final solution obtained by applying E-NAUTILUS, the company always had SKUs to cover unpredicted demand at least the same amount as the safety level.

Fig. 5 shows that the company could improve inventory management with the final solution obtained. Before using the proposed optimization model, the company had excess inventory at the beginning of the period. The inventory level could not be controlled by the model before week seven because of the lead time. By using the final solution, zero orders were set for the first three weeks, which can be seen in the incoming lot sizes for weeks seven to nine in the figure. Because the decision maker was more interested in ITO than the other objectives for this SKU, after that period, the final solution suggested to order in similar amounts as the demand data. With this strategy, the company will have the possibility to balance between the inventory planning conflicts, namely meeting the unpredicted demand and keeping the inventory value controlled. At the end of the period, one can see a decrease in the demand. In this situation, buying SKUs in similar amounts as demand did not meet the minimum order quantity and would increase the ordering cost. Therefore, in the final solution, the company was suggested to order more SKUs in week 44 so that no order in week 45 was needed. Then, the company should order more SKUs in week 46 to satisfy demand until the end of the period considered.

For SKU 2, the reference point was $z(10) =$ (276 936.75, 1 074.71, 0.99, 34.33), and the final solution improved by the projection to $z_{final} =$ (225 332.50, 722.98, 0.997, 54.12). The corresponding lot sizes can be seen in Fig. 6. The other decision variables were SS= 3, SOT= 3 days. For this SKU, the decision maker was more interested in CSL than the other objectives, which made the safety level higher and almost similar to the inventory level and the demand. As in the case of SKU 1, the company had excess inventory at the beginning of the period considered, and because of the lead time, the effects of the final solution can be only seen after week eight. The decision maker was then more interested in ITO values than both of the cost objectives. Therefore, in the final solution, the lot sizes were in similar amounts as the demand until week 44. At the end of the period, for the similar reason as for SKU 1, no order in weeks 45, 47 and 48 was needed because the demand for these weeks had been satisfied by the previous order. Thanks to the minimum order quantity.

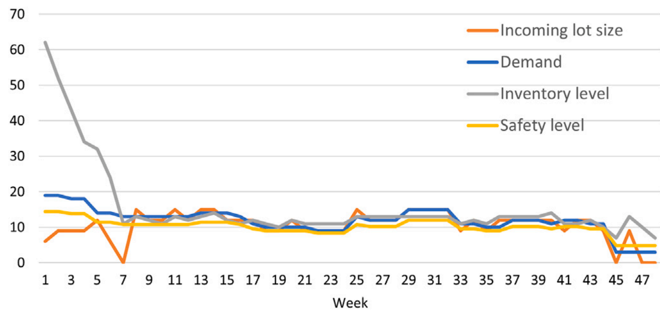

Fig. 6. Result for SKU 2.

The presented results showed that multiobjective optimization is a valuable tool for solving the integrated lot sizing problem with safety strategy placement. We managed to find a final solution for each SKU which was confirmed to be the most preferred solution by the decision maker. The decision maker was very happy with the interactive E-NAUTILUS method, which helped him in making a good decision that reflected his preferences well. He realized that an improvement can be made in his inventory management system by implementing our proposed multiobjective optimization model and solving it with an appropriate method. He appreciated the fact that the method enabled him to think of improvements in objectives rather than focusing on trade-offs.

In particular, the decision maker highlighted the usefulness of SOT for inventory control. SOT supports measuring the success of his day-to-day operations because it responds faster than SS. The usefulness of SOT is particularly pronounced in an industrial environment, where demand fluctuates rapidly. More generally, SOT provides a quick way to assess the relevance of inventory control and, thus, serves the needs of the management well.

6. Conclusions

In this paper, we developed a multiobjective optimization model to solve a single-item multi-period lot sizing problem in periodic review policy under stochastic environment on demand. We proposed the concept of SOT which can handle high fluctuation of demand better than SS. The combination of SS and SOT increased the preparedness of handling demand uncertainty. We then proposed a multiobjective optimization model with four objectives and four constraints to solve this problem. By using the proposed model, we determined the optimal order quantity in each period and simultaneously decided the optimal values of SS and SOT.

As a proof of concept, two SKUs, one with high demand and another with low demand, were studied with real data from a manufacturing company to demonstrate the performance and applicability of the proposed model. Even though interactive methods have many desirable properties, they have not been applied widely in lot sizing. For the first time in this field, we used the trade-off-free interactive E-NAUTILUS method, designed for solving computationally expensive problems. A novel web-based graphical user interface was developed in this research to help the decision maker in finding his most preferred solution using the E-NAUTILUS method. By applying this method, the decision maker could avoid thinking of sacrifices and trade-offs as most other multiobjective optimization methods would have necessitated. The decision maker provided different preferences for the two SKUs, and was satisfied with both results.

The decision maker, who was a supply chain manager of the company, found the model and SOT useful in his daily operations. He greatly appreciated SOT that efficiently handles dynamic stock to manage the demand stochasticity. He also appreciated the proposed model and the interactive E-NAUTILUS method, as well as the user interface, that allowed him to consider POC, HC, CSL and ITO simultaneously without having to trade-off among the objectives. He was pleased with the objective function values and the corresponding order quantities, SS and SOT. He found the model, the interactive solution process and the results useful and was willing to adopt them more widely for inventory planning and control in his company. This demonstrates the strengths of the model and the interactive method applied.

Some assumptions have been made in this research: no capacity limit and no backorder cost. Including them in the model is a future research direction to extend its applicability. Moreover, the number of SKUs to be considered in this research is limited since the decision maker needs to repeat the interactive solution process for each SKU. Considering many SKUs is a further possibility to extend this work. In addition, considering additional uncertainties in the model, such as lead time uncertainty, is another future direction. It would make the model more realistic, but computationally more demanding.

CRediT authorship contribution statement

Adhe Kania: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Juha Sipilä:** Validation, Data curation, Writing – review & editing. **Giovanni Misitano:** Software, Writing – original draft. **Kaisa Miettinen:** Conceptualization, Writing – review & editing, Supervision. **Jussi Lehtimäki:** Conceptualization, Resources, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This research was partly funded by LPDP, the Indonesian Endowment Fund for Education (grant number S-5302/LPDP.4/2020) and the Academy of Finland (grant number 322221). The research is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyväskylä. We thank Dr. Bekir Afsar for his valuable suggestions in polishing this paper.

Appendix A. Detailed steps of interactive decision making stage

The detailed steps of the decision making process in every iteration of SKU 1 and SKU 2 can be seen in Tables A.1 and A.2, respectively. In each iteration (h), four candidates were shown to the decision maker, together with the best reachable values from each candidate. The decision maker then selected one candidate among them, which is shown in bold face, to proceed with in the next iteration.

Appendix B. Details of the pre-processing stage

The pre-processing stage is the most time-consuming part in applying the E-NAUTILUS method. As mentioned in Section 5.2, the NSGA-III method was used to generate nondominated solutions in this stage. Because of the challenge of generating a sufficient number of nondominated solutions, we needed to rerun the method for several times with different sizes of the initial population. We used the structured approach described in Das and Dennis (1998) with the number of partitions from 1 until 20. The reason of having different sizes of initial

Table A.1
Interactive decision making stage of SKU 1.

h	Candidates	Best reachable values			
		POC	HC	CSL	ITO
1	$z(1,1)$=**(1 025 459.60, 8 590.38, 0.50, 22.77)**	**747 820**	**2 717.24**	**1.00**	**252.82**
	$z(1,2)$=(1 024 809.20, 8 598.50, 0.55, 16.64)	747 820	2 717.24	1.00	124.94
	$z(1,3)$=(1 020 226.80, 8 575.23, 0.53, 18)	747 820	2 717.24	1.00	148.42
	$z(1,4)$=(1 018,290.80, 8,754.71, 0.55, 14.86)	749 296	2 717.24	1.00	124.94
2	$z(2,1)$=(997 781.87, 8 157.48, 0.55, 23.95)	749 296	2 717.24	1.00	124.94
	$z(2,2)$=**(1 004 891.20, 8 047.24, 0.51, 31.88)**	**748 420**	**2 717.24**	**1.00**	**252.82**
	$z(2,3)$=(1 007 384.09, 8 046.84, 0.55, 25.47)	749 296	2 717.24	1.00	124.94
	$z(2,4)$=(999 560.53, 8 014.37, 0.53, 26.89)	747 820	2 717.24	1.00	148.42
3	$z(3,1)$=(981 768.30, 7 503.88, 0.56, 33.05)	749 496	2 820.06	1.00	113.45
	$z(3,2)$=(979 238.80, 7 470.28, 0.54, 34.83)	748 420	2 717.24	1.00	124.94
	$z(3,3)$=**(987 940.30, 7 518.98, 0.51, 47.24)**	**748 420**	**2 717.24**	**0.96**	**252.82**
	$z(3,4)$=(980 461.30, 7 418.42, 0.51, 36.92)	748 420	2 717.24	1.00	148.42
4	$z(4,1)$=(962 441.97, 6 875.80, 0.52, 50.80)	748 420	2 717.24	0.93	148.42
	$z(4,2)$=**(967 533.97, 6 920.64, 0.56, 48.57)**	**754 724**	**2 717.24**	**0.93**	**124.94**
	$z(4,3)$=(961 580.83, 6 907.24, 0.54, 49.28)	748 420	2 717.24	0.93	148.42
	$z(4,4)$=(969 563.69, 7 069.32, 0.51, 63.39)	766 380	2 717.24	0.82	252.82
5	$z(5,1)$=(938 339.64, 6 334.91, 0.59, 50.45)	754 724	2 820.06	0.93	110.70
	$z(5,2)$=(942 058.31, 6 237.21, 0.57, 54.54)	763 228	2 820.06	0.90	113.45
	$z(5,3)$=**(947 127.64, 6 322.29, 0.61, 49.90)**	**754 724**	**2 820.06**	**0.93**	**97.01**
	$z(5,4)$=(951 258.98, 6 287.11, 0.57, 55.56)	763 228	2 820.06	0.90	113.45
6	$z(6,1)$=(913 349.31, 5 744.86, 0.64, 51.17)	754 724	2 890.41	0.90	83.81
	$z(6,2)$=(918 386.91, 5 727.19, 0.62, 53.02)	754 724	2 820.06	0.90	90.54
	$z(6,3)$=**(923 009.31, 5 709.51, 0.67, 50.29)**	**754 724**	**2 890.41**	**0.93**	**77.96**
	$z(6,4)$=(926 266.11, 5 728.63, 0.65, 52.51)	754 724	2 890.41	0.90	77.96
7	$z(7,1)$=(902 961.99, 5 120.63, 0.70, 53.46)	754 724	2 944.52	0.90	77.96
	$z(7,2)$=(886 815.99, 5 140.92, 0.69, 51.78)	754 724	2 944.52	0.90	77.96
	$z(7,3)$=**(898 890.99, 5 096.73, 0.73, 50.69)**	**754 724**	**2 991.42**	**0.91**	**68.68**
	$z(7,4)$=(893 481.99, 5 082.30, 0.68, 52.89)	754 724	2 890.41	0.90	77.96
8	$z(8,1)$=(850 835.32, 4 596.99, 0.73, 51.93)	754 724	3 088.83	0.90	67.98
	$z(8,2)$=**(871 379.32, 4 486.96, 0.78, 52.16)**	**778 636**	**3 101.46**	**0.90**	**67.98**
	$z(8,3)$=(880 200.66, 4 515.82, 0.76, 54.78)	788 016	3 088.83	0.90	67.98
	$z(8,4)$=(867 760.66, 4 532.05, 0.74, 53.29)	754 724	3 088.83	0.90	67.98
9	$z(9,1)$=(857 937.66, 3 801.42, 0.80, 60.07)	842 820	3 115.89	0.82	67.98
	$z(9,2)$=**(843 867.66, 3 877.18, 0.83, 53.62)**	**810 528**	**3 119.49**	**0.90**	**59.87**
	$z(9,3)$=(832 411.66, 3 995.33, 0.78, 52.17)	793 444	3 130.32	0.90	59.87
	$z(9,4)$=(850 533.66, 4 047.65, 0.81, 56.01)	818 232	3 115.89	0.87	67.98
10	$z(10,1)$=(833 440, 3 119.49, 0.90, 55.85)				
	$z(10,2)$=(818 232, 3 310.70, 0.84, 57.79)				
	$z(10,3)$=(816 356, 3 267.41, 0.88, 55.08)				
	$z(10,4)$=**(810 528, 3 355.80, 0.90, 54.48)**				

Table A.2
Interactive decision making stage of SKU 2.

h	Candidates	Best reachable values			
		POC	HC	CSL	ITO
1	$z(1,1)$=(409 483.89, 3 968.73, 0.51, 15.60)	220 829.40	770.44	1.00	91.65
	$z(1,2)$=(410 619.97, 4 022.61, 0.55, 12.53)	220 829.40	770.44	1.00	91.65
	$z(1,3)$=(414 679.15, 3 972.64, 0.50, 18.23)	220 829.40	770.44	1.00	107.02
	$z(1,4)$=**(410 144.51, 3 959.10, 0.54, 14.15)**	**220 829.40**	**770.44**	**1.00**	**91.65**
2	$z(2,1)$=(393 319.35, 3 658.46, 0.55, 19.08)	220 829.40	770.44	1.00	91.65
	$z(2,2)$=(389 615.57, 3 752.13, 0.59, 14.92)	220 829.40	770.44	1.00	91.65
	$z(2,3)$=(391 339.51, 3 638.77, 0.58, 17.20)	220 829.40	770.44	1.00	91.65
	$z(2,4)$=**(393 180.29, 3 661.26, 0.59, 15.93)**	**220 829.40**	**770.44**	**1.00**	**91.65**
3	$z(3,1)$=(375 340.31, 3 388.36, 0.64, 17.08)	221 680.95	866.74	1.00	60.95
	$z(3,2)$=(376 916.86, 3 355.38, 0.59, 22.04)	220 829.40	770.44	1.00	91.65
	$z(3,3)$=(372 305.76, 3 391.15, 0.59, 19.53)	220 829.40	770.44	1.00	91.65
	$z(3,4)$=**(374 301.64, 3 333.05, 0.63, 19.39)**	**221 680.95**	**866.74**	**1.00**	**60.95**
4	$z(4,1)$=(355 244.20, 3 010.63, 0.66, 22.56)	221 680.95	866.74	1.00	60.95
	$z(4,2)$=(354 829.47, 3 045.72, 0.68, 20.71)	221 680.95	866.74	1.00	60.95
	$z(4,3)$=(357 053.34, 2 994.08, 0.64, 23.94)	221 680.95	866.74	1.00	60.95
	$z(4,4)$=**(362 466.02, 3 065.26, 0.69, 20.15)**	**221 680.95**	**866.74**	**1.00**	**60.95**
5	$z(5,1)$=(342 204.94, 2 733.74, 0.71, 23.72)	221 680.95	866.74	1.00	54.55
	$z(5,2)$=(349 562.69, 2 781.89, 0.73, 21.20)	221 680.95	866.74	1.00	54.55
	$z(5,3)$=(344 315.60, 2 714.43, 0.69, 25.34)	221 680.95	866.74	1.00	60.95
	$z(5,4)$=**(342 446.86, 2 766.30, 0.73, 22.10)**	**221 680.95**	**866.74**	**1.00**	**54.55**
6	$z(6,1)$=(330 966.70, 2 486.05, 0.78, 22.98)	221 680.95	866.74	1.00	54.55
	$z(6,2)$=(318 783.99, 2 453.66, 0.76, 24.09)	221 680.95	866.74	1.00	54.55
	$z(6,3)$=(322 427.71, 2 467.34, 0.78, 24.06)	221 680.95	866.74	1.00	54.55
	$z(6,4)$=**(326 281.74, 2 416.54, 0.76, 27.01)**	**221 680.95**	**866.74**	**1.00**	**54.55**
7	$z(7,1)$=(304 210.42, 2 107.25, 0.81, 28.16)	223 732.50	866.74	1.00	50.42
	$z(7,2)$=(304 936.19, 2 081.43, 0.78, 30.66)	223 932.50	866.74	1.00	54.55
	$z(7,3)$=**(314 984.16, 2 117.72, 0.82, 27.79)**	**223 732.50**	**866.74**	**1.00**	**50.42**
	$z(7,4)$=(312 243.94, 2 068.87, 0.77, 32.60)	223 932.50	866.74	1.00	54.55
8	$z(8,1)$=(289 121.59, 1 805.41, 0.87, 28.94)	223 732.50	971.42	1.00	42.53
	$z(8,2)$=(290 706.47, 1 757.02, 0.83, 33.06)	223 932.50	866.74	0.99	50.42
	$z(8,3)$=**(300 516.80, 1 826.81, 0.87, 29.53)**	**223 732.50**	**971.42**	**1.00**	**42.53**
	$z(8,4)$=(286 502.34, 1 812.85, 0.85, 30.21)	223 932.50	866.74	1.00	50.42
9	$z(9,1)$=(277 440.15, 1 427.73, 0.92, 34.48)	230 738.70	996.54	0.98	42.53
	$z(9,2)$=(265 027.75, 1 514.96, 0.89, 32.29)	223 932.50	971.42	1.00	42.53
	$z(9,3)$=(269 256.63, 1 503.10, 0.92, 30.57)	228 687.15	996.54	1.00	42.53
	$z(9,4)$=**(277 440.15, 1 466.81, 0.94, 31.11)**	**230 138.70**	**1 007.71**	**1.00**	**39.44**
10	$z(10,1)$=(238 196.45, 1 099.83, 0.97, 35.27)				
	$z(10,2)$=(254 363.50, 1 028.65, 0.97, 39.44)				
	$z(10,3)$=**(276 936.75, 1 074.71, 0.99, 34.33)**				
	$z(10,4)$=(252 911.95, 1 211.49, 0.99, 30.24)				

populations is to get more different solutions (Deb & Jain, 2014). We then combined all of the generated solutions and deleted the dominated ones.

In NSGA-III, we used simulated binary crossover for integer variables with crossover probability 0.9 and polynomial mutation for integer variables with mutation probabilities 0.9. We found that these parameters are good enough for our needs after several experiments. More detailed information related to these operators can be seen in Deb, Sindhya, and Okabe (2007). For other parameters, we used the default values in pymoo (Blank & Deb, 2020).

References

Açıkgöz, N., Çağıl, G., & Uyaroğlu, Y. (2020). The experimental analysis on safety stock effect of chaotic supply chain attractor. *Computers & Industrial Engineering*, *150*, Article 106881.

Agrell, P. J. (1995). A multicriteria framework for inventory control. *International Journal of Production Economics*, *41*(1), 59–70, Proceedings of the 12th International Conference on Production Research.

Alem, D., Curcio, E., Amorim, P., & Almada-Lobo, B. (2018). A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers & Operations Research*, *90*, 125–141.

Alfares, H., & Ghaithan, A. (2019). EOQ and EPQ production-inventory models with variable holding cost: State-of-the-art review. *Arabian Journal for Science and Engineering*, *44*, 1737–1755.

Amorim, P., Antunes, C., & Almada-Lobo, B. (2011). Multi-objective lot-sizing and scheduling dealing with perishability issues. *Industrial and Engineering Chemistry Research*, *50*(6), 3371–3381.

Andriolo, A., Battini, D., Grubbström, R., Persona, A., & Sgarbossa, F. (2014). A century of evolution from Harris's basic lot size model: Survey and research agenda. *International Journal of Production Economics*, *155*, 16–38.

Aslam, T., & Amos, H. (2010). Multi-objective optimization for supply chain management: A literature review and new development. In *8th International conference on supply chain management and information* (pp. 1–8). IEEE.

Azadnia, A., Saman, M., & Wong, K. (2015). Sustainable supplier selection and order lot-sizing: An integrated multi-objective decision-making process. *International Journal of Productions Research*, *53*(2), 383–408.

Bahl, H., Ritzman, L., & Gupta, J. (1987). Determining lot sizes and resource requirements: A review. *Operations Research*, *35*(3), 329–345.

Benayoun, R., de Montgolfier, J., Tergny, J., & Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, *1*, 366–375.

Bitran, G., & Yanasse, H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, *28*(10), 1174–1186.

Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, *8*, 89497–89509.

Bouchery, Y., Ghaffari, A., Jemai, Z., & Dallery, Y. (2012). Including sustainability criteria into inventory models. *European Journal of Operational Research*, *222*(2), 229–240.

Buchanan, J., & Corner, J. (1997). The effects of anchoring in interactive MCDM solution methods. *Computers & Operations Research*, *24*(10), 907–918.

Chan, F., & Chan, H. (2006). A simulation study with quantity flexibility in a supply chain subjected to uncertainties. *International Journal of Computer Integrated Manufacturing*, *19*(2), 148–160.

Chopra, S., & Meindl, P. (2016). *Supply chain management: strategy, planning, and operation* (6). Pearson.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, *8*(3), 631–657.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*(4), 577–601.

Deb, K., Sindhya, K., & Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the 9th Annual conference on genetic and evolutionary computation* (pp. 1187–1194). ACM.

Efthymiou, K., Mourtzis, D., Pagoropoulos, A., Papakostas, N., & Chryssolouris, G. (2016). Manufacturing systems complexity analysis methods review. *International Journal of Computer Integrated Manufacturing*, *29*(9), 1025–1044.

Glock, C., Grosse, E., & Ries, J. (2014). The lot sizing problem: A tertiary study. *International Journal of Production Economics*, *155*, 39–51.

Grant, D., Lambert, D., Stock, J., & Ellram, L. (2006). *Fundamentals of logistics management*. The McGraw-Hill Companies.

Graves, S. (1988). Safety stocks in manufacturing systems. *Journal of Manufacturing and Operations Management*, *1*, 67–101.

Guide, V., & Srivastava, R. (2000). A review of techniques for buffering against uncertainty with MRP systems. *Production Planning & Control*, *11*(3), 223–233.

Haimes, Y., Lasdon, L., & Wismer, D. (1971). On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-1*(3), 296–297.

Harris, F. (1913). How many parts to make at once. *Factory, the Magazine of Management*, *10*, 135–136.

Heikkinen, R., Sipilä, J., Ojalehto, V., & Miettinen, K. (2021). Flexible data driven inventory management with interactive multiobjective lot size optimization. *International Journal of Logistics Systems and Management*, http://dx.doi.org/10.1504/IJLSM.2021.10042370, in press.

Inderfurth, K., & Vogelgesang, S. (2013). Concepts for safety stock determination under stochastic demand and different types of random production yield. *European Journal of Operational Research*, *224*(2), 293–301.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–291.

Kumar, K., & Aouam, T. (2018). Integrated lot sizing and safety stock placement in a network of production facilities. *International Journal of Production Economics*, *195*, 74–95.

Land, A., & Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica*, *28*(3), 497–520.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, *36*(1), 3–37.

Miettinen, K., & Hakanen, J. (2009). Why use interactive multi-objective optimization in chemical process design? In G. Rangaiah (Ed.), *Multi-objective optimization: techniques and applications in chemical engineering* (pp. 153–188). World Scientific.

Miettinen, K., Hakanen, J., & Podkopaev, D. (2016). Interactive nonlinear multiobjective optimization methods. In S. Greco, M. Ehrgott, & J. Figueira (Eds.), *Multiple criteria decision analysis: state of the art surveys* (2). (pp. 931–980). Springer, New York.

Miettinen, K., & Ruiz, F. (2016). NAUTILUS framework: Towards trade-off-free interaction in multiobjective optimization. *Journal of Business Economics*, *86*(1), 5–21.

Miettinen, K., Ruiz, F., & Wierzbicki, A. (2008). Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective optimization: interactive and evolutionary approaches, vol. 5252* (pp. 27–57). Springer, Berlin, Heidelberg.

New, C. (1975). Safety stocks for requirements planning. *Production and Inventory Management*, *12*, 1–18.

Ojalehto, V., & Miettinen, K. (2019). DESDEO: An open framework for interactive multiobjective optimization. In S. Huber, M. Geiger, & A. de Almeida (Eds.), *Multiple criteria decision making and aiding: cases on models and methods with computer implementations* (pp. 67–94). Springer.

Rafiei, R., Nourelfath, M., Gaudreault, J., De Santa-Eulalia, L., & Bouchard, M. (2015). Dynamic safety stock in co-production demand-driven wood remanufacturing mills: A case study. *International Journal of Production Economics*, *165*, 90–99.

Rashid, R., Bozorgi-Amiri, A., & Seyedhoseini, S. (2015). Developing a new stochastic competitive model regarding inventory and price. *Journal of Industrial Engineering International*, *11*(3), 309–318.

Rezaei, J., & Davoodi, M. (2011). Multi-objective models for lot-sizing with supplier selection. *International Journal of Production Economics*, *130*(1), 77–86.

Roberts, J. (2007). State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth international conference on coordinated and multiple views in exploratory visualization (CMV 2007)* (pp. 61–71). IEEE.

Ruiz, A., Sindhya, K., Miettinen, K., Ruiz, F., & Luque, M. (2015). E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method. *European Journal of Operational Research*, *246*, 218–231.

Sawaya, W., & Giauque, W. (1986). *Production and Operations Management*. Harcourt Brace Jovanovich.

Silver, E., Pyke, D., & Thomas, D. (2017). *Inventory and production management in supply chains* (4). CRC Press.

Talluri, S., Cetin, K., & Gardner, A. (2004). Integrating demand and supply variability into safety stock evaluations. *International Journal of Physical Distribution and Logistics Management*, *34*(1), 62–69.

Tavaghof-Gigloo, D., & Minner, S. (2021). Planning approaches for stochastic capacitated lot-sizing with service level constraints. *International Journal of Productions Research*, *59*(17), 5087–5107.

Teunter, R., Syntetos, A., & Babai, M. (2017). Stock keeping unit fill rate specification. *European Journal of Operational Research*, *259*(3), 917–925.

Ustun, O., & Demirtas, E. (2008). An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection. *Omega*, *36*(4), 509–521.

Wagner, H., & Whitin, T. (1958). Dynamic version of the economic lot size model. *Management Science*, *5*, 89–96.

Wierzbicki, A. (1980). The use of reference objectives in multiobjective optimization. In G. Fandel, & T. Gal (Eds.), *Lecture Notes in Economics and Mathematical Systems: vol. 177*, *Multiple Criteria Decision Making Theory and Application* (pp. 468–486). Springer, Berlin, Heidelberg.

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

Zipkin, P. (2000). *Foundation of inventory management*. McGraw-Hill Higher Education.